二层交换机实验

【实验题目】**二层交换机实验**

【实验目的】掌握二层交换机的基本配置和使用方法。

【预备知识】

✧ **ping 命令可以用来测试网络的连通性。**

每次 ping 都将发出 4 个 echo 请求包给目的主机，目的主机每收到一个 echo 请求包(echo request)之后都将发回 echo 响应包(echo reply)。因此，ping 可以用来检测网络的双向连通性。

✧ **ping 命令：**

C:\>ping 目的主机的 IP 地址 ！发出 4 个请求包，例如，C:\>ping 192.168.1.2

C:\>ping -t 目的主机的 IP 地址 ！持续发出请求包，例如，C:\>ping -t 192.168.1.2

【注意事项】

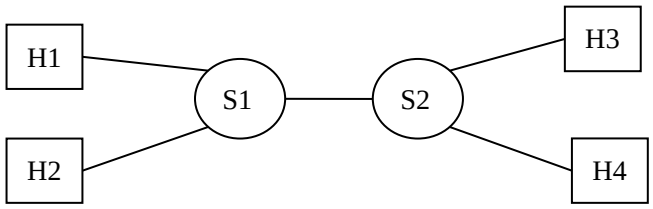
1、查看主机的校园网网卡的 IP 地址和子网掩码。 四台主机的 IP 地址为 172.16.X.2~172.16.X.5， 子网掩码都是 255.255.0.0,默认网关为 172.16.0.1，其中，X 为组号。如果没有设置好要设置一下。

2、每次做实验前先用#reload 重启设备,否则，可能会遗留前面配置的内容。

3、主机上禁用 Windows 防火墙（控制面板/系统和安全），否则防火墙可能会禁用 ping。

【实验内容】

(1)在两个交换机之间连接一条网线，每台交换机连两台主机。



四台主机配置 IPv4 地址：192.168.1.1，192.168.1.2 ，192.168.1.3 和 192.168.1.4,子网掩码均为 255.255.255.0。

1A、用 ipconfig 命令查出四台主机的 MAC 地址(注意：查实验网接口，不是校园网接口)：

(1) IP 地址：192.168.1.1 **MAC 地址：44-33-4c-0e-c8-29**

(2) IP 地址：192.168.1.2 **MAC 地址：44-33-4c-0e-c2-69**

(3) IP 地址：192.168.1.3 **MAC 地址：44-33-4c-0e-c8-5b**

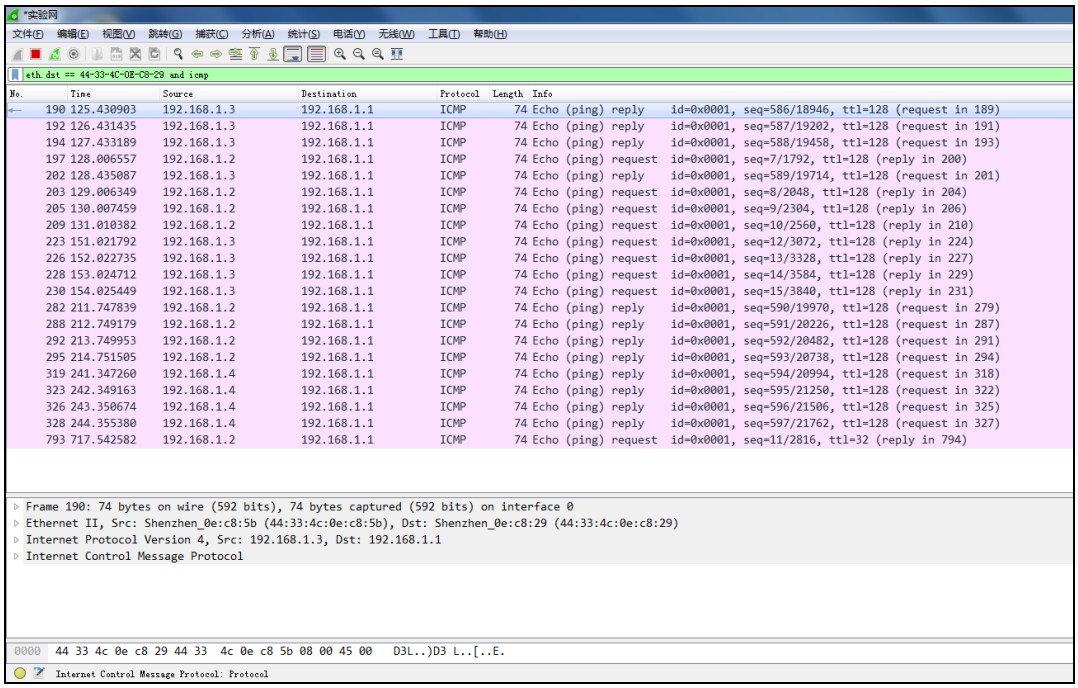
(4) IP 地址：192.168.1.4 **MAC 地址：4c-cc-6a-dc-4d-19**（自带电脑）

1B、Wireshark 以太网帧（DIXv2）截屏：

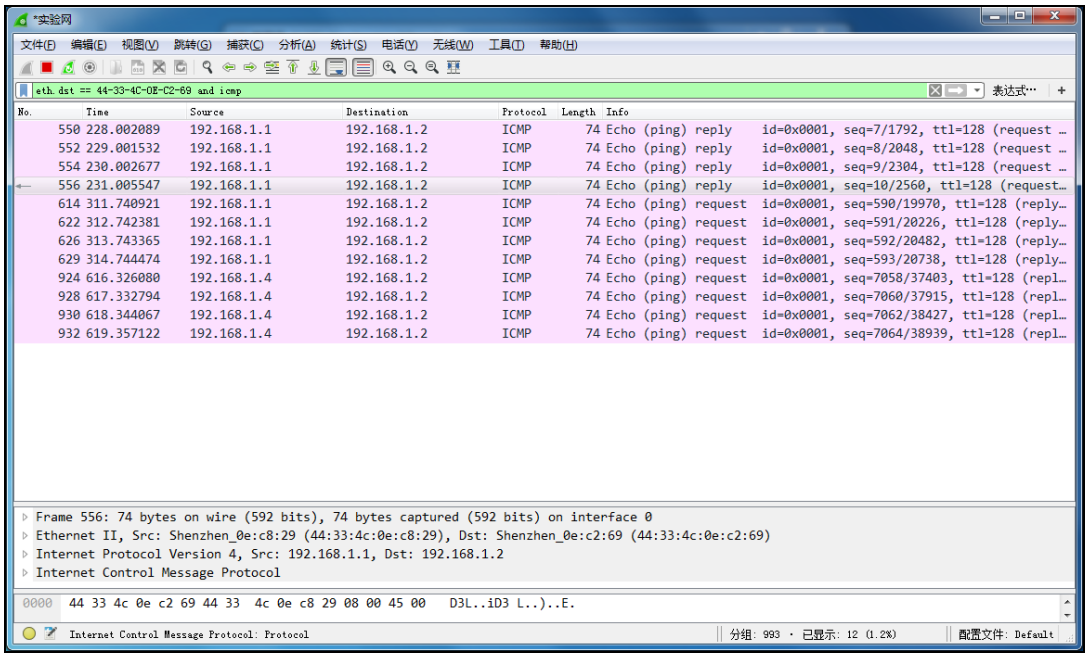
每台主机用 Wireshark 检测出一个其它主机发给自己的以太网帧并截屏（用 anysend 或者用 ping IP 地址产生包）。Wireshark Filter：eth.dst == 84-A6-C8-C0-BB-CF(主机的实验网网卡地址)进行过滤。

