

# 网络技术与应用课程实验报告

## 实验名称：互联网组网与路由器配置

学号： 2211489 姓名： 冯佳明 专业： 物联网工程

### 一、实验要求：通过编程获取 IP 地址与 MAC 地址的对应关系

#### 1. 实体环境下互联网组网与路由器配置

在实体环境下完成互联网组网与路由器配置，要求如下：

- （1）在机房实验室环境下，通过将局域网划分为不同子网，用多 IP 主机作为路由器，组建互联网。
- （2）在命令行方式下，按照静态路由方式，配置路由器和主机，测试互联网的连通性。

#### 2. 仿真环境下的互联网组网与路由器配置

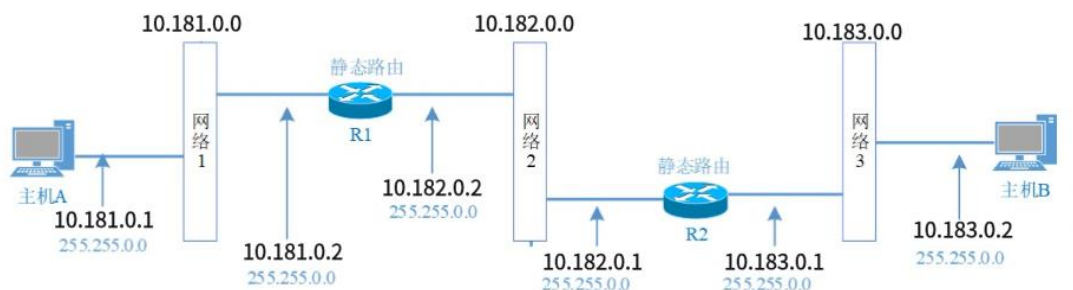
在仿真环境下完成互联网组网与路由器配置，要求如下：

- （1）学习路由器的配置方法和配置命令。
- （2）参考实体实验，组建由多个路由器组成的互联网。物理网络可以由集线器、交换机构成。
- （3）按照静态路由方式配置路由器和主机，测试互联网的连通性。
- （4）利用动态路由方式配置路由器和主机，测试互联网的连通性。
- （5）在仿真环境的“模拟”方式中观察数据包在互联网中的传递过程，并进行分析。

### 二、实验步骤

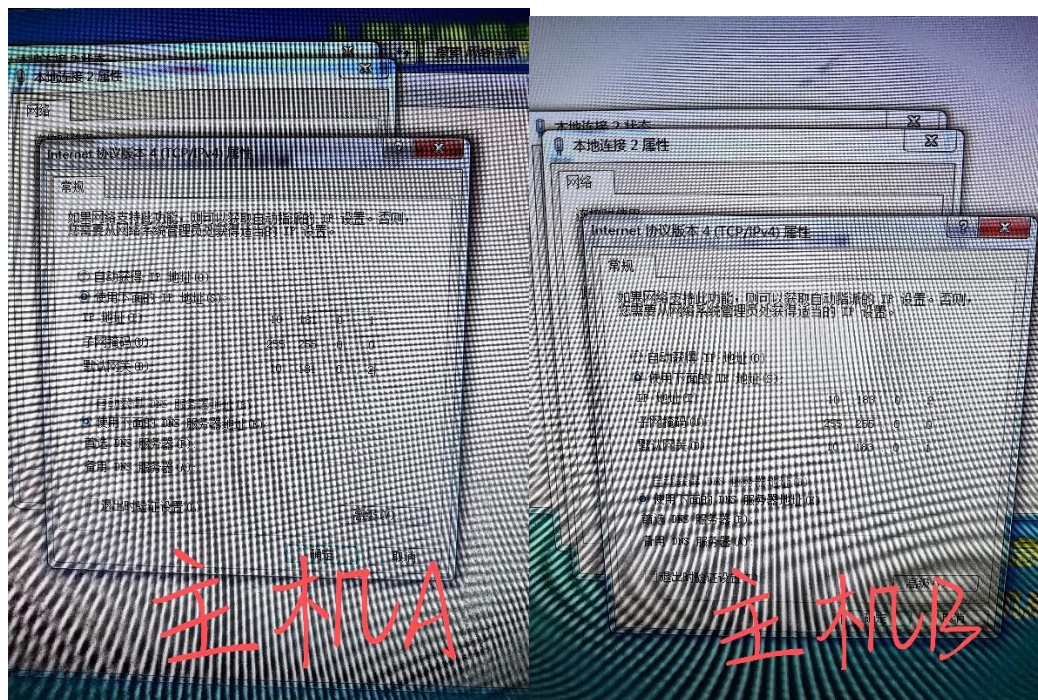
#### （一）实体环境下互联网组网与路由器配置

1. 准备工作：实验拓扑如下图所示，这里没有使用实验指导视频中的 IP 地址，因为在实验室的环境下，教室内的多台主机处于同一个局域网下，无法支持在该网络中存在多台主机的 ip 地址相同，故选择不同的 ip 来完成这个实验。



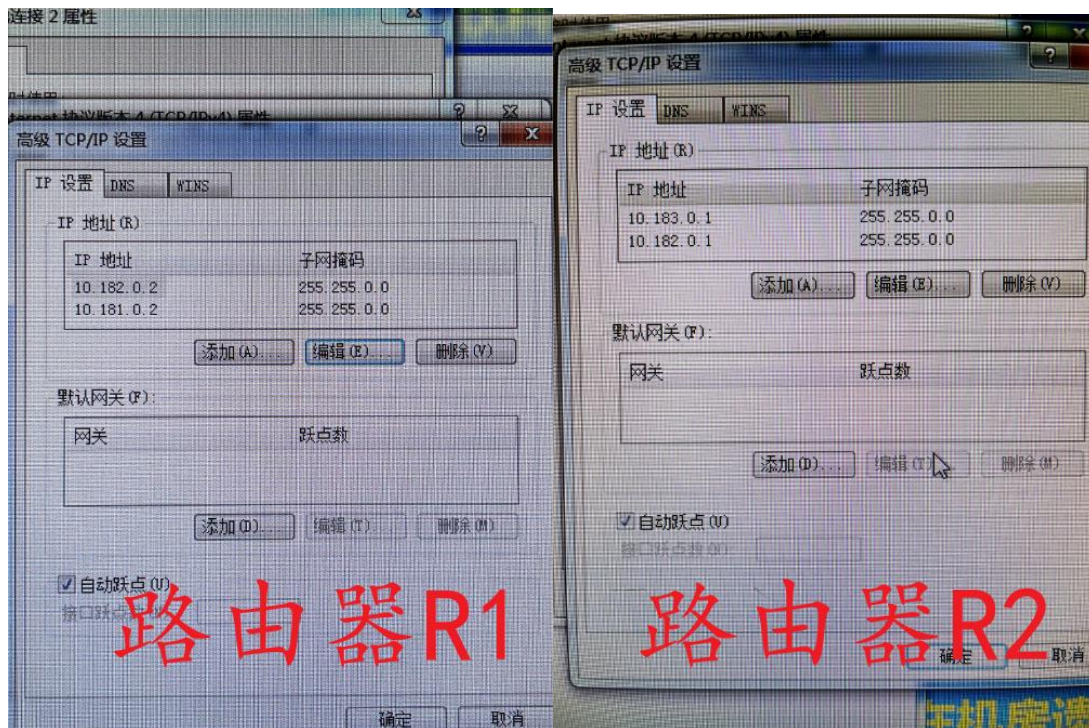
#### 2. 配置主机 IP 地址

	IP 地址	子网掩码	默认网关
主机 A	10.181.0.1	255.255.0.0	10.181.0.2
主机 B	10.183.0.2	255.255.0.0	10.183.0.1



