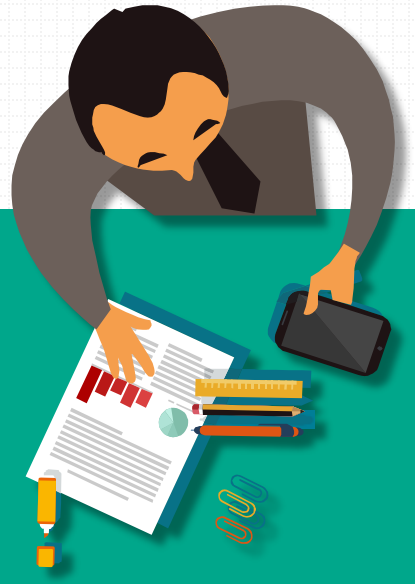
[Lab] 網路封包分析器(Network Packet Analyzer)

異質多網多媒體服務

國立臺北科技大學電子工程系
授課教師：李昭賢 副教授
電子郵件：chlee@ntut.edu.tw
校內分機：2288



學習目標 Outline

- Packet Analyzer / Sniffer
  - Wireshark
- HTTP
  - Chrome Browser
  - Postman

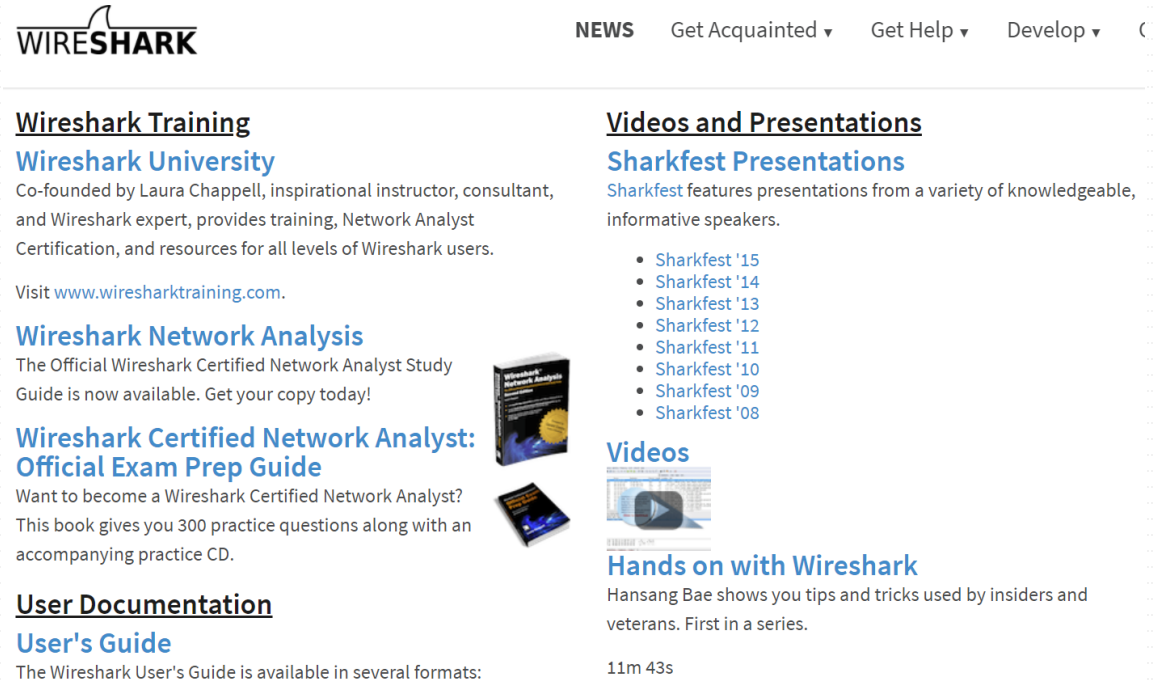
網路封包分析器(Network Packet Analyzer)

- 封包分析(Analysis)/監聽(Sniffing)
  - 當資料流經網路時，將其擷取並解譯。
  - 呈現特定封包的結構與內容。
  - 掌握網路資源使用情況。
    - 何人、何事佔用頻寬
    - 識別網路攻擊或惡意行為
- 封包分析器(Analyzer)/監聽器(Sniffer)
  - 封包擷取、流量統計之工具
  - TCPdump、Wireshark (Ethereal)、MS Message Analyzer (Network Monitor)
  - [Library/API] libpcap (Linux), WinPcap, Npcap (MS Windows)

Wireshark (Ethereal)

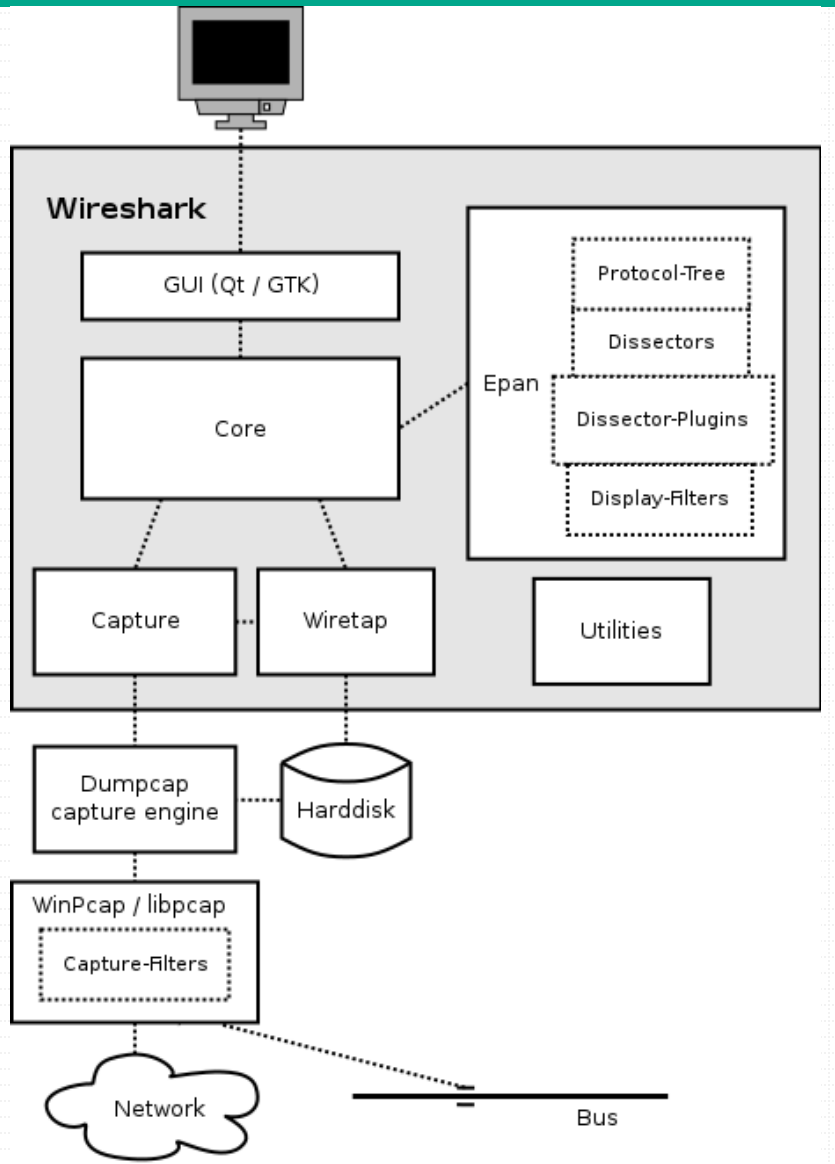
- 免費開源的網路封包分析軟體
  - 1998年Gerald Combs開發Ethereal
  - 2006年更名為Wireshark
- 目前全世界最廣泛的網路封包分析軟體
  - 官方網站：https://www.wireshark.org/
  - 官方說明：https://www.wireshark.org/docs/

ETHEREAL
PACKET SNIFFER

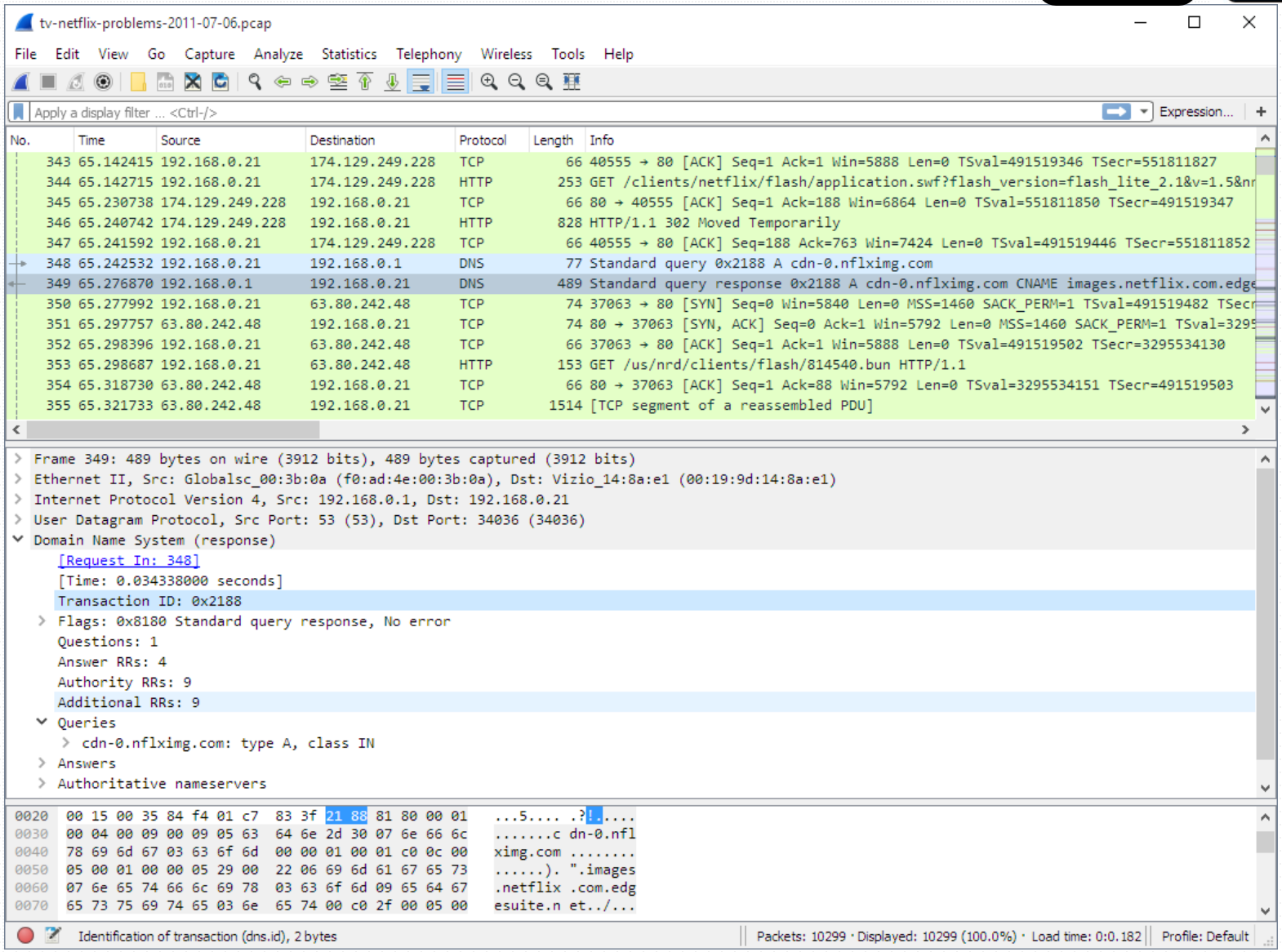


Wireshark (Ethereal)

- GUI
  - Handling of all user input/output.
- Core
  - Main "glue code" that holds the other blocks together.
- Epan
  - Enhanced Packet Analyzer
- Wiretap
  - The wiretap library is used to read and write capture files in libpcap, pcapng, and many other file formats.
- Capture
  - The interface with the capture engine.



