

实验二 数据链路层实验

- 在网络课程学习中，802.3 和 ETHERNETII 规定了以太网 MAC 层的报文格式分为 7 字节的前导符、1 字节的起始符、6 字节的目的 MAC 地址、6 字节的源 MAC 地址、2 字节的类型、数据字段和 4 字节的数据校验字段。对于选中的报文，缺少哪些字段，为什么？
 - 缺少前导符和起始符，和数据校验字段，这两个字段和校验字段在网卡接收 MAC 帧时被去掉了，因此实验抓包软件的报文中没有这些字段。
- 查看交换机的 MAC 地址表，结果为：

```
<S1>dis mac-address
No Multicast Mac addresses found.
MAC ADDR          VLAN ID          STATE          PORT INDEX      AGING TIME(s)
000c-2900-d32d     1                LEARNED        Ethernet1/0/1    AGING
3c8c-4068-d371     1                LEARNED        Ethernet1/0/1    AGING
3c8c-4068-d372     1                LEARNED        Ethernet1/0/2    AGING
a036-9f0a-1dcb     1                LEARNED        Ethernet1/0/2    AGING
a036-9f0a-237d     1                LEARNED        Ethernet1/0/1    AGING
<S1>
```

1)、解释 MAC 地址表中各字段的含义？

- MAC ADDR 为设备的 MAC 地址
- VLAN ID 为端口所在的 VLAN 编号
- PORT INDEX 表示源 MAC 地址为由该端口号学习来的
- STATE 表示该记录怎么得来的（学习/配置）
- AGING TIME 表示该记录的生命时间

2)、这个实验能够说明 MAC 地址表的学习是来源于数据帧的源 MAC 地址而非目的 MAC 地址吗？如果能，为什么？如果不能，试给出一个验证方法。

- 不能。方法：清空交换机的 MAC 地址表，断开交换机与 PCB 连线，然后 ping PCB，查看交换机的 MAC 地址表，这时 MAC 中只有 PCA 的 MAC 地址学习记录。

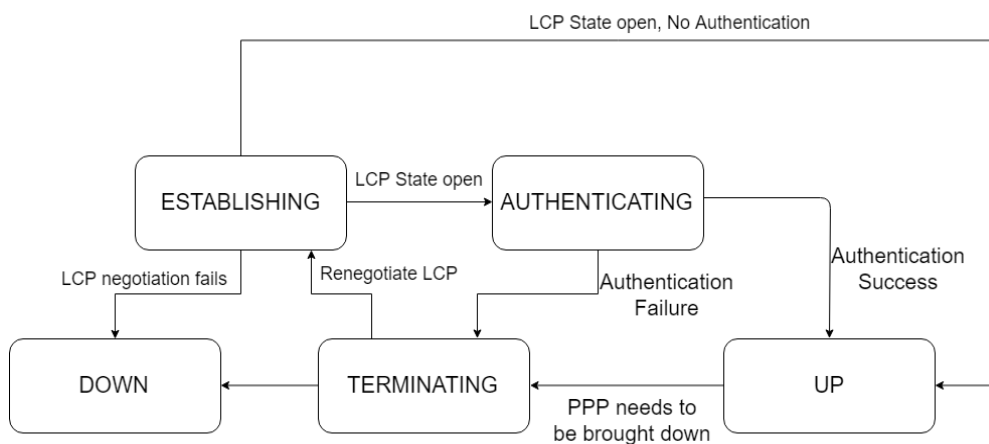
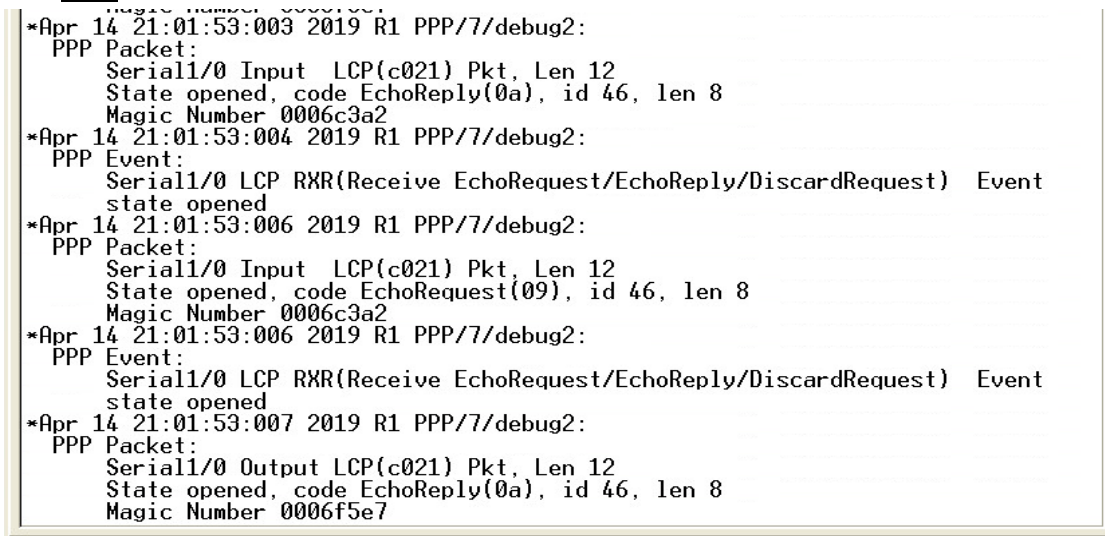
3. 在 VLAN 实验中，实验中的计算机能否通讯，请将结果填入下表：