### 3. 配置路由器 IP 地址

	IP 地址	子网掩码
路由器 R1	10.181.0.2	255.255.0.0
路由器 R1	10.182.0.2	255.255.0.0
路由器 R2	10.182.0.1	255.255.0.0
路由器 R2	10.183.0.1	255.255.0.0



### 4. 设置路由表

- (1) 使用 route ADD 10.183.0.0 MASK 255.255.0.0 10.182.0.1 添加路由器 R1 的路由表



```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation. 保留所有权利。

C:\Windows\system32>route ADD 10.181.0.0 MASK 255.255.0.0 10.182.0.1
操作完成!

C:\Windows\system32>route print
=====
接口列表
17...18 66 da 05 11 e9 .....Intel(R) Ethernet Connection I217-LM
1.....Software Loopback Interface 1
12...00 00 00 00 00 00 00 e0 Microsoft ISAIAF Adapter
=====

IPv4 路由表
=====
活动路由:
网络目标      网络掩码      网关      接口      跃点数
10.181.0.0      255.255.0.0      255.255.0.0      在链路上      10.181.0.2      266
10.181.0.2      255.255.255.255      255.255.255.255      在链路上      10.181.0.2      266
10.181.255.255  255.255.255.255      255.255.255.255      在链路上      10.181.0.2      266
10.182.0.0      255.255.0.0      255.255.0.0      在链路上      10.181.0.2      266
10.182.0.2      255.255.255.255      255.255.255.255      在链路上      10.181.0.2      266
10.182.255.255  255.255.255.255      255.255.255.255      在链路上      10.181.0.2      266
10.183.0.0      255.255.0.0      255.255.0.0      10.182.0.1      10.181.0.2      11
127.0.0.0      255.0.0.0      255.0.0.0      在链路上      127.0.0.1      306
127.0.0.1      255.255.255.255      255.255.255.255      在链路上      127.0.0.1      306
127.255.255.255  255.255.255.255      255.255.255.255      在链路上      127.0.0.1      306
224.0.0.0      240.0.0.0      240.0.0.0      在链路上      127.0.0.1      306
224.0.0.0      240.0.0.0      240.0.0.0      在链路上      10.181.0.2      266
255.255.255.255  255.255.255.255      255.255.255.255      在链路上      127.0.0.1      306
255.255.255.255  255.255.255.255      255.255.255.255      在链路上      10.181.0.2      266
=====
永久路由:
无

IPv6 路由表
=====
活动路由:
如果跃点数网络目标      网关      接口      跃点数
1      306 ::1/128      在链路上      1
17     266 fe80::/64      在链路上      17
17     266 fe80::64fb:ea9d:646d:e3d9/128 在链路上      17
1      306 ff00::/8      在链路上      1
17     266 ff00::/8      在链路上      17
=====
永久路由:
无

C:\Windows\system32>
```

(2) 使用 route ADD 10.181.0.0 MASK 255.255.0.0 10.182.0.2 添加路由器 R2 的路由表

```
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Windows\system32>route ADD 10.183.0.0 MASK 255.255.0.0 10.182.0.1
操作完成!

C:\Windows\system32>route print
=====
接口列表
17...18 66 da 05 11 e9 .....Intel(R) Ethernet Connection I217-LM
1.....Software Loopback Interface 1
12...00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
=====

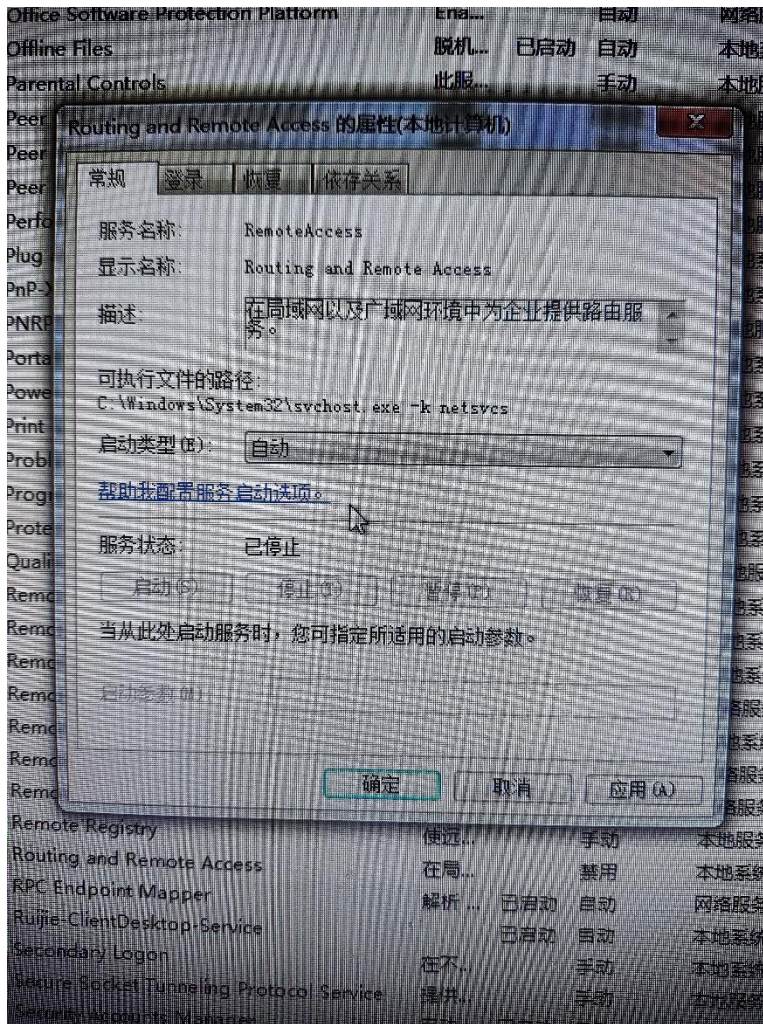
IPv4 路由表
=====
活动路由:
网络目标      网络掩码      网关      接口      跃点数
10.181.0.0      255.255.0.0      在链路上      10.181.0.2      266
10.181.0.2      255.255.255.255      在链路上      10.181.0.2      266
10.181.255.255  255.255.255.255      在链路上      10.181.0.2      266
10.182.0.0      255.255.0.0      在链路上      10.181.0.2      266
10.182.0.2      255.255.255.255      在链路上      10.181.0.2      266
10.182.255.255  255.255.255.255      在链路上      10.181.0.2      266
10.183.0.0      255.255.0.0      10.182.0.1      10.181.0.2      11
127.0.0.0      255.0.0.0      在链路上      127.0.0.1      306
127.0.0.1      255.255.255.255      在链路上      127.0.0.1      306
127.255.255.255  255.255.255.255      在链路上      127.0.0.1      306
224.0.0.0      240.0.0.0      在链路上      127.0.0.1      306
224.0.0.0      240.0.0.0      在链路上      10.181.0.2      266
255.255.255.255  255.255.255.255      在链路上      127.0.0.1      306
255.255.255.255  255.255.255.255      在链路上      10.181.0.2      266
=====
永久路由:
无

IPv6 路由表
=====
活动路由:
如果跃点数网络目标      网关      接口      跃点数
1      306 ::1/128      在链路上      1
17     266 fe80::/64      在链路上      17
17     266 fe80::64fb:ea9d:646d:e3d9/128 在链路上      17
1      306 ff00::/8      在链路上      1
17     266 ff00::/8      在链路上      17
=====
永久路由:
无

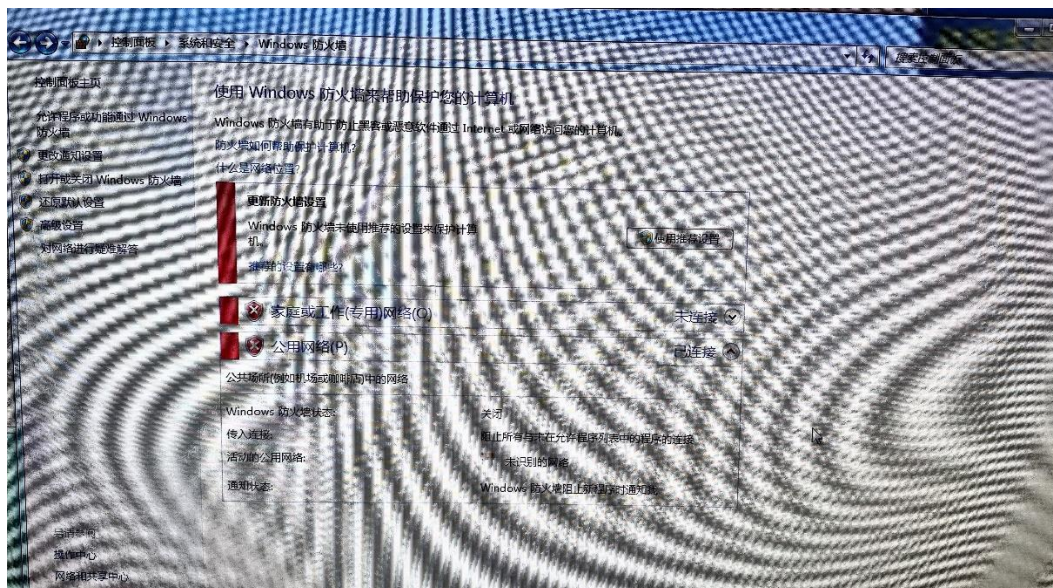
C:\Windows\system32>
```

