

Universidade Tecnológica Federal do Paraná - UTFPR

Campus:
Campo Mourão

Professor:
Dr. Luiz Arthur Feitosa dos Santos

E-mail:
luizsantos@utfpr.edu.br

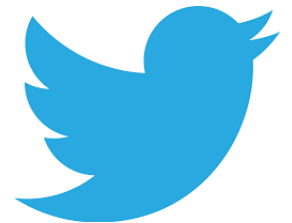
Sumário:

- **Roteadores;**
- **Datagrama IP;**
- **Tabela de roteamento;**
- **Roteamento estático (teoria, Linux e CISCO);**
- **Rota padrão;**
- **Vantagens e desvantagens do roteamento estático;**

Atenção - este material/slides são em grande parte um apanhado dos conteúdos dos livros apresentados na Bibliografia (último slide) e que estão disponíveis na biblioteca da universidade. Tais slides não dispensam o uso dos referentes livros e de outros materiais de apoio.

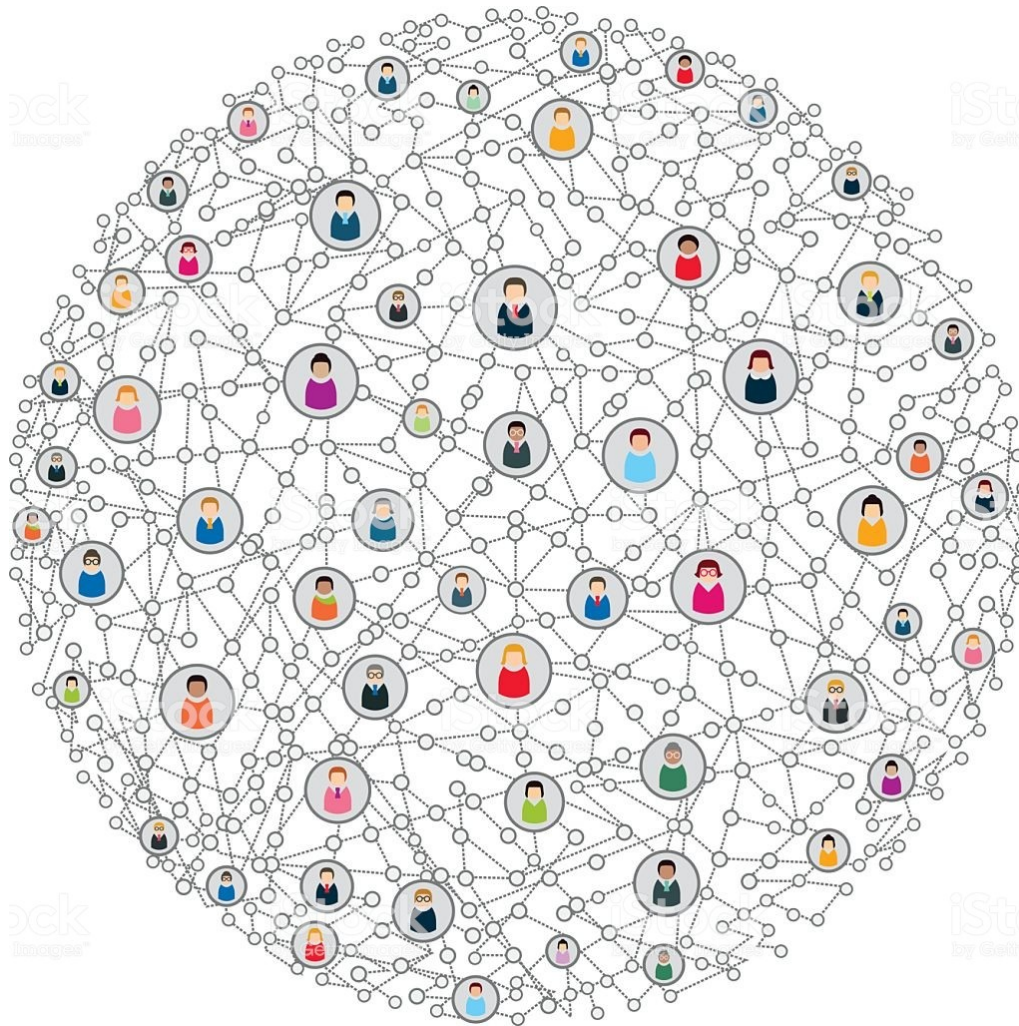
As redes de computadores, principalmente a Internet, permite que pessoas se comuniquem, colaborem e interajam de várias formas.

Por exemplo, redes são utilizadas para acessar páginas Web, conversar utilizando telefones IPs, participar de videoconferências, jogar, fazer compras, estudar etc.



Bem, há várias tecnologias e complexidades para se fazer algo como a Internet funcionar, mas uma grande questão é:

Como enviar uma informação da sua origem para o seu destino na Internet?



A resposta pode não ser tão trivial, pois como já foi dito uma rede como a Internet é complexa.

Em redes de computadores locais, as informações/quadros, podem ir da origem para o destino através de *switches* Ethernet (Camada 2).

Contudo, pensando em grandes redes, no caso a Internet, é necessário subir o nível (Camada 3) e utilizar o protocolo IP e seu esquema de roteamento (datagrama IP e roteadores) para conseguir conectar milhões ou até mesmo bilhões de *hosts*.

Então, a resposta para a pergunta anterior é:

Na Internet as informações conseguem chegar ao seu destino utilizando-se da tecnologia IP e seus roteadores.



Ok, mas o que é um roteador e como ele funciona?



Roteadores são computadores!

Atualmente a maioria dos computadores podem ser roteadores (computadores pessoais, *tablets*, *smartphones*, dispositivos embarcados, etc).

Mas na prática quando se fala de roteador, espera-se que esse seja um computador especializado nessa tarefa (fazer roteamento de pacotes de rede).

Isso normalmente significa que tal computador tem CPU, memória principal e secundária, especialmente desenvolvido ou dedicado a tratar e armazenar sistema operacional e aplicativos também especializados na tarefa de roteamento de pacotes de rede.

Então em resumo, um roteador normalmente é uma “caixinha” na qual pode-se conectar cabos de rede para interconectar múltiplas redes. Por exemplo, através do roteador podemos conectar uma rede local à Internet.

Segundo **CISCO** 2014:

Um roteador conecta uma rede em outra rede. O roteador é responsável por entregar pacotes entre redes diferentes.

Conforme **TANENBAUM** 2011:

A principal função da camada de rede é rotear pacotes da origem até o destino. O pacote pode ter que fazer múltiplos saltos (passar de um roteador para outro) até chegar ao destino.

Segundo **FOUROUZAN** 2013:

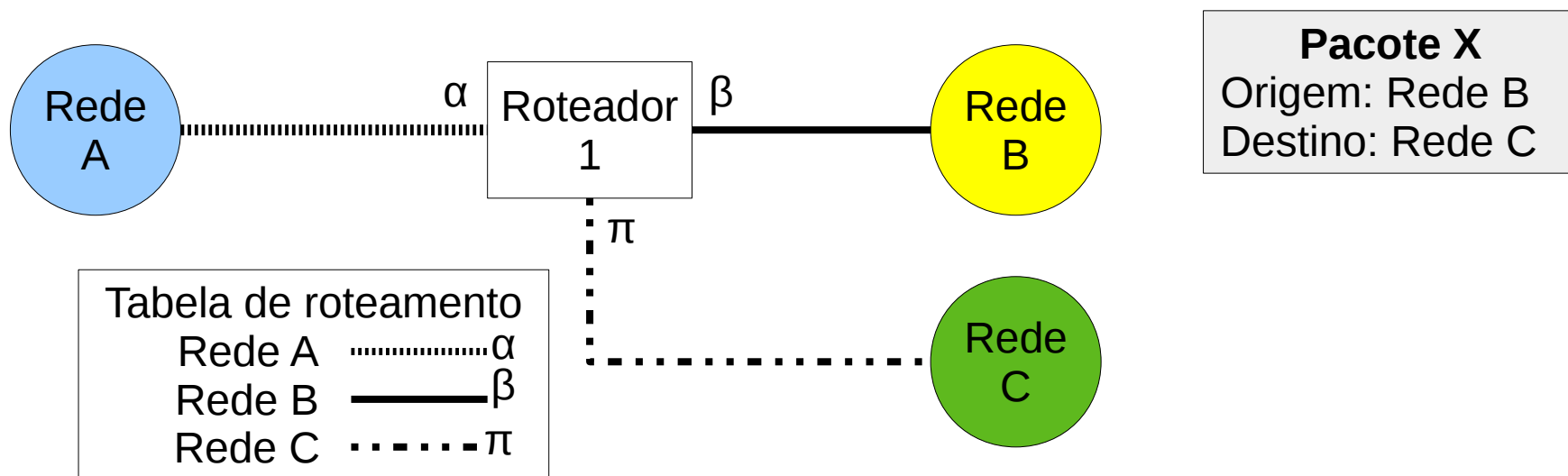
O roteamento é possível se o roteador possui uma tabela de roteamento que permite encaminhar pacotes para o próximo *host*, até alcançar o destino (roteamento *unicast*) ou destinos (roteamento *multicast*) final. Para criar tabelas de roteamento, nos roteadores, a Internet precisa de protocolos de roteamento, que devem ser responsáveis por criar e manter tais rotas.

Ok, então:

Roteador são computadores interconectados em duas ou mais redes, que permitem que pacotes cruzem tal roteador, rumo ao seu destino final.

Os roteadores sabem para onde enviar pacotes graças as tabelas de roteamento que são criadas, armazenadas e mantidas em cada roteador.

A tabela de roteamento basicamente contém informações a respeito de redes que podem ser acessadas pelos pacotes que cruzam o roteador. Cada linha dessa tabela indica uma rede que pode ser alcançada pelo roteador e como isso é feito (por exemplo, placa de rede que dá o acesso ou endereço do próximo salto/*host* que dá o acesso a rede em questão).



Portanto há três elementos básicos envolvidos quando a questão é “enviar datagramas IPs de uma rede para outra”, sendo esses: (i) Datagrama IP; (ii) tabela de roteamento e (iii) alguma forma de preencher corretamente as tabelas de roteamento com as devidas rotas. A seguir serão apresentados mais detalhes a respeito desses elementos.

• Datagrama IP

O datagrama IP contém duas informações vitais para enviar informações via rede: (i) endereço IP de origem (*source address*) e (ii) endereço IP de destino (*destination address*). O datagrama pode ser visto como uma carta de correio, com remetente e destinatário.

0	4	8	16	24	32
Ver	IHL	Service Type	Total Length		
Identifier			Flags	Fragment Offset	
Time to Live		Protocol	Header Checksum		
32 bit Source Address					
32 bit Destination Address					
Options and Padding					

• Tabela de roteamento

Basicamente todo *host* na rede possui uma tabela de roteamento, pois é assim que o *host* sabe para onde enviar uma dada informação. Por exemplo, a saída do comando `route -n` de uma máquina Linux pode ser algo similar com:

```
# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags      Metric    Ref    Use Iface
0.0.0.0          192.168.237.1   0.0.0.0          UG         600       0      0 wlan0
127.0.0.0        0.0.0.0         255.0.0.0        U          0         0      0 lo
192.168.237.0    0.0.0.0         255.255.255.0    U          600       0      0 wlan0
```

Tal tabela, funciona da seguinte forma, os pacote que chegam no *host* (criados ou roteados), tem o endereço IP de destino (*destination address* do datagrama IP) submetido a tabela de roteamento. Nesse Linux, as rotas são analisadas de baixo para cima, ou seja, da última linha/rota (que inicia com 192.168.237.0) até a primeira (inicia com 0.0.0.0). A rota que primeiro casar ganha (ou seja, o pacote é enviado por ela). Casar neste caso significa, calcular o endereço de rede baseado no endereço de destino do datagrama IP, com a máscara de rede da rota (Genmask). Caso o resultado seja o valor que está em Destination, significa que o datagrama deve ser enviado pela rota em questão.