四台主机之间相互可以 ping 通，被 ping 的主机可以接到相应的 ICMP 数据包。这些数据包的截图如下，选中一个 ICMP 包条目，即可在下方窗口的“Ethernet II”中查看以太网帧的内容：

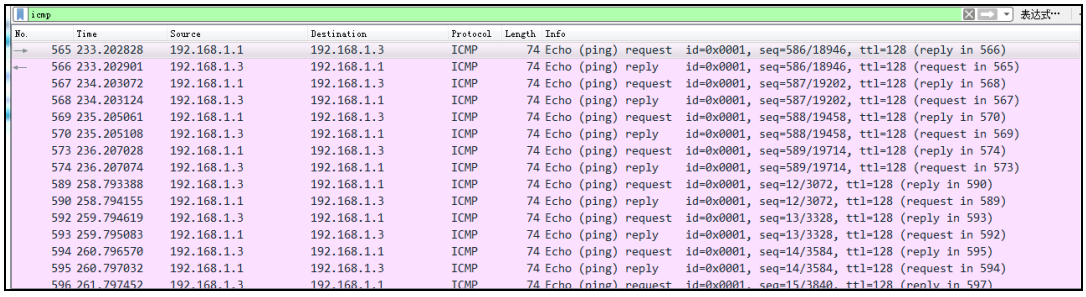
(1)


*实验网
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
eth.dst == 44-33-4C-0E-C8-29 and icmp
No. Time Source Destination Protocol Length Info
190 125.430903 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=586/18946, ttl=128 (request in 189)
192 126.431435 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=587/19202, ttl=128 (request in 191)
194 127.433189 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=588/19458, ttl=128 (request in 193)
197 128.006557 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 200)
202 128.435087 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=589/19714, ttl=128 (request in 201)
203 129.006349 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 204)
205 130.007459 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 206)
209 131.010382 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 210)
223 151.021792 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 224)
226 152.022735 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 227)
228 153.024712 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 229)
230 154.025449 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 231)
282 211.747839 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=590/19970, ttl=128 (request in 279)
288 212.749179 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=591/20226, ttl=128 (request in 287)
292 213.749953 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=592/20482, ttl=128 (request in 291)
295 214.751505 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=593/20738, ttl=128 (request in 294)
319 241.347260 192.168.1.4 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=594/20994, ttl=128 (request in 318)
323 242.349163 192.168.1.4 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=595/21250, ttl=128 (request in 322)
326 243.350674 192.168.1.4 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=596/21506, ttl=128 (request in 325)
328 244.355380 192.168.1.4 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=597/21762, ttl=128 (request in 327)
793 717.542582 192.168.1.2 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=11/2816, ttl=32 (reply in 794)
Frame 190: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: Shenzhen_0e:c8:5b (44:33:4c:0e:c8:5b), Dst: Shenzhen_0e:c8:29 (44:33:4c:0e:c8:29)
Internet Protocol Version 4, Src: 192.168.1.3, Dst: 192.168.1.1
Internet Control Message Protocol
0000 44 33 4c 0e c8 29 44 33 4c 0e c8 5b 08 00 45 00 D3L..)D3 L..[..E.
Internet Control Message Protocol: Protocol

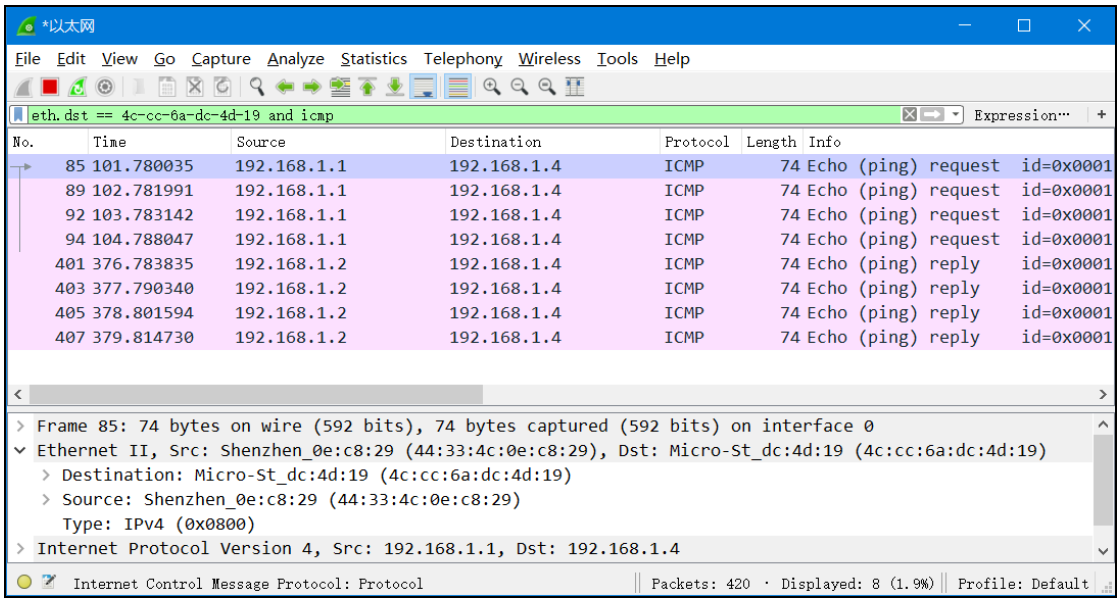
(2)


