

# Sistemas Distribuídos

2016.1

---

PROF. MARCIAL PORTO FERNANDEZ  
MARCIAL@LARCES.UECE.BR

PROF. ANDRÉ RIBEIRO CARDOSO  
ANDREC@LARCES.UECE.BR

# A. Introdução ao Mininet

---

# Sumário

---

## Ensino de Computação

### Plataformas para Redes e SDs

- Emulação de Redes
- Emulação de Redes usando Virtualização
- Exemplo de Virtualização de rede usando servidor

### Mininet

- Usando Mininet

# Sumário

---

## Ensino de Computação

### Plataformas para Redes e SDs

- Emulação de Redes
- Emulação de Redes usando Virtualização
- Exemplo de Virtualização de rede usando servidor

### Mininet

- Usando Mininet

# Ensino de Computação

---

Computação é uma ciência aplicada.

Não adianta teoria sem uma aplicação

- As teorias mais recentes tem mais de 30 anos....

Não adianta aplicação sem o fundamento teórico

- A não ser que você queira resolver apenas problemas triviais já conhecidos.

Para aprender Sistemas Distribuídos é necessário ter uma experiência prática.

# Ensino de Computação

---

Aprender fazendo (“Learning by doing”) deve ser a metodologia para ensino da Ciência da Computação.

Em Sistemas de Computação isso pode ser atingido construindo sistemas, modificando, usando e experimentando sistemas em funcionamento.

Redes e Sistemas Distribuídos exigem laboratórios complicados incluindo múltiplos servidores, switches e desktops.

Que tal um sistema para avaliar Sistema Distribuído parecido com o de programação?

# Sumário

---

Ensino de Computação

## Plataformas para Redes e SDs

- Emulação de Redes
- Emulação de Redes usando Virtualização
- Exemplo de Virtualização de rede usando servidor

## Mininet

- Usando Mininet

# Plataformas para ensino e pesquisa de Redes e SD

Plataforma	Vantagens	Desvantagens
Testbed	Execução rápida, preciso (sistema real)	Caro, difícil de configurar, difícil de modificar
Simulador	Barato, flexível, fácil de construir cenários, tempo virtual (tempo é o mesmo independente do hardware utilizado), fácil de coletar informações	Aplicações precisam ser modificadas, precisão duvidosa, pode ter execução lenta, tempo virtual pode causar distorções
Emulador	Barato, flexível, razoavelmente preciso, execução razoavelmente rápida	Mais lento que testbed, precisão pode ser baixa devido a multiplexação, dificuldade de coletar informações



# Plataformas para experimentação de redes

---

## Emuladores baseados em container:

- CORE, virtual Emulab, Trellis, Imunes, ns-3 (emulation), **Mininet**

## Emuladores Full-VM

- DieCast, GNS3

## Emuladores UML (User Mode Linux)

- NetKit

## Simuladores:

- Ns-3, OPNET

## Testbeds:

- Emulab, GENI, PlanetLab, ORBIT, PanLab, G-Lab

# Sumário

---

Ensino de Computação

Plataformas para Redes e SDs

- **Emulação de Redes**
- Emulação de Redes usando Virtualização
- Exemplo de Virtualização de rede usando servidor

