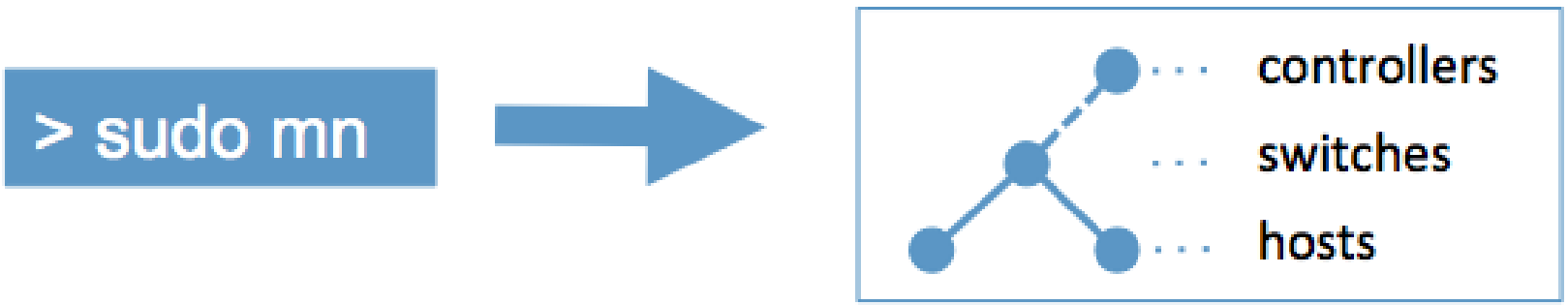
Conceitos e Tecnologias para Dispositivos Conectados (C115)

Prof. Samuel Baraldi Mafra

Inatel
Instituto Nacional de Telecomunicações

Mininet



`http://mininet.org/`

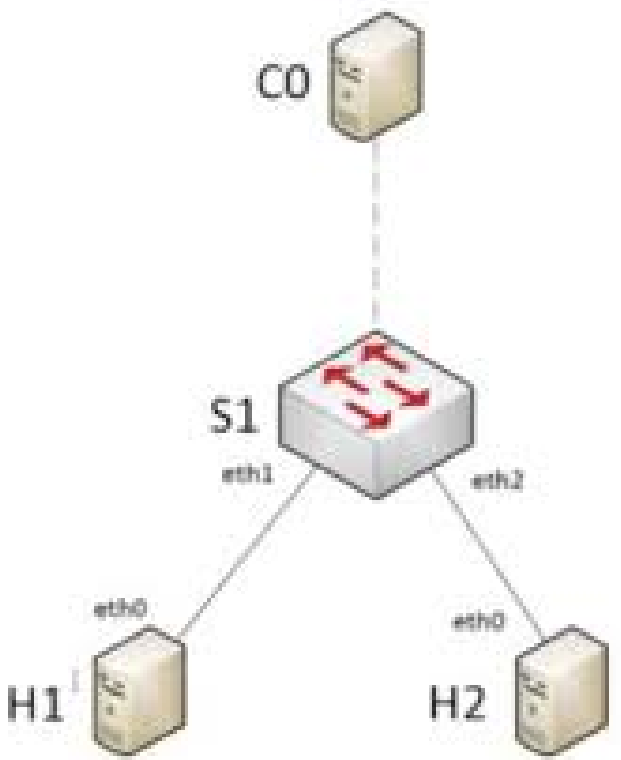
Analisar as tabelas de fluxos

Comandos para visualização e alteração de fluxos:

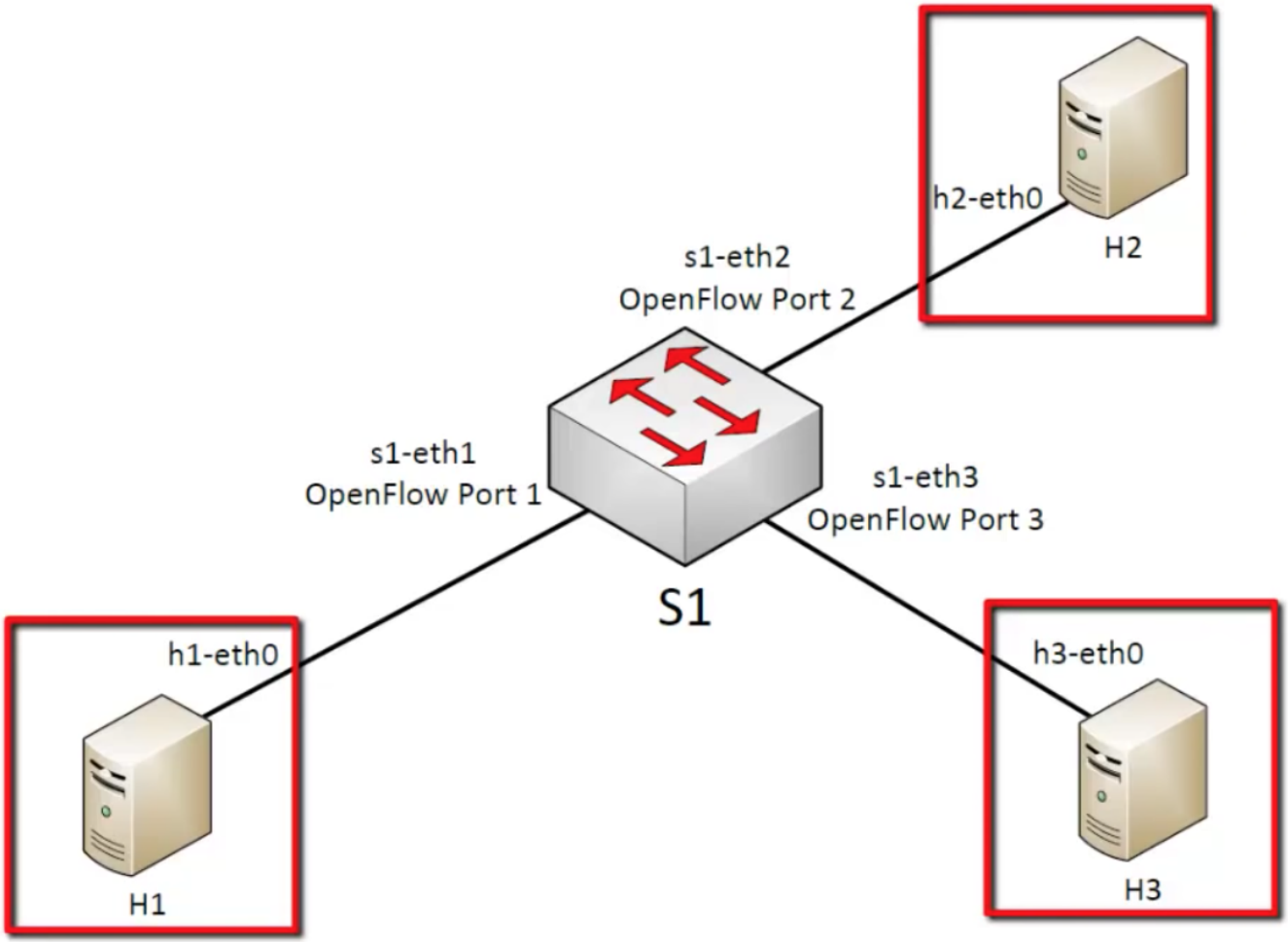
- Open vSwitch (OVS)
- ovs-ofctl: utilitário de linha de comando que envia mensagens OpenFlow rápidas, útil para visualizar a porta do switch e estatísticas de fluxo ou inserir entradas de fluxo manualmente.
- ovs-ofctl show s1: mostra as informações do switch s1
- ovs-ofctl dump-flows s1: Mostra a tabela de fluxos de s1.
- ovs-ofctl add-flow s1 in_port=1,actions=output:2 - Inclusão de regras
- ovs-ofctl del-flows s1 -Exclusão de regras
- tcpdump -XX -n -i h2-eth0 um utilitário para imprimir os pacotes vistos por um host:

teste na topologia padrão

- sudo mn
- xterm h1
  - sudo ovs-ofctl show s1
  - sudo ovs-ofctl dump-flows s1



sudo mn --topo=single,3 --controller=none --mac



- sudo mn --topo=single,3 --controller=none --mac
- xterm h1
  - sudo ovs-ofctl show s1
  - sudo ovs-ofctl add-flow s1 action=normal
- pingall
  - sudo ovs-ofctl dump-flows s1
  - sudo ovs-ofctl del-flows s1
  - sudo ovs-ofctl dump-flows s1
- pingall

