

华东师范大学数据科学与工程学院实验报告

课程名称：计算机网络与编程

年级：2021

上机实践成绩：

指导教师：张召

姓名：温兆和

学号：10205501432

上机实践名称：UDP 和 TCP 协议分析

上机实践日期：2022.05.20

上机实践编号：13

组号：001-432

上机实践时间：13:00

一、实验目的

了解 UDP 协议的工作原理；

了解 TCP 协议的工作原理；

学习 TCP 建立连接三次握手的过程；

学习 TCP 断开连接四次挥手的过程。

二、实验任务

使用 Wireshark 快速了解 UDP 协议；

使用 Wireshark 快速了解 TCP 协议。

三、使用环境

Wireshark

四、实验过程

Task1. 从跟踪中选择一个 UDP 数据包。从此数据包中，识别并确定 UDP 首部字段，请为这些字段命名并将实验结果附在实验报告中。

Wireshark network traffic capture showing a UDP packet. The packet list shows a UDP packet from 192.168.0.1 to 255.255.255.255. The packet details show the UDP header fields: Source Port (1024), Destination Port (5001), Length (15), and Checksum (0xf2c1). The packet payload is a User Datagram Protocol (UDP) packet. Red handwritten labels are added to the packet details: '源端口号' (Source Port) for Source Port, '目的端口号' (Destination Port) for Destination Port, '长度' (Length) for Length, '校验和' (Checksum) for Checksum, and '应用层报文' (Application Layer Payload) for the Data field.

Task2. UDP 首部中的长度字段指的是什么，以及为什么需要这样设计？使用捕获的 UDP 数据包进行验证，请将实验结果附在实验报告中。

指的是数据字段长度加上首部字段长度，在上例中是 $143+8=151$ 。

这样设计是因为不同 UDP 网络包中数据字段长度不同，所以需要有一个明确的长度。

Task3. UDP 有效负载中可包含的最大字节数是多少？请将实验结果附在实验报告中。

总长度：65536bit

首部字段：8bit

应用层报文中可容纳的最大长度： $65536-8=65528$ bit

Task4. 观察发送 UDP 数据包后接收响应的 UDP 数据包，这是对发送的 UDP 数据包的回复，请描述两个数据包中端口号之间的联系。请将实验结果附在实验报告中。

上面抓到的 UDP 数据包并没有响应数据包，所以我们抓取一个 DNS 数据包并找到它的响应报文，观察源端口号和目的端口号之间的关系。DNS 使用 UDP。

正在捕获 WLAN

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)

udp

No.	Time	Source	Destination	Protocol	Length	Info
522	158.607151	192.168.0.1	255.255.255.255	UDP	185	1024 → 5001 Len=143
639	164.732316	192.168.0.107	192.168.1.1	DNS	73	Standard query 0x4d30 A mmbiz.qpic.cn
641	164.757489	192.168.1.1	192.168.0.107	DNS	424	Standard query response 0x4d30 A mmbiz.qpic.cn CNA
1247	166.995460	192.168.0.107	192.168.1.1	DNS	76	Standard query 0x82f6 A mp.weixin.qq.com
1248	167.004329	192.168.1.1	192.168.0.107	DNS	128	Standard query response 0x82f6 A mp.weixin.qq.com
1288	167.388855	192.168.0.107	192.168.1.1	DNS	73	Standard query 0xb28e A res.wx.qq.com
1289	167.398279	192.168.1.1	192.168.0.107	DNS	457	Standard query response 0xb28e A res.wx.qq.com CNA
1958	168.045484	192.168.0.107	192.168.1.1	DNS	82	Standard query 0x3017 A badjs.weixinbridge.com
1959	168.060168	192.168.1.1	192.168.0.107	DNS	98	Standard query response 0x3017 A badjs.weixinbridge.com
1006	173.847500	192.168.0.107	192.168.1.1	DNS	73	Standard query 0xb094 A mmbiz.qpic.cn

Header Checksum: 0x0000 [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.0.107
 Destination Address: 192.168.1.1