Mininet

- Usando Mininet

# Emulação de Redes

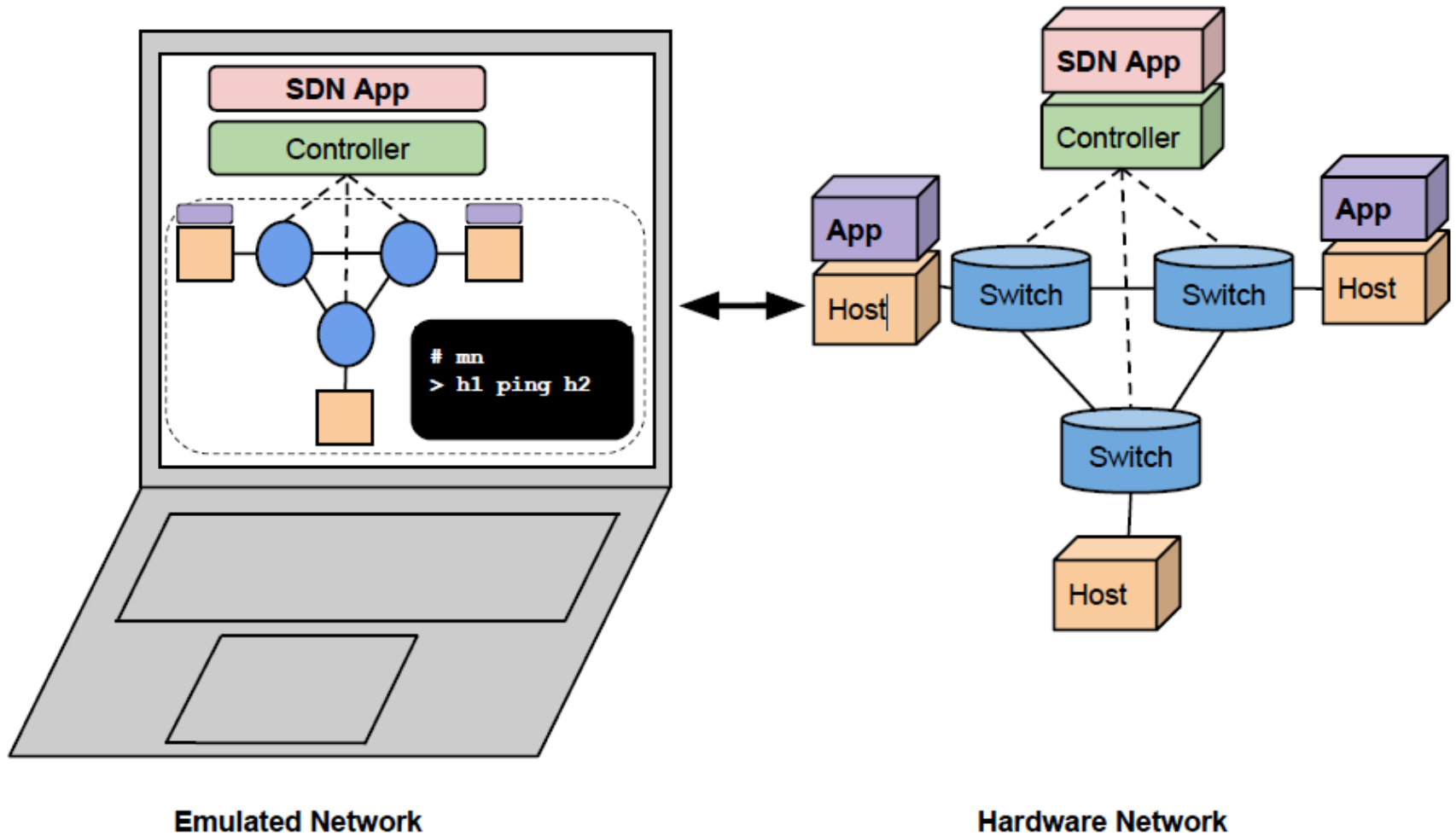
---

Emulação, ou executando em um emulador, significa executar código não modificado interativamente sobre um hardware virtual em um computador PC, provendo realismo e facilidade de uso a um custo baixo, com algumas limitações (velocidade, detalhamento).

Um testbed é mais rápido e mais real, mas custa caro.

Um simulador é barato, tem mais detalhes, porém a execução é lenta, exige modificação no código (que não pode ser usado na vida real) e carece de fidelidade.

# Aplicação real



# Sumário

---

Ensino de Computação

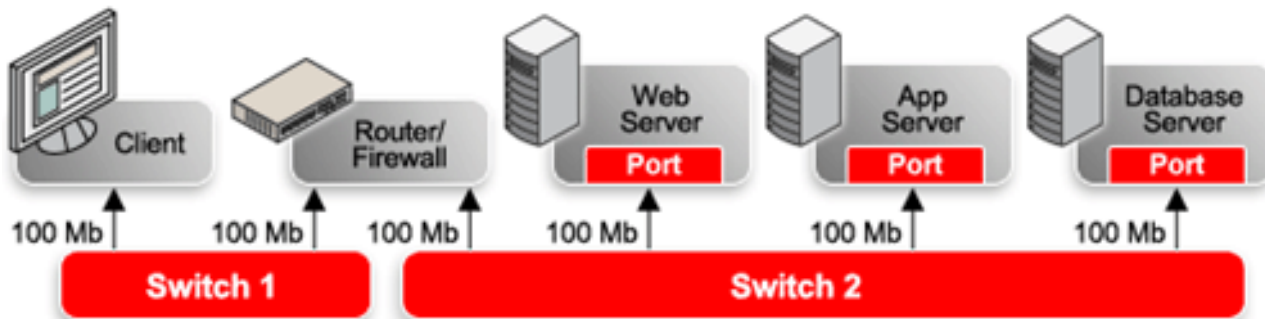
Plataformas para Redes e SDs

- Emulação de Redes
- **Emulação de Redes usando Virtualização**
- Exemplo de Virtualização de rede usando servidor

