



实验报告

课程名称 计算机网络

开课学期 2019年 - 2020年第二学期

指导教师 刘雷

实验室 巡天楼 328

班 级 软件工程 4 班

学 号 2017070030429

姓 名 钟祯

成绩: _____ (五级)

实验课程 评分表标准

	全勤、学习态度端正、实验认真、积极回答问题、操作过程正确，结果准确，实验报告内容规范	偶有缺勤、实验认真、回答问题较积极、操作过程正确，结果准确，实验报告内容规范	旷课 2 次以内、偶有迟到、实验认真、回答问题较好、操作过程基本正确，结果基本准确，实验报告内容较规范	旷课 2 次以上、学习态度一般、基本能回答出问题、操作过程较正确，结果基本准确，实验报告内容基本规范	经常旷课，实验过程不认真、问题回答不积极、实验报告不符合要求或未交
	优秀 (90—100)	良好 (80—89)	中 (70—79)	及格 (60—69)	不及格 (<59)
实验一					
实验二					
实验三					
实验四					
实验五					
实验六					
实验七					
实验八					
实验成绩总评（五级制）					

说明： 1. 每次实验结束，学生完成一份实验报告，课程结束后汇总，加封面装订成册存档；2. 各系（部）可在以上五项栏目的基础上，可根据实验课程和实验项目的具体需要，统一设计和调整项目内容，但封面格式应统一；3. 对于设计性实验，只要求说明实验的目的要求、提出可供实验的基本条件和注意事项，实验方案和步骤的设置、仪器的安排等，可由学生自己设计；4. 可根据实验数量自行添加行数。打印到封面背面

桂林航天工业学院学生实验报告一

课程名称	计算机网络 (A)		实验项目名称	直通双绞线制作与测试	
开课教学单位及实验室	计算机科学与工程学院 巡天 309		实验日期	2020 年 06 月 12 日	
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	刘雷		实验成绩		

一、实验目的

- 1、了解双绞线的特性与应用场合，掌握双绞线的制作方法。
- 2、掌握使用双绞线作为传输介质的网络连接方法。
- 3、掌握测线仪的使用方法。

二、实验原理

网线水晶头有两种做法标准：标准分别为 TIA/EIA 568B 和 TIA/EIA 568A。制作水晶头首先将水晶头有卡的一面向下，有铜片的一面朝上，有开口的一方朝向自己身体，从左至右排序为 12345678，下面是 TIA/EIA 568B 和 TIA/EIA 568Av 网线线序（优先选择 568B 网线接法，通常认为该标准对电磁干扰的屏蔽更好）。

若双绞线两端水晶头采用相同标准（如两端都是 568B）则为直通线。两端不同标准（一端 568A，一端 568B）则为交叉线。

三、实验操作方法和步骤

- 步骤 1：准备好 5 类线、RJ-45 插头和一把专用的压线钳。
- 步骤 2：用压线钳的剥线刀口将 5 类线的外保护套管划开，刀口距端头至少 2 厘米。
- 步骤 3：将划开的外保护套管剥去（旋转、向外抽）。
- 步骤 4：露出 5 类线电缆中的 4 对双绞线。
- 步骤 5：按照 EIA/TIA-568B 标准和导线颜色将导线按规定的序号排好。
- 步骤 6：将 8 根导线平坦整齐地平行排列，导线间不留空隙
- 步骤 7：准备用压线钳的剪线刀口将 8 根导线剪断。
- 步骤 8：剪断电缆线。一定要剪得很整齐。剥开的导线长度不可太短。
- 步骤 9：将剪断的电缆线放入 RJ-45 插头，电缆线的外保护层最后应能够在 RJ-45 插头内的凹槽内。

陷处被压实。反复进行调整。

步骤 10：在确认一切都正确后（特别要注意不要将导线的顺序排列反了），将 RJ-45 插头放入压线钳的压头槽内，准备最后的压实。

步骤 11：双手紧握压线钳的手柄，用力压紧。

步骤 12：完成。用测试仪测试双绞线的通断性。

四、实验结果与分析

描述实验结果，并进行分析和总结。

实验结果：成功制作双绞线，并成功连接。

分析总结：本次实验使我接触到了手工制作网线这样更为具有实际操作性质的课程，这种实验要求我们用更多的精力去注意是否达到了实验要求，是否能正确制作出成功的接线口。在实验过程中，我们全程很认真，神情凝聚排线序、剪线、插线以及水晶头的压制。全心全意参与实验过程，我也是第一次接触到制作水晶头这样手动操作的实验，虽然实验要求和实验步骤并没有很复杂，但是更谨小慎微。对于此次实验我们受益良多，细心耐心上都得到很大锻炼与提高。这次实验不仅在于其具有很高实用价值，同时我们对计算机网络这门课程也有了新的收获，是一次难得的动手实践课程。

桂林航天工业学院学生实验报告二

课程名称	计算机网络 (A)		实验项目名称	CRC 循环冗余校验的计算与验证	
开课教学单位及实验室	计算机科学与工程学院		实验日期	2020 年 03 月 31 日	
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	刘雷		实验成绩		

一、实验目的

- 掌握 CRC 循环冗余校验方法。
- 熟练掌握交换机的基本配置。

二、实验原理

1、CRC 校验

用二进制的模 2 运算进行 2^n 乘 M 的运算，这相当于在 M 后面添加 n 个 0。

得到的 $(k + n)$ 位的数除以事先选定好的长度为 $(n + 1)$ 位的除数 P，得出商是 Q 而余数是 R，余数 R 比除数 P 少 1 位，即 R 是 n 位。

将余数 R 作为冗余码拼接在数据 M 后面发送出去。

2、交换机的基本配置

启动交换机后，使用 enable、configure terminal、interface (FastEthernet/Serial)x/x、no shutdown 等命令对其进行操作。

网络拓扑搭建完成之后，通过 ping 命令测试各主机之间的连通性。

三、实验操作方法和步骤

1、编程实现 CRC 循环冗余校验（编程语言不限）

- 编写程序，计算输入比特序列的 CRC 冗余码。
- 手动计算 CRC 冗余码。
- 两者进行对比，验证计算正确性。

2、交换机的基本配置（使用 packet tracer 或 gns3）

- (1) 交换机在各种模式下的切换，主要包括：用户模式、特权模式、全局配置模式、端口模式等。
- (2) 设计并搭建一个简单的网络拓扑结构，要求包含至少两个主机和一个交换机，确保各主机之间能相互通信（使用 ping 验证）。

四、实验结果与分析

1、对 CRC 循环冗余校验代码关键部分截图，进行简要说明。

```

1  function CRC(data) {
2    const x = 20170700304;
3    const y = 29;
4
5    let k = 0;
6    let crc = 0;
7    let length = 16;
8
9    while (length--) {
10      for (let i = 128; i != 0; i = i >> 1) {
11        crc = crc * 2;
12
13        if ((crc & x) != 0) crc = crc ^ y;
14
15        if ((data[k] & i) != 0) crc = crc ^ (x ^ y);
16      }
17
18      k++;
19    }
20
21    return crc.toString(16);
22  }
23
24  const data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
25
26  // 计算出结果
27  const result = CRC(data);
28  console.log("Result: " + result);
29
30  // push 进 data 数组
31  data.push(result);
32  console.log("Push data: ", data);
33
34  // 再次计算
35  console.log("Push result: " + CRC(data));
36

```

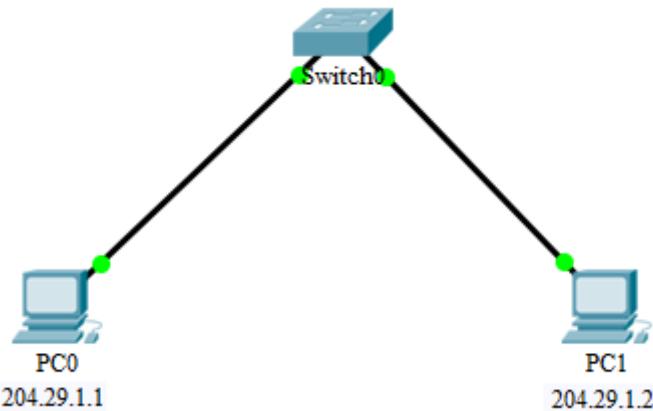
```

Administrator@zhenzhenChange MINGW64 ~/Desktop
$ node index.js
Result: b
Push data: [
  1, 2, 3, 4, 5,
  6, 7, 8, 9, 10,
  'b'
]
Push result: b

```

分析：通过计算，得出冗余校验码。因此将其追加回数组，则是发送端真正发送的数据。在接收端接收到数据时，根据相同的多项式做同样的计算，得到的校验码若为 0，则说明传输过程中比特没有产生差错，否则出错。

2、对网络拓扑结构和测试结果截图，进行简要说明。



```
Switch>
Switch>enable
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int f0/1
Switch(config-if)#no shut
Switch(config-if)#exit
Switch(config)#

```

```
C:\>ping 204.29.1.2
Pinging 204.29.1.2 with 32 bytes of data:

Reply from 204.29.1.2: bytes=32 time<1ms TTL=128
Reply from 204.29.1.2: bytes=32 time<1ms TTL=128
Reply from 204.29.1.2: bytes=32 time<1ms TTL=128
Reply from 204.29.1.2: bytes=32 time=1ms TTL=128

Ping statistics for 204.29.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

配置好主机的 IP 地址，通过交换机相连，两者之间能互通

桂林航天工业学院学生实验报告三

课程名称	计算机网络 (A)		实验项目名称	VLAN 的配置与管理	
开课教学单位及实验室	计算机科学与工程学院		实验日期	2020 年 04 月 14 日	
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	刘雷		实验成绩		

一、实验目的

- 1、熟练掌握交换机的基本配置。
- 2、熟悉交换机 VLAN 的相关概念。
- 3、掌握基于端口的 VLAN 的配置方法，能根据网络应用需求，将网络进行 VLAN 划分和配置

二、实验原理

利用以太网交换机可以很方便地实现虚拟局域网 VLAN (Virtual LAN)。

虚拟局域网 VLAN 是由一些局域网网段构成的与物理位置无关的逻辑组，而这些网段具有某些共同的需求。每一个 VLAN 的帧都有一个明确的标识符，指明发送这个帧的计算机是属于哪一个 VLAN。

虚拟局域网其实只是局域网给用户提供的一种服务，而并不是一种新型局域网。

由于虚拟局域网是用户和网络资源的逻辑组合，因此可按照需要将有关设备和资源非常方便地重新组合，使用户从不同的服务器或数据库中存取所需的资源。

配置过程中主要的命令：

