

# 实验二：路由器基本配置

(原为个人信息，此处隐藏)

## 一、实验目的

- 掌握路由器的基本知识
- 掌握路由器端口的配置
- 掌握路由协议的基本配置
- 熟悉使用 **Boson Netsim** 的相关操作

## 二、实验内容

- 使用 **IOS** 命令配置路由器
- 掌握 **静态路由** 和 **动态路由** (**RIP**、**OSPF**) 的配置方法

## 三、实验要求

- 本实验要求自行构建一个 **网络拓扑**，要求包括3个以上路由器（路由器采用 **串行连接**），用于连接两个以太网，每个以太网至少包括1台主机；
- 完成路由器、主机等设备的配置；使用 **RIP** 或 **OSPF** 来维护路由器的路由表；
- 实验配置完成后，两台主机要能够相互ping通；
- 实验报告要包括 **网络拓扑**、**配置** 以及 **结果**。

## 四、实验内容及分析

### 1. 搭建网络拓扑

本实验的网络结构较为简单，总共有 **2** 台 **PC机** (**PC1**、**PC2**) 与 **3** 个 **路由器** (**Router1**、**Router2**、**Router3**) 组成。其中 **PC1** 与 **Router1** 连接，各路由器 **串行连接**，另外的一台PC机与另一边的 **Router3** 连接，形成线性的拓扑结构。PC机与路由器之间通过 **以太网接口 (Ethernet)** 连接，而路由器之间通过 **串行接口 (Serial)** 连接，因此本实验需要选择同时具有两种接口的路由器；同时，本结构较为简单，最基础的 **以太网** 技术就可以胜任。经过查阅相关资料，发现 **2500 Series** 的路由器同时具备两种接口，因此三台路由器均选择 **2500** 型号，每台有2个 **串口** 和1个 **以太口**。**PC1** 与 **Router1** 的两个 **Ethernet0** 端口相连，**Router1** 与

Router2 的两个 Serial0 相连，而 Router2 和 Router3 的 Serial1 相连，最后 Router3 和 PC2 的 Ethernet0 相连。形成如下的拓扑结构：