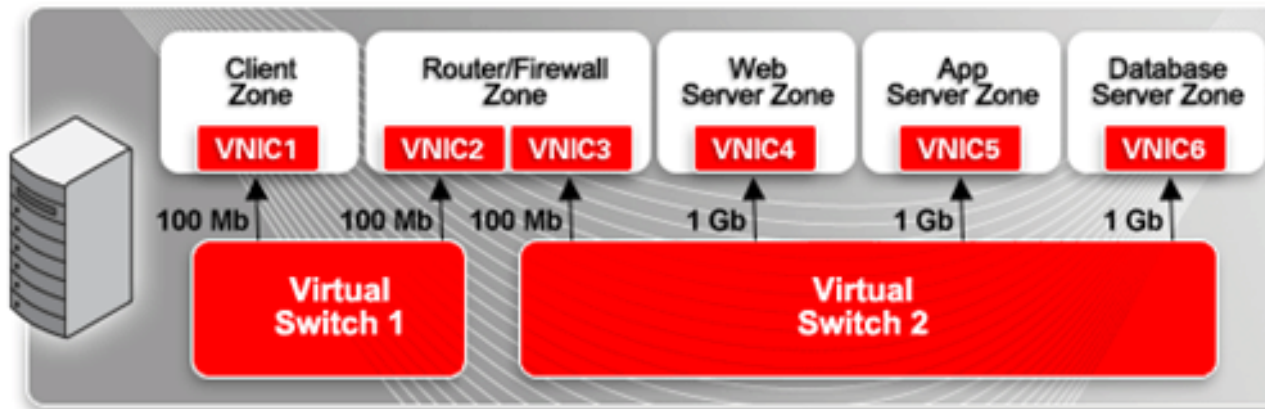
Mininet

- Usando Mininet

# Virtualização de Redes usando servidor



AFTER CONSOLIDATION



# Sumário

---

Ensino de Computação

Plataformas para Redes e SDs

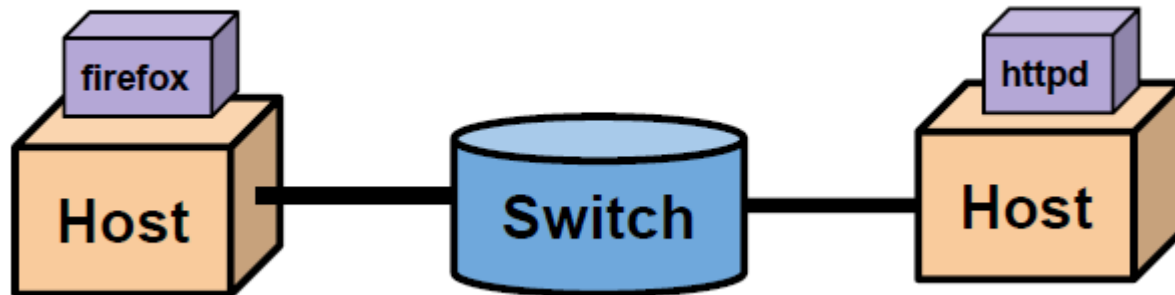
- Emulação de Redes
- Emulação de Redes usando Virtualização
- **Exemplo de Virtualização de rede usando servidor**

