Redes Definidas por Software com Mininet e Onos

Bruno Moreto de Oliveira

Agosto 2022

SUMÁRIO

1 INTRODUÇÃO

O uso de redes definidas por software tem sido uma nova forma de pensar sobre grandes redes de comunicação, representam uma nova maneira de olhar a forma como as redes são controladas, configuradas e operadas e isso, são redes que usam controladores com base em software ou APIs, para direcionar o tráfego na rede e se comunicar com uma outra infraestrutura de hardwares. Entretanto, por ser uma forma nova de se trabalhar com redes de computadores, o nível de complexidade para montar um ambiente com essas características é maior, o ambiente é composto pelos switches físicos conectados por fibra óptica, é caracterizada por existir um sistema de controle (software) que pode controlar o mecanismo de encaminhamento dos elementos de comutação da rede por uma interface de programação que é bem definida. Neste trabalho desenvolvemos um ambiente físico com 4 switches ópticos para desenvolver um ambiente de testes com suporte a programabilidade para redes ópticas definidas por software, permitindo carregar tabelas de roteamento e inserir códigos de roteamento baseados em diversos critérios como logística, inteligência de negócios. E futuramente disponibilizar para pesquisadores do laboratório CIDIG realizarem testes de seus códigos.de gerenciamento de pacotes de redes a nível de chip, ou gravar a própria tabela de roteamento de pacotes para obter melhores resultados. Com o ambiente montado realizamos testes com alguns códigos e guardamos as métricas de desempenho que serão apresentadas em gráficos. O ambiente por fim ficou como esperávamos conseguiu rodar os códigos propostos e otimizou o roteamento de pacotes, agora o próximo passou é disponibilizar esse ambiente para que pesquisadores façam o acesso remoto ao ambiente.

1.1 FUNDAMENTOS CONCEITUAIS

1.1.1 Redes definidas por software(SDN)

Redes Definidas por Software (RDS, ou SDN, do inglês Software Defined Network) permite a programação de elementos de rede, tais como roteadores, comutadores, etc. Se fundamenta na separação dos planos de Controle e de Dados, tornando a rede programável e flexível. Ela é formada por três camadas principais: camada de aplicação, camada de controle e camada de infraestrutura. A camada de aplicação utiliza uma API para especificar o comportamento da rede. A camada de controle é responsável por traduzir requisitos da camada de aplicação para a camada de infraestrutura, e oferece uma visão abstrata da rede às aplicações. A camada de infraestrutura é composta pelos elementos da rede, e dispositivos que realizam a transmissão de dados. Segundo (Dorgival Guedes et al,2012) (GUEDES et al., 2012) , possibilita que a rede seja controlada de forma estensível atraves de aplicações, expressas em software. A esse novo paradigma, deu-se o nome de Redes Definidas por Software.

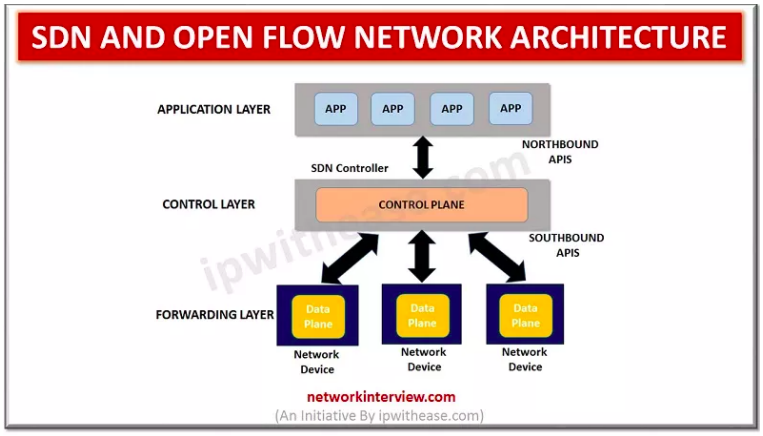
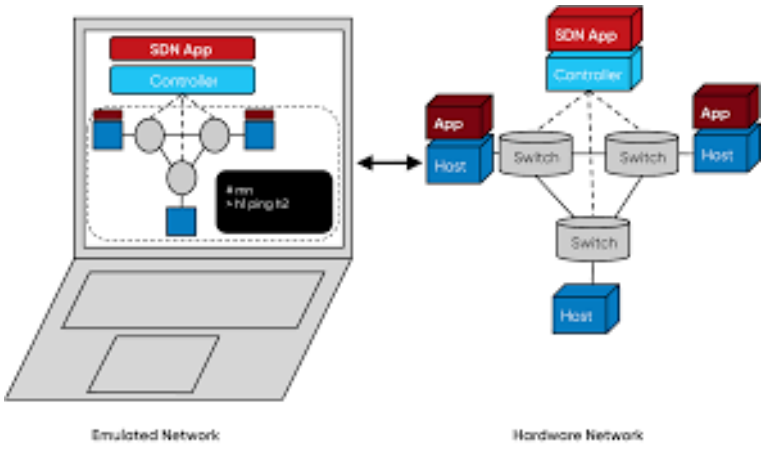