		Ping 命令	能否 ping 通
同一 VLAN 中	PCA ping PCB	Ping 192.168.2.11	能
不同 VLAN 中	PCB ping PCC	Ping 192.168.2.12	不能

4. 交换机在没有配置 VLAN 时，冲突域和广播域各有哪些端口？配置了 VLAN 以后呢？

- 交换机在没有配置 VLAN 时：
 - 广播域：交换机所有的端口是一个广播域
 - 冲突域：每个端口是一个冲突域
- 配置了 VLAN 以后：
 - 广播域：同一个 VLAN 属于一个广播域
 - 冲突域：每个端口是一个冲突域

5. 根据 R1 上的 debug 显示信息，画出 LCP 协议在协商过程中的状态转移图（事件驱动、状态转移）。



6. 根据 debug 显示信息，画出 PPP 协议 PAP 验证过程的状态转移图。

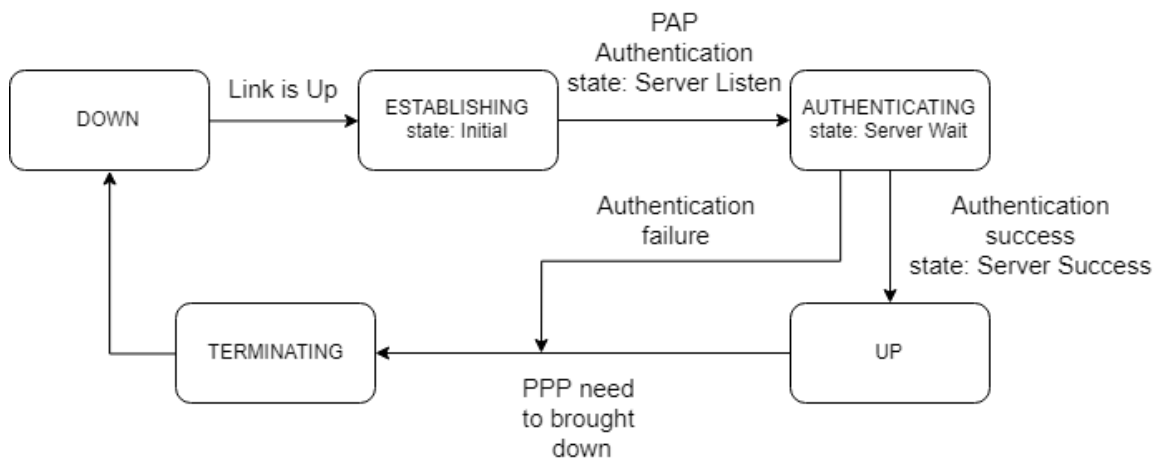
```

Serial1/0 PAP : ServerSuccess --> Initial
[R1-Serial1/0]undo shutdown
[R1-Serial1/0]
%Apr 14 21:26:30:820 2019 R1 IFNET/3/LINK_UPDOWN: Serial1/0 link status is UP.
%Apr 14 21:26:33:007 2019 R1 IFNET/5/LINEPROTO_UPDOWN: Line protocol on the inte
rface Serial1/0 is UP.
*Apr 14 21:26:33:007 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 PAP Initial Event
state Initial
*Apr 14 21:26:33:008 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 PAP Server Lower Up Event
state Initial
*Apr 14 21:26:33:008 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 PAP : Initial --> ServerListen
*Apr 14 21:26:33:014 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Input PAP(c023) Pkt, Len 16

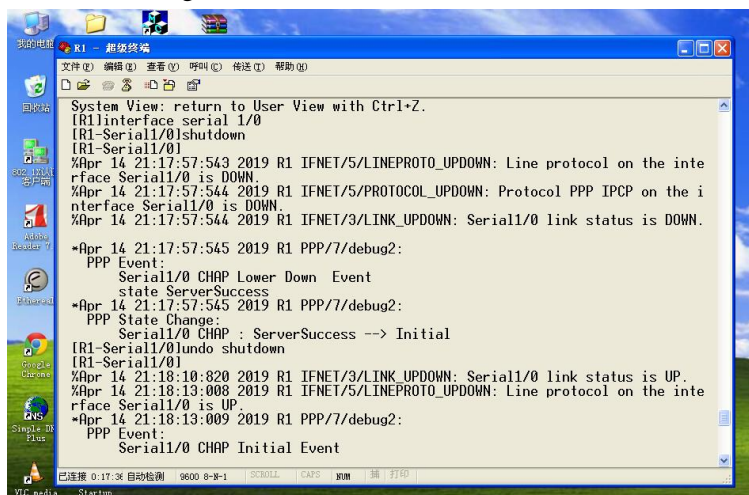
State ServerListen, code Request(01), id 1, len 12
Host Len: 3 Name:RTB
Pwd:*****
*Apr 14 21:26:33:014 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 PAP Receive Request Event
state ServerListen
*Apr 14 21:26:33:014 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 PAP : ServerListen --> WaitAAA
*Apr 14 21:26:33:017 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 PAP AAA Result Event
state WaitAAA

PPP Event:
Serial1/0 PAP Server Lower Up Event
state Initial
*Apr 14 21:26:33:008 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 PAP : Initial --> ServerListen
*Apr 14 21:26:33:014 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Input PAP(c023) Pkt, Len 16

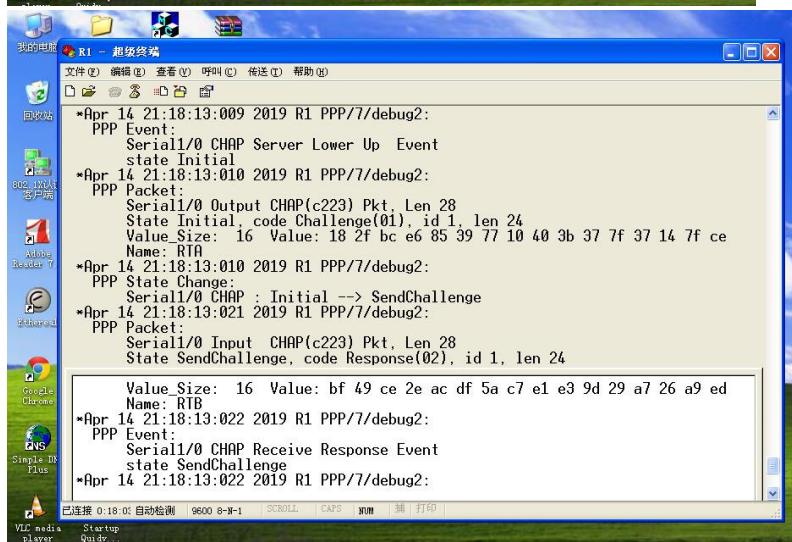
State ServerListen, code Request(01), id 1, len 12
Host Len: 3 Name:RTB
Pwd:*****
*Apr 14 21:26:33:014 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 PAP Receive Request Event
state ServerListen
*Apr 14 21:26:33:014 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 PAP : ServerListen --> WaitAAA
*Apr 14 21:26:33:017 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 PAP AAA Result Event
state WaitAAA
*Apr 14 21:26:33:018 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Output PAP(c023) Pkt, Len 36
State WaitAAA, code Ack(02), id 1, len 32
Msg Len: 27 Msg:Welcome to use this device.
*Apr 14 21:26:33:118 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 PAP : WaitAAA --> ServerSuccess
%Apr 14 21:26:33:269 2019 R1 IFNET/5/PROTOCOL_UPDOWN: Protocol PPP IPCP on the i
nterface Serial1/0 is UP.
  