Mininet

- Usando Mininet

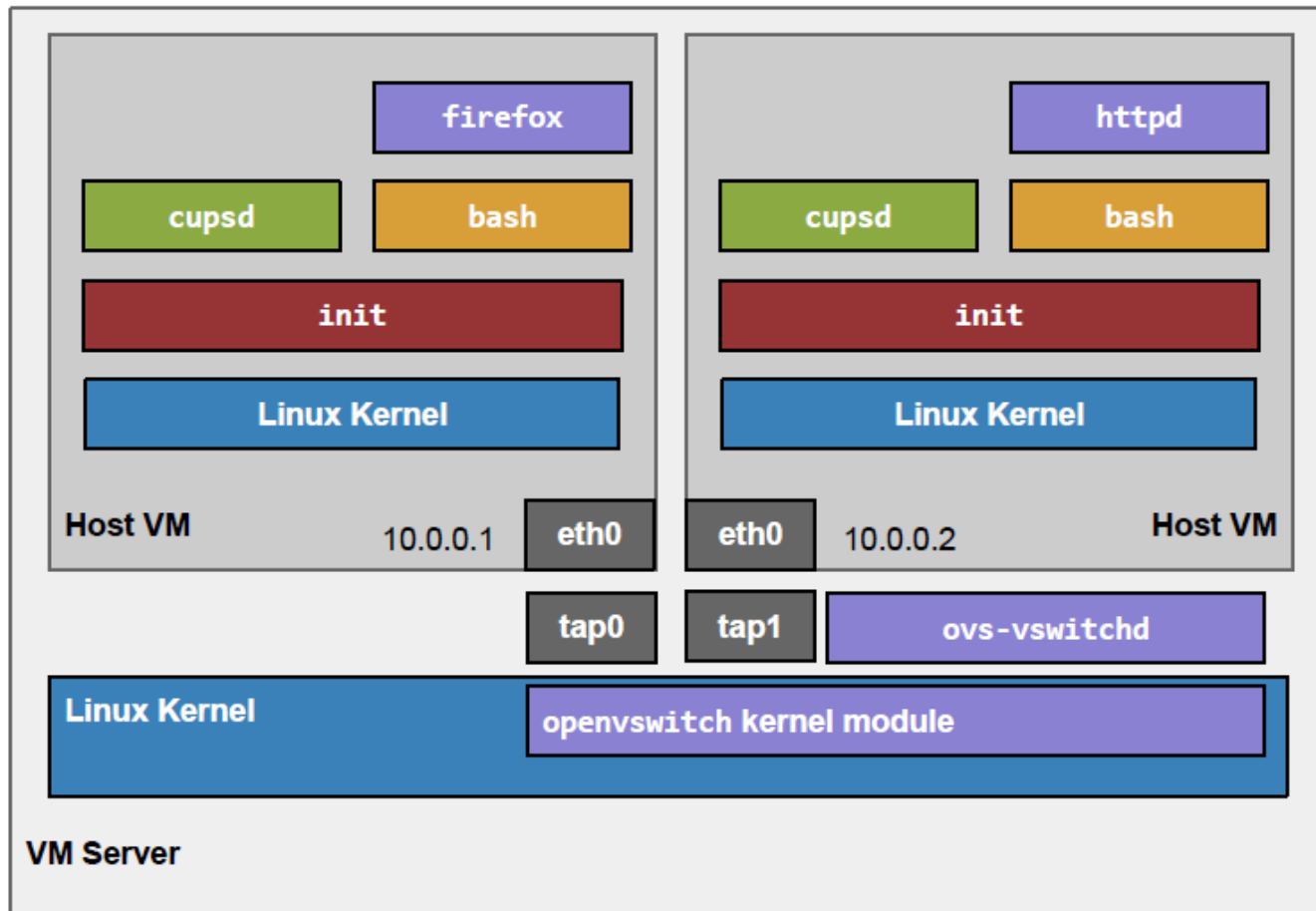
# Uma rede simples

---

