

MININET

ASSIS TIAGO DE OLIVEIRA FILHO

ASSIS.TIAGO@GPRT.UFPE.BR

ATOF@CIN.UFPE.BR



SUMÁRIO

- Introdução
 - Motivação
- Instalação
- Mininet Tutorial
 - Uso da interface de linha de comando
 - Uso da interface de programação
- Demo
 - Demo scenarios
 - Demo presentation
- Experiências com Mininet

PLATAFORMAS PARA REDES/ENSINO DE SISTEMAS

Plataforma	Vantagens	Desvantagens
Hardware testbed	<ul style="list-style-type: none">▪ Rápido▪ Preciso: "verdade fundamental"	<ul style="list-style-type: none">▪ Caro▪ Difícil de reconfigurar▪ Difícil de mudar▪ Difícil de baixar
Simulador	<ul style="list-style-type: none">▪ Barato, flexível▪ Detalhado▪ Fácil de "baixar"▪ Tempo virtual	<ul style="list-style-type: none">▪ Pode exigir alterações no aplicativo▪ Pode não executar o código OS▪ Pode não ser "crível"▪ Pode ser lento / não interativo
Emulador	<ul style="list-style-type: none">▪ Barato, flexível▪ Código real▪ Razoavelmente preciso▪ Fácil de "baixar"▪ Uso rápido / interativo	<ul style="list-style-type: none">▪ Mais lento que o hardware▪ Possível imprecisão da multiplexação

REDE DE TESTES DE REDE BASEADA EM HARDWARE(1 / 2)

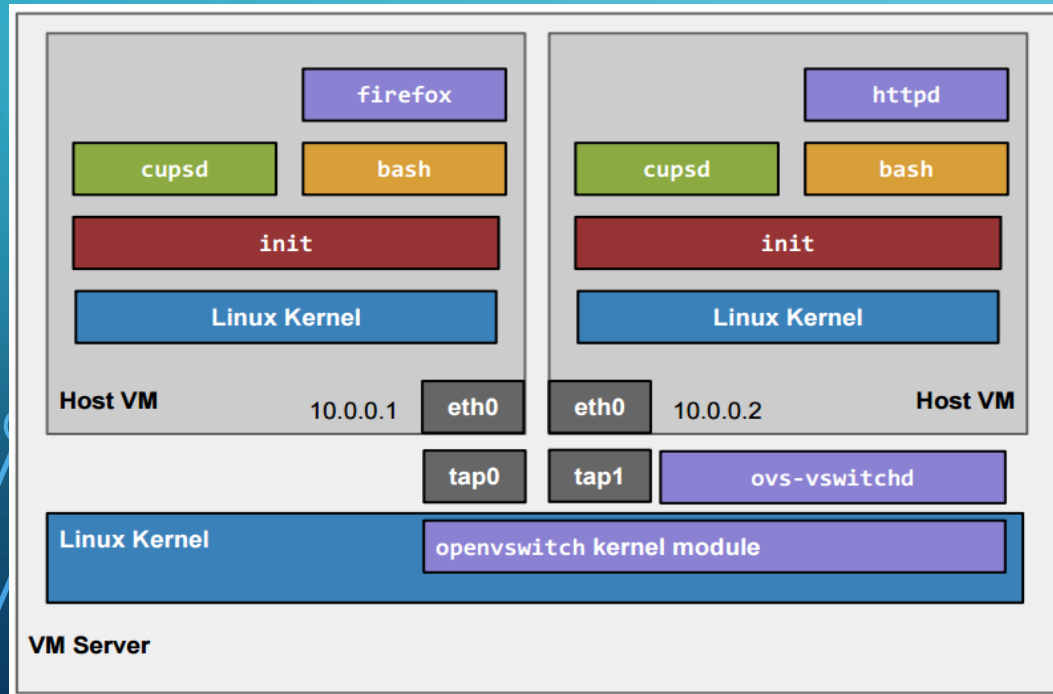
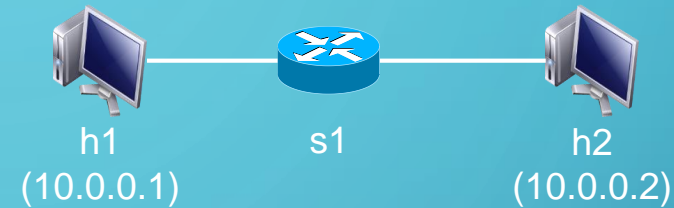
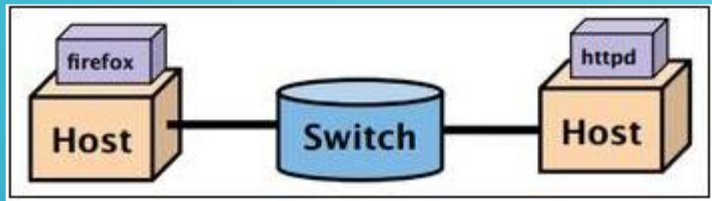