5. 启动路由服务：在服务中找到 Routing and Remote Access 服务，并打开





## 6. 关闭主机防火墙



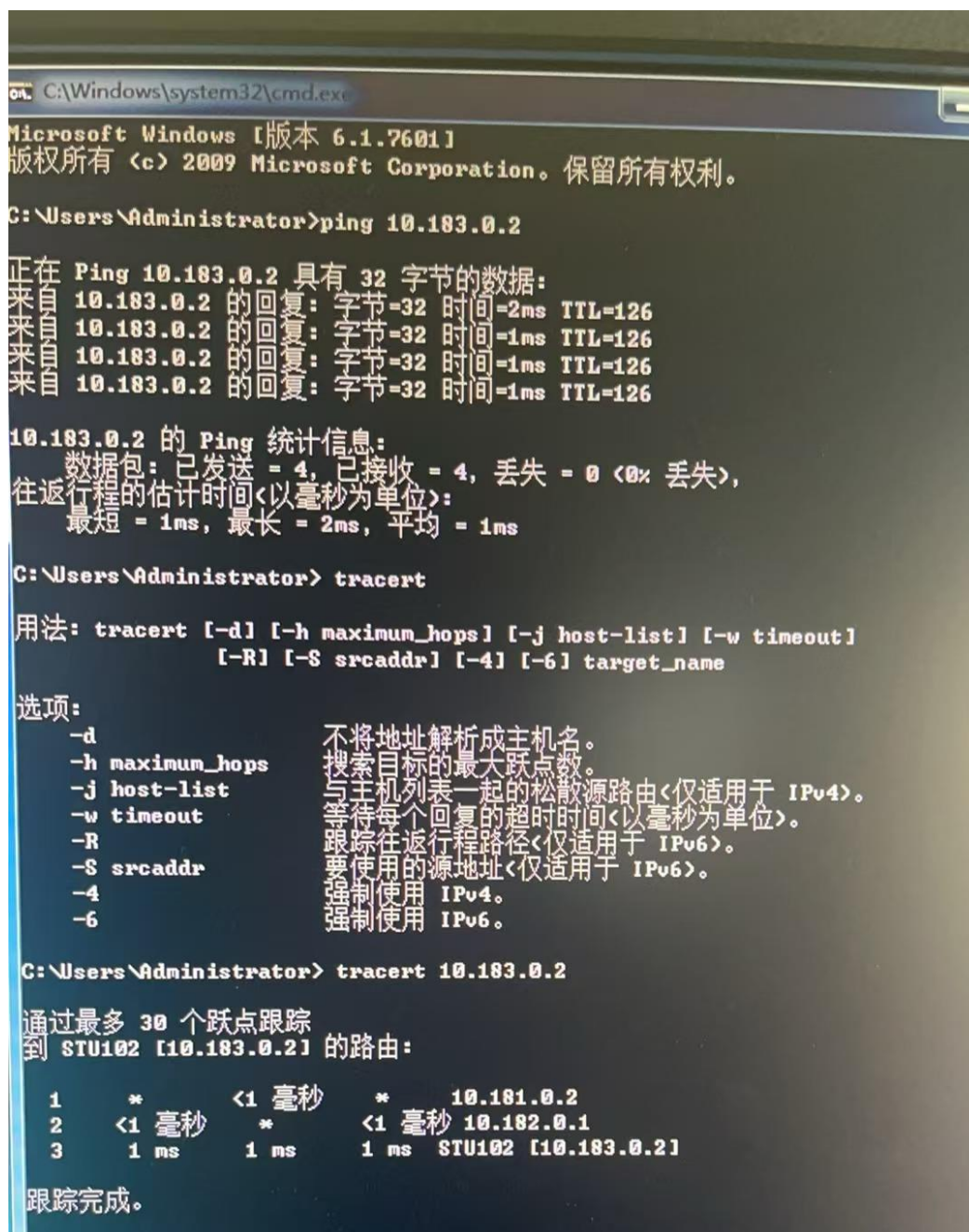
## 7. 进行连通性测试



- (1) 在主机 A ping 主机 B, 并使用 tracert

结果如下图所示, 可以看到主机 A 成功 ping 通主机 B;

数据包先发向主机 A 的默认网关 10.181.0.2, 然后路由器 R1 根据路由转发规则, 将数据包转发至 10.182.0.1, 最后路由器 R2 据其路由表项匹配规则直接投递数据包至主机 B 10.183.0.2



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>ping 10.183.0.2

正在 Ping 10.183.0.2 具有 32 字节的数据:
来自 10.183.0.2 的回复: 字节=32 时间=2ms TTL=126
来自 10.183.0.2 的回复: 字节=32 时间=1ms TTL=126
来自 10.183.0.2 的回复: 字节=32 时间=1ms TTL=126
来自 10.183.0.2 的回复: 字节=32 时间=1ms TTL=126

10.183.0.2 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间<以毫秒为单位>:
        最短 = 1ms, 最长 = 2ms, 平均 = 1ms

C:\Users\Administrator>tracert

用法: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
        [-R] [-S srcaddr] [-4] [-6] target_name

选项:
    -d          不将地址解析成主机名。
    -h maximum_hops  搜索目标的最大跃点数。
    -j host-list  与主机列表一起的松散源路由<仅适用于 IPv4>。
    -w timeout   等待每个回复的超时时间<以毫秒为单位>。
    -R          跟踪往返行程路径<仅适用于 IPv6>。
    -S srcaddr   要使用的源地址<仅适用于 IPv6>。
    -4          强制使用 IPv4。
    -6          强制使用 IPv6。