Wireshark (Ethereal)



Main Menu

Display Filter

Packet List

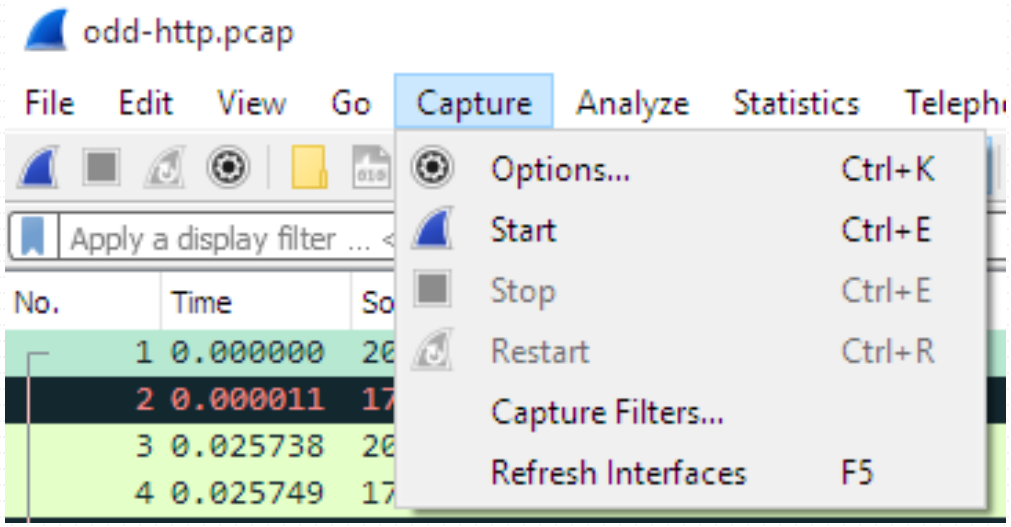
Packet Details

Packet Bytes

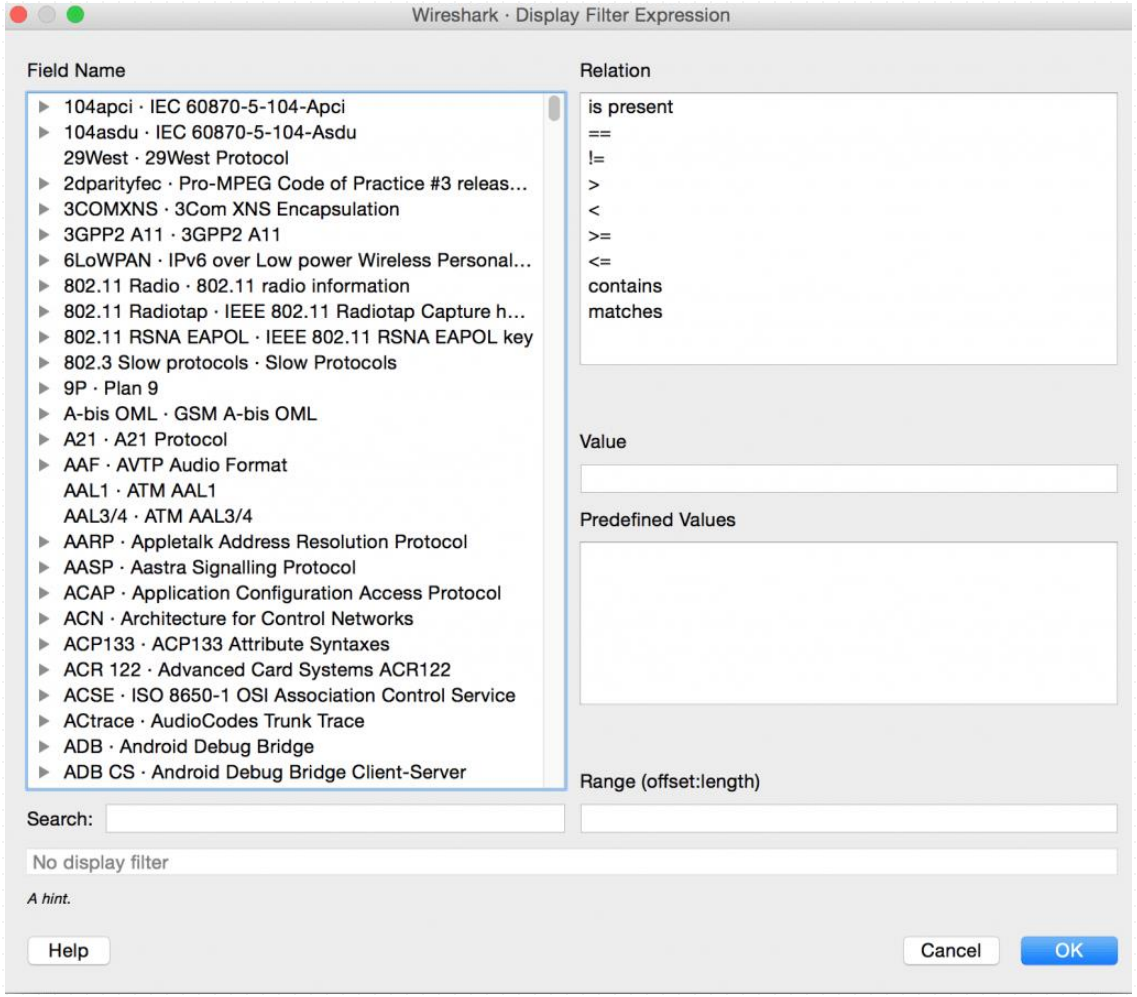
圖片來源：https://www.wireshark.org/docs/wsug_html/

Wireshark (Ethereal)

- Capture → Start

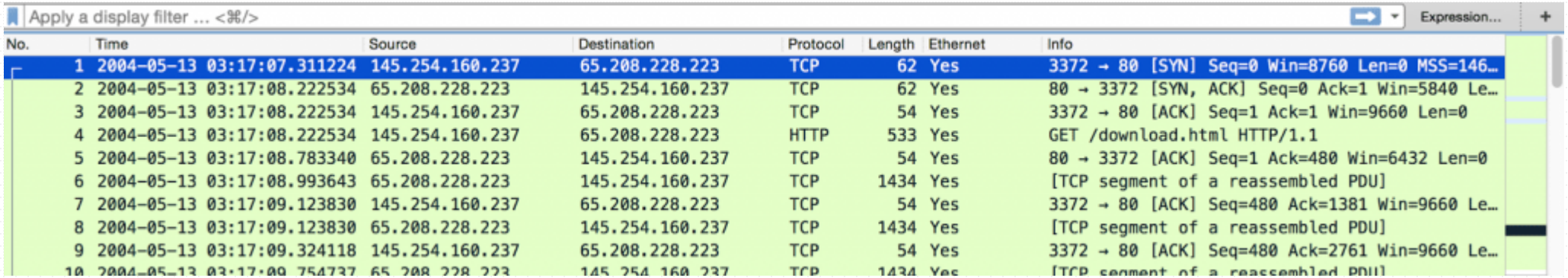


- Display Filter



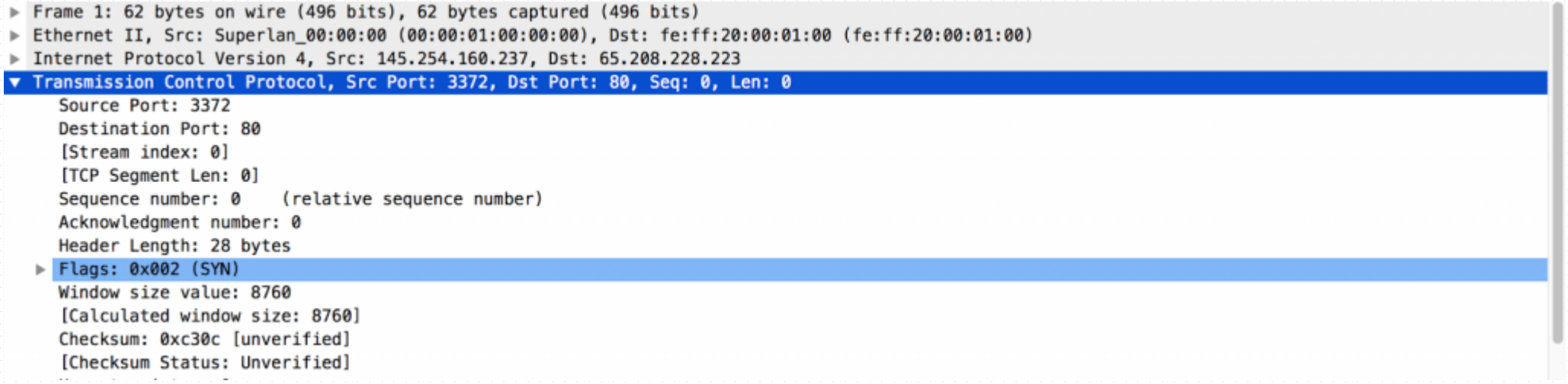
Wireshark (Ethereal)

- Packet List



No.	Time	Source	Destination	Protocol	Length	Ethernet	Info
1	2004-05-13 03:17:07.311224	145.254.160.237	65.208.228.223	TCP	62	Yes	3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=146…
2	2004-05-13 03:17:08.222534	65.208.228.223	145.254.160.237	TCP	62	Yes	80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Le…
3	2004-05-13 03:17:08.222534	145.254.160.237	65.208.228.223	TCP	54	Yes	3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
4	2004-05-13 03:17:08.222534	145.254.160.237	65.208.228.223	HTTP	533	Yes	GET /download.html HTTP/1.1
5	2004-05-13 03:17:08.783340	65.208.228.223	145.254.160.237	TCP	54	Yes	80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=0
6	2004-05-13 03:17:08.993643	65.208.228.223	145.254.160.237	TCP	1434	Yes	[TCP segment of a reassembled PDU]
7	2004-05-13 03:17:09.123830	145.254.160.237	65.208.228.223	TCP	54	Yes	3372 → 80 [ACK] Seq=480 Ack=1381 Win=9660 Le…
8	2004-05-13 03:17:09.123830	65.208.228.223	145.254.160.237	TCP	1434	Yes	[TCP segment of a reassembled PDU]
9	2004-05-13 03:17:09.324118	145.254.160.237	65.208.228.223	TCP	54	Yes	3372 → 80 [ACK] Seq=480 Ack=2761 Win=9660 Le…
10	2004-05-13 03:17:09.754737	65.208.228.223	145.254.160.237	TCP	1434	Yes	[TCP segment of a reassembled PDU]

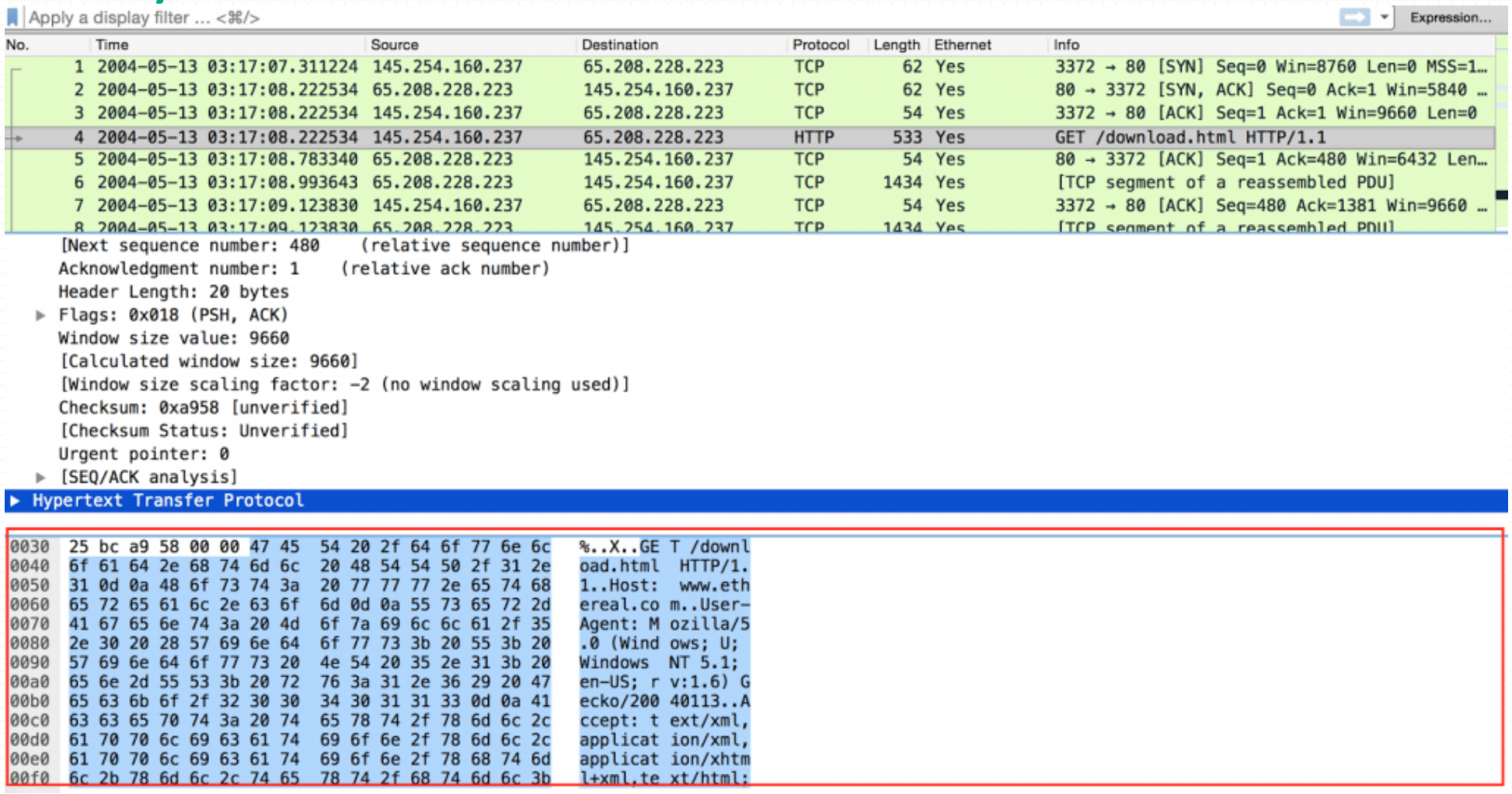
- Packet Details



```
▶ Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
▶ Ethernet II, Src: Superlan_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
▶ Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
▼ Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 3372
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 0
    Header Length: 28 bytes
  ▶ Flags: 0x002 (SYN)
    Window size value: 8760
    [Calculated window size: 8760]
    Checksum: 0xc30c [unverified]
    [Checksum Status: Unverified]
```