Figura 2: Virtualização completa dos dispositivos

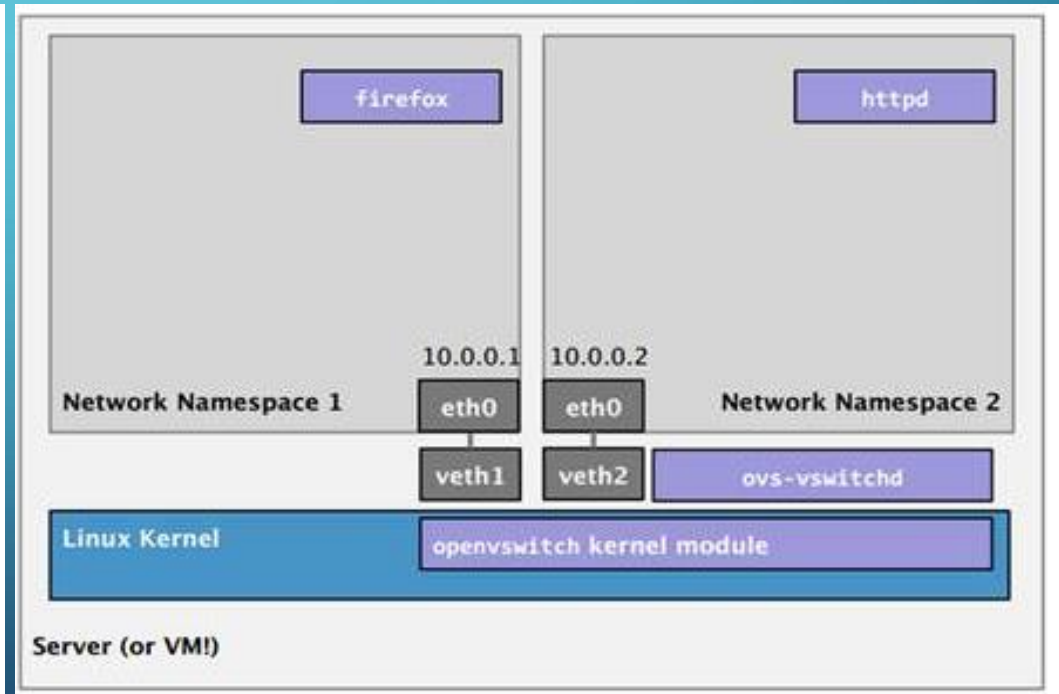


Figura 3: Virtualização "leve" dos dispositivos

REDE DE TESTES DE REDE BASEADA EM HARDWARE (2/2)

- Problemas

- Muito trabalho, mesmo para criar uma topologia de rede tão simples
- Não programável

```
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
```

REDE DE TESTES BASEADA EM EMULAÇÃO

- Uma simples ferramenta de linha de comando / API que pode facilitar o trabalho
- A solução deve nos permitir criar facilmente topologias para tamanhos variados, até centenas e milhares de nós