C:\Users\Administrator>tracert 10.183.0.2

通过最多 30 个跃点跟踪
到 STU102 [10.183.0.2] 的路由:

 1  *          <1 毫秒          *          10.181.0.2
 2  <1 毫秒    *          <1 毫秒  10.182.0.1
 3  1 ms       1 ms       1 ms     STU102 [10.183.0.2]

跟踪完成。
```

- (2) 在主机 B ping 主机 A, 并使用 tracert

结果如下图所示, 可以看到主机 B 成功 ping 通主机 A;

数据包先发向主机 B 的默认网关 10.183.0.1, 然后路由器 R2 根据路由转发规则, 将数据包转发至 10.182.0.2, 最后路由器 R1 根据其路由表项匹配规则直接投递数据包至主机 A 10.181.0.1

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 10.181.0.1

正在 Ping 10.181.0.1 具有 32 字节的数据:
来自 10.181.0.1 的回复: 字节=32 时间=3ms TTL=126
来自 10.181.0.1 的回复: 字节=32 时间=1ms TTL=126
来自 10.181.0.1 的回复: 字节=32 时间=1ms TTL=126
来自 10.181.0.1 的回复: 字节=32 时间=1ms TTL=126

10.181.0.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 1ms, 最长 = 3ms, 平均 = 1ms

C:\Users\Administrator>tracert 10.181.0.1

通过最多 30 个跃点跟踪
到 STU101 [10.181.0.1] 的路由:

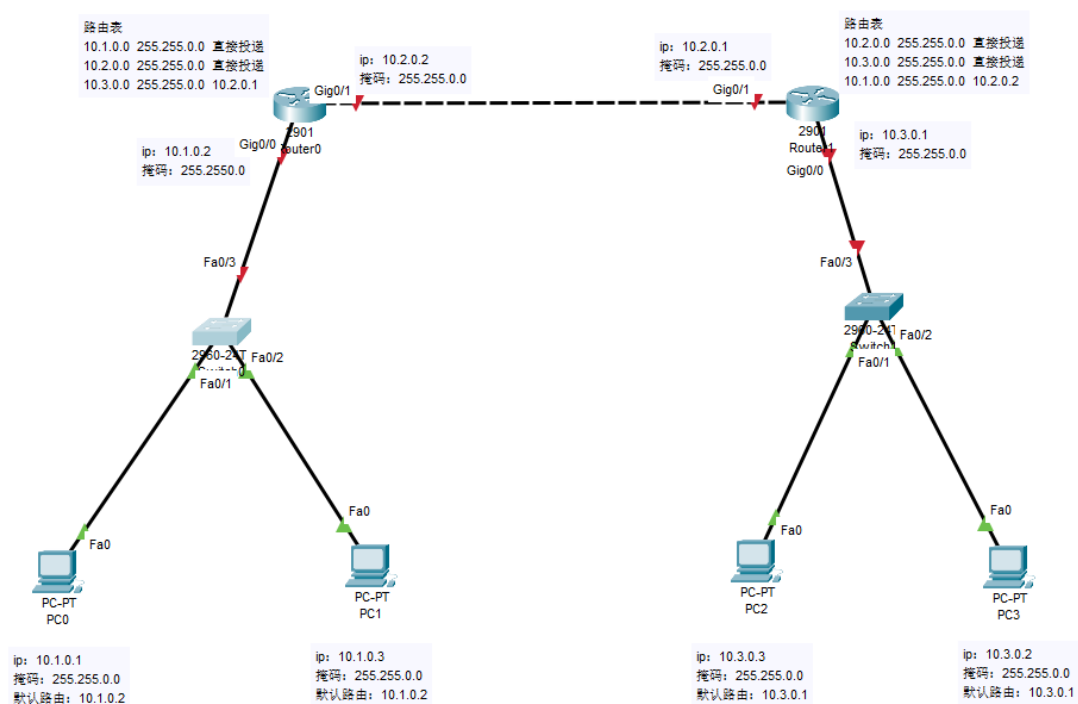
 1  *          <1 毫秒          *          10.183.0.1
 2  1 ms       *              1 ms      10.182.0.2
 3  1 ms       1 ms          1 ms      STU101 [10.181.0.1]

跟踪完成。

C:\Users\Administrator>
```

## (二) 仿真环境下的互联网组网与路由器配置

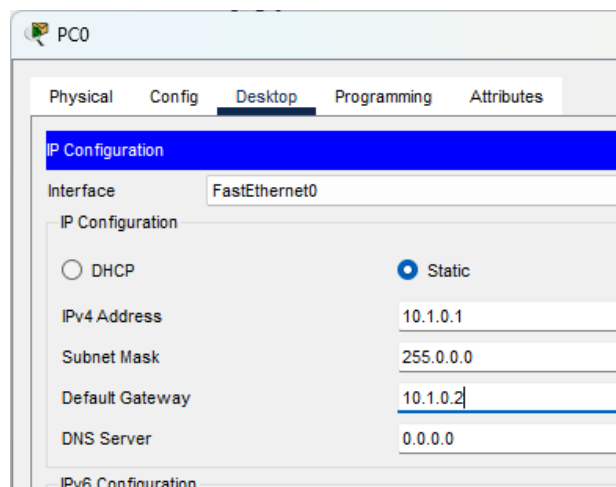
1. 拓扑结构如下图所示



## 2. 按照静态路由方式配置路由器和主机，测试互联网的连通性

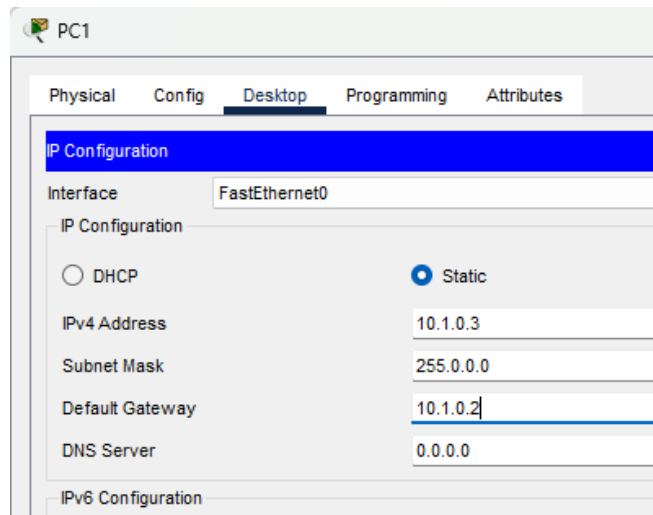
(1) 按照上述拓扑结构分别配置四台主机的 IP 地址、子网掩码、默认路由

PC0:

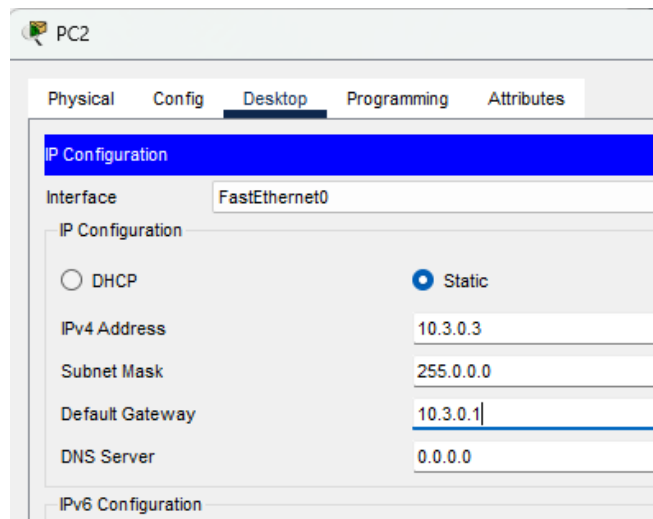


PC1:

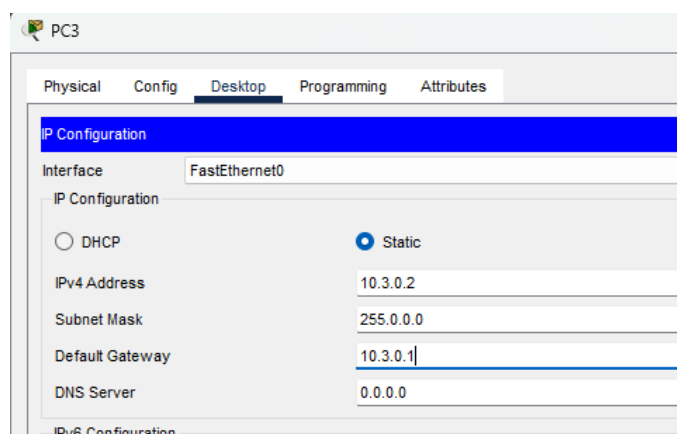




PC2:



PC3:

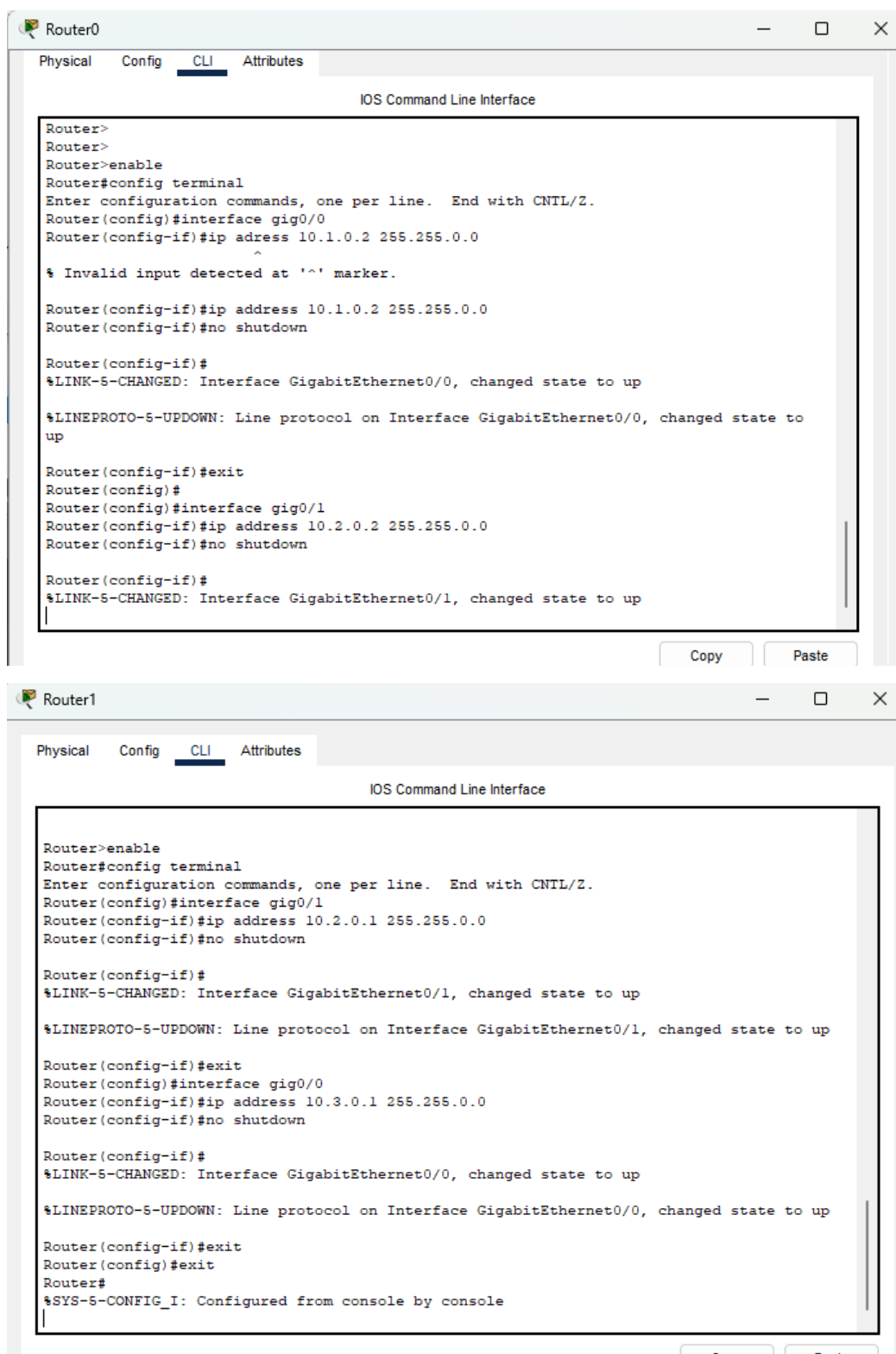


(2) 对路由器进行配置:

配置 router0: 进入 CLI 界面, 依次执行下面的指令

- ① Router>enable 进入特权模式
- ② Router#config terminal 进入全局操作模式
- ③ Router (config)# interface gig0/0 进入接口配置模式

- ④ Router (config-if)# ip address 10.1.0.2 255.255.0.0 配置接口的 ip 地址和掩码
- ⑤ Router (config-if)# no shutdown 将端口激活
- 两台路由器的配置如下图所示：



The image displays two screenshots of the Cisco Packet Tracer interface, showing the configuration of two routers, Router0 and Router1, in their CLI windows.

**Router0 Configuration:**

```
Router>
Router>
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface gig0/0
Router(config-if)#ip address 10.1.0.2 255.255.0.0
^
% Invalid input detected at '^' marker.

Router(config-if)#ip address 10.1.0.2 255.255.0.0
Router(config-if)#no shutdown

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#
Router(config)#interface gig0/1
Router(config-if)#ip address 10.2.0.2 255.255.0.0
Router(config-if)#no shutdown

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

**Router1 Configuration:**

```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface gig0/1
Router(config-if)#ip address 10.2.0.1 255.255.0.0
Router(config-if)#no shutdown

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

Router(config-if)#exit
Router(config)#interface gig0/0
Router(config-if)#ip address 10.3.0.1 255.255.0.0
Router(config-if)#no shutdown

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

- (3) 配置路由器的静态路由，两台路由器的配置如下图所示



Router0

Physical Config CLI Attributes

IOS Command Line Interface

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.3.0.0 255.255.0.0 10.2.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.1.0.0/16 is directly connected, GigabitEthernet0/0
L       10.1.0.2/32 is directly connected, GigabitEthernet0/0
C       10.2.0.0/16 is directly connected, GigabitEthernet0/1
L       10.2.0.2/32 is directly connected, GigabitEthernet0/1
S       10.3.0.0/16 [1/0] via 10.2.0.1

Router#
```

Copy Paste

Router1

Physical Config CLI Attributes

IOS Command Line Interface

```
%SYS-5-CONFIG_I: Configured from console by console

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.1.0.0 255.255.0.0 10.2.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show route
^
% Invalid input detected at '^' marker.

Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
S       10.1.0.0/16 [1/0] via 10.2.0.2
C       10.2.0.0/16 is directly connected, GigabitEthernet0/1
L       10.2.0.1/32 is directly connected, GigabitEthernet0/1
C       10.3.0.0/16 is directly connected, GigabitEthernet0/0
L       10.3.0.1/32 is directly connected, GigabitEthernet0/0

Router#
```

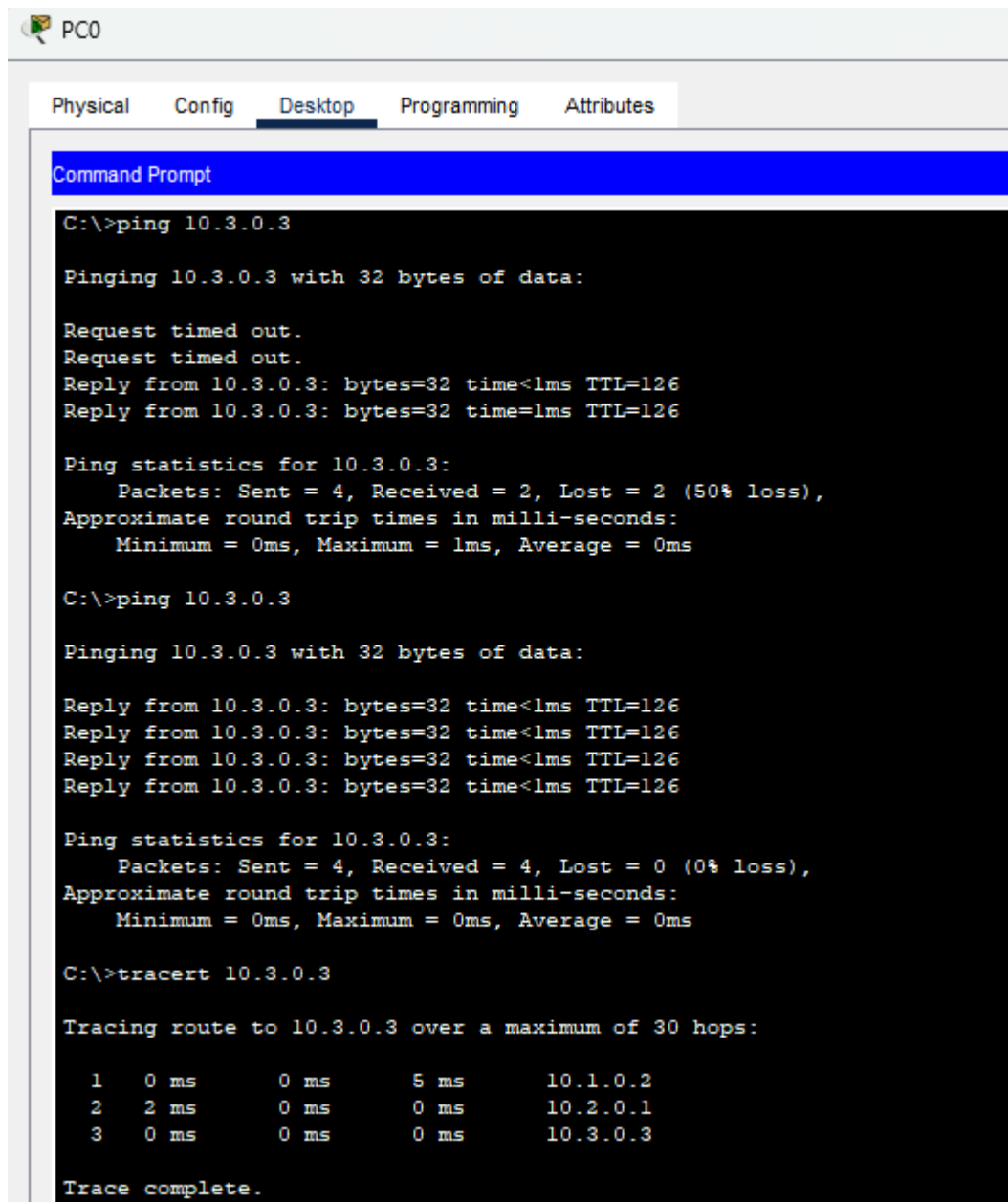
Copy Paste

#### (4) 测试网络连通性

① 使用 PC0 ping PC2: 观察到在第一次 ping 时, 前两个包丢失, 后两个包成

功返回，在第二次 ping 时，四个包都成功返回（该原因在后续的仿真中进行分析）

- ② 使用 tracert 10.3.0.3 查看数据包的路径，由路由器 0 转发至路由器 1，而后抵达 PC2



The screenshot shows the Command Prompt window on PC0. The window has tabs for Physical, Config, Desktop (selected), Programming, and Attributes. The Command Prompt displays the following text:

```
C:\>ping 10.3.0.3

Pinging 10.3.0.3 with 32 bytes of data:

Request timed out.
Request timed out.
Reply from 10.3.0.3: bytes=32 time<1ms TTL=126
Reply from 10.3.0.3: bytes=32 time=1ms TTL=126

Ping statistics for 10.3.0.3:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 10.3.0.3

Pinging 10.3.0.3 with 32 bytes of data:

Reply from 10.3.0.3: bytes=32 time<1ms TTL=126
Reply from 10.3.0.3: bytes=32 time<1ms TTL=126
Reply from 10.3.0.3: bytes=32 time<1ms TTL=126
Reply from 10.3.0.3: bytes=32 time<1ms TTL=126

Ping statistics for 10.3.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>tracert 10.3.0.3

Tracing route to 10.3.0.3 over a maximum of 30 hops:

  0  0 ms    0 ms    5 ms    10.1.0.2
  1  2 ms    0 ms    0 ms    10.2.0.1
  2  0 ms    0 ms    0 ms    10.3.0.3

Trace complete.
```

- (5) 在仿真环境的“模拟”方式中观察数据包在互联网中的传递过程，并进行分析

- ① PC1 ping PC3，运行结果如下图所示，观察到前两个包超时，但成功接收到了后两个包，此时，查看仿真结果
- ② 根据仿真结果进行分析：

可以看到，PC1 第一次向 PC3 发包时，由于 PC1 不知道路由器的 MAC 地址，所以主机 1 第一次发送的是 ARP 包，路由器将自身 MAC 地址返回 PC1；

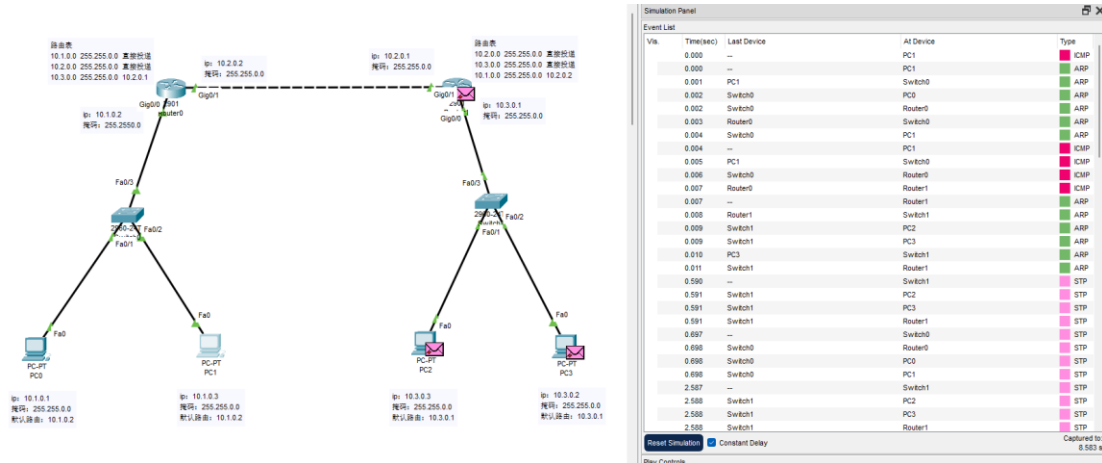
此时，因为 PC1 已知路由器地址，所以直接发送 ICMP 报文，但是由



于路由器 0 不知道路由器 1 的地址，所以在此处转发的是 ARP 数据包形式，路由器 1 返回 MAC 地址；

因此前两次的包都没有成功到达 PC3；

第三次发包通过路由器转发后，PC3 可以回复来自 PC1 的报文



### 3. 利用动态路由方式配置路由器和主机，测试互联网的连通性

#### (1) 利用 RIP 动态配置环境

##### 配置 router 0

