Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Uso de controlador SDN ONOS para garantia de tráfego em ambiente multirotas

Filipe Fuscaldi de Assis Theodoro

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:

Erik de Britto e Silva

COORIENTAÇÃO:

Marlon Paolo Lima

Julho, 2018

João Monlevade–MG

Filipe Fuscaldi de Assis Theodoro

Uso de controlador SDN ONOS para garantia de tráfego em ambiente multirotas

Orientador: Erik de Britto e Silva

Coorientador: Marlon Paolo Lima

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina "Trabalho de Conclusão de Curso II".

Universidade Federal de Ouro Preto

João Monlevade

Julho de 2018

T388u Theodoro, Filipe Fuscaldi de Assis.
Uso de controlador SDN ONOS para garantia de tráfego em ambiente multirotas [manuscrito] / Filipe Fuscaldi de Assis Theodoro. - 2018.

36f.: il.: color; grafs; tabs.

Orientador: Prof. MSc. Erik de Britto Silva.
Coorientador: Prof. Dr. Marlon Paolo Lima.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Sistemas de informação. 2. Controladores programáveis. 3. Sistema autônomo. I. Silva, Erik de Britto. II. Lima, Marlon Paolo. III. Universidade Federal de Ouro Preto. IV. Titulo.

CDU: 004.72:681.5

Catalogação: ficha.sisbin@ufop.edu.br

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DE ENGENHARIA DA COMPUTAÇÃO

ATA DE DEFESA

Aos 13 dias do mês de Julho de 2018, às 18 horas e 00 minutos, na sala C 304 do ICEA, foi realizada a defesa de Monografia pelo aluno Filipe Fuscaldi de Assis Theodoro, sendo a Comissão Examinadora constituída pelos professores: Prof. MSc. Erik de Britto e Silva, Prof. MSc. Marlon Paolo Lima, Prof. MSc. Theo da Silva Lins, Prof. MSc. Bruno Cerqueira Hott e MSc. Vinicius Fonseca e Silva. O candidato apresentou a monografia intitulada: *"USO DO CONTROLADOR SDN ONOS PARA GARANTIA DE TRÁFEGO EM AMBIENTE MULTIROTAS"*. A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, concedendo-lhe o prazo de 15 dias para incorporação no texto final das alterações sugeridas. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo formando.

João Monlevade, 13 de Julho de 2018.

Prof. MSc. Erik de Britto e Silva
Professor Orientador/Presidente

Prof. MSc. Marlon Paolo Lima
Professor Coorientador

Prof. MSc. Theo Silva Lins
Professor Convidado

Prof. MSc. Bruno Cerqueira Hott
Professor Convidado

MSc. Vinicius Fonseca e Silva
Pesquisador Convidado

Filipe Fuscaldi de Assis Theodoro
Formando

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO

ICEA

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

USO DE CONTROLADOR SDN ONO PARA GARANTIA DE TRÁFEGO EM AMBIENTE MULTIROTAS

Filipe Fuscaldi de Assis Theodoro

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI496 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia de Computação e aprovada pela Banca Examinadora abaixo assinada:

Prof. MSc. Erik de Britto e Silva
DECSI – UFOP
Professor Orientador

Prof. MSc. Marlon de Paolo Lima
DECSI – UFOP
Professor Coorientador

Prof. MSc. Theo Silva Lins
DECSI – UFOP
Professor Convidado

Prof. MSc. Bruno Cerqueira Hott
DECSI – UFOP
Professor Convidado

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO

ICEA

Vinicius Fonseca e Silva

MSc.Vinicius Fonseca e Silva
Pesquisador Convidado

João Monlevade, 13 de Julho de 2018

Rua Trinta e Seis, 115 – Bairro Loanda – CEP 35931-008 – João Monlevade – MG – Brasil
http://www.icea.ufop.br – coec@icea.ufop.br – **(31) 3852-8709**

Agradecimentos

A todos que estiveram presentes nessa caminhada eu só tenho a agradecer. Vocês fazer parte dessa vitoria que foi alcançada com muita luta e muito esforço. Acreditem que jamais irei me esquecer que esta felicidade só foi possível porque estive rodeado das melhores pessoas.

Agradeço a todos os professores que tive, pelas lições ensinadas e pelas oportunidades que me foram dadas. Aos meus companheiros e amigos, agradeço pelas vivencias, pelos momentos de descontração e por tudo que aprendemos e superamos juntos. Cada um de vocês tiveram papel fundamental no meu crescimento e serei eternamente grato a todos por tudo.

Agradeço principalmente a minha família. Pelo carinho, pelo acolhimento, pelos ensinamentos e por me guiarem para ser quem sou hoje, sem vocês eu nada seria.

"Em algum lugar, alguma coisa incrível está esperando para ser descoberta."

— Carl Sagan

Resumo

A popularização da Internet, causada pelo barateamento e avanço nas tecnologias de rede, fez surgir na sociedade diversas aplicações que são amplamente utilizadas, e como alguns desses serviços são altamente dependentes do bom funcionamento da rede, é importante garantir que em caso de falha em algum enlace, uma rota alternativa fosse gerada no menor tempo possível. Com isso, o objetivo desse trabalho é avaliar e analisar experimentalmente a ordem de grandeza do tempo, em que o controlador SDN ONOS demora para gerar uma nova rota em um ambiente multirotas, de forma autônoma, quando há falha em um enlace. Os resultados mostram que utilizando um controlador SDN ONOS é possível restabelecer a comunicação, gerando uma rota alternativa de forma automatizada, na ordem dos milissegundos.

Palavras-chaves: SDN. ONOS. Detecção de falha. Sistema Autônomo. Garantia de tráfego.

Abstract

The Internet's popularity, caused by the cheapening and advancement of network technologies, has given rise in society to several applications that are widely used, and since some of these services are highly dependent on the proper functioning of the network, it is important to ensure that in case of a link failure some alternative route can be generated in the shortest time possible. Therefore, the objective of this work is to evaluate and experimentally analyze the order of magnitude of the time in which the ONOS SDN controller takes to generate a new route, autonomously, when a link failure is discovered. The results show that using an ONOS SDN controller it is possible to reestablish the communication, generating an alternative route automatically, in the order of milliseconds.