```
vlan database  
vlan 101 name vlan101  
switchport mode trunk  
switchport mode access  
switchport access vlan 101
```

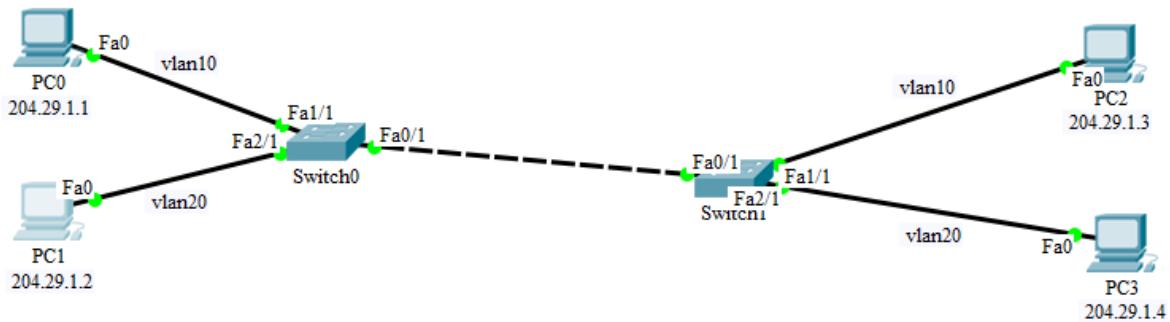
三、实验操作方法和步骤 (packet tracer 或 gns3)

- 1、搭建一个网络拓扑结构，至少包含 2 个交换机。
- 2、配置 PC 机的 IP 地址。

- 3、配置两台交换机，创建 VLAN10 和 VLAN20。
- 4、验证相同 VLAN 的连通性。
- 5、利用三层交换机实现不同 VLAN 之间的通信（扩展内容）。

四、实验结果与分析

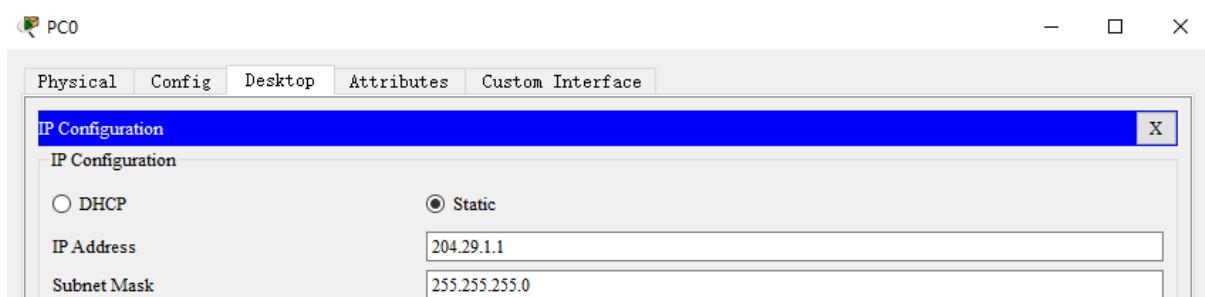
1、对网络拓扑结构进行截图，说明预期的 VLAN 划分方式。



将 Switch0 的 Fa1/1 接口放到 VLAN10，Fa2/1 接口放到 VLAN20

将 Switch1 的 Fa1/1 接口放到 VLAN10，Fa2/1 接口放到 VLAN20

2、说明 PC 机 IP 地址的配置方法和流程，进行相应截图。



单击 PC 机 -> 选择 Desktop -> 选择 IP Configuration -> 选择 Static -> 填写 IP Address 选项

3、说明交换机的配置方法和流程，进行相应截图

- (1) 创建 vlan 并命名
- (2) 选择 Fa 接口
- (3) 执行配置命令
- (4) 对于与另一个交换机相连的接口，需要配置为 trunk 模式
- (5) 保存退出

```
Switch(config)#vlan 10
Switch(config-vlan)#name VLAN10
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name VLAN20
Switch(config-vlan)#exit
Switch(config)#interface f1/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#interface f2/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface f0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#exit
Switch(config)#
```

```
Switch#show vlan
```

VLAN	Name	Status	Ports
1	default	active	Fa3/1, Fa4/1, Fa5/1
10	VLAN10	active	Fa1/1
20	VLAN20	active	Fa2/1
1002	fddi-default	active	
1003	token-ring-default	active	
1004	fddinet-default	active	
1005	trnet-default	active	

Switch0 与 Switch 1 配置相同

4、对验证结果截图，进行简要分析说明。

```
C:\>ping 204.29.1.3

Pinging 204.29.1.3 with 32 bytes of data:

Reply from 204.29.1.3: bytes=32 time<1ms TTL=128
Reply from 204.29.1.3: bytes=32 time=1ms TTL=128
Reply from 204.29.1.3: bytes=32 time<1ms TTL=128
Reply from 204.29.1.3: bytes=32 time=1ms TTL=128

Ping statistics for 204.29.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 204.29.1.2

Pinging 204.29.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 204.29.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

使用 PC0 ping PC1 (不通) 与 PC2 (相通)

结果表明，相同 VLAN 可以互通，不同 VLAN 不能通信

桂林航天工业学院学生实验报告四

课程名称	计算机网络 (A)		实验项目名称	静态路由的配置	
开课教学单位及实验室	计算机科学与工程学院		实验日期	2020 年 04 月 21 日	
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	刘雷		实验成绩		

一、实验目的

- 1、了解路由器的基本配置和相关的配置命令。
- 2、掌握配置静态路由的方法。
- 3、掌握路由器中提供的网络连通性测试命令.

二、实验原理

路由器分组转发算法

- (1) 从数据报的首部提取目的主机的 IP 地址 D, 得出目的网络地址为 N。
- (2) 若网络 N 与此路由器直接相连, 则把数据报直接交付目的主机 D; 否则是间接交付, 执行(3)。
- (3) 若路由表中有目的地址为 D 的特定主机路由, 则把数据报传送给路由表中所指明的下一跳路由器; 否则, 执行(4)。
- (4) 若路由表中有到达网络 N 的路由, 则把数据报传送给路由表指明的下一跳路由器; 否则, 执行(5)。
- (5) 若路由表中有一个默认路由, 则把数据报传送给路由表中所指明的默认路由器; 否则, 执行(6)。
- (6) 报告转发分组出错。

配置静态路由: Router(config)#ip route 192.168.3.0 255.255.255.0 192.168.2.2

配置端口地址: ip address 192.168.1.1 255.255.255.0

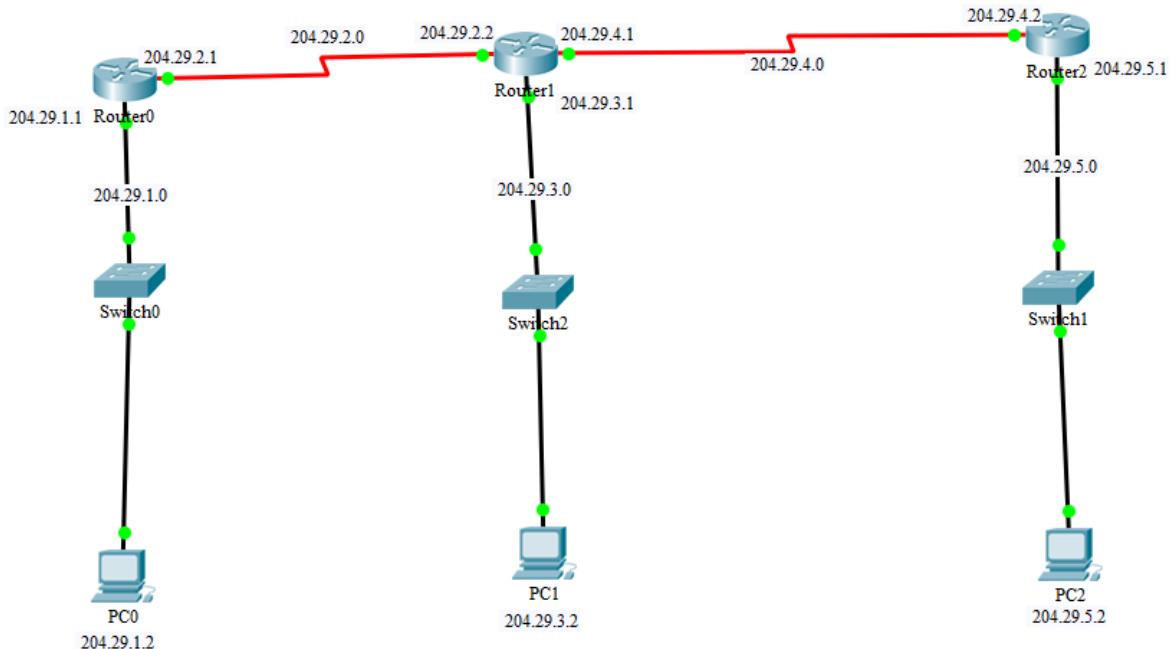
三、实验操作方法和步骤 (packet tracer 或 gns3)

- 1、搭建一个网络拓扑结构, 至少包含 3 个交换机。

- 2、练习路由器的基本配置命令。
- 3、练习在路由器的各种配置模式下的切换。
- 4、配置拓扑结构的静态路由，并查看路由状态。
- 5、删除已经配置的静态路由，观察结果。
- 6、配置汇总路由并测试连通性。