*实验网
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
eth.dst == 44-33-4C-0E-C2-69 and icmp
表达式…
No. Time Source Destination Protocol Length Info
550 228.002089 192.168.1.1 192.168.1.2 ICMP 74 Echo (ping) reply id=0x0001, seq=7/1792, ttl=128 (request …
552 229.001532 192.168.1.1 192.168.1.2 ICMP 74 Echo (ping) reply id=0x0001, seq=8/2048, ttl=128 (request …
554 230.002677 192.168.1.1 192.168.1.2 ICMP 74 Echo (ping) reply id=0x0001, seq=9/2304, ttl=128 (request …
556 231.005547 192.168.1.1 192.168.1.2 ICMP 74 Echo (ping) reply id=0x0001, seq=10/2560, ttl=128 (request…
614 311.740921 192.168.1.1 192.168.1.2 ICMP 74 Echo (ping) request id=0x0001, seq=590/19970, ttl=128 (reply…
622 312.742381 192.168.1.1 192.168.1.2 ICMP 74 Echo (ping) request id=0x0001, seq=591/20226, ttl=128 (reply…
626 313.743365 192.168.1.1 192.168.1.2 ICMP 74 Echo (ping) request id=0x0001, seq=592/20482, ttl=128 (reply…
629 314.744474 192.168.1.1 192.168.1.2 ICMP 74 Echo (ping) request id=0x0001, seq=593/20738, ttl=128 (reply…
924 616.326080 192.168.1.4 192.168.1.2 ICMP 74 Echo (ping) request id=0x0001, seq=7058/37403, ttl=128 (repl…
928 617.332794 192.168.1.4 192.168.1.2 ICMP 74 Echo (ping) request id=0x0001, seq=7060/37915, ttl=128 (repl…
930 618.344067 192.168.1.4 192.168.1.2 ICMP 74 Echo (ping) request id=0x0001, seq=7062/38427, ttl=128 (repl…
932 619.357122 192.168.1.4 192.168.1.2 ICMP 74 Echo (ping) request id=0x0001, seq=7064/38939, ttl=128 (repl…
Frame 556: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: Shenzhen_0e:c8:29 (44:33:4c:0e:c8:29), Dst: Shenzhen_0e:c2:69 (44:33:4c:0e:c2:69)
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.2
Internet Control Message Protocol
0000 44 33 4c 0e c2 69 44 33 4c 0e c8 29 08 00 45 00 D3L..iD3 L..)..E.
Internet Control Message Protocol: Protocol
分组: 993 · 已显示: 12 (1.2%)
配置文件: Default

(3)


icmp
表达式…
No. Time Source Destination Protocol Length Info
565 233.202828 192.168.1.1 192.168.1.3 ICMP 74 Echo (ping) request id=0x0001, seq=586/18946, ttl=128 (reply in 566)
566 233.202901 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=586/18946, ttl=128 (request in 565)
567 234.203072 192.168.1.1 192.168.1.3 ICMP 74 Echo (ping) request id=0x0001, seq=587/19202, ttl=128 (reply in 568)
568 234.203124 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=587/19202, ttl=128 (request in 567)
569 235.205061 192.168.1.1 192.168.1.3 ICMP 74 Echo (ping) request id=0x0001, seq=588/19458, ttl=128 (reply in 570)
570 235.205108 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=588/19458, ttl=128 (request in 569)
573 236.207028 192.168.1.1 192.168.1.3 ICMP 74 Echo (ping) request id=0x0001, seq=589/19714, ttl=128 (reply in 574)
574 236.207074 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) reply id=0x0001, seq=589/19714, ttl=128 (request in 573)
589 258.793388 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 590)
590 258.794155 192.168.1.1 192.168.1.3 ICMP 74 Echo (ping) reply id=0x0001, seq=12/3072, ttl=128 (request in 589)
592 259.794619 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 593)
593 259.795083 192.168.1.1 192.168.1.3 ICMP 74 Echo (ping) reply id=0x0001, seq=13/3328, ttl=128 (request in 592)
594 260.796570 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 595)
595 260.797032 192.168.1.1 192.168.1.3 ICMP 74 Echo (ping) reply id=0x0001, seq=14/3584, ttl=128 (request in 594)
596 261.797452 192.168.1.3 192.168.1.1 ICMP 74 Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 597)

(4)