Key-words: SDN. ONOS. Failure detection. Autonomous system. Traffic assurance.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

O sucesso da Internet é indiscutível, está presente em praticamente todos os campos da nossa sociedade, e cada vez há mais soluções para nossos problemas que dependem do bom desempenho da rede. Um relatório da Cisco [1] mostra como o tráfego ultrapassará 3 Zettabytes(10^{31} bytes) em 2021. Devido às proporções que a Internet tomou, atualmente é praticamente impossível parar a rede para efetuar alguma mudança, uma vez que há serviços críticos na rede que não podem ser interrompidos (DAHLIN et al., 2003). Para solucionar esse problema, podemos utilizar o paradigma de redes definidas por software (SDN - *Software Defined Network*), que tem como principal característica a separação do plano de controle e do plano de dados, que executa o encaminhamento (KREUTZ et al., 2014). Essa separação permite uma visão mais estruturada e mais ágil da rede, facilitando seu controle, além de permitir que novas tecnologias sejam testadas em ambiente real sem afetar a disponibilidade da mesma. O mais importante dessa abordagem é que ela pode sem implantada gradualmente, substituindo os switches e roteadores atuais por equipamentos que permitam o uso de um protocolo de comunicação, como o OpenFlow (MCKEOWN et al., 2008), entre os planos de dados e controle.

Com a disseminação da Internet, hoje podemos facilmente acessar um datacenter do outro lado do planeta, seja para fazer a transferência de algum arquivo ou a requisição de algum serviço. Chamamos essas redes de longa distância de redes WAN *Wide Area Network*. As redes WAN em geral são compostas por redes de menor porte. Os roteadores (P. Almquist, 1994) são equipamentos responsáveis pelo encaminhamento dos dados, fazendo a ligação entre redes distintas. Temos como exemplos as seguintes aplicações em redes WAN: a migração de sistemas, requisições em banco de dados e visualização de vídeos sob demanda (DAHLIN et al., 2003).

1.1 O problema de pesquisa

Existem diversos sistemas que são críticos, ou seja, um conjunto de aplicações que se forem interrompidas ou tiverem um atraso muito grande na entrega dos dados, podem inviabilizar o uso de um ou mais serviços e resultar em grandes prejuízos, inclusive morte (KNIGHT, 2002). O esforço para manter o bom funcionamento da rede e garantir que essas aplicações não falhem deve ser constante. Como falhas podem ocorrer, é de extrema importância garantir que a detecção das mesmas e o seu reparo sejam feitos o mais rápido possível, evitando assim perdas e prejuízos (WOOD et al., 2010). Para

[1] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf

solucionar esse problema, utilizamos a arquitetura de redes definidas por software no presente trabalho, através do controlador ONOS (BERDE et al., 2014). Assim a detecção de falhas é automatizada e são realizadas ações caso uma falha tenha sido encontrada, evitando que uma pessoa fique encarregada por realizar essas tarefas.

1.2 Objetivos

1.2.1 Objetivos Gerais

O projeto tem como objetivo utilizar o controlador SDN open source ONOS, para que em caso de falhas em um enlace, uma rota alternativa possa ser criada automaticamente, garantindo a entrega do conteúdo em uma topologia multirotas (*multipath*).

1.2.2 Objetivos Específicos

Este trabalho possui como objetivos específicos os seguintes itens:

- Utilizar uma implementação do controlador ONOS em uma rede WAN simulada.
- Manter um fluxo de comunicação de dados entre dois hospedeiros na rede WAN.
- Desenvolver um experimento para mensurar o tempo de inatividade ao gerar uma nova rota ao provocar a queda manual de um enlace na rota do fluxo entre os dois hospedeiros.

1.3 Justificativa

Automatizar esse processo evita a necessidade de uma pessoa ter que identificar e solucionar a falha que gerou a queda de um link. Esse processo de identificar e solucionar a falha pode demorar muito, ou nem acontecer caso passe despercebido por um ser humano, e em uma aplicação crítica, onde o tráfego não pode ser interrompido, reduzir esse tempo é relevante. A redução do tempo se dá pela automatização do processo.

1.4 Estrutura do trabalho

O restante deste trabalho é organizado como segue. O Capítulo 2 apresenta a revisão bibliográfica sobre o tema, além de conceitos básicos que são necessários para o entendimento deste trabalho. O Capítulo 3 apresenta os métodos e procedimentos sobre o desenvolvimento do trabalho. O Capítulo 4 apresenta o resultado dos dados coletados pelo teste proposto. O Capítulo 5 apresenta os principais aspectos e os desafios encontrados neste trabalho.

2 Revisão bibliográfica

Neste capítulo serão apresentados os conceitos, os métodos e as tecnologias necessárias para o desenvolvimento do projeto.

2.1 Redes de computadores

O avanço das tecnologias embutidas nos computadores e o crescimento de aplicações foi proporcionado graças ao surgimento das redes de computadores. Com essa rede é possível que dois terminais distintos se comuniquem trocando informações. Esses terminais podem estar ligados através de cabos, como par trançado e fibra ótica, ou via ar, utilizando redes sem fio. Denominamos que os hosts são os terminais, ou seja, quem envia e quem recebe a mensagem. Quando um host conecta a uma rede, é definido um endereço IP para ele. Esse endereço serve como uma identificação, e é através dele que outros dispositivos na rede conseguem lhe enviar mensagens (KUROSE; ROSS, 2009).

Para que essa comunicação fosse eficiente e funcional foram (e são) criados vários protocolos, que são padronizações ou procedimentos de como os terminais devem agir para se comunicar. "Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou no recebimento de uma mensagem ou outro evento"(KUROSE; ROSS, 2009).

Uma rede é composta por roteadores, switches ou comutadores, que fazem a ligação do hospedeiro a uma rede, e são os responsáveis por direcionar a mensagem para o destino (P. Almquist, 1994). Esses equipamentos podem ser programáveis, como no caso das SDNs, ou serem engessadas, como é o caso da maioria presente no mercado atual.

Na época em que foi concebida a ideia de redes jamais imaginariam a proporção que ela tomaria, isso explica a dificuldade na evolução e na inovação da infraestrutura da rede, por isso é comum falar que a estrutura da nossa rede está calcificada (MCKEOWN et al., 2008).