```



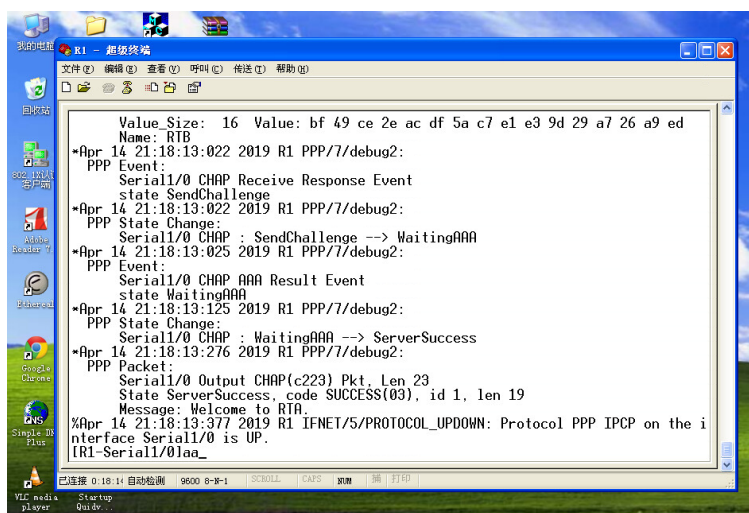
7. 根据 debug 显示信息，画出 PPP 协议的 CHAP 验证的状态转移图。（选作）