Caso um pacote não case com uma rota, esse é submetido a próxima rota. Se nenhuma rota casar com o pacote, esse é descartado pelo *host*, que pode enviar um datagrama ICMP informando do “problema”.

Por exemplo, imaginando que se queira enviar um pacote para o *host* 192.168.237.100, tal IP é submetido a máscara da rota (Genmask), o *and* lógico entre a máscara 255.255.255.0 e o IP 192.168.237.100, dá o IP de rede 192.168.237.0, que combina com o campo Destination dessa rota. Então, o datagrama será enviado para a placa de rede wlan0, ou seja, para a primeira placa de rede WiFi do *host* Linux e deve chegar ao destino.

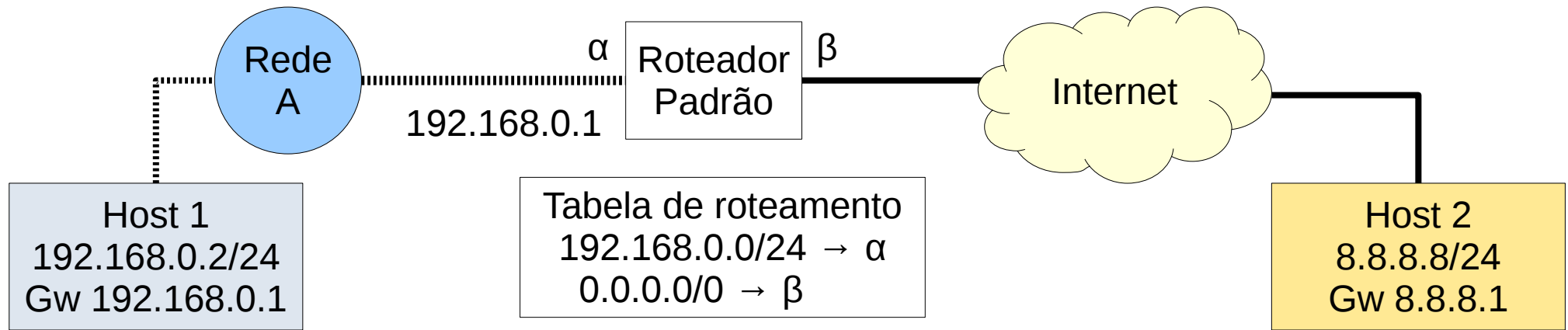
Entendido? Se sim (usando a tabela de roteamento do exemplo anterior):

- Por qual rota seguirá um pacote destinado ao *host* 172.217.29.3?
- Por qual rota seguirá um pacote destinado ao *host* 127.0.0.2?
- Todos/quaisquer pacotes casam com alguma rota nesse caso?

Ainda usando o exemplo anterior, a rota que inicia com 0.0.0.0, no campo Destination, representa a rota padrão ou *gateway* padrão.

Quando um *host* envia um pacote para outro, de uma rede diferente, o pacote é encaminhado para o *gateway* padrão, pois o *host* não pode se comunicar diretamente com dispositivos fora da sua rede local.

O roteador padrão é o destino dos pacotes destinados a outras redes (redes remotas), que não a rede local do dispositivo. O *gateway* padrão ou roteador padrão é normalmente usado para conectar a rede local à Internet.



Analisando os conceitos de roteamento apresentados até o presente momento, pode-se concluir que roteador usa sua tabela de roteamento para determinar o caminho a ser usado para a entrega de pacotes. Desta forma, a eficiência da interligação de redes depende muito da capacidade do roteador em encaminhar os pacotes da maneira eficiente.

Mas ai surge outra questão, quem é que preenche ou como são preenchidas as tabelas de roteamento?

route -n

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use Iface
0.0.0.0	192.168.237.1	0.0.0.0	UG	600	0	0 wlan0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0 lo
192.168.237.0	0.0.0.0	255.255.255.0	U	600	0	0 wlan0



Essa resposta está no último item/elementos básicos de roteamento:

- **Alguma forma de preencher as tabelas de roteamento com as devidas rotas**

Bem, há basicamente duas respostas para a pergunta anterior: (i) rotas estáticas e (ii) rotas dinâmicas. Nesses slides iremos focar no roteamento estático.

- **Rotas estáticas**

As rotas estáticas são rotas configuradas manualmente por “alguém” e há basicamente duas formas de fazer isso: (i) ao se configurar/atribuir endereços IPs às placas de rede; (ii) adicionando rotas manualmente.

Utilizando novamente um *host* com o sistema operacional Linux como exemplo e imaginando que nesse *host* não há nenhuma placa de rede configurada (não há IPs/máscaras atribuídos), esse *host* não terá nenhuma rota (ou apenas a rota de *loopback* – que normalmente existe, havendo placa de rede no computador ou não). Inclusive é muito importante que a placa de rede de *loopback* esteja ativa para o bom funcionamento da máquina.

Então, imaginando um computador com duas placas de rede (eth0 e wlan0), sem nada configurado, ao se executar o comando `ifconfig -a` (para ver as configurações das placas de rede, ativas e inativas) e `route -a` (ver as rotas deste *host*), as saídas serão:

ifconfig -a

```
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether 54:53:ed:3b:44:07 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
     inet 127.0.0.1 netmask 255.0.0.0
     inet6 ::1 prefixlen 128 scopeid 0x10<host>
     loop txqueuelen 1 (Local Loopback)
     (... restante da saída omitida)
```

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
       ether b8:76:3f:f5:cd:d3 txqueuelen 1000 (Ethernet)
       (... restante da saída omitida)
```

route -n

Kernel IP routing table

Destination	Gateway	Genmask		Flags	Metric	Ref	Use	Iface
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo	

Sem nenhum IP atribuído a nenhuma interface de rede real (eth0/wlan0), a única rota que há é para a interface virtual de *loopback* (rota para o destino 127.0.0.0).

Agora, caso seja atribuído um IP/máscara para alguma das interfaces, surgirá automaticamente uma rota para a rede em questão. Exemplo:

```
# ifconfig wlan0 192.168.237.6 netmask 255.255.255.0 up
```

```
# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
127.0.0.0	0.0.0.0	255.0.0.0	U 0	0	0	lo	
192.168.237.0	0.0.0.0	255.255.255.0	U 600	0	0	wlan0	

Nesse exemplo, é atribuído o endereço 192.168.237.6/24 para a placa de rede wlan0, o *host* Linux calcula automaticamente o endereço de rede, que é 192.168.237.0 e adiciona uma rota para essa rede. Isso significa que todo pacote destinado para a rede 192.168.237.0/24, deve ser enviado diretamente à placa de rede wlan0.

Na prática isso normalmente representa que o *host* foi configurado para acessar uma rede local. Caso se queira conectar à Internet (outras redes), será necessário configurar a rota padrão (que também é estática, neste caso). Exemplo:

```
# route add default gw 192.168.237.1
```

```
# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.237.1	0.0.0.0	UG	600	0	0	wlan0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
192.168.237.0	0.0.0.0	255.255.255.0	U	600	0	0	wlan0

O comando `route add default gw...`, irá adicionar uma rota padrão estática (0.0.0.0 192.168.237.1 0.0.0.0 ... wlan0), que no exemplo, diz basicamente que: qualquer pacote que não combinar com a rota 192.168.237.0/24 ou com a rota 127.0.0.0/8 (já que essas rotas são analisadas primeiro que a rota padrão – ver slides anteriores) deve ser enviado para o roteador 192.168.237.1, através da placa de rede wlan0. A placa de rede wlan0 foi atribuída automaticamente, pois é a placa de rede que dá acesso à rede 192.168.237.0/24, que é a rede onde está o roteador padrão.

Observação: o endereço IP do roteador/*gateway* padrão deve ser informado/ conhecido pelo administrador da rede ou obtido automaticamente via DHCP. Esse é o roteador de borda, que conecta a rede local às redes externas. Atualmente, um bom exemplo de roteador padrão é o modem ADSL de uma casa.

Essa configuração de rotas, vistas até aqui, é a configuração básica de um *host* normal (não roteador). Em um roteador, espera-se que seja utilizando algum comando para adicionar rotas, tais rotas normalmente estão fora da faixa de rede atribuída as placas de rede do *host*. Voltando ao exemplo anterior, a placa de rede está atrelada a rede 192.168.237.0/24, mas um comando poderia adicionar rotas para as redes 10.1.0.0/26 e 10.2.0.0/16:

```
# route add -net 10.1.0.0/16 dev eth0
# route add -net 10.2.0.0/16 dev wlan0
```

```
# route -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.237.1	0.0.0.0	UG	600	0	0	wlan0
10.1.0.0	0.0.0.0	255.255.0.0	U	0	0	0	eth0
10.2.0.0	0.0.0.0	255.255.0.0	U	0	0	0	wlan0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
192.168.237.0	0.0.0.0	255.255.255.0	U	600	0	0	wlan0

Como pode-se observar na saída do comando `route -n`, o primeiro comando `route` adicionou uma rota para a rede 10.1.0.0, através da placa de rede `eth0`. Já o segundo comando `route`, adicionou uma rota para a rede 10.2.0.0, via `wlan0`. Ou seja, agora essa máquina ou *hosts* que utilizem essa máquina como roteador padrão, podem acessar as redes 10.1.0.0 e 10.2.0.0 através dessas rotas.

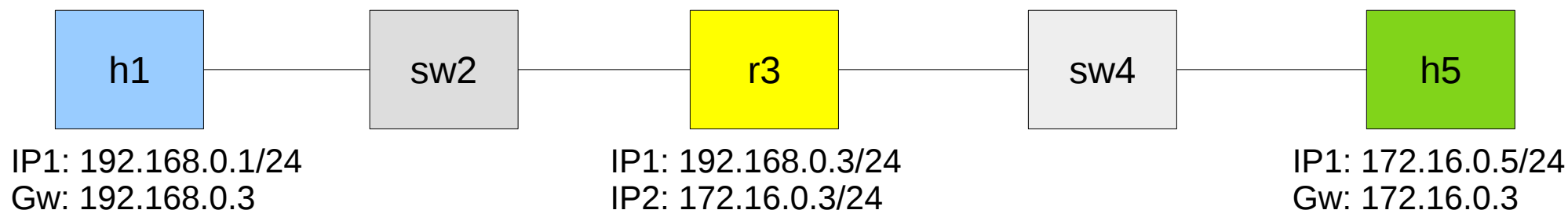
Observação importante: para configurar o *host* Linux como roteador é necessário utilizar o comando a seguir:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Para desativar o host como roteador, basta trocar o 1 por 0, no comando anterior.

Atividade 1

Dados os exemplo anteriores, configure o cenário de rede a seguir de forma que o *host* h1 consiga se comunicar com o *host* h5. Tal cenário deve ser implementado/simulado no Mininet (<http://mininet.org/>). O slide seguinte apresenta o arquivo de criação da rede no Mininet.