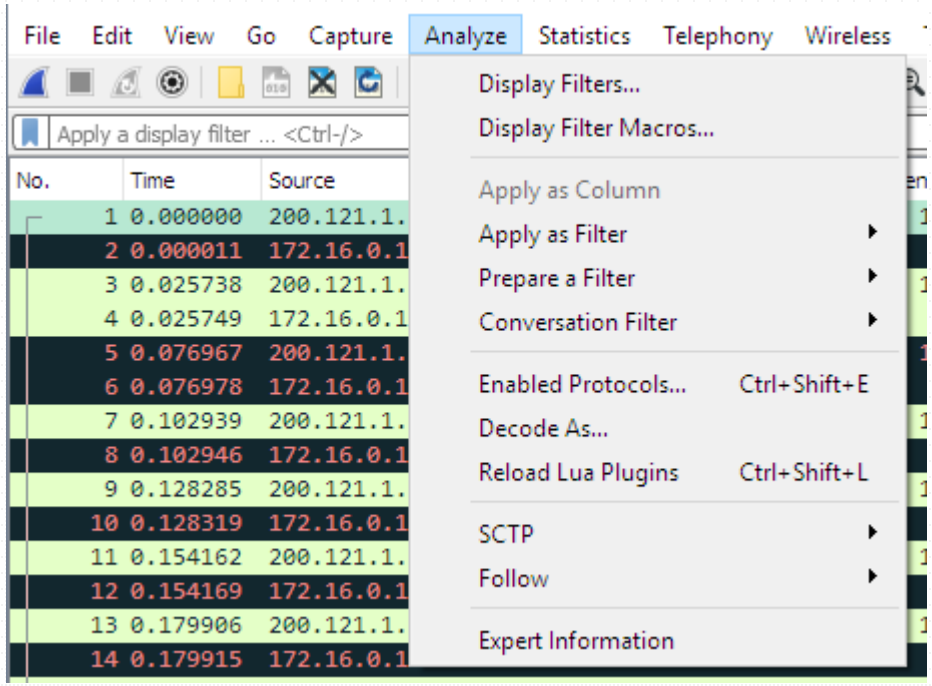
Wireshark (Ethereal)

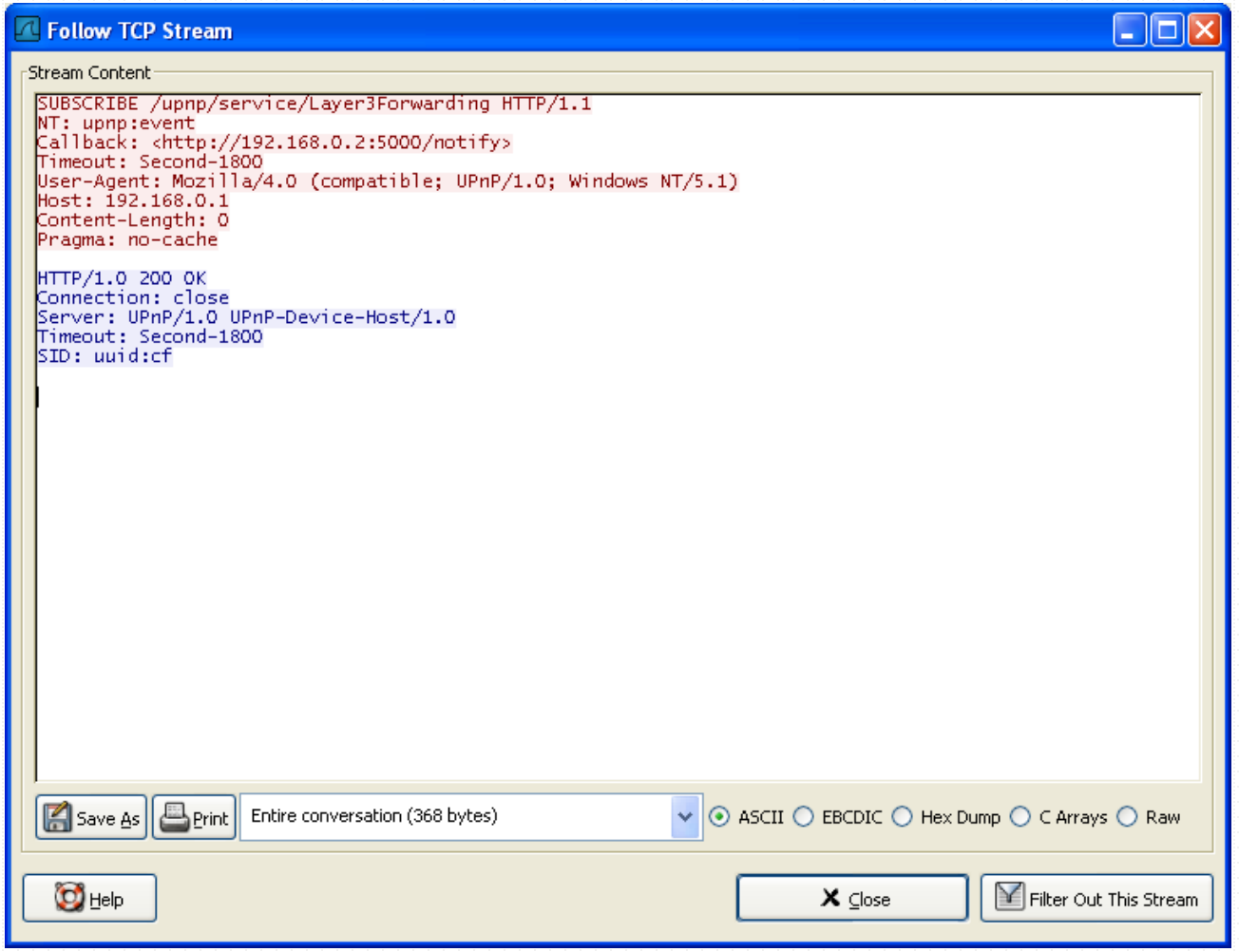
- Packet Bytes



Wireshark (Ethereal)

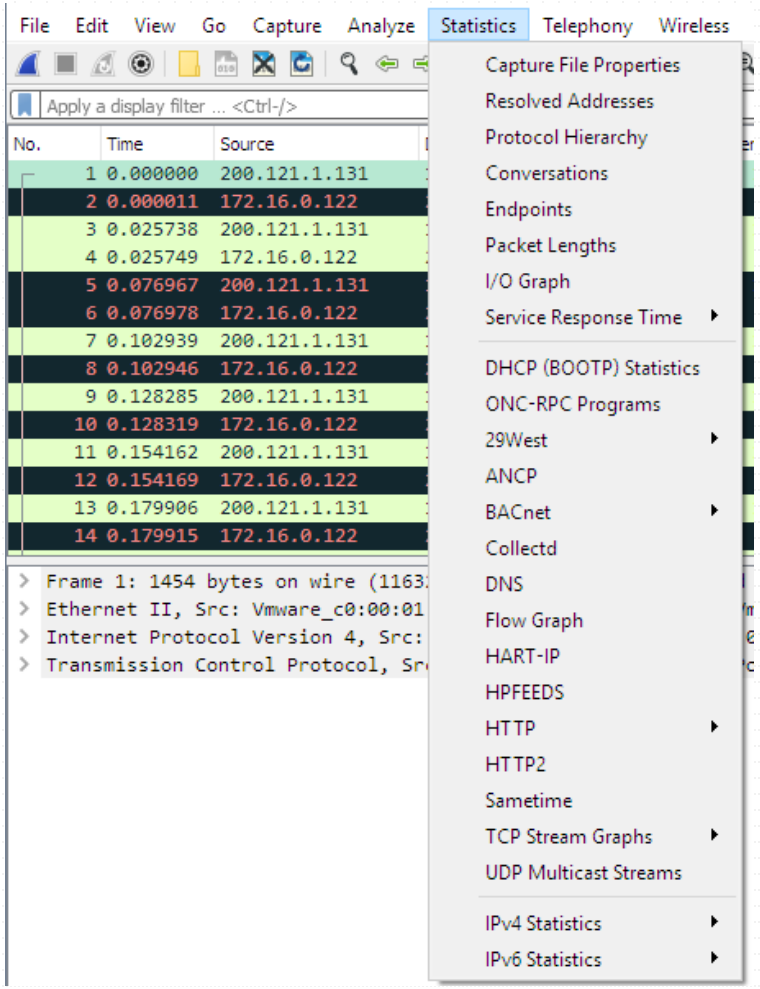
- Analyze → Follow TCP Stream

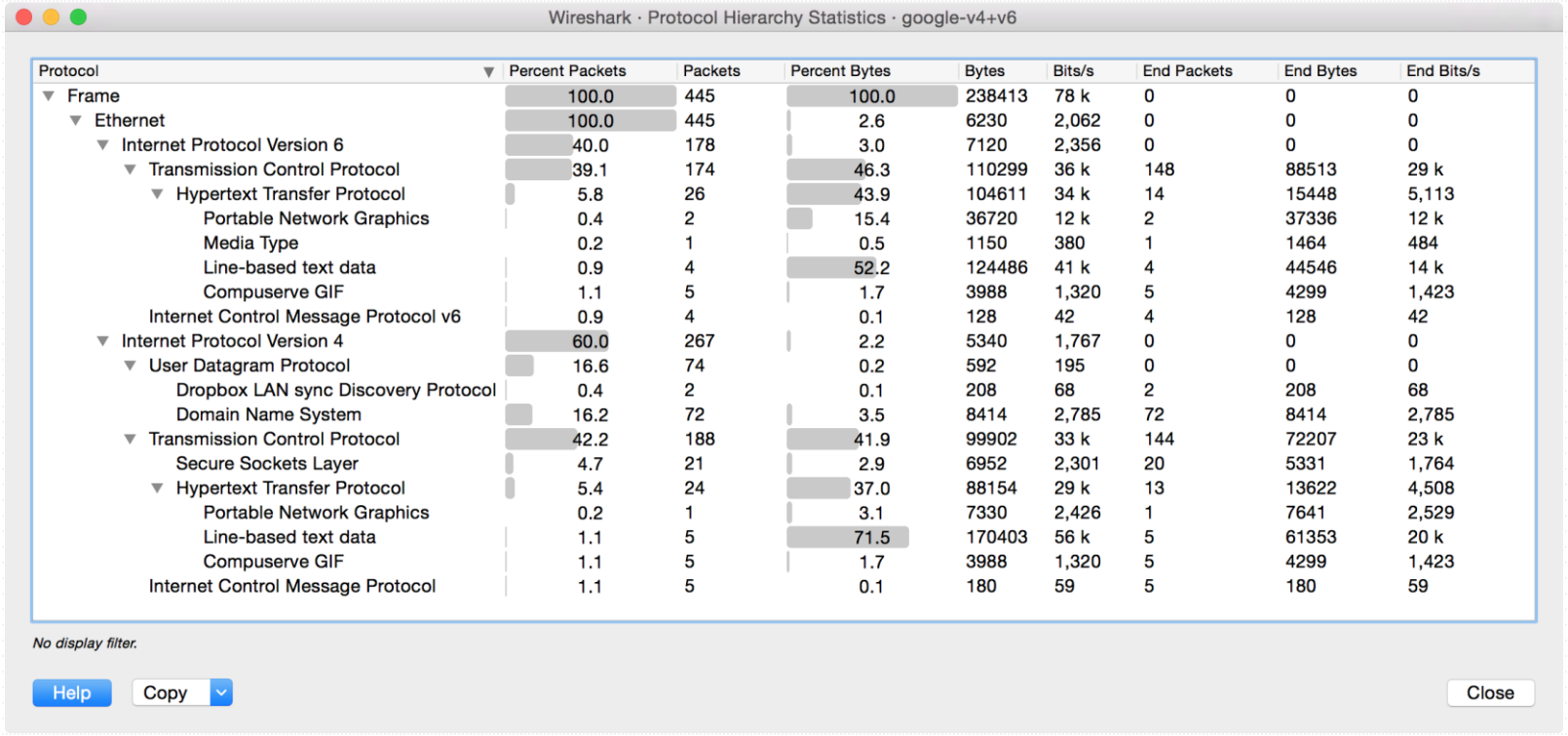




Wireshark (Ethereal)