fluxos camada 1

- sudo ovs-ofctl add-flow s1 in_port=1,actions=output:2
- sudo ovs-ofctl add-flow s1 in_port=2,actions=output:1
- sudo ovs-ofctl dump-flows s1
- Teste ping para h2 e h3
- tcpdump -XX -n -i h2-eth0
- tcpdump -XX -n -i h3-eth0
- sudo ovs-ofctl add-flow s1 in_port=3,actions=flood
- sudo ovs-ofctl add-flow s1 in_port=2,actions=drop
- sudo ovs-ofctl del-flows s1

Fluxos camada 2 MAC

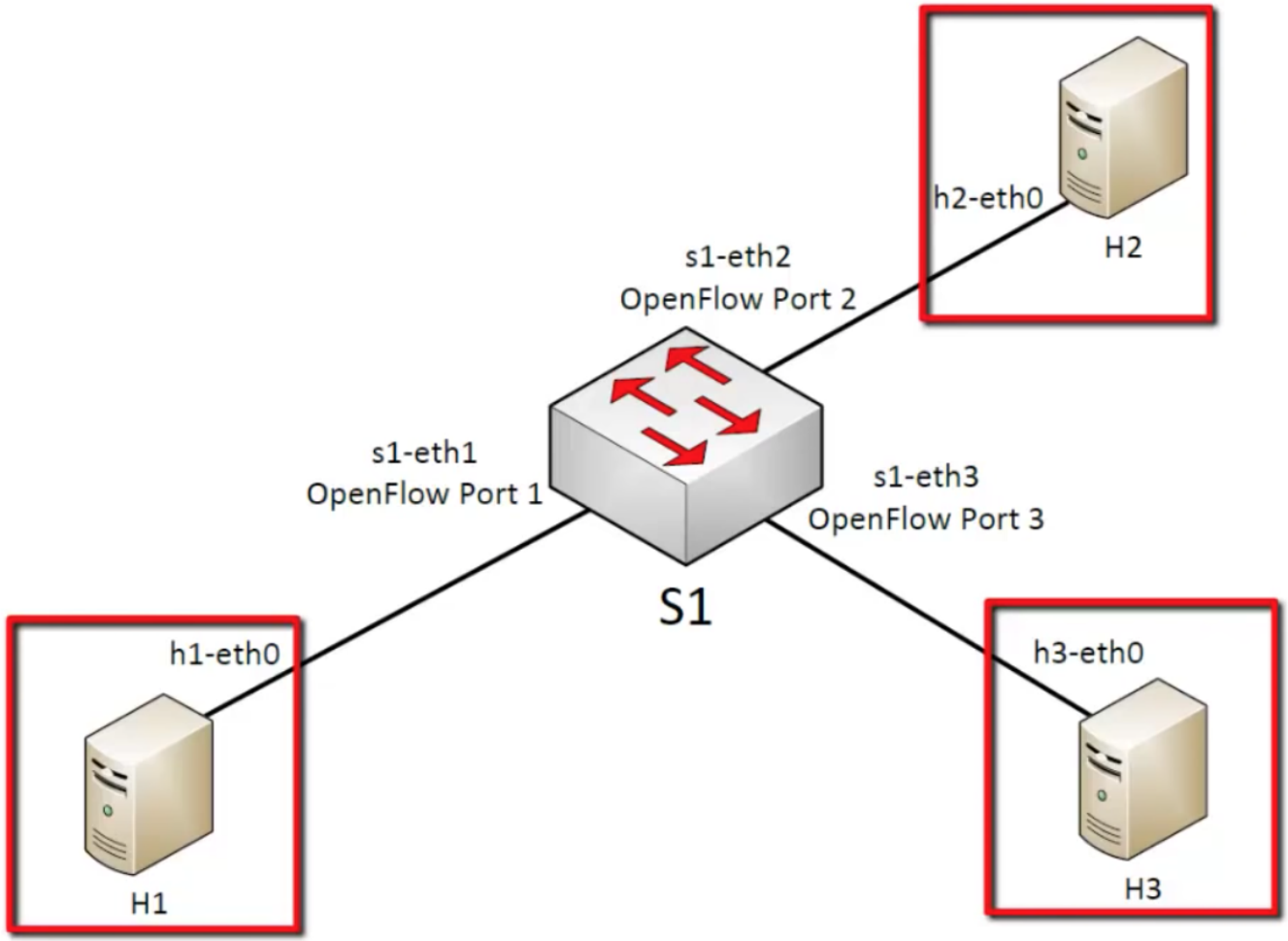
- sudo ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,actions=output:2
- sudo ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,actions=output:1
- Verificar pingall