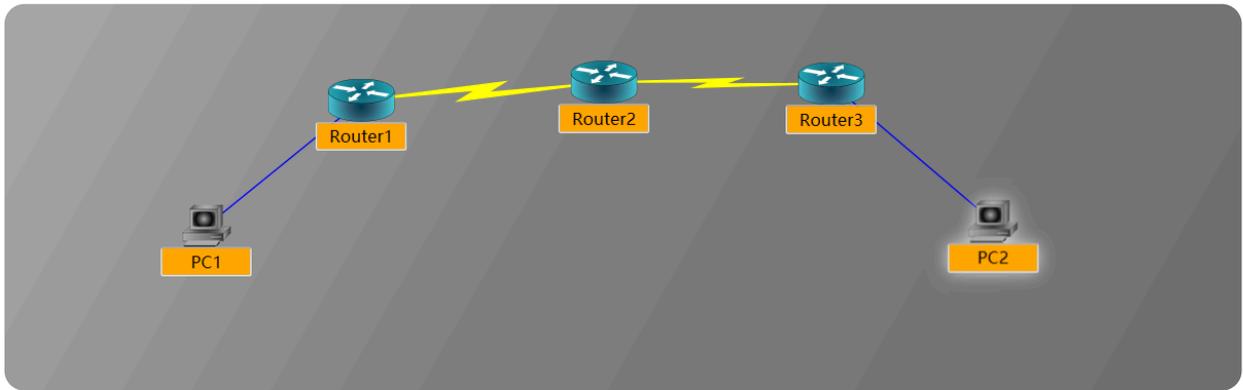


图1 拓扑结构简图

用 Mermaid 标记语言绘制网络图，表示如下：

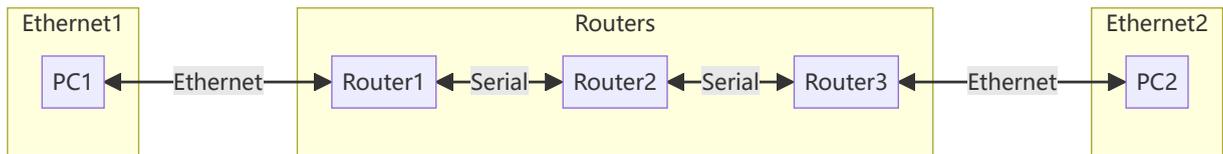
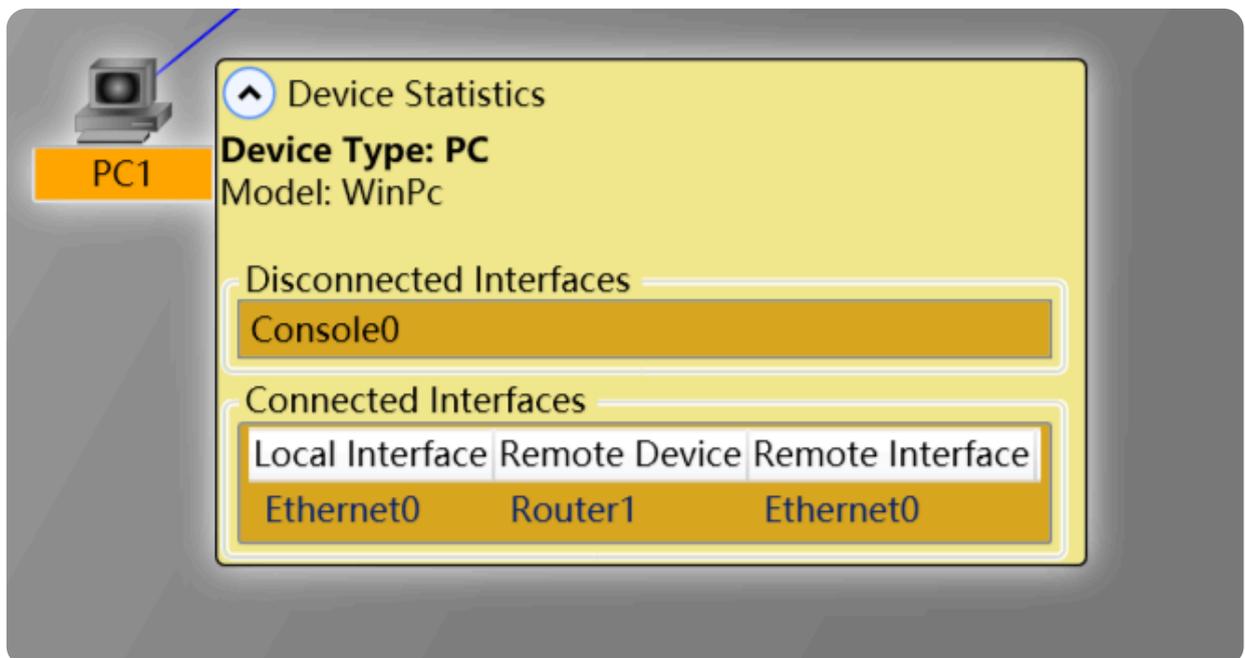


图2 网络拓扑图绘制

端口连接对应情况如下：





Router1

Device Statistics

Device Type: Router

Model: 2501

Series: 2500

Disconnected Interfaces

BRI0

Serial1

Console0

Connected Interfaces

Local Interface	Remote Device	Remote Interface
-----------------	---------------	------------------

Ethernet0	PC1	Ethernet0
-----------	-----	-----------

Serial0	Router2	Serial0
---------	---------	---------



Router2

Device Statistics

Device Type: Router

Model: 2501

Series: 2500

Disconnected Interfaces

BRI0

Ethernet0

Console0

Connected Interfaces

Local Interface	Remote Device	Remote Interface
-----------------	---------------	------------------

