

比較了好久，還是這篇文章把 Wireshark 使用技巧講活了！

[51CTO 網工運維之家](https://zhuanlan.zhihu.com/p/391807271)

<https://zhuanlan.zhihu.com/p/391807271>

39 人贊同了該文章

Wireshark 是非常流行的網絡封包分析軟件，可以截取各種網絡數據包，並顯示數據包詳細信息。常用於開發測試過程各種問題定位。本文主要內容包括：

- 1、Wireshark 軟件下載和安裝以及 Wireshark 主界面介紹。
- 2、Wireshark 簡單抓包示例。通過該例子學會怎麼抓包以及如何簡單查看分析數據包內容。
- 3、Wireshark 過濾器使用。過濾器包含兩種類型，一種是抓包過濾器，就是抓取前設置過濾規則。另外一種是顯示過濾器，就是在數據包分析時進行過濾數據使用。通過過濾器可以篩選出想要分析的內容。包括按照協議過濾、端口和主機名過濾、數據包內容過濾。具體規則和實例可以查看正文。

## Wireshark 軟件安裝

軟件下載路徑：

<https://www.wireshark.org/>

按照系統版本選擇下載，下載完成後，按照軟件提示一路 Next 安裝。

## Wireshark 開始抓包示例

先介紹一個使用 wireshark 工具抓取 ping 命令操作的示例，讓讀者可以先上手操作感受一下抓包的具體過程。

1、打開 wireshark 2.6.5，主界面如下：



2、選擇菜單欄上 Capture -> Option，勾選 WLAN 網卡（這裡需要根據各自電腦網卡使用情況選擇，簡單的辦法可以看使用的 IP 對應的網卡）。點擊 Start。啟動抓包。

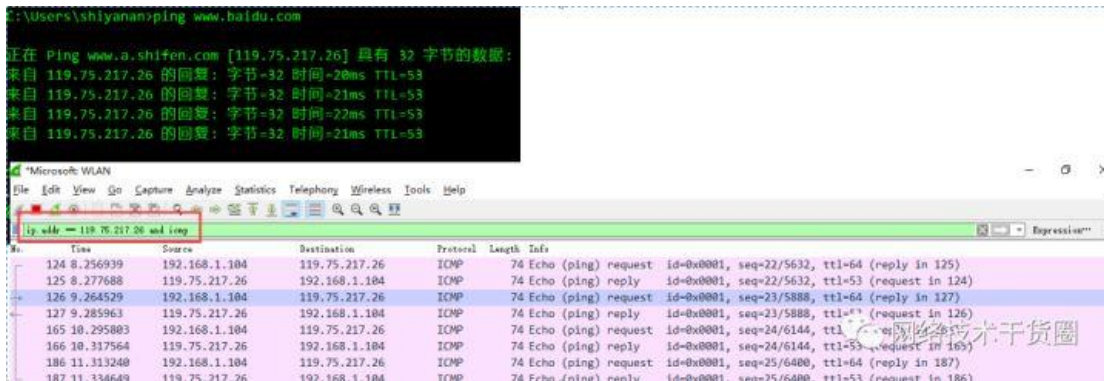
3、wireshark 啟動後，wireshark 處於抓包狀態中。

4、執行需要抓包的操作，如在 cmd 窗口下執行 '\$ ping baidu.com'。

5、操作完成後相關數據包就抓取到了。為避免其他無用的數據包影響分析，可以通過在過濾欄設置過濾條件進行數據包列表過濾，獲取結果如下。

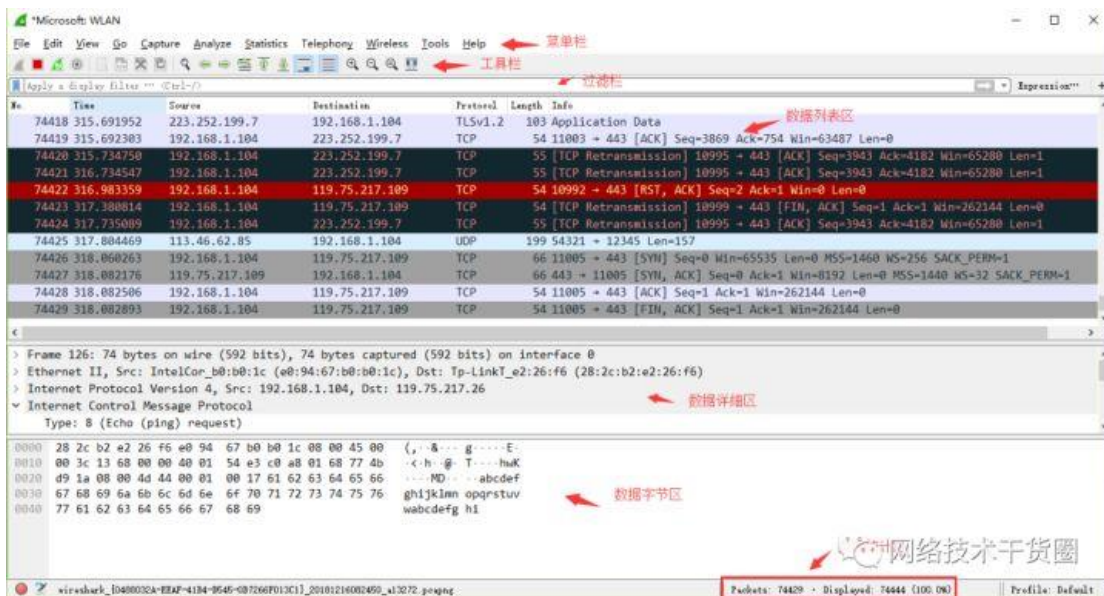
說明：`ip.addr == 119.75.217.26 and icmp` 表示只顯示 ICMP 協議且源主機 IP 或者目的主機 IP 為 119.75.217.26 的數據包。

“說明：協議名稱 icmp 要小寫。”

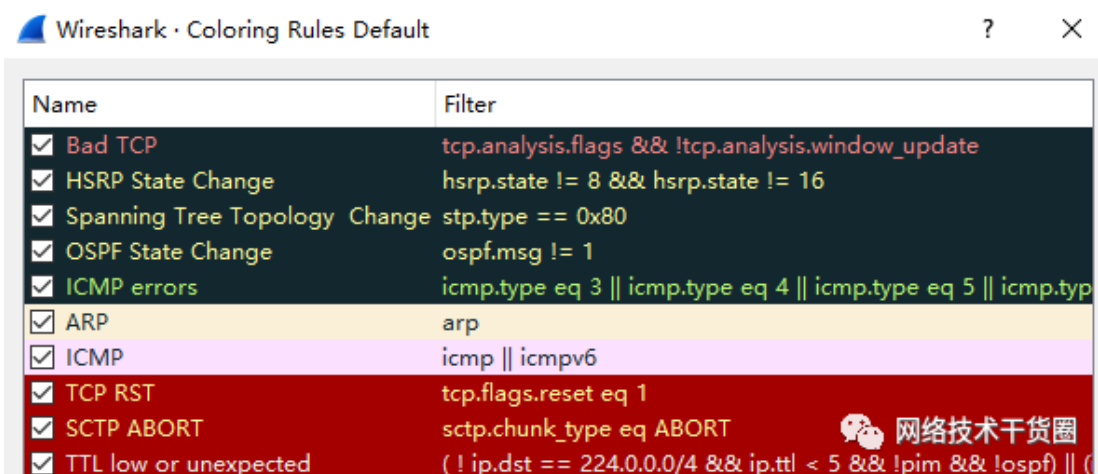


6、wireshark 抓包完成，就這麼簡單。關於 wireshark 顯示過濾條件、抓包過濾條件、以及如何查看數據包中的詳細內容在後面介紹。

## Wireshark 抓包界面介紹



說明：數據包列表區中不同的協議使用了不同的顏色區分。協議顏色標識定位在菜單欄 View --> Coloring Rules。如下所示



## WireShark 主要分為這幾個界面

1. Display Filter(顯示過濾器), 用於設置過濾條件進行數據包列表過濾。

菜單路徑: Analyze --> Display Filters.