Figura - 1 Arquitetura das Redes Definidas por Software [1]

1.1.2 Mininet

Mininet, é um emulador de rede que cria redes com servidores, switches, controladores e enlaces virtuais, com ele é possível criar uma rede virtual realística rodando em um núcleo real (Kernel Linux), switches e códigos de aplicação em uma única máquina física ou virtual (Virtual Machine - VM) que pode ser nativa ou em nuvem (Cloud). O Mininet possui linha de comando própria e APIs, que permitem a criação de rede e serviços, customização e compartilhamento com outros usuários, e também a implantação em hardware real. A capacidade de banda disponível e CPU que não podem exceder à capacidade e banda disponível no servidor onde o Mininet está instalado. Para entender o Mininet segundo (FLAUZINO, 2019) é ter em mente que ele não é um simulador, mas sim um emulador. Na prática, isso significa que componentes da rede emulada podem executar aplicações reais e até se comunicar com redes reais. É possível, por exemplo, executar scripts, códigos (programas) próprios ou mesmo aplicações prontas (servidor Web, DNS, Firewall, etc.) nos hosts da rede emulada.



[1] https://networkinterview.com/basics-of-sdn-and-open-flow-network-architecture/

Figura 2 - Arquitetura Mininet [2]

1.1.3 ONOS

Open Network Operating System(ONOS), é um projeto open source de rede definida por software, ele faz parte da Linux Foundation e tem como objetivo melhorar a disponibilidade, desempenho e escalabilidade. Segundo (FERREIRA, 2016) ONOS é um projeto desenvolvido pela equipe do ON.Lab, uma organização sem fins lucrativos especializada em tecnologias SDN e OpenFlow. O projeto do controlador é pautado por algumas metas, tais como: liberar o desenvolvedor da aplicação de conhecer detalhes intrínsecos do hardware proprietário; libertar o operador de rede de interfaces e protocolos proprietários e permitir a evolução de componentes de hardware e software, i.e., cada um poderá seguir a linha do tempo de mercado de forma independente.

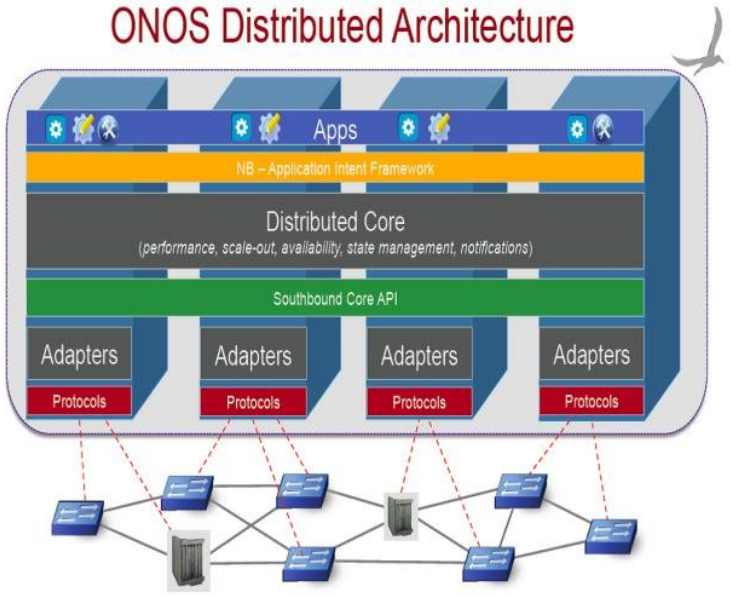


Figura 3 - Arquitetura ONOS [3]

1.1.4 Stratum

Segundo o site da opennetworking Stratum é um sistema operacional de switch independente de código aberto para redes definidas por software. Ele suporta vários switches de caixa branca e expõe um conjunto de interfaces SDN de última geração, incluindo P4Runtime e OpenConfig/gNMI.Stratum evita o aprisionamento do fornecedor dos planos de dados atuais (ou seja, interfaces proprietárias e APIs de software fechadas) e permite a fácil integração de dispositivos nas redes das operadoras. Ele oferece uma solução completa de comutação de caixa branca para cumprir a promessa de SDN 'definida por

[2] https://opennetworking.org/mininet/
[3] Experimenting with ONOS Scalability on Software Defined Network,

software'. Segundo o site plvision.eu/ o Stratum amplia o escopo do SDN para incluir controle completo do ciclo de vida, configuração e interfaces de operações. Ele fornece uma distribuição aberta e mínima pronta para produção para switches de caixa branca e permite a intercambialidade de dispositivos de encaminhamento, bem como a programação de comportamentos de encaminhamento.