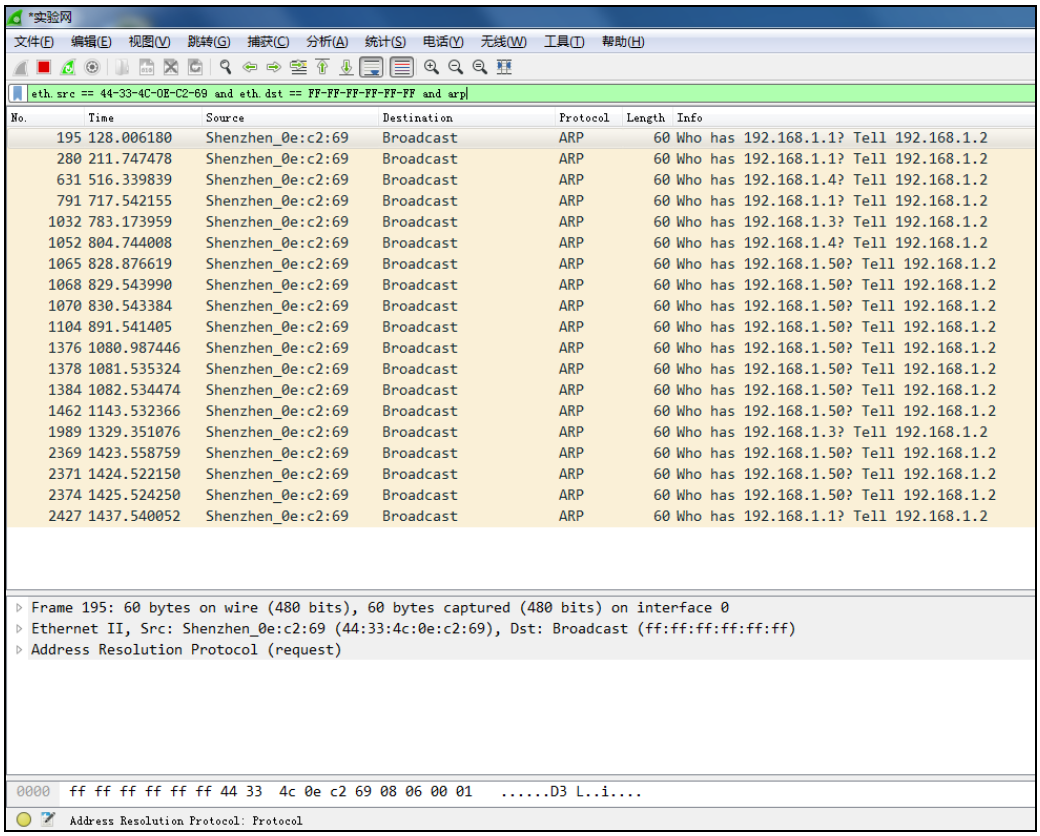


(2)直接用 anysend 发送一个广播帧，或者用一台主机（例如，192.168.1.1）ping 一个子网中不存在的 IP 地址(例如，192.168.1.50)来产生广播帧(ARP 包)。在所有主机上用 Wireshark 检测这个以太网广播帧（源 MAC 地址为该主机的地址，目的 MAC 地址为广播地址）并截屏。Wireshark Filter：eth.src == 84-A6-C8-C0-BB-CF and eth.dst == FF-FF-FF-FF-FF-FF。

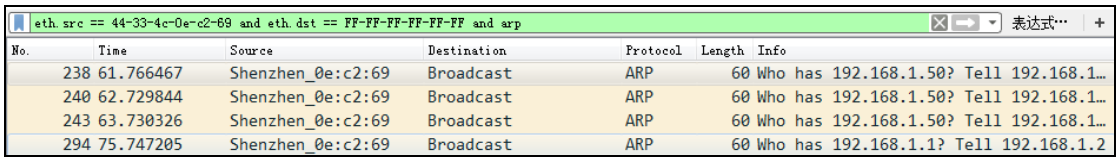
2A、在四台主机上捕捉发给自己的广播帧并截屏：

我们采用 ping 一个不存在的 IP 地址（192.168.1.50）来产生广播帧。发出 ping 的主机不知道 192.168.1.50 的 MAC 地址，因此它会广播出 ARP 包，然后可以在其他主机上用 Wireshark 截获这些包。为了方便起见，我们在过滤条件中增加了“and arp”以筛选出 ARP 包。如下：

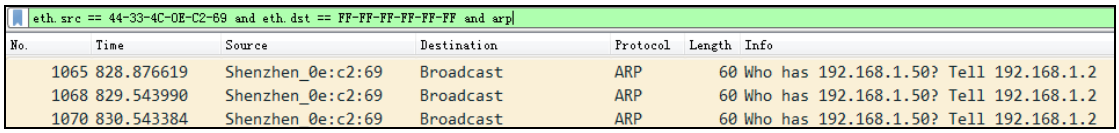
(1)



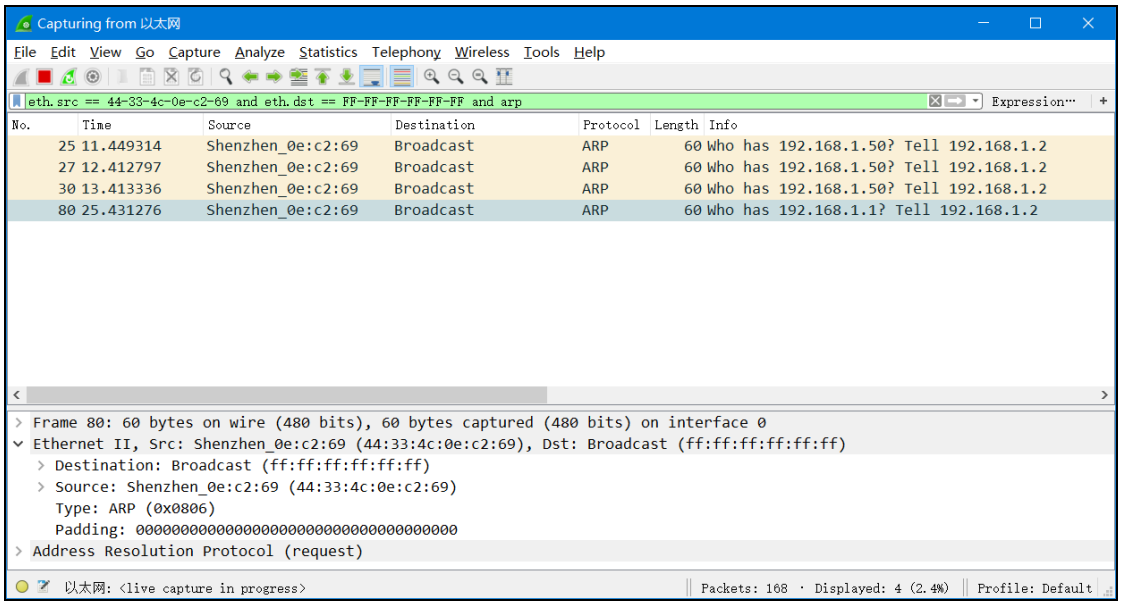
(2)



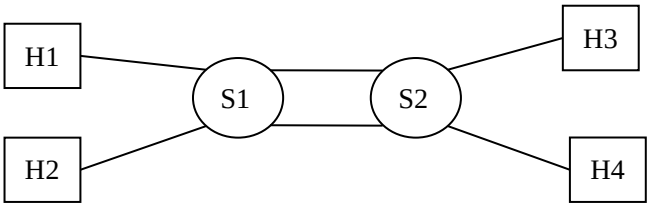
(3)



(4)