2.1.1 Protocolo TCP/IP

As primeiras redes foram desenvolvidos com fins militares em 1969 na criação do ARPANET ou *Advanced Research Project Agency Network* (MOREIRA et al., 2013), que visava conectar bases militares distantes entre si, para que elas pudessem trocar informações. Devido ao aumento de usuários na rede, se fez necessário a criação do TCP/IP, que são alguns dos protocolos de comunicação utilizados na Internet.

Com o protocolo TCP, *Transmission Control Protocol*, é possível ter garantia de entrega das mensagens em ordem, assim caso uma mensagem se perca no meio do caminho ela é reenviada. O TCP também mantém o controle da rede, evitando o congestionamento e que o emissor não envie mais mensagens que o receptor consiga suportar (SOCOLOFSKY; KALE, 1991).

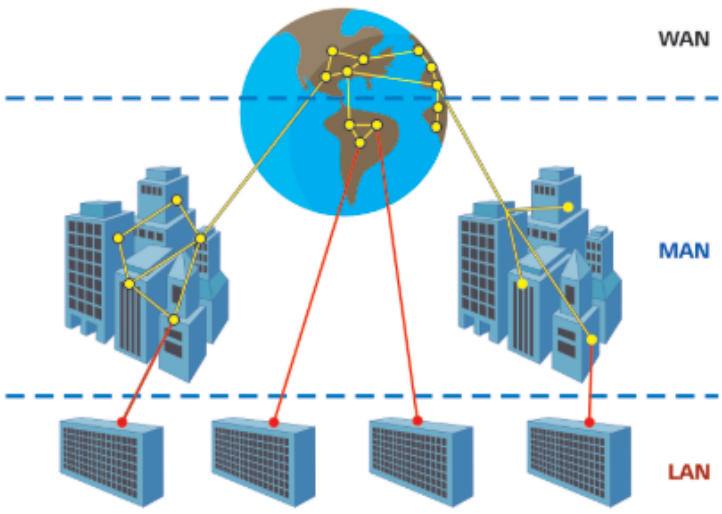
O protocolo IP permite que blocos de dados (ou datagramas) sejam transmitidos entre remetente e destinatário. Os usuários remetente e destinatário são identificados pelo endereço IP. No caso do IPv4 são 32 bits para endereçamento, ou cerca de 4 bilhões de endereços, já a nova versão do protocolo, o IPv6, utiliza 128 bits e tem cerca de $3,4 * 10^{38}$ endereços (KUROSE; ROSS, 2009).

2.1.2 Tipos de redes

Os tipos de redes são definidos pelo seu tamanho, são elas: a rede de área local LAN *Local Area Network*, a rede de área metropolitana MAN *Metropolian Area Network* e a rede de área extensa WAN *Wide Area Network*(KUROSE; ROSS, 2009). Cada uma dessas redes possuem características distintas, como diferentes larguras de banda, e umas são mais sensíveis a falha que outras.

A popularização da tecnologia e o constante avanço para deixá-la mais eficiente, proporcionou o desenvolvimento de aplicações onde o serviço está sendo feito do outro lado do planeta. Atualmente, podemos facilmente fazer uma videochamada entre Washington e Brasília, embora a distância dessas cidades seja de quase 7000 Km.

Figura 1 – Tipos de rede



Fonte: Retirado de (KREUTZ et al., 2014)

2.1.3 Ambiente multirotas

Durante qualquer período de tempo, podem existir infinitos caminhos interligando dois hosts (SAVAGE et al., 1999). Cada um desses enlaces possui características únicas, como latência, congestionamento e largura de banda. Escolher dentre esses caminhos qual é o melhor é custoso, e em um ambiente real onde essas características variam com o tempo, inviabiliza esse processo. Em um ambiente multirotas, caso um enlace venha a falhar, a chance de ter um caminho alternativo é maior, o que faz reduzir o tempo sem comunicação entre os hosts.

2.1.4 Tráfego

Com o barateamento dos dispositivos e o sucesso da internet, o volume de dados na rede aumentou e vem aumentando exponencialmente (Incorporated ULC, 2016). É por isso que ela é fundamental para a nossa sociedade e está presente em praticamente todos os campos, é possível desde assistir filmes, como os oferecidos pela Netflix, e transferir criptomoedas para outra pessoa, como o Bitcoin (NAKAMOTO, 2008).

No estudo feito pela Sandvine (Incorporated ULC, 2016), é possível visualizar que os serviços de streaming de vídeo compõem a maior parte do tráfego da américa latina, seguido pela navegação na Web. Em um outro estudo (Cisco Mobile, 2017), realizado pela empresa Cisco, em 2021 é esperado que o tráfego anual global chegue a 3,3 Zettabytes (10^{21} bytes).

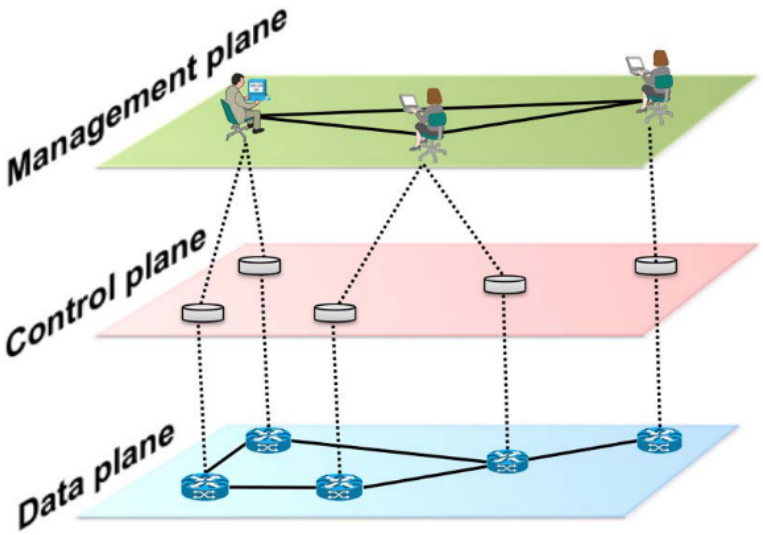
2.2 Redes definidas por software

Redes definidas por software, ou *Software Defined Networks* (SDN) (KIM; FEAMSTER, 2013), é um novo paradigma de rede que está emergindo e veio para solucionar alguns dos problemas da rede, uma vez que com ele é possível separar o plano de controle do plano de dados, facilitando o controle dos fluxos e o encaminhamento dos pacotes. Esse paradigma vem chamando a atenção da comunidade acadêmica e também da indústria, pois há cada vez mais estudos e novas soluções para melhorar sua eficiência.