Serial0	Router1	Serial0
---------	---------	---------

Serial1	Router3	Serial1
---------	---------	---------

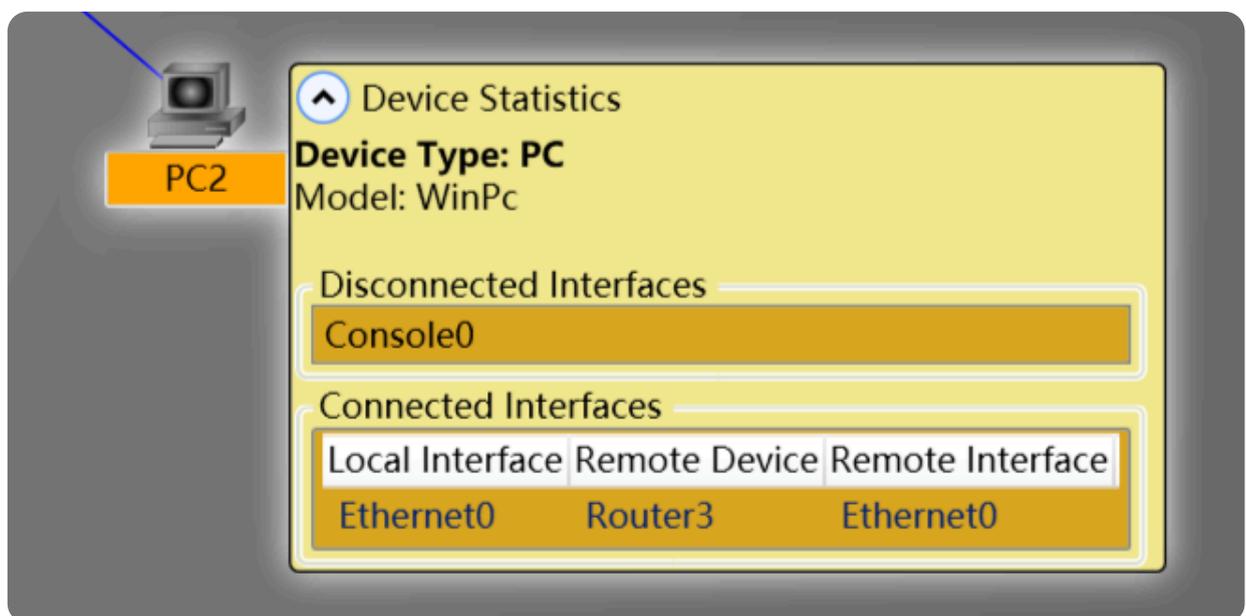
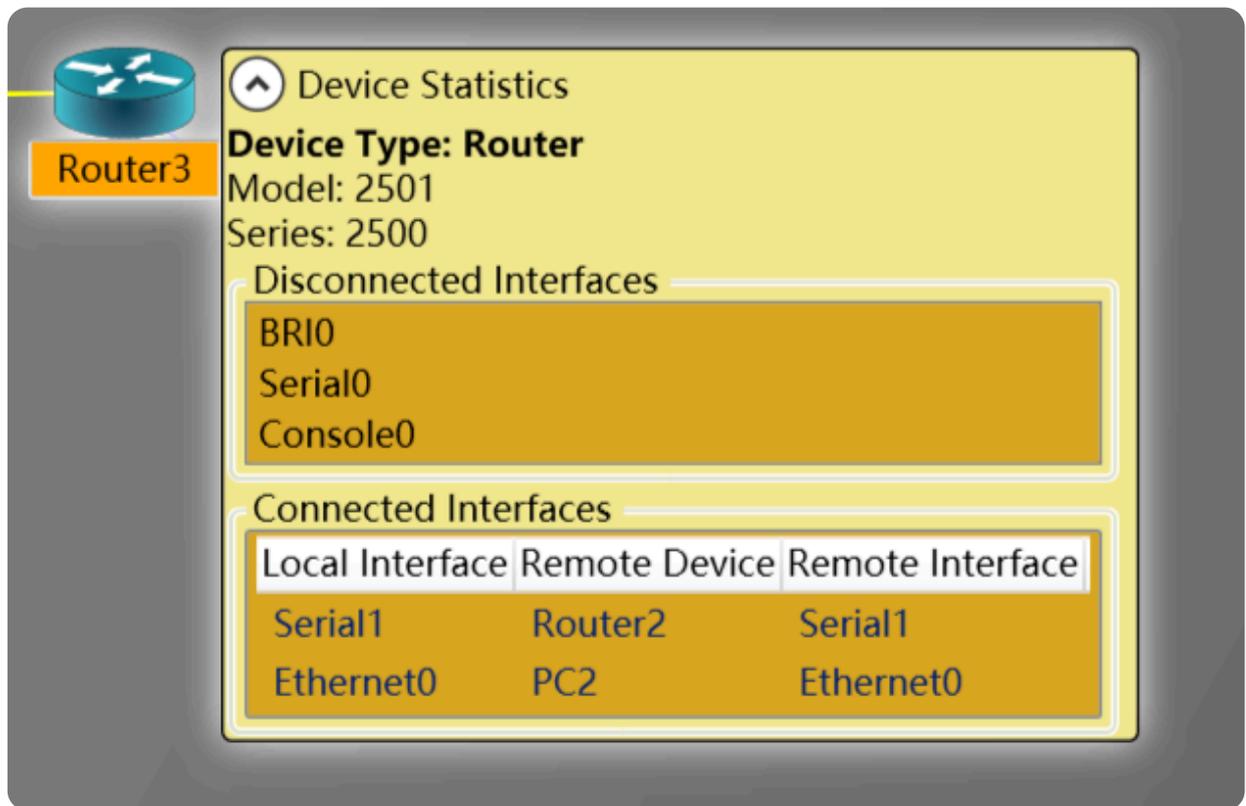