Fluxos camada 2 MAC

- sudo ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,actions=output:2
- sudo ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,actions=output:1
- Verificar pingall
- Ping trabalha a nivel de IP e a primeira coisa que faz é usar o ARP para descobrir os endereços MAC dos diferentes hosts. Nosso switch, com as regras que possui atualmente, está filtrando o tráfego de dados ARP.
- sudo ovs-ofctl add-flow s1 dl_type=0x806,nw_proto=1,action=flood
- pingall

Controlador POX

sudo mn --topo single,3 --mac --switch ovsk --controller remote



- Abrir duas conexões com o mininet;
- Executar na pasta pox ./pox.py log.level --DEBUG misc.of_tutorial: Iniciar o pox em modo debug exemplo of_tutorial da pasta misc;
  - cd pox
  - ./pox.py log.level --DEBUG misc.of_tutorial
- Em outro terminal Executar sudo mn --topo single,3 --mac --switch ovsk --controller remote
- Modo HUB.

- xterm h1 h2 h3
- tcpdump -XX -n -i h2-eth0
- tcpdump -XX -n -i h3-eth0
- ping -c1 10.0.0.2
- Os pacotes de ping agora estão indo para o controlador, que os inunda em todas as interfaces, exceto a de envio. Você deve ver pacotes ARP e ICMP idênticos correspondentes ao ping em ambos os xterms executando tcpdump.
- É assim que um hub funciona; ele envia todos os pacotes para todas as portas da rede.

Agora, veja o que acontece quando um host inexistente não responde. De h1 xterm

- ping -c1 10.0.0.5

learning switch

- Executar o controlador: ./pox.py log.level --DEBUG forwarding.l2_learning
  - cd pox
  - ./pox.py log.level --DEBUG forwarding.l2_learning
- Executar sudo mn --topo single,3 --mac --switch ovsk --controller remote
- Executar pings
- Fases do switch
- (1) hub,
- (2) controller-based Ethernet learning switch, and
- (3) flow-accelerated learning switch.