- Statistics → Protocol Hierarchy

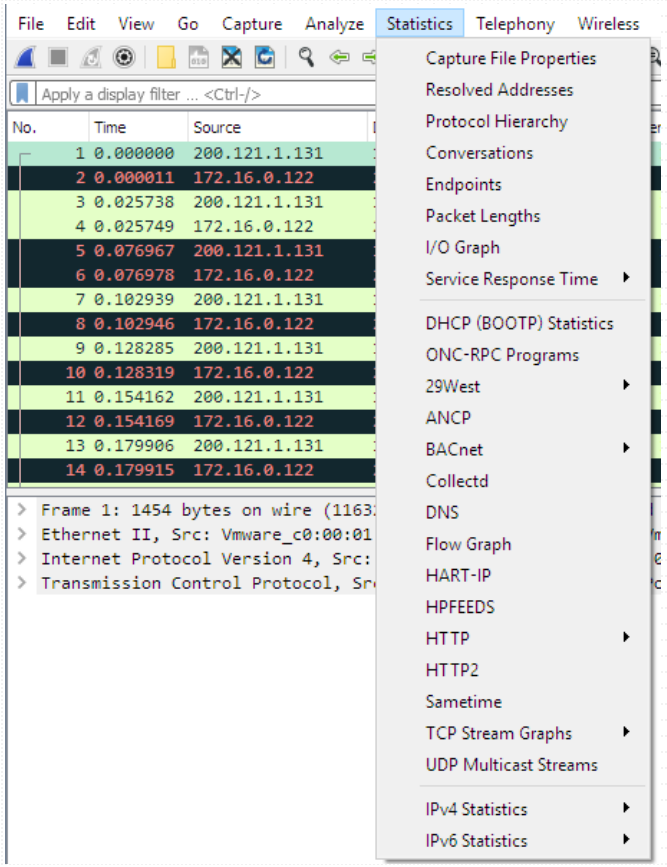


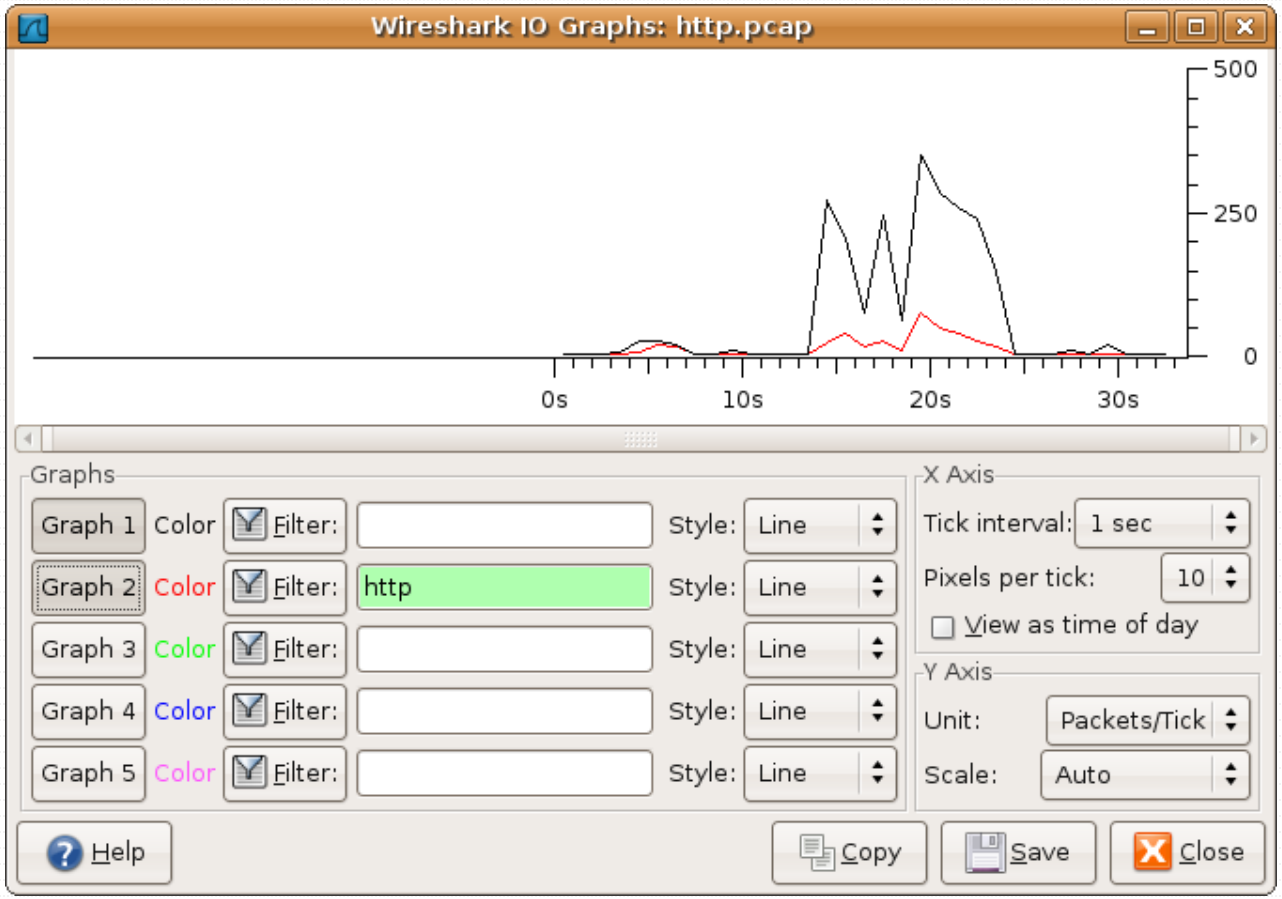


Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	445	100.0	238413	78 k	0	0	0
Ethernet	100.0	445	2.6	6230	2,062	0	0	0
Internet Protocol Version 6	40.0	178	3.0	7120	2,356	0	0	0
Transmission Control Protocol	39.1	174	46.3	110299	36 k	148	88513	29 k
Hypertext Transfer Protocol	5.8	26	43.9	104611	34 k	14	15448	5,113
Portable Network Graphics	0.4	2	15.4	36720	12 k	2	37336	12 k
Media Type	0.2	1	0.5	1150	380	1	1464	484
Line-based text data	0.9	4	52.2	124486	41 k	4	44546	14 k
Compuserve GIF	1.1	5	1.7	3988	1,320	5	4299	1,423
Internet Control Message Protocol v6	0.9	4	0.1	128	42	4	128	42
Internet Protocol Version 4	60.0	267	2.2	5340	1,767	0	0	0
User Datagram Protocol	16.6	74	0.2	592	195	0	0	0
Dropbox LAN sync Discovery Protocol	0.4	2	0.1	208	68	2	208	68
Domain Name System	16.2	72	3.5	8414	2,785	72	8414	2,785
Transmission Control Protocol	42.2	188	41.9	99902	33 k	144	72207	23 k
Secure Sockets Layer	4.7	21	2.9	6952	2,301	20	5331	1,764
Hypertext Transfer Protocol	5.4	24	37.0	88154	29 k	13	13622	4,508
Portable Network Graphics	0.2	1	3.1	7330	2,426	1	7641	2,529
Line-based text data	1.1	5	71.5	170403	56 k	5	61353	20 k
Compuserve GIF	1.1	5	1.7	3988	1,320	5	4299	1,423
Internet Control Message Protocol	1.1	5	0.1	180	59	5	180	59

Wireshark (Ethereal)

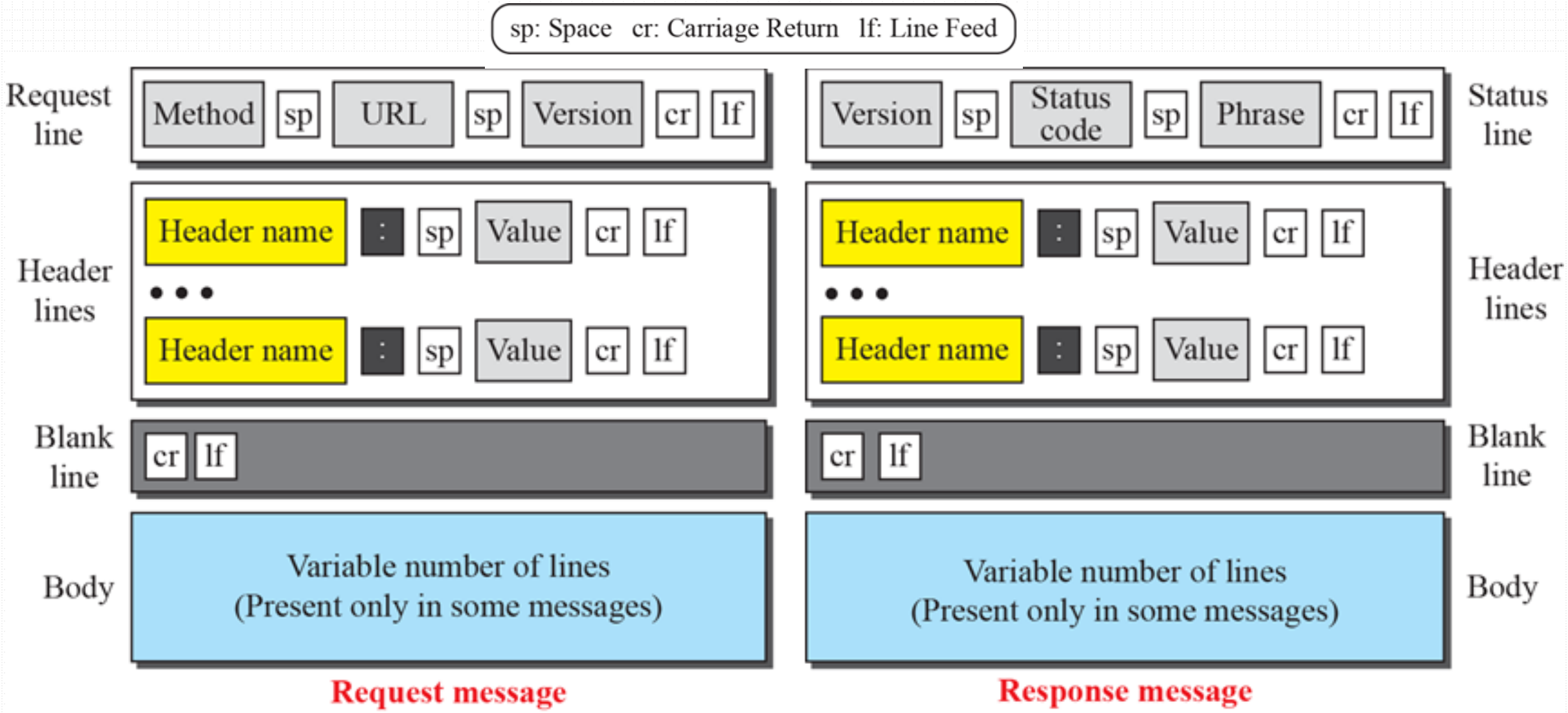
- Statistics → I/O Graph (throughput)
  - How about loss, delay, and jitter?





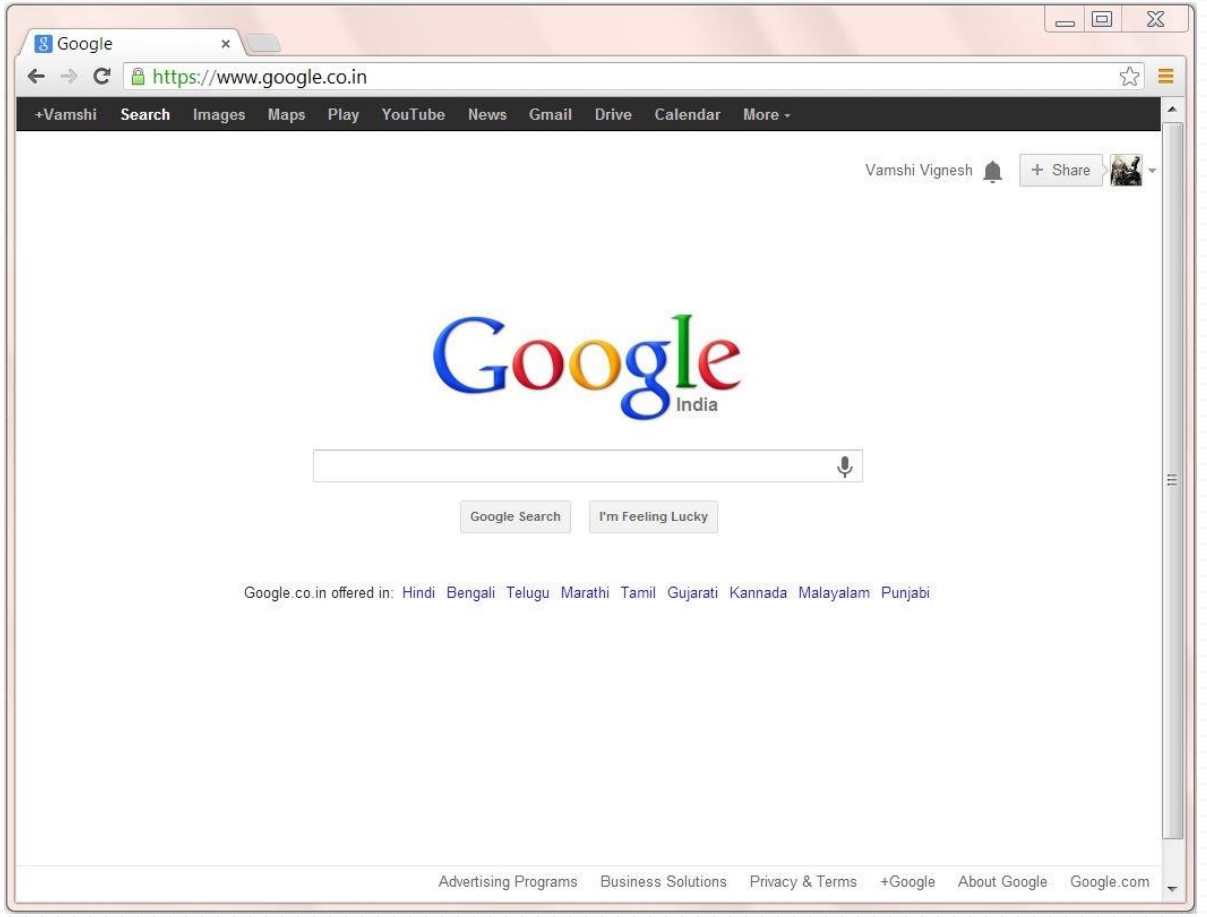
HTTP Packet Format

圖片來源：Computer Networks - A Top Down Approach，McGraw-Hill出版



Google Chrome Browser

- 由Google開發的免費網頁瀏覽器
  - 對應之開放原始碼計劃名為Chromium
  - Google Chrome本身是非自由軟體，未開放原始碼
- 官方網站：https://www.google.com.tw/chrome/