(3)在两个交换机之间再连接一条网线。

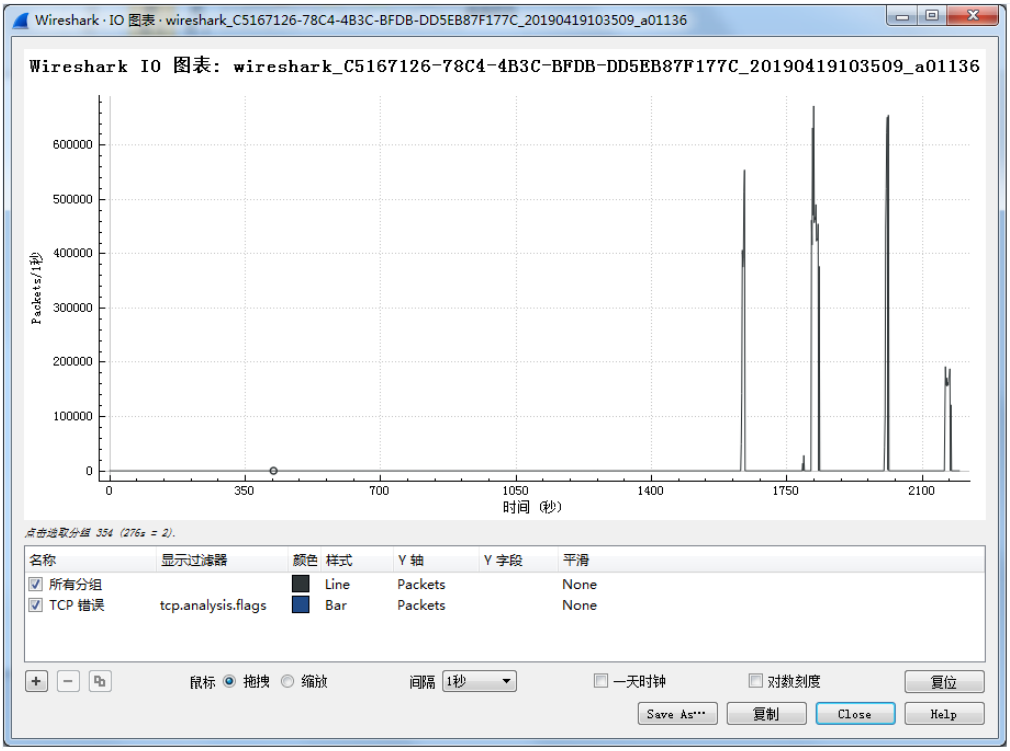


用步骤(2)的方法产生广播帧，并用 Wireshark 检测广播风暴（capture/interfaces），得到实验网接口收发包的速度(packets/s)，截屏该画面。注意：当发现广播风暴时要**及时断开**其中一条网线以避免死机。

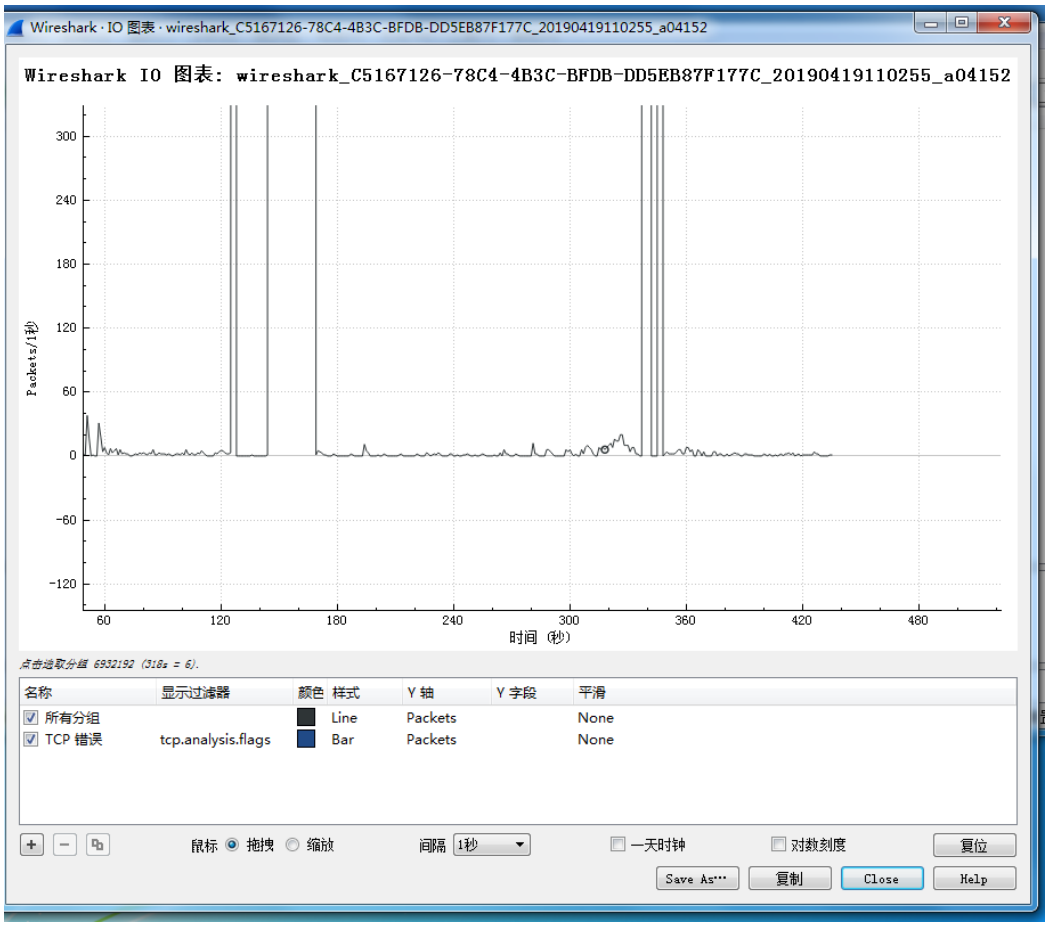
在四台主机上捕捉广播风暴并截屏收发包的速度：

使用 Wireshark 的“IO 图表（IO Graph)”功能观察主机收发包的速度及其随时间的变化曲线。在连接了第二条网线后，可以观察到每台主机抓到数据包的速度都超过了 10 万个/秒，由此可见的确产生了广播风暴。