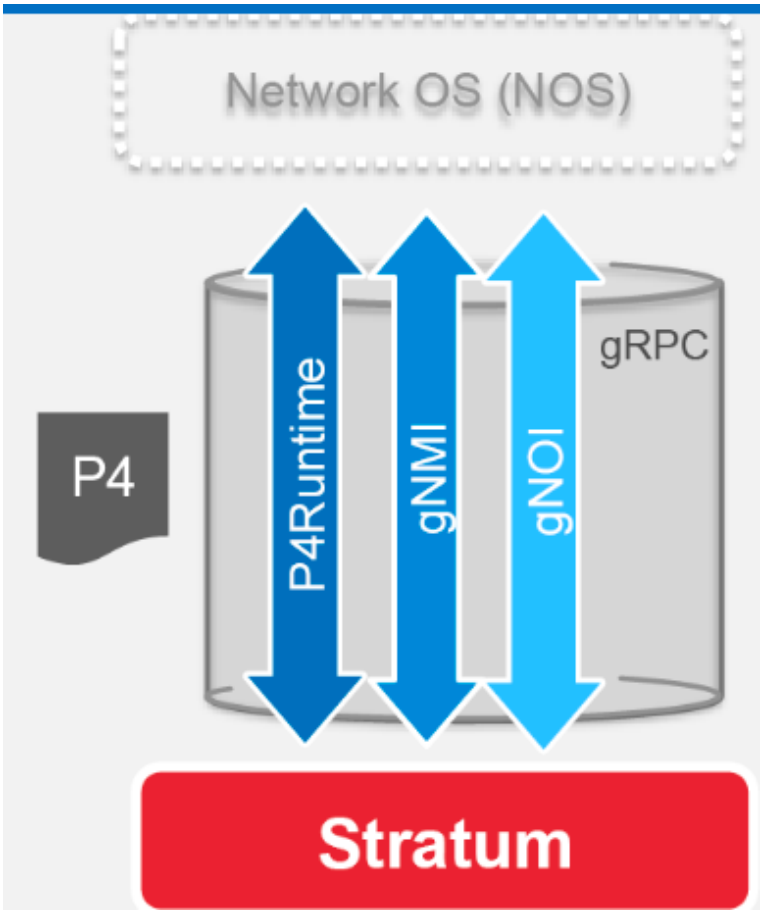


Figura 4 - Arquitetura Stratum [4]

1.1.5 SONIC

SONIC foi criado pela Microsoft para seus data centers do Azure. Segundo (LOVATO, 2022), é um sistema operacional de rede de código aberto baseado em Linux que roda em switches de vários fornecedores e ASICs. O SONiC oferece um conjunto completo de funcionalidades de rede, como BGP e RDMA, que foram reforçados pela produção nos data centers de alguns dos maiores provedores de serviços em nuvem. Ele oferece às equipes a flexibilidade de criar as soluções de rede de que precisam, aproveitando a força coletiva de um grande ecossistema e comunidade.

[4] https://opennetworking.org/stratum/

1.1.6 OpenFlow

Foi proposto pela Universidade de Stanford para atender à demanda de validação de novas propostas de arquiteturas e protocolos de rede. Segundo (ANDRÉ, 2013) foi Publicado em 2008, ele visa suportar novas propostas de protocolos e arquiteturas de rede sobre equipamentos comercialmente disponíveis. Descreve uma tecnologia para controlar as possíveis opções de encaminhamento de pacotes em switches e roteadores, entre outros dispositivos. A inteligência para a tomada destas decisões é responsabilidade de um elemento externo, chamado controlador OpenFlow, o qual pode ser instalado em servidores comuns. Existe um protocolo padronizado, chamado OpenFlow, que é responsável pela comunicação entre as camadas de controle e de infraestrutura. Um switch OpenFlow é responsável por encaminhar fluxo de pacotes para determinada porta, encapsular fluxo de pacotes e enviar ao Controlador e descartar fluxo de pacotes (e se nenhuma ação for especificada, o pacote é descartado).

1.2 TRABALHOS CORRELATOS

1.2.1 Trabalho 1

Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em Redes de Computadores. Autores: Dorgival Guedes, Luiz Filipe Menezes Vieira, Marcos Menezes Vieira, Henrique Rodrigues e Rogério Vinhal Nunes Neste trabalho os autores fazem uma abordagem sobre o tema redes definidas por software, são discutidos componentes de um sistema de rede definido por software, incluindo as soluções para virtualização dos elementos de rede, sistemas operacionais de rede e novas aplicações, bem como os desafios de pesquisa que esse paradigma ainda precisa enfrentar e os diversos esforços de pesquisa em andamento no Brasil e no mundo. O artigo inicia falando um pouco da história das redes definidas por software, como surgiu a iniciativa e como chegou a criação da interface e do protocolo OpenFlow, o que se destaca desse início é o trecho em que diz que os elementos de encaminhamento oferecem uma interface de programação simples que lhes permite estender o acesso e controle da tabela de consulta utilizada pelo hardware para determinar o próximo passo de cada pacote recebido. Em seguida é falado um pouco da motivação que surgiu a SDN, como foi aumentando sua importância, a quantidade de trabalhos que foram surgindo nessa linha de pesquisa, os autores citam como a proposta de uma interface de programação para os elementos de comutação da rede, como o padrão OpenFlow, atraio muitos fabricantes e pesquisadores gostaram da ideia de uma estrutura de redes onde a comutação de pacotes não precisa mais ser definida pelo princípio de roteamento de redes Ethernet ou IP, mas que pode ser controlado por aplicações(software) desenvolvido independentemente do hardware de rede. Nesse trabalho também é lembrado que os estudos relacionado as Redes Definidas por Software também abrem a possibilidade de se desenvolver novas aplicações que controlem os elementos de