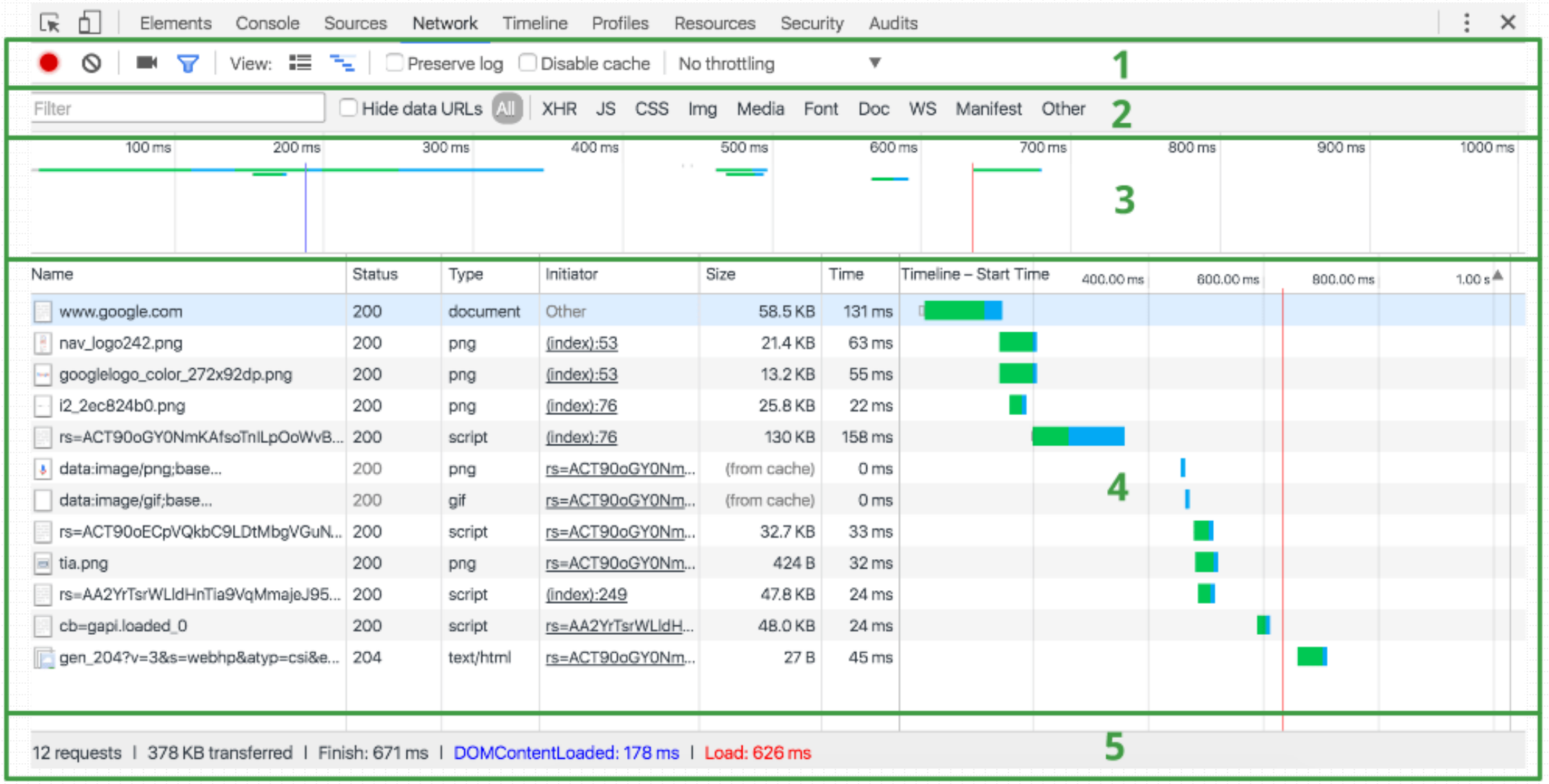
Google Chrome Browser

- Chrome 開發人員工具
  - 更多工具 → 開發者工具
    - Network
      - 瞭解請求和下載的資源文件並優化網頁加載性能
      - 官方說明
        - https://developers.google.com/web/tools/chrome-devtools/network-performance/resource-loading?hl=zh-tw
        - https://developers.google.com/web/tools/chrome-devtools/network-performance/understanding-resource-timing?hl=zh-tw
        - https://developers.google.com/web/tools/chrome-devtools/network-performance/network-conditions?hl=zh-tw

Google Chrome Browser

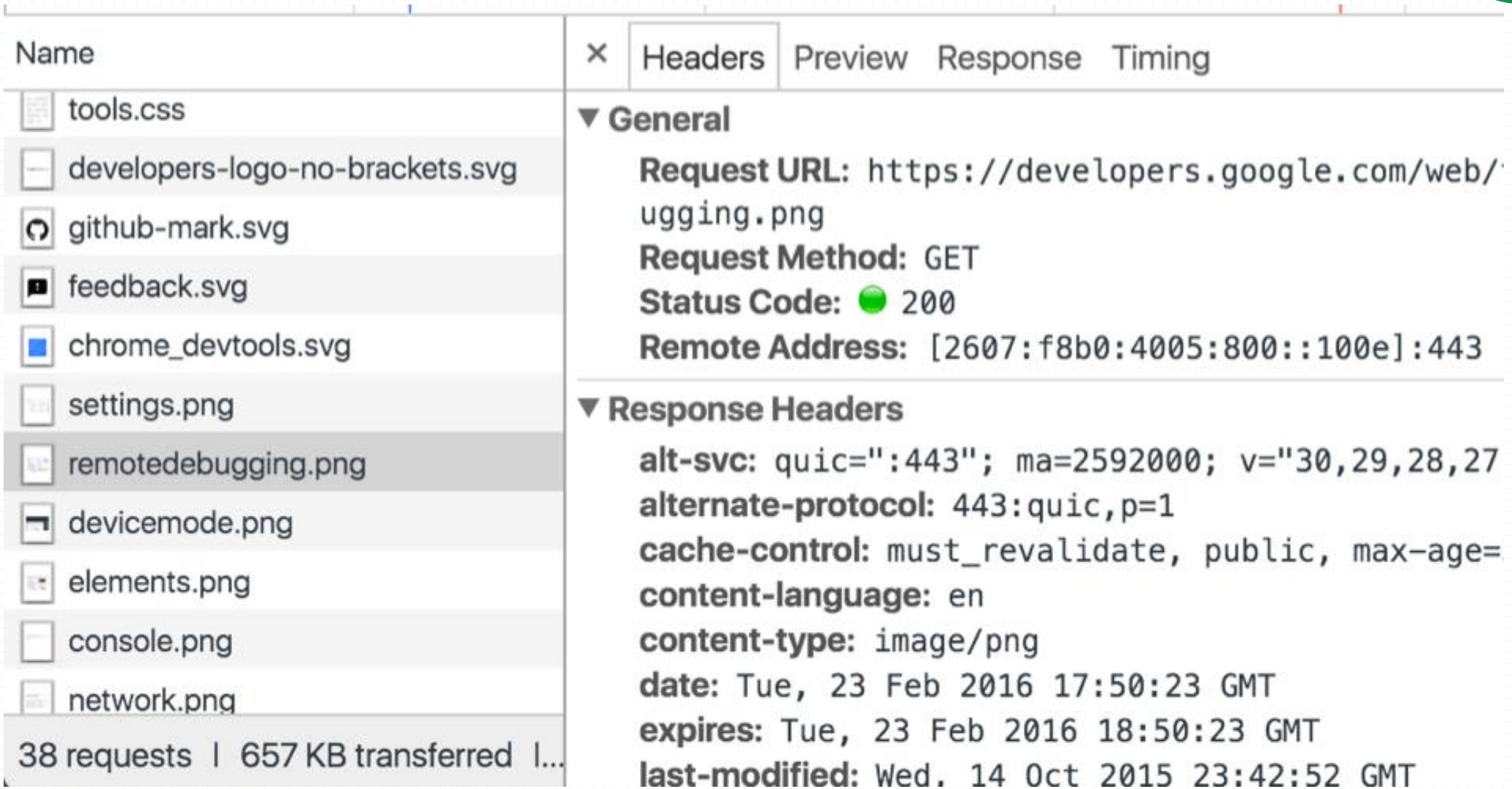
- Network 面板由五個窗格組成：
  1. Controls
     - 使用這些選項可以控制 Network 面板的外觀和功能。
  2. Filters
     - 使用這些選項可以控制在 Requests Table 中顯示哪些資源。提示：按住 Cmd (Mac) 或 Ctrl (Windows/Linux) 並點擊過濾器可以同時選擇多個過濾器。
  3. Overview
     - 此圖表顯示了資源檢索時間的時間線。如果您看到多條豎線堆疊在一起，則說明這些資源被同時檢索。
  4. Requests Table
     - 此表格列出了檢索的每一個資源。默認情況下，此表格按時間順序排序，最早的資源在頂部。點擊資源的名稱可以顯示更多信息。提示：右鍵點擊 Timeline 以外的任何一個表格標題可以添加或移除信息列。
  5. Summary
     - 此窗格可以一目瞭然地告訴您請求總數、傳輸的數據量和加載時間。

Google Chrome Browser



圖片來源：https://developers.google.com/web/tools/chrome-devtools/network-performance/resource-loading?hl=zh-tw

Google Chrome Browser



圖片來源：https://developers.google.com/web/tools/chrome-devtools/network-performance/resource-loading?hl=zh-tw

查核點：網頁還原

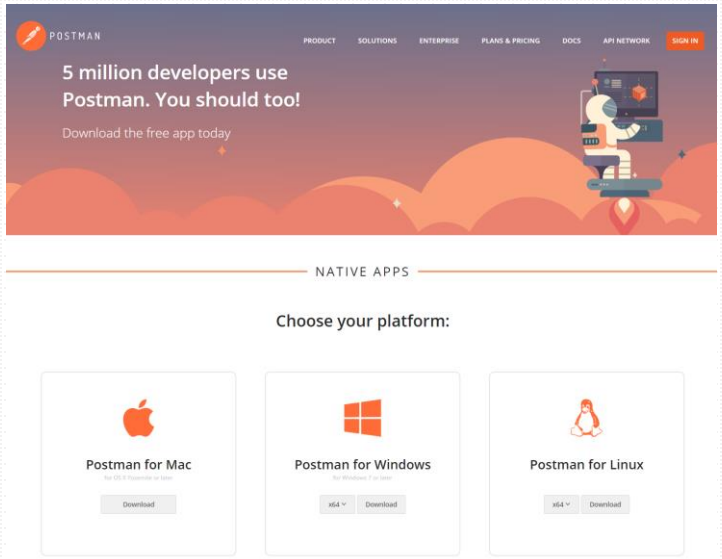
- 目標
  - 請任選一個網站，使用Chrome開啟該網站過程中，利用Wireshark擷取HTTP封包後，從中還原出下載的網頁檔案。
- 檢查項目
  - 使用Chrome開發人員工具顯示網頁下載的HTTP Header
  - 使用Wireshark顯示網頁下載的HTTP Header
  - 顯示利用Wireshark還原的網頁檔案