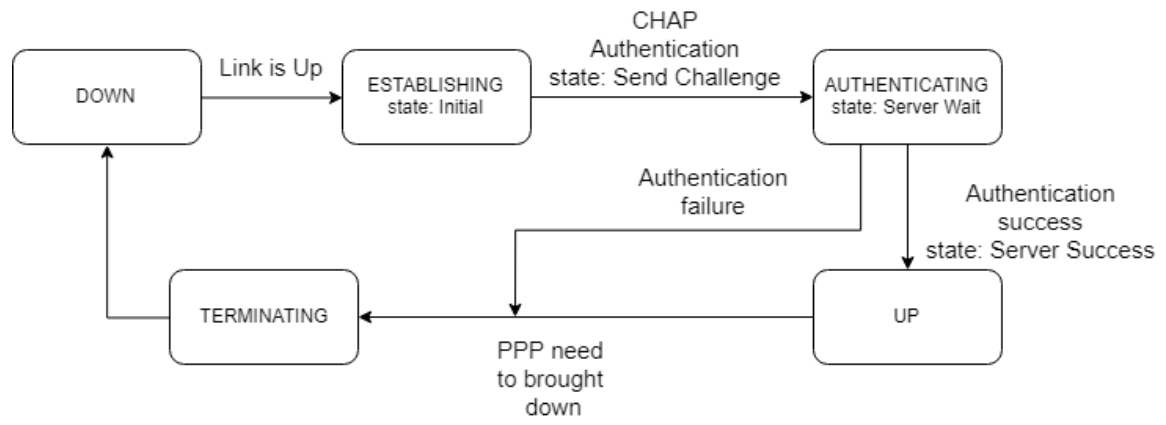
```
System View: return to User View with Ctrl+Z.
[R1]interface serial 1/0
[R1-Serial1/0]shutdown
[R1-Serial1/0]
%Apr 14 21:17:57:543 2019 R1 IFNET/5/LINEPROTO_UPDOWN: Line protocol on the inte
rface Serial1/0 is DOWN.
%Apr 14 21:17:57:544 2019 R1 IFNET/5/PROTOCOL_UPDOWN: Protocol PPP IPCP on the i
nterface Serial1/0 is DOWN.
%Apr 14 21:17:57:544 2019 R1 IFNET/3/LINK_UPDOWN: Serial1/0 link status is DOWN.
*Apr 14 21:17:57:545 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 CHAP Lower Down Event
state ServerSuccess
*Apr 14 21:17:57:545 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 CHAP : ServerSuccess --> Initial
[R1-Serial1/0]undo shutdown
[R1-Serial1/0]
%Apr 14 21:18:10:820 2019 R1 IFNET/3/LINK_UPDOWN: Serial1/0 link status is UP.
%Apr 14 21:18:13:008 2019 R1 IFNET/5/LINEPROTO_UPDOWN: Line protocol on the inte
rface Serial1/0 is UP.
*Apr 14 21:18:13:009 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 CHAP Initial Event
```



```
*Apr 14 21:18:13:009 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 CHAP Server Lower Up Event
state Initial
*Apr 14 21:18:13:010 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Output CHAP(c223) Pkt, Len 28
State Initial, code Challenge(01), id 1, len 24
Value_Size: 16 Value: 18 2f bc e6 85 39 77 10 40 3b 37 7f 37 14 7f ce
Name: RTA
*Apr 14 21:18:13:010 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 CHAP : Initial --> SendChallenge
*Apr 14 21:18:13:021 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Input CHAP(c223) Pkt, Len 28
State SendChallenge, code Response(02), id 1, len 24
Value_Size: 16 Value: bf 49 ce 2e ac df 5a c7 e1 e3 9d 29 a7 26 a9 ed
Name: RTB
*Apr 14 21:18:13:022 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 CHAP Receive Response Event
state SendChallenge
*Apr 14 21:18:13:022 2019 R1 PPP/7/debug2:
```



```
Value_Size: 16 Value: bf 49 ce 2e ac df 5a c7 e1 e3 9d 29 a7 26 a9 ed
Name: RTB
*Apr 14 21:18:13:022 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 CHAP Receive Response Event
state SendChallenge
*Apr 14 21:18:13:022 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 CHAP : SendChallenge --> WaitingAAA
*Apr 14 21:18:13:025 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 CHAP AAA Result Event
state WaitingAAA
*Apr 14 21:18:13:125 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 CHAP : WaitingAAA --> ServerSuccess
*Apr 14 21:18:13:276 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Output CHAP(c223) Pkt, Len 23
State ServerSuccess, code SUCCESS(03), id 1, len 19
Message: Welcome to RTA.
%Apr 14 21:18:13:377 2019 R1 IFNET/5/PROTOCOL_UPDOWN: Protocol PPP IPCP on the i
nterface Serial1/0 is UP.
[R1-Serial1/0]aa_
```



实验三 网络层实验

1. 将执行命令的结果填入下表：

2.6.1 中步骤 2 中的执行结果	No ARP Entries Found
2.6.1 中步骤 4 中的执行结果	Interface : 192.168.1.22 --- 0x2 Internet Address Physical Address Type 192.168.1.21 00-0c-29-5f-7b-e7 dynamic
2.6.2 中步骤 11 中的执行结果	Interface : 192.168.1.22 --- 0x2 Internet Address Physical Address Type 192.168.1.10 3c-8c-40-29-f2-c7 dynamic

2. 分析 2.6.1 步骤 6 中截获的报文，统计“Protocol”字段填空：有 2 个 ARP 报文，有 8 个 ICMP 报文。在所有报文中，ARP 报文中 ARP 协议树的“Opcode”字段有两个取值 1，2，两个取值分别表达什么信息？

- 1 是表达 request
- 2 是表达 reply

3. 根据 2.6.1 步骤 6 分析 ARP 报文结构：选中第一条 ARP 请求报文和第一条 ARP 应答报文，将 ARP 请求报文和 ARP 应答报文中的字段信息填入下表：（这报文是 516 和 517 的）

字段项	ARP 请求数据报文	ARP 应答数据报文
链路层 Destination 项	Broadcast (ff:ff:ff:ff:ff:ff)	Vmware_58:a0:03 (00:0c:29:58:a0:03)
链路层 Source 项	Vmware_58:a0:03 (00:0c:29:58:a0:03)	Vmware_5f:7b:e7 (00:0c:29:5f:7b:e7)
网络层 Sender MAC Address	Vmware_58:a0:03 (00:0c:29:58:a0:03)	Vmware_5f:7b:e7 (00:0c:29:5f:7b:e7)
网络层 Sender IP Address	192.168.1.22	192.168.1.21
网络层 Target MAC Address	00:00:00_00:00:00 (00:00:00:00:00:00)	Vmware_58:a0:03 (00:0c:29:58:a0:03)
网络层 Target IP Address	192.168.1.21	192.168.1.22

4. (1) 比较 ping1-学号中截获的报文信息，少了什么报文？简述 ARP Cache 的作用。

- 少了 ARP 报文。主机的 ARP Cache 存放最近的 IP 地址与 MAC 地址的对应关系，一旦受到 ARP 应答，主机将获得的 IP 地址和 MAC 地址的对应关系存到 ARP Cache 中，当发送报文时，首先去 ARP Cache 中查找相应的项，若找到相应的项那么将报文直接发送。

(2) 按照图-4 重新进行组网，并确保连线正确。修改计算机的 IP 地址，并将 PC A 的默认网关修改为 192.168.1.10，PC B 的默认网关修改为 192.168.2.10。考虑如果不设置默认网关会有什么后果？

- 如果不设置默认网关那么无法访问不同网段的主机。

5.根据 2.6.2 步骤 12 分析 ARP 报文结构：选中第一条 ARP 请求报文和第一条 ARP 应答报文，将 ARP 请求报文和 ARP 应答报文中的字段信息与上表进行对比。与 ARP 协议在相同网段内解析的过程相比较，有何异同点？