comutação de uma rede físicas de formas impensadas no passado, como algoritmos de logísticas que podem ser implementados na hora rotear um pacote, até mesmo questões de inteligência de negócios podem ser implementadas

1.2.2 Trabalho 2

OpenFlow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes. Autores: Christian Esteve Rothenberg* , Marcelo Ribeiro Nascimento, Marcos Rogério Salvador, Maurício Ferreira Magalhães** Nesse artigo inicialmente os autores falam sobre o conceito de redes definidas por software, começam falando sobre a arquitetura da internet como o crescimento dela foi causando mudanças e como ela se tornou comercial, e como isso causou o que os autores chamam de engessamento da internet onde ocorre implementações integradas verticalmente baseadas em software fechado sobre hardware proprietário. A ideia é que no novo modelo possa mover grande parte da lógica de tomada de decisão dos dispositivos de rede para controladores externos. Os autores introduzem os princípios da tecnologia OpenFlow, contam um pouco da sua história de criação que foi para atender à demanda de validação de novas propostas de arquiteturas e protocolos de rede. Um conceito importante citado pelos autores é sobre como um fluxo é constituído, que é feito pela combinação de campos do cabeçalho do pacote a ser processado pelo dispositivo. Seguindo a leitura entende-se que uma rede programável com OpenFlow é constituída de equipamentos de rede que são habilitados para que o estado das tabelas de fluxos possa ser instalado através de um canal seguro, conforme as decisões de um controlador em software. São citados alguns cenários possíveis para aplicação como redes corporativas, domésticas, celulares, backbone. Por fim, o trabalho apresenta a arquitetura RouteFlow como exemplo de inovação para a função de roteamento em um ambiente aberto (open source), executado sobre hardware comercial (commodity). Consiste em uma oferta de serviços de roteamento IP remoto de forma centralizada, e que visa um desacoplamento efetivo entre o plano de encaminhamento e o plano de controle.

1.2.3 Softawares

Para desenvolver o trabalho foi necessário utilizar o Stratum, como SO do switch , em conjunto o ONOS que é o controlador que nos fornece diversas informações da rede definida por software,o Mininet para criar a rede virtual, SONIC para realizar testes de compatibilidade, foi outra alternativa além do Stratum, Virtual Box para rodar o ONOS, e computadores com Ubuntu 20.04

1.3 METODOLOGIA DE DESENVOLVIMENTO

Para montar o ambiente RDS teremos que trabalhar com uma arquitetura específica, que utiliza equipamentos específicos que são controlados por um software onde envolve

a preparação do equipamento de rede no caso deste trabalho, são switches ópticos, que não possuem software proprietário, utilizando o ONIE que é uma ferramenta utilizada para se instalar um sistema operacional no switch. Com essa ferramenta podemos fazer a instalação do Stratum que é um sistema operacional de switch de código aberto para redes definidas por software, ele nos fornece um conjunto de interfaces SDN, ele trabalha com P4Runtime e OpenConfig. O Stratum evita que se tenha que trabalhar com aplicações e APIs que são desenvolvidas por empresas privadas que cobram pelo uso. Outra ferramenta que está dentro da nossa arquitetura é o ONOS(Open Network Operating System), é um controlador para se trabalhar com redes definidas por software, ele possui também o código aberto e é líder para a construção de soluções SDN. Temos ele rodando em uma VM para fazer comunicação com o Stratum instalado no switch. Nesse estudo são utilizados três switches ópticos da Edge-core do modelo AS7712-32X. Com o ambiente pronto conseguimos pré-carregar tabelas de endereços MAC nesses switches e também outros que conectarem a essa rede experimental. Além de tornar possível inserir códigos, a nível de chip, como de gerenciamento de pacotes baseados em regras como logística, distância/custo, cálculos matemáticos como algoritmos que encontram melhor rota de um ponto a outro e consideram seus custos que rodamos no seu chip.

2 RESULTADOS

2.1 APRESENTAÇÃO DOS RESULTADOS

A rede que foi trabalhada inicialmente foi um switch OpenFlow com stratum e dois host conectados a ele. Como podemos ver na imagem abaixo a topologia desse ambiente.