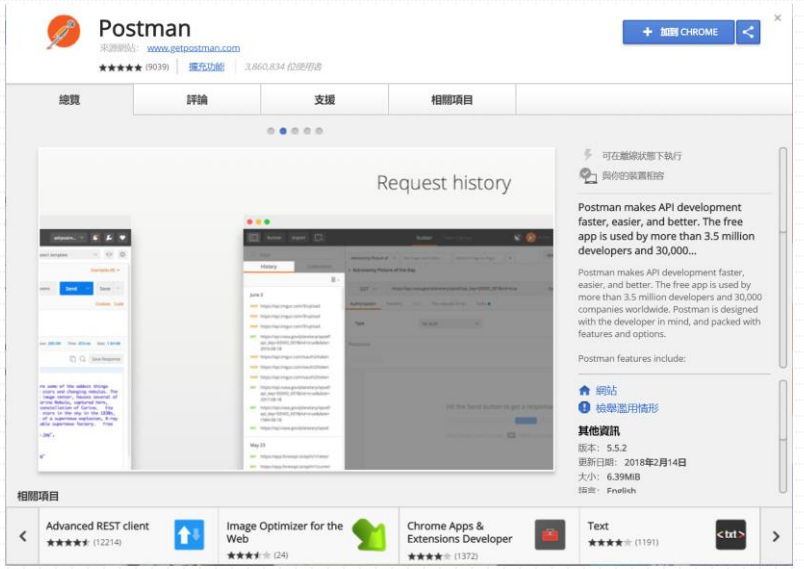
查核點：影片還原

- 目標：
  - 請任選一個具有URL的視訊檔案
    - 如：http://140.124.70.120/course/video.html 內有多個video
  - 使用Chrome開啟視訊檔案的URL，利用Wireshark擷取HTTP封包後，從中還原出下載的視訊檔案。
- 檢查項目：
  - 播放還原的視訊檔案

Postman

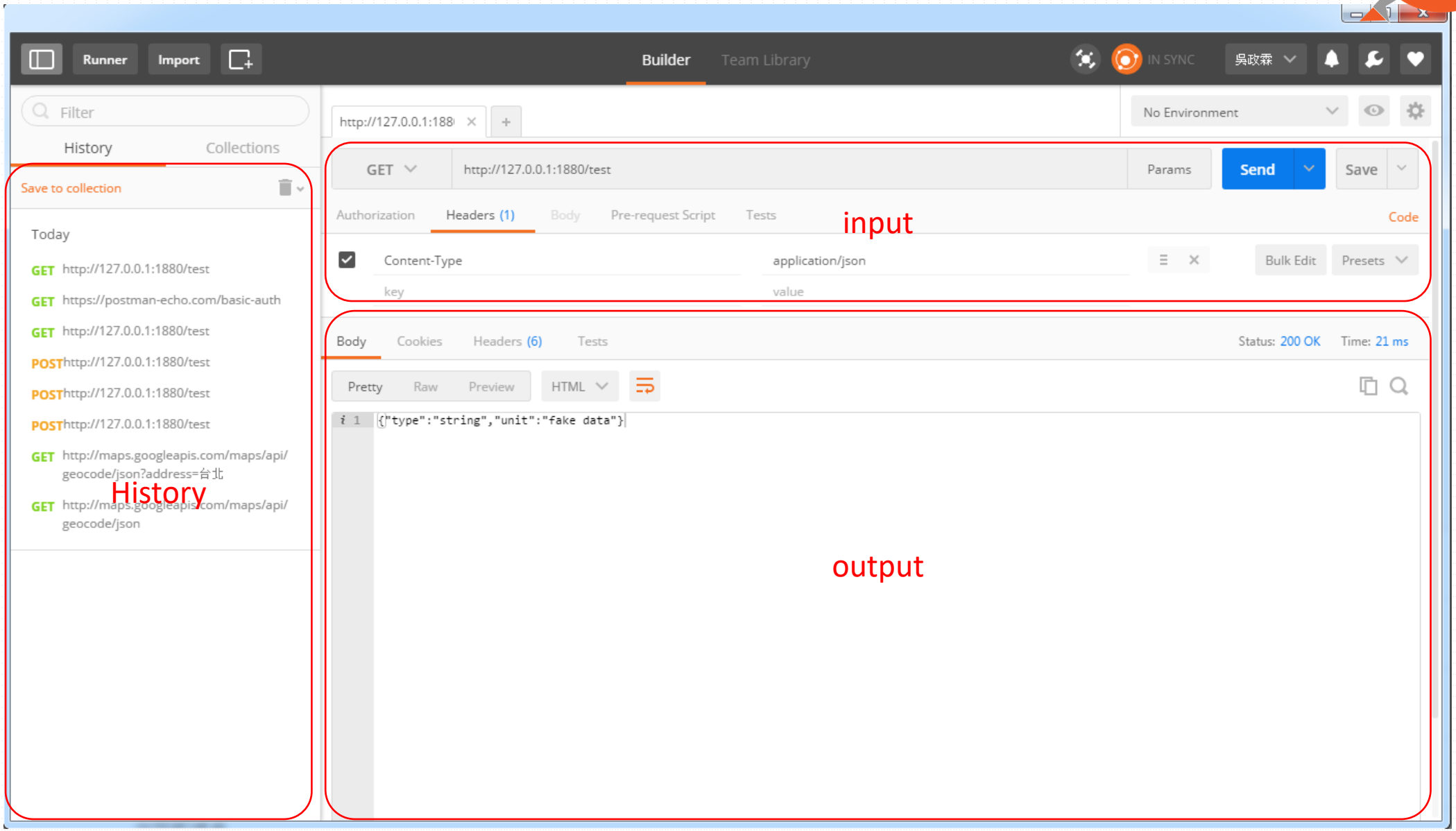
- 用來測試 Web Service的方便工具，可快速產生複雜之 HTTP Requests
  - 可選安裝可執行檔，或Chrome的擴充元件





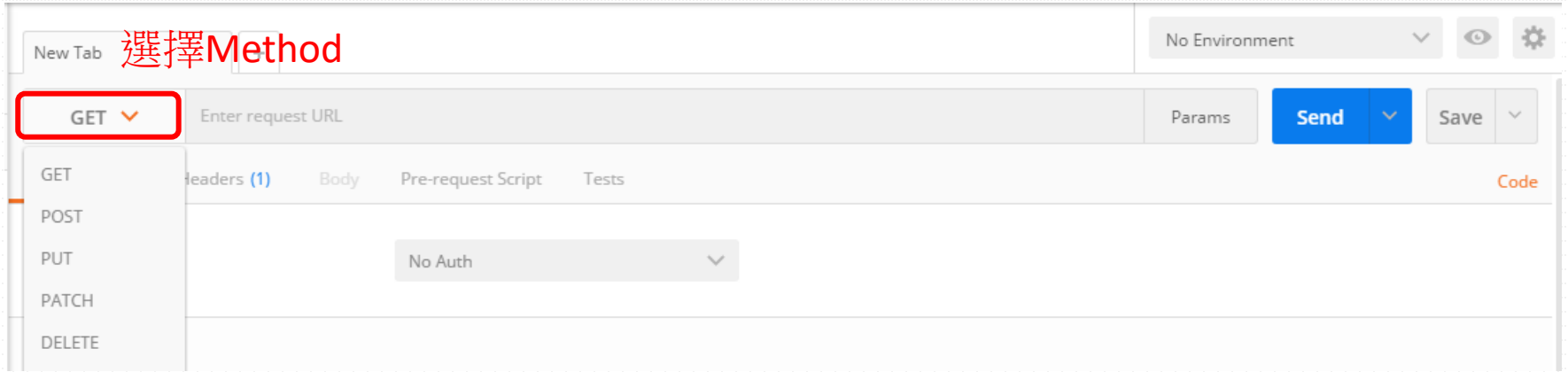
- 官方網站：https://www.getpostman.com/
- 官方說明：https://www.getpostman.com/docs/v6/

Postman

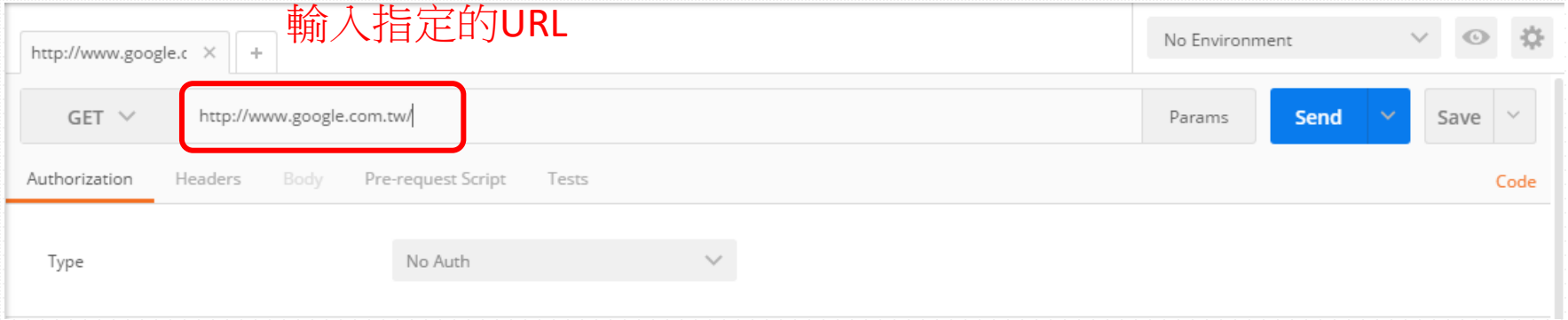

Runner
Import
Builder
Team Library
IN SYNC
吳政霖
Filter
History
Collections
Save to collection
Today
GET http://127.0.0.1:1880/test
GET https://postman-echo.com/basic-auth
GET http://127.0.0.1:1880/test
POST http://127.0.0.1:1880/test
POST http://127.0.0.1:1880/test
POST http://127.0.0.1:1880/test
GET http://maps.googleapis.com/maps/api/geocode/json?address=台北
GET http://maps.googleapis.com/maps/api/geocode/json
History
http://127.0.0.1:1880
No Environment
GET
http://127.0.0.1:1880/test
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Code
input
Content-Type
application/json
Bulk Edit
Presets
key
value
Body
Cookies
Headers (6)
Tests
Status: 200 OK
Time: 21 ms
Pretty
Raw
Preview
HTML
1 {"type":"string","unit":"fake data"}
output

Postman

- 設定HTTP Request - Method

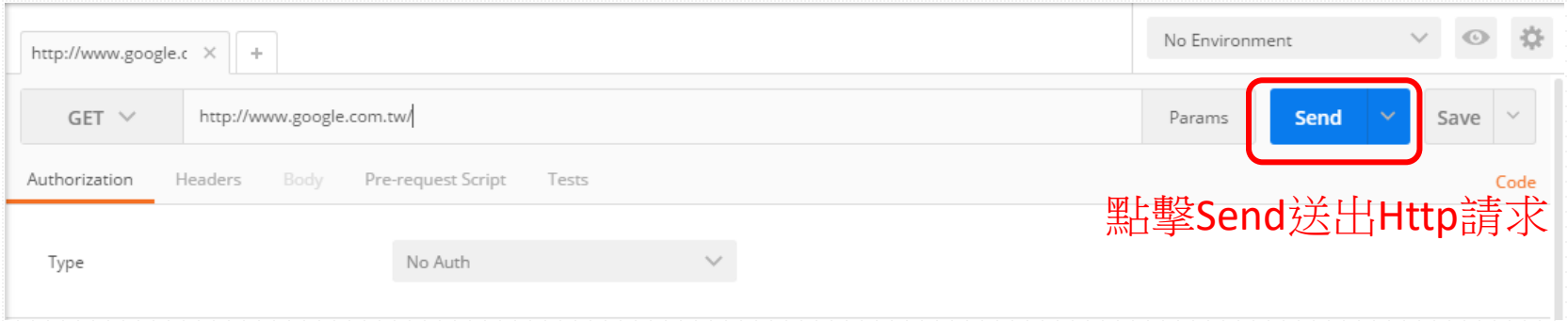


- 設定HTTP Request - URL

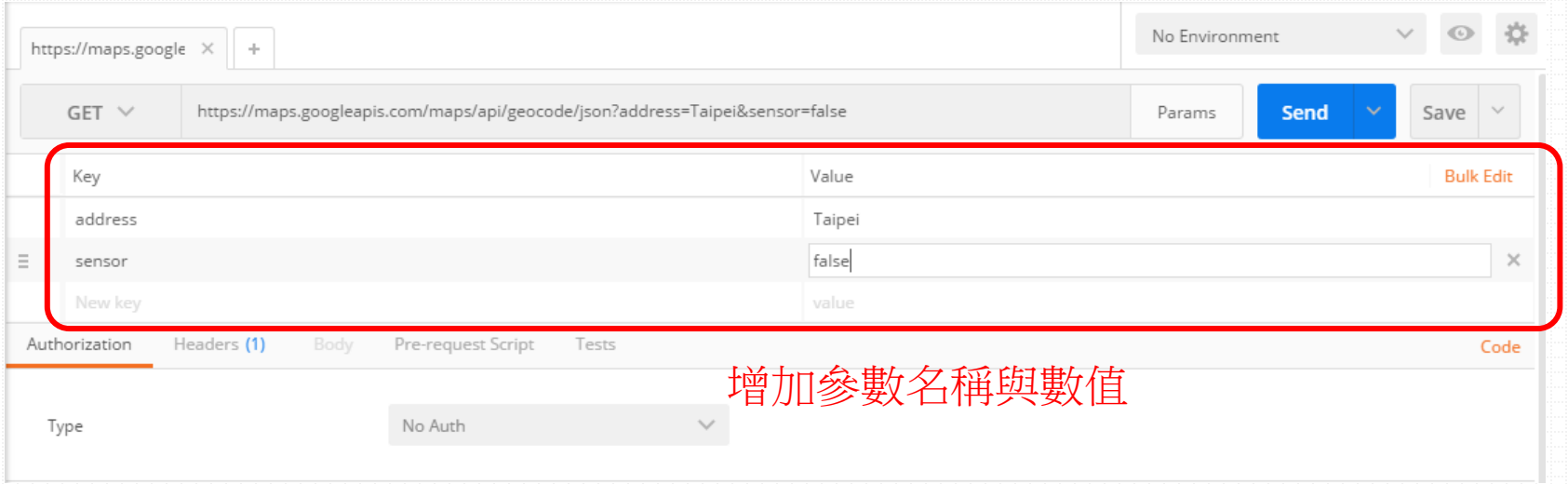


Postman

- 送出HTTP Request



- [若有需求] 設定HTTP Request - 參數



Postman

- 察看HTTP Response - Body

選擇Body

HTTP狀態碼、響應時間

Postman

- 察看HTTP Response - Header

選擇Headers

Body Cookies Headers (10) Tests Status: 200 OK Time: 360 ms

alt-svc → quic=":443"; ma=2592000; v="37,36,35"

cache-control → private, max-age=0

content-encoding → gzip

content-type → text/html; charset=UTF-8 Body資料格式為html，編碼為utf-8

date → Sun, 23 Apr 2017 17:42:05 GMT

expires → -1

server → gws

status → 200 狀態碼為200(OK)