O SDN possui como característica principal (KREUTZ et al., 2014) que os planos de controle e de dados são desacoplados, removendo dos dispositivos da rede o controle, os deixando apenas para encaminhar pacotes. As decisões de encaminhamento são baseadas em fluxos, o que facilita a flexibilidade da rede, que é limitada apenas pela criação de novas tabelas de fluxo.

O controlador é semelhante a um sistema operacional tradicional, ou seja, fornece uma abstração que facilita o desenvolvimento de novas aplicações. Mas a característica mais importante é o fato de ser possível programar a rede usando aplicações de software, o

Figura 2 – Camadas de uma SDN



Fonte: Retirado de (KREUTZ et al., 2014)

que resulta em um controle muito maior, podendo criar regras como descartar pacotes de um determinado emissor, ou criar uma nova rota quando é detectada queda em um enlace.

2.2.1 Funcionamento e arquitetura

A arquitetura de uma SDN é dividida em duas camadas (GUEDES et al., 2012), de controle e de infraestrutura, as quais podem ser observadas na Figura 2. A camada de infraestrutura é responsável por encaminhar pacotes, lendo apenas a tabela de encaminhamento e direcionando o pacote para a porta especificada. A camada de controle, ou controlador, tem como papel tratar e gerenciar todos os aspectos da rede, como mecanismos de segurança, e proporcionar uma visão global da rede.

O controlador se comunica com a camada de infraestrutura através do protocolo OpenFlow, que foi criado para padronizar a comunicação dos switches com os controladores. Atualmente muitos dos switches presentes na rede não podem ter seu código fonte modificado, porque depende do fornecimento do mesmo pela fabricante, o que não ocorre em SDN. Em SDN também existe a camada de aplicação, que utiliza os dados que são providos pelo controlador para construir aplicações, como firewall e balanceamento de carga.

Utilizando essa abordagem é possível manter o alto desempenho no encaminhamento dos pacotes via hardware, além de aumentar a flexibilidade da rede, permitindo adicionar ou remover fluxos de dados através das aplicações do controlador. Toda vez que um pacote é recebido pelo switch OpenFlow, é feita uma consulta na tabela de encaminhamento do switch para determinar seu destino, e caso ainda não tenha nenhuma regra do que fazer

com ele, ou para onde enviar, o pacote pode ser descartado ou uma nova regra pode ser gerada. Caso a consulta tenha sucesso, o pacote é então encaminhado para o seu destino.

2.2.2 OpenFlow

O OpenFlow (MCKEOWN et al., 2008) é um padrão de comunicação que faz o intermédio entre a camada de controle e a de dados em diferentes switches e roteadores. Esse padrão foi criado com intenção de testar novos protocolos em ambientes reais, sem afetar a comunicação nessas redes. De uma forma simples o OpenFlow atua sob os switches de acordo com as regras determinadas pelos controladores.

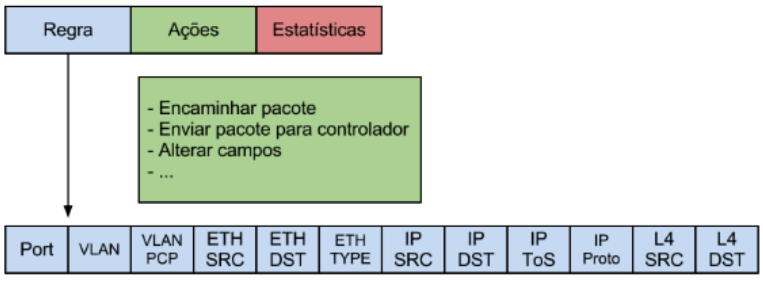
Os switches OpenFlow são divididos em três partes. A tabela de encaminhamento, que tem uma ação para cada fluxo, um canal seguro para a comunicação entre o switch e o controlador, e o protocolo OpenFlow, que permite o controlador alterar a tabela de encaminhamento do controlador.

Sempre que um pacote é recebido pelo comutador OpenFlow, ele pode sofrer algumas ações.

- O pacote pode ser enviado para uma determinada porta.
- O pacote pode ser encapsulado e enviado para o controlador via o canal seguro, (Normalmente essa ação ocorre com o primeiro pacote de um fluxo.)
- O pacote pode ser descartado.

A tabela de encaminhamento de um switch OpenFlow representada na Figura 3 é composta por um cabeçalho, que contém informações sobre o fluxo, por uma ação, que define o que fazer com o pacote, e pelas estatísticas do fluxo, que informam o número de pacotes e o tamanho da transmissão. Por isso essas tabelas, permitem ao desenvolvedor criar diversas aplicações, como por exemplo balanceamento de carga (MOURA et al.,) e entrega de conteúdo (NAM; CALIN; SCHULZRINNE, 2015).

Figura 3 – Tabela de Fluxo OpenFlow



Fonte: Retirado de (Lucas Rodrigues Costa, 2013)

2.2.3 Controladores de rede

Os controladores de rede utilizam uma abstração que permite ele se comunicar com os switches de diferentes marcas, como é caso do OpenFlow. Além disso, eles também são encarregados de disponibilizar algumas funções essenciais, como por exemplo o descobrimento de novos dispositivos na rede, e a coleta de informações sobre o status da rede (LARA; KOLASANI; RAMAMURTHY, 2014).

Com isso, os controladores têm papel fundamental no funcionamento das redes definidas por software. Por eles terem uma visão centralizada do status da rede, e quem a gerencia, eles podem configurar as regras de encaminhamento de acordo com sua necessidade. É importante ressaltar que ter uma visão centralizada não impede que se tenha mais de um controlador responsável pela mesma rede (MUQADDAS et al., 2016).

Desde o surgimento de SDN, vários controladores já foram criados com os mais diversos propósitos. Alguns desses controladores atuam de forma centralizada, devido a simplicidade, enquanto outros buscam maior escalabilidade e disponibilidade de seus sistemas.