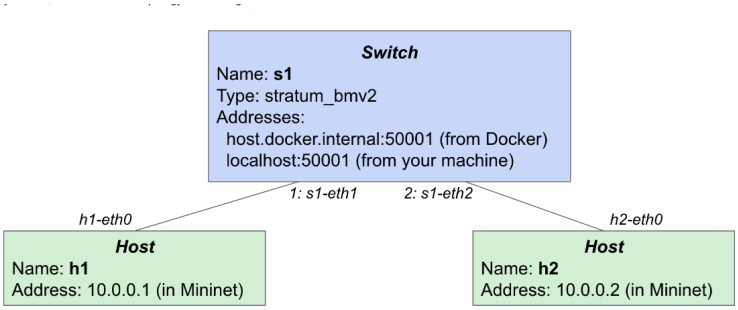


Figura 7 - Topologia da Rede Emulada no Mininet

Conseguimos emular nosso ambiente utilizando o Mininet, inicialmente começamos a trabalhar com um Switch que suporta o software Stratum que faz o controle desse equipamento. Ligado a este switch temos 2 host, inicialmente eles não se comunicam é necessário realizar a configuração da tabela de roteamento do switch utilizando a liguagem p4.

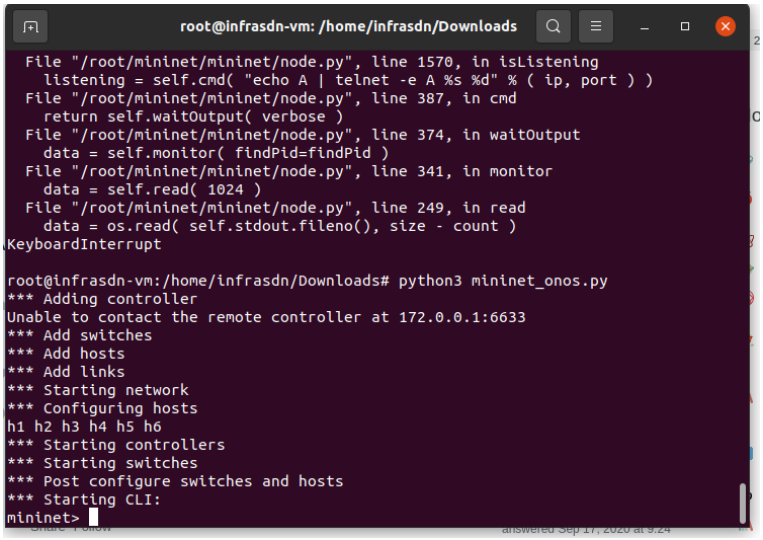


Figura 8 - Rede Emulada no Mininet Rodando

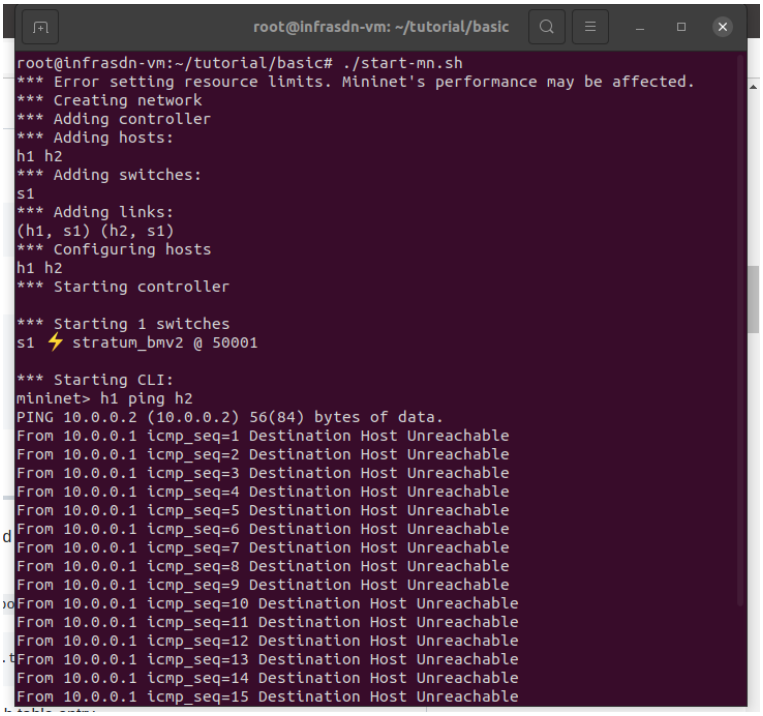


Figura 9 - Ping Host 1 para Host 2 na rede Mininet com erro

Na imagem abaixo podemos ver a configuração da tabela com stratum para permitir a comunicação.

Figura 10 - Configuração da Rota

E depois de configurar o ping começa a funcionar.