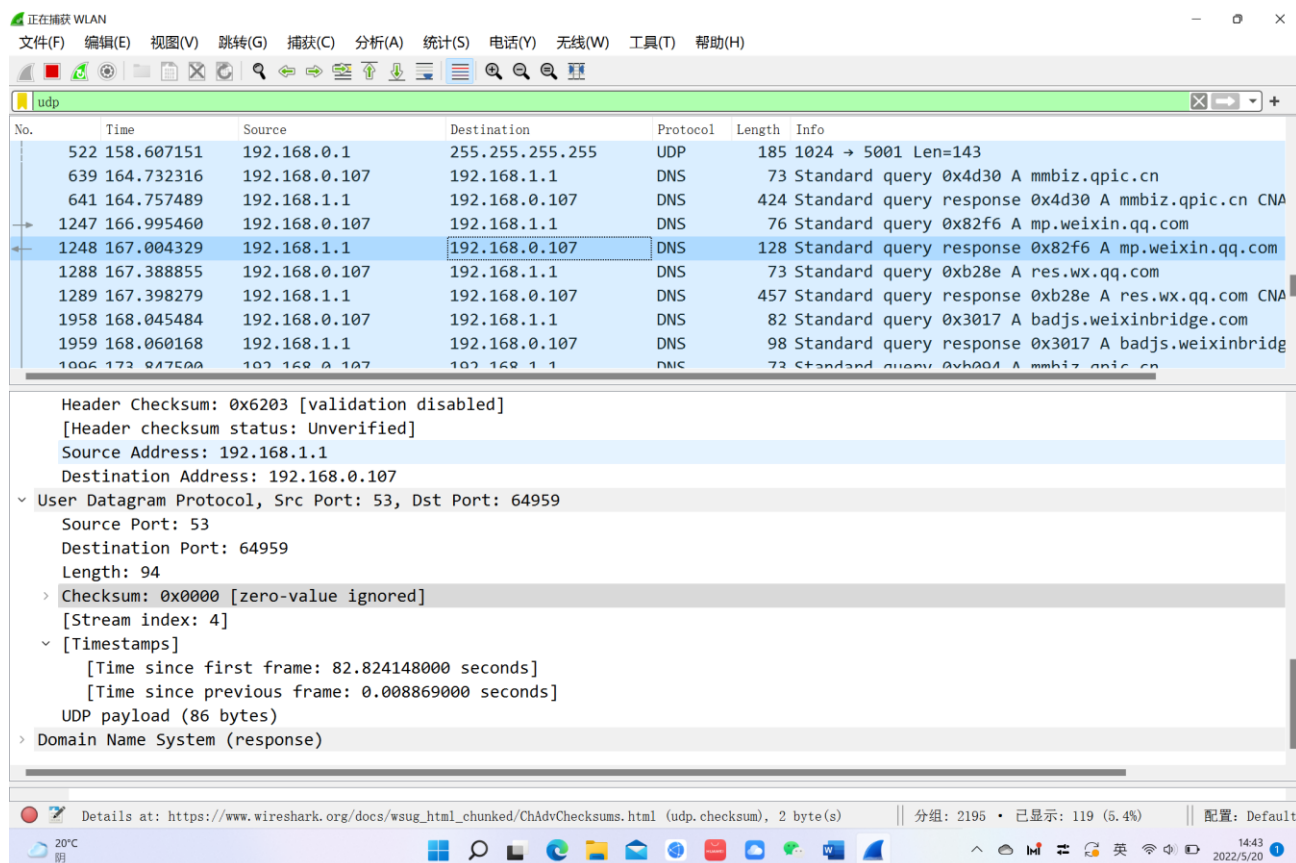
▼ User Datagram Protocol, Src Port: 64959, Dst Port: 53
 Source Port: 64959
 Destination Port: 53
 Length: 42
 Checksum: 0x82f8 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 4]

▼ [Timestamps]
 [Time since first frame: 82.815279000 seconds]
 [Time since previous frame: 2.237971000 seconds]

UDP payload (34 bytes)
 Domain Name System (query)

Details at: https://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html (udp.checksum), 2 byte(s) | 分组: 2143 · 已显示: 107 (5.0%) | 配置: Default

20°C 阴 14:43 2022/5/20



我们可以看到，在 UDP 协议中，响应数据包的目的端口号是原数据包的源端口号，源端口号是原数据包的目的端口号。

Task5. 利用 Wireshark 抓取一个 TCP 数据包，查看其具体数据结构和实际的数据（要求根据报文结构正确标识每个部分），请将实验结果附在实验报告中。