Dentre os controladores existentes, são destacáveis o NOX (GUDE et al., 2008), o POX (Ligia Rodrigues Prete et al., 2014) e o Onix (KOPONEN et al., 2010).

2.2.4 Mininet

O MININET foi criado com o propósito de simular redes, evitando assim a necessidade de ter uma rede real com todos os componentes. Essa ferramenta permite que, utilizando apenas um computador, seja possível fazer a prototipação de uma grande rede sem muito custo nem muita dificuldade.

Uma das principais características do MININET é a criação, por meio de software, de switches OpenFlow, que ao serem virtualizados, possuem as mesmas características dos switches físicos. Por isso é possível criar uma rede definida por software de forma rápida, e sem ter a necessidade dos componentes físicos da rede.

Com o surgimento de SDN, o MININET se faz importante por conseguir, sem muito esforço, que um pesquisador simule uma rede complexa com diversos nós interligados. Isso tudo é graças aos atributos do MININET de flexibilidade, escalabilidade, realidade, interatividade, compartilhamento e modulação. (LANTZ; HELLER; MCKEOWN, 2010)

2.2.5 Controlador Onos

Como já explicado anteriormente, os controladores têm papel fundamental em uma rede definida por software. O *Open Network Operating System* (ONOS) (BERDE et al., 2014) é um controlador de código aberto que foi criado para resolver requisitos feitos pelos

operadores de rede. A alta vazão, baixa latência, alta disponibilidade e o volume da rede global, são os requisitos necessários para um bom controlador.

O ONOS segue os passos do seu antecessor o Onix (KOPONEN et al., 2010) de disponibilizar um controlador SDN distribuído, com a diferença de ser código aberto, o que permite que a comunidade possa examinar, testar e contribuir para um melhor controlador.

Por ser um controlador com o núcleo distribuído (MUQADDAS et al., 2016), o ONOS consegue ter uma alta performance, ser escalável e ter uma alta disponibilidade. O núcleo do ONOS é responsável por manter o estado da rede, interagir com os dispositivos da rede por meio de APIs e fornecer serviços para as aplicações também usando APIs.

3 Desenvolvimento

Este capítulo descreve os materiais e ferramentas utilizadas para calcular o tempo gasto pelo controlador ONOS para refazer uma rota. Para o desenvolvimento do trabalho foi utilizado um notebook Dell Inspiron 1501 com processador Intel Core i5 de 2.30GHz, 6GB de memória principal e 120 GB de armazenamento.

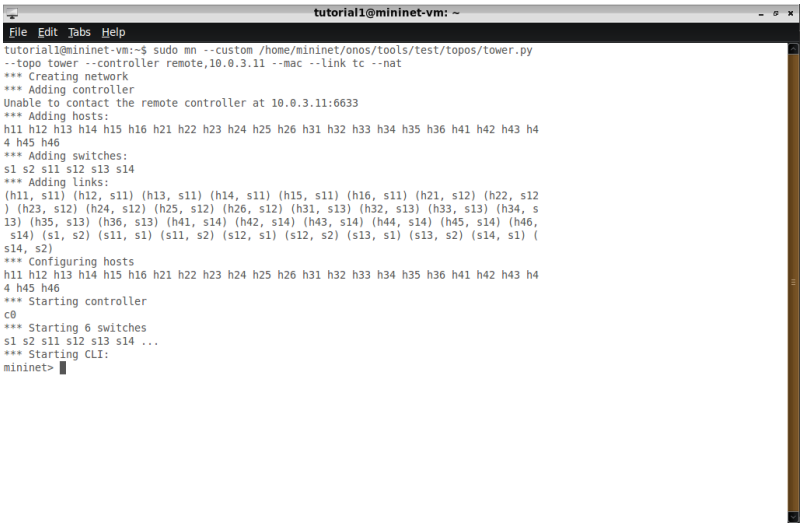
Para realizar a emulação de uma rede de computadores foi utilizado o Mininet na versão 2.2.1. O controlador utilizado pela rede SDN foi o ONOS na versão 1.1.0, que utiliza o OpenFlow 1,5. Para gerar um tráfego entre dois hospedeiros foi utilizada a ferramenta Bwping-UDP. Todas essas ferramentas foram executadas em um ambiente Linux Ubuntu 14.10.

Os experimentos foram realizados no laboratório da Universidade Federal de Ouro Preto.

3.1 Criação de ambiente de simulação usando o Mininet

Como explicado anteriormente, o Mininet[1] é uma ferramenta para simulação de redes gratuita. Para simular um ambiente real, foi utilizada uma topologia composta de 24 hosts e 6 switches. Cada host tem acesso à internet por meio do protocolo NAT. A Figura 5 ilustra a topologia simulada.

Figura 4 – Execução Mininet



Fonte: Produzido pelo autor

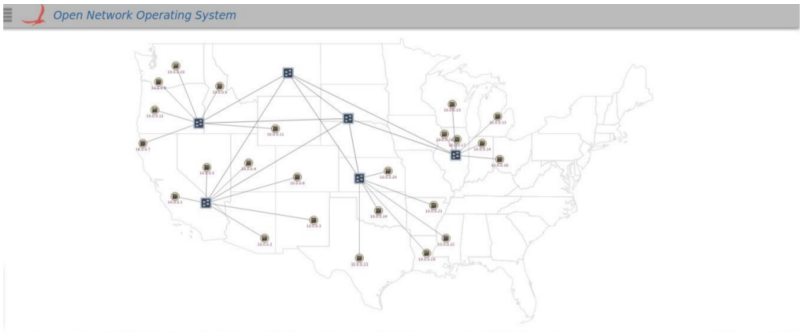
[1] http://mininet.org/

O Mininet pode ser executado pelo terminal utilizando o comando *sudo mn –custom /home/mininet/onos/tools/test/topos/tower.py –topo tower –controller remote,10.0.3.11 –mac –link tc –nat*, que tem como parâmetros o endereço IP do controlador, a topologia usada, além de habilitar o uso do protocolo NAT. A execução do Mininet com os parâmetros descritos pode ser visto na Figura 4.

3.2 Executando o controlador ONOS

O controlador ONOS é desenvolvido na linguagem Java [2]. Ele está disponível em uma maquina virtual no SDN-hub [3] contendo o controlador ONOS e o MININET já configurados e prontos para uso. O ONOS fornece uma interface gráfica onde se tem uma melhor visão da rede. Nele também é possível visualizar a topologia da rede, como exibida na Figura 5.