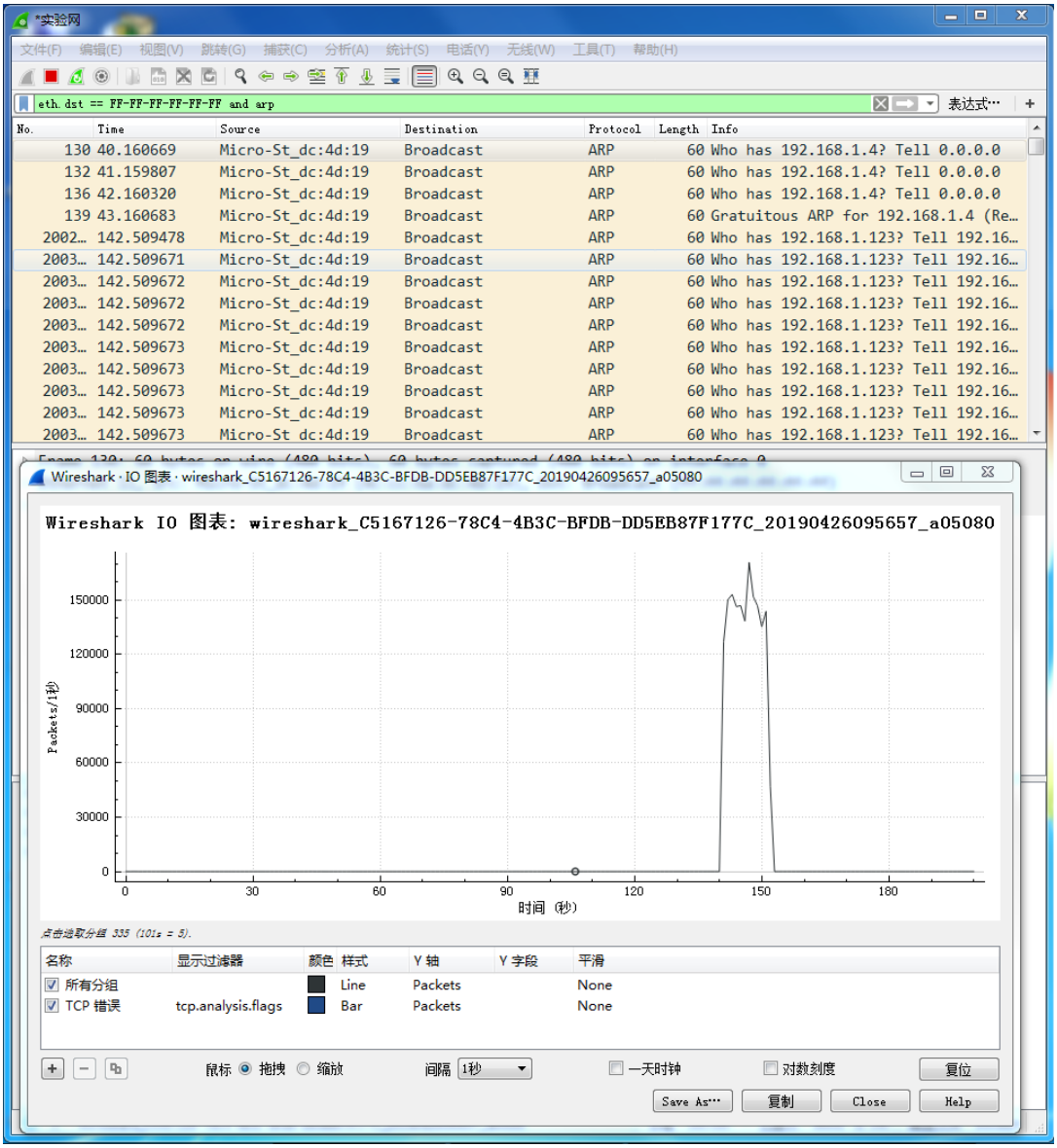
(1)


Wireshark · IO 图表 · wireshark_C5167126-78C4-4B3C-BFDB-DD5EB87F177C_20190419103509_a01136
Wireshark IO 图表: wireshark_C5167126-78C4-4B3C-BFDB-DD5EB87F177C_20190419103509_a01136
Packets/1秒
600000
500000
400000
300000
200000
100000
0
0
350
700
1050
1400
1750
2100
时间 (秒)
名称 显示过滤器 颜色 样式 Y 轴 Y 字段 平滑
所有分组 Line Packets None
TCP 错误 tcp.analysis.flags Bar Packets None
鼠标 拖拽 缩放
间隔 1秒
一天时钟
对数刻度
复位
Save As···
复制
Close
Help

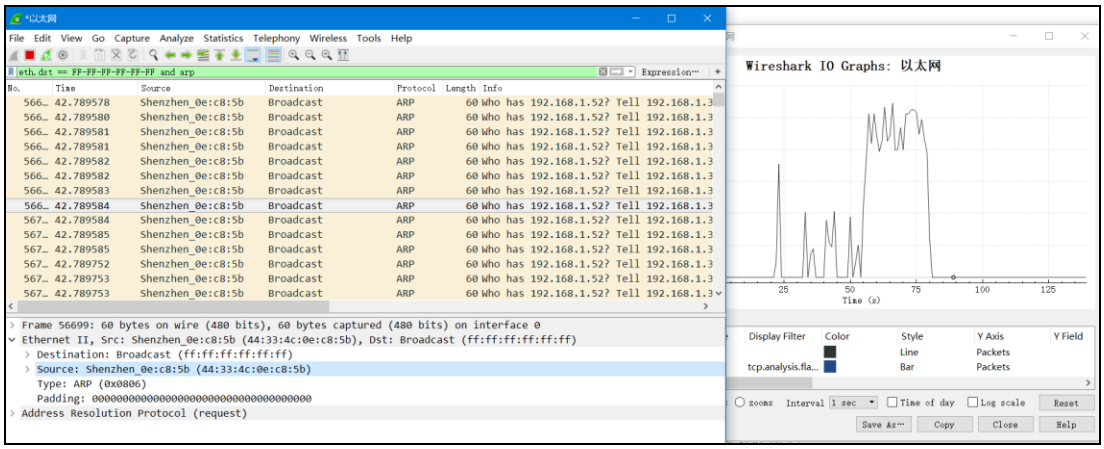
(2)


Wireshark · IO 图表 · wireshark_C5167126-78C4-4B3C-BFDB-DD5EB87F177C_20190419110255_a04152
Wireshark IO 图表: wireshark_C5167126-78C4-4B3C-BFDB-DD5EB87F177C_20190419110255_a04152
Packets/1秒
300
240
180
120
60
0
-60
-120
60
120
180
240
300
360
420
480
时间 (秒)
名称 显示过滤器 颜色 样式 Y 轴 Y 字段 平滑
所有分组 Line Packets None
TCP 错误 tcp.analysis.flags Bar Packets None
鼠标 拖拽 缩放
间隔 1秒
一天时钟
对数刻度
复位
Save As···
复制
Close
Help