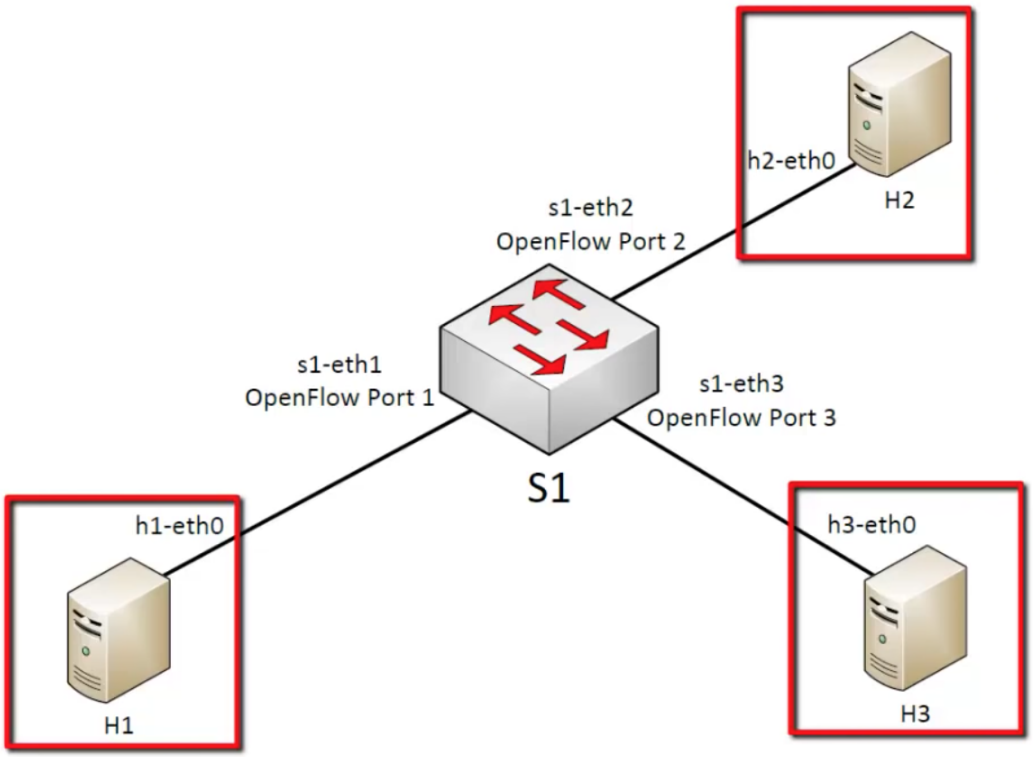
Topologias customizadas

Interação computador e mininet

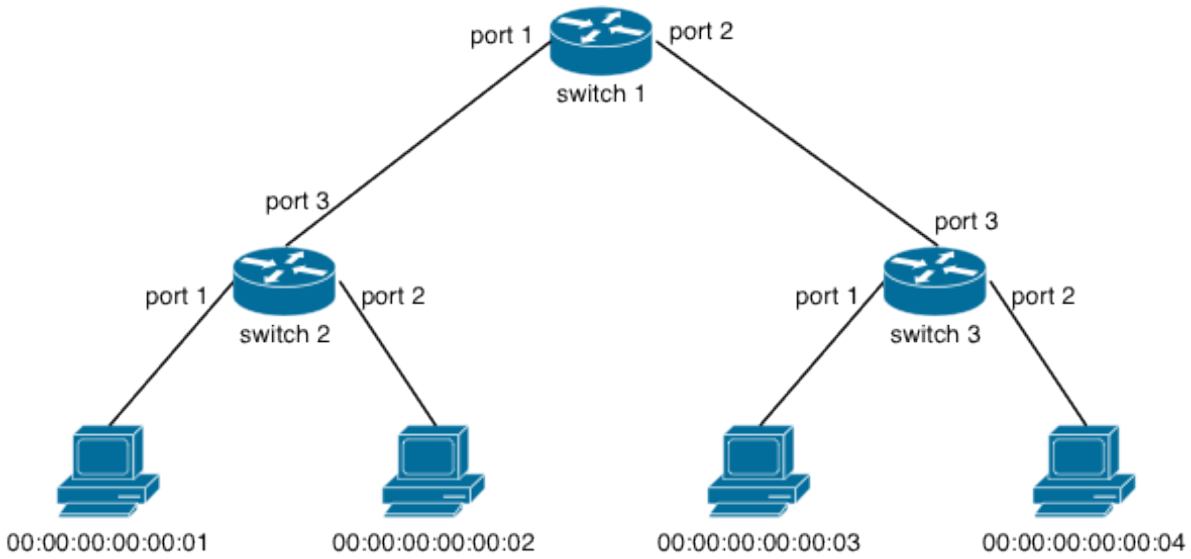
- Github;
  - `https://github.com/git-guides/install-git`
- firefox.
  - sudo apt-get install firefox;
  - firefox &