O objetivo dessa atividade é: (i) aprender sobre roteamento estático, (ii) como configurar máquinas Linux e (iii) como utilizar simuladores.

Configuração de rede no Mininet para o cenário da Atividade 1

```
#!/usr/bin/python

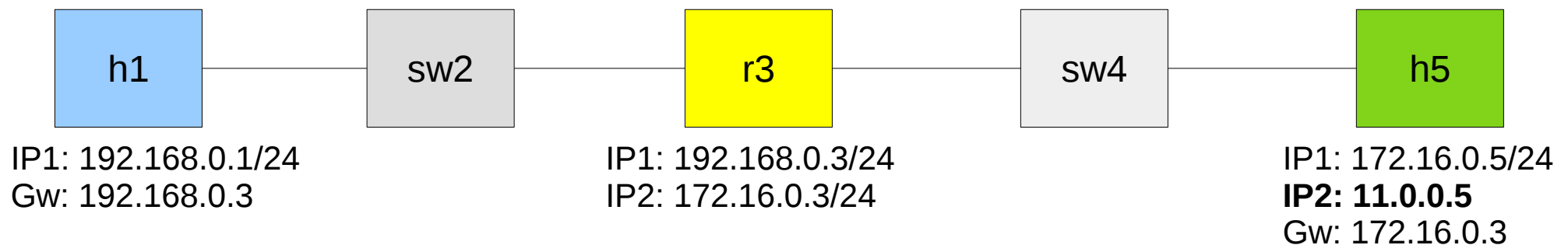
from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info

def emptyNet():
    net = Mininet(controller=Controller )
    net.addController( 'c0' )
    h1 = net.addHost( 'h1' )
    sw2 = net.addSwitch( 'sw2' )
    r3 = net.addHost( 'r3' )
    sw4 = net.addSwitch( 'sw4' )
    h5 = net.addHost( 'h5' )
    net.addLink( h1, sw2 )
    net.addLink( r3, sw2 )
    net.addLink( r3, sw4 )
    net.addLink( h5, sw4 )
    net.start()
    CLI( net )
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()
```

Atividade 1.2

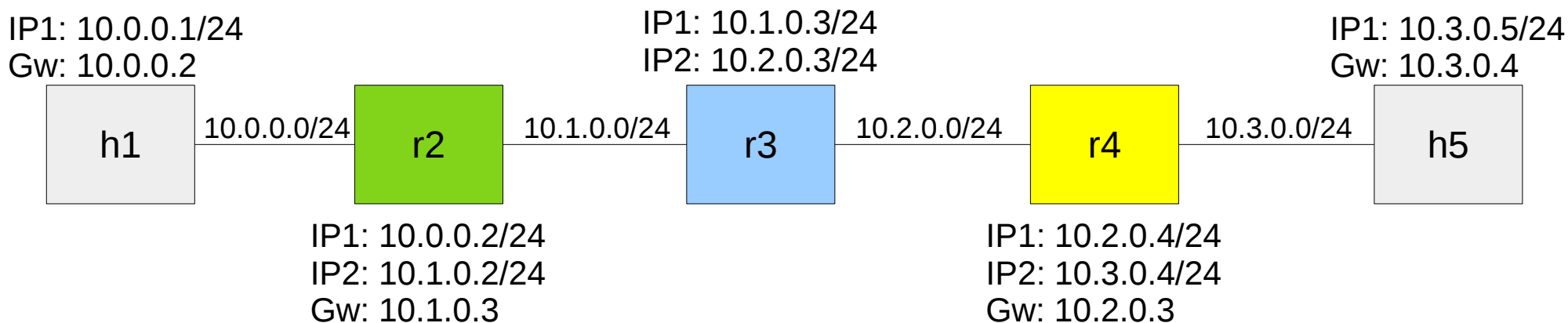
Altere o *host* h5 para que tenha dois IPs na mesma placa de rede (ifconfig h5-eth0:1 11.0.0.5), e configure o roteador, para que esse consiga rotear pacotes para esse novo IP.



Observação: não é necessário alterar o arquivo do Mininet. É possível fazer tal atividade adicionando rota para rede ou rota para *host* h5 com IP 11.0.0.5.

Atividade 2

Utilizando roteamento estático no Linux no Mininet, configure o seguinte cenário de rede:



Nessa atividade o r3 é um roteador central, no qual pode ser necessário configurar suas rotas da seguinte forma:

```
# route add -net <rede> gw <ip_rotador_que_da_acesso_a_rede>
```

No comando anterior <rede> é a rede para os quais os pacotes serão roteados e <ip_rotador_que_da_acesso_a_rede> é próximo *host* que dá acesso a essa rede. Pesquise a respeito do uso desse comando/parâmetros.

Configure todo esse cenário de rede, de forma que o h1 “pingue” h5.

Configuração de rede no Mininet para o cenário da Atividade 2

```
#!/usr/bin/python

from mininet.net import Mininet
from mininet.node import Controller
from mininet.cli import CLI
from mininet.log import setLogLevel, info

def emptyNet():
    net = Mininet(controller=Controller )
    net.addController( 'c0' )
    h1 = net.addHost( 'h1' )
    r2 = net.addHost( 'r2' )
    r3 = net.addHost( 'r3' )
    r4 = net.addHost( 'r4' )
    h5 = net.addHost( 'h5' )
    net.addLink( h1, r2 )
    net.addLink( r2, r3 )
    net.addLink( r3, r4 )
    net.addLink( r4, h5 )
    net.start()
    CLI( net )
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    emptyNet()
```

Rotas estáticas em roteadores CISCO

Bem, até agora utilizamos máquinas com Linux para fazer experimentos. O Linux é bom para tarefas de rede, tal como servidores. Contudo no mundo corporativo, na maioria das vezes, são empregadas soluções proprietárias, principalmente devido ao suporte.

Assim, vamos estudar também roteamento utilizando roteadores da CISCO, que é reconhecida como uma das melhores empresas para soluções de redes de computadores.

Infelizmente, é claro que, os equipamentos da CISCO são caros, sendo então recomendável/necessário utilizar simuladores para estudar/praticar a configuração de rede nesses equipamentos. Desta forma, vamos utilizar o simulador Packet Tracer da CISCO:

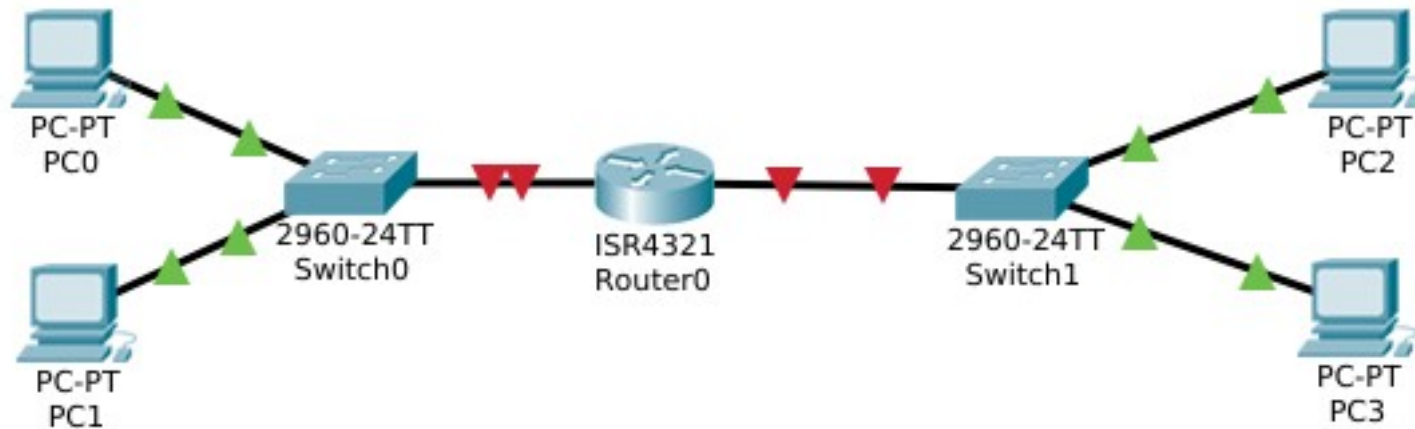
<https://www.netacad.com/pt-br/courses/packet-tracer>

Tal simulador permitirá construir cenários de rede utilizando uma grande gama de equipamentos, tais como: switches, roteadores, servidores, hosts, etc.

O objetivo aqui não será virar um *expert* em CISCO e sim apenas entender os conceitos necessário para roteamento de pacotes.

Exemplo 1

No Cisco Packet Tracer, vamos criar uma rede bem simples:

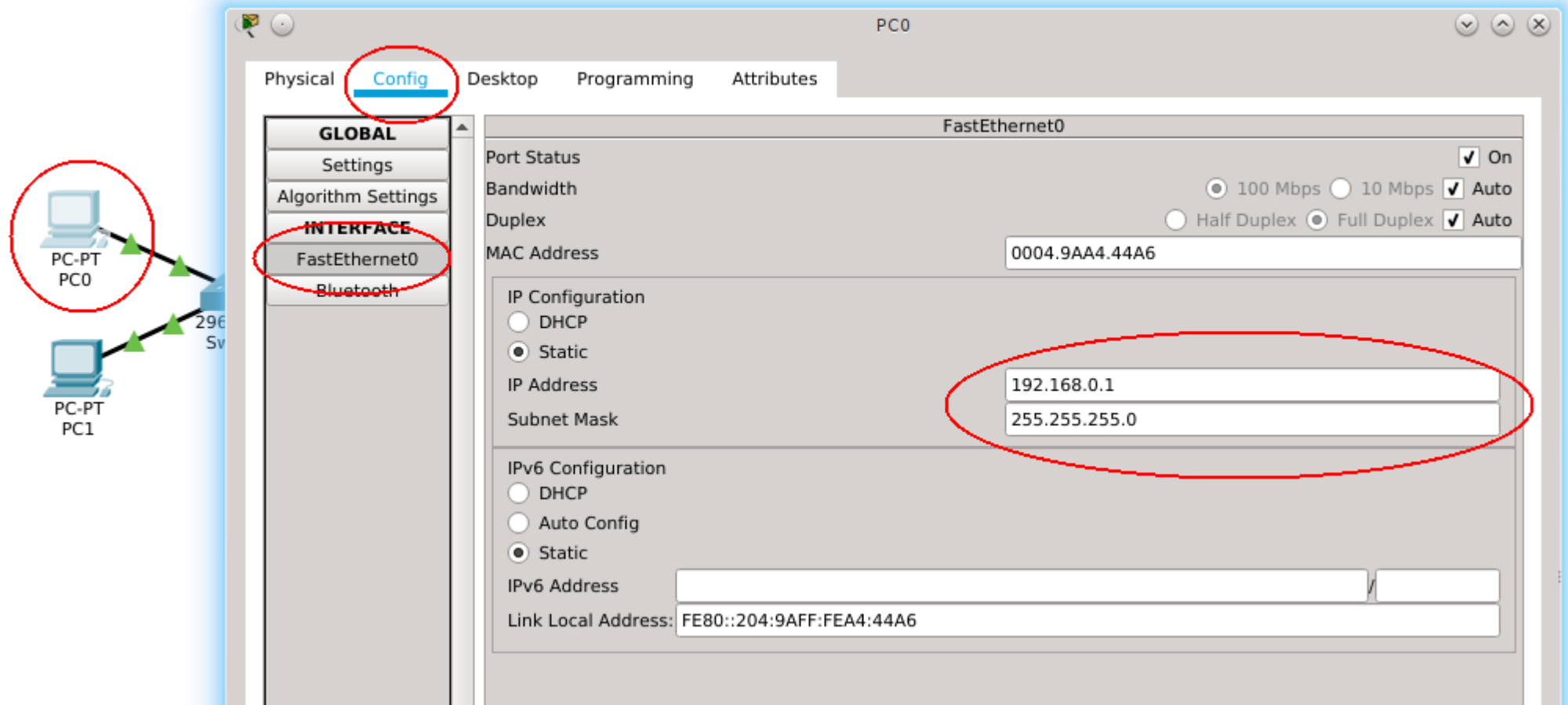


É para ser bem intuitivo, procure no canto inferior esquerdo os equipamentos de rede e adicione:

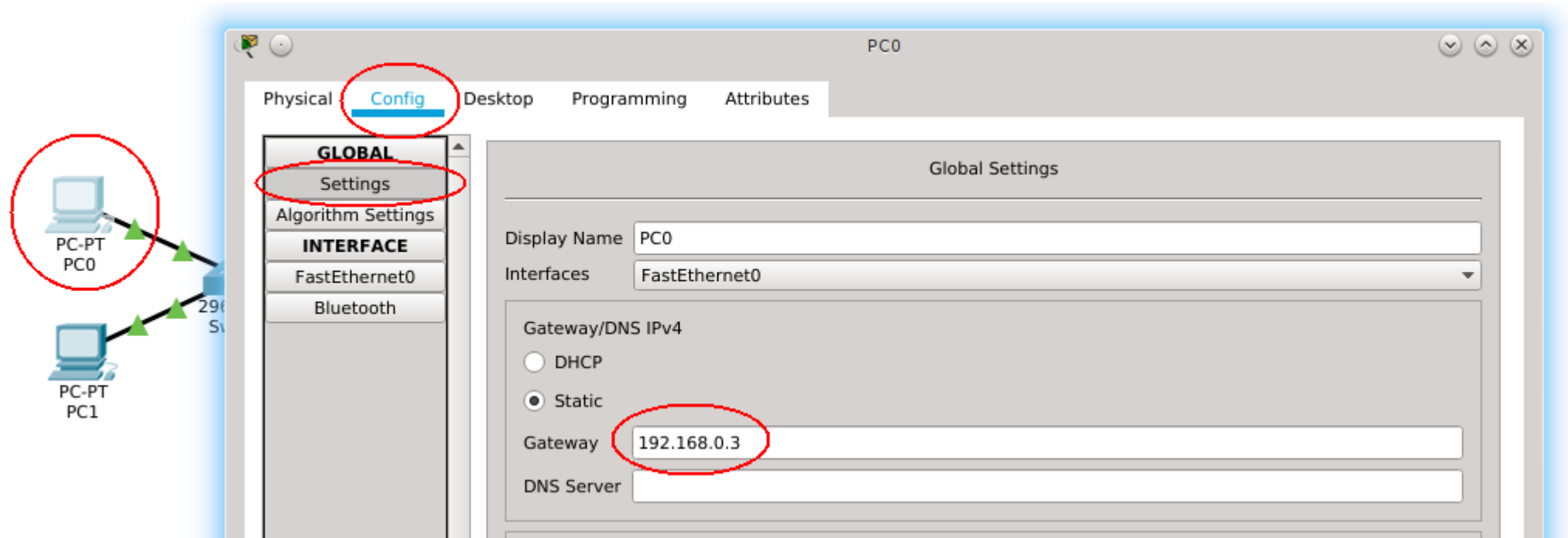
- 4 PCs;
- 2 *switches* 2960;
- 1 *router* 4321.

Depois interligue os computadores, tal como na figura, usando o cabo *copper straight-through* (cabo trançado), utilizando as portas *fast-ethernet*.

Agora vamos configurar os PCs, clique em cima do PC0 e então abrirá uma caixa para configuração. Então, clique em Config e FastEthernet0, para configura o IP Address e Subnet Mask, com os valores 192.168.0.1 e 255.255.255.0, respectivamente.



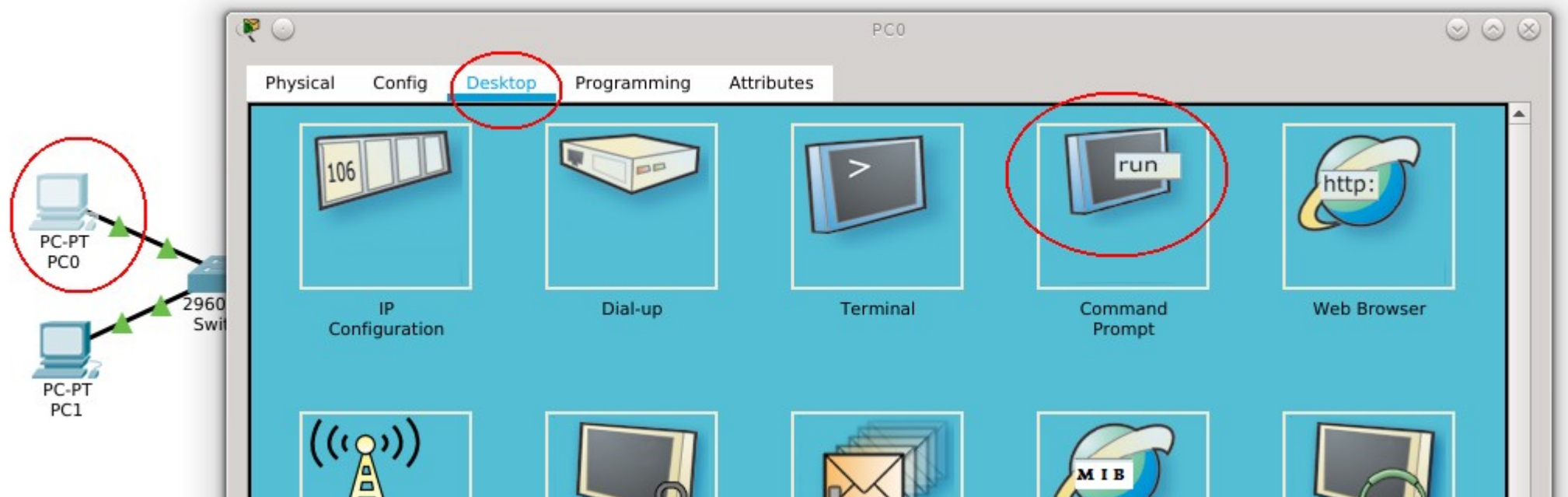
Ainda nessa tela de configuração do PC0, agora clique em Settings e configure o Gateway com o valor 192.168.0.3.



Faça esse procedimento para todos os PCs, de forma que esses fiquem com as seguintes configurações.

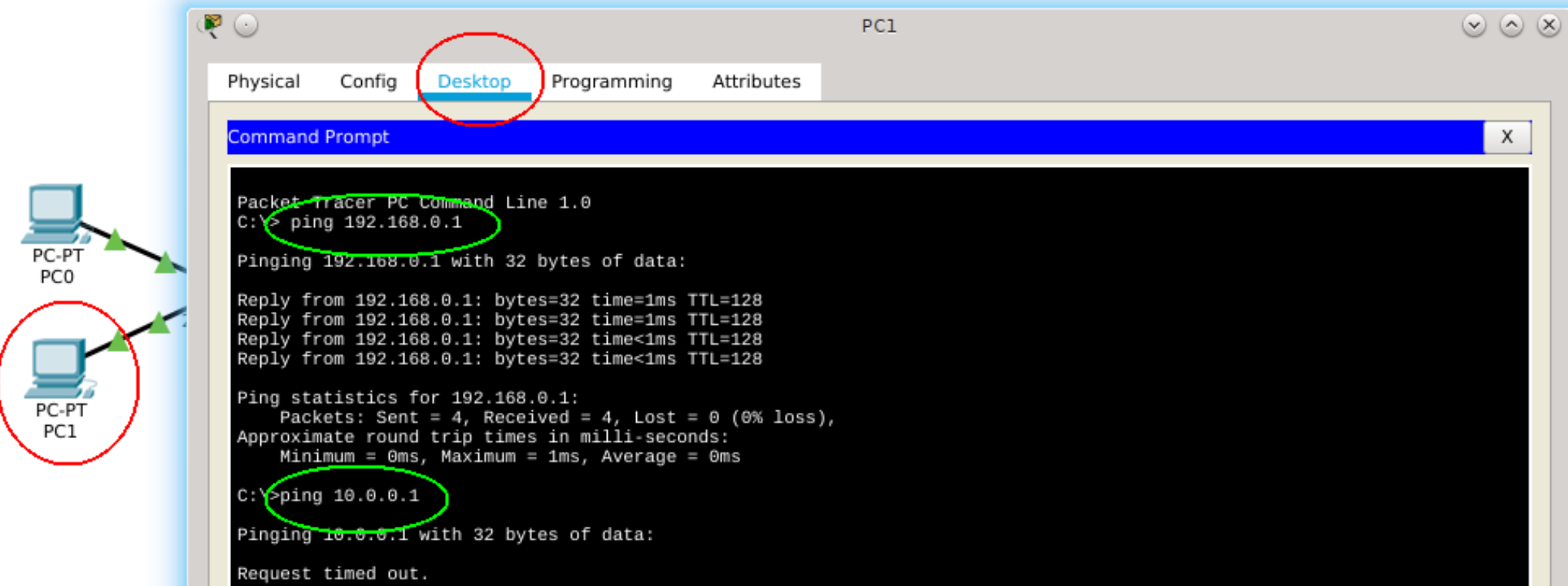
- PC0: IP 192.168.0.1/24 e GW (Gateway) 192.168.0.3;
- PC1: IP 192.168.0.2/24 e GW 192.168.0.3;
- PC2: IP 10.0.0.1/8 e GW 10.0.0.3;
- PC3: IP 10.0.0.2/8 GW 10.0.0.3.

Depois de configurar todos os PCs teste sua conectividade com o comando ping. Para isso, por exemplo, clique em cima do PC0, na caixa de diálogo clique em Desktop e Command Prompt, então será possível digitar comandos no PC em questão.