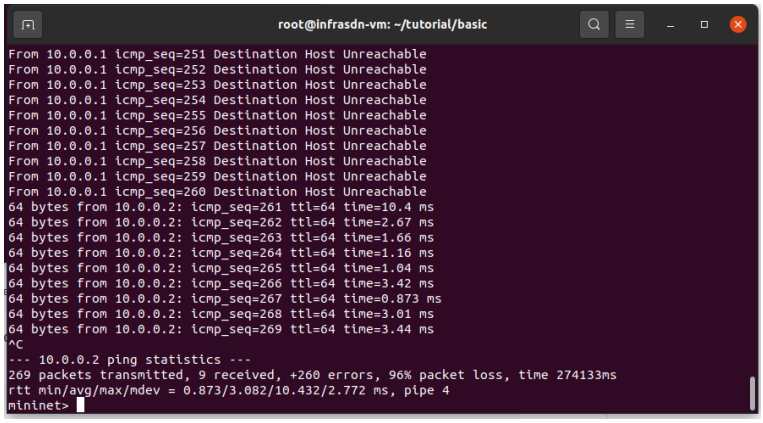


Figura 11 - Ping Host 1 para Host 2 na rede Mininet funcionando

Depois com o gNMI CLI que é um Python CLI que permite fazer chamadas gNMI para um dispositivo Stratum. Por exemplo podemos listar as interfaces do switch para obter informações sobre elas, podemos chamar um contador de pacotes que passam por determinada porta, alem de poder desativar ou ativar alguma porta para não receber mais pacotes. Por fim utilizamos o netconf para mandar as configurações do ambiente no controlador ONOS, em sua interface podemos obter diversas informações da rede, equipamentos que fazem parte, hosts, switch e os links entre eles, a topologia da rede etc.

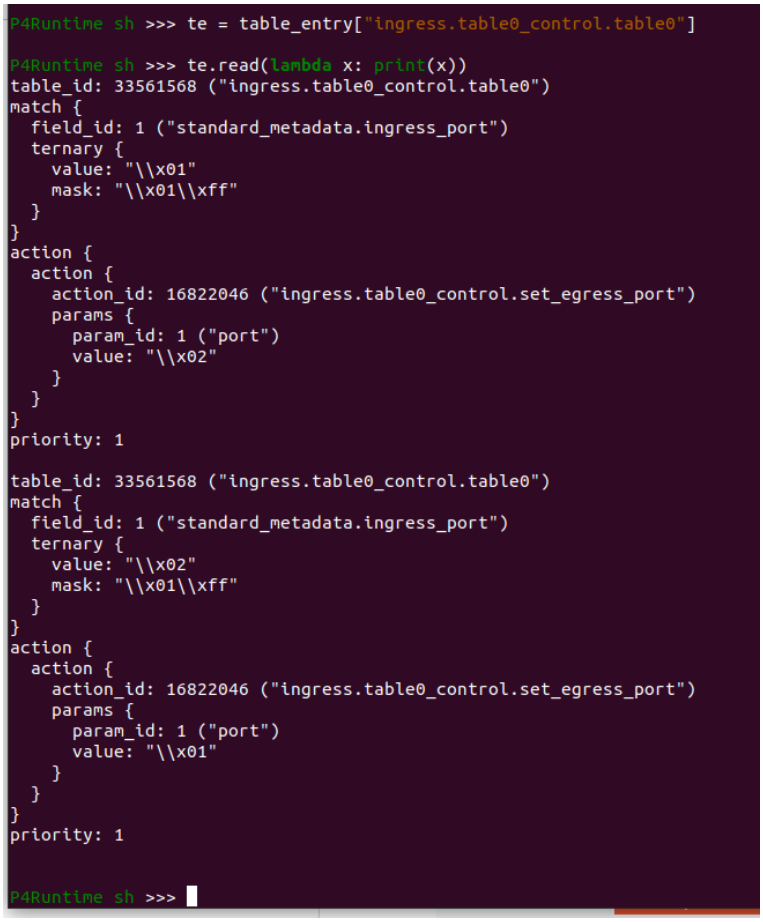


Figura 12 - Interfaces do switch utilizando gNMI CLI

Conseguimos extrair diversas informações do switch como status das portas.

```
***************************
***************************
RESPONSE
update {
  timestamp: 1656882360124957490
  update {
    path {
      elem {
        name: "interfaces"
      }
      elem {
        name: "interface"
        key {
          key: "name"
          value: "s1-eth1"
        }
      }
      elem {
        name: "state"
      }
      elem {
        name: "oper-status"
      }
    }
    val {
      string_val: "UP"
    }
  }
}

***************************
***************************
RESPONSE
sync_response: true

***************************
```

Figura 13 - Status das portas utilizando gNMI CLI

Por fim vemos esse rede no controlador ONOS.