源端口号
目的端口号
序号
确认号
首部长度
标志字段
接收窗口
检验和
紧急数据指针
选项
数据

Task6. 根据 TCP 三次握手的交互图以及 TCP 报文段结构图逐步分析三次握手过程，请将实验结果附在实验报告中。

第一次握手：客户端发送 SYN 报文

Flags: 0x002 (SYN)
Window: 64240
[Calculated window size: 64240]
Checksum: 0xe849 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK per
[Timestamps]
[Time since first frame in this TCP stream: 0.000000000 seconds]
[Time since previous frame in this TCP stream: 0.000000000 seconds]

第二次握手：服务端发送 SYN 和 ACK 报文

应用显示过滤器: `<Ctrl-/>`

No.	Time	Source	Destination	Protocol	Length	Info
531	8.910520	192.168.1.1	192.168.0.107	DNS	91	Standard query response 0x927e A idc.ecnu.edu.cn A
532	8.911639	192.168.0.107	202.120.92.151	TCP	66	61728 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
533	8.933597	202.89.233.100	192.168.0.107	TCP	54	443 → 61708 [ACK] Seq=9917 Ack=14563 Win=4193536 L

[TCP Segment Len: 0]
 Sequence Number: 0 (relative sequence number)
 Sequence Number (raw): 428471850
 [Next Sequence Number: 1 (relative sequence number)]
 Acknowledgment Number: 1 (relative ack number)
 Acknowledgment number (raw): 1513010170
 1000 = Header Length: 32 bytes (8)
 > Flags: 0x012 (SYN, ACK)
 Window: 29200
 [Calculated window size: 29200]
 Checksum: 0x05cf [unverified]
 [Checksum Status: Unverified]
 Urgent Pointer: 0
 > Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Windo
 > [Timestamps]
 [Time since first frame in this TCP stream: 0.021958000 seconds]
 [Time since previous frame in this TCP stream: 0.021958000 seconds]
 > [SEQ/ACK analysis]
 [This is an ACK to the segment in frame: 532]
 [The RTT to ACK the segment was: 0.021958000 seconds]
 [iRTT: 0.022102000 seconds]

Time delta from previous frame in this TCP stream (tcp.time_delta) | 分组: 4454 • 已显示: 4454 (100.0%) | 配置: Default

第三次握手：客户端发送 ACK 报文

应用显示过滤器: `<Ctrl-/>`

No.	Time	Source	Destination	Protocol	Length	Info
531	8.910520	192.168.1.1	192.168.0.107	DNS	91	Standard query response 0x927e A idc.ecnu.edu.cn A
532	8.911639	192.168.0.107	202.120.92.151	TCP	66	61728 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
533	8.933597	202.89.233.100	192.168.0.107	TCP	54	443 → 61708 [ACK] Seq=9917 Ack=14563 Win=4193536 L
534	8.933597	202.120.92.151	192.168.0.107	TCP	66	80 → 61728 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
535	8.933741	192.168.0.107	202.120.92.151	TCP	54	61728 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
536	8.933993	192.168.0.107	202.120.92.151	HTTP	547	GET / HTTP/1.1
537	8.945061	202.120.92.151	192.168.0.107	TCP	54	80 → 61728 [ACK] Seq=1 Ack=494 Win=30464 Len=0
538	8.945130	202.120.92.151	192.168.0.107	HTTP	391	HTTP/1.1 302 Moved Temporarily (text/html)
539	8.946859	192.168.0.107	192.168.1.1	DNS	82	Standard query 0x73bb A portal2020.ecnu.edu.cn
540	8.953258	40.79.189.59	192.168.0.107	TCP	54	443 → 61727 [ACK] Seq=6361 Ack=16916 Win=524800 Le

0101 = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)
 Window: 512
 [Calculated window size: 131072]
 [Window size scaling factor: 256]
 Checksum: 0xe83d [unverified]
 [Checksum Status: Unverified]
 Urgent Pointer: 0
 > [Timestamps]
 [Time since first frame in this TCP stream: 0.022102000 seconds]
 [Time since previous frame in this TCP stream: 0.000144000 seconds]
 > [SEQ/ACK analysis]
 [This is an ACK to the segment in frame: 534]
 [The RTT to ACK the segment was: 0.000144000 seconds]
 [iRTT: 0.022102000 seconds]

Time delta from previous frame in this TCP stream (tcp.time_delta) | 分组: 4548 • 已显示: 4548 (100.0%) | 配置: Default

Task7. 根据 TCP 四次挥手的交互图以及 TCP 报文段结构图逐步分析四次挥手过程，请

将实验结果附在实验报告中。

第一次挥手：主动关闭方发送 FIN

The screenshot shows a Wireshark packet capture of a network session. The packet list at the top shows a sequence of packets. Packet 1716 is selected, which is a TCP packet from 59.36.89.180 to 192.168.0.107, port 80 to 61964, with the FIN flag set. The packet details pane shows the TCP flags as 0x011 (FIN, ACK). The packet bytes pane shows the urgent flag set.