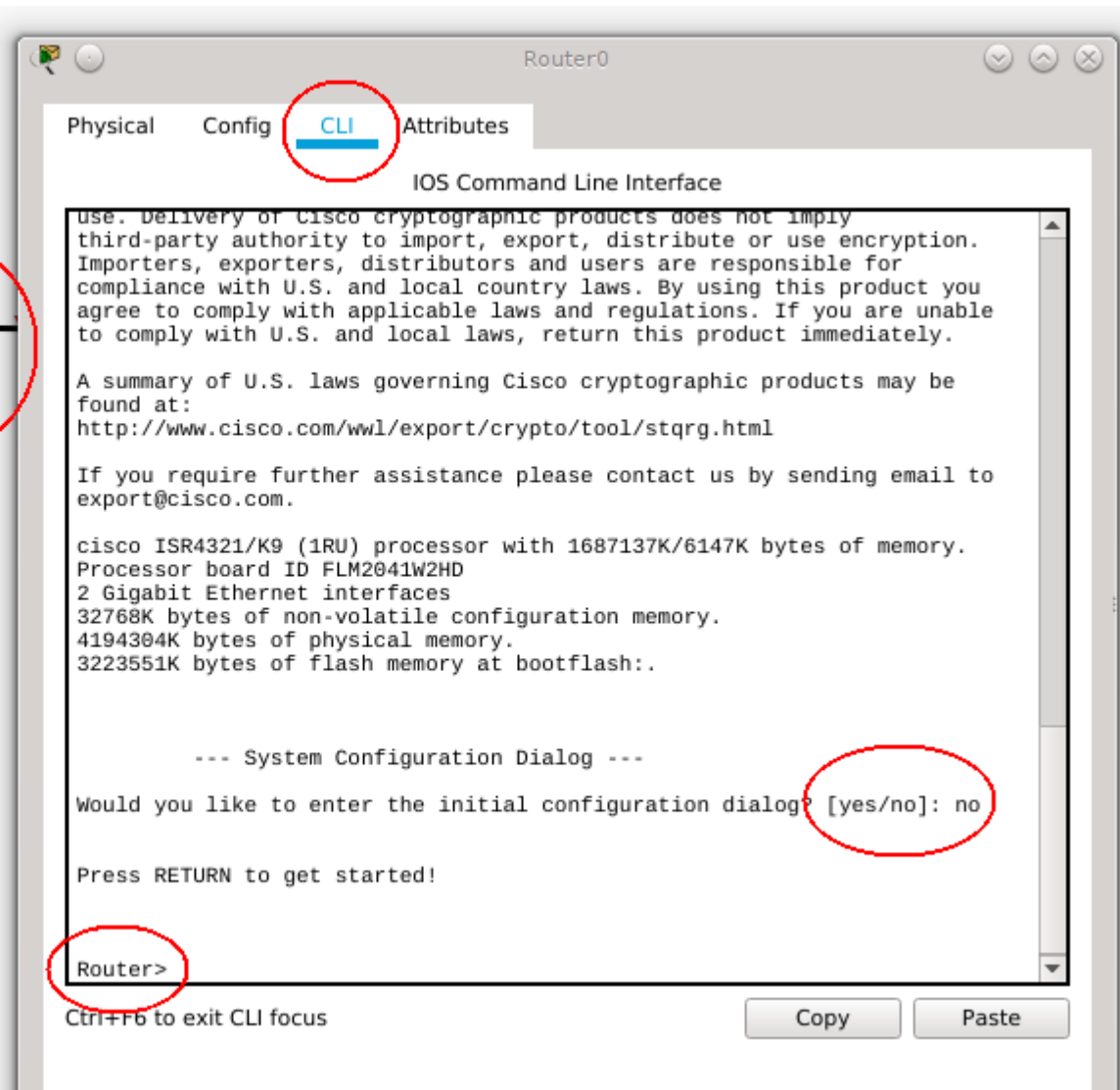
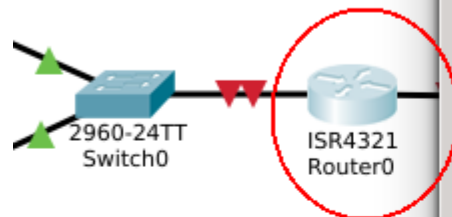
No *prompt* de comando do PC1 teste a conectividade entre os PCs, utilizando o comando ping:

- ping 192.168.0.1
- ping 10.0.0.1



Com a configuração atual será possível “pingar” apenas o PC0, que está na mesma rede, já o PC2 não responderá, pois está em uma rede diferente e precisa do roteador configurado corretamente para isso.

Então, para haver comunicação entre todas as máquinas do cenário é necessário configurar o roteador (router0). Para isso, clique no Router0, depois na aba CLI. Nessa configuração não iremos ao dialogo de configuração inicial, então digite no. Assim, teremos acesso ao terminal do roteador (Router>).



No terminal do roteador, vamos iniciar configurando os IPs das placas de rede (os comandos a serem digitados estão em negrito):

```
Router>enable  
Router#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#interface gigabitethernet0/0/0  
Router(config-if)#ip address 192.168.0.3 255.255.255.0  
Router(config-if)#no shutdown
```

```
Router(config-if)#  
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed  
state to up  
interface gigabitethernet0/0/1  
Router(config-if)#ip address 10.0.0.3 255.0.0.0  
Router(config-if)#no shutdown
```

```
Router(config-if)#  
%LINK-5-CHANGED: Interface GigabitEthernet0/0/1, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/1, changed  
state to up  
exit  
Router(config)#  
Router#
```


Os comandos anteriores têm as seguintes funções:

- `enable` – inicia sessão no modo privilegiado;
- `configure terminal` – Coloca no modo de configuração global;
- `interface` – leva ao modo de configuração da interface. No caso do comando `interface gigabitethernet0/0/0`, é para configurar especificamente tal placa de rede;
- `ip address` – configura o endereço IP e máscara de uma interface;
- `no shutdown` – configura a interface para que ela não seja desligada, ou seja, fique ativa;
- `exit` – sai do modo de configuração.

Note que alguns comandos não ficaram na frente do *prompt* de comando, devido a saída do terminal. Isso aconteceu toda vez que foi executado o `no shutdown`, e por consequência os comandos seguintes (`interface gigabitethernet0/0/1` e `exit`), ficaram fora do prompt de comando, mas não há problemas, é só digitar o comando e dar enter.

Depois disso é possível utilizar o comando `ping` no próprio terminal do roteador ou dos hosts para testar a conectividade entre ambas redes.

Podemos utilizar o comando `show` para visualizar algumas configurações/status do roteador. Por exemplo para ver os IPs atribuídos nas interfaces de rede do roteador:

```
Router#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet0/0/0	192.168.0.3	YES	manual	up	down
GigabitEthernet0/0/1	10.0.0.3	YES	manual	up	down
Vlan1	unassigned	YES	unset	administratively down	down

No cenário do exemplo podemos ver que os IPs foram atribuídos as interfaces.

Outro comando útil é o para ver rotas no roteador:

```
Router#show ip route
```

(...saída omitida)

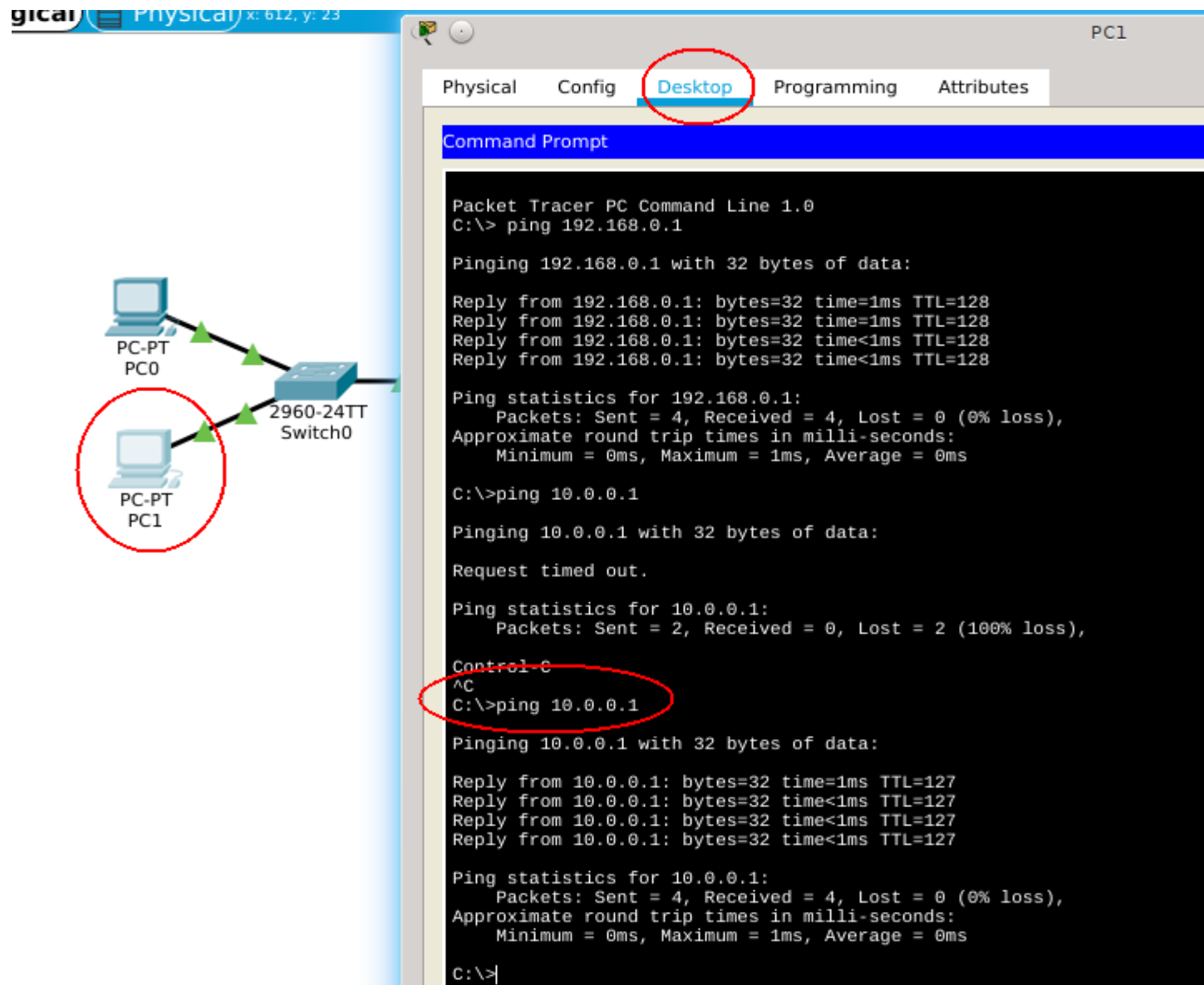
```

    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/8 is directly connected, GigabitEthernet0/0/1
L       10.0.0.3/32 is directly connected, GigabitEthernet0/0/1
    192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, GigabitEthernet0/0/0
L       192.168.0.3/32 is directly connected, GigabitEthernet0/0/0

```

Com a saída é possível notar que há rotas para as redes **10.0.0.0/8** via interface **GigabitEthernet0/0/1** e **192.168.0.0/24** via **GigabitEthernet0/0/0**. O **C** indica que é uma rede conectada a placa de rede e o **L** informa que o IP local dessa placa de rede.

Com tudo configurado, vamos testar a conectividade do cenário via ping a partir do PC1. Note que antes de configurar o roteador não era possível “pingar” o PC2, mas depois da configuração feita no roteador, o PC1 consegue se comunicar com o PC2.



The screenshot shows a Packet Tracer workspace with a network diagram on the left and a PC1 configuration window on the right. In the network diagram, PC-PT PC0 and PC-PT PC1 are connected to a 2960-24TT Switch0. PC1 is circled in red. The PC1 configuration window has the 'Desktop' tab selected, showing a Command Prompt. The Command Prompt displays the results of a ping test to 192.168.0.1, which is successful, and a ping test to 10.0.0.1, which fails with a 100% loss. The 'Control-C' command is also shown, and the ping test to 10.0.0.1 is repeated, showing a successful result.

Physical Config **Desktop** Programming Attributes

Command Prompt

```
Packet Tracer PC Command Line 1.0
C:\> ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:

Reply from 192.168.0.1: bytes=32 time=1ms TTL=128
Reply from 192.168.0.1: bytes=32 time=1ms TTL=128
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),