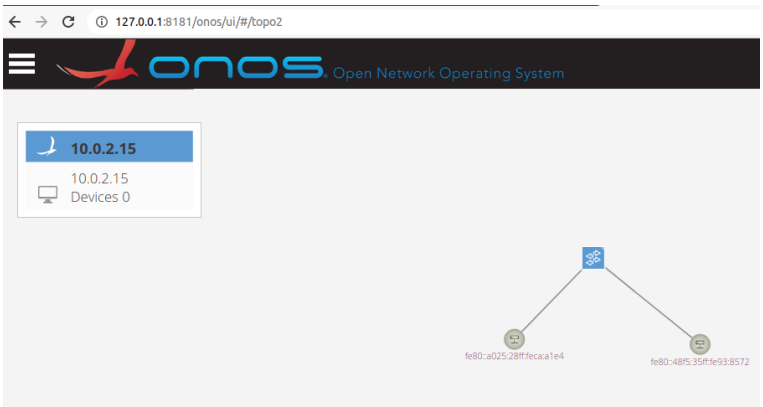


Figura 14 - Rede configurada no ONOS GUI.

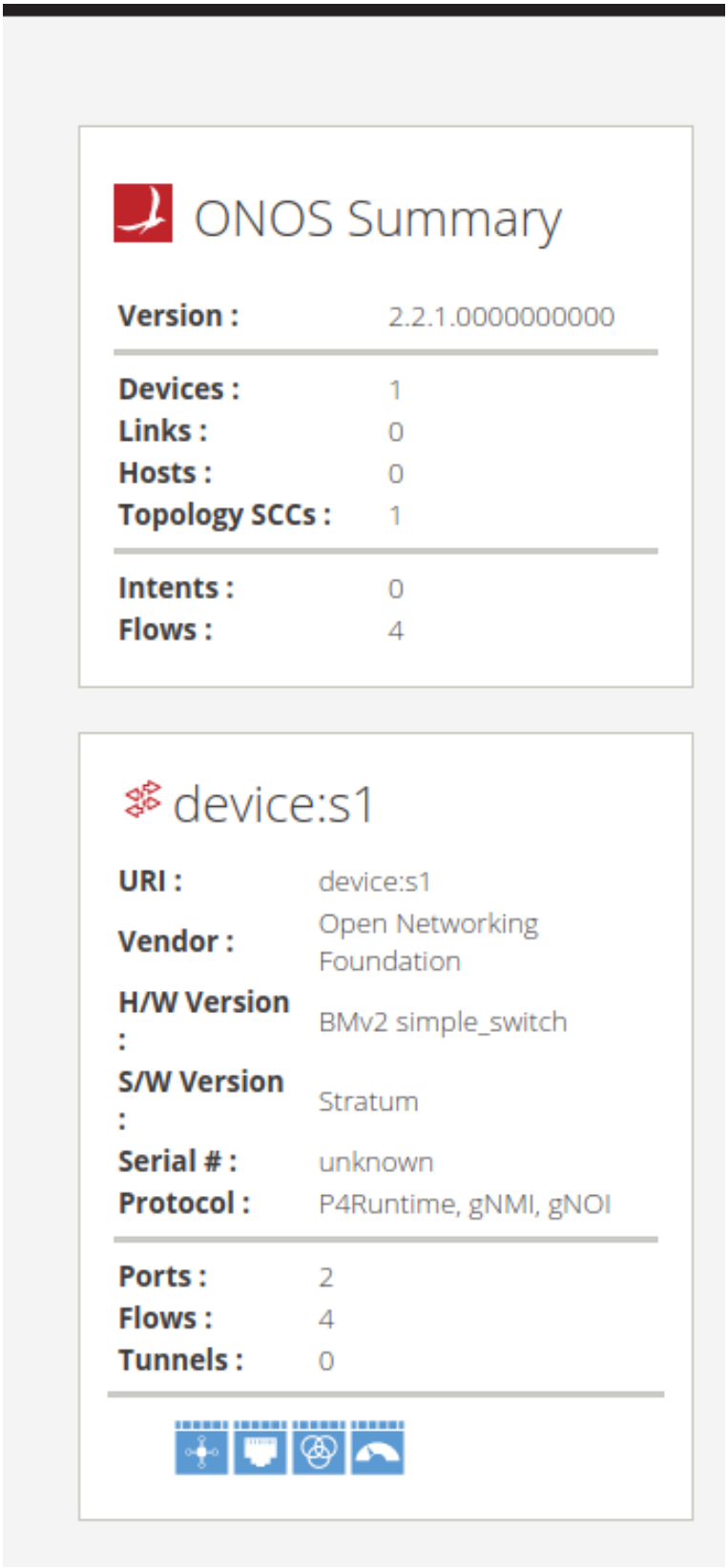


Figura 15 - Versão do ONOS e Switch com Stratum e informando os protocolos.

Algumas aplicações são necessárias ser ativas para toda rede funcionar.