四、实验结果与分析

- 1、对网络拓扑结构进行截图，并做简单说明。



搭建拓扑图并分配 IP 地址

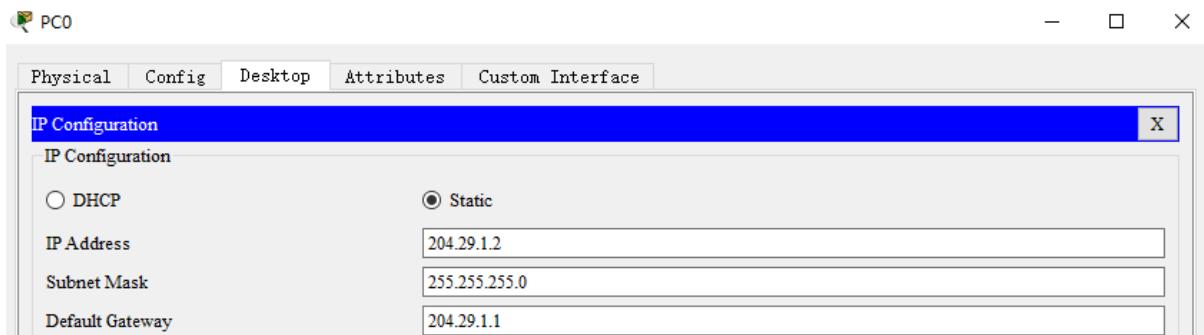
- 2、描述配置过程，说明关键步骤的配置方法和目的，并截图。
- (1) 确定 PC 以及 Router 的 IP 地址
 - (2) 执行配置路由器端口 IP 地址命令
 - (3) 配置 PC 的 IP 地址及默认网关
 - (4) 配置静态路由
 - (5) 测试连通性
 - (6) 清除静态路由再测试连通性

```

Router(config)#interface g0/0
Router(config-if)#ip address 204.29.1.1 255.255.255.0
Router(config-if)#exit
Router(config)#

```

每个路由器端口的配置都是类似的



PC 机 IP 地址及默认网关的配置

```

Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)#ip route 204.29.1.0 255.255.255.0 204.29.2.1
Router(config)#ip route 204.29.5.0 255.255.255.0 204.29.4.2
Router(config)#end
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

S    204.29.1.0/24 [1/0] via 204.29.2.1
      204.29.2.0/24 is variably subnetted, 2 subnets, 2 masks
C      204.29.2.0/24 is directly connected, Serial0/0/0
L      204.29.2.2/32 is directly connected, Serial0/0/0
      204.29.3.0/24 is variably subnetted, 2 subnets, 2 masks
C      204.29.3.0/24 is directly connected, GigabitEthernet0/0
L      204.29.3.1/32 is directly connected, GigabitEthernet0/0
      204.29.4.0/24 is variably subnetted, 2 subnets, 2 masks
C      204.29.4.0/24 is directly connected, Serial0/0/1
L      204.29.4.1/32 is directly connected, Serial0/0/1
S    204.29.5.0/24 [1/0] via 204.29.4.2

Router#

```

给 Router1 配置静态路由

```

Router(config)#ip route 0.0.0.0 0.0.0.0 204.29.2.2
Router(config)#end
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 204.29.2.2 to network 0.0.0.0

  204.29.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        204.29.1.0/24 is directly connected, GigabitEthernet0/0
L        204.29.1.1/32 is directly connected, GigabitEthernet0/0
  204.29.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        204.29.2.0/24 is directly connected, Serial0/0/1
L        204.29.2.1/32 is directly connected, Serial0/0/1
S*    0.0.0.0/0 [1/0] via 204.29.2.2

Router#

```

给 Router0 配置默认路由

```

Router(config)#ip route 0.0.0.0 0.0.0.0 204.29.4.1
Router(config)#end
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 204.29.4.1 to network 0.0.0.0

  204.29.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        204.29.4.0/24 is directly connected, Serial0/0/0
L        204.29.4.2/32 is directly connected, Serial0/0/0
  204.29.5.0/24 is variably subnetted, 2 subnets, 2 masks
C        204.29.5.0/24 is directly connected, GigabitEthernet0/0
L        204.29.5.1/32 is directly connected, GigabitEthernet0/0
S*    0.0.0.0/0 [1/0] via 204.29.4.1

Router#

```

给 Router2 配置默认路由

3、对连通性测试结果截图，进行简要分析说明。

```
C:\>ping 204.29.3.2
Pinging 204.29.3.2 with 32 bytes of data:
Reply from 204.29.3.2: bytes=32 time=7ms TTL=126
Reply from 204.29.3.2: bytes=32 time=1ms TTL=126
Reply from 204.29.3.2: bytes=32 time=1ms TTL=126
Reply from 204.29.3.2: bytes=32 time=2ms TTL=126

Ping statistics for 204.29.3.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 7ms, Average = 2ms

C:\>ping 204.29.5.2
Pinging 204.29.5.2 with 32 bytes of data:
Reply from 204.29.5.2: bytes=32 time=2ms TTL=125
Reply from 204.29.5.2: bytes=32 time=3ms TTL=125
Reply from 204.29.5.2: bytes=32 time=11ms TTL=125
Reply from 204.29.5.2: bytes=32 time=2ms TTL=125

Ping statistics for 204.29.5.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 11ms, Average = 4ms

C:\>
```

使用 PC0 ping PC1 与 PC2 皆能互通

```
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip route 204.29.5.0 255.255.255.0 204.29.4.2
Router(config)#no ip route 204.29.1.0 255.255.255.0 204.29.2.1
Router(config)#

```

清除 Router1 的静态路由

```
Router(config)#no ip route 0.0.0.0 0.0.0.0 204.29.2.2
```

```
Router(config)#no ip route 0.0.0.0 0.0.0.0 204.29.4.1
```

分别清除 Router0 与 Router2 的默认路由

```
C:\>ping 204.29.5.2
Pinging 204.29.5.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 204.29.5.2:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 204.29.3.2
Pinging 204.29.3.2 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 204.29.1.1: Destination host unreachable.
Reply from 204.29.1.1: Destination host unreachable.

Ping statistics for 204.29.3.2:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 204.29.1.1
Pinging 204.29.1.1 with 32 bytes of data:
Reply from 204.29.1.1: bytes=32 time<1ms TTL=255
Reply from 204.29.1.1: bytes=32 time<1ms TTL=255
Reply from 204.29.1.1: bytes=32 time<1ms TTL=255
Reply from 204.29.1.1: bytes=32 time=1ms TTL=255

Ping statistics for 204.29.1.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>
```

再次测试，此时发现只能 ping 通主机的默认网关，而其他主机之间无法连通

分析：由于没有配置静态路由和默认路由，一个网络里的主机只能 ping 通本网络内的主机和直接与本网络相连的路由器，无法 Ping 通其他网络的主机。

桂林航天工业学院学生实验报告五

课程名称	计算机网络 (A)		实验项目名称	RIP 协议的配置	
开课教学单位及实验室	计算机科学与工程学院		实验日期	2020 年 04 月 23 日	
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	刘雷		实验成绩		

一、实验目的

- 1、掌握在路由器上配置单区域 RIP 的方法。
- 2、分析 RIP 路由协议。
- 3、调试 RIP 路由协议。

二、实验原理

1、RIP 工作原理

路由信息协议 RIP (Routing Information Protocol) 是内部网关协议 IGP 中最先得到广泛使用的协议。RIP 是一种分布式的、基于距离向量的路由选择协议。RIP 协议要求网络中的每一个路由器都要维护从它自己到其他每一个目的网络的距离记录。

2、RIP 协议特点

- (1) 仅和相邻路由器交换信息。
- (2) 交换的信息是当前本路由器所知道的全部信息，即自己的路由表。
- (3) 按固定的时间间隔交换路由信息，例如，每隔 30 秒。当网络拓扑发生变化时，路由器也及时向相邻路由器通告拓扑变化后的路由信息。

3、路由表的建立

- (1) 路由器在刚刚开始工作时，只知道到直接连接的网络的距离（此距离定义为 1）。它的路由表是空的。
- (2) 以后，每一个路由器也只和数目非常有限的相邻路由器交换并更新路由信息。
- (3) 经过若干次更新后，所有的路由器最终都会知道到达本自治系统中任何一个网络的最短距离和下一跳路由器的地址。

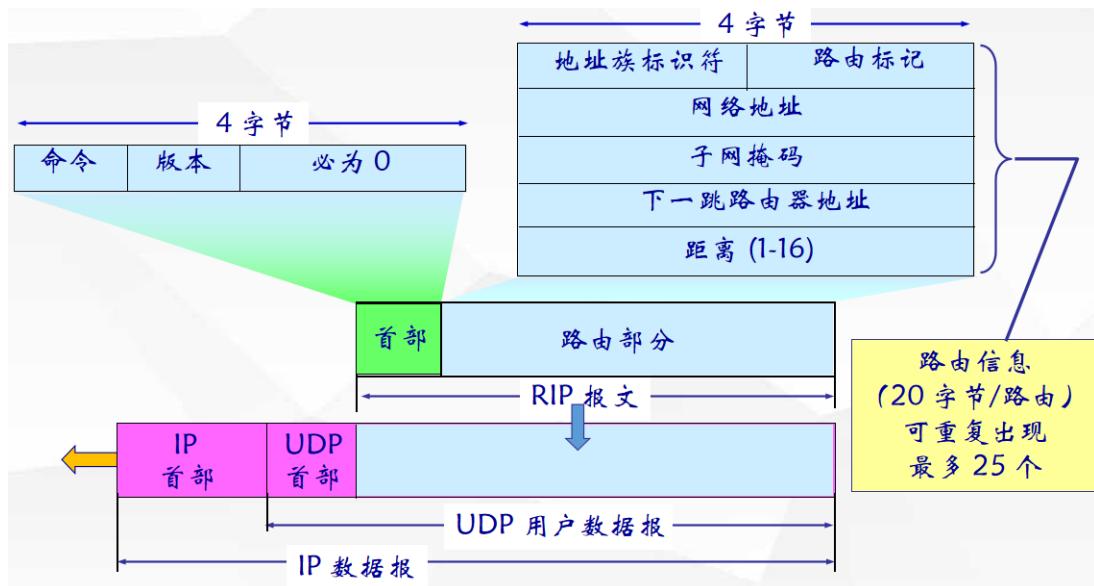
4、距离向量算法

距离向量算法的基础就是 Bellman-Ford 算法（或 Ford-Fulkerson 算法）。

这种算法的要点是这样的：

- (1) 设 X 是结点 A 到 B 的最短路径上的一个结点。
- (2) 若把路径 A→B 拆成两段路径 A→X 和 X→B，则每一段路径 A→X 和 X→B 也都分别是结点 A 到 X 和结点 X 到 B 的最短路径。