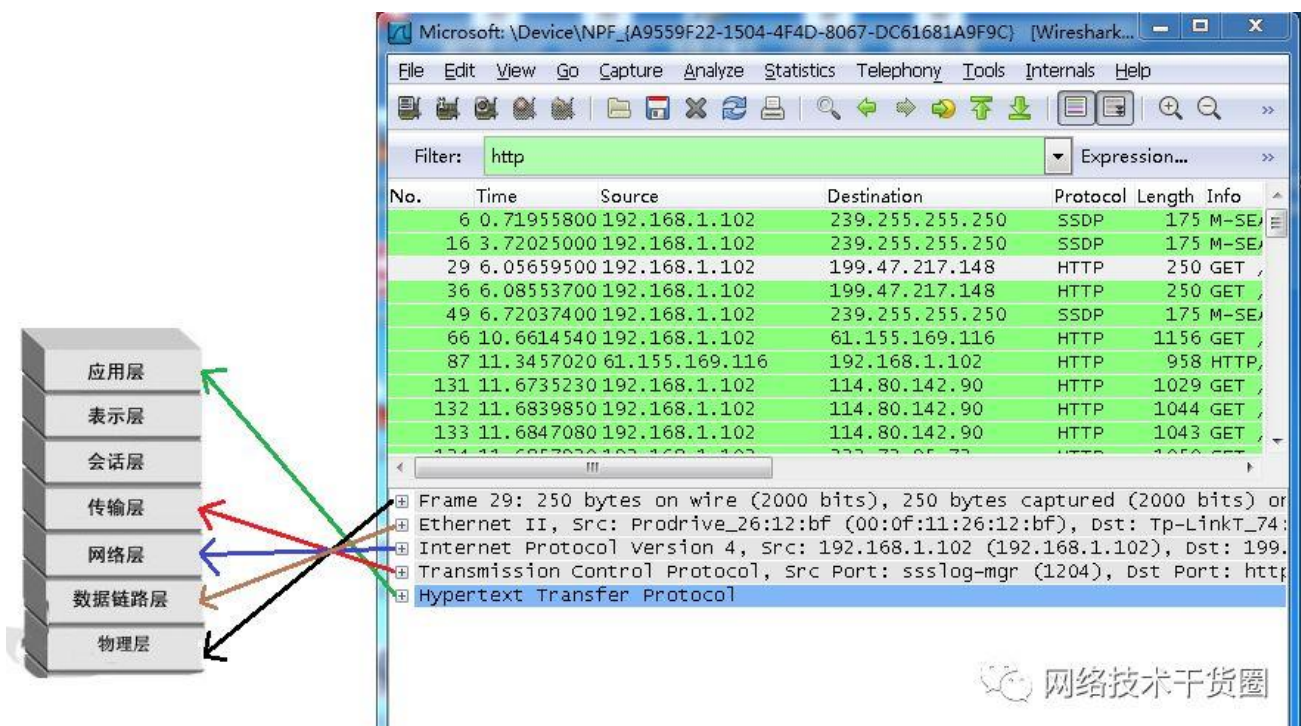


1. Packet List Pane(數據包列表), 顯示捕獲到的數據包, 每個數據包包含編號, 時間戳, 源地址, 目標地址, 協議, 長度, 以及數據包信息。不同協議的數據包使用了不同的顏色區分顯示。

No.	Time	Source	Destination	Protocol	Length	Info
2577	2018-12-1...	192.168.1.104	223.252.199.7	TCP	54	4496 → 443 [ACK] Seq=3869 Ack=754
2578	2018-12-1...	192.168.1.104	59.111.181.155	TCP	590	4263 → 80 [ACK] Seq=3869 Ack=223
2579	2018-12-1...	192.168.1.104	59.111.181.155	TLSv1.2	236	Application Data

1. Packet Details Pane(數據包詳細信息)，在數據包列表中選擇指定數據包，在數據包詳細信息中會顯示數據包的所有詳細信息內容。數據包詳細信息面板是最重要的，用來查看協議中的每一個字段。各行信息分別為

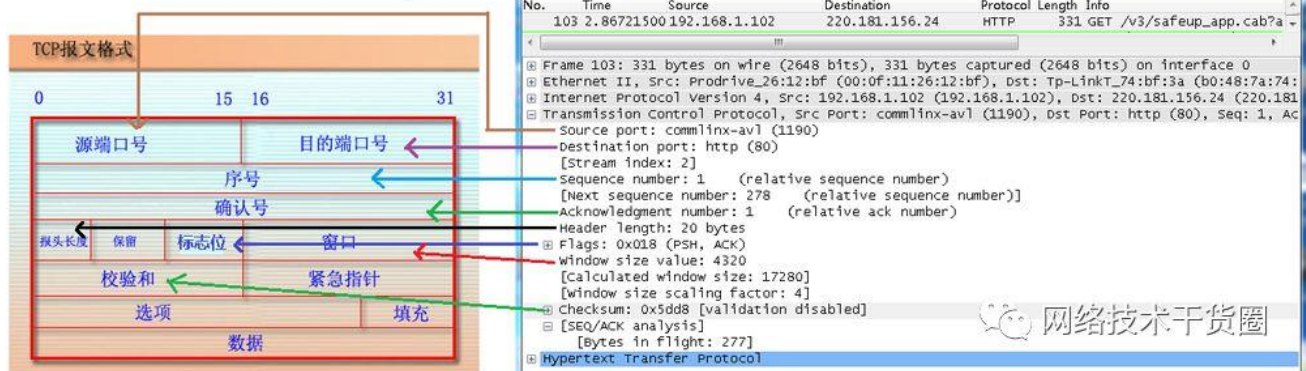
- Frame：物理層的數據幀概況
- Ethernet II：數據鏈路層以太網幀頭部信息
- Internet Protocol Version 4：互聯網層 IP 包頭部信息
- Transmission Control Protocol：傳輸層 T 的數據段頭部信息，此處是 TCP
- Hypertext Transfer Protocol：應用層的信息，此處是 HTTP 協議