Control-C
^C
C:\>ping 10.0.0.1

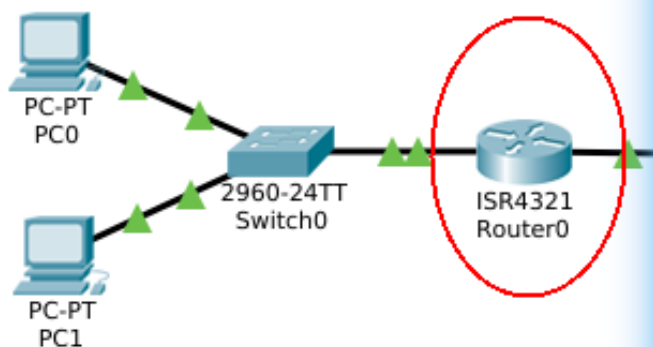
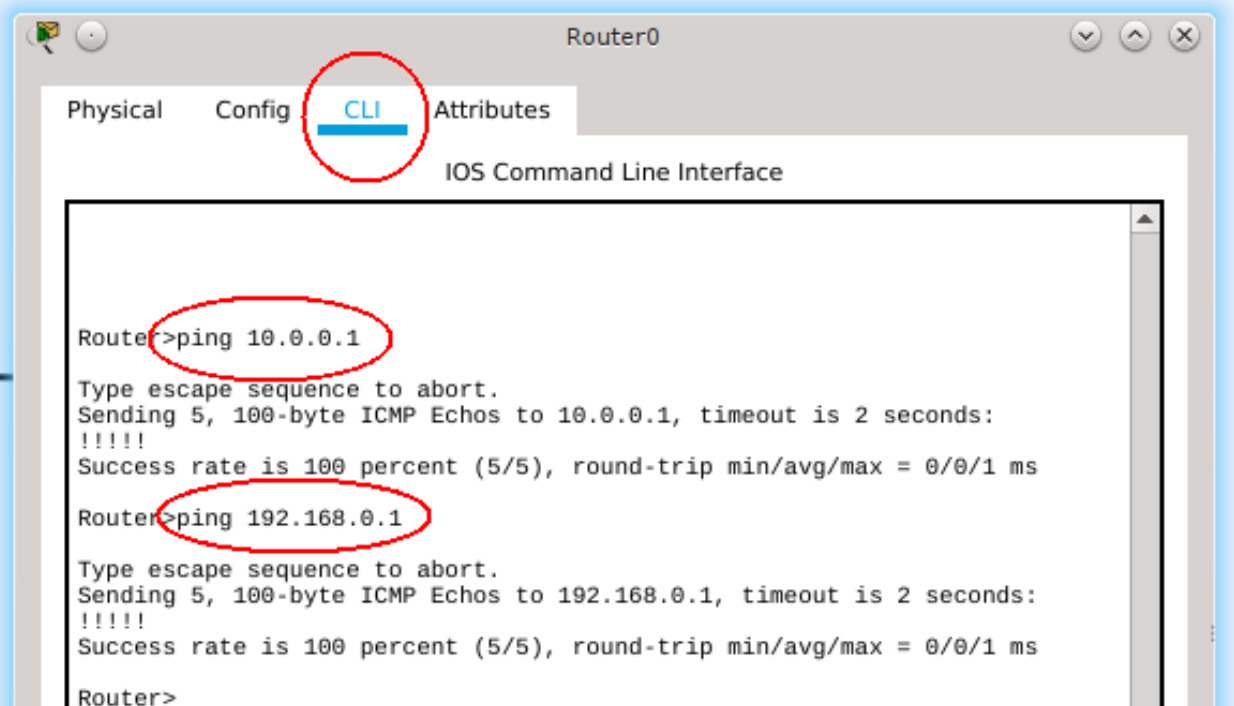
Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=1ms TTL=127
Reply from 10.0.0.1: bytes=32 time<1ms TTL=127
Reply from 10.0.0.1: bytes=32 time<1ms TTL=127
Reply from 10.0.0.1: bytes=32 time<1ms TTL=127

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>|
```

O mesmo teste pode ser feito a partir do roteador!

```

Router0
Physical Config CLI Attributes
IOS Command Line Interface

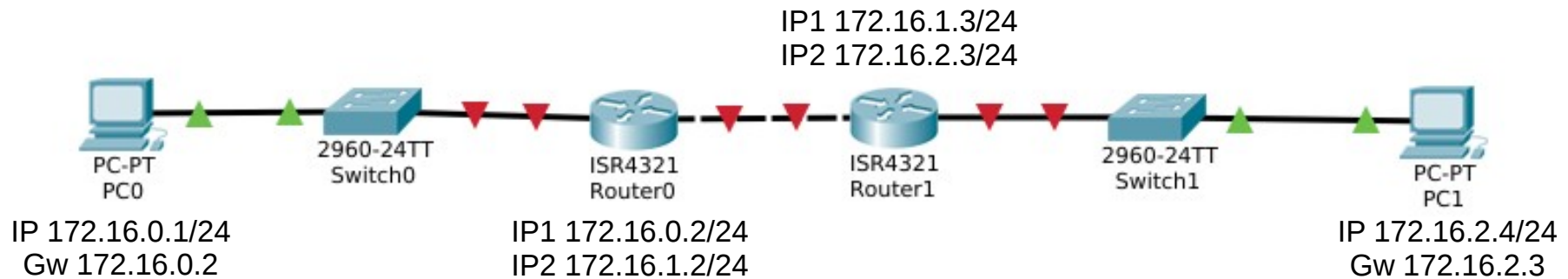
Router>ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms

Router>ping 192.168.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms

Router>
  
```

Exemplo 2

Vamos agora configurar o seguinte cenário de rede:

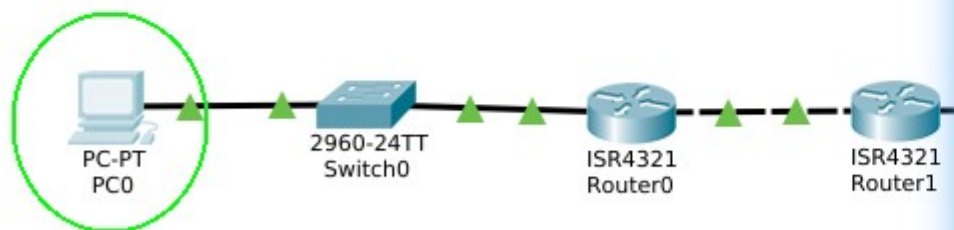


Bem, crie tal cenário no simulador da CISCO e configure PC0 e PC1, tal como no exemplo anterior. Configure também as interfaces de rede do Router0 e Router1, baseando-se também no exemplo anterior.

Após essa configuração faça o teste de conectividade (ver imagem do próximo slide), contudo não são todas as redes que estão completamente acessíveis. Pois, faltam rotas nos roteadores. Ou seja, o router0 sabe da existência da rede 172.16.0.0/24 e da rede 172.16.1.0/24, mas não conhece a rede 172.16.2.0/24, por exemplo.

Visualize as rotas presentes nos roteadores com o comando `show ip route`.

Neste teste no PC0 não é possível acessar a rede 172.16.2.0/24, pois o Router0 não conhece a rota para essa rede!



PC0

Physical Config **Desktop** Programming Attributes

Command Prompt

```

Packet Tracer PC Command Line 1.0
C:\>ping 172.16.0.2

Pinging 172.16.0.2 with 32 bytes of data:

Reply from 172.16.0.2: bytes=32 time=1ms TTL=255

Ping statistics for 172.16.0.2:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

Control-C
^C
C:\>ping 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.1.2: bytes=32 time<1ms TTL=255
Reply from 172.16.1.2: bytes=32 time<1ms TTL=255

Ping statistics for 172.16.1.2:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

Control-C
^C
C:\>ping 172.16.2.3

Pinging 172.16.2.3 with 32 bytes of data:

Reply from 172.16.0.2: Destination host unreachable.

Ping statistics for 172.16.2.3:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),

Control-C
^C
C:\>
  
```

Bem, para que no teste anterior o PC0 possa acessar a rede 172.16.3.0/24, podemos criar rotas estáticas em Router0 e Router1 (já que Router1 também não sabe da existência da rede 172.16.0.0/24). Para isso executamos os seguintes comandos em Router0:

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 172.16.2.0 255.255.255.0 g0/0/1
%Default route without gateway, if not a point-to-point interface, may impact
performance
Router(config)#ip route 172.16.2.0 255.255.255.0 172.16.1.3
Router(config)#
```

Já no Router1 execute:

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 172.16.0.0 255.255.255.0 g0/0/0
%Default route without gateway, if not a point-to-point interface, may impact
performance
Router(config)#ip route 172.16.0.0 255.255.255.0 172.16.1.2
Router(config)#
```

Na configuração anterior o comando `ip route`, adiciona rotas estáticas, sendo ainda que o primeiro comando adiciona rota indicando apenas a placa de rede e o segundo indica o endereço do próximo salto (*host* para o qual o pacote deve ser roteado).

Agora executando o comando `show ip route` no Router0, é possível verificar uma rota estática para a rede 172.16.2.0/24:

```
Router#show ip route
```

```
Gateway of last resort is not set
```

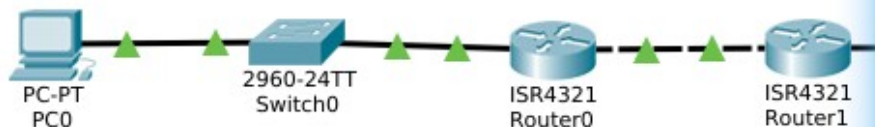
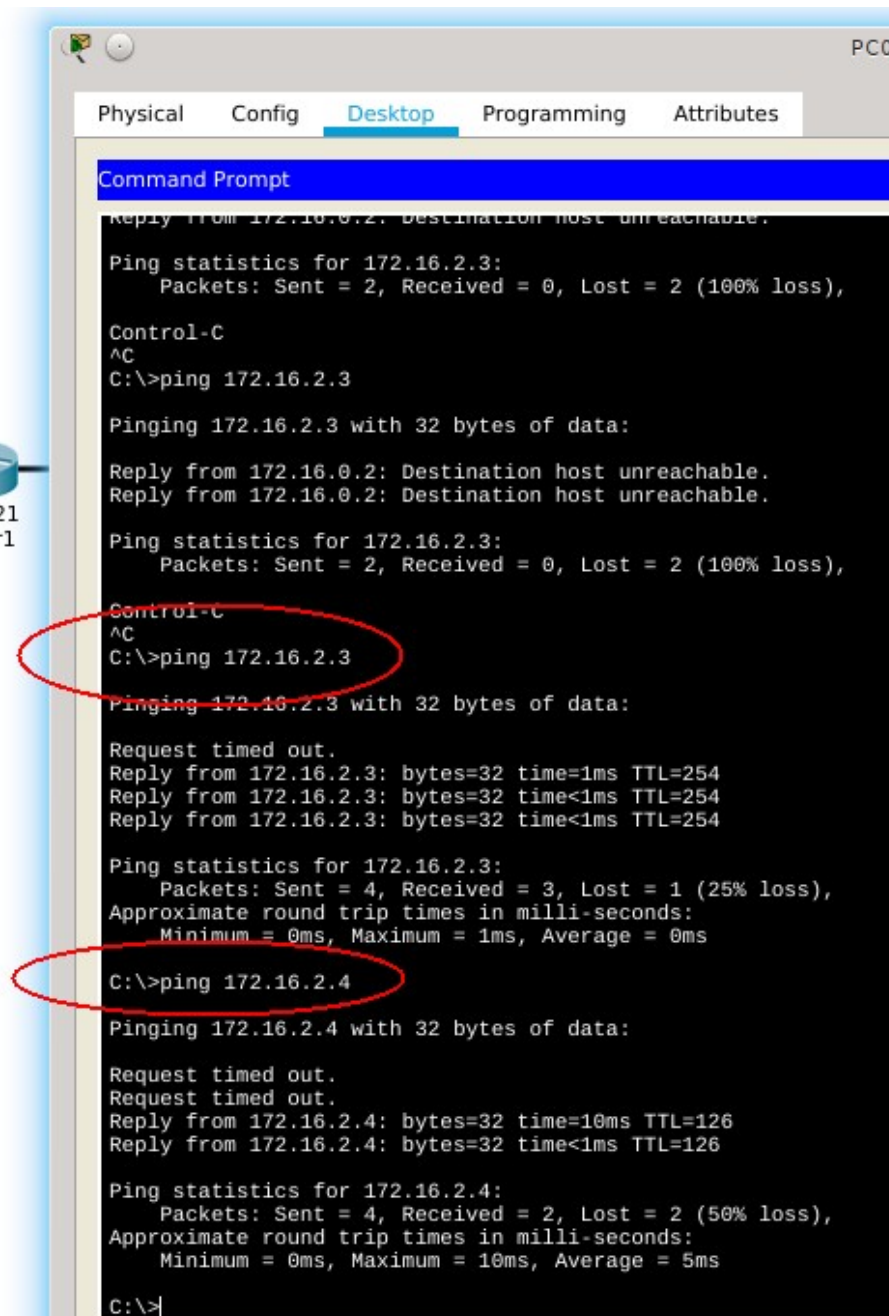
```