- 请求报文中，相同的网段网络层中的 Target IP Address 为 PCB 的 (192.168.1.21)，而不同网段网络层的 Target IP Address 为 PCA 的默认网关的 IP (192.168.1.10)。应答报文中，相同的网段链路层的 Source 和网络层的 Sender MAC Address 都是 PCB 的 MAC 地址 (Vmware_5f:7b:e7 (00:0c:29:5f:7b:e7))，而不同的网段链路层的 Source 和网络层的 Sender MAC Address 都是 PCA 默认网关 S1 e1/0/1 的 MAC 地址 (Hangzhou_29:f2:c7 (3c:8c:40:29:f2:c7))，相同网段网络层的 Sender IP address 为 PCB 的 IP (192.168.1.21)，而不同网段网络层的 Sender IP Address 为 PCA 的默认网关的 IP (192.168.1.10)。

6.根据 3.6.1 步骤 2——在 PC A 和 PC B 上启动 Wireshark 软件进行报文截获，然后 PC A ping PC B，分析截获的 ICMP 报文：共有 8 个 ICMP 报文，分别属于哪些种类？对应的种类和代码字段分别是什么？请分析报文中的哪些字段保证了回送请求报文和回送应答报文的一一对应？

- 这报文都是询问报文。第 1, 3, 5, 7 (83, 86, 91, 95) 属于 request 类型,对应的字段为 Type: 8 (Echo (Ping) request)。第 2, 4, 6, 8 (84, 87, 92, 96) 属于 reply 类型，对应的字段为 Type: 0 (Echo (Ping) Reply)。网络层的 Source 和 destination 字段保证了回送请求报文和回送应答报文的一一对。

7.根据 3.6.1 步骤 3——在 PC A 和 PC B 上启动 Wireshark 软件进行报文截获，运行 pingtest 程序，设置地址掩码请求报文参数，分析截获报文填写下表：

地址掩码请求报文		地址掩码应答报文	
ICMP 字段名	字段值	ICMP 字段名	字段值
Type	17 (Address mask request)	Type	18 (Address mask reply)
Code	0	Code	0
Checksum	0xe3ff [correct]	Checksum	0xe3fe [correct]
Identifier (BE)	2560 (0x0a00)	Identifier (BE)	2560 (0x0a00)
Identifier (LE)	10 (0x000a)	Identifier (LE)	10 (0x000a)
Sequence number (BE)	256 (0x0100)	Sequence number (BE)	256 (0x0100)
Sequence number (LE)	1 (0x0001)	Sequence number (LE)	1 (0x0001)
Address Mask	0.0.0.0	Address Mask	255.255.255.0

8.根据 3.6.1 步骤 4——在 PC A 和 PC B 上启动 Wireshark 软件进行报文截获,运行 pingtest 程序,设置时间戳请求报文参数,分析截获报文填写下表:

时间戳请求报文		时间戳应答报文	
ICMP 字段名	字段值	ICMP 字段名	字段值
Type	13 (Timestamp request)	Type	14 (Timestamp reply)
Code	0	Code	0
Checksum	0xe7ff [correct]	Checksum	0xb0f3 [correct]
Identifier (BE)	2560 (0x0a00)	Identifier (BE)	2560 (0x0a00)
Identifier (LE)	10 (0x000a)	Identifier (LE)	10 (0x000a)
Sequence number (BE)	256 (0x0100)	Sequence number (BE)	256 (0x0100)
Sequence number (LE)	1 (0x0001)	Sequence number (LE)	1 (0x0001)
Originate timestamp	0 (0 seconds after midnight UTC)	Originate timestamp	0 (0 seconds after midnight UTC)
Receive timestamp	0 (0 seconds after midnight UTC)	Receive timestamp	3036833283 (15 hours, 50 minutes, 17.013 seconds after midnight UTC)
Transmit timestamp	0 (0 seconds after midnight UTC)	Transmit timestamp	3036833283 (15 hours, 50 minutes, 17.013 seconds after midnight UTC)

通过上述实验,仔细体会 ICMP 询问报文的作用。

9.根据 3.6.2 中步骤 5 回答:

(1) 请比较这两种情况有何不同?

- 因为 Ping 的 IP 地址都和 PCA 不是 PCA 不是同一网段,所以 PCA 都会将报文发送给默认网关 S1 的 E0/1。在第一种情况,10.1.3.20 在 S1 的 E0/23 端口的子网内,所以 S1 会将报文发送至 E0/23 将报文发送至 E0/23 端口,而在第二种情况,10.1.4.10 不在 S1 的路由表内,所以 S1 认为该 IP 不可达,从而回复 Destination unreachable。

(2) 截获了哪种 ICMP 差错报文? 其类型和代码字段值是什么? 此报文的 ICMP 协议部分又分为了几部分? 其作用是什么?