## TCP 包的具體內容

從下圖可以看到 wireshark 捕獲到的 TCP 包中的每個字段。





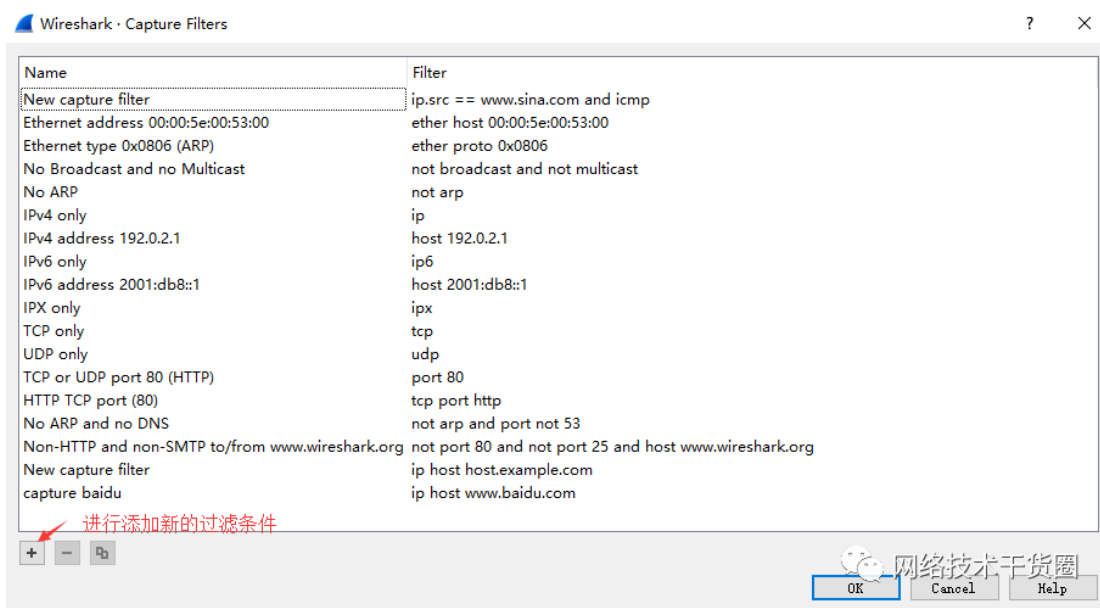
## 1. Dissector Pane(數據包字節區)。

## Wireshark 過濾器設置

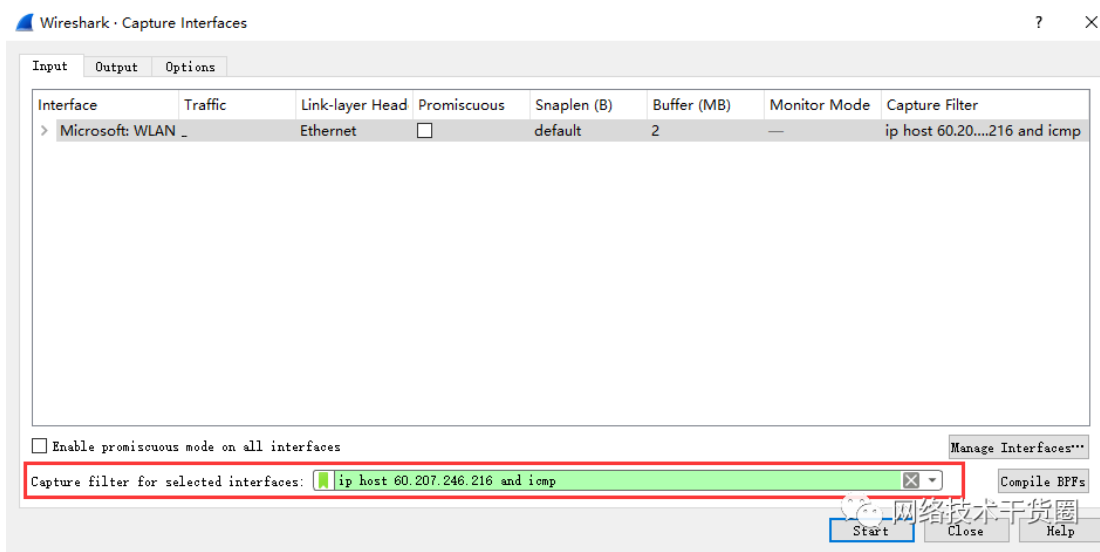
初學者使用 wireshark 時，將會得到大量的冗餘數據包列表，以至於很難找到自己需要抓取的數據包部分。wireshark 工具中自帶了兩種類型的過濾器，學會使用這兩種過濾器會幫助我們在大量的數據中迅速找到我們需要的信息。

### (1) 抓包過濾器

捕獲過濾器的菜單欄路徑為 Capture --> Capture Filters。用於在抓取數據包前設置。



如何使用？可以在抓取數據包前設置如下。



ip host 60.207.246.216 and icmp 表示只捕獲主機 IP 為 60.207.246.216 的 ICMP 數據包。獲取結果如下：

No.	Time	Source	Destination	Protocol	Length	Info
1	2018-12-1...	192.168.1.104	60.207.246.216	ICMP	74	Echo (ping) request id=0x0001, seq=41/10496, ttl=64 (reply in 2)
2	2018-12-1...	60.207.246.216	192.168.1.104	ICMP	74	Echo (ping) reply id=0x0001, seq=41/10496, ttl=54 (request in 1)
3	2018-12-1...	192.168.1.104	60.207.246.216	ICMP	74	Echo (ping) request id=0x0001, seq=42/10752, ttl=64 (reply in 4)
4	2018-12-1...	60.207.246.216	192.168.1.104	ICMP	74	Echo (ping) reply id=0x0001, seq=42/10752, ttl=54 (request in 3)
5	2018-12-1...	192.168.1.104	60.207.246.216	ICMP	74	Echo (ping) request id=0x0001, seq=43/11008, ttl=64 (reply in 6)
6	2018-12-1...	60.207.246.216	192.168.1.104	ICMP	74	Echo (ping) reply id=0x0001, seq=43/11008, ttl=54 (request in 5)
7	2018-12-1...	192.168.1.104	60.207.246.216	ICMP	74	Echo (ping) request id=0x0001, seq=44/11264, ttl=64 (reply in 8)
8	2018-12-1...	60.207.246.216	192.168.1.104	ICMP	74	Echo (ping) reply id=0x0001, seq=44/11264, ttl=54 (request in 7)

## (2) 顯示過濾器

顯示過濾器是用於在抓取數據包後設置過濾條件進行過濾數據包。通常是在抓取數據包時設置條件相對寬泛或者沒有設置導致抓取的數據包內容較多時使用顯示過濾器設置條件過濾以方便分析。同樣上述場景，在捕獲時未設置抓包過濾規則直接通過網卡進行抓取所有數據包，如下



執行'\$ ping huawei.com' 獲取的數據包列表如下



No.	Time	Source	Destination	Protocol	Length	Info
170	2018-12-1...	223.252.199.7	192.168.1.104	TCP	54	443 → 4817 [ACK] Seq=3276 Ack=2117 Win=7168 Len=0
171	2018-12-1...	223.252.199.7	192.168.1.104	TCP	54	443 → 4817 [ACK] Seq=3276 Ack=3944 Win=10752 Len=0
172	2018-12-1...	223.252.199.7	192.168.1.104	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
173	2018-12-1...	223.252.199.7	192.168.1.104	TLSv1.2	621	Application Data
174	2018-12-1...	223.252.199.7	192.168.1.104	TLSv1.2	103	Application Data
175	2018-12-1...	192.168.1.104	223.252.199.7	TCP	54	4817 → 443 [ACK] Seq=3944 Ack=4150 Win=65280 Len=0
176	2018-12-1...	192.168.1.104	59.111.181.155	TLSv1.2	772	Application Data
177	2018-12-1...	192.168.1.104	180.89.255.2	DNS	75	Standard query 0x8f78 AAAA maw.netease.com
178	2018-12-1...	180.89.255.2	192.168.1.104	DNS	130	Standard query response 0x8f78 AAAA maw.netease.com SOA ns4.netease.net
179	2018-12-1...	59.111.181.155	192.168.1.104	TLSv1.2	524	Application Data
180	2018-12-1...	192.168.1.104	59.111.181.155	TCP	54	4771 → 443 [ACK] Seq=2873 Ack=1881 Win=256 Len=0
181	2018-12-1...	192.168.1.1	224.0.0.1	IGMPv3	50	Membership Query, general
182	2018-12-1...	192.168.1.1	224.0.0.1	IGMPv3	50	Membership Query, general