```
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 10.1.0.0
Router(config-router)#network 10.2.0.0
Router(config-router)#exit
Router(config)#exit
```

##### 查看 router 0 的路由表

```
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C    10.1.0.0/16 is directly connected, GigabitEthernet0/0
L    10.1.0.2/32 is directly connected, GigabitEthernet0/0
C    10.2.0.0/16 is directly connected, GigabitEthernet0/1
L    10.2.0.2/32 is directly connected, GigabitEthernet0/1
R    10.3.0.0/16 [120/1] via 10.2.0.1, 00:00:20,
GigabitEthernet0/1
```

Router#

##### 配置 router 1

```
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 10.2.0.0
Router(config-router)#network 10.3.0.0
Router(config-router)#exit
Router(config)#exit
Router#
```

##### 查看 router 1 的路由表

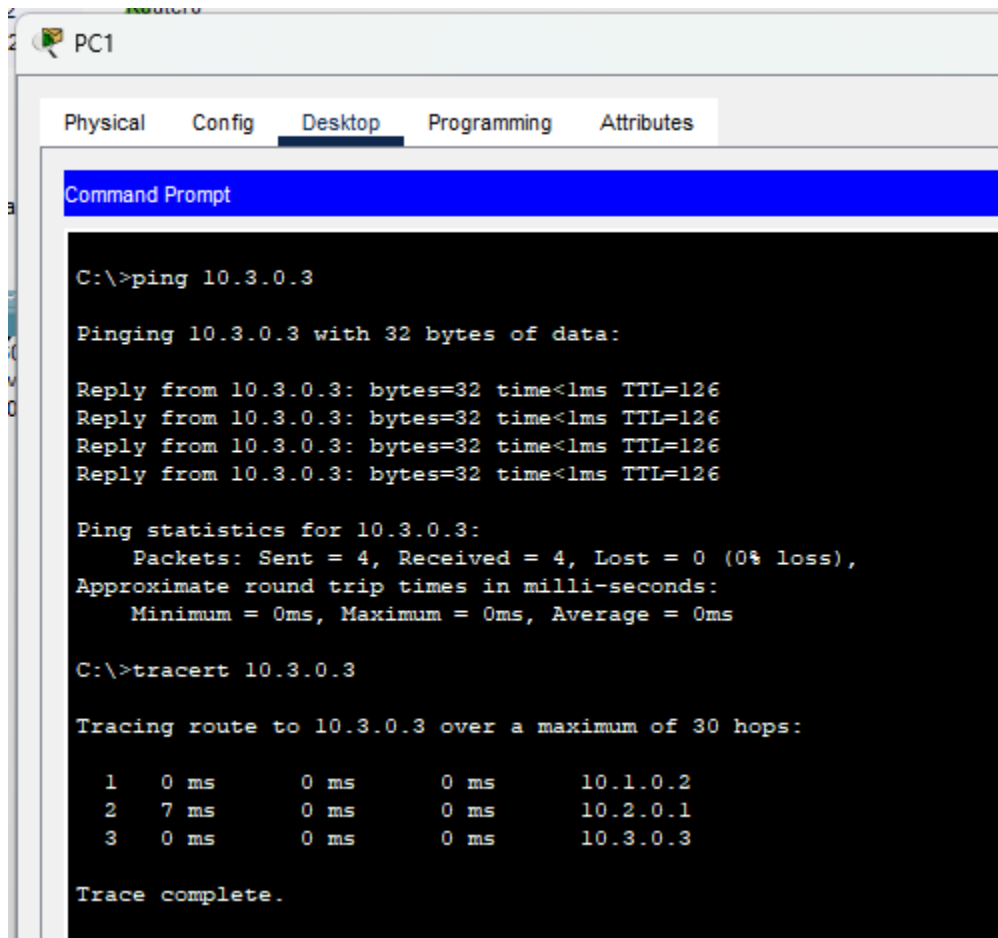
```

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
R      10.1.0.0/16 [120/1] via 10.2.0.2, 00:00:24,
GigabitEthernet0/1
C      10.2.0.0/16 is directly connected, GigabitEthernet0/1
L      10.2.0.1/32 is directly connected, GigabitEthernet0/1
C      10.3.0.0/16 is directly connected, GigabitEthernet0/0
L      10.3.0.1/32 is directly connected, GigabitEthernet0/0
Router#

```

(2) 测试网络连通性

- ① 使用 PC1 ping PC2: 观察到在第一次 ping 时, 四个包都成功返回 (该原因在后续的仿真中进行分析)
- ② 使用 tracert 10.3.0.3 查看数据包的路径, 由路由器 0 转发至路由器 1, 而后抵达 PC2



The screenshot shows a PC1 window with the 'Desktop' tab selected. A Command Prompt window is open, displaying the results of a ping and a tracert command. The ping command shows four successful replies from 10.3.0.3. The tracert command shows the path from 10.1.0.2 to 10.2.0.1 to 10.3.0.3.

```

C:\>ping 10.3.0.3

Pinging 10.3.0.3 with 32 bytes of data:

Reply from 10.3.0.3: bytes=32 time<1ms TTL=126
Reply from 10.3.0.3: bytes=32 time<1ms TTL=126
Reply from 10.3.0.3: bytes=32 time<1ms TTL=126
Reply from 10.3.0.3: bytes=32 time<1ms TTL=126

Ping statistics for 10.3.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>tracert 10.3.0.3

Tracing route to 10.3.0.3 over a maximum of 30 hops:

  0  0 ms    0 ms    0 ms    10.1.0.2
  1  7 ms    0 ms    0 ms    10.2.0.1
  2  0 ms    0 ms    0 ms    10.3.0.3