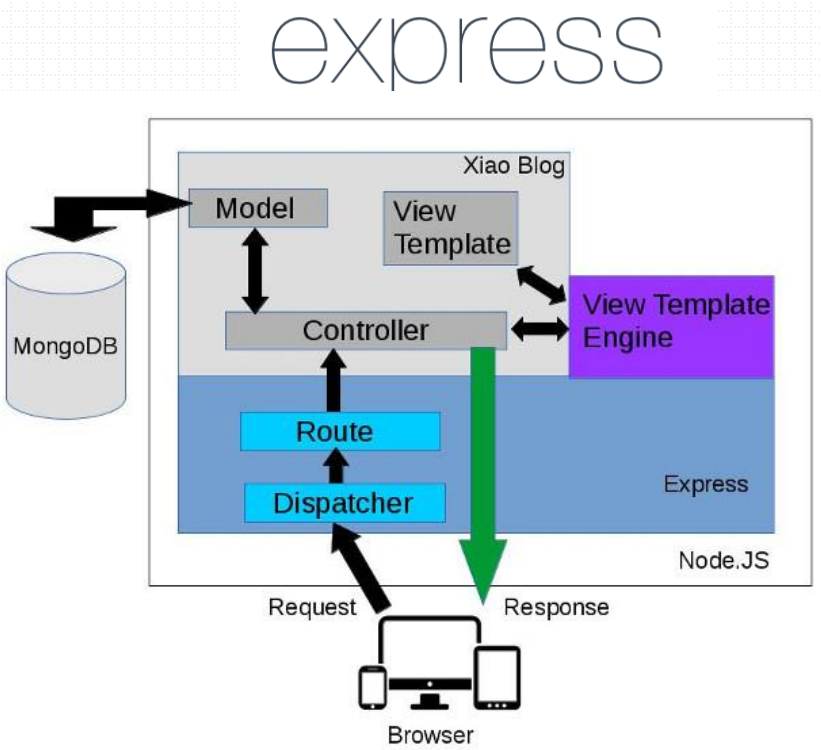
x-frame-options → SAMEORIGIN

x-xss-protection → 1; mode=block

HTTP response (Headers)

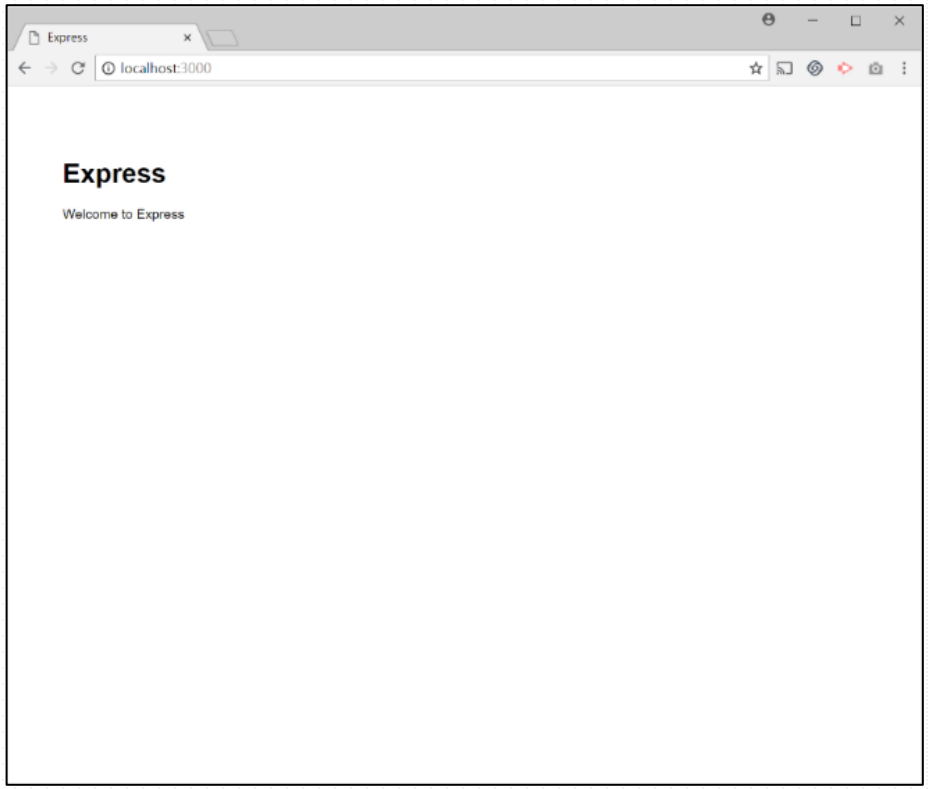
Express

- Express是最小又靈活的Node.js Web應用程式架構，為Web與行動式應用程式提供一組健全的特性。
  - http://expressjs.com/
- 大量的HTTP公用程式方法與中介軟體供您支配，能夠快速又輕鬆的建立完整的API。



Express-Generator

- Express提供快速建立應用程式架構的工具
  1. npm install express-generator -g
  2. express myExpressApp
  3. cd myExpressApp
  4. npm install
  5. npm start
  6. 用瀏覽器打開http://localhost:3000
  7. Ctrl+C可以中斷指令執行



基礎路由(Basic Routing)

- http://expressjs.com/en/starter/basic-routing.html
- 基礎語法：app.METHOD(PATH, HANDLER)
  - app是express的實例(instance)。
  - METHOD是HTTP要求方法，如：get、post、put、delete。
  - PATH是伺服器上的路徑。
  - HANDLER是當路由相符時要執行的函數。
- 示範：

```
app.get('/', function (req, res)
{
  res.send('Hello World!');
});
```

路由器層次中介軟體(Router-level Middleware)

- http://expressjs.com/en/guide/using-middleware.html
- Express中介軟體(Middleware)
  - 代表有權存取(1)要求物件(req)、(2)回應物件(res)以及(3)呼叫下一個中介軟體函數的函數。

```
var app = express();
var router = express.Router();
// a middleware function with no mount path.
// This code is executed for every request to the router
router.use(function (req, res, next) {
      console.log('Time:', Date.now());
      next();
});
```

靜態檔案(Static Files)

- http://expressjs.com/en/starter/static-files.html
- 使用內建函式：express.static(root, [options])
  - 示範： `app.use(express.static('public'));`
  - 代表可以載入位於public目錄中的檔案
    - http://localhost:3000/images/kitten.jpg
    - http://localhost:3000/css/style.css
    - http://localhost:3000/js/app.js
    - http://localhost:3000/images/bg.png
    - http://localhost:3000/hello.html

參數處理

- https://flaviocopes.com/express-request-parameters/
- 若使用GET，參數夾帶在URL中，則使用req.query物件

```
GET /test?name=fred&tel=0926xxx572

app.get('/test', function(req, res) {
        console.log(req.query.name);
        console.log(req.query.tel);
});
```

參數處理

- https://flaviocopes.com/express-request-parameters/
- 若使用POST，參數夾帶在HTTP封包中，則使用req.body物件

```
<form action='/test' method='post’>
    <input type='text' name='name' value='fred’>
    <input type='text' name='tel' value='0926xxx572’>
    <input type='submit' value='Submit’>
</form>

app.post('/test', function(req, res) {
    console.log(req.query.id);
    console.log(req.body.name);
    console.log(req.body.tel);
});
```

查核點：協定追蹤

- 目標：
  - 利用Express設計出自己的REST/RESTful API (可接收表單參數)
  - 利用Postman送出HTTP Request 附帶表單參數
  - 利用Wireshark擷取封包後，找出Express回傳的資料
- 查核項目：
  - 使用Wireshark顯示回傳的資料內容