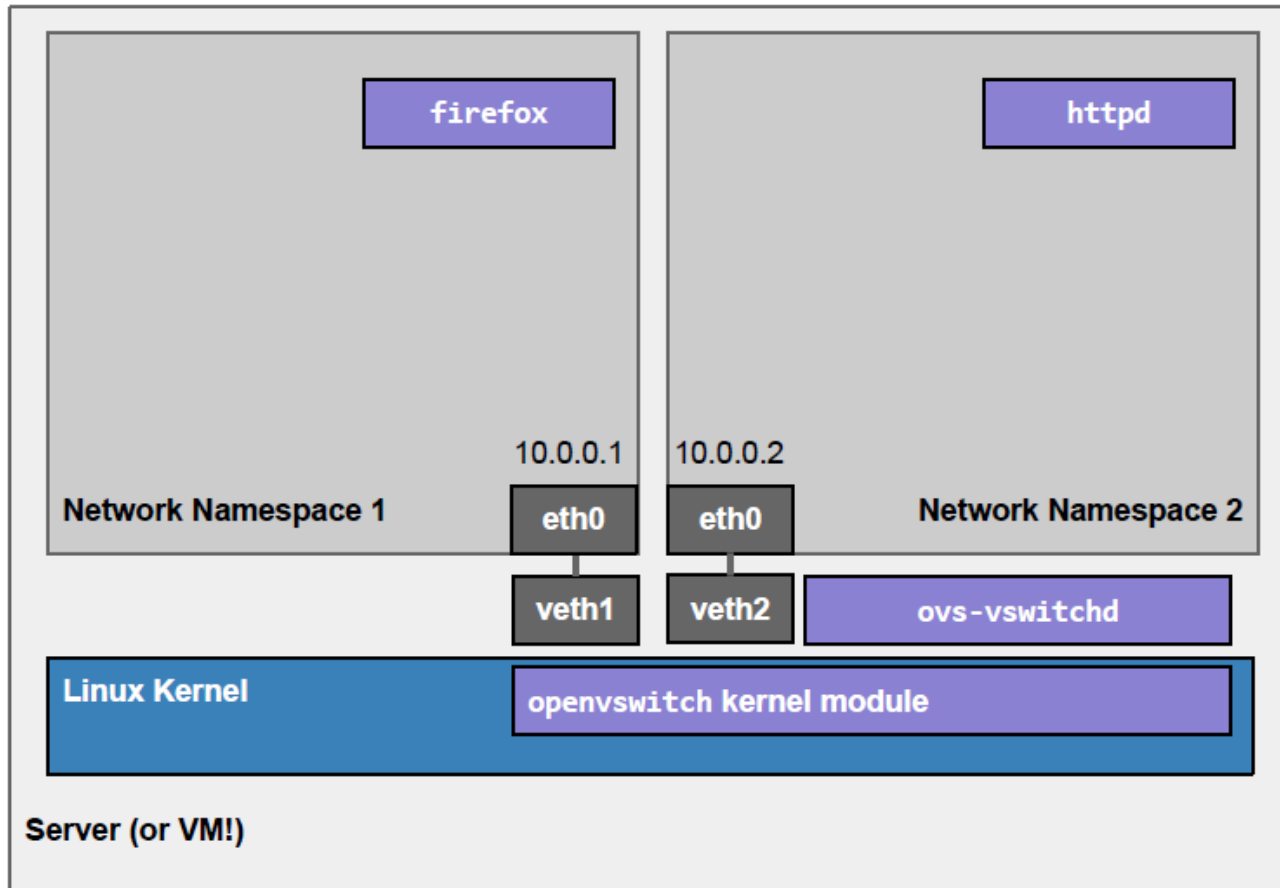




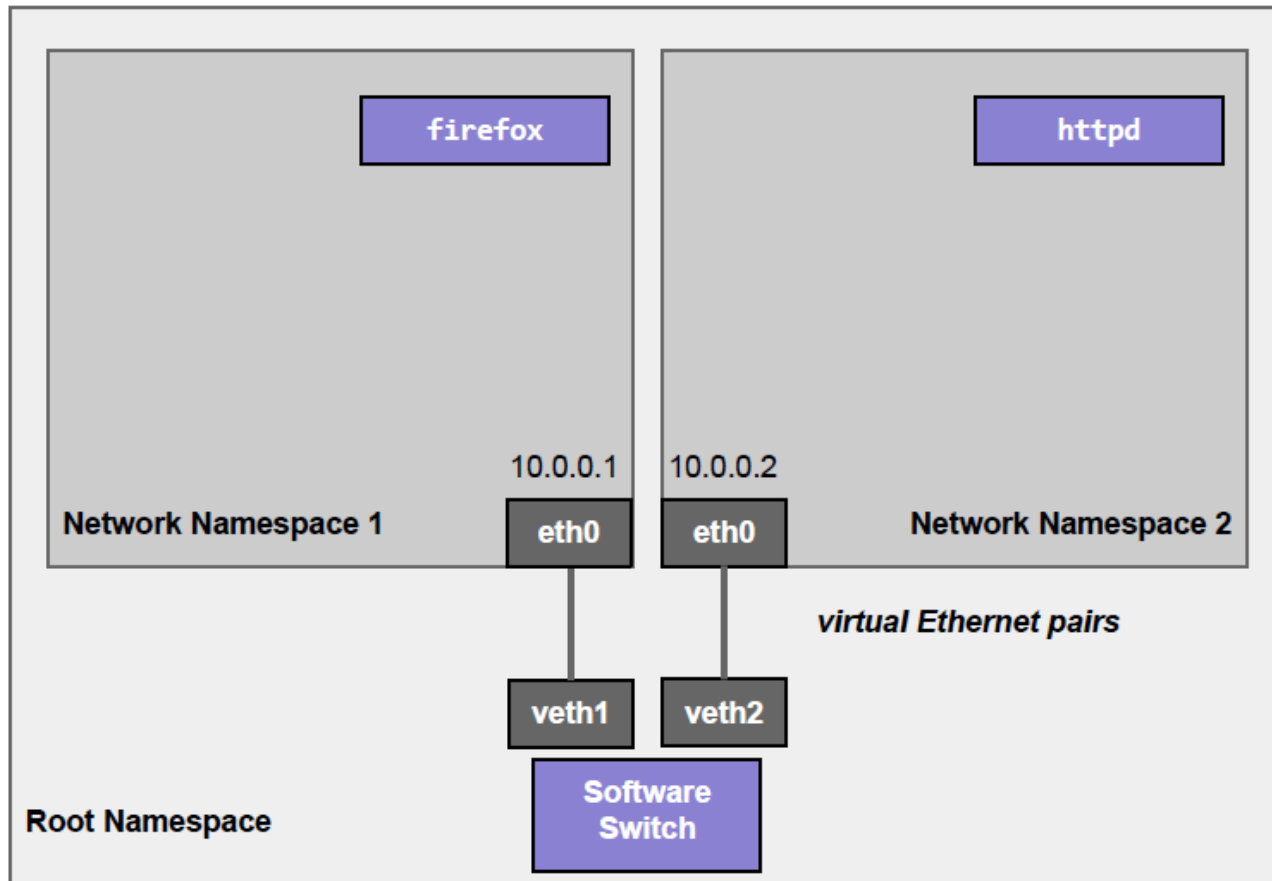
# Rede simples usando Full Virtualization