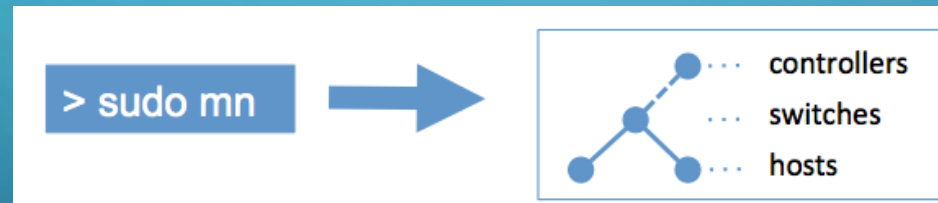


```
h1 = net.addHost( 'h1' )
h2 = net.addHost( 'h2' )
s1 = net.addSwitch( 's1' )
c0 = net.addController( 'c0' )
net.addLink( h1, s1 )
net.addLink( h2, s1 )
net.start()
CLI( net )
```

Geração de topologia usando o Mininet API

INTRODUÇÃO PARA O MININET

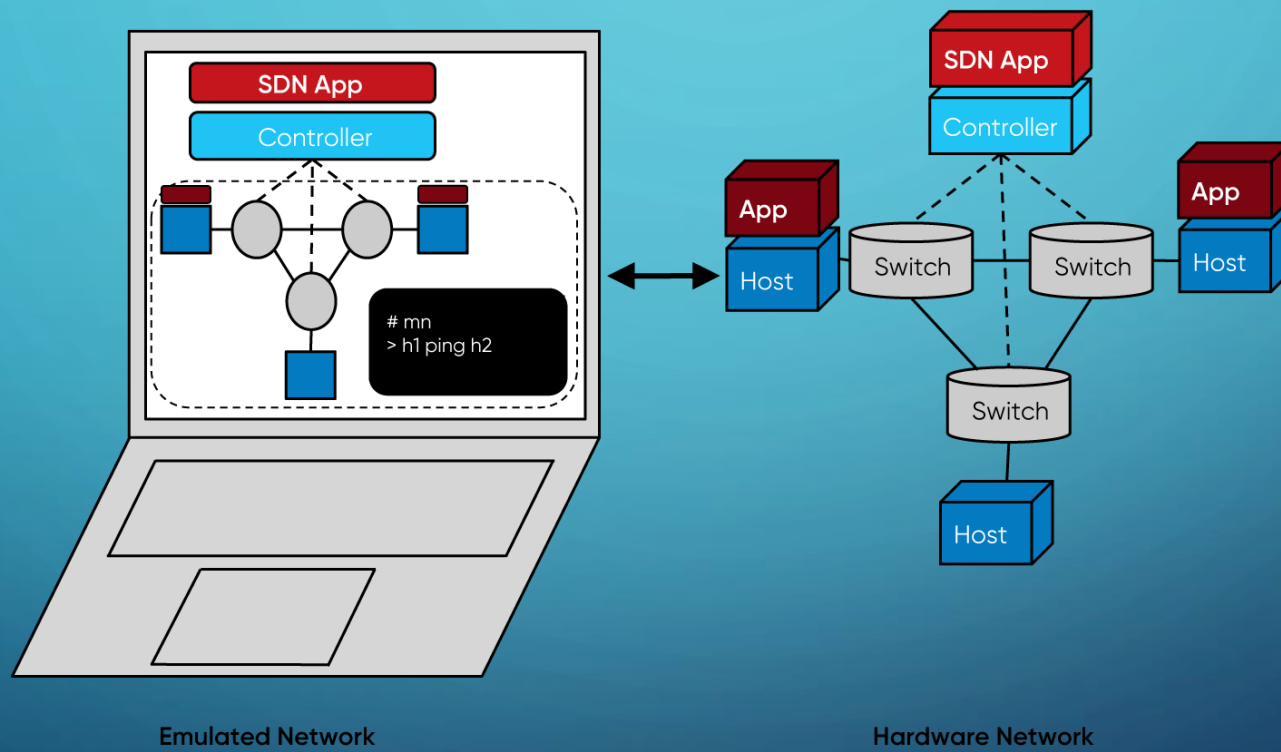
- Mininet
 - Um emulador de rede que cria uma rede virtual realista
 - Executa código real de kernel, comutador e aplicativo em uma única máquina