No.	Time	Source	Destination	Protocol	Length	Info
1711	81.538618	59.36.89.180	192.168.0.107	TCP	54	80 → 61964 [ACK] Seq=1 Ack=706 Win=15872 Len=0
1712	81.566324	192.168.0.107	204.79.197.203	TLSv1.2	312	Application Data
1713	81.567824	192.168.0.107	204.79.197.203	TLSv1.2	334	Application Data
1714	81.568158	192.168.0.107	204.79.197.203	TLSv1.2	258	Application Data
1715	81.568619	59.36.89.180	192.168.0.107	HTTP	592	HTTP/1.1 200 OK
1716	81.568619	59.36.89.180	192.168.0.107	TCP	54	80 → 61964 [FIN, ACK] Seq=539 Ack=706 Win=15872 Len=0
1717	81.568724	192.168.0.107	59.36.89.180	TCP	54	61964 → 80 [ACK] Seq=706 Ack=540 Win=130816 Len=0
1718	81.568990	192.168.0.107	204.79.197.203	TLSv1.2	257	Application Data
1719	81.569600	192.168.0.107	59.36.89.180	TCP	54	61964 → 80 [FIN, ACK] Seq=706 Ack=540 Win=130816 Len=0
1720	81.570380	192.168.0.107	204.79.197.203	TLSv1.2	255	Application Data

Flags: 0x011 (FIN, ACK)

- 000. = Reserved: Not set
- ...0 = Nonce: Not set
- 0... = Congestion Window Reduced (CWR): Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-1 = Acknowledgment: Set
- 0... = Push: Not set
-0.. = Reset: Not set
-0. = Syn: Not set
- >1 = Fin: Set
- > [TCP Flags:A...F]

Window: 124
[Calculated window size: 15872]
[Window size scaling factor: 128]

Urgent (tcp.flags.urg), 1 byte(s)

分组: 4921 · 已显示: 4921 (100.0%) 配置: Default

20°C 阴 15:53 2022/5/20

第二次挥手：被动关闭方发送 ACK

The screenshot shows a Wireshark packet capture of a network session. The packet list at the top shows a sequence of packets. Packet 1717 is selected, which is a TCP packet from 192.168.0.107 to 59.36.89.180, port 61964 to 80, with the ACK flag set. The packet details pane shows the TCP flags as 0x010 (ACK). The packet bytes pane shows the urgent flag set.

No.	Time	Source	Destination	Protocol	Length	Info
1711	81.538618	59.36.89.180	192.168.0.107	TCP	54	80 → 61964 [ACK] Seq=1 Ack=706 Win=15872 Len=0
1712	81.566324	192.168.0.107	204.79.197.203	TLSv1.2	312	Application Data
1713	81.567824	192.168.0.107	204.79.197.203	TLSv1.2	334	Application Data
1714	81.568158	192.168.0.107	204.79.197.203	TLSv1.2	258	Application Data
1715	81.568619	59.36.89.180	192.168.0.107	HTTP	592	HTTP/1.1 200 OK
1716	81.568619	59.36.89.180	192.168.0.107	TCP	54	80 → 61964 [FIN, ACK] Seq=539 Ack=706 Win=15872 Len=0
1717	81.568724	192.168.0.107	59.36.89.180	TCP	54	61964 → 80 [ACK] Seq=706 Ack=540 Win=130816 Len=0
1718	81.568990	192.168.0.107	204.79.197.203	TLSv1.2	257	Application Data
1719	81.569600	192.168.0.107	59.36.89.180	TCP	54	61964 → 80 [FIN, ACK] Seq=706 Ack=540 Win=130816 Len=0
1720	81.570380	192.168.0.107	204.79.197.203	TLSv1.2	255	Application Data

Flags: 0x010 (ACK)

- 000. = Reserved: Not set
- ...0 = Nonce: Not set
- 0... = Congestion Window Reduced (CWR): Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-1 = Acknowledgment: Set
- 0... = Push: Not set
-0.. = Reset: Not set
-0. = Syn: Not set
-0 = Fin: Not set
- [TCP Flags:A....]