# Rede simples usando Virtualização Leve (Container)



# Rede Ethernet Virtual usando Namespace



# Criando rede com Linux

```
sudo bash
```

```
# Create host namespaces
```

```
ip netns add h1
```

```
ip netns add h2
```

```
# Create switch
```

```
ovs-vsctl add-br s1
```

```
# Create links
```

```
ip link add h1-eth0 type veth peer name s1-eth1
```

```
ip link add h2-eth0 type veth peer name s1-eth2
```

```
ip link show
```

```
# Move host ports into namespaces
```

```
ip link set h1-eth0 netns h1
```

```
ip link set h2-eth0 netns h2
```

```
ip netns exec h1 ip link show
```

```
ip netns exec h2 ip link show
```

```
# Connect switch ports to OVS
```

```
ovs-vsctl add-port s1 s1-eth1
```

```
ovs-vsctl add-port s1 s1-eth2
```

```
ovs-vsctl show
```

```
# Set up OpenFlow controller
```

```
ovs-vsctl set-controller s1 tcp:127.0.0.1
```

```
ovs-controller ptcp: &
```

```
ovs-vsctl show
```

```
# Configure network
```

```
ip netns exec h1 ifconfig h1-eth0 10.1
```

```
ip netns exec h1 ifconfig lo up
```

```
ip netns exec h2 ifconfig h2-eth0 10.2
```

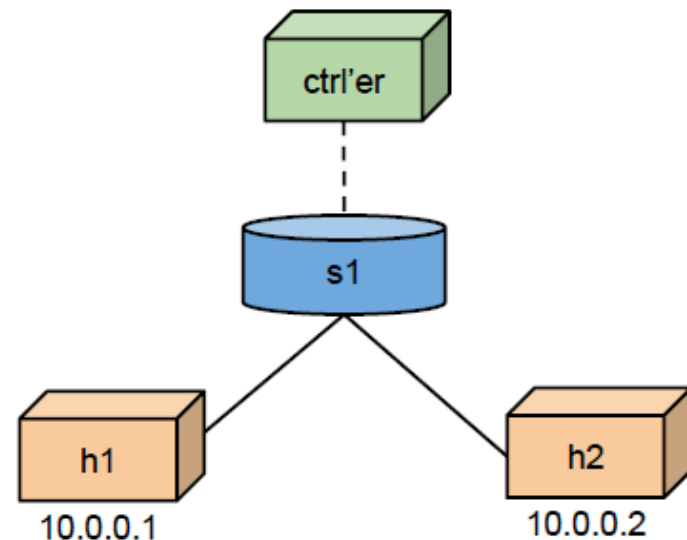
```
ip netns exec h1 ifconfig lo up
```

```
ifconfig s1-eth1 up
```

```
ifconfig s1-eth2 up
```

```
# Test network
```

```
ip netns exec h1 ping -c1 10.2
```



# Seria ótimo se....

---

...tivesse uma ferramenta de linha de comando simples e/ou uma API que construísse a rede automaticamente?

...essa ferramenta permitisse criar facilmente topologias de vários tamanhos, com centenas de nós, switches e hosts?

...permitisse criar links reais com banda limitada, delay, taxa descarte?