- 终点不可达差错报文,类型字段为 3 (Destination unreachable)。此报文的 ICMP 协议部分又包括 Code 为 0 表示网络不可达,即无路由到主机。封装的源 Echo 请求 ICMP 报文的 IP 层和 ICMP 层表示该差错报文来源与一个从 10.1.2.10 到 10.1.4.10 的 Echo 请求。

10.根据 3.6.2 中步骤 6 回答:

- (1) 结合报文内容, 简述 `tracert` 的工作过程。
 - PCA 运行 `tracert` 向目的地址 PCB 发送具有不同生存时间 (TTL) 的 ICMP Echo 请求报文, 在 PCA 到 PCB 路径上的每个路由器都要在转发该 ICMP 报文时将其 TTL 值减 1。当 TTL 值减为 0 时, 路由器就向源主机 PCA 发送 ICMP 超时差错报文。而 PCA 通过向 PCB 发送 TTL 为 1, 2,, n 的 Echo 报文就可以获得从 PCA 到 PCB 的所有路径信息。
- (2) 截获了哪种 ICMP 差错报文? 其类型和代码字段值是什么?
 - 获取了超时报文, 其类型有 Type: 11 (Time-to-live exceeded), Code: 0 (Time to live exceeded in transit), Checksum: 0xf4ff [correct], Unused: 00000000。封装的源 Echo 请求 ICMP 报文的 IP 层和 ICMP 层表示该差错报文来源于一个从 10.1.2.10 到 10.1.3.10 的 Echo 请求。

7	2.621562	169.254.233.9	169.254.255.255	NBNS	92 Name query NB ISATAP<00>
8	3.499079	fe80::505c:de3c:bf1:e909	ff02::1:3	LLMNR	86 Standard query 0xe569 A isatap
9	3.499088	169.254.233.9	224.0.0.252	LLMNR	66 Standard query 0xe569 A isatap
10	3.586656	10.1.2.10	10.1.3.10	ICMP	106 Echo (ping) request id=0x0200, seq=5376/21, ttl=1 (no response found!)
11	3.588212	10.1.2.1	10.1.2.10	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
12	3.588715	10.1.2.10	10.1.3.10	ICMP	106 Echo (ping) request id=0x0200, seq=5632/22, ttl=1 (no response found!)
13	3.589897	10.1.2.1	10.1.2.10	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
14	3.590352	10.1.2.10	10.1.3.10	ICMP	106 Echo (ping) request id=0x0200, seq=5888/23, ttl=1 (no response found!)
15	3.591748	10.1.2.1	10.1.2.10	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
16	3.698875	169.254.233.9	169.254.255.255	NBNS	92 Name query NB ISATAP<00>
17	4.466196	169.254.233.9	169.254.255.255	NBNS	92 Name query NB ISATAP<00>
18	4.586899	10.1.2.10	10.1.3.10	ICMP	106 Echo (ping) request id=0x0200, seq=6144/24, ttl=2 (reply in 19)

Internet Control Message Protocol Type: 11 (Time-to-live exceeded) Code: 0 (Time to live exceeded in transit) Checksum: 0xf4ff [correct] [Checksum Status: Good] Unused: 00000000
Internet Protocol Version 4, Src: 10.1.2.10, Dst: 10.1.3.10 0100 = Version: 4 0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 92 Identification: 0x011f (287) > Flags: 0x0000 > Time to live: 1 Protocol: ICMP (1) Header checksum: 0x9f6d [validation disabled] [Header checksum status: Unverified] Source: 10.1.2.10 Destination: 10.1.3.10 > Internet Control Message Protocol

11 根据 4.6 中步骤 1, 写出 tracert 命令用到了 IP 协议报文的哪几个字段?

- Tracert 命令用到了 IP 协议报文的第 1 字段 (ICMP)

```

▼ Internet Protocol Version 4, Src: 10.1.2.1, Dst: 10.1.2.10
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
    .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 120
  Identification: 0x0013 (19)
  ▼ Flags: 0x0000
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment offset: 0
  Time to live: 255
  Protocol: ICMP (1)
  Header checksum: 0xa365 [validation disabled]
  [Header checksum status: Unverified]
  Source: 10.1.2.1
  Destination: 10.1.2.10

```

12. 根据 4.6 中步骤 2 回答: 观察 PC A 和 PC B 能否 ping 通, 结合截获报文分析原因。

- PC A 和 PC B 不能 ping 通, 当 A (10.1.2.10) 的子网掩码 255.255.0.0 (B 类), 而 B (10.1.3.10) 的子网掩码 255.255.255.0 (C 类) 时。因为他们俩的子网掩码不同于是他们俩不能 ping 通, 首先 PCA ping PCB 时 PCA 认为 PCB 的网段 10.1.0.0 (同一网段), 但 PCB 应答 PCA 时他认为 PCA 的网段 (10.1.0.0) 但 PCB 的网段 (10.1.3.0) 于是 PCB 认为 PCA 不在同一网段, 所以在 PCA 会发现 ICMP 的超时报文 (no response found)。

13. 根据 4.6 中步骤 3 填写下表:

Destination/Mask	Protocol	Pre	Cost	Nextthop	Interface
10.1.2.0/24	Direct	0	0	10.1.2.1	Vlan2
10.1.2.1/32	Direct	0	0	127.0.0.1	InLoop0
10.1.3.0/24	Direct	0	0	10.1.3.1	Vlan3
10.1.3.1/32	Direct	0	0	127.0.0.1	InLoop0
127.0.0.0/8	Direct	0	0	127.0.0.1	InLoop0
127.0.0.1/32	Direct	0	0	127.0.0.1	InLoop0

```

S1 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
<S1>
#Feb 6 05:14:21:20 2010 S1 SHELL/4/LOGIN:
Trap 1.3.6.1.4.1.25506.2.2.1.1.3.0.1<hh3cLogIn>: login from Console
%Feb 6 05:14:21:260 2010 S1 SHELL/5/SHELL_LOGIN: Console logged in from aux0.
<S1>sys
System View: return to User View with Ctrl+Z.
[Sys]ip ttl-expires enable
[Sys]ip unreachable enable
[Sys]dis ip routing-table
Routing Tables: Public
Destinations : 6          Routes : 6

Destination/Mask    Proto Pre  Cost      NextHop          Interface
10.1.2.0/24         Direct 0     0          10.1.2.1          Vlan2
10.1.2.1/32         Direct 0     0          127.0.0.1         InLoop0
10.1.3.0/24         Direct 0     0          10.1.3.1          Vlan3
10.1.3.1/32         Direct 0     0          127.0.0.1         InLoop0
127.0.0.0/8         Direct 0     0          127.0.0.1         InLoop0
127.0.0.1/32        Direct 0     0          127.0.0.1         InLoop0
  