觀察上述獲取的數據包列表，含有大量的無效數據。這時可以通過設置顯示器過濾條件進行提取分析信息。ip.addr == 211.162.2.183 and icmp。並進行過濾。

No.	Time	Source	Destination	Protocol	Length	Info
85	2018-12-1...	192.168.1.104	211.162.2.183	ICMP	74	Echo (ping) request id=0x0001, seq=45/11520, ttl=64 (reply in 87)
87	2018-12-1...	211.162.2.183	192.168.1.104	ICMP	74	Echo (ping) reply id=0x0001, seq=45/11520, ttl=52 (request in 85)
88	2018-12-1...	192.168.1.104	211.162.2.183	ICMP	74	Echo (ping) request id=0x0001, seq=46/11776, ttl=64 (reply in 89)
89	2018-12-1...	211.162.2.183	192.168.1.104	ICMP	74	Echo (ping) reply id=0x0001, seq=46/11776, ttl=52 (request in 88)
91	2018-12-1...	192.168.1.104	211.162.2.183	ICMP	74	Echo (ping) request id=0x0001, seq=47/12032, ttl=64 (reply in 91)
92	2018-12-1...	211.162.2.183	192.168.1.104	ICMP	74	Echo (ping) reply id=0x0001, seq=47/12032, ttl=52 (request in 91)
135	2018-12-1...	192.168.1.104	211.162.2.183	ICMP	74	Echo (ping) request id=0x0001, seq=48/12288, ttl=64 (reply in 136)
136	2018-12-1...	211.162.2.183	192.168.1.104	ICMP	74	Echo (ping) reply id=0x0001, seq=48/12288, ttl=52 (request in 135)

上述介紹了抓包過濾器和顯示過濾器的基本使用方法。在組網不複雜或者流量不大情況下，使用顯示器過濾器進行抓包後處理就可以滿足我們使用。下面介紹一下兩者間的語法以及它們的區別。

## wireshark 過濾器表達式的規則

### 1、抓包過濾器語法和實例

抓包過濾器類型 Type (host、net、port) 、方向 Dir (src、dst) 、協議 Proto (ether、ip、tcp、udp、http、icmp、ftp 等) 、邏輯運算符 (&& 與、|| 或、! 非)

## (1) 協議過濾

比較簡單，直接在抓包過濾框中直接輸入協議名即可。

- tcp, 只顯示 TCP 協議的數據包列表
- http, 只查看 HTTP 協議的數據包列表
- icmp, 只顯示 ICMP 協議的數據包列表

## (2) IP 過濾

host 192.168.1.104

src host 192.168.1.104

dst host 192.168.1.104

## (3) 端口過濾

port 80

src port 80

dst port 80

## (4) 邏輯運算符&& 與、|| 或、! 非

抓取主機地址為 192.168.1.80、目的端口為 80 的數據包

```
src host 192.168.1.104 && dst port 80
```

抓取主機為 192.168.1.104 或者 192.168.1.102 的數據包

```
host 192.168.1.104 || host 192.168.1.102
```

不抓取廣播數據包

```
! broadcast
```

## 2、顯示過濾器語法和實例

### (1) 比較操作符

比較操作符有== 等於、!= 不等於、> 大於、< 小於、>= 大於等於、<=小於等於。

### (2) 協議過濾

比較簡單，直接在 Filter 框中直接輸入協議名即可。

“注意：協議名稱需要輸入小寫。

”

- tcp, 只顯示 TCP 協議的數據包列表
- http, 只查看 HTTP 協議的數據包列表
- icmp, 只顯示 ICMP 協議的數據包列表

No.	Time	Source	Destination	Protocol	Length	Info
937	100.153605	192.168.1.104	119.75.217.26	ICMP	74	Echo (ping) request id=0x0001,
938	100.175141	119.75.217.26	192.168.1.104	ICMP	74	Echo (ping) response id=0x0001,
943	101.171744	192.168.1.104	119.75.217.26	ICMP	74	Echo (ping) request id=0x0001,

### (3) ip 過濾

顯示源地址為 192.168.1.104 的數據包列表

```
ip.src == 192.168.1.104
```

顯示目標地址為 192.168.1.104 的數據包列表

```
ip.dst == 192.168.1.104,
```

顯示源 IP 地址或目標 IP 地址為 192.168.1.104 的數據包列表

```
ip.addr == 192.168.1.104
```

No.	Time	Source	Destination	Protocol	Length	Info
925	97.971645	192.168.1.104	101.247.50.254	DNS	75	Standard query 0x2dcb AAAA
926	97.974107	101.247.50.254	192.168.1.104	DNS	130	Standard query response 0x2dcb
927	98.026392	59.111.181.155	192.168.1.104	TLSv1.2	524	Application/javascript
928	98.067066	192.168.1.104	59.111.181.155	TCP	54	11584 → 443 [ACK] Seq=13643

### (4) 端口過濾

顯示源主機或者目的主機端口為 80 的數據包列表。

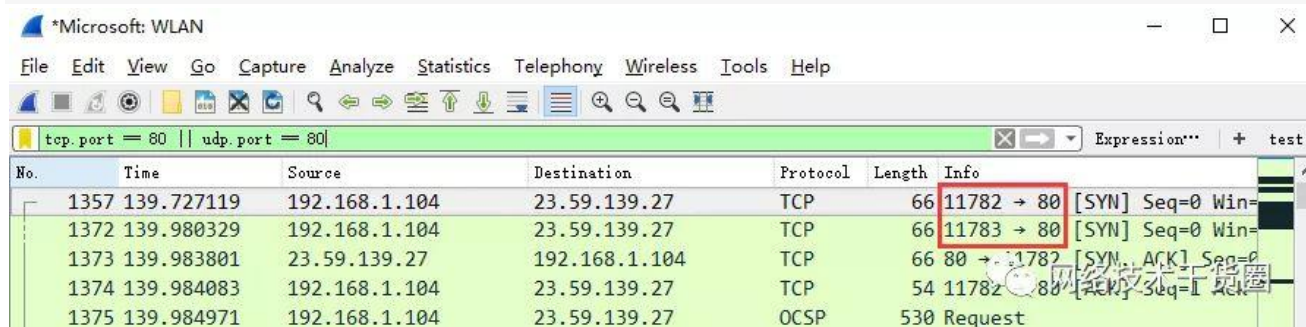
```
tcp.port == 80
```

只顯示 TCP 協議的源主機端口為 80 的數據包列表。

```
tcp.srcport == 80
```

只顯示 TCP 協議的目的主機端口為 80 的數據包列表。

```
tcp.dstport == 80
```



No.	Time	Source	Destination	Protocol	Length	Info
1357	139.727119	192.168.1.104	23.59.139.27	TCP	66	11782 → 80 [SYN] Seq=0 Win=
1372	139.980329	192.168.1.104	23.59.139.27	TCP	66	11783 → 80 [SYN] Seq=0 Win=
1373	139.983801	23.59.139.27	192.168.1.104	TCP	66	80 → 11782 [SYN, ACK] Seq=0
1374	139.984083	192.168.1.104	23.59.139.27	TCP	54	11782 → 80 [ACK] Seq=1
1375	139.984971	192.168.1.104	23.59.139.27	OCSP	530	Request