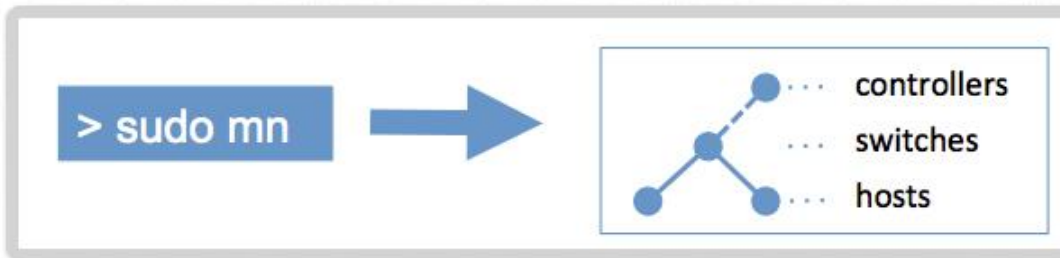
...pudesse utilizar ferramentas de teste e monitoração conhecidos como, Wireshark, nmap, iperf, etc?

...estivesse incluído no Ubuntu?

# Mininet

An Instant Virtual Network on your Laptop (or other PC)

Mininet creates a **realistic virtual network**, running **real kernel, switch and application code**, on a single machine (VM, cloud or native), in seconds, with a single command:



Because you can easily [interact with](#) your network using the Mininet [CLI](#) (and [API](#)), [customize](#) it, [share](#) it with others, or [deploy](#) it on real hardware, Mininet is useful for [development](#), [teaching](#), and [research](#).

Mininet is also a great way to develop, share, and experiment with [OpenFlow](#) and Software-Defined Networking systems.

Mininet is actively developed and supported, and is released under a permissive BSD Open Source license. We encourage you to [contribute](#) code, bug reports/fixes, documentation, and anything else that can improve the system!

## Get Started

[Download](#) a Mininet VM, do the [walkthrough](#) and run the [OpenFlow tutorial](#).

## Support

Read the [FAQ](#), read the [documentation](#), and join our mailing list, [mininet-discuss](#).

## Contribute

File a [bug](#), download the [source](#), or submit a [pull request](#) - all on GitHub.

## Mininet

### [Get Started](#)

### [Sample Workflow](#)

### [Walkthrough](#)

### [Overview](#)

### [Download](#)

### [Documentation](#)

### [Videos](#)

### [Source Code](#)

### [Apps](#)

### [FAQ](#)

### [Wiki](#)

### [Papers](#)

### [Support](#)

### [Contribute](#)

### [News Archives](#)

### [Credits](#)

## News

[Announcing Mininet 2.2.0!](#)

[Mininet Tutorial at SIGCOMM](#)

# Sumário

---

Ensino de Computação

Plataformas para Redes e SDs

- Emulação de Redes
- Emulação de Redes usando Virtualização
- Exemplo de Virtualização de rede usando servidor

**Mininet**

- Usando Mininet

# Arquitetura Mininet

---

## Virtualização baseada em container (Namespace)

- Python usado para configuração e orquestração
- C++ compilado para emulação (switches, Linux, Apps)

## Possibilidade de uso:

- Usando a CLI Mininet
- Usando script Python

## Instalação

- Instalação de pacote Debian/Ubuntu
- Baixar uma VM pronta para uso



# Sumário

---

Ensino de Computação

Plataformas para Redes e SDs

- Emulação de Redes
- Emulação de Redes usando Virtualização
- Exemplo de Virtualização de rede usando servidor

Mininet

- Usando Mininet

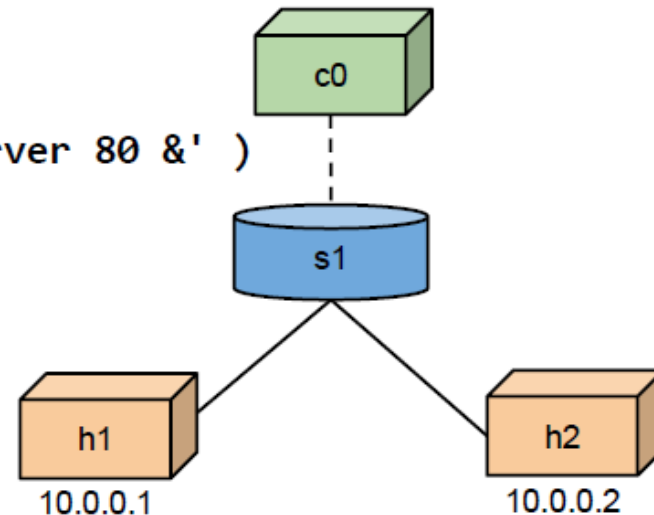
# Mininet CLI

---

```
# mn
# mn --topo tree,depth=3,fanout=3 --
link=tc,bw=10
mininet> xterm h1 h2
h1# wireshark &
h2# python -m SimpleHTTPServer 80 &
h1# firefox &
# mn --topo linear,100
# mn --custom custom.py --topo mytopo
```