INTRODUÇÃO PARA O MININET

- Fornece interface de linha de comando (CLI) e interface de programação de aplicativo (API)
 - CLI: comando interativo
 - API: automação
- Abstração
 - Host: emulado como um processo no nível do sistema operacional
 - Switch: emulado usando o switch baseado em software
 - E.g., Open vSwitch, SoftSwitch

MININET



INSTALAÇÃO MININET (1 / 2)

- Mininet VM Instalação
 - A maneira mais fácil e à prova de falhas de instalar o Mininet
 - Baixe a imagem da [VM](#) pré-instalada do Mininet
 - Faça o download e instale um dos hypervisors (por exemplo, VirtualBox, Qemu, VMware Workstation, VMware Fusion ou KVM)
 - Ou imagem do [SDN HUB](#)
- Instalação nativa da fonte
 - SO Recomendado: Ubuntu 11.10 e posterior
 - Procedimentos
 - Download source do github : `$ git clone git://github.com/mininet/mininet`
 - Instalação completa: Mininet + Open vSwitch + wireshark + etc. : `$ mininet/util/install.sh -a`
 - Mínima instalação: + Mininet + Open vSwitch : `$ mininet/util/install.sh -fnv`

INSTALAÇÃO MININET (2/2)

- SO Recomendado: Ubuntu 12.04 e posterior

- Procedimentos

- Remova todos os Mininet instalados anteriormente e o Open vSwitch

```
$ sudo rm -rf /usr/local/bin/mn /usr/local/bin/mnexec \  
/usr/local/lib/python*/*/mininet* \  
/usr/local/bin/ovs-* /usr/local/sbin/ovs-*
```

- Instale o pacote Mininet de acordo com a sua versão do Ubuntu (escolha uma delas!

```
$ sudo apt-get install mininet  
$ sudo apt-get install mininet/quantal-backports  
$ sudo apt-get install mininet/precise-backports
```

- Controlador OpenvSwitch desativado se estiver em execução)

```
$ sudo service openvswitch-controller stop  
$ sudo update-rc.d openvswitch-controller disable
```

- Você também pode instalar software adicional através do source do mininet

```
$ git clone git://github.com/mininet/mininet  
$ mininet/util/install.sh -fw
```

MININET TUTORIAL (1 / 7)

- Uso da interface de linha de comando do Mininet
 - Interagir com hosts e switches
 - Inicie uma topologia mínima
 - `$ sudo mn`
 - Inicie uma topologia mínima usando um controle remoto
 - `$ sudo mn --controller=remote,ip=[IP_ADDDR],port=[listening port]`
 - Inicie uma topologia personalizada
 - `$ sudo mn --custom [topo_script_path] --topo=[topo_name]`
 - Display nós
 - `mininet> nodes`
 - Mostrar links
 - `mininet> net`
 - Descarregar informações sobre todos os nós
 - `mininet> dump`

MININET TUTORIAL (2/7)

- Interagir com hosts e switches
 - Verifique o endereço IP de um determinado nó
 - `mininet> h1 ifconfig -a`
 - Imprimir a lista de processos de um processo de host
 - `mininet> h1 ps -a`
- Teste a conectividade entre hosts
 - Verifique a conectividade fazendo ping de host0 para host1
 - `mininet> h1 ping -c 1 h2`
 - Verificar a conectividade entre todos os hosts
 - `mininet> pingall`

MININET TUTORIAL (3/7)