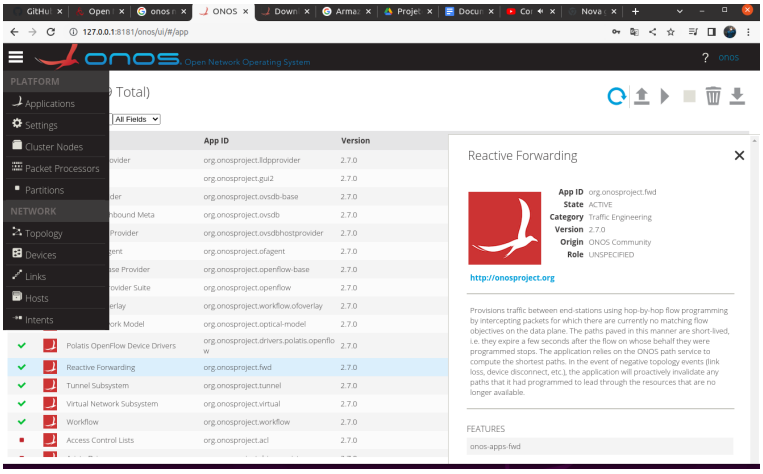


Figura 16 - GUI do ONOS mostrando a parte de Aplicações ativadas.

2.2 DISCUSSÃO DOS RESULTADOS

Conseguimos criar nosso ambiente computacional com um switch com stratum instalado, e 2 hosts conectados a ele. Configuramos as tabelas de roteamento no switch utilizando a linguagem p4, conseguimos extrair informações estatisticas sobre as interfaces do switch e sobre o trafego da rede. Para o ambiente fisíco conseguimos identificar o equipamento a ser utilizado no projeto, ele suporta a linguagem p4. Realizamos testes com o sistema SONIC feito para gerenciar o switch, mas decidimos seguir com o STRATUM, estamos com ele rodando no switch.

2.3 VALIDAÇÃO

O ambiente emulado foi validado utilizando o Controlador ONOS, que consegue enxergar a rede criada e no fornece diversas informações sobre ela, como a topologia, os links entre os equipamentos, os ips e informações dos host e switchs. Futuramente será solicitado a outros pesquisadores para que testem seus códigos no ambiente, como alunos de mestrado e doutorado.

2.4 DESEMPENHO

Iremos realizar testes de desempenho no ambiente computacional emulado, verificar estátistica como a latência de pacotes que trafegam pela rede, comportamento com protocolos e roteamento inseridos.

3 CONCLUSÃO

Com esse trabalho podemos compreender melhor os conceitos de redes definidas por Software e ver uma implementação desse tipo de rede, para contruir uma SDN precisamos de equipamentos específicos que suportam softwares que trabalham com esse tipo de conceito. Com o ambiente virtual configurado conseguimos enxergar os principais desafios para montar o ambiente computacional que suporta programabilidade de SDN, além disso compreendemos o funcionamento do controlador ONOS e como ele nos ajuda a trabalhar com SDN.

REFERÊNCIAS

ANDRÉ. *OpenFlow: uma rede definida por software - Revista Infra Magazine 11*. [S.l.], 2013. Disponível em: <https://www.devmedia.com.br/openflow-uma-rede-definida-por-software-revista-infra-magazine-11/27894>.

FERREIRA, Caio César. *Análise Comparativa de Controladores para Redes Definidas por Software de Classe Carrier Grade*. [S.l.], 2016. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/18139/1/AnaliseComparativaContraladores.pdf>.

FLAUZINO, José. *Emulador Mininet*. [S.l.], 2019. Disponível em: <www.vivaolinux.com.br/artigo/Emulador-de-Redes-Mininet>.

GUEDES, Dorgival. et al. *Redes Definidas por Software: uma abordagem sistemica para o desenvolvimento das pesquisas em Redes de Computadores*. [S.l.], 2012. Disponível em: <https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/minicurso-sdn.pdf>.

LOVATO, Jill. *Software para Rede Aberta na Nuvem (SONiC) migra para a Linux Foundation*. [S.l.], 2022. Disponível em: <https://www.prnewswire.com/news-releases/software-for-open-networking-in-the-cloud-sonic-moves-to-the-linux-foundation-301525831.html>.

https://github.com/stratum/

https://opennetworking.org/onos/

https://opennetworking.org/p4/

https://opennetworking.org/mininet/

https://opennetworking.org/stratum/

https://github.com/stratum/stratum/wiki/Talks

https://opennetworking.org/stratum/

https://github.com/stratum/tutorial/blob/master/README.md

https://wiki.opennetworking.org/display/COM/Stratum

https://wiki.onosproject.org/

https://wiki.opennetworking.org/display/COM/Stratum

https://github.com/stratum/tutorial/blob/master/README.md

https://www.researchgate.net/publication/260346033$_{R}edes_{D}efinidas_{p}or_{S}oftware_{u}ma_{a}bordagem_{s}ist$

https://intrig.dca.fee.unicamp.br/wp-content/papercite-data/pdf/rothenberg2010openflow.pdf