图3 端口连接情况

## 2. 基本配置

### (1) IP配置原则

本实验试图实现两个以太网之间的通信，两台PC机各自处于不同的以太网之内，分别设这两个以太网的 IPv4 地址为 192.168.16.0 与 192.168.17.0。因为设备较少，不需要增加交换机等设备，令PC机和各路由器的 Ethernet 端口直接相连就好，路由器与PC机相连端口的IP会成为PC机的默认网关地址。PC机的IP地址需要与直连的路由器 Ethernet 端口处在同一子网中，两个路由器之间的 Serial 端口也需要处在同一子网当中。这是因为当设备需要通信时，会使用 ARP (地址解析协议) 来解析对方的 MAC 地址。ARP 广播只能在同一个子网内传播。如果两个设备不在同一个子网，ARP

请求无法到达对方，通信也无法建立。而同一个设备的两个端口能不能处于同一个子网呢？答案也是**不能**。因为在通信时路由器的每个接口都相当于一个独立的网络（广播域）。路由器是需要通过**路由表**转发收到的**数据包**，如果两个接口配置在同一子网中，路由器无法区分流量来自哪个接口，可能导致路由表混乱和通信失败。因此最后得到的IP地址分配情况如下：

设备	端口	IP地址	子网掩码
PC1	Ethernet0	192.168.16.2	255.255.255.0
Router1	Ethernet0	192.168.16.1	255.255.255.0
Router1	Serial0	192.168.1.1	255.255.255.0
Router2	Serial0	192.168.1.2	255.255.255.0
Router2	Serial1	192.168.2.1	255.255.255.0
Router3	Serial1	192.168.2.2	255.255.255.0
Router3	Ethernet1	192.168.17.1	255.255.255.0
PC2	Ethernet0	192.168.17.2	255.255.255.0

表1 各设备端口IP分配情况

## (2) 数据链路层协议选择

数据链路层的位于**OSI参考模型**的第二层，其主要任务是在物理线路上基于**比特流**提供**点对点 (Point to Point)**的、**可靠**的数据传输，使之对网络层呈现为一条**“无错”**的线路。该层之封装协议的主要作用是来自**网络层**的数据包封装成**帧 (Frame)**，实现**帧定界**、**透明传输**、**差错控制**等任务，并提供**逻辑链路控制 (LLC)**和**介质访问控制 (MAC)**的功能。数据链路层封装协议的具体形式因网络类型和技术而异，如**以太网**、**PPP (点对点协议)**、**HDLC (高层数据链路控制)**、**帧中继**等。串行线上最常用的两个数据链路层通信协议是**串行线路网际协议 (SLIP)**和**点对点协议**，后者产生时间较晚，但其**灵活性**、广泛的**协议支持**和基本的**身份验证**功能，使其逐渐成为**点对点通信**的标准协议。

特性	PPP	HDLC
协议支持	支持多种网络层协议（如 IPv4、IPv6）。	Cisco HDLC 支持多协议，标准 HDLC 不支持。
认证机制	支持 <b>PAP</b> 和 <b>CHAP</b> 身份验证。	不支持身份验证。