- Execute um teste de regressão
 - O tráfego recebe preparação
 - `mininet> iperf -s -u -p [port_num] &`
 - Geração de tráfego do cliente
 - `mininet> iperf -c [IP] -u -t [duration] -b [bandwidth] -p [port_num]`
- Variação de Link
 - `$ sudo mn -link tc,bw=[bandwidth],delay=[delay_in_millisecond]`
- Interpretador python
 - Imprimir variáveis locais acessíveis
 - `$ py locals()`
 - Executar um método através da chamada da mininet AP
 - `$ py [mininet_name_space].[method]`

MININET TUTORIAL (4/7)

- Uso da interface de programação de aplicativos Mininet
 - API de baixo nível: nós e links
 - `mininet.node.Node`
 - Um nó de rede virtual, que é simplesmente um namespace de rede
 - `mininet.link.Link`
 - Um link básico, que é representado como um par de nós

Classe	Método	Descrição
Node	MAC/setMAC	Retornar / Atribuir endereço MAC de um nó ou interface específica
	IP/setIP	Retornar / Atribuir endereço IP de um nó ou interface específica
	cmd	Envie um comando, aguarde a saída e retorne-o
	terminate	Envie o sinal kill para o nó e limpe-o depois
Link	Link	Crie um link para outro nó, faça duas novas interfaces

MININET TUTORIAL (4/7)

- `h1 = Host('h1')`
- `h2 = Host('h2')`
- `s1 = OVSSwitch('s1', inNamespace=False)`
- `c0 = Controller('c0', inNamespace=False)`
- `Link(h1, s1)`
- `Link(h2, s1)`
- `h1.setIP('10.1/8')`
- `h2.setIP('10.2/8')`

```
c0.start()  
s1.start( [ c0 ] )  
print h1.cmd( 'ping -c1',  
h2.IP() )  
s1.stop()  
c0.stop()
```

MININET TUTORIAL (5/7)

- Uso da interface de programação de aplicativos
 - Mininet API de nível médio: objeto de rede
 - `mininet.net.Mininet`
 - Emulação de rede com hosts gerados em namespaces de rede

Classe	Método	Descrição
Net	<code>addHost</code>	Adicionar um host à rede
	<code>addSwitch</code>	Adicionar um switch à rede
	<code>addLink</code>	Ligue dois nós em conjunto
	<code>addController</code>	Adicionar um controlador à rede
	<code>getNodeByName</code>	Retornar nó (s) com nome (s) próprio (s)
	<code>start</code>	Iniciar o controlador e os interruptores
	<code>stop</code>	Para o controlador, switches e hosts
	<code>ping</code>	Ping entre todos os hosts especificados e retornar todos os dados

MININET TUTORIAL (5/7)

```
net = Mininet()  
h1 = net.addHost( 'h1' )  
h2 = net.addHost( 'h2' )  
s1 = net.addSwitch( 's1' )  
c0 = net.addController( 'c0' )  
net.addLink( h1, s1 )  
net.addLink( h2, s1 )
```

```
net.start()  
print h1.cmd( 'ping -c1', h2.IP() )  
CLI( net )  
net.stop()
```

MININET TUTORIAL (6/7)

- Uso da interface de programação de aplicativos Mininet
 - API de alto nível: modelos de topologia
 - mininet.topo.Topo
 - Representação da rede do data center para multi-árvores estruturadas

Class	Method	Description
Topo	Methods similar to net	E.g., addHost, addSwitch, addLink,
	addNode	Add node to graph
	addPort	Generate port mapping for new edge
	switches	Return all switches
	Hosts/nodes/switches/links	Return all hosts
	isSwitch	Return true if node is a switch, return false otherwise

MININET TUTORIAL (6/7)

```
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 )
```

```
net = Mininet( topo=SingleSwitchTopo( 3 ) )  
net.start()  
CLI( net )  
net.stop()
```

MININET TUTORIAL (7/7)

- Uso da interface de programação de aplicativos
 - Mininet Topologia personalizada