5、RIP2 协议的报文格式



6、关键命令

```
Router(config)#router rip  
Router(config-router)#network 192.168.1.1  
Router#show ip route  
Router#show ip rip database
```

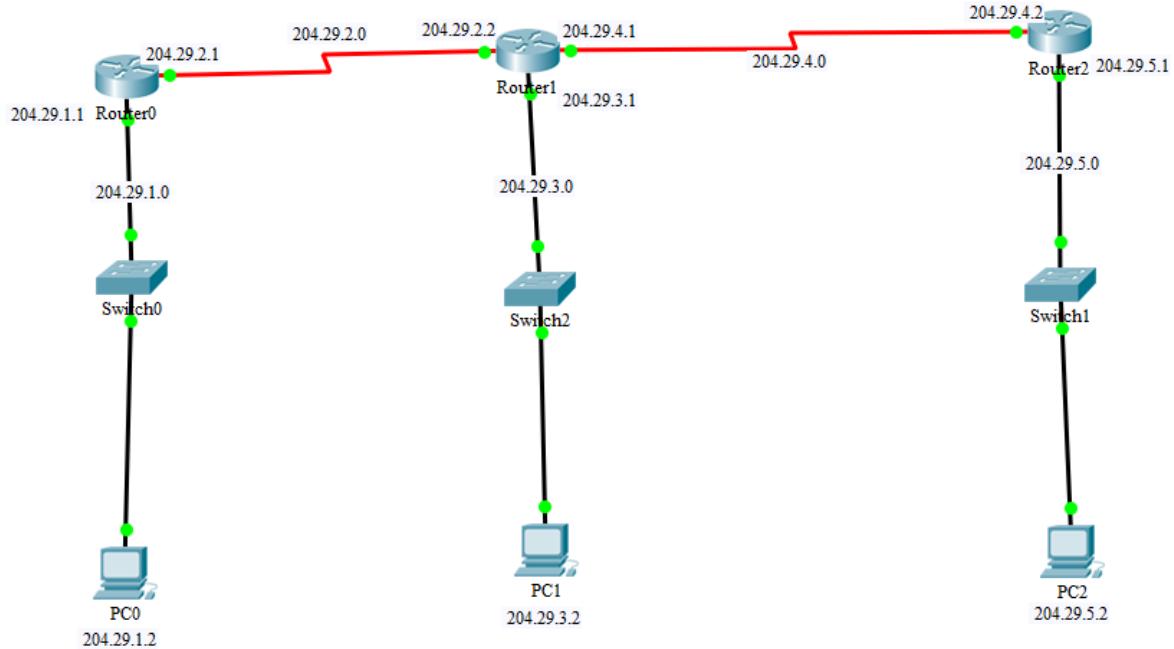
三、实验操作方法和步骤 (packet tracer 或 gns3)

- 1、设计实验的网络拓扑结构。
- 2、配置 PC 的 IP 地址。
- 3、在路由器上配置接口地址、时钟频率等基本信息。
- 4、创建 RIP 路由进程 route rip。
- 5、定义关联网络。
- 6、测试网络连通性。

7、调试 RIP: debug ip rip 并分析结果。

四、实验结果与分析

1、对网络拓扑结构进行截图，并做简单说明。



延用实验 4 网络拓扑图，清除其静态路由，恢复到主机、路由器端口 IP 地址配置完毕状态

2、描述路由器 RIP 配置过程，说明关键步骤的配置方法和目的，并截图。

- (1) 分别给路由器配置动态路由
- (2) 查看路由器相互学习（自主交换信息）而来的路由表
- (3) 测试连通性

```
Router(config)#router rip
Router(config-router)#network 204.29.2.0
Router(config-router)#network 204.29.3.0
Router(config-router)#network 204.29.4.0
Router(config-router)#end
```

给 Router1 配置 RIP

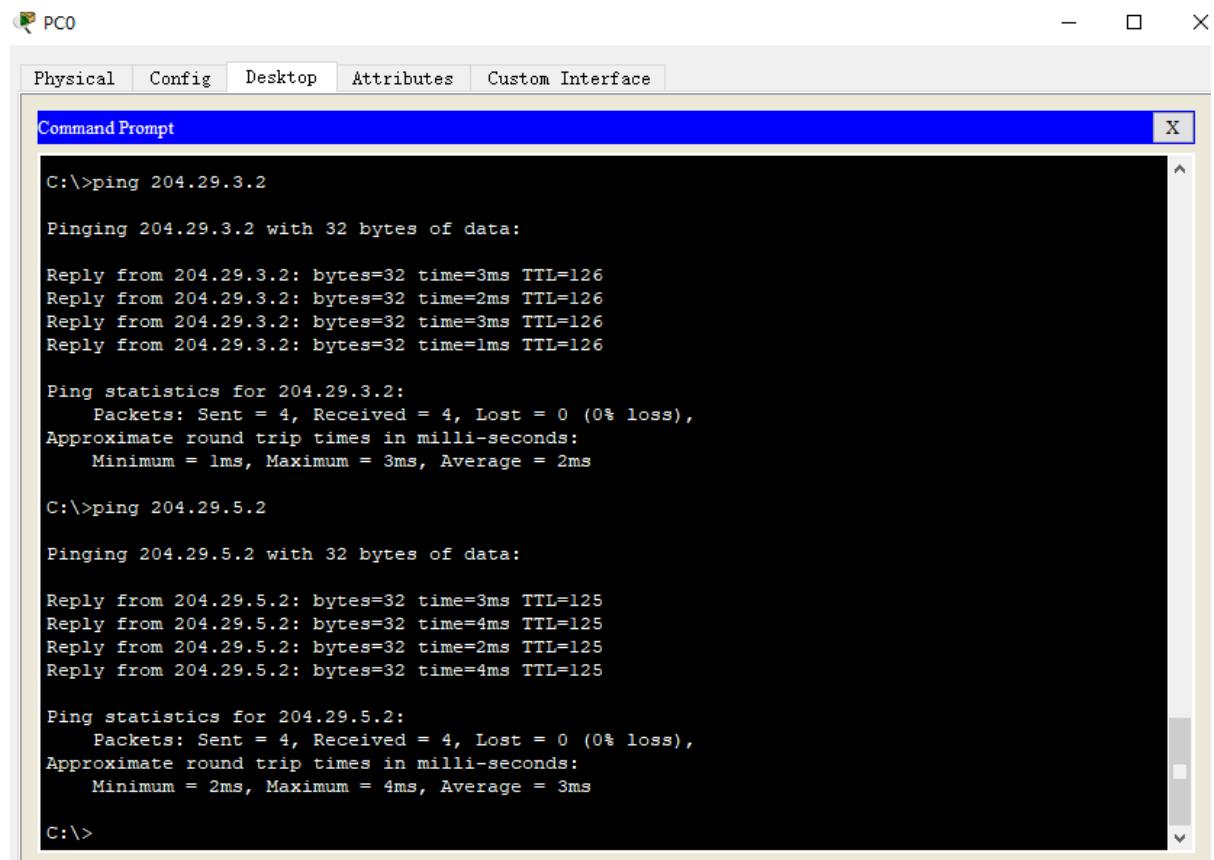
```
Router(config)#router rip  
Router(config-router)#network 204.29.1.0  
Router(config-router)#network 204.29.2.0  
Router(config-router)#end
```

给 Router0 配置 RIP

```
Router(config)#router rip  
Router(config-router)#network 204.29.4.0  
Router(config-router)#network 204.29.5.0  
Router(config-router)#end
```

给 Router2 配置 RIP

3、对连通性测试结果截图，进行简要分析说明。



The screenshot shows a Windows Command Prompt window titled "PC0". The window has tabs at the top: Physical, Config, Desktop, Attributes, and Custom Interface. The main area is a "Command Prompt" window with a blue title bar. It displays the following command and its output:

```
C:\>ping 204.29.3.2  
Pinging 204.29.3.2 with 32 bytes of data:  
  
Reply from 204.29.3.2: bytes=32 time=3ms TTL=126  
Reply from 204.29.3.2: bytes=32 time=2ms TTL=126  
Reply from 204.29.3.2: bytes=32 time=3ms TTL=126  
Reply from 204.29.3.2: bytes=32 time=1ms TTL=126  
  
Ping statistics for 204.29.3.2:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 1ms, Maximum = 3ms, Average = 2ms  
  
C:\>ping 204.29.5.2  
  
Pinging 204.29.5.2 with 32 bytes of data:  
  
Reply from 204.29.5.2: bytes=32 time=3ms TTL=125  
Reply from 204.29.5.2: bytes=32 time=4ms TTL=125  
Reply from 204.29.5.2: bytes=32 time=2ms TTL=125  
Reply from 204.29.5.2: bytes=32 time=4ms TTL=125  
  
Ping statistics for 204.29.5.2:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 2ms, Maximum = 4ms, Average = 3ms  
  
C:\>
```

使用 PC0 ping PC1 与 PC2，结果表明配置成功

4、对某一路由器的路由表截图，并进行简要分析说明。

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

R 204.29.1.0/24 [120/1] via 204.29.2.1, 00:00:14, Serial0/0/0
204.29.2.0/24 is variably subnetted, 2 subnets, 2 masks
C 204.29.2.0/24 is directly connected, Serial0/0/0
L 204.29.2.2/32 is directly connected, Serial0/0/0
204.29.3.0/24 is variably subnetted, 2 subnets, 2 masks
C 204.29.3.0/24 is directly connected, GigabitEthernet0/0
L 204.29.3.1/32 is directly connected, GigabitEthernet0/0
204.29.4.0/24 is variably subnetted, 2 subnets, 2 masks
C 204.29.4.0/24 is directly connected, Serial0/0/1
L 204.29.4.1/32 is directly connected, Serial0/0/1
R 204.29.5.0/24 [120/1] via 204.29.4.2, 00:00:11, Serial0/0/1