Objetivo: Criação e execução de uma topologia customizada no mininet

sudo mn --topo=single,3

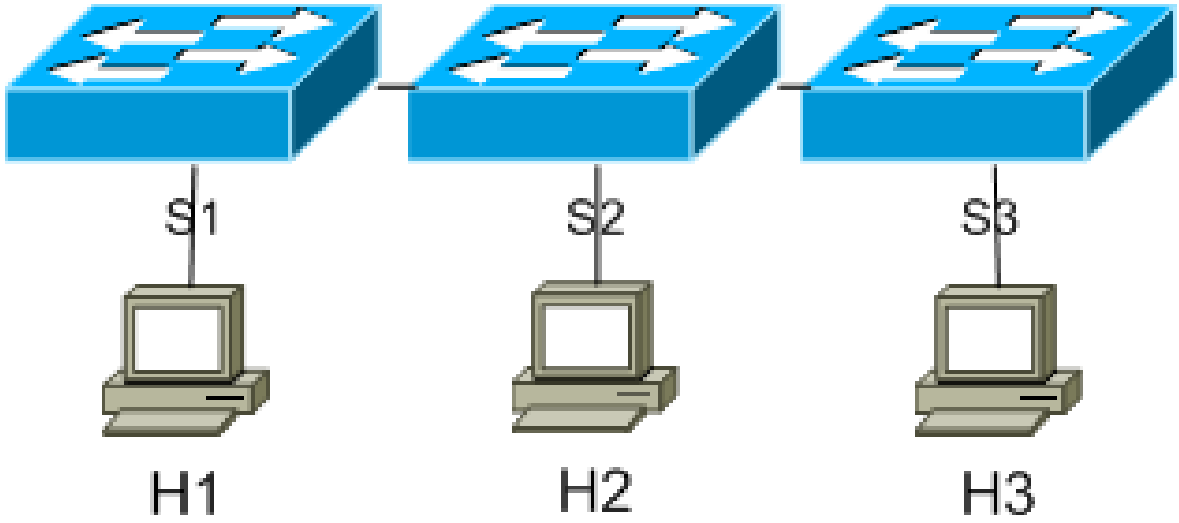


sudo mn --topo=tree,2,2



- O primeiro termo refere-se a o numero de camadas de switchs enquanto o segundo especifica a ramificação em cada switch

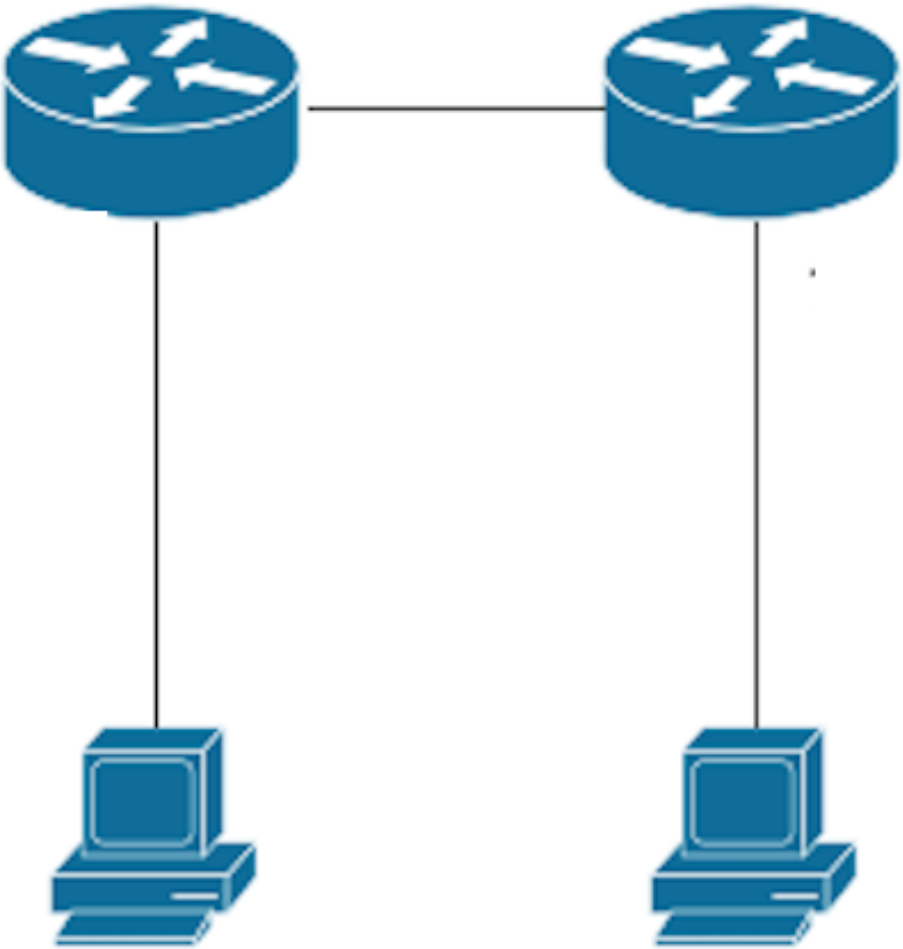
sudo mn --topo=linear,3



Linear (3)

Vamos partir de uma configuração conhecida e criar uma topologia customizada.