```

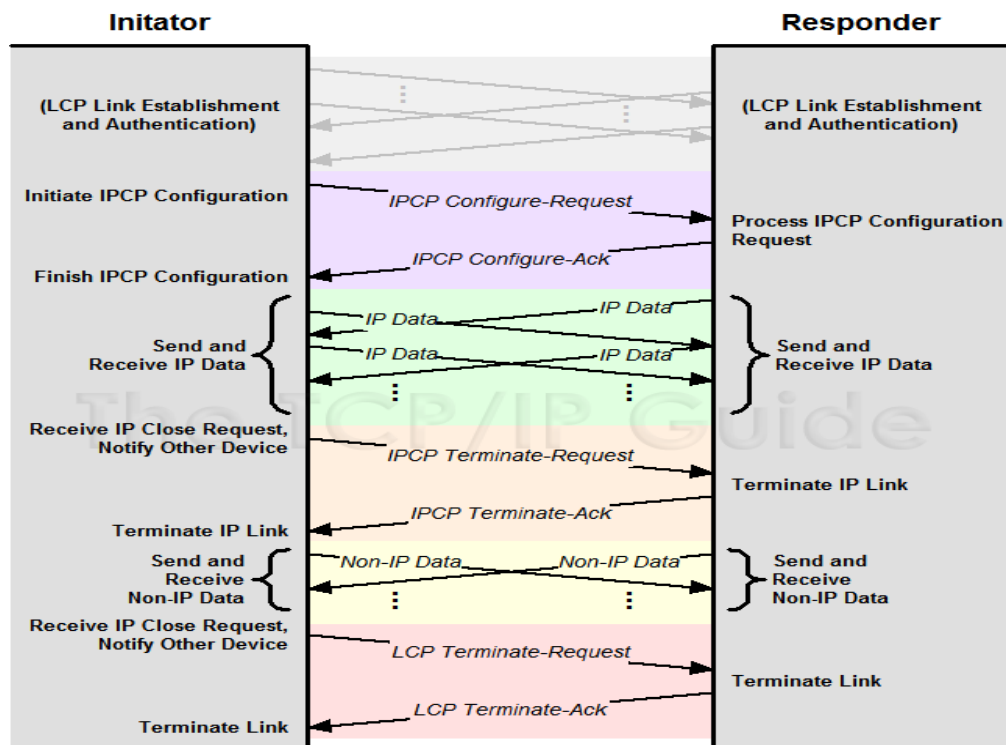
4. (1) 按照实验2的5.5节（PPP协议实验）图17配置路由器，两个路由器相互 ping，看能否 ping 通。根据 R1 上的 debug 显示信息，画出 IPCP 协议在协商过程中的状态转移图（事件驱动、状态转移）。

- 两个路由器相互 ping，能 ping 通

```

State reqsent, code ConfReq(01), id 0, len 10
IP Address(3), len 6, val c0000002
*Apr 14 23:34:25:740 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 IPCP RCR+(Receive Config Good Request) Event
state reqsent
*Apr 14 23:34:25:891 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Output IPCP(8021) Pkt, Len 14

State reqsent, code ConfAck(02), id 0, len 10
IP Address(3), len 6, val c0000002
*Apr 14 23:34:25:992 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 IPCP : reqsent --> acksent
*Apr 14 23:34:26:142 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Input IPCP(8021) Pkt, Len 14
State acksent, code ConfAck(02), id 0, len 10
IP Address(3), len 6, val c0000001
*Apr 14 23:34:26:243 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 IPCP RCA(Receive Config Ack) Event
state acksent
*Apr 14 23:34:26:344 2019 R1 PPP/7/debug2:
PPP State Change:
Serial1/0 IPCP : acksent --> opened
*Apr 14 23:34:26:494 2019 R1 PPP/7/debug2:
PPP Event:
Serial1/0 : Add VLink = 5
%Apr 14 23:34:26:695 2019 R1 IFNET/5/LINEPROTO_UPDOWN: Line protocol on the inte
rface Serial1/0 is UP.
%Apr 14 23:34:26:896 2019 R1 IFNET/5/PROTOCOL_UPDOWN: Protocol PPP IPCP on the i
nterface Serial1/0 is UP.
  
```



(2) 将路由器 R2 的接口 S0/0 的 IP 地址改为 10.0.0.1/24, 两台路由器能否 ping 通? 并解释为什么? 注意体会 IPCP 协议的特点。(查看 IPCP 协议协商过程的 debug 信息)