Router#show ip route rip ←
R 204.29.1.0/24 [120/1] via 204.29.2.1, 00:00:15, Serial0/0/0
204.29.4.0/24 is variably subnetted, 2 subnets, 2 masks
R 204.29.5.0/24 [120/1] via 204.29.4.2, 00:00:13, Serial0/0/1

Router#

Routing Table for Router1

Type	Network	Port	Next Hop IP	Metric
R	204.29.1.0/24	Serial0/0/0	204.29.2.1	120/1
C	204.29.2.0/24	Serial0/0/0	---	0/0
L	204.29.2.2/32	Serial0/0/0	---	0/0
C	204.29.3.0/24	GigabitEthernet0/0	---	0/0
L	204.29.3.1/32	GigabitEthernet0/0	---	0/0
C	204.29.4.0/24	Serial0/0/1	---	0/0
L	204.29.4.1/32	Serial0/0/1	---	0/0
R	204.29.5.0/24	Serial0/0/1	204.29.4.2	120/1

查看 Router1 的路由表

分析：路由器配置了动态路由，路由器与路由器之间就能相互学习（交换路由信息）。

5、对 RIP 协议报文截图，并进行简要分析说明。

```
Router#debug ip rip
RIP protocol debugging is on
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#RIP: sending v1 update to 255.255.255.255 via Serial0/0/0 (204.29.2.2)
RIP: build update entries
    network 204.29.3.0 metric 1
    network 204.29.4.0 metric 1
    network 204.29.5.0 metric 2
RIP: sending v1 update to 255.255.255.255 via GigabitEthernet0/0 (204.29.3.1)
RIP: build update entries
    network 204.29.1.0 metric 2
    network 204.29.2.0 metric 1
    network 204.29.4.0 metric 1
    network 204.29.5.0 metric 2
RIP: sending v1 update to 255.255.255.255 via Serial0/0/1 (204.29.4.1)
RIP: build update entries
    network 204.29.1.0 metric 2
    network 204.29.2.0 metric 1
    network 204.29.3.0 metric 1
RIP: received v1 update from 204.29.2.1 on Serial0/0/0
    204.29.1.0 in 1 hops
RIP: received v1 update from 204.29.4.2 on Serial0/0/1
    204.29.5.0 in 1 hops
RIP: sending v1 update to 255.255.255.255 via Serial0/0/0 (204.29.2.2)
RIP: build update entries
    network 204.29.3.0 metric 1
    network 204.29.4.0 metric 1
    network 204.29.5.0 metric 2
RIP: sending v1 update to 255.255.255.255 via GigabitEthernet0/0 (204.29.3.1)
```

桂林航天工业学院学生实验报告六

课程名称	计算机网络 (A)		实验项目名称	OSPF 协议的配置	
开课教学单位及实验室	计算机科学与工程学院 巡天 309		实验日期	2020 年 06 月 03 日	
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	刘雷		实验成绩		

一、实验目的

- 1、掌握在路由器上配置单区域 OSPF 的方法。
- 2、分析 OSPF 路由协议。
- 3、调试 OSPF 路由协议。

二、实验原理

1、OSPF 的三个要点

(1) 向本自治系统中所有路由器发送信息，这里使用的方法是洪泛法。

(2) 发送的信息就是与本路由器相邻的所有路由器的链路状态，但这只是路由器所知道的部分信息。

“链路状态”就是说明本路由器都和哪些路由器相邻，以及该链路的“度量”(metric)。

(3) 只有当链路状态发生变化时，路由器才用洪泛法向所有路由器发送此信息。

2、OSPF 的区域 (area)

为了使 OSPF 能够用于规模很大的网络，OSPF 将一个自治系统再划分为若干个更小的范围，叫作区域。

每一个区域都有一个 32 位的区域标识符（用点分十进制表示）。

区域也不能太大，在一个区域内的路由器最好不超过 200 个。

3、划分区域

划分区域的好处就是将利用洪泛法交换链路状态信息的范围局限于每一个区域而不是整个的自治系统，这就减少了整个网络上的通信量。

在一个区域内部的路由器只知道本区域的完整网络拓扑，而不知道其他区域的网络拓扑的情况。

OSPF 使用层次结构的区域划分。在上层的区域叫作主干区域 (backbone area)。

主干区域的标识符规定为 0.0.0.0。主干区域的作用是用来连通其他在下层的区域。

4、OSPF 直接用 IP 数据报传送

OSPF 不用 UDP 而是直接用 IP 数据报传送。

OSPF 构成的数据报很短。这样做可减少路由信息的通信量。

数据报很短的另一好处是可以不必将长的数据报分片传送。

但分片传送的数据报只要丢失一个，就无法组装成原来的数据报，而整个数据报就必须重传。

5、OSPF 的五种分组类型

类型 1，问候 (Hello) 分组。

类型 2，数据库描述 (Database Description) 分组。

类型 3，链路状态请求 (Link State Request) 分组。

类型 4，链路状态更新 (Link State Update) 分组，用洪泛法对全网更新链路状态。

类型 5，链路状态确认 (Link State Acknowledgment) 分组。

6、关键命令

```
Router(config)#router ospf 100
```

```
Router(config-router)#network 192.168.1.0 0.0.0.255 area 0
```

三、实验操作方法和步骤 (packet tracer 或 gns3)

1、设计实验的网络拓扑结构。

2、配置 PC 的 IP 地址。

3、在路由器上配置接口地址、时钟频率等基本信息。

4、配置 OSPF 路由协议。

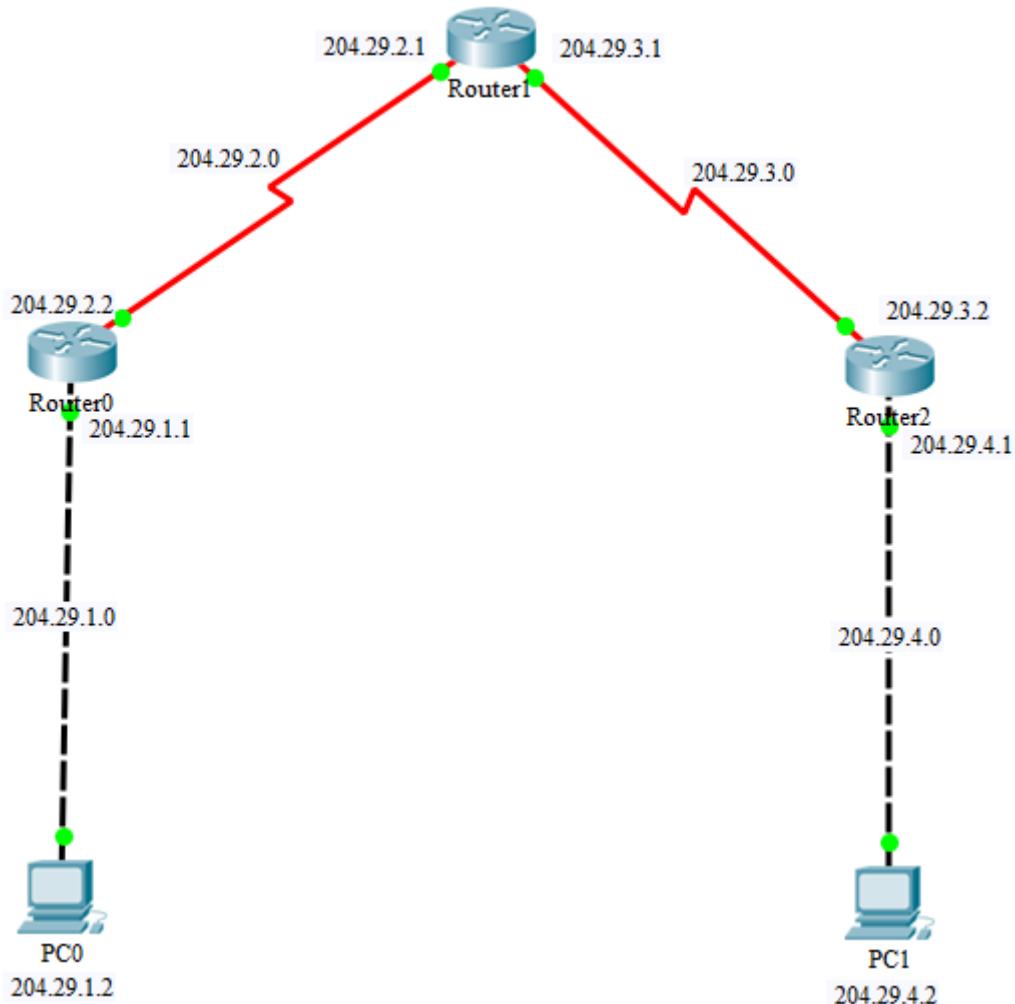
5、查看 OSPF 路由状态 show ip route。

6、查看 OSPF 邻居的基本信息 show ip ospf neighbor。

7、显示 OSPF 链路状态数据库的信息 show ip ospf database。

四、实验结果与分析

1、对网络拓扑结构进行截图，并做简单说明。



给 PC 机与路由器端口分配 IP 地址

2、描述路由器 OSPF 配置过程，说明关键步骤的配置方法和目的，并截图。

- (1) 分配 OSPF 协议进程
- (2) 宣告自身网段

```
Router(config)#router ospf 29
Router(config-router)#network 204.29.2.0 0.0.0.255 area 29
Router(config-router)#network 204.29.3.0 0.0.0.255 area 29
Router(config-router)#end
```

在 Router1 配置 OSPF

```
Router(config)#router ospf 29
Router(config-router)#network 204.29.1.0 0.0.0.255 area 29
Router(config-router)#network 204.29.2.0 0.0.0.255 area 29
Router(config-router)#end
```