Figura 5 – Topologia criada pelo Mininet exibida na interface do ONOS



Fonte: Produzido pelo autor

A inicialização do ONOS pode ser feita pelo terminal do Linux executando o comando */home/mininet/apache-karaf-3.0.3/bin/client -u karaf -h 10.0.3.11*. Após ser iniciado é necessário ativar o controlador para utilizar o *Intent framework* com o comando *onos:app activate org.onosproject.ifwd*. Esse processo é demonstrado no Figura 6.

O *Intent framework* é uma abstração usada pelo ONOS que permite a criação de regras complexas sem a preocupação com os detalhes dos componentes da rede (SANVITO et al., 2018), que permite os programadores preocuparem com o que deve ser feito, não em como é feito. Cada *intent* é compilado de forma separada dos outros, o que permite que em caso de alguma mudança na topologia do caminho, o objetivo permaneça funcionando.

[2] https://www.java.com/
[3] http://sdnhub.org/

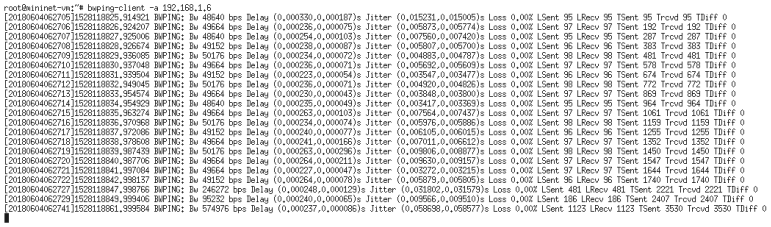
Figura 6 – Execução ONOS

Fonte: Produzido pelo autor

3.3 Criação de fluxo utilizando BWPING-UDP

O BWPING-UDP[4] é um *software* livre que permite calcular o desempenho da rede entre dois hosts utilizando o protocolo UDP (MOURA; SILVA; MACEDO, 2017). O BWPING-UDP é composto de um servidor e um cliente, e um tráfego é gerado entre eles, permitindo medir as taxas de perda de pacotes, atraso e variação do atraso (*Jitter*).

Figura 7 – Execução do BWPING-UDP Cliente

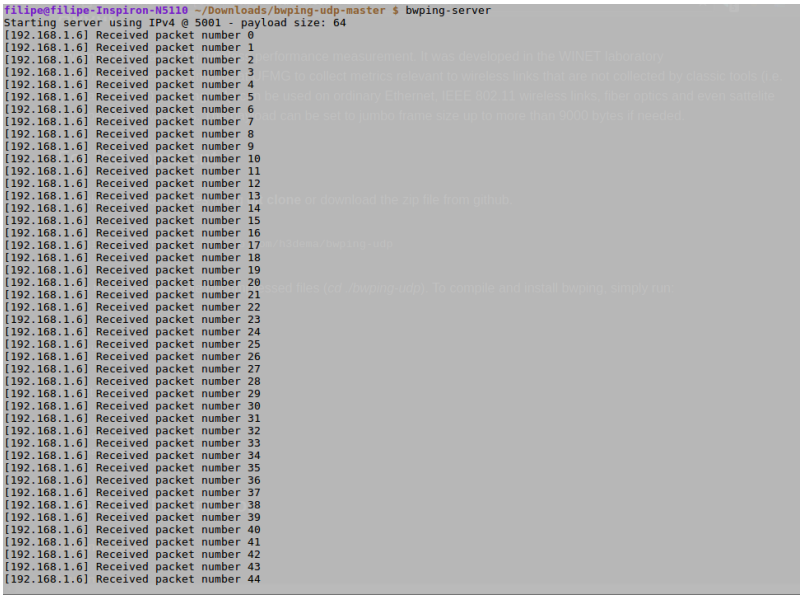


Fonte: Produzido pelo autor

Para usar o BWPING-UDP, primeiro é necessário executar o servidor em um host, que pode ser feito com o comando *bwping-server*, como demostrado na Figura 8. Como configuração padrão a porta de comunicação é definida como 5001 e o tamanho do pacote é de 64 bytes. Com o servidor configurado, podemos executar o cliente. Para isso utilizamos o comando *bwping-client -a 200.239.155.147*, com o endereço do servidor, como demonstrado na Figura 7. O parâmetro -a define a utilização de IPv4, o endereço IP do servidor também é passado como parâmetro.

[4] https://github.com/h3dema/bwping-udp

Figura 8 – Execução do BWPING-UDP Servidor

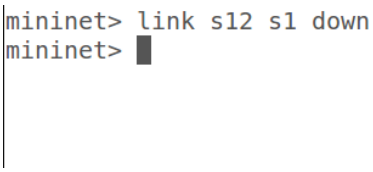


Fonte: Produzido pelo autor

3.4 O experimento

Foram realizados um total de 33 experimentos, onde foram medidos os tempos de inatividade quando uma falha manual em um enlace é gerada. A topologia representada pela Figura 5 é utilizada para simular uma rede WAN multirotas. Para a realização do experimento primeiro é gerado tráfego entre dois hosts, que é feito pelo BWPING-UDP. A interrupção da comunicação é feita manualmente desativando um enlace pelo MININET utilizando o comando *link s12 s1 down*, como pode ser visto na Figura 9. Esse comando tem como parâmetros os switches que compõe o enlace, e a ação a ser feita, onde *down* representa desligar.

Figura 9 – Desativando um enlace pelo MININET



Fonte: Produzido pelo autor

Como a topologia simulada é um ambiente multirotas, o controlador ONOS ao detectar a queda do enlace gera uma nova rota entre os dois mesmos hosts de forma automática. O tempo entre a queda do enlace até o restabelecimento da comunicação pode ser calculado pela razão entre a quantidade de pacotes perdidos a segundo e a quantidade

de pacotes enviados em um segundo. O BWPING-UDP fornece estatística da comunicação a cada segundo, contendo os números de pacotes enviados e perdidos.

