以太网帧分析

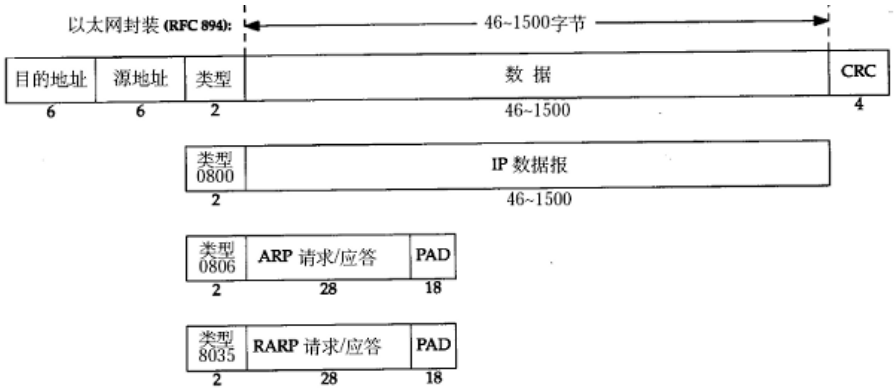
一 实验目的

1. 掌握以太网帧的报文格式
2. 掌握网络协议分析软件的基本使用方法

二 预备知识

数据链路层的传输单位为帧（Frame），在发送端数据链路层将网络层的数据按照一定格式打包为帧并发送给物理层，在接收端数据链路层将物理层的数据按照一定格式解包为帧并发送给网络层。

目前，在数据链路层使用比较多的是以太网（Ethernet）协议。以太网帧格式如下：

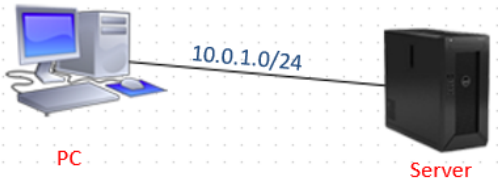


各字段含义如下：

1. **目的 Mac 地址**：下一跳的 Mac 地址，帧每经过一跳（即每经过一台网络设备如交换机）该地址会被替换，直到最后一跳被替换为接收端的 Mac 地址。
2. **源 Mac 地址**：发送端 Mac 地址。
3. **类型**：用来指出以太网帧内所含的上层协议。对于 IP 报文来说，该字段值是 0x0800；对于 ARP 请求/应答报文，字段的值是 0x0806；RARP 请求/应答报文对应值为 0x8035。
4. **数据**：从上层或下层传来的有效数据，如果少于 46 个字节，会填充到 46 个字节。
5. **校验码**：CRC 校验码，校验该帧在传输过程中是否出错。

三 实验环境

在右上方的实验拓扑图中选择“**以太网帧分析**”，实验拓扑及网络子网网段（10.0.2.0/24）如图所示：



子网网段可随意设置，设备所获取的 ip 地址可能会有所不同，实际进行实验时请进入虚拟机进行查看。本实验中各个设备 ip 地址为：

PC：10.0.1.10　　Server：10.0.1.6

设置完子网网段后点击“提交实验”按钮，耐心等待资源分配，资源分配中请不要进行其他操作。

四 实验内容

1. 进入虚拟机

资源分配成功后，点击图标在弹出窗口中点击“全屏访问”即可进入虚拟机，centos 系统的登录用户名密码均为：**centos**。

2. 查看 IP 地址

进入 Server，使用 **ifconfig** 命令可以查看到本机 ip 地址为 10.0.1.6。

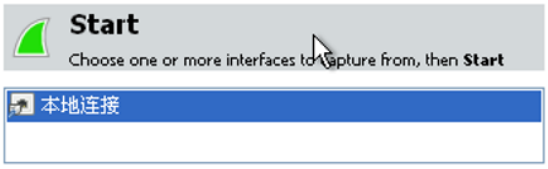
```
[root@centos7 centos]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1450
        inet 10.0.1.6  netmask 255.255.255.0  broadcast 10.0.1.255
        inet6 fe80::f816:3eff:fe5c:800c  prefixlen 64  scopeid 0x20<link>
        ether fa:16:3e:5c:80:0c  txqueuelen 1000  (Ethernet)
        RX packets 156  bytes 19356 (18.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 222  bytes 21651 (21.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

进入 PC，打开命令提示符，使用 **ipconfig** 查看 IP 地址。

```
C:\Documents and Settings\admin>ipconfig

Windows IP Configuration


Ethernet adapter 本地连接:

        Connection-specific DNS Suffix  . : openstacklocal
        IP Address. . . . . . . . . . . . : 10.0.1.10
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.0.1.1
```

3. 抓取以太网帧数据包

然后进入 PC，打开桌面的 wireshark 程序，然后选择“本地连接”网卡 ，然后单击”Start”按钮开始抓包：



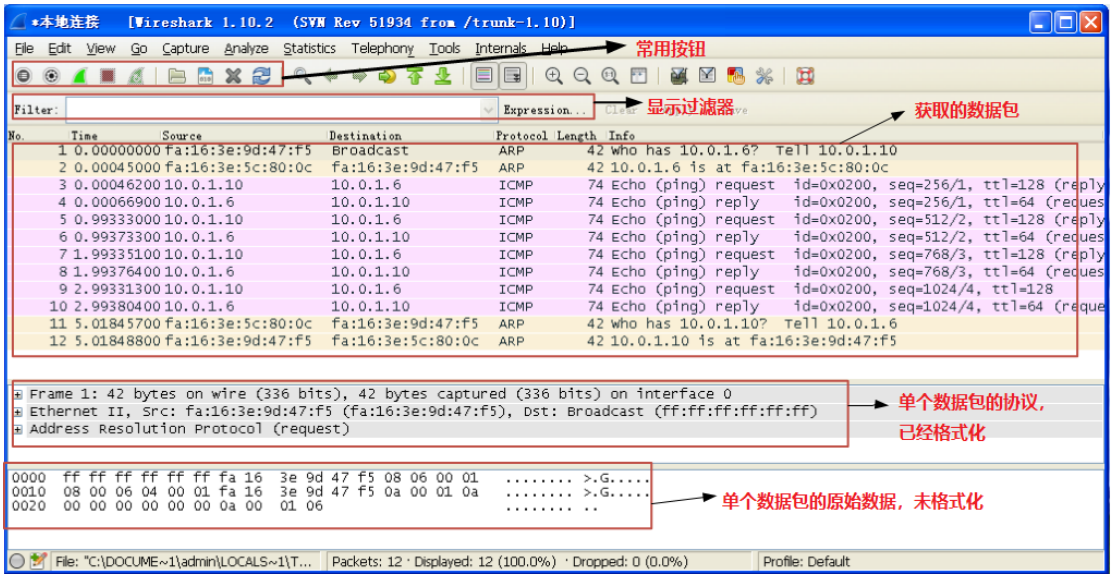
接下来打开命令提示符，输入 **ping 10.0.1.6**，然后查看抓包信息。

1	0.00000000	fa:16:3e:9d:47:f5	Broadcast	ARP	42 Who has 10.0.1.6? Tell 10.0.1.10
2	0.00045000	fa:16:3e:5c:80:0c	fa:16:3e:9d:47:f5	ARP	42 10.0.1.6 is at fa:16:3e:5c:80:0c
3	0.00046200	10.0.1.10	10.0.1.6	ICMP	74 Echo (ping) request id=0x0200, seq=256/1, ttl=128 (reply
4	0.00066900	10.0.1.6	10.0.1.10	ICMP	74 Echo (ping) reply id=0x0200, seq=256/1, ttl=64 (request
5	0.99333000	10.0.1.10	10.0.1.6	ICMP	74 Echo (ping) request id=0x0200, seq=512/2, ttl=128 (reply
6	0.99373300	10.0.1.6	10.0.1.10	ICMP	74 Echo (ping) reply id=0x0200, seq=512/2, ttl=64 (request
7	1.99335100	10.0.1.10	10.0.1.6	ICMP	74 Echo (ping) request id=0x0200, seq=768/3, ttl=128 (reply
8	1.99376400	10.0.1.6	10.0.1.10	ICMP	74 Echo (ping) reply id=0x0200, seq=768/3, ttl=64 (request
9	2.99331300	10.0.1.10	10.0.1.6	ICMP	74 Echo (ping) request id=0x0200, seq=1024/4, ttl=128
10	2.99380400	10.0.1.6	10.0.1.10	ICMP	74 Echo (ping) reply id=0x0200, seq=1024/4, ttl=64 (reques

4. 数据包分析

1）Wireshark 的使用

分析数据包之前我们先来认识一下 Wireshark，比较常用的功能如下图所示：



其中常用按钮从左到右的功能依次是：

1. 列出可用接口。
2. 抓包时需要设置的一些选项。一般会保留最后一次的设置结果。
3. 开始新的一次抓包。
4. 暂停抓包。
5. 继续进行本次抓包。
6. 打开抓包文件。可以打开之前抓包保存后的文件。不仅可以打开 wireshark 软件保存的文件，也可以打开 tcpdump 使用-w 参数保存的文件。
7. 保存文件。把本次抓包或者分析的结果进行保存。
8. 关闭打开的文件。文件被关闭后，就会切换到初始界面。
9. 重载抓包文件。

2）分析数据包

首先我们了解到IP地址只在网络层可以识别，数据链路层进行通信需要的是MAC地址，并不认识IP。于是PC会先发出一个广播，询问谁的IP地址是10.0.1.6？

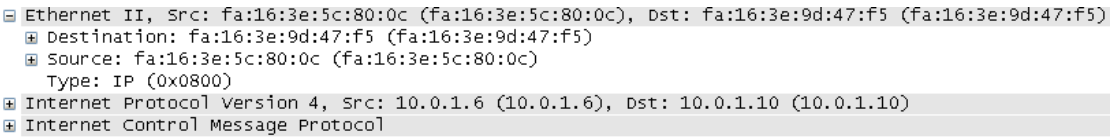
```
Ethernet II, Src: fa:16:3e:9d:47:f5 (fa:16:3e:9d:47:f5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: fa:16:3e:9d:47:f5 (fa:16:3e:9d:47:f5)
  Type: ARP (0x0806)
```

从帧的头部信息可以看出，该帧在数据链路层使用的是 Ethernet II 协议，目的地址为Broadcast(ff:ff:ff:ff:ff:ff)，这是16进制的广播地址，即向整个链路上的机器询问谁的IP地址为10.0.1.6；源地址字段的值为fa:16:3e:9d:47:f5，告诉接收方本机的Mac地址；类型字段Type为ARP(0x0806)，表明了帧的上层协议为ARP，如下图所示：

```
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: fa:16:3e:9d:47:f5 (fa:16:3e:9d:47:f5)
    Sender IP address: 10.0.1.10 (10.0.1.10)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.0.1.6 (10.0.1.6)
```

该数据包是一个 ARP 请求报文，各个字段的值给出了请求主机和目的主机的相关信息。

PC 通过 ARP 协议获取到 Server 的 MAC 地址之后，便开始向 Server 正常发送 ICMP 数据包，对应的以太网帧内容如下：



帧首部的 Type 字段(0x0800)指明了上层协议为 IP，以太网帧的数据区域即为一个 IP 数据包，而 ICMP 又是基于 IP 协议的。我们所使用的 ping 命令正是基于 ICMP 来实现的。