## (5) Http 模式過濾

只顯示 HTTP GET 方法的。

```
http.request.method=="GET"
```

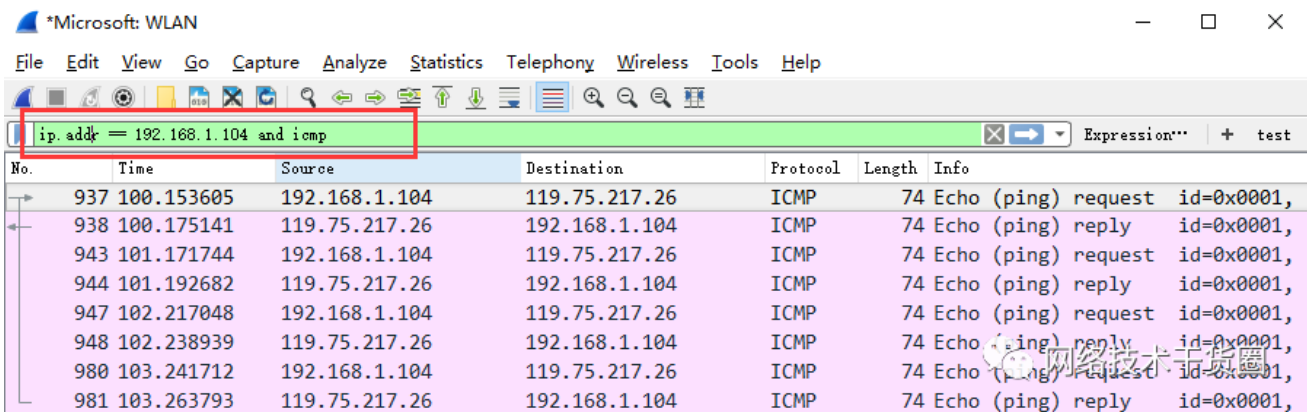
## (6) 邏輯運算符為 and/or/not

過濾多個條件組合時，使用 and/or。

比如獲取 IP 地址為 192.168.1.104 的 ICMP 數據包表達式為

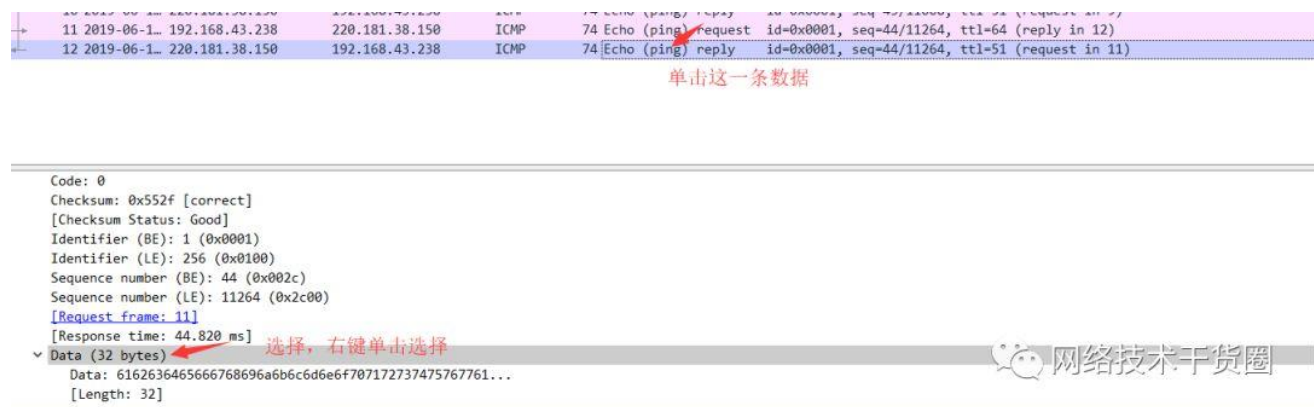
```
ip.addr == 192.168.1.104 and icmp
```