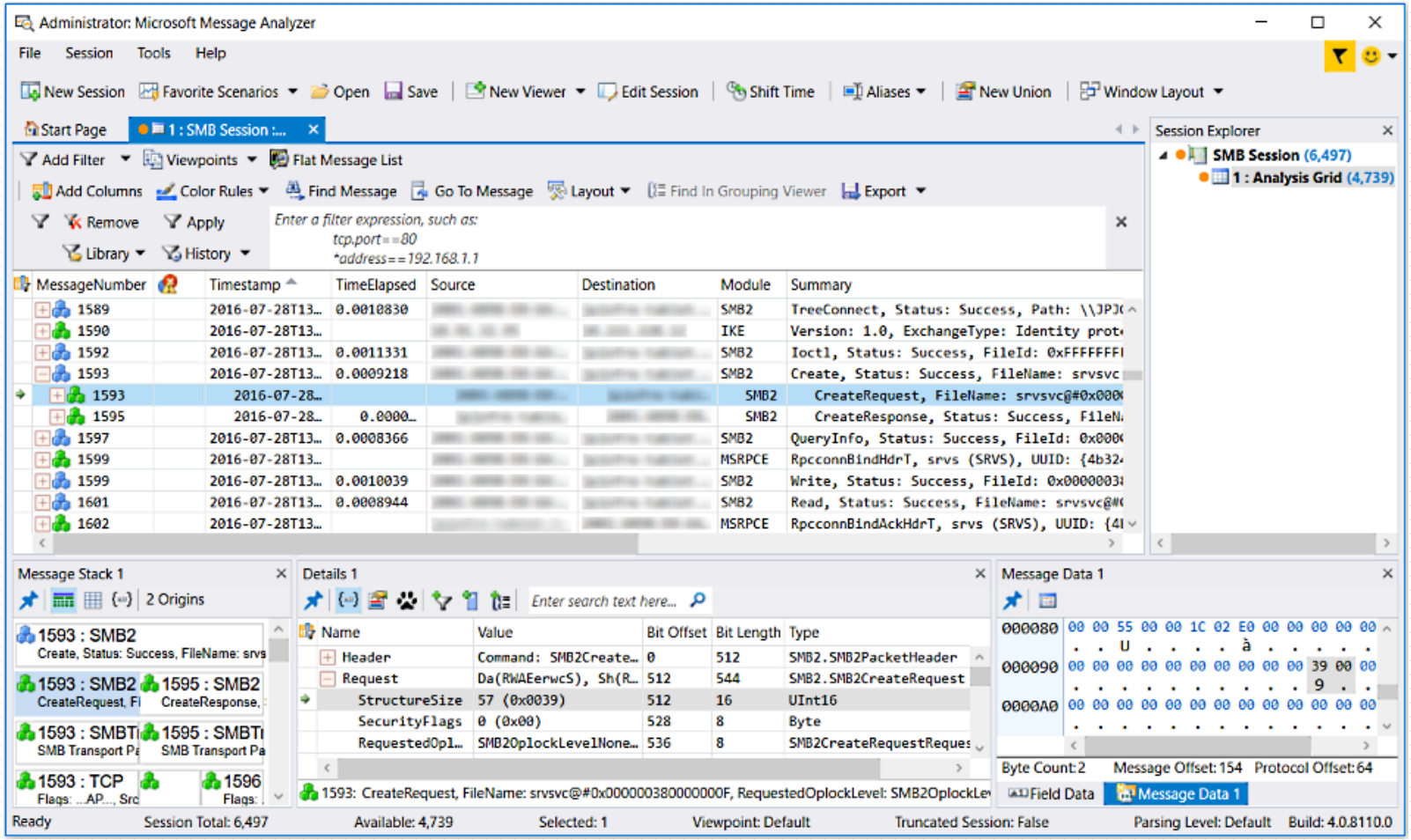
自行探索

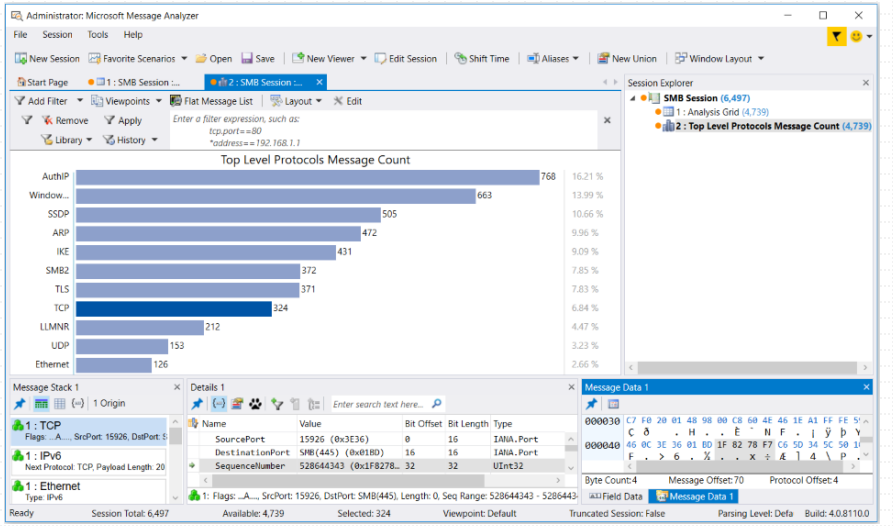
NTUT NESL

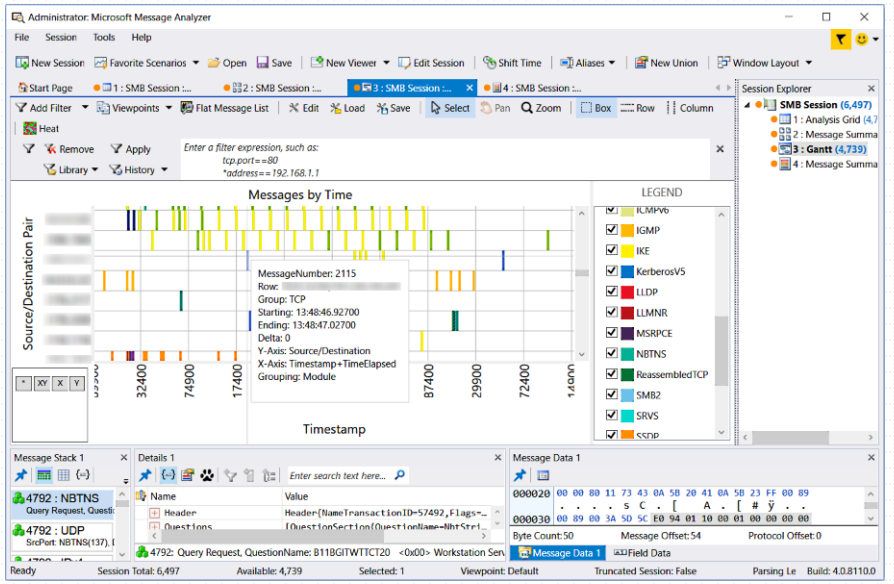
MS Message Analyzer (Network Monitor)

- A new tool for capturing, displaying, and analyzing protocol messaging traffic, events, and other system or application messages in network troubleshooting and other diagnostic scenarios.
  - It is the successor to Microsoft Network Monitor 3.4 and is a key component in the Protocol Engineering Framework (PEF).
- 下載網址：https://www.microsoft.com/en-us/download/details.aspx?id=44226
- 官方說明：https://docs.microsoft.com/en-us/message-analyzer/microsoft-message-analyzer-operating-guide

MS Message Analyzer (Network Monitor)







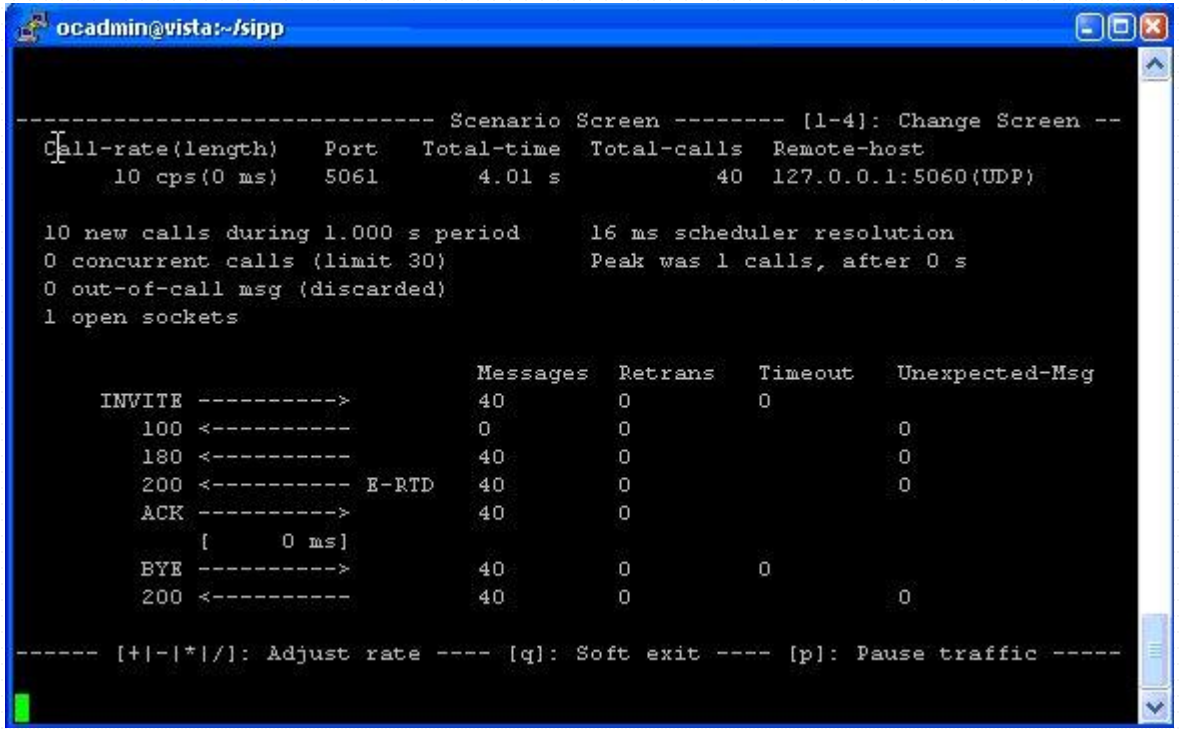
其他網路封包分析器

	Creator	Latest release	User interface	Software license	Cost
Analyze This	Comoe Networks		Web GUI	N/A	?
Cain and Abel	Massimiliano Montoro	4.9.56 / April 7, 2014	GUI	Freeware	Free
Capsa	Colasoft	10.0 / July 26, 2017[1]	GUI	Proprietary	$0-$995, depending on version[2]
Carnivore	Federal Bureau of Investigation	?	?	N/A	?
Charles Web Debugging Proxy	Karl van Randow	4.1.4 / July 10, 2017	GUI	?	$30-$50 (Free Trial)
Clarified Analyzer	Clarified Networks		GUI	Proprietary	Non-free
Clusterpoint Network Traffic Surveillance System	Clusterpoint		web GUI	Proprietary	?
CommView	TamoSoft	6.5	GUI	Proprietary	$299-$599, $149 1 year subscription
Debookee	iwaxx	6.0.0b2 (2278) / July 21, 2017[3]	GUI	Proprietary	$29.90-$69.90
dSniff	Dug Song	2.3 / December 17, 2000[4]	CLI	BSD License	Free
EtherApe	Juan Toledo	0.9.14 / February 6, 2016[5]	GUI	GNU General Public License	Free
Ettercap	ALoR and NaGA	0.8.2-Ferri / March 14, 2015[6]	Both	GNU General Public License	Free
Fiddler	Eric Lawrence	4.6.3.50306 / 9 December 2016	GUI	Freeware	Free
justniffer	The Justniffer team	0.5.15 / March 21, 2016[7]	CLI	GNU General Public License	Free
Kismet	Mike Kershaw (dragorn)	2016-01-R1 / January 31, 2016[8]	CLI	GNU General Public License	Free
Microsoft Message Analyzer	Microsoft	1.4 / October 28, 2016 [9]	GUI	Proprietary	Free
Microsoft Network Monitor	Microsoft	3.4 / June 24, 2010	GUI	Proprietary	Free
netsniff-ng	Daniel Borkmann	0.6.2 / November 7, 2016	CLI	GNU General Public License	Free
ngrep	Jordan Ritter	1.45 *(11/18/06)*	CLI	BSD-style	Free
Observer	Viavi Solutions (formerly Network Instruments)		GUI	Proprietary	Price on request
OmniPeek (formerly AiroPeek, EtherPeek)	Savvius (formerly WildPackets)	11.1 / November, 2017	GUI	Proprietary	$1194-$5994, depending on version[10]
SteelCentral Transaction Analyzer	OPNET Technologies/Riverbed Technology	17.0.T-PL1 / June 9, 2014[11]	GUI	Proprietary	Non-free
snoop	Sun Microsystems	Solaris 10 / December 11, 2006	CLI	CDDL	Free
tcpdump	The Tcpdump team	4.8.1 / October 25, 2016[12]	CLI	BSD License	Free
Tranalyzer	The Tranalyzer team	0.7.5 / February 10, 2018[13]	CLI	GNU General Public License	Free
Wireshark (formerly Ethereal)	The Wireshark team	2.4.5 / February 23, 2018[14]	Both	GNU General Public License	Free
Xplico	The Xplico team	1.2.0 / February 1, 2017[15]	Both	GNU General Public License	Free

圖片來源：https://en.wikipedia.org/wiki/Comparison_of_packet_analyzers

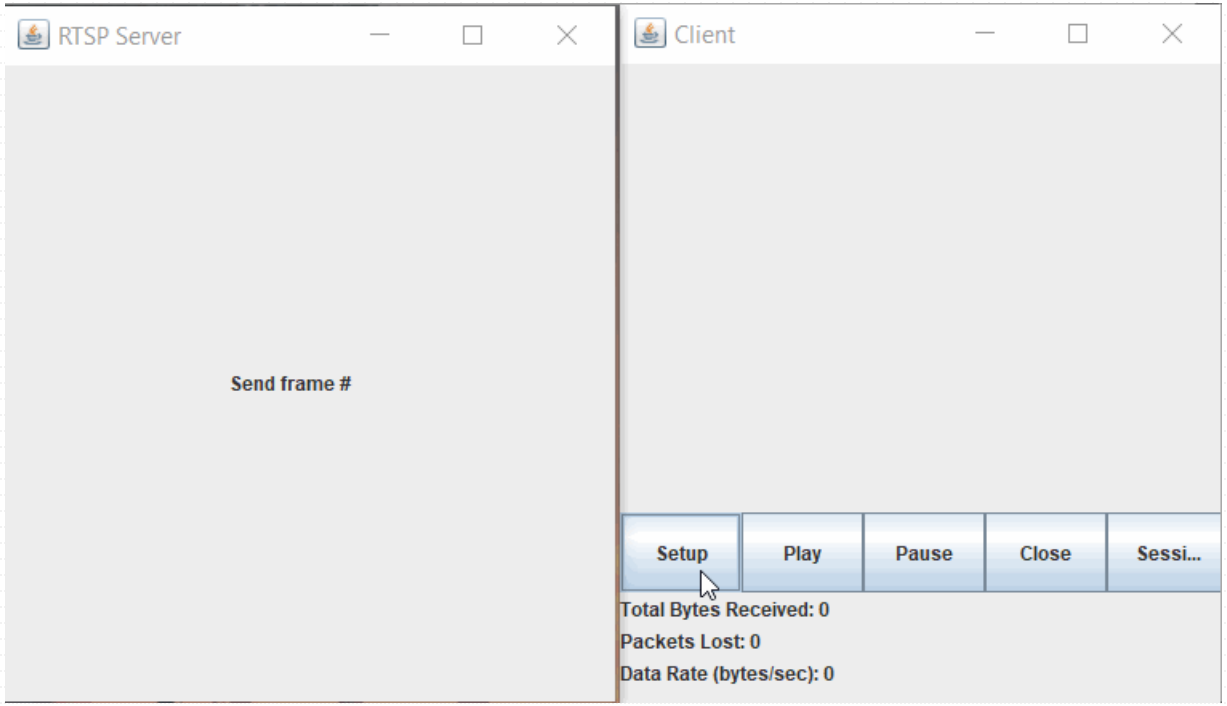
SIP

- SIPp
  - http://sipp.sourceforge.net/index.html
  - http://sipp-wip.readthedocs.io/en/latest/



RTP/RTCP/RTSP

- RTSP Client & Server
  - https://www.csee.umbc.edu/~pmundur/courses/CMSC691C/lab5-kurose-ross.html
  - https://github.com/mutaphore/RTSP-Client-Server



NTUT NESL

總結

- 網路封包分析器(如：Wireshark)可幫助具體了解網路封包傳遞與網路資料交換行為。
  - 封包標頭(Header)解析
  - 封包交換先後順序
  - 傳輸效能分析與統計