- 两台路由器能 ping 通, 因为 IPCP 协议的特点是 IPCP 协议中并未规定点对点两端的 IP 地址必须在同一网段。当一端接收到 Config-Request 报文后, 它从报文的配置参数选项中可获知对端的 IP 地址, 但并不与本端的 IP 地址进行比较。因此说点对点通信的两端如果是手动设置每一端的 IP 地址时, 无须双方地址在同一网段。

```

PPP Packet:
Serial1/0 Output IP(0021) Pkt, Len 88
*Apr 14 23:37:19:427 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Input IP(0021) Pkt, Len 88
Reply from 10.0.0.1: bytes=56 Sequence=1 ttl=255 time=26 ms
*Apr 14 23:37:19:629 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Output IP(0021) Pkt, Len 88
*Apr 14 23:37:19:653 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Input IP(0021) Pkt, Len 88
Reply from 10.0.0.1: bytes=56 Sequence=2 ttl=255 time=26 ms
*Apr 14 23:37:19:855 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Output IP(0021) Pkt, Len 88
*Apr 14 23:37:19:879 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Input IP(0021) Pkt, Len 88
Reply from 10.0.0.1: bytes=56 Sequence=3 ttl=255 time=26 ms
*Apr 14 23:37:20:081 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Output IP(0021) Pkt, Len 88
*Apr 14 23:37:20:105 2019 R1 PPP/7/debug2:
PPP Packet:
Serial1/0 Input IP(0021) Pkt, Len 88
Reply from 10.0.0.1: bytes=56 Sequence=4 ttl=255 time=26 ms

--- 10.0.0.1 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 26/26/26 ms

```

5.根据 5.6 中步骤 5:

(1) 在截获报文中, 有 5 个 ARP 报文, 10 个 ICMP:Echo 报文, 5 个 ICMP:Echo Reply 报文, 1 个 IP 报文。

(2) 据 ping 命令执行过程的分析, 将本属于同一个数据报文信息的报文截取出来, 例如下列的报文, 从信息栏中可以看出, 报文 1、2、3、4 属于同一数据段。

```

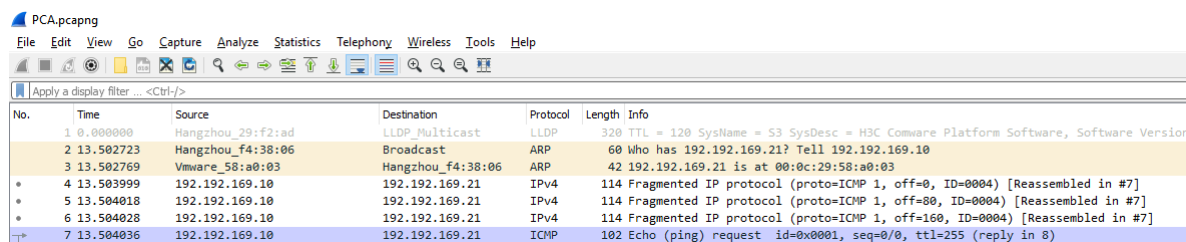
1 C 192.192.169.10 192.192.169.20 ICMP Echo (ping) request
2 C 192.192.169.10 192.192.169.20 IP Fragmented IP protocol (proto=ICMP 0x01, off=80)
3 C 192.192.169.10 192.192.169.20 IP Fragmented IP protocol (proto=ICMP 0x01, off=160)
4 C 192.192.169.10 192.192.169.20 IP Fragmented IP protocol (proto=ICMP 0x01, off=240)

```

将第一个 ICMP Request 的报文分片信息填写下表。

字段名称	分片序号 1	分片序号 2	分片序号 3	分片序号 4
“Identification” 字段值	0x0004 (4)	0x0004 (4)	0x0004 (4)	0x0004 (4)
“Flag” 字段值	0x2000	0x200a	0x2014	0x001e
“Frame offset” 字段值	0	10	20	30
传输的数据量	80 bytes (0 - 79)	80 bytes (80-159)	80 bytes (160-239)	68 bytes (240-307)

分析表格内容，根据 IP 首部字段设置，体会分片过程。



The screenshot shows a Wireshark capture of a network packet. The packet list on the left shows a sequence of events: an LLDMP Multicast packet, an ARP request, and three fragmented IP packets (labeled 4, 5, and 6). The selected packet (7) is an ICMP Echo (ping) request. The packet details pane on the right shows the ICMP header and the payload, which is the Echo (ping) request. The packet bytes pane at the bottom shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Hangzhou_29:f2:ad	LLDP_Multicast	LLDP	320	TTL = 120 SysName = S3 SysDesc = H3C Comware Platform Software, Software Versior
2	13.502723	Hangzhou_f4:38:06	Broadcast	ARP	60	Who has 192.192.169.21? Tell 192.192.169.10
3	13.502769	Vmware_58:a0:03	Hangzhou_f4:38:06	ARP	42	192.192.169.21 is at 00:0c:29:58:a0:03
4	13.503999	192.192.169.10	192.192.169.21	IPv4	114	Fragmented IP protocol (proto=ICMP 1, off=0, ID=0004) [Reassembled in #7]
5	13.504018	192.192.169.10	192.192.169.21	IPv4	114	Fragmented IP protocol (proto=ICMP 1, off=80, ID=0004) [Reassembled in #7]
6	13.504028	192.192.169.10	192.192.169.21	IPv4	114	Fragmented IP protocol (proto=ICMP 1, off=160, ID=0004) [Reassembled in #7]
7	13.504036	192.192.169.10	192.192.169.21	ICMP	102	Echo (ping) request id=0x0001, seq=0/0, ttl=255 (reply in 8)

(3) ping 的数据部分为 300 字节，路由器的以太网端口 MTU 设为 100 字节。回送请求报文为何被分片为 4 片而不是 3 片？数据部分长度为多少时报文正好被分为 3 片？

- 因为假设 300 字节，20 字节是 IP 的头部于是 MTU 的值为 $100 - 20 = 80$ ，每一片的数据部分最多为 80 字节，于是 $300 / 80 = 3.75 \approx 4$ 片。假设报文正好被分为 3 片，数据部分长度必须为 240 字节 ($240 / 80 = 3$ 片)。