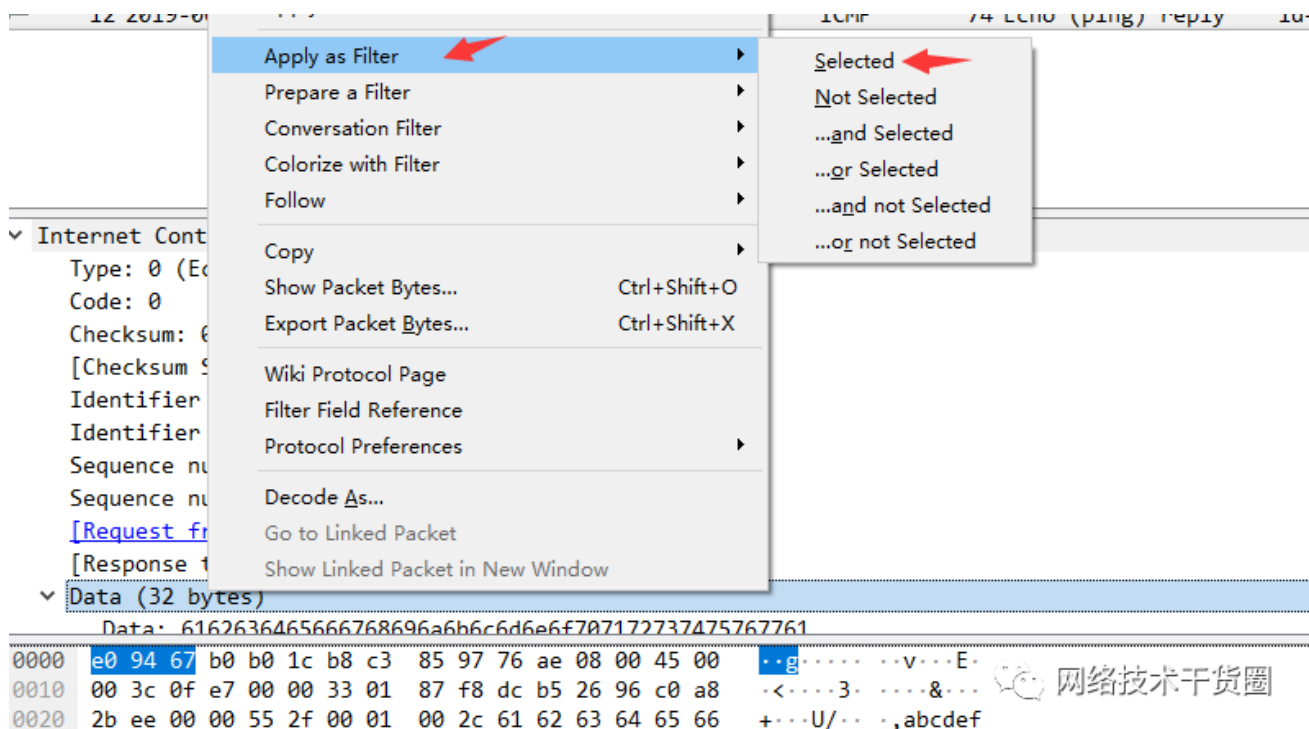


## (7) 按照數據包內容過濾。

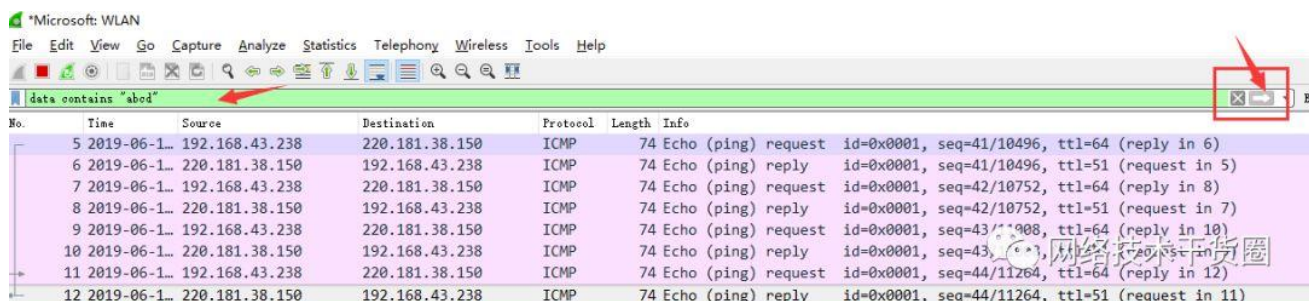
假設我要以 ICMP 層中的內容進行過濾，可以單擊選中界面中的碼流，在下方進行選中數據。如下



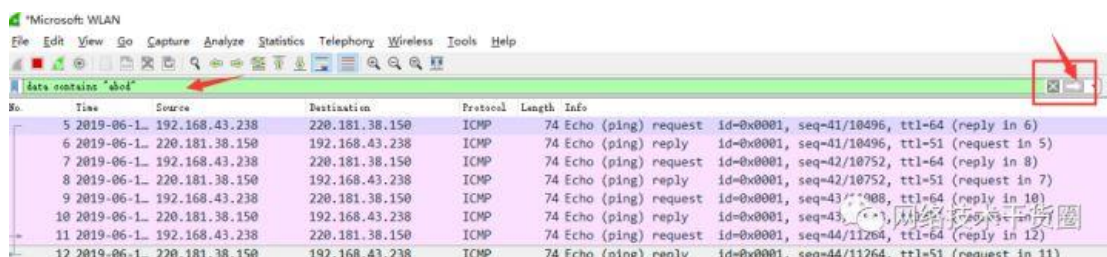
右鍵單擊選中後出現如下界面



選中 Select 後在過濾器中顯示如下



後面條件表達式就需要自己填寫。如下我想過濾出 data 數據包中包含 "abcd" 內容的數據流。包含的關鍵詞是 contains 後面跟上內容。



看到這，基本上對 wireshak 有了初步瞭解。

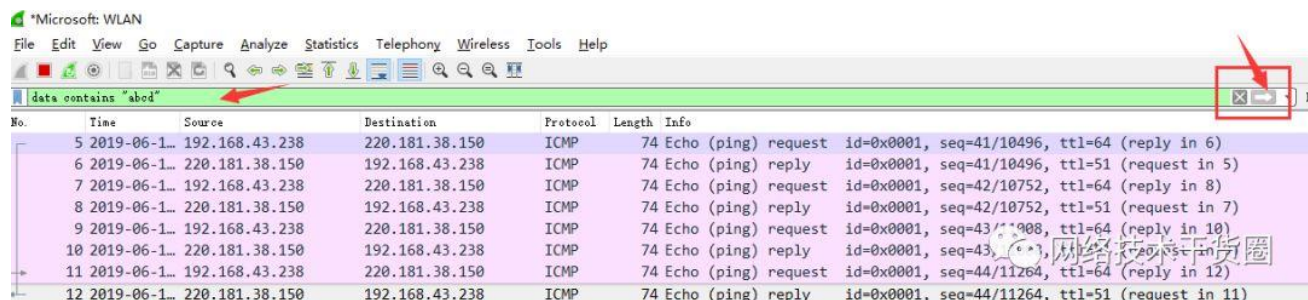
## Wireshark 抓包分析 TCP 三次握手

### (1) TCP 三次握手連接建立過程

Step1: 客戶端發送一個 SYN=1, ACK=0 標志的數據包給服務端，請求進行連接，這是第一次握手；

Step2: 服務端收到請求並且允許連接的話，就會發送一個 SYN=1, ACK=1 標志的數據包給發送端，告訴它，可以通訊了，並且讓客戶端發送一個確認數據包，這是第二次握手；

Step3: 服務端發送一個 SYN=0, ACK=1 的數據包給客戶端端，告訴它連接已被確認，這就是第三次握手。TCP 連接建立，開始通訊。



No.	Time	Source	Destination	Protocol	Length	Info
5	2019-06-1...	192.168.43.238	220.181.38.150	ICMP	74	Echo (ping) request id=0x0001, seq=41/10496, ttl=64 (reply in 6)
6	2019-06-1...	220.181.38.150	192.168.43.238	ICMP	74	Echo (ping) reply id=0x0001, seq=41/10496, ttl=51 (request in 5)
7	2019-06-1...	192.168.43.238	220.181.38.150	ICMP	74	Echo (ping) request id=0x0001, seq=42/10752, ttl=64 (reply in 8)
8	2019-06-1...	220.181.38.150	192.168.43.238	ICMP	74	Echo (ping) reply id=0x0001, seq=42/10752, ttl=51 (request in 7)
9	2019-06-1...	192.168.43.238	220.181.38.150	ICMP	74	Echo (ping) request id=0x0001, seq=43/11008, ttl=64 (reply in 10)
10	2019-06-1...	220.181.38.150	192.168.43.238	ICMP	74	Echo (ping) reply id=0x0001, seq=43/11008, ttl=51 (request in 9)
11	2019-06-1...	192.168.43.238	220.181.38.150	ICMP	74	Echo (ping) request id=0x0001, seq=44/11264, ttl=64 (reply in 12)
12	2019-06-1...	220.181.38.150	192.168.43.238	ICMP	74	Echo (ping) reply id=0x0001, seq=44/11264, ttl=51 (request in 11)

### (2) wireshark 抓包獲取訪問指定服務端數據包

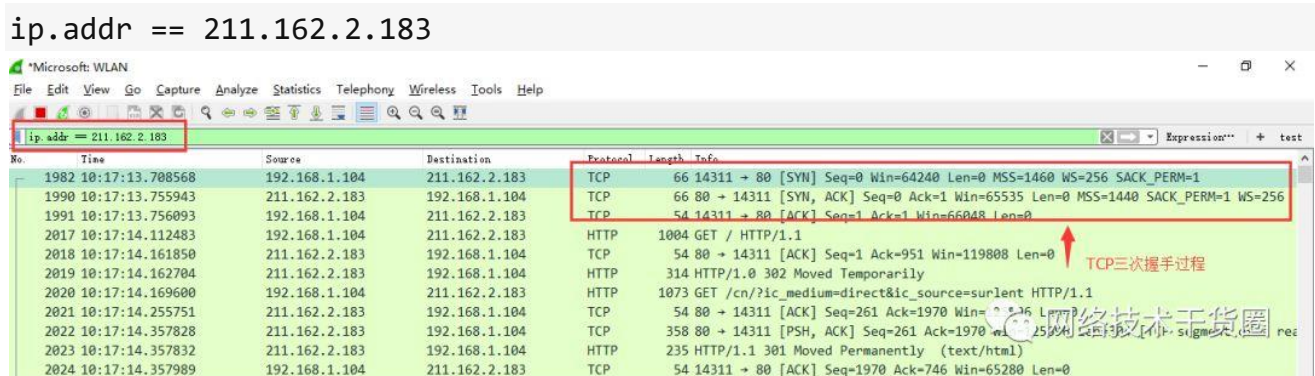
Step1: 啟動 wireshark 抓包，打開瀏覽器輸入 [\\_huawei.com](http://huawei.com)。