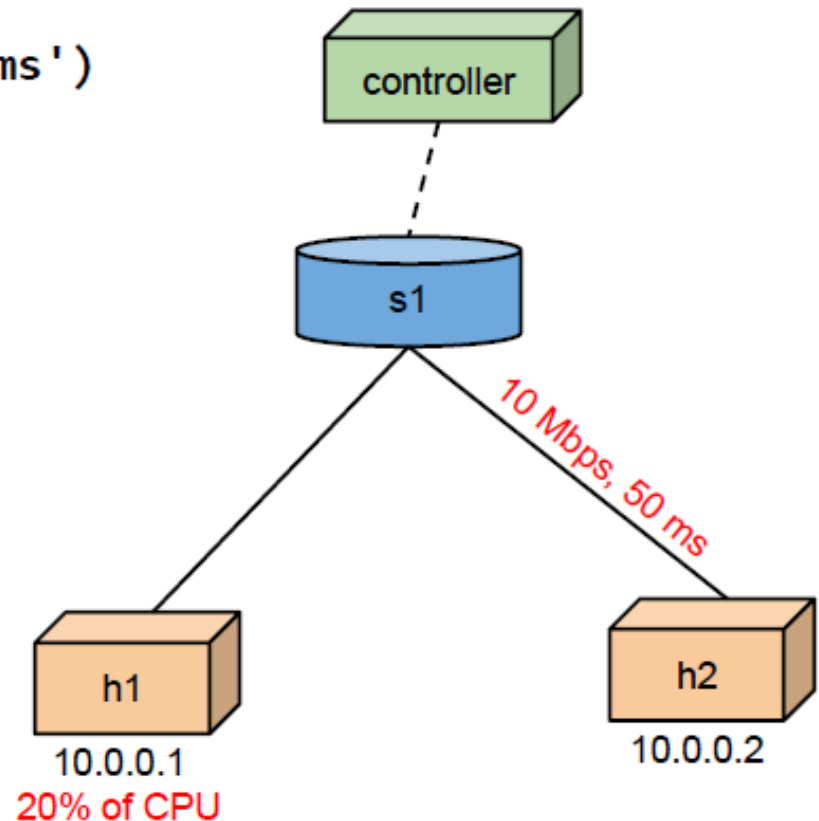
# Mininet Python API

```
net = Mininet()                # net is a Mininet() object
h1 = net.addHost( 'h1' )      # h1 is a Host() object
h2 = net.addHost( 'h2' )      # h2 is a Host()
s1 = net.addSwitch( 's1' )    # s1 is a Switch() object
c0 = net.addController( 'c0' ) # c0 is a Controller()
net.addLink( h1, s1 )         # creates a Link() object
net.addLink( h2, s1 )
net.start()
h2.cmd( 'python -m SimpleHTTPServer 80 &' )
sleep( 2 )
h1.cmd( 'curl', h2.IP() )
CLI( net )
h2.cmd('kill %python')
net.stop()
```



# Modelando desempenho

```
# Use performance-modeling link and host classes
net = Mininet(link=TCLink, host=CPULimitedHost)
# Limit link bandwidth and add delay
net.addLink(h2, s1, bw=10, delay='50ms')
# Limit CPU bandwidth
net.addHost('h1', cpu=.2)
```



# Arquivo de topologia

---

```
# cat custom.py
from mininet.topo import Topo
class SingleSwitchTopo( Topo ):
    "Single Switch Topology"
    def build( self, count=1):
        hosts = [ self.addHost( 'h%d' % i )
                  for i in range( 1, count + 1 ) ]
        s1 = self.addSwitch( 's1' )
        for h in hosts:
            self.addLink( h, s1 )
topos = { 'mytopo': SingleSwitchTopo }

# mn --custom custom.py --topo mytopo,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
```

# Cuidado!!!!

---

Os hosts no Mininet tem o mesmo file system.

- Escrever `h1:/etc/test.conf` vai sobrescrever o arquivo `h2:/etc/test.conf`
- Solução: Criar um diretório para cada host `/file/h1 /e /file/h2/` e escrever o arquivo `/file/h1/test.conf` e `/file/h2/test.conf`

Mininet exige a presença de controlador SDN

- Para criar redes legadas usar controlador que simula learning L2 switch
  - `sudo mn -controller ref`
  - `sudo mn -controller ovsc`
- Criar um nó com função de roteador IP (usando learning sw controller)
  - Veja `example/linuxrouter.py`

# Finalizando

---

Mininet is cool...  
...and funny.

Enjoy!!!!