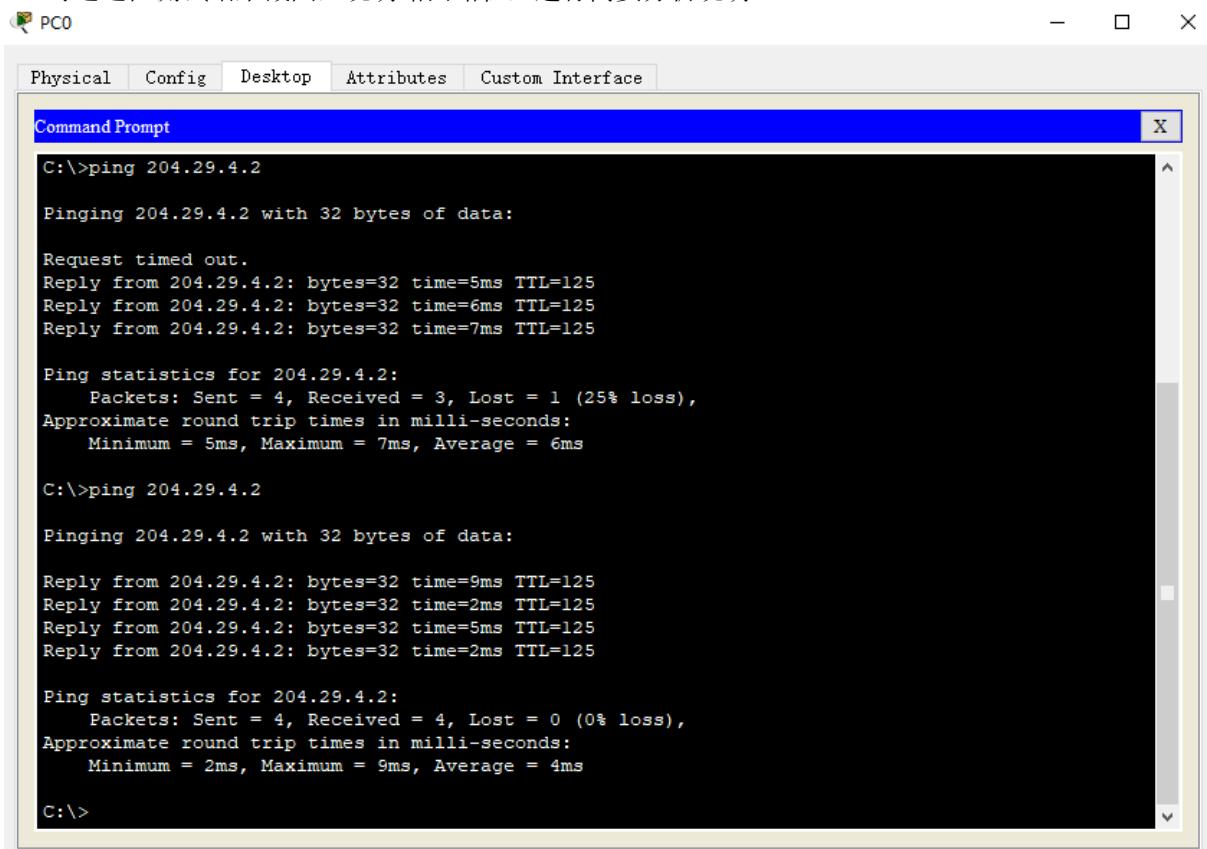
在 Router0 配置 OSPF

```
Router(config)#router ospf 29
Router(config-router)#network 204.29.3.0 0.0.0.255 area 29
Router(config-router)#network 204.29.4.0 0.0.0.255 area 29
Router(config-router)#

```

在 Router2 配置 OSPF

5、对连通性测试结果截图，说明路由路径，进行简要分析说明。



The screenshot shows a Windows Command Prompt window titled "PC0". The window has tabs at the top: Physical, Config, Desktop, Attributes, Custom Interface, and a selected tab labeled "Command Prompt". The main area displays the output of several "ping" commands. The first two pings are to IP address 204.29.4.2, resulting in a request timed out and three replies from the target host. The third and fourth pings are also to 204.29.4.2, showing successful round-trip times. The final ping command is to 204.29.4.2, showing a successful connection.

```
C:\>ping 204.29.4.2

Pinging 204.29.4.2 with 32 bytes of data:

Request timed out.
Reply from 204.29.4.2: bytes=32 time=5ms TTL=125
Reply from 204.29.4.2: bytes=32 time=6ms TTL=125
Reply from 204.29.4.2: bytes=32 time=7ms TTL=125

Ping statistics for 204.29.4.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 7ms, Average = 6ms

C:\>ping 204.29.4.2

Pinging 204.29.4.2 with 32 bytes of data:

Reply from 204.29.4.2: bytes=32 time=9ms TTL=125
Reply from 204.29.4.2: bytes=32 time=2ms TTL=125
Reply from 204.29.4.2: bytes=32 time=5ms TTL=125
Reply from 204.29.4.2: bytes=32 time=2ms TTL=125

Ping statistics for 204.29.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 9ms, Average = 4ms

C:\>
```

使用 PC0 ping PC1，结果表明已连通

6、对某一路由器的路由表截图，并结合拓扑结构进行简要分析说明。

```
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

O  204.29.1.0/24 [110/65] via 204.29.2.2, 00:03:30, Serial0/0/0
  204.29.2.0/24 is variably subnetted, 2 subnets, 2 masks
C  204.29.2.0/24 is directly connected, Serial0/0/0
L  204.29.2.1/32 is directly connected, Serial0/0/0
  204.29.3.0/24 is variably subnetted, 2 subnets, 2 masks
C  204.29.3.0/24 is directly connected, Serial0/0/1
L  204.29.3.1/32 is directly connected, Serial0/0/1
O  204.29.4.0/24 [110/65] via 204.29.3.2, 00:02:22, Serial0/0/1

Router#show ip route ospf
O  204.29.1.0 [110/65] via 204.29.2.2, 00:03:37, Serial0/0/0
O  204.29.4.0 [110/65] via 204.29.3.2, 00:02:29, Serial0/0/1

Router#
```

Routing Table for Router1

Type	Network	Port	Next Hop IP	Metric
O	204.29.1.0/24	Serial0/0/0	204.29.2.2	110/65
C	204.29.2.0/24	Serial0/0/0	---	0/0
L	204.29.2.1/32	Serial0/0/0	---	0/0
C	204.29.3.0/24	Serial0/0/1	---	0/0
L	204.29.3.1/32	Serial0/0/1	---	0/0
O	204.29.4.0/24	Serial0/0/1	204.29.3.2	110/65

Router1 路由表

分析：在应用了 OSPF 协议后，在 OSPF 中划分区域的目的就是在于控制链路状态信息 LSA 泛洪的范围、减小链路状态数据库 LSDB 的大小、改善网络的可扩展性、达到快速收敛，在大型网络下，这种优势尤为明显。

5、查看 OSPF 邻居的基本信息，并结合拓扑结构进行说明。

```
Router#show ip ospf neighbor

Neighbor ID      Pri   State        Dead Time     Address          Interface
204.29.4.1        0    FULL/       00:00:37     204.29.3.2      Serial0/0/1
204.29.2.2        0    FULL/       00:00:36     204.29.2.2      Serial0/0/0
Router#show ip ospf database
              OSPF Router with ID (204.29.3.1) (Process ID 29)

              Router Link States (Area 29)

Link ID          ADV Router      Age      Seq#      Checksum Link count
204.29.2.2      204.29.2.2    464      0x80000003 0x0046de 3
204.29.3.1      204.29.3.1    418      0x80000004 0x008137 4
204.29.4.1      204.29.4.1    396      0x80000003 0x008a93 3
```

Router 1 OSPF 的邻居及数据库信息

分析说明：根据信息，Router1 与 Router2 成为了邻居，在 PC0 ping PC1 的过程中，依次经过 Router0 → Router1 → Router2，网关依次经过 204.29.2.2 → 204.29.3.1 → 204.29.4.1

桂林航天工业学院学生实验报告七

课程名称	计算机网络 (A)		实验项目名称	Wireshark 捕捉数据包并分析	
开课教学单位及实验室	计算机科学与工程学院 巡天 328		实验日期	2020 年 06 月 09 日	
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	刘雷		实验成绩		

一、实验目的

- 1、掌握以太网帧的构成，了解各个字段的含义。
- 2、理解 ping 命令的实现原理
- 3、熟悉 TCP 的首部格式，理解建立连接和释放连接的过程。
- 4、掌握网络协议分析软件的基本使用方法。

二、实验原理

1、以太网的帧包括目的地址（6 字节）、原地址（6 字节）、类型（2 字节）、数据（46-1500 字节）、FCS（4 字节）。

2、PING (Packet InterNet Groper)

PING 用来测试两个主机之间的连通性。

PING 使用了 ICMP 回送请求与回送回答报文。

PING 是应用层直接使用网络层 ICMP 的例子，它没有通过运输层的 TCP 或 UDP。

2、TCP 连接的建立

(1) 客户端发送 syn 包 ($seq=j$) 到服务器，并进入 SYN_SENT 状态，等待服务器确认；SYN：同步序列编号。

(2) 服务器收到 syn 包，必须确认客户的 SYN ($ack=j+1$)，同时自己也发送一个 SYN 包 ($seq=k$)，即 SYN+ACK 包，此时服务器进入 SYN_RECV 状态。

(3) 客户端收到服务器的 SYN+ACK 包，向服务器发送确认包 ACK ($ack=k+1$)，此包发送完毕，客户端和服务端进入 ESTABLISHED (TCP 连接成功) 状态，完成三次握手。

3、TCP 连接的释放

(1) 当主机 A 的应用程序通知 TCP 数据已经发送完毕时，TCP 向主机 B 发送一个带有 FIN 附加标

记的报文段。

- (2) 主机 B 收到这个 FIN 报文段之后，并不立即用 FIN 报文段回复主机 A，而是先向主机 A 发送一个确认序号 ACK，同时通知自己相应的应用程序：对方要求关闭连接。
- (3) 主机 B 的应用程序告诉 TCP：我要彻底的关闭连接，TCP 向主机 A 送一个 FIN 报文段。
- (4) 主机 A 收到这个 FIN 报文段后，向主机 B 发送一个 ACK 表示连接彻底释放。

三、实验操作方法和步骤 (Wireshark)

- 1、运行网络协议分析软件 (Wireshark)。
- 2、列表框中列出了本地计算机中存在的网络适配器，选择网络适配器，点击 Start 便可开始抓包。
- 3、使用 ping 命令测试本机与网关的连通性。
- 4、观察协议树区中以太网帧结构，是否符合 EthernetII 帧结构。
- 5、重新开始网络数据包捕获，浏览一些网页之后停止捕获，查看捕获到的数据包，记录以太网数据帧的类型字段的值是什么？对应什么协议。
- 6、分析 TCP 连接的建立和释放过程。

四、实验结果与分析

- 1、结合抓包结果，分析以太网的帧结构。

The screenshot shows two windows. The top window is Wireshark displaying network traffic. A red circle highlights the source and destination IP addresses in the first few rows of the packet list. The bottom window is a terminal window titled 'Administrator@zhenzhenChange MINGW64 ~' showing the command '\$ ping 47.93.41.216' and its output. The output shows four ICMP echo replies from the target host.