特性	PPP	HDLC
错误检测	提供 <b>CRC</b> 错误检测。	提供 <b>CRC</b> 错误检测。
链路管理	提供 <b>LCP</b> 和 <b>NCP</b> 功能。	不支持链路管理功能。
互操作性	设备间广泛兼容。	Cisco HDLC 仅限于 Cisco 设备之间。
扩展性	支持压缩、加密等扩展功能。	不支持扩展。
性能开销	较高（由于支持多种功能）。	较低（功能简单）。
使用场景	广泛应用于 WAN 链路、拨号网络（ <b>PPPoE</b> ）等。	通常用于简单的 <b>点对点链路</b> 。

表2 PPP与HDLC部分特性比较

本实验采用**PPP**协议作为三个路由器之间传输帧的封装协议。

### 3. 路由配置

具体各网络互联设备的路由配置参照以上原则确定。首先配置**路由器**。



Press Enter to Start

Router> #Router1界面

Router>enable # 进入特权模式，随后标志符会变成'#'，可以查看路由信息，进行ping操作等，但是还无法具体配置各个端口。

Router#configure terminal # 进入全局配置模式，可以进行一些全局配置操作，但是无法具体配置特定端口

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#hostname Router1 # 改名

Router1(config)#interface ethernet 0 #切换到端口配置模式，单独配置Ethernet端口各种措施

Router1(config-if)#ip address 192.168.16.1 255.255.255.0 #设置Ethernet0处端口IPv4地址和子网掩码，必须和PC机1在一个子网

Router1(config-if)#no shutdown #启用以太网接口

%LINK-3-UPDOWN: Interface Ethernet0, changed state to up

Router1(config-if)#interface serial 0 #转入Serial 0串口，与Router2相连

Router1(config-if)#encapsulation ppp #设置封装协议为PPP

```
Router1(config-if)#clock rate 64000 #设置clock rate（非必须）以同步通信速率，通过show controllers serial <interface>命令来确定当前接口类型，如果是DCE则需要设置。
```

```
Router1(config-if)#ip address 192.168.2.1 255.255.255.0
```

```
Router1(config-if)#no shutdown
```

```
%LINK-3-UPDOWN: Interface Ethernet0, changed state to up
```

```
%LINK-3-UPDOWN: Interface Ethernet0, changed state to down
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
```

```
Router1(config-if)#
```

按照类似的流程，设置剩下两个路由器。接下来设置**PC机**。



Press Enter to begin

```
C:> #PC1的命令行界面，PC2操作基本与之一致
```

```
C:>ipconfig /ip 192.168.16.2 255.255.255.0
```

```
C:>ipconfig /dg 192.168.16.1 # 设置默认网关，IP地址与直连路由器的端口IP地址相同，路由器接口IP作为设备的默认网关
```

```
C:>
```

## 4. 连通性测试

此时我们使用 **ping** 命令来测试网络的连通性。

```
Router1>ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router1>192.168.16.2
      ^
% Invalid input detected at '^' marker.

Router1>ping 192.168.16.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.16.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router1>ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router1>192.168.2.1
      ^
% Invalid input detected at '^' marker.

Router1>ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
Router1>
```

图4 路由器1的ping命令测试

```

Router2>ping 192.168.16.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.16.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
Router2>192.168.16.1
      ^
% Invalid input detected at '^' marker.

Router2>ping 192.168.16.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.16.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
Router2>ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router2>ping 192.168.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router2>ping 192.168.17.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.17.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
Router2>

```

图5 路由器2的ping命令测试

```

Press Enter to Start

00:00:06: %LINK-3-UPDOWN: Interface Serial1, changed state to up
00:00:06: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state to up
00:00:06: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
00:00:06: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to up
Router3>ping 192.168.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router3>ping 192.168.17.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.17.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router3>ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5), round-trip min/avg/max = 1/2/4 ms
Router3>