Mais exemplos podem ser encontrados em:
<https://github.com/mininet/mininet/tree/master/examples>

```
# cat custom.py
```

```
LEN_DPID = 16
from mininet.topo import Topo
class MyTopo( Topo ):
```

```
    def name_dpid( self, index ):
        dpid = '%02d' % ( index )
        zeros = '0' * ( LEN_DPID - len( dpid ) )
        name = 's%02d' % ( index )
        return { 'name':name, 'dpid':zeros + dpid }
```

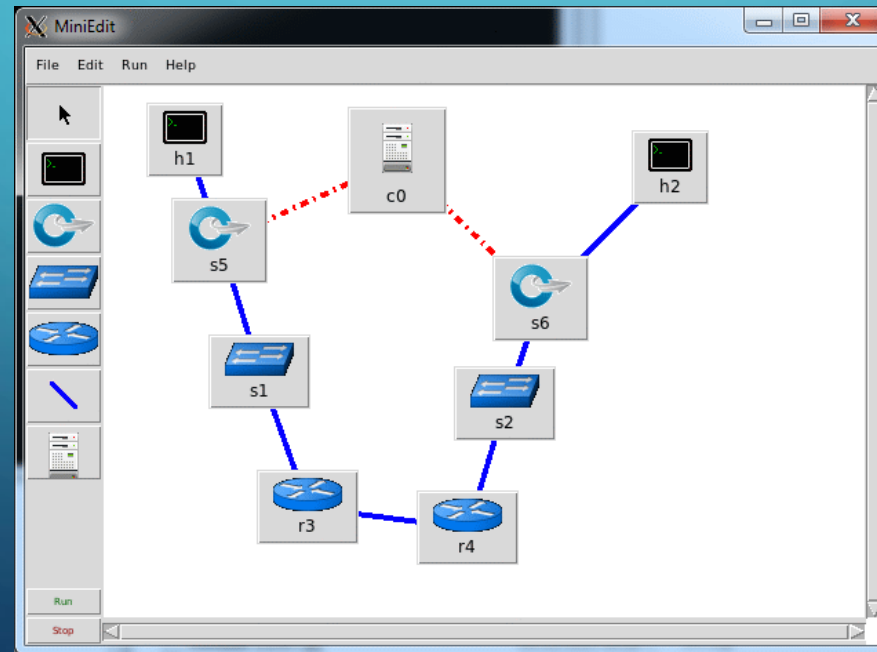
```
    def build( self, count=1):
        hosts = [ self.addHost( 'h%d' % i )
                  for i in range( 1, count + 1 ) ]
        s1 = self.addSwitch( **self.name_dpid(1) )
        for h in hosts:
            self.addLink( h, s1 )
```

```
topos = { 'mytopo': MyTopo }
```

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

APLICAÇÕES MININET

- MiniEdit
 - Uma aplicação GUI que facilita a geração da topologia Mininet
 - Salve a topologia ou exporte como um script python Mininet




```
from mininet.topo import Topo
```

```
class MyTopo( Topo ):
```

```
    def __init__( self ):
```

```
        Topo.__init__( self )
```

```
        h1 = self.addHost( 'h1' )
```

```
        h2 = self.addHost( 'h2' )
```

```
        h3 = self.addHost( 'h3' )
```

```
        s1 = self.addSwitch( 's1' )
```

```
        s2 = self.addSwitch( 's2' )
```

```
        s3 = self.addSwitch( 's3' )
```

```
        s4 = self.addSwitch( 's4' )
```

```
        s5 = self.addSwitch( 's5' )
```

```
        #Adicionando links
```

```
        self.addLink( h1, s1, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
        self.addLink( h2, s1, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
        self.addLink( h3, s5, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
        self.addLink( s1, s2, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
        self.addLink( s1, s3, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
        self.addLink( s1, s4, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
        self.addLink( s2, s5, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
        self.addLink( s3, s5, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
        self.addLink( s4, s5, bw=100,delay='5ms',loss=1,max_queue_size=10000,use_htb=True )
```

```
topos = { 'mytopo': ( lambda: MyTopo() ) }
```