Step2: 使用 ping [\\_huawei.com](http://huawei.com) 獲取 IP。

```
C:\Users\shiyanan>ping www.huawei.com

正在 Ping huawei.dtwscache.ourwebcdn.com [211.162.2.183] 具有 32 字节的数据:
来自 211.162.2.183 的回复: 字节=32 时间=57ms TTL=51
```

Step3: 輸入過濾條件獲取待分析數據包列表



圖中可以看到 wireshark 截獲到了三次握手的三個數據包。第四個包才是 HTTP 的，這說明 HTTP 的確是使用 TCP 建立連接的。

## 第一次握手數據包

客戶端發送一個 TCP，標志位為 SYN，序列號為 0，代表客戶端請求建立連接。如下圖。



Microsoft WLAN

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 211.162.2.183

No.	Time	Source	Destination	Protocol	Length	Info
1982	10:17:13.708568	192.168.1.104	211.162.2.183	TCP	66	14311 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1990	10:17:13.755943	211.162.2.183	192.168.1.104	TCP	66	80 → 14311 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM=1 WS=256
1991	10:17:13.756093	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily
2020	10:17:14.169600	192.168.1.104	211.162.2.183	HTTP	1073	GET /cn/?ic_medium=direct&ic_source=surlent HTTP/1.1
2021	10:17:14.255751	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=261 Ack=1970 Win=0 Len=0
2022	10:17:14.357828	211.162.2.183	192.168.1.104	TCP	358	80 → 14311 [PSH, ACK] Seq=261 Ack=1970 Win=0 Len=25
2023	10:17:14.357832	211.162.2.183	192.168.1.104	HTTP	235	HTTP/1.1 301 Moved Permanently (text/html)
2024	10:17:14.357989	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1970 Ack=746 Win=65280 Len=0

TCP三次握手过程

數據包的關鍵屬性如下：

Microsoft WLAN

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 211.162.2.183

No.	Time	Source	Destination	Protocol	Length	Info
1982	10:17:13.708568	192.168.1.104	211.162.2.183	TCP	66	14311 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1990	10:17:13.755943	211.162.2.183	192.168.1.104	TCP	66	80 → 14311 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM=1 WS=256
1991	10:17:13.756093	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily
2020	10:17:14.169600	192.168.1.104	211.162.2.183	HTTP	1073	GET /cn/?ic_medium=direct&ic_source=surlent HTTP/1.1
2021	10:17:14.255751	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=261 Ack=1970 Win=0 Len=0
2022	10:17:14.357828	211.162.2.183	192.168.1.104	TCP	358	80 → 14311 [PSH, ACK] Seq=261 Ack=1970 Win=0 Len=25
2023	10:17:14.357832	211.162.2.183	192.168.1.104	HTTP	235	HTTP/1.1 301 Moved Permanently (text/html)
2024	10:17:14.357989	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1970 Ack=746 Win=65280 Len=0

TCP三次握手过程

SYN：标志位，表示请求建立连接

- Seq = 0：初始建立连接值为 0，数据包的相对序列号从 0 开始，表示当前还没有发送数据
- Ack = 0：初始建立连接值为 0，已经收到包的数量，表示当前没有接收到数据

## 第二次握手的数据包

服务器发回确认包，标志位为 SYN,ACK。将确认序号(Acknowledgement Number)设置为客户端的 ISN 加 1 以。即 0+1=1，如下图



Microsoft WLAN

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 211.162.2.183

No.	Time	Source	Destination	Protocol	Length	Info
1982	10:17:13.708568	192.168.1.104	211.162.2.183	TCP	66	14311 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1990	10:17:13.755943	211.162.2.183	192.168.1.104	TCP	66	80 → 14311 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM=1 WS=256
1991	10:17:13.756093	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1 Ack=1 Win=65048 Len=0
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily
2020	10:17:14.169600	192.168.1.104	211.162.2.183	HTTP	1073	GET /cn/?ic_medium=direct&ic_source=surlent HTTP/1.1
2021	10:17:14.255751	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=261 Ack=1970 Win=0 Len=0
2022	10:17:14.357828	211.162.2.183	192.168.1.104	TCP	358	80 → 14311 [PSH, ACK] Seq=261 Ack=1970 Win=0 Len=25
2023	10:17:14.357832	211.162.2.183	192.168.1.104	HTTP	235	HTTP/1.1 301 Moved Permanently (text/html)
2024	10:17:14.357989	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1970 Ack=746 Win=65280 Len=0

TCP三次握手过程

數據包的關鍵屬性如下：

- Seq = 0：初始建立值為 0，表示當前還沒有發送數據
- Ack = 1：表示當前端成功接收的數據位數，雖然客戶端沒有發送任何有效數據，確認號還是被加 1，因為包含 SYN 或 FIN 標志位。（並不會對有效數據的計數產生影響，因為含有 SYN 或 FIN 標志位的包並不攜帶有效數據）

## 第三次握手的數據包

客戶端再次發送確認包(ACK) SYN 標志位為 0,ACK 標志位為 1.並且把服務器發來 ACK 的序號字段+1,放在確定字段中發送給對方.並且在數據段放寫 ISN 的+1，如下圖：

No.	Time	Source	Destination	Protocol	Length	Info
1982	10:17:13.708568	192.168.1.104	211.162.2.183	TCP	66	14311 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1990	10:17:13.755943	211.162.2.183	192.168.1.104	TCP	66	80 → 14311 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM=1 WS=256
1991	10:17:13.756093	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily

Transmission Control Protocol, Src Port: 14311, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

Source Port: 14311

Destination Port: 80

[Stream index: 17]

[TCP Segment Len: 0]

Sequence number: 1 (relative sequence number) ←

[Next sequence number: 1 (relative sequence number)]

Acknowledgment number: 1 (relative ack number) ←

0101 .... = Header Length: 20 bytes (5)

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....]