(3)



(4)



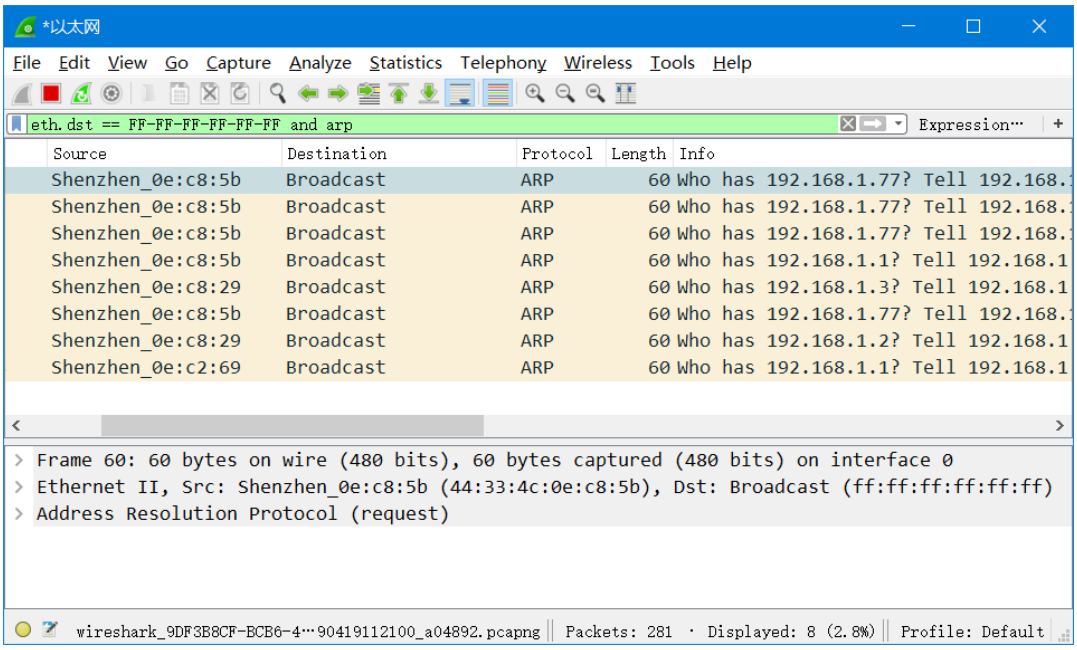
(4) 先在两台交换机上启动生成树算法，然后在它们之间重新连接两条网线，检测是否会出现广播风暴，截屏 Wireshark。启动生成树算法的命令：(config)#spanning-tree

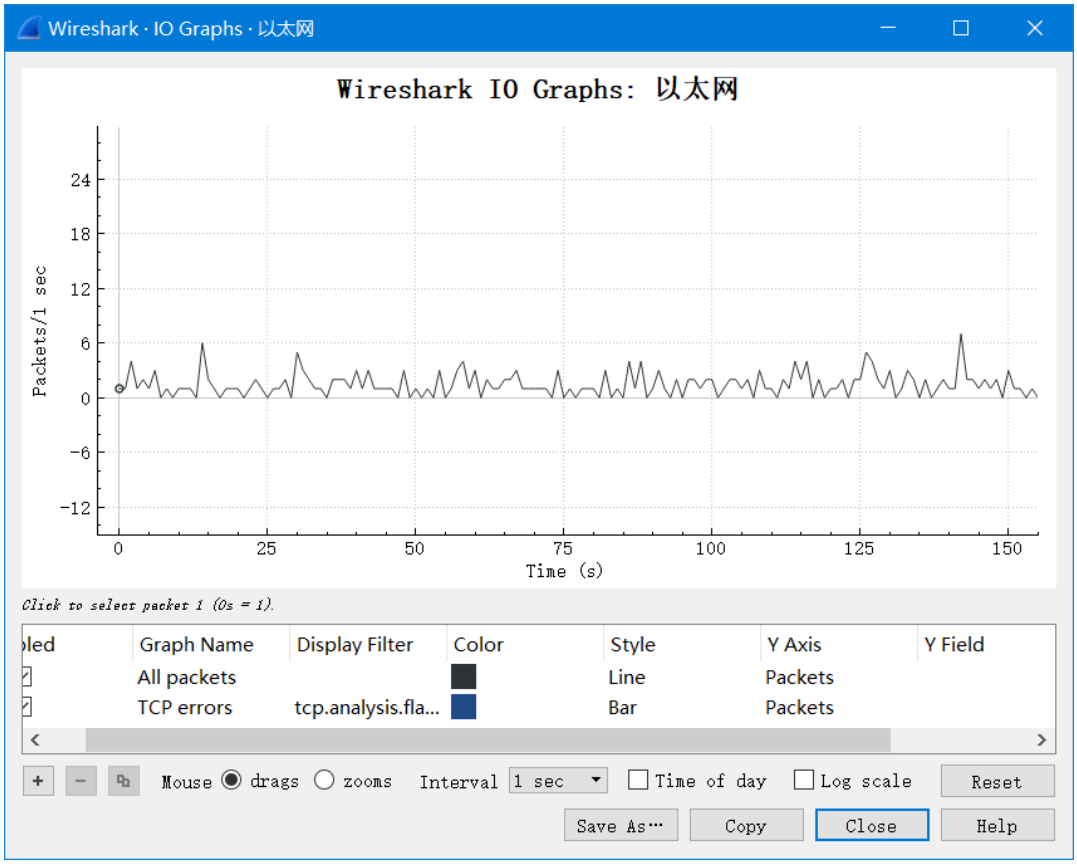
4A. 是否存在广播风暴？(是/否)

否。

4B. 经过 2 分钟截屏 Wireshark（capture/interfaces）：

每台主机的抓包结果相似，收发数据包的速度都在 10 个/秒以下，远远小于之前的超过 10 万个/秒，因此说明没有产生广播风暴。下面的两张截图取自 192.168.1.4 的 Wireshark。