    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
C       172.16.0.0/24 is directly connected, GigabitEthernet0/0/0
L       172.16.0.2/32 is directly connected, GigabitEthernet0/0/0
C       172.16.1.0/24 is directly connected, GigabitEthernet0/0/1
L       172.16.1.2/32 is directly connected, GigabitEthernet0/0/1
S       172.16.2.0/24 is directly connected, GigabitEthernet0/0/1
        [1/0] via 172.16.1.3

```

O S na frente da rota da rede 172.16.2.0/24 indica que é uma rota estática. Também é possível ver apenas as rotas estáticas com o comando `show ip route static`.

Agora, realizando o teste de conectividade, realizado anteriormente, o PC0 pode “pingar” o PC1 na rede 172.16.3.0/24.

```

PC0
Physical Config Desktop Programming Attributes
Command Prompt
Reply from 172.16.0.2: Destination host unreachable.
Ping statistics for 172.16.2.3:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),

Control-C
^C
C:\>ping 172.16.2.3

Pinging 172.16.2.3 with 32 bytes of data:

Reply from 172.16.0.2: Destination host unreachable.
Reply from 172.16.0.2: Destination host unreachable.

Ping statistics for 172.16.2.3:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),

Control-C
^C
C:\>ping 172.16.2.3

Pinging 172.16.2.3 with 32 bytes of data:

Request timed out.
Reply from 172.16.2.3: bytes=32 time=1ms TTL=254
Reply from 172.16.2.3: bytes=32 time<1ms TTL=254
Reply from 172.16.2.3: bytes=32 time<1ms TTL=254

Ping statistics for 172.16.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 172.16.2.4

Pinging 172.16.2.4 with 32 bytes of data:

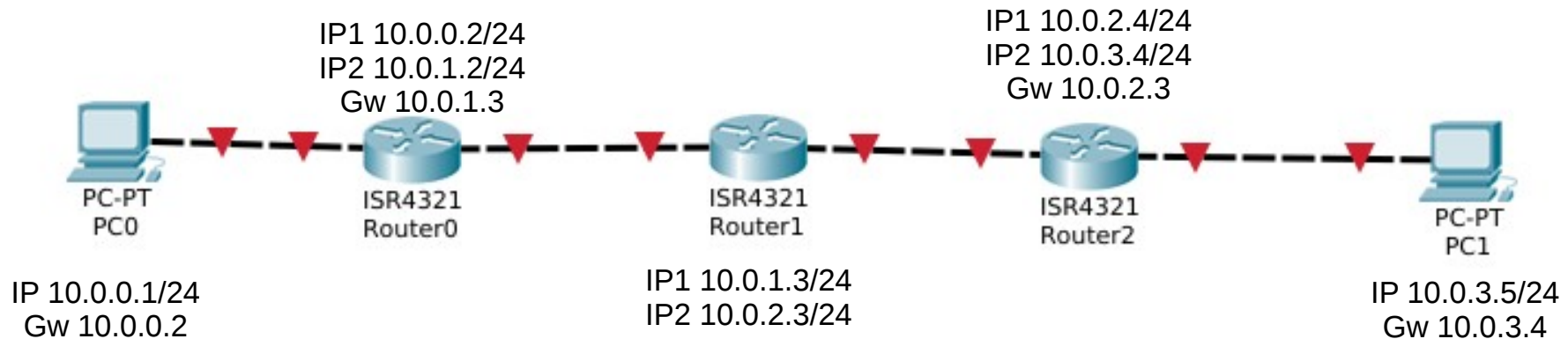
Request timed out.
Request timed out.
Reply from 172.16.2.4: bytes=32 time=10ms TTL=126
Reply from 172.16.2.4: bytes=32 time<1ms TTL=126

Ping statistics for 172.16.2.4:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 5ms

C:\>
  
```

Atividade 3

Nessa atividade vamos utilizar rota padrão para configurar o cenário de rede. Para isso configure o seguinte cenário:



Neste cenário você deve configurar no Router0 e Router2 uma rota padrão. Isso pode ser feito com o comando:

```
ip route 0.0.0.0 0.0.0.0 <iproteadorpadrão ou interface_de_saída>
```

Exemplo:

```
ip route 0.0.0.0 0.0.0.0 192.168.0.1
ip route 0.0.0.0 0.0.0.0 g0/0/0
```

No qual o 192.168.0.1 é o endereço IP do roteador/gateway padrão e g0/0/0 é a interface de rede de saída que estão conectada ao roteador padrão.

Rota padrão em roteadores CISCO

O slide anterior apresentou como criar rotas padrão em roteadores cisco,s sendo basicamente o comando:

```
ip route 0.0.0.0 0.0.0.0 <ip_rotador_padrão ou interface_de_saída>
```

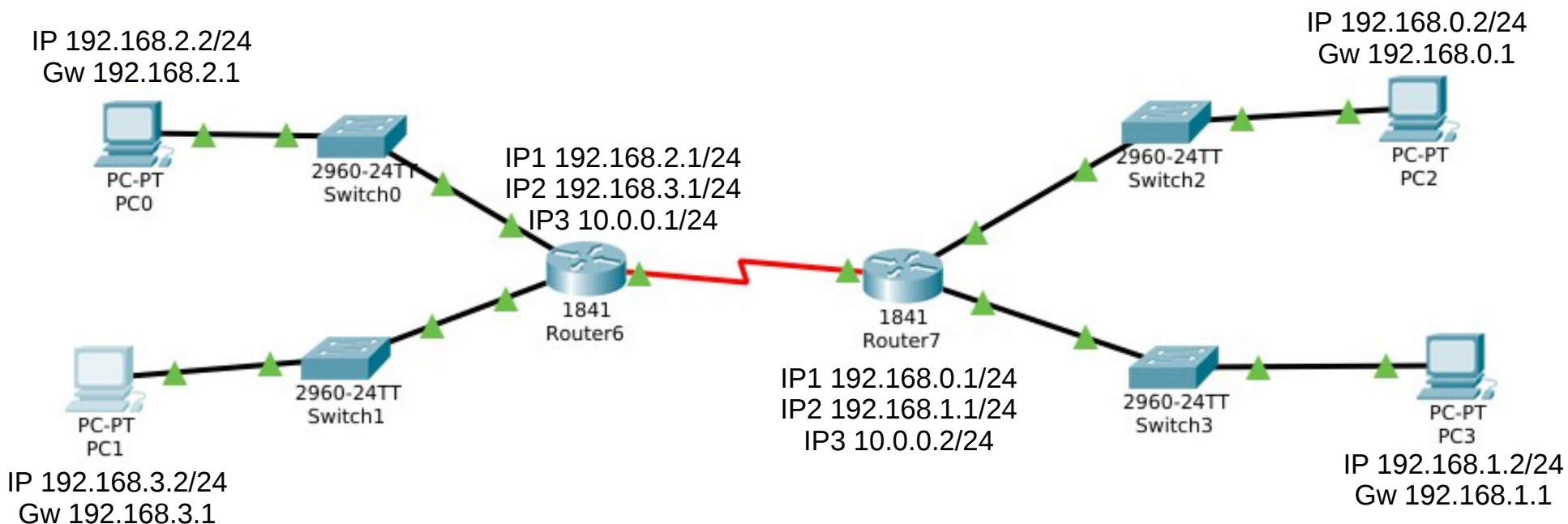
Contudo nos sistemas anteriores ao IOS 12.x é necessário utilizar o comando `ip classless` quando rotas padrão são configuradas, isso porque os roteadores CISCO trabalham por padrão com o esquema *ClassFul*, logo a máscara da rota padrão seria ignorada e falharia. Dos sistemas IOS 12.x em diante, o comando `classless` encontra-se ativado por padrão e por isso não é necessário digitá-lo.

Só lembrando que o roteamento padrão é usado no envio de pacotes para redes que não se encontram na tabela de roteamento. Ou seja, se o roteador não conhece a rota para o destino, o pacote é descartado. Porém, usando-se a rota padrão, podemos instruir o roteador, que em vez de simplesmente descartar o pacote com o destino desconhecido, o encaminhe para uma interface de saída ou endereço IP de um roteador que deve saber como entregar o pacote ao destino.

Rotas padrão, são mais indicadas em redes chamadas de *stub* (redes que possuem apenas uma interface de saída para outras redes). Como por exemplo, o modem ADSL utilizados em residências, onde os pacotes dessas redes ou acessam a rede local ou seguem para a interface de saída que leva à Internet (que é o cabo do telefone). Ou seja, não tem outra rede para acessar fora essas. Mas na prática podemos usar rota padrão em qualquer tipo de rede.

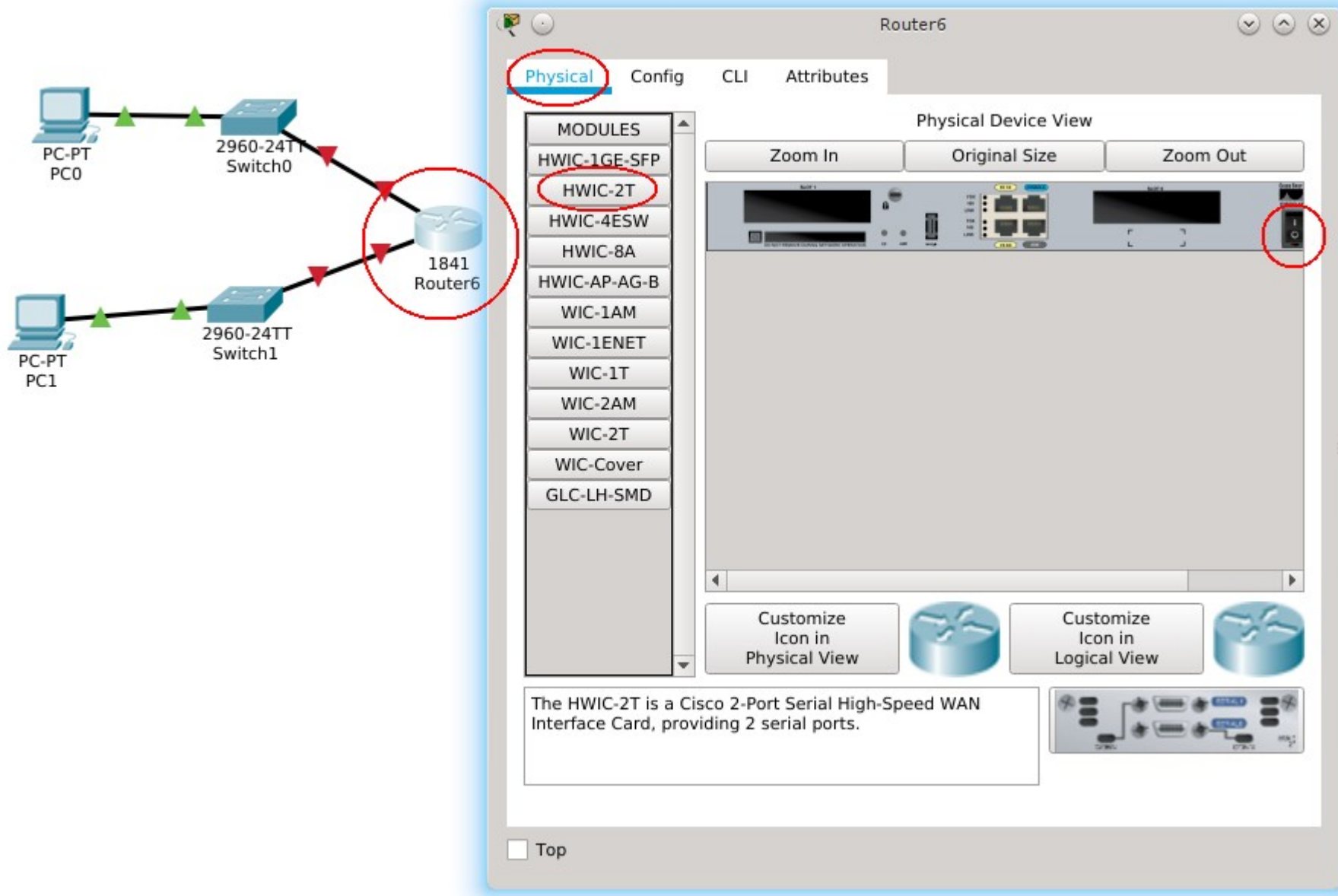
Atividade 4

Vamos agora configurar o seguinte cenário de rede:



Note que há uma ligação em vermelho, entre os roteadores. Essa é uma ligação serial (uma placa de rede). Tal placa serial não tem por padrão no roteador, então devemos adicioná-la, como um módulo, para que ele possa fazer a interligação dos roteadores, dando a possibilidade de interligar mais equipamentos, já que o padrão seria apenas duas conexões por cabo par trançado. Os próximos slides apresentam como adicionar esse módulo (note também que mudamos o modelo do roteador - 1841).

Para adicionar o módulo clique no roteador, depois na aba Physical e escolha o módulo HWIC-2T. Agora desligue o roteador, é só clicar na chave de liga/desliga da imagem.

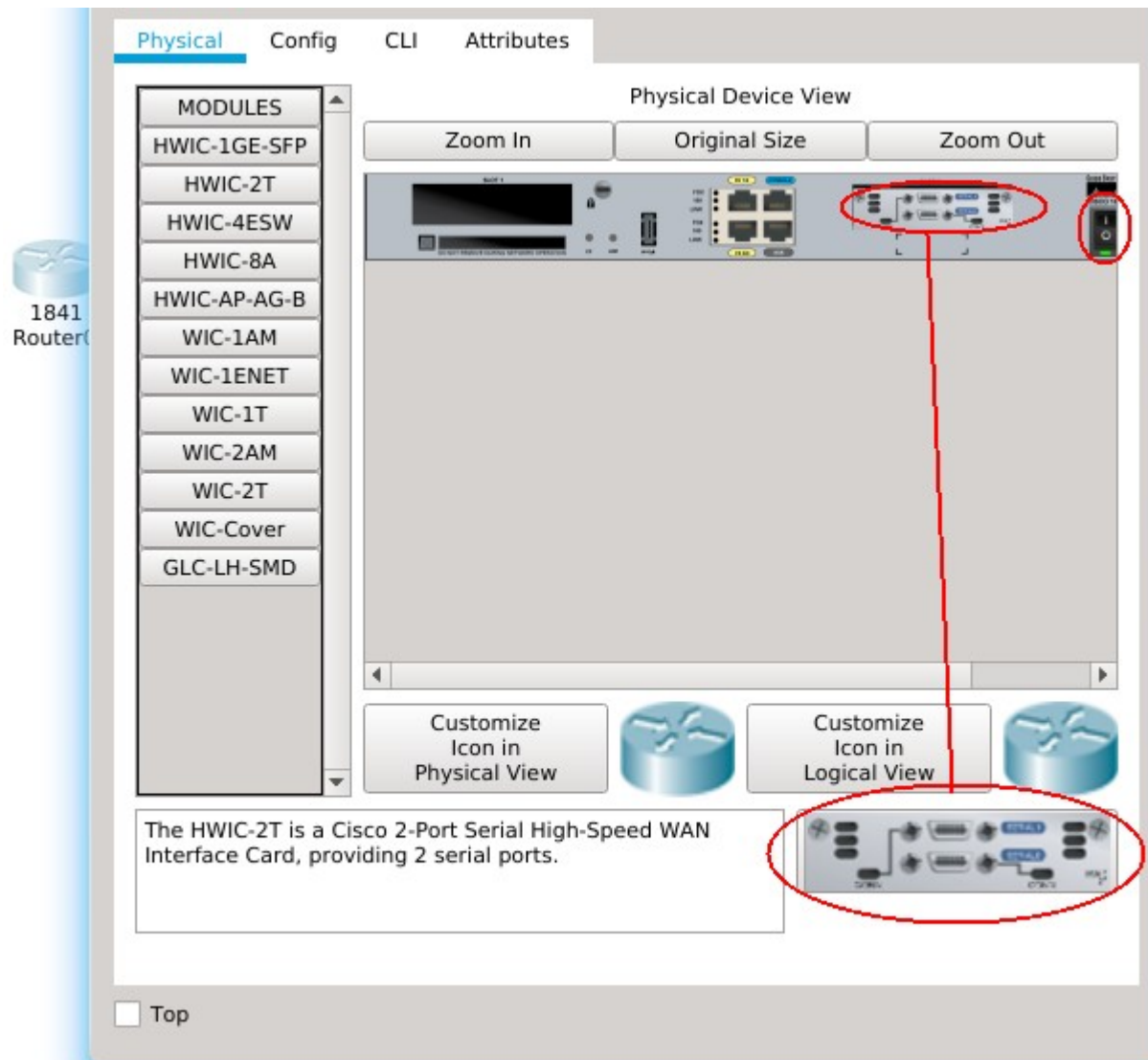


The image shows a network diagram on the left and a configuration window for 'Router6' on the right.

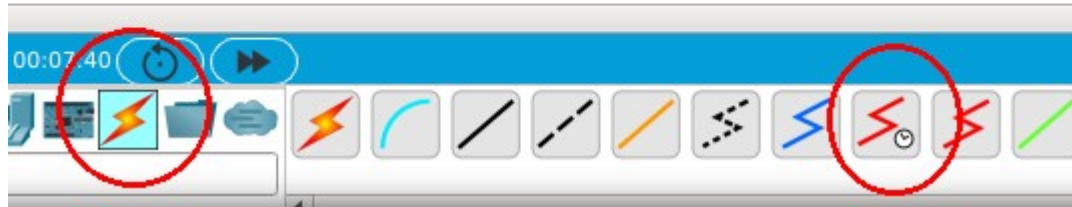
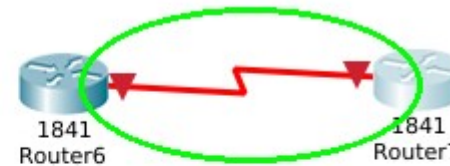
Network Diagram: Two PCs, PC-PT PC0 and PC-PT PC1, are connected to switches 2960-24TT Switch0 and 2960-24TT Switch1 respectively. Both switches are connected to a central router labeled '1841 Router6'.

Router6 Configuration Window: The window has tabs for 'Physical', 'Config', 'CLI', and 'Attributes'. The 'Physical' tab is selected and circled in red. On the left, under 'MODULES', the 'HWIC-2T' module is selected and circled in red. On the right, the 'Physical Device View' shows a router chassis with a power button circled in red. Below the chassis view are buttons for 'Customize Icon in Physical View' and 'Customize Icon in Logical View'. At the bottom, a text box states: 'The HWIC-2T is a Cisco 2-Port Serial High-Speed WAN Interface Card, providing 2 serial ports.' A 'Top' button is at the bottom left.

Agora você deve clicar na imagem do módulo no canto inferior direito e arrastá-la para o “buraco do módulo”, na imagem do roteador e soltar a imagem lá! Tal como está na figura a seguir. Por fim, você deve ligar o roteador no botão de liga/desliga da imagem. O processo de *boot* do roteador pode demorar alguns segundos.



Após isso, é possível fazer a ligação entre os roteadores através do cabo serial DCE (linha vermelha com relógio, no simulado). Depois de tal cabo, clique em um dos roteadores e aparecerá um menu, escolha uma das interfaces seriais, tal como serial0/0/0. Na sequência clique no outro roteador e também escolha uma interface serial.



Pronto, agora você pode continuar com a configuração do cenário da Atividade, tal como nós vimos nos slides anteriores. Ou seja, continue a atividade a partir daqui.

Vantagens e desvantagens do roteamento estático

Agora que já vimos teoria e prática do roteamento estático, vamos citar seus prós e contras:

- Vantagens:
 - Redução do *overhead* na CPU do roteador;
 - Não há utilização de largura de banda entre os roteadores;
 - Segurança, já que é o administrador que adiciona as rotas e por isso ele tem total controle das rotas;
- Desvantagens:
 - O administrador precisa, efetivamente, possuir profundo conhecimento da rede;
 - Se uma rede for adicionada, o administrador deve, manualmente, adicionar a rota de como alcançá-la a cada um dos roteadores;
 - Não é viável em redes de grande porte.

• Rotas dinâmicas

O roteamento dinâmico será detalhado em outro arquivo de slide Introdução ao Roteamento 2.

Referência:

FOROUZAN, Behrouz A. Data Communications and Networking. 5ª Edição. 2013.

TANENBAUM, Andrew S. Computer Networks. 5ª Edição. 2011.

CISCO. Routing and Switching Essentials Companion Guide. 3ª Edição. 2014.

COMER, Douglas E. Interligação de Redes em TCP/IP Volume 1. 5ª Edição. 2006.

FILIPPETTI, Marco A. CCNA 4.0 Guia Completo de Estudo. 2006.

ODOM, Wendell. CCENT/CCNA ICND1 Guia Oficial de Certificação do Exame. 2ª Edição. 2008.

Fim!