Configuração linear: 2 host 2 switch



Comandos para criação de topologia customizada em python:

- Criar os hosts- função addHost: host1 = self.addHost('h1');
- Criar o switch- função addSwitch: Switch1 = self.addSwitch('s1');
- Criar links função addLink: self.addLink(host1, Switch1)

Abrir arquivo da topologia customizada: topo-2sw-2host.py

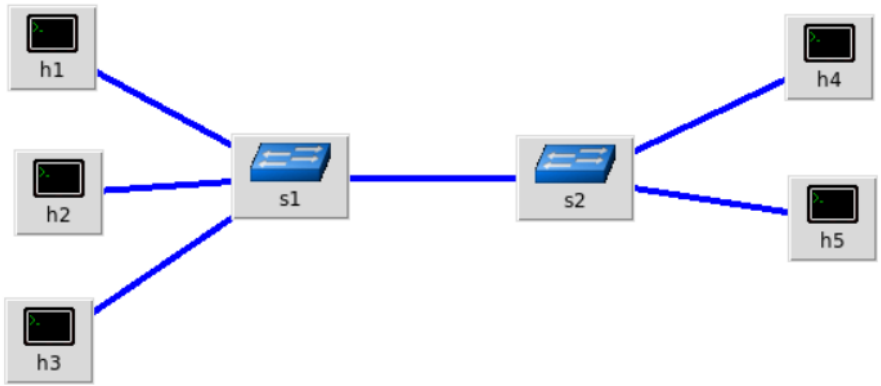
- Para executar uma topologia customizada execute:
- sudo mn --custom localização do arquivo --topo mytopo
- Para a topologia presente sudo mn --custom topo-2sw-2host.py --topo mytopo
- Vamos executar primeiro considerando o controlador do mininet.
- Vamos descobrir algumas informações da topologia e depois fazer testes com ping e iperf.

- Verificação dos nós: nodes;
- Verificação das interfaces dos links: net;
- Verificação das conexões e endereços IP: dump.
- Verificação das configurações de um host: h1 ifconfig -a

- Teste de conexão: pingall
- h1 ping h2 ;
- iperf: analisa a largura de banda alcançada.

Lembrar sempre que finalizar uma simulação de executar
sudo mn -c
para apagar a ultima análise.

- Vamos criar uma nova topologia com 3 hosts do lado esquerdo e 2 hosts do lado direito;



- Vamos alterar o arquivo já existente para criar a nova topologia.
- Para isso devemos incluir mais dois host do lado esquerdo e um host do lado direito.
- Conectar os novos hosts aos switchs.
- Salvar o arquivo no mesmo diretório (topo-3host-2sw-2host.py)

- sudo mn --custom topo-3host-2sw-2host.py --topo mytopo
- Verificação dos nós: nodes;
- Verificação das interfaces dos links: net;
- Verificação das conexões e endereços IP: dump.

- Teste de conexão: pingall
- h1 ping h1a ;
- h1 ping h2a;

- Vamos executar agota com um controlador manual:
- sudo mn --custom topo-3host-2sw-2host.py --topo mytopo --controller=none --mac
- xterm h1 h2 h1a
- Mostrar informações das portas dos switches sudo ovs-ofctl show s3 e sudo ovs-ofctl show s4 (verificar qual é o label dos switchs criados)
- Switch normal: sudo ovs-ofctl add-flow s3 action=normal
- sudo ovs-ofctl add-flow s4 action=normal
- Testar ping
- Verificar os pacotes na entrada de h2: tcpdump -XX -n -i h2-eth0
- tcpdump -XX -n -i h1a-eth0

- Mostra tabela de fluxos: sudo ovs-ofctl dump-flows s3
- Apaga os fluxos do switch sudo ovs-ofctl del-flows s3

Fluxos camada 2 MAC

- Executar ifconfig nos xterms dos hosts para descobrir os endereços MAC.
- Protocolo ARP sudo ovs-ofctl add-flow s3 dl_type=0x806,nw_proto=1,action=flood
- sudo ovs-ofctl add-flow s4 dl_type=0x806,nw_proto=1,action=flood
- sudo ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,actions=output:2
- sudo ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,actions=output:1

Fluxos camada 2 MAC

- sudo ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04,actions=output:4
- sudo ovs-ofctl add-flow s4 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04,actions=output:2
- sudo ovs-ofctl add-flow s4 dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01,actions=output:1
- sudo ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:00:01,actions=output:1
- Para completar a tabela incluir dados para todos os fluxos.