4 Resultados

No cenário de comunicação em uma rede WAN multirotas, primeiramente foi mensurado o tempo em que o controlador ONOS, ao detectar uma falha em um enlace, demora para gerar uma nova rota, restabelecendo a comunicação entre os terminais. Com a amostra coletada por meio da ferramenta BWPING-UDP descrita na seção 3.3, foi feito o cálculo da *Comulative Distribution Function* (CDF).

Tabela 1 – Número de pacotes enviados e perdidos em 1 segundo

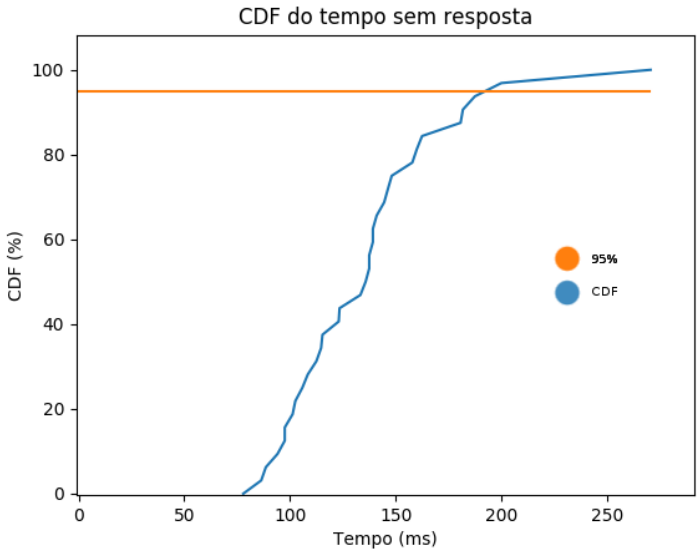
Pacotes Enviados	Pacotes Perdidos	Tempo sem comunicacao (ms)
78	9	115.38
85	8	94.12
80	13	162.5
82	8	97.56
75	10	133.33
81	12	148.15
83	15	180.72
79	11	139.24
83	9	108.43
81	11	135.8
79	8	101.27
78	8	102.56
79	7	88.61
80	15	187.5
70	14	200
81	10	123.46
77	6	77.92
83	12	144.58
77	14	181.82
80	11	137.5
85	9	105.88
81	7	86.42
80	9	112.5
61	7	114.75
80	11	137.5
76	12	157.89
85	23	270.59
65	8	123.08
82	8	97.56
79	11	139.24
82	12	146.34
75	12	160
78	11	141.03

Fonte – Produzido pelo autor.

A partir do cálculo da CDF, é possível saber a porcentagem dos valores menores que um tempo estipulado. Como pode ser visto na Figura 10, em 95% dos 33 testes feitos o tempo em que o controlador ONOS demorou para gerar uma nova rota de forma

automática para a comunicação foi de aproximadamente 200 milissegundos.

Figura 10 – CDF do tempo para gerar nova rota



Fonte: Produzido pelo autor

Como a amostra coletada apresentou um baixo desvio padrão, foi detectado que 33 experimentos seriam suficientes para uma conclusão satisfatória. A Tabela 2 apresenta o tempo mínimo, máximo, a média e o desvio padrão em milissegundos do tempo em que a comunicação interrompida.

Por meio da análise da CDF, observamos que o tempo entre a interrupção e o restabelecimento da comunicação entre os dois hosts está na casa dos milissegundos. Esse atraso, causado pela geração de uma nova rota, pode impactar a experiência dos usuários utilizando um serviço, por exemplo, uma video chamada poderia ser interrompida, assim como o download de algum arquivo.

Tabela 2 – Tempos medidos

min (ms)	max (ms)	média (ms)	desvio padrão (ms)
77.92	270.59	134,94	39,37

Fonte – Produzido pelo autor.

5 Conclusão

Com a demanda pelos serviços disponibilizados na Internet cada vez maior, os provedores de serviço têm como desafio garantir uma alta confiabilidade da rede, o que deixa evidente a necessidade de mecanismos com melhor garantia do tráfego.

De acordo com os resultados, em busca da garantia de tráfego, o uso do controlador ONOS se mostra eficiente. Vale ressaltar que todo o processo, desde a detecção da falha, o cálculo para a melhor rota, a configuração dos switches e o restabelecimento da comunicação é automatizado pelo ONOS, evitando a necessidade de intervenção humana.

A média de latência entre dois terminais em uma rede WAN é de 104.1 milissegundos, sendo o mínimo 103.3 e o máximo 106.8 milisegundos (AMIR; DANILOV; STANTON, 2000). Podemos concluir que ao utilizar o controlador ONOS, a interrupção do tráfego entre dois hosts por alguma eventualidade não afeta a comunicação dos mesmos, uma vez que gerar uma nova rota demora no máximo 270.59 milissegundos, apenas cerca de 3 vezes mais que a latência média em uma rede WAN multirotas. Para que uma falha fosse perceptível para o usuário, o tempo sem comunicação teria que ser pelo menos 10 vezes maior do que a latência.

Como trabalhos futuros, pretendemos analisar o uso do controlador ONOS em ambientes reais ao invés de simulados. Assim, poderemos ter uma visão melhor da garantia do tráfego quando forem utilizados equipamentos reais, aproximando-se assim da realidade.

Referências

AMIR, Y.; DANILOV, C.; STANTON, J. A low latency, loss tolerant architecture and protocol for wide area group communication. In: *Proceeding International Conference on Dependable Systems and Networks. DSN 2000*. IEEE Comput. Soc, 2000. p. 327–336. ISBN 0-7695-0707-7. Disponível em: <http://ieeexplore.ieee.org/document/857557/>. Citado na página 26.

BERDE, P. et al. Onos. *Proceedings of the third workshop on Hot topics in software defined networking - HotSDN '14*, p. 1–6, 2014. Disponível em: <http://dl.acm.org/citation.cfm?doid=2620728.2620744>. Citado 2 vezes nas páginas 11 e 17.

Cisco Mobile. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper*. [S.l.], 2017. 2016–2021 p. Citado na página 14.

DAHLIN, M. et al. End-to-End WAN service availability. *IEEE/ACM Transactions on Networking*, v. 11, n. 2, p. 300–313, 2003. ISSN 10636692. Citado na página 10.

GUDE, N. et al. NOX. *ACM SIGCOMM Computer Communication Review*, v. 38, n. 3, p. 105, jul 2008. ISSN 01464833. Disponível em: <http://portal.acm.org/citation.cfm?doid=1384609.1384625>. Citado na página 17.