```

图6 路由器3的ping命令测试

Devices: PC1 [Device #4]

```
Use the command help to get started

Press Enter to begin
C:>ping 192.168.16.1

Pinging 192.168.16.1 with 32 bytes of data:
Reply from 192.168.16.1: bytes=32 time=63ms TTL=241
Reply from 192.168.16.1: bytes=32 time=61ms TTL=241
Reply from 192.168.16.1: bytes=32 time=55ms TTL=241
Reply from 192.168.16.1: bytes=32 time=65ms TTL=241
Reply from 192.168.16.1: bytes=32 time=61ms TTL=241

Ping statistics for 192.168.16.1:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 55ms, Maximum = 65ms, Average = 61ms

C:>ping 192.168.16.2

Pinging 192.168.16.2 with 32 bytes of data:
Reply from 192.168.16.2: bytes=32 time=62ms TTL=241
Reply from 192.168.16.2: bytes=32 time=69ms TTL=241
Reply from 192.168.16.2: bytes=32 time=49ms TTL=241
Reply from 192.168.16.2: bytes=32 time=65ms TTL=241
Reply from 192.168.16.2: bytes=32 time=65ms TTL=241

Ping statistics for 192.168.16.2:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 49ms, Maximum = 69ms, Average = 62ms

C:>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=52ms TTL=241
Reply from 192.168.1.1: bytes=32 time=70ms TTL=241
Reply from 192.168.1.1: bytes=32 time=58ms TTL=241
Reply from 192.168.1.1: bytes=32 time=69ms TTL=241
Reply from 192.168.1.1: bytes=32 time=50ms TTL=241

Ping statistics for 192.168.1.1:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 50ms, Maximum = 70ms, Average = 60ms

C:>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.

Ping statistics for 192.168.1.2:
    Packets: Sent = 5, Received = 0, Lost = 5 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

图7 PC机1的ping命令测试

```
Boson BOSS 5.0
Copyright 1998-2024 Boson Software, LLC.
Use the command help to get started

Press Enter to begin
C:>ping 192.168.16.1

Pinging 192.168.16.1 with 32 bytes of data:
Request timed out.

Ping statistics for 192.168.16.1:
    Packets: Sent = 5, Received = 0, Lost = 5 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time=64ms TTL=241
Reply from 192.168.2.2: bytes=32 time=55ms TTL=241
Reply from 192.168.2.2: bytes=32 time=64ms TTL=241
Reply from 192.168.2.2: bytes=32 time=57ms TTL=241
Reply from 192.168.2.2: bytes=32 time=59ms TTL=241

Ping statistics for 192.168.2.2:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 55ms, Maximum = 64ms, Average = 60ms

C:>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:
Request timed out.

Ping statistics for 192.168.2.1:
    Packets: Sent = 5, Received = 0, Lost = 5 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:>
```

图8 PC机2的ping命令测试

总结而言，所有路由器只能ping通与之相邻的路由器或PC机的对应通信端口IP，而不能ping通相邻路由器的另外一端口；而PC机只能ping通与之相邻的路由器的位处同一子网的端口。这是正常的现象，因为此时还没有配置**路由协议**，各个设备之内的路由表并没有到达不直接相连的设备的路由选择。总结一下，各路由器路由表与PC机的IP配置信息展示如下：

```
C:>ipconfig /all
```

HELP

Manipulates ip information for Workstation.

```
IPCONFIG [/all | /dg | /dns | /ip | /ipv6 | /release | /renew]
/all          Display full configuration information
/dg          Adds the default gateway to the workstation
/dns         Adds the dns server to the workstation
/ip          Adds the ip address and subnet mask to the workstation
/release     Release the IPV4 address to the workstation
/renew       Renew the IPV4 address to the workstation
/release6    Release the IPV6 address to the workstation
/renew6      Renew the IPV6 address to the workstation
```

Examples:

```
⌘ ipconfig /ip 157.1.1.12 255.0.0.0
⌘ ipconfig /dg 157.1.1.1
```

Boson BOSS 5.0 IP Configuration

Ethernet adapter Local Area Connection:

```
Connection-specific DNS Suffix . :
IP Address. . . . . : 192.168.16.2
Subnet Mask . . . . . : 255.255.255.0
```

```
C:>
```

Router1 ✘ Router2 ✘ Router3 ✘ PC1 ✘ PC2 ✘

```
C:>ipconfig
```

HELP

Manipulates ip information for Workstation.