4C. 在两台交换机上执行显示生成树参数的命令并截屏：

(config)#show spanning-tree

从以下截图的参数中可以看出，目前 Switch2 是根网桥。

Switch1：

```
20-S5750-1(config)#show spanning-tree
StpVersion : MSTP
SysStpStatus : ENABLED
MaxAge : 20
HelloTime : 2
ForwardDelay : 15
BridgeMaxAge : 20
BridgeHelloTime : 2
BridgeForwardDelay : 15
MaxHops: 20
TxHoldCount : 3
PathCostMethod : Long
BPDUGuard : Disabled
BPDUFilter : Disabled
LoopGuardDef  : Disabled

###### mst 0 vlans map : ALL
BridgeAddr : 5869.6c15.59e2
Priority: 32768
TimeSinceTopologyChange : 0d:0h:15m:9s
TopologyChanges : 1
DesignatedRoot : 32768.5869.6c15.59ca
RootCost : 0
RootPort : GigabitEthernet 0/13
CistRegionRoot : 32768.5869.6c15.59ca
CistPathCost : 20000
```

Switch2：

```
20-S5750-2(config)#show spanning-tree
StpVersion : MSTP
SysStpStatus : ENABLED
MaxAge : 20
HelloTime : 2
ForwardDelay : 15
BridgeMaxAge : 20
BridgeHelloTime : 2
BridgeForwardDelay : 15
MaxHops: 20
TxHoldCount : 3
PathCostMethod : Long
BPDUGuard : Disabled
BPDUFilter : Disabled
LoopGuardDef  : Disabled

###### mst 0 vlans map : ALL
BridgeAddr : 5869.6c15.59ca
Priority: 32768
TimeSinceTopologyChange : 0d:0h:15m:15s
TopologyChanges : 1
DesignatedRoot : 32768.5869.6c15.59ca
RootCost : 0
RootPort : 0
CistRegionRoot : 32768.5869.6c15.59ca
CistPathCost : 0
```

4D. 在两台交换机上执行显示接口 f0/1 和 f0/2 的生成树参数的命令并截屏：

(config)#show spanning-tree interface *f0/2* 或 f0/1

在我们的实验中，两台交换机分别以接口 g0/13- g0/13 和接口 g0/14- g0/14 互相连接。

Switch1：

```
20-S5750-1(config)#show spanning-tree interface g0/13

PortAdminPortFast : Disabled
PortOperPortFast : Disabled
PortAdminAutoEdge : Enabled
PortOperAutoEdge : Disabled
PortAdminLinkType : auto
PortOperLinkType : point-to-point
PortBPDUGuard : Disabled
PortBPDUFilter : Disabled
PortGuardmode  : None

###### MST 0 vlans mapped :ALL
PortState : forwarding
PortPriority : 128
PortDesignatedRoot : 32768.5869.6c15.59ca
PortDesignatedCost : 0
PortDesignatedBridge :32768.5869.6c15.59ca
PortDesignatedPortPriority : 128
PortDesignatedPort : 13
PortForwardTransitions : 1
PortAdminPathCost : 20000
PortOperPathCost : 20000
Inconsistent states : normal
PortRole : rootPort
```

Switch2：

```
20-S5750-2(config)#show spanning-tree interface g0/13

PortAdminPortFast : Disabled
PortOperPortFast : Disabled
PortAdminAutoEdge : Enabled
PortOperAutoEdge : Disabled
PortAdminLinkType : auto
PortOperLinkType : point-to-point
PortBPDUGuard : Disabled
PortBPDUFilter : Disabled
PortGuardmode  : None

###### MST 0 vlans mapped :ALL
PortState : forwarding
PortPriority : 128
PortDesignatedRoot : 32768.5869.6c15.59ca
PortDesignatedCost : 0
PortDesignatedBridge :32768.5869.6c15.59ca
PortDesignatedPortPriority : 128
PortDesignatedPort : 13
PortForwardTransitions : 1
PortAdminPathCost : 20000
PortOperPathCost : 20000
Inconsistent states : normal
PortRole : designatedPort
```

4E. 根据上面结果填表

	网桥优先权（priority）	网桥 MAC 地址 (BridgeAddr)	根网桥 ID (DesignatedRoot)	到根的距离 (RootCost)	根端口 (RootPort)	指定端口 (Designated)
Switch1	32768	5869.6c15.59e2	32768.5869.6c15.59ca	0	g0/13	13
Switch2	32768	5869.6c15.59ca	32768.5869.6c15.59ca	0	0	13

4F. 显示两台交换机的 MAC 地址表，通过 ping 让每个 MAC 地址表包含全部主机的 MAC 地址，然后截屏：

命令：**#show mac-address-table**

Switch1 的 MAC 地址表截屏：

```
20-S5750-1(config)#show mac-address-table
Vlan        MAC Address          Type      Interface
----------  -------------------- --------  --------------------
   1        0088.9900.1376       DYNAMIC   GigabitEthernet 0/17
   1        4433.4c0e.b6ef       DYNAMIC   GigabitEthernet 0/13
   1        4433.4c0e.ce18       DYNAMIC   GigabitEthernet 0/19
   1        4ccc.6adc.4d19       DYNAMIC   GigabitEthernet 0/13
   1        5869.6c15.59ca       DYNAMIC   GigabitEthernet 0/13
```

Switch2 的 MAC 地址表截屏：

```
20-S5750-2(config)#show mac-address-table
Vlan        MAC Address          Type      Interface
----------  -------------------- --------  --------------------
   1        0088.9900.1376       DYNAMIC   GigabitEthernet 0/13
   1        4433.4c0e.b6ef       DYNAMIC   GigabitEthernet 0/17
   1        4433.4c0e.ce18       DYNAMIC   GigabitEthernet 0/13
   1        4ccc.6adc.4d19       DYNAMIC   GigabitEthernet 0/21
   1        5869.6c15.59e2       DYNAMIC   GigabitEthernet 0/13
```

(5) 在(4)的基础上，修改优先权令另一台交换机成为根网桥，ping 通后查看生成树信息并填表：

(config)#spanning-tree priority *4096*　　!设置交换机优先权为 4096。默认优先权为 32768

	网桥优先权	网桥 MAC 地址	根网桥 ID	到根的距离	根端口	指定端口
Switch1	4096	5869.6c15.59e2	4096.5869.6c15.59e2	0	0	13
Switch2	32768	5869.6c15.59ca	4096.5869.6c15.59e2	0	g0/13	13