GUEDES, D. et al. Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em Redes de Computadores. *Sbrc*, p. 160–210, 2012. Citado na página 15.

Incorporated ULC, S. 2016 - Global Internet Phenomena - Latin America & North America. 2016. Disponível em: <https://www.sandvine.com/hubfs/downloads/archive/2016-global-internet-phenomena-report-latin-america-and-north-america.pdf>. Citado na página 14.

KIM, H.; FEAMSTER, N. Improving network management with software defined networking. *IEEE Communications Magazine*, v. 51, n. 2, p. 114–119, 2013. ISSN 01636804. Citado na página 14.

KNIGHT, J. C. Safety critical systems: challenges and directions. *Proceedings of the 24rd International Conference on Software Engineering (ICSE), 2002. IEEE.*, p. 547 – 550, 2002. Citado na página 10.

KOPONEN, T. et al. Onix: A Distributed Control Platform for Large-scale Production Networks. 2010. Disponível em: <http://static.usenix.org/events/osdi10/tech/full{_}papers/Koponen.> Citado 2 vezes nas páginas 17 e 18.

KREUTZ, D. et al. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, v. 103, n. 1, p. 14 – 76, 2014. ISSN 0018-9219. Disponível em: <http://arxiv.org/abs/1407.6062>. Citado 4 vezes nas páginas 10, 13, 14 e 15.

KUROSE; ROSS. *Redes de computadores e a Internet*. [S.l.: s.n.], 2009. Citado 2 vezes nas páginas 12 e 13.

LANTZ, B.; HELLER, B.; MCKEOWN, N. A network in a laptop: rapid prototyping for software-defined networks. *. . . Workshop on Hot Topics in Networks*, p. 1–6, 2010. ISSN 1450304095. Disponível em: <http://dl.acm.org/citation.cfm?id=1868466>. Citado na página 17.

LARA, A.; KOLASANI, A.; RAMAMURTHY, B. Network innovation using open flow: A survey. *IEEE Communications Surveys and Tutorials*, v. 16, n. 1, p. 493–512, 2014. ISSN 1553877X. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6587999>. Citado na página 17.

Ligia Rodrigues Prete et al. Simulation in an SDN network scenario using the POX Controller. In: *2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*. IEEE, 2014. p. 1–6. ISBN 978-1-4799-4340-1. Disponível em: <http://ieeexplore.ieee.org/document/6860403/>. Citado na página 17.

Lucas Rodrigues Costa. OpenFlow e o Paradigma de Redes Definidas por Software. p. 1–155, 2013. Citado na página 16.

MCKEOWN, N. et al. OpenFlow. *ACM SIGCOMM Computer Communication Review*, v. 38, n. 2, p. 69, 2008. ISSN 01464833. Disponível em: <http://portal.acm.org/citation.cfm?doid=1355734.1355746>. Citado 3 vezes nas páginas 10, 12 e 16.

MOREIRA, M. D. D. et al. Internet do futuro: um novo horizonte. 2013. Disponível em: <http://ce-resd.facom.ufms.br/sbrc/2009/080.pdf>. Citado na página 12.

MOURA, H. et al. Comparação de Políticas de Divisão de Tráfego em Data Center empregando SDN. Citado na página 16.

MOURA, H. D.; SILVA, E. de Britto e; MACEDO, D. F. BWPING-UDP – Avaliando o Desempenho de Redes Sem Fio. 2017. Citado na página 21.

MUQADDAS, A. S. et al. Inter-controller traffic in ONOS clusters for SDN networks. *2016 IEEE International Conference on Communications, ICC 2016*, 2016. Citado 2 vezes nas páginas 17 e 18.

NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Www.Bitcoin.Org*, p. 9, 2008. ISSN 09254560. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Citado na página 14.

NAM, H.; CALIN, D.; SCHULZRINNE, H. Intelligent content delivery over wireless via SDN. *2015 IEEE Wireless Communications and Networking Conference, WCNC 2015*, p. 2185–2190, 2015. ISSN 1525-3511. Citado na página 16.

P. Almquist. RFC 1716. 1994. Citado 2 vezes nas páginas 10 e 12.

SANVITO, D. et al. ONOS Intent Monitor and Reroute service: enabling plug&play routing logic. 2018. Citado na página 20.

SAVAGE, S. et al. The end-to-end effects of Internet path selection. *ACM SIGCOMM Comput. Commun. Review*, v. 29, n. 4, p. 289–299, 1999. ISSN 01464833. Citado na página 14.

SOCOLOFSKY, T.; KALE, C. RFC1180. p. 1–28, 1991. Citado na página 13.

WOOD, T. et al. Disaster recovery as a cloud service: Economic benefits & deployment challenges. *2nd USENIX Workshop on Hot Topics in Cloud Computing. Boston, MA*, p. 1–7, 2010. Disponível em: <http://www.usenix.org/>. Citado na página 10.

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO

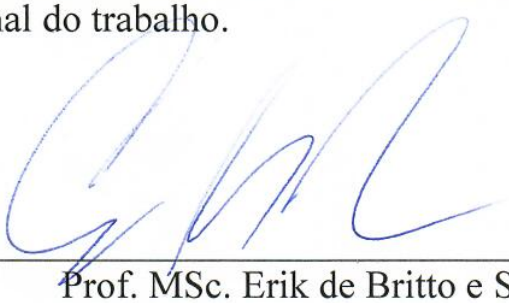
TERMO DE RESPONSABILIDADE

Eu, Filipe Fuscaldi de Assis Theodoro, declaro que o texto do trabalho de conclusão de curso intitulado "USO DE CONTROLADOR SDN ONOS PARA GARANTIA DE TRÁFEGO EM AMBIENTE MULTIROTAS" é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 13 de Julho de 2018

Filipe Fuscaldi de Assis Theodoro

Certifico que o aluno **Filipe Fuscaldi de Assis Theodoro,** autor do trabalho de conclusão de curso intitulado "**Uso de Controlador SDN ONOS para Garantia de Tráfego em Ambiente Multirotas**", efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.



Prof. MSc. Erik de Britto e Silva

Orientador

João Monlevade, 25 de julho de 2018