```
IPCONFIG [/all | /dg | /dns | /ip | /ipv6 | /release | /renew]
/all          Display full configuration information
/dg          Adds the default gateway to the workstation
/dns         Adds the dns server to the workstation
/ip          Adds the ip address and subnet mask to the workstation
/release     Release the IPV4 address to the workstation
/renew       Renew the IPV4 address to the workstation
/release6    Release the IPV6 address to the workstation
/renew6      Renew the IPV6 address to the workstation
```

Examples:

```
⌘ ipconfig /ip 157.1.1.12 255.0.0.0
⌘ ipconfig /dg 157.1.1.1
```

Boson BOSS 5.0 IP Configuration

Ethernet adapter Local Area Connection:

```
Connection-specific DNS Suffix . :
IP Address. . . . . : 192.168.17.2
Subnet Mask . . . . . : 255.255.255.0
```

```

Router1>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       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, * - candidate default
       U - per-user static route

Gateway of last resort is not set

C     192.168.1.0 is directly connected, Serial0
C     192.168.16.0 is directly connected, Ethernet0

```

```

Router2>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       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, * - candidate default
       U - per-user static route

Gateway of last resort is not set

C     192.168.1.0 is directly connected, Serial0
C     192.168.2.0 is directly connected, Serial1

```

```

Router3>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       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, * - candidate default
       U - per-user static route

Gateway of last resort is not set

C     192.168.2.0 is directly connected, Serial1
C     192.168.17.0 is directly connected, Ethernet0

```

图9 路由表与IP配置情况一览

以下配置路由协议，之后再进行连通性测试。

## 5. 配置路由协议

如果要实现两台不直接连接的设备之间的通信，需要两台设备的路由表中存储着相应的路径。路由设置方法分为**静态路由**和**动态路由**。静态路由由网络管理员自行设置，一旦设置后不会自动更新，因此这样的网络工作时**资源占用较少**，但是**扩展性较差**，在网络互联设备状况发生变化的时候需要管理员手动设置新路由，灵活性差。而动态路由可以通过路由协议来**自适应**地更新网络路由，保证网络的连通性。

**路由协议**是位于**网络层**的网络协议，用于**路由器之间或路由器内部交换路由信息**，并根据这些信息动态计算数据包到达目标地址的最佳路径。路由协议的主要目的是实现网络设备之间的高效通信，自动适应网络拓扑的变化。这样的路由**拓展性更强**，**网络健壮性更佳**，但是在通信时相应的**资源消耗**也就更多。

**RIP (Routing Information Protocol)** 和 **OSPF (Open Shortest Path First)** 是两种常用的动态路由协议，前者基于**距离向量算法**的动态路由协议，以**跳数 (Hop Count)**作为路径的度量标准。每经过一个路由器，跳数 +1，在超过**15**时表示路径不可达。它的最大跳数限制了网络规模，且未考虑带宽问题，在实际应用中可能准确性并不佳；另外这样的算法可能会发生路由

环，需要借助“**毒性反转**”等技术解决。因此其更适合在**网络规模小，拓扑简单**的情况下使用，且不需要考虑**链路带宽**和**复杂路由选择**。而**OSPF (Open Shortest Path First)**是一种基于**链路状态算法 (Dijkstra算法)**的动态路由协议，专为中大型网络设计。它使用链路的“**成本 (Cost)**”作为路径的度量标准。默认情况下，成本与链路带宽成反比，带宽越高成本越低。OSPF支持将网络划分为多个**区域 (Area)**，骨干区域为**Area 0**。区域划分减少了路由信息的传播范围，提高了协议的效率。

由于本实验的网络拓扑结构非常简单，故使用**RIP**协议来配置路由。具体操作如下：



#在RIP中，用 `network` 命令声明路由器的接口所在的子网。RIP会自动将这些子网加入路由表，并通告给其他运行RIP协议的邻居路由器。声明的子网将被作为“可达网络”通告给邻居路由器，邻居根据这些信息更新自己的路由表。同时，路由器也会从邻居处学习其他子网的信息。

```
Router1(config)#router rip
Router1(config-router)#network 192.168.16.0
Router1(config-router)#network 192.168.1.0
Router1(config-router)#end
Router1#
...
Router2(config)#router rip
Router2(config-router)#network 192.168.1.0
Router2(config-router)#network 192.168.2.0
Router2(config-router)#end
Router2#
...
Router3(config)#router rip
Router3(config-router)#network 192.168.2.0
Router3(config-router)#network 192.168.17.0
Router3(config-router)#end
Router3#
```

此时查看各路由表，结果如下：

```
Router1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       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, * - candidate default
       U - per-user static route

Gateway of last resort is not set

C    192.168.1.0 is directly connected, Serial0
C    192.168.16.0 is directly connected, Ethernet0
R    192.168.2.0 [120/1] via 192.168.1.2, 00:07:40, Serial0
R    192.168.17.0 [120/2] via 192.168.1.2, 00:02:19, Serial0