No.	Time	Source	Destination	Protocol	Length	Info
535	7.497492	172.16.192.157	47.93.41.216	ICMP	74	Echo (ping) request id=0x0001, seq=8662/54817, ttl=128 (reply in 536)
598	8.499313	172.16.192.157	47.93.41.216	ICMP	74	Echo (ping) request id=0x0001, seq=8663/55073, ttl=128 (reply in 592)
607	8.770126	172.16.192.157	47.93.41.216	TCP	54	65141 → 80 [ACK] Seq=1 Ack=2 Win=1023 Len=0
611	8.798673	172.16.192.157	47.93.41.216	TCP	54	65145 → 80 [ACK] Seq=1 Ack=2 Win=1029 Len=0
616	8.821062	172.16.192.157	47.93.41.216	TCP	54	65142 → 80 [ACK] Seq=1 Ack=2 Win=1024 Len=0
687	9.503840	172.16.192.157	47.93.41.216	TCP	54	65146 → 80 [ACK] Seq=1 Ack=2 Win=1029 Len=0
689	9.508498	172.16.192.157	47.93.41.216	TCP	54	65144 → 80 [ACK] Seq=1 Ack=2 Win=8244 Len=0
741	10.207598	172.16.192.157	47.93.41.216	TCP	54	65143 → 80 [ACK] Seq=1 Ack=2 Win=1029 Len=0
825	11.951361	172.16.192.157	47.93.41.216	TCP	54	65145 → 80 [FIN, ACK] Seq=1 Ack=2 Win=1029 Len=0
826	11.951418	172.16.192.157	47.93.41.216	TCP	54	65141 → 80 [FIN, ACK] Seq=1 Ack=2 Win=1023 Len=0
827	11.951455	172.16.192.157	47.93.41.216	TCP	54	65142 → 80 [FIN, ACK] Seq=1 Ack=2 Win=1024 Len=0
828	11.951488	172.16.192.157	47.93.41.216	TCP	54	65144 → 80 [FIN, ACK] Seq=1 Ack=2 Win=8244 Len=0
829	11.951524	172.16.192.157	47.93.41.216	TCP	54	65146 → 80 [FIN, ACK] Seq=1 Ack=2 Win=1029 Len=0
830	11.951557	172.16.192.157	47.93.41.216	TCP	54	65143 → 80 [FIN, ACK] Seq=1 Ack=2 Win=1029 Len=0

```
Administrator@zhenzhenChange MINGW64 ~
$ ping 47.93.41.216

正在 Ping 47.93.41.216 具有 32 字节的数据:
来自 47.93.41.216 的回复: 字节=32 时间=54ms TTL=52
来自 47.93.41.216 的回复: 字节=32 时间=55ms TTL=52
来自 47.93.41.216 的回复: 字节=32 时间=54ms TTL=52
来自 47.93.41.216 的回复: 字节=32 时间=55ms TTL=52

47.93.41.216 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 = 54ms, 最长 = 55ms, 平均 = 54ms
```

```
▼ Ethernet II, Src: ASUSTekC_5e:0f:2a (04:92:26:5e:0f:2a), Dst: NewH3CTe_5a:f4:01 (78:2c:29:5a:f4:01)
  > Destination: NewH3CTe_5a:f4:01 (78:2c:29:5a:f4:01)
  > Source: ASUSTekC_5e:0f:2a (04:92:26:5e:0f:2a)
  Type: IPv4 (0x0800)
```

Ethernet II 帧类型值大于等于 1536 (0x0600)，以太网数据帧的长度在 64–1518 字节之间

2、结合抓包结果和 ICMP 协议，分析 ping 命令的实现原理。

ping 命令是基于 ICMP 协议来工作的，ICMP 全称为 Internet 控制报文协议（Internet Control Message Protocol）。ping 命令会发送一份 ICMP 回显请求报文给目标主机，并等待目标主机返回 ICMP 回显应答。因为 ICMP 协议会要求目标主机在收到消息之后，必须返回 ICMP 应答消息给源主机，如果源主机在一定时间内收到了目标主机的应答，则表明两台主机之间网络是可达的。

3、结合抓包结果和 TCP 协议，分析连接建立和释放的过程。

连接建立过程

1. 客户端发送 SYN 请求，进入 SYN-SEND（同步发送状态）
2. 服务端收到请求，回复 ACK 报文并发送 SYN 请求，进入 SYN-RECV（同步接收状态）
3. 客户端收到回复和 SYN 请求，然后向服务端发送 ACK 报文，接着进入 ESTABLISH（连接建立状态），服务端收到客户端的 ACK 后也进入了 ESTABLISH（连接建立状态）
4. 至此，两端的通信连接建立完毕。

释放过程

1. TCP 客户端发送一个 FIN，关闭客户端到服务器端的数据传送。（客户端不再发送报文给服务器端，但可接受服务器端报文）
2. 服务器收到这个 FIN，它发回一个 ACK，确认序号为收到的序号加 1
3. 服务器关闭客户端的连接，发送一个 FIN 给客户端。（服务器端关闭到客户端的数据传送）
4. 客户端发回 ACK 报文确认，并将确认序号设置为收到序号加 1

桂林航天工业学院学生实验报告八

课程名称	计算机网络 (A)		实验项目名称	FTP 服务器配置	
开课教学单位及实验室	计算机科学与工程学院 巡天 328		实验日期	2020 年 06 月 16 日	
学生姓名	钟祯	学号	2017070030429	专业班级	17 软件工程 4 班
指导教师	刘雷		实验成绩		

一、实验目的

- 1、掌握常用 FTP 服务器软件的配置与管理。
- 2、掌握常用 FTP 客户端工具的使用。
- 3、更深入地了解 FTP 协议。

二、实验原理

1、FTP 的主要工作步骤

- (1) 打开熟知端口（端口号为 21），使客户进程能够连接上。
- (2) 等待客户进程发出连接请求。
- (3) 启动从属进程来处理客户进程发来的请求。从属进程对客户进程的请求处理完毕后即终止，但从属进程在运行期间根据需要还可能创建其他一些子进程。
- (4) 回到等待状态，继续接受其他客户进程发来的请求。主进程与从属进程的处理是并发地进行。

2、FTP 的两个连接

控制连接在整个会话期间一直保持打开，FTP 客户发出的传送请求通过控制连接发送给服务器端的控制进程，但控制连接不用来传送文件。

实际用于传输文件的是“数据连接”。服务器端的控制进程在接收到 FTP 客户发送来的文件传输请求后就创建“数据传送进程”和“数据连接”，用来连接客户端和服务器端的数据传送进程。

数据传送进程实际完成文件的传送，在传送完毕后关闭“数据传送连接”并结束运行。

3、两个不同的端口号

当客户进程向服务器进程发出建立连接请求时，要寻找连接服务器进程的熟知端口（21），同时还要告诉服务器进程自己的另一个端口号，用于建立数据传送连接。

接着，服务器进程用自己传送数据的熟知端口（20）与客户进程所提供的端口号建立数据传送

连接。

由于 FTP 使用了两个不同的端口号，所以数据连接与控制连接不会发生混乱。

4、主动模式与被动模式

FTP 支持两种模式，一种方式叫做 Standard (也就是 PORT 方式，主动方式)，一种是 Passive(也就是 PASV，被动方式)。主动模式 FTP 的客户端发送 PORT 命令到 FTP 服务器。被动模式 FTP 的客户端发送 PASV 命令到 FTP Server。

FTP 客户端首先和 FTP 服务器的 TCP 21 端口建立连接，通过这个通道发送命令，客户端需要接收数据的时候在这个通道上发送 PORT 命令。PORT 命令包含了客户端用什么端口接收数据。在传送数据的时候，服务器端通过自己的 TCP 20 端口连接至客户端的指定端口发送数据。FTP server 必须和客户端建立一个新的连接用来传送数据。

被动模式在建立控制通道的时候和主动模式类似，但建立连接后发送的不是 Port 命令，而是 Pasv 命令。FTP 服务器收到 Pasv 命令后，随机打开一个高端端口（端口号大于 1024）并且通知客户端在这个端口上传送数据的请求，客户端连接 FTP 服务器此端口，通过三次握手建立通道，然后 FTP 服务器将通过这个端口进行数据的传送。

三、实验操作方法和步骤 (Wireshark、FileZilla)

1、打开 FileZilla 服务端，建立站点

- (1) 点击新站点。
- (2) 在新站点内输入服务器/虚拟主机提供的 IP 地址、用户名和密码，然后点击“连接”按钮。
- (3) 设置共享文件夹。

2、打开 Wireshark 软件进行抓包。

3、使用 FileZilla 客户端，用主动模式下载文件。

4、使用 FileZilla 客户端，用被动模式下载文件。

四、实验结果与分析

The screenshot shows a browser window with the URL <ftp://172.16.192.157>. The page title is '/ 的索引'. Below the title is a table listing files:

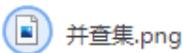
	名称	大小	修改日期
□	hinesboy-mavonEditor- mavonEditor - A markdown editor based on Vue that supports a variety of personalized features.url	65 B	2020/6/24 上午7:07:00
□	VisuAlgo - 数据结构和算法动态可视化 (Chinese).url	49 B	2020/6/9 上午7:05:00
□	Vue integration - Docs - TinyMCE.url	110 B	2020/6/24 上午7:14:00
□	使用vue实现排序算法演示动画 - 挖金.url	73 B	2020/5/23 上午8:00:00
□	图片.jpg	163 kB	2020/5/13 上午8:00:00
□	并查集.png	39.0 kB	2020/6/9 上午6:11:00
□	快速上手 - Ant Design Vue.url	76 B	2020/6/23 上午4:51:00
□	算法导论 第13章 红黑树 (图文详细解说) _数据结构与算法_程序猿-CSDN博客.url	83 B	2020/5/21 上午8:00:00

1、结合抓包结果，分析主动模式下的 FTP 工作流程。

The screenshot shows the FileZilla interface with an active session labeled 'ftp'. The log pane displays the following traffic capture:

No.	Time	Source	Destination	Protocol	Length	Info
14	3.155951	172.16.192.157	172.16.192.157	FTP	187	Response: 220-FileZilla Server 0.9.60 beta
22	3.156125	172.16.192.157	172.16.192.157	FTP	60	Request: USER anonymous
28	3.156267	172.16.192.157	172.16.192.157	FTP	81	Response: 331 Password required for anonymous
35	3.156384	172.16.192.157	172.16.192.157	FTP	69	Request: PASS chrome@example.com
40	3.156485	172.16.192.157	172.16.192.157	FTP	78	Response: 530 Login or password incorrect!
45	3.156555	172.16.192.157	172.16.192.157	FTP	50	Request: QUIT
50	3.156639	172.16.192.157	172.16.192.157	FTP	57	Response: 221 Goodbye
115	3.158138	172.16.192.157	172.16.192.157	FTP	187	Response: 220-FileZilla Server 0.9.60 beta
120	3.158301	172.16.192.157	172.16.192.157	FTP	60	Request: USER zhongzhen
127	3.158453	172.16.192.157	172.16.192.157	FTP	81	Response: 331 Password required for zhongzhen
134	3.158554	172.16.192.157	172.16.192.157	FTP	60	Request: PASS zhongzhen
137	3.158746	172.16.192.157	172.16.192.157	FTP	59	Response: 230 Logged on
142	3.158867	172.16.192.157	172.16.192.157	FTP	50	Request: SYST
147	3.158958	172.16.192.157	172.16.192.157	FTP	76	Response: 215 UNIX emulated by FileZilla
153	3.159021	172.16.192.157	172.16.192.157	FTP	49	Request: PWD
158	3.159096	172.16.192.157	172.16.192.157	FTP	75	Response: 257 "/" is current directory.
164	3.159159	172.16.192.157	172.16.192.157	FTP	52	Request: TYPE I
170	3.159240	172.16.192.157	172.16.192.157	FTP	63	Response: 200 Type set to I
176	3.159299	172.16.192.157	172.16.192.157	FTP	62	Request: SIZE /图片.jpg
187	3.159427	172.16.192.157	172.16.192.157	FTP	56	Response: 213 166558
194	3.159495	172.16.192.157	172.16.192.157	FTP	61	Request: CWD /图片.jpg
203	3.159588	172.16.192.157	172.16.192.157	FTP	97	Response: 550 CWD failed. "/图片.jpg": directory not found.
206	3.159647	172.16.192.157	172.16.192.157	FTP	50	Request: PASV
221	3.159922	172.16.192.157	172.16.192.157	FTP	96	Response: 227 Entering Passive Mode (172,16,192,157,218,255)
250	3.160500	172.16.192.157	172.16.192.157	FTP	62	Request: RETR /图片.jpg
264	3.160750	172.16.192.157	172.16.192.157	FTP	117	Response: 150 Opening data channel for file download from server of "/图片.jpg"
318	3.161964	172.16.192.157	172.16.192.157	FTP	88	Response: 226 Successfully transferred "/图片.jpg"

ftp://172.16.192.157/并查集.png

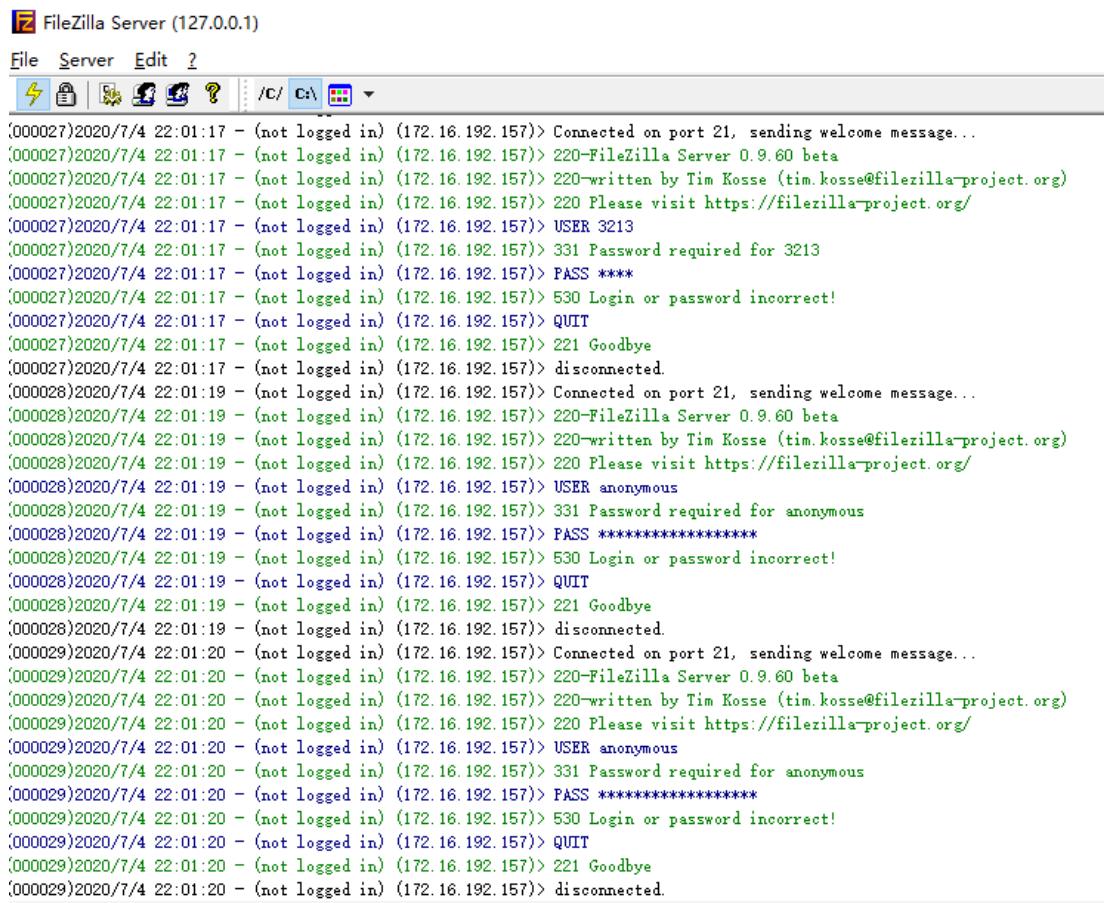


1. 客户端的命令端口与 FTP 服务器的命令端口建立连接，并发送命令“PORT 端口”。
2. FTP 服务器给客户端的命令端口返回一个“ACK”。
3. FTP 服务器发起一个从它自己的数据端口到客户端先前指定的数据端口的连接
4. 最后客户端给服务器端返回一个“ACK”。

2、结合抓包结果，分析被动模式下的FTP工作流程。

```
50 Request: PASV
94 Response: 227 Entering Passive Mode (172,16,192,157,206,4)
53 Request: LIST -l
99 Response: 150 Opening data channel for directory listing of "/"
78 Response: 226 Successfully transferred "/"
50 Request: QUIT
57 Response: 221 Goodbye
```

1. 客户端的命令端口与服务器的命令端口建立连接，并发送命令“PASV”
2. 服务器返回命令“PORT 端口”，告诉客户端（服务器）用哪个端口侦听数据连接
3. 客户端初始化一个从自己的数据端口到服务器端指定的数据端口的数据连接
4. 给客户端的数据端口返回一个“ACK”响应



The screenshot shows the FileZilla Server interface with a log window displaying multiple connection attempts. The log entries are color-coded in green, indicating successful connections and responses. The log entries show various commands such as PASV, PORT, LIST, QUIT, and responses like 227, 150, 226, 221, and 530. The log also includes welcome messages from the server and information about the FileZilla Server version.

```
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> Connected on port 21, sending welcome message...
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> 220-FileZilla Server 0.9.60 beta
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> 220-written by Tim Kosse (tim.kosse@filezilla-project.org)
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> 220 Please visit https://filezilla-project.org/
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> USER 3213
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> 331 Password required for 3213
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> PASS *****
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> 530 Login or password incorrect!
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> QUIT
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> 221 Goodbye
[000027]2020/7/4 22:01:17 - (not logged in) (172.16.192.157)> disconnected.
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> Connected on port 21, sending welcome message...
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> 220-FileZilla Server 0.9.60 beta
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> 220-written by Tim Kosse (tim.kosse@filezilla-project.org)
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> 220 Please visit https://filezilla-project.org/
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> USER anonymous
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> 331 Password required for anonymous
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> PASS *****
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> 530 Login or password incorrect!
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> QUIT
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> 221 Goodbye
[000028]2020/7/4 22:01:19 - (not logged in) (172.16.192.157)> disconnected.
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> Connected on port 21, sending welcome message...
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> 220-FileZilla Server 0.9.60 beta
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> 220-written by Tim Kosse (tim.kosse@filezilla-project.org)
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> 220 Please visit https://filezilla-project.org/
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> USER anonymous
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> 331 Password required for anonymous
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> PASS *****
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> 530 Login or password incorrect!
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> QUIT
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> 221 Goodbye
[000029]2020/7/4 22:01:20 - (not logged in) (172.16.192.157)> disconnected.
```