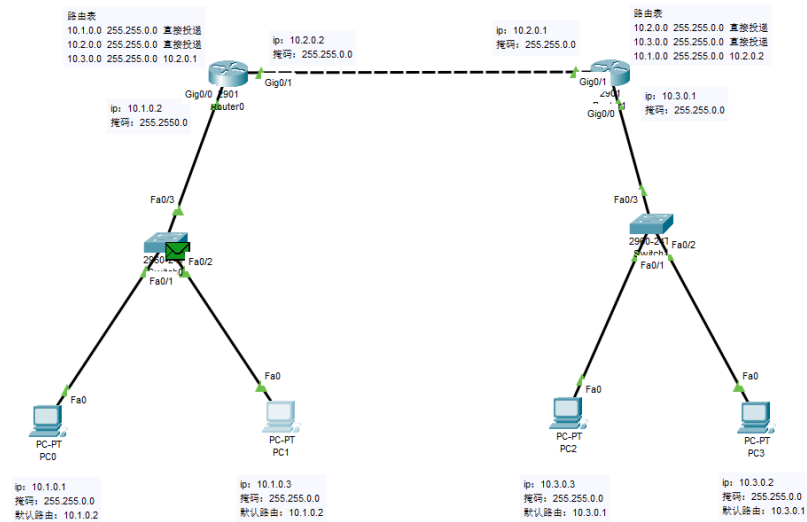
Trace complete.

```

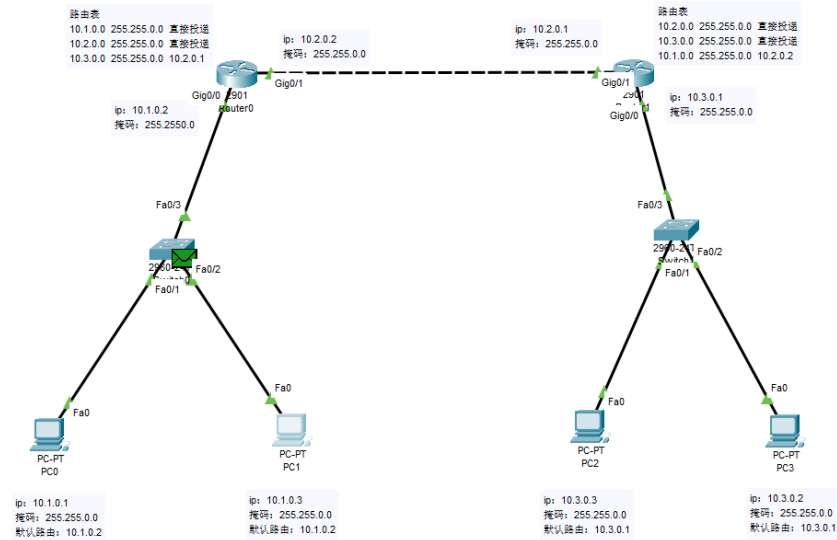
(3) 在仿真环境的“模拟”方式中观察数据包在互联网中的传递过程, 并进行分析, 观察 PC1 ping PC3

- ① PC1 将 ICMP 报文发送给 switch0

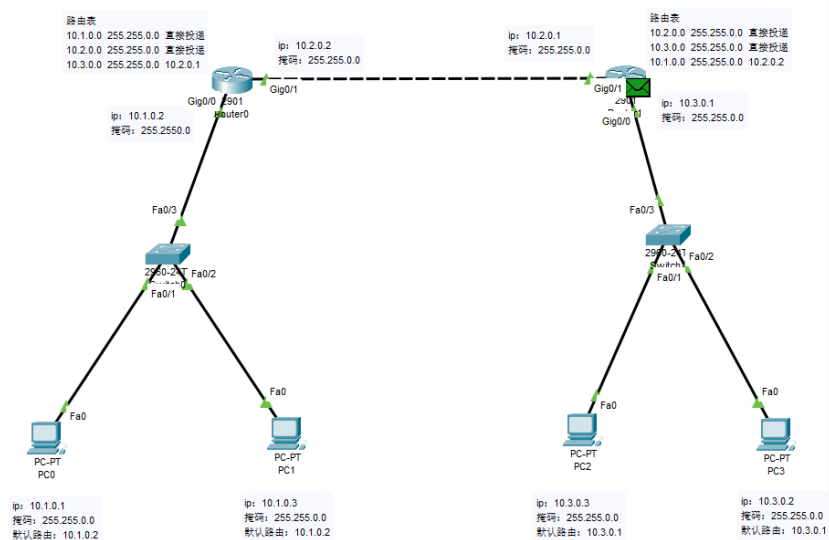




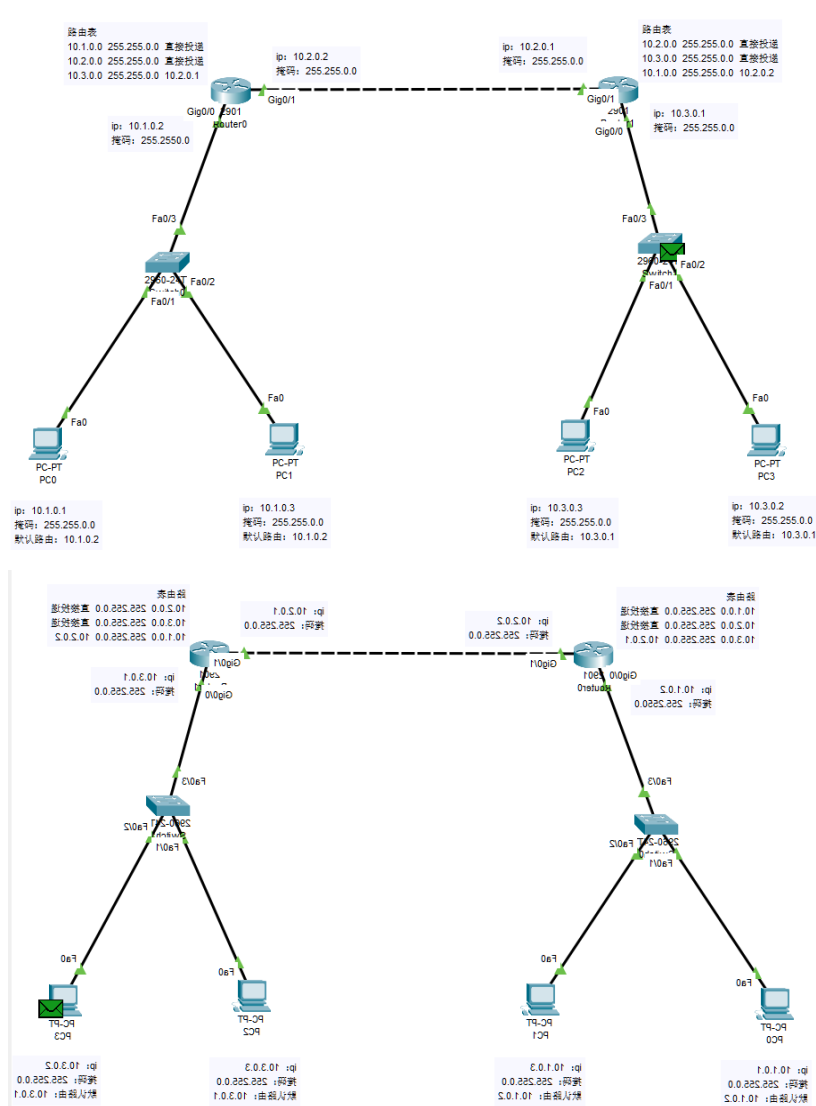
## ② Switch0 将报文发送给 router0



## ③ Router0 按照路由表将报文发送给 router1



#### ④ 报文经 switch1 发送至 PC3



#### ⑤ 随后进行返程，过程与之前类似，不再赘述

Event List				
Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC1	ICMP
	0.001	PC1	Switch0	ICMP
	0.002	Switch0	Router0	ICMP
	0.003	Router0	Router1	ICMP
	0.004	Router1	Switch1	ICMP
	0.005	Switch1	PC3	ICMP
	0.006	PC3	Switch1	ICMP
	0.007	Switch1	Router1	ICMP
	0.008	Router1	Router0	ICMP
	0.009	Router0	Switch0	ICMP
	0.010	Switch0	PC1	ICMP