Router1#
```

```

Router2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       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, * - candidate default
       U - per-user static route

Gateway of last resort is not set

C     192.168.1.0 is directly connected, Serial0
C     192.168.2.0 is directly connected, Serial1
R     192.168.16.0 [120/1] via 192.168.1.1, 00:06:15, Serial0
R     192.168.17.0 [120/1] via 192.168.2.2, 00:06:28, Serial1

Router2#

```

```

Router3(config)#router rip
Router3(config-router)#network 192.168.17.0
Router3(config-router)#network 192.168.2.0
Router3(config-router)#end
Router3#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       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, * - candidate default
       U - per-user static route

Gateway of last resort is not set

C     192.168.2.0 is directly connected, Serial1
C     192.168.17.0 is directly connected, Ethernet0
R     192.168.1.0 [120/1] via 192.168.2.1, 00:06:44, Serial1
R     192.168.16.0 [120/2] via 192.168.2.1, 00:04:21, Serial1

```

图10 配置RIP后路由表一览

明显看到，各个路由器通过RIP的学习，路由表中包含所有子网的IP地址了。现在使用ping命令测试一下。

## 5. 连通性再测试

使用ping命令，观察PC1能否与PC2成功连接：

```

C:>ping 192.168.17.2

Pinging 192.168.17.2 with 32 bytes of data:
Reply from 192.168.17.2: bytes=32 time=56ms TTL=241
Reply from 192.168.17.2: bytes=32 time=58ms TTL=241
Reply from 192.168.17.2: bytes=32 time=54ms TTL=241
Reply from 192.168.17.2: bytes=32 time=66ms TTL=241
Reply from 192.168.17.2: bytes=32 time=48ms TTL=241

Ping statistics for 192.168.17.2:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 48ms, Maximum = 66ms, Average = 56ms

C:>

```

Router1 × Router2 × Router3 × PC1 × PC2 ×

```
C:>ping 192.168.16.2

Pinging 192.168.16.2 with 32 bytes of data:
Reply from 192.168.16.2: bytes=32 time=53ms TTL=241
Reply from 192.168.16.2: bytes=32 time=71ms TTL=241
Reply from 192.168.16.2: bytes=32 time=62ms TTL=241
Reply from 192.168.16.2: bytes=32 time=63ms TTL=241
Reply from 192.168.16.2: bytes=32 time=48ms TTL=241

Ping statistics for 192.168.16.2:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 48ms, Maximum = 71ms, Average = 59ms

C:>
```

Router1 ✘ Router2 ✘ Router3 ✘ PC1 ✘ PC2 ✘

图11 PC机互ping结果

成功双向ping通。

## 五、收获与感想

已经到这里，不得不感慨一句我好辛苦……其实前一份报告对应的实验很简单，只不过我把它搞得复杂了，写了很多字。这个报告原本只想着粗制滥造，快速结束战斗，结果由于某种完美主义作祟，又写了不少。这个实验的最大困难在于学习软件操作本身。完成报告的五天时间里，有接近四天都是在琢磨为什么这个软件打不开了/卡死了/里面的路由设置都丢失了……如今离考试只有十多天了，不知道能复习成什么样子。不过还是很感谢这两次实验，在完成路由器和主机等设备的配置过程中，我学会了如何使用**IOS命令**来配置路由器、设置**静态路由**和**动态路由**（RIP）。我了解到静态路由和动态路由的不同之处，并且学会了在路由器上配置这两种路由的方法。在实验最后，我成功实现了两台主机相互ping通，证明了路由器配置的正确性和网络的连通性。这次实验让我对网络拓扑的设计、设备配置和路由协议有了更深入的理解。实验报告中包括了网络拓扑图、详细的配置步骤以及实验结果的截图，这些都是对我学习过程的很好总结，并且也为今后的网络配置和管理提供了宝贵的经验。剩下就是宝贵的**复习/预习**时间了。

另：不知因何缘故，明明已经配置完成并ping通，并请另两位同学测试了一下，均无问题。过一日再次打开配置文件发现竟无法ping通，不知道这背后是什么神奇的道理……只希望在老师的电脑里能够跑通吧！我的结果确实是来源于自己的配置文件，如果有问题的话，只能尝试在这里申辩了。