网络技术干货圈

數據包的關鍵屬性如下：

- ACK：標志位，表示已經收到記錄
- Seq = 1：表示當前已經發送 1 個數據
- Ack = 1：表示當前端成功接收的數據位數，雖然服務端沒有發送任何有效數據，確認號還是被加 1，因為包含 SYN 或 FIN 標志位（並不會對有效數據的計數產生影響，因為含有 SYN 或 FIN 標志位的包並不攜帶有效數據）。

就這樣通過了 TCP 三次握手，建立了連接。開始進行數據交互

No.	Time	Source	Destination	Protocol	Length	Info
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily
2020	10:17:14.169600	192.168.1.104	211.162.2.183	HTTP	1073	GET /cn/?ic_medium=direct&ic_source=surlent HTTP/1.1
2021	10:17:14.255751	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=261 Ack=1970 Win=125696 Len=0

Transmission Control Protocol, Src Port: 14311, Dst Port: 80, Seq: 1, Ack: 1, Len: 950

Source Port: 14311

Destination Port: 80

[Stream index: 17]

[TCP Segment Len: 950]

Sequence number: 1 (relative sequence number)

[Next sequence number: 951 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

0101 .... = Header Length: 20 bytes (5)

Flags: 0x018 (PSH, ACK)

Window size value: 258

[Calculated window size: 66048]

[Window size scaling factor: 256]

Checksum: 0x4dc6 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

[SEQ/ACK analysis]

[iRTT: 0.047525000 seconds]

[Bytes in flight: 950]

[Bytes sent since last PSH flag: 950]

网络技术干货圈

No.	Time	Source	Destination	Protocol	Length	Info
1982	10:17:13.708568	192.168.1.104	211.162.2.183	TCP	66	14311 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1990	10:17:13.755943	211.162.2.183	192.168.1.104	TCP	66	80 → 14311 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM=1 WS=256
1991	10:17:13.756093	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily
2020	10:17:14.169600	192.168.1.104	211.162.2.183	HTTP	1073	GET /cn/?ic_medium=direct&ic_source=surlent HTTP/1.1
2021	10:17:14.255751	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=261 Ack=1970 Win=125696 Len=0
2022	10:17:14.357828	211.162.2.183	192.168.1.104	TCP	358	80 → 14311 [PSH, ACK] Seq=261 Ack=1970 Win=125696 Len=304 [TCP segment of a re
2023	10:17:14.357832	211.162.2.183	192.168.1.104	HTTP	235	HTTP/1.1 301 Moved Permanently (text/html)
2024	10:17:14.357989	192.168.1.104	211.162.2.183	TCP	54	14311 → 80 [ACK] Seq=1970 Ack=746 Win=65280 Len=0
5163	10:17:24.357599	192.168.1.104	211.162.2.183	TCP	55	[TCP Keep-Alive] 14311 → 80 [ACK] Seq=1969 Ack=746 Win=65280 Len=1
5170	10:17:24.405625	211.162.2.183	192.168.1.104	TCP	66	[TCP Keep-Alive ACK] 80 → 14311 [ACK] Seq=746 Ack=1970 Win=125696 Len=0 SLE=19
7152	10:17:34.406679	192.168.1.104	211.162.2.183	TCP	55	[TCP Keep-Alive] 14311 → 80 [ACK] Seq=1969 Ack=746 Win=65280 Len=1
7153	10:17:34.454915	211.162.2.183	192.168.1.104	TCP	66	[TCP Keep-Alive ACK] 80 → 14311 [ACK] Seq=746 Ack=1970 Win=125696 Len=0 SLE=19
9120	10:17:44.457849	192.168.1.104	211.162.2.183	TCP	55	[TCP Keep-Alive] 14311 → 80 [ACK] Seq=1969 Ack=746 Win=65280 Len=1
9122	10:17:44.506466	211.162.2.183	192.168.1.104	TCP	66	[TCP Keep-Alive ACK] 80 → 14311 [ACK] Seq=746 Ack=1970 Win=125696 Len=0 SLE=19

网络技术干货圈

下面針對數據交互過程的數據包進行一些說明：

No.	Time	Source	Destination	Protocol	Length	Info
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily
2020	10:17:14.169600	192.168.1.104	211.162.2.183	HTTP	1073	GET /cn/?ic_medium=direct&ic_source=surlent HTTP/1.1
2021	10:17:14.255751	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=261 Ack=1970 Win=125696 Len=0

Transmission Control Protocol, Src Port: 14311, Dst Port: 80, Seq: 1, Ack: 1, Len: 950

Source Port: 14311

Destination Port: 80

[Stream index: 17]

[TCP Segment Len: 950]

Sequence number: 1 (relative sequence number)

[Next sequence number: 951 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

0101 .... = Header Length: 20 bytes (5)

Flags: 0x018 (PSH, ACK)

Window size value: 258

[Calculated window size: 66048]

[Window size scaling factor: 256]

Checksum: 0x4dc6 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

[SEQ/ACK analysis]

[iRTT: 0.047525000 seconds]

[Bytes in flight: 950]

[Bytes sent since last PSH flag: 950]

网络技术干货圈

## 數據包的關鍵屬性說明

Seq: 1

Ack: 1: 說明現在共收到 1 字節數據

No.	Time	Source	Destination	Protocol	Length	Info
2017	10:17:14.112483	192.168.1.104	211.162.2.183	HTTP	1004	GET / HTTP/1.1
2018	10:17:14.161850	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=1 Ack=951 Win=119808 Len=0
2019	10:17:14.162704	211.162.2.183	192.168.1.104	HTTP	314	HTTP/1.0 302 Moved Temporarily
2020	10:17:14.169600	192.168.1.104	211.162.2.183	HTTP	1073	GET /cn/?pic_medium=direct&ic_source=surlent HTTP/1.1
2021	10:17:14.255751	211.162.2.183	192.168.1.104	TCP	54	80 → 14311 [ACK] Seq=261 Ack=1970 Win=125696 Len=0

Transmission Control Protocol, Src Port: 14311, Dst Port: 80, Seq: 1, Ack: 1, Len: 950
Source Port: 14311
Destination Port: 80
[Stream index: 17]
[TCP Segment Len: 950]
Sequence number: 1 (relative sequence number)
[Next sequence number: 951 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
0101 ... = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window size value: 258
[Calculated window size: 66048]
[Window size scaling factor: 256]
Checksum: 0x4dc6 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
[SEQ/ACK analysis]
[iRTT: 0.047525000 seconds]
[Bytes in flight: 950]
[Bytes sent since last PSH flag: 950]

Seq: 1 Ack: 951: 說明現在服務端共收到 951 字節數據

在 TCP 層，有個 FLAGS 字段，這個字段有以下幾個標識：SYN，FIN，ACK，PSH，RST，URG。如下



```

1000 .... ..0... ..0... ..0... ..0... ..0... ..0... ..0... ..0...
Header Length: 32 bytes (8)
▼ Flags: 0x012 (SYN, 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
    > .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
[TCP Flags: .....A..S.]

```

其中，對於我們日常的分析有用的就是前面的五個字段。它們的含義是：SYN 表示建立連接，FIN 表示關閉連接，ACK 表示響應，PSH 表示有 DATA 數據傳輸，RST 表示連接重置。

調整數據包列表中時間戳顯示格式。

No.	Time	Source	Destination	Protocol	Length	Info
1986	2018-12-16 11:37:44.583026	211.162.2.183	192.168.1.104	TCP	54	443 → 2206 [FIN, ACK] Seq=73867 Ack=3410 Win=137472 Len=0
1987	2018-12-16 11:37:44.583302	211.162.2.183	192.168.1.104	TCP	54	2201 → 443 [ACK] Seq=3435 Ack=64910 Win=66048 Len=0
1988	2018-12-16 11:37:44.583441	211.162.2.183	192.168.1.104	TCP	54	2201 → 443 [FIN, ACK] Seq=3435 Ack=64910 Win=66048 Len=0
1989	2018-12-16 11:37:44.583460	211.162.2.183	192.168.1.104	TCP	54	2206 → 443 [ACK] Seq=3435 Ack=64910 Win=66048 Len=0
1990	2018-12-16 11:37:44.583904	211.162.2.183	192.168.1.104	TCP	54	2206 → 443 [FIN, ACK] Seq=3410 Ack=73868 Win=66048 Len=0
1991	2018-12-16 11:37:44.587146	211.162.2.183	192.168.1.104	TLShv.1.2	85	Encrypted Alert