Window: 511
[Calculated window size: 130816]
[Window size scaling factor: 256]

Urgent (tcp.flags.urg), 1 byte(s)

分组: 4981 · 已显示: 4981 (100.0%) 配置: Default

20°C 阴 15:54 2022/5/20

第三次挥手：被动关闭方发送 FIN

WLAN

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)

应用显示过滤器 ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1711	81.538618	59.36.89.180	192.168.0.107	TCP	54	80 → 61964 [ACK] Seq=1 Ack=706 Win=15872 Len=0
1712	81.566324	192.168.0.107	204.79.197.203	TLSv1.2	312	Application Data
1713	81.567824	192.168.0.107	204.79.197.203	TLSv1.2	334	Application Data
1714	81.568158	192.168.0.107	204.79.197.203	TLSv1.2	258	Application Data
1715	81.568619	59.36.89.180	192.168.0.107	HTTP	592	HTTP/1.1 200 OK
1716	81.568619	59.36.89.180	192.168.0.107	TCP	54	80 → 61964 [FIN, ACK] Seq=539 Ack=706 Win=15872 Len=0
1717	81.568724	192.168.0.107	59.36.89.180	TCP	54	61964 → 80 [ACK] Seq=706 Ack=540 Win=130816 Len=0
1718	81.568990	192.168.0.107	204.79.197.203	TLSv1.2	257	Application Data
1719	81.569600	192.168.0.107	59.36.89.180	TCP	54	61964 → 80 [FIN, ACK] Seq=706 Ack=540 Win=130816 Len=0
1720	81.570380	192.168.0.107	204.79.197.203	TLSv1.2	255	Application Data

Flags: 0x011 (FIN, ACK)

- 000. = Reserved: Not set
- ...0 = Nonce: Not set
- 0... = Congestion Window Reduced (CWR): Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-1 = Acknowledgment: Set
- 0... = Push: Not set
-0.. = Reset: Not set
-0. = Syn: Not set
- >1 = Fin: Set
- > [TCP Flags:A...F]

Window: 511
[Calculated window size: 130816]
[Window size scaling factor: 256]

Urgent (tcp.flags.urg), 1 byte(s)

分组: 5034 • 已显示: 5034 (100.0%) 配置: Default

20°C 阴

15:54 2022/5/20

第四次挥手：主动关闭方发送 ACK，连接断开

WLAN

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)

应用显示过滤器 ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1714	81.568158	192.168.0.107	204.79.197.203	TLSv1.2	258	Application Data
1715	81.568619	59.36.89.180	192.168.0.107	HTTP	592	HTTP/1.1 200 OK
1716	81.568619	59.36.89.180	192.168.0.107	TCP	54	80 → 61964 [FIN, ACK] Seq=539 Ack=706 Win=15872 Len=0
1717	81.568724	192.168.0.107	59.36.89.180	TCP	54	61964 → 80 [ACK] Seq=706 Ack=540 Win=130816 Len=0
1718	81.568990	192.168.0.107	204.79.197.203	TLSv1.2	257	Application Data
1719	81.569600	192.168.0.107	59.36.89.180	TCP	54	61964 → 80 [FIN, ACK] Seq=706 Ack=540 Win=130816 Len=0
1720	81.570380	192.168.0.107	204.79.197.203	TLSv1.2	255	Application Data
1721	81.586842	52.182.143.210	192.168.0.107	TCP	1454	[TCP Previous segment not captured] 443 → 61963 [P
1722	81.586918	192.168.0.107	52.182.143.210	TCP	66	[TCP Dup ACK 1562#1] 61963 → 443 [ACK] Seq=518 Ack
1723	81.601982	59.36.89.180	192.168.0.107	TCP	54	80 → 61964 [RST] Seq=540 Win=0 Len=0

0101 = Header Length: 20 bytes (5)

Flags: 0x004 (RST)

- 000. = Reserved: Not set
- ...0 = Nonce: Not set
- 0... = Congestion Window Reduced (CWR): Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-0 = Acknowledgment: Not set
- 0... = Push: Not set
- >1.. = Reset: Set
-0. = Syn: Not set
-0 = Fin: Not set
- [TCP Flags:R..]

Window: 0
[Calculated window size: 0]

Urgent (tcp.flags.urg), 1 byte(s)

分组: 5228 • 已显示: 5228 (100.0%) 配置: Default

20°C 阴

15:56 2022/5/20

五、总结

在本周的实验中，我们使用 Wireshark 快速了解了 TCP 和 UDP 协议，为以后的学习打好了基础。