

# 淺談網路跟 HTTP

# 什麼是網路

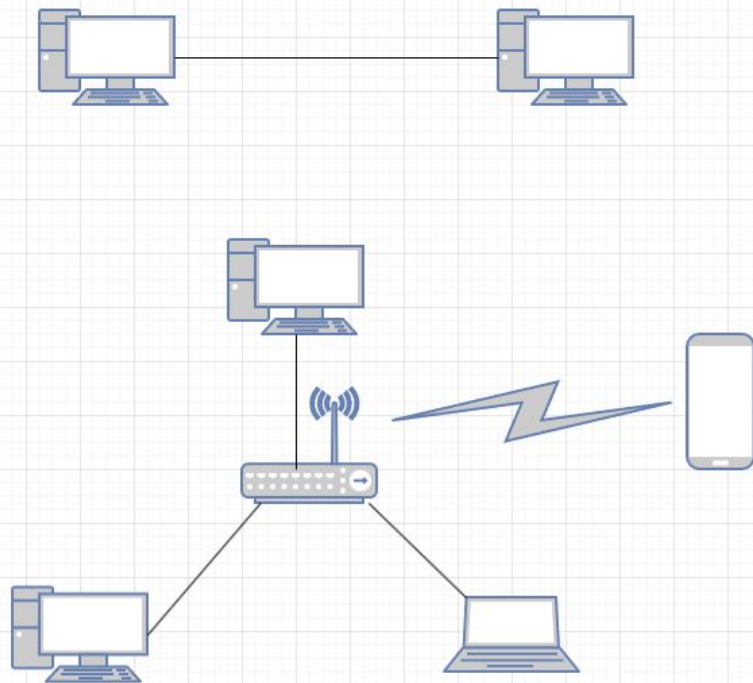
在日常生活中我們有許許多多的事情是離不開網路的，例如說看影片、玩遊戲、甚至於上課、交作業等等。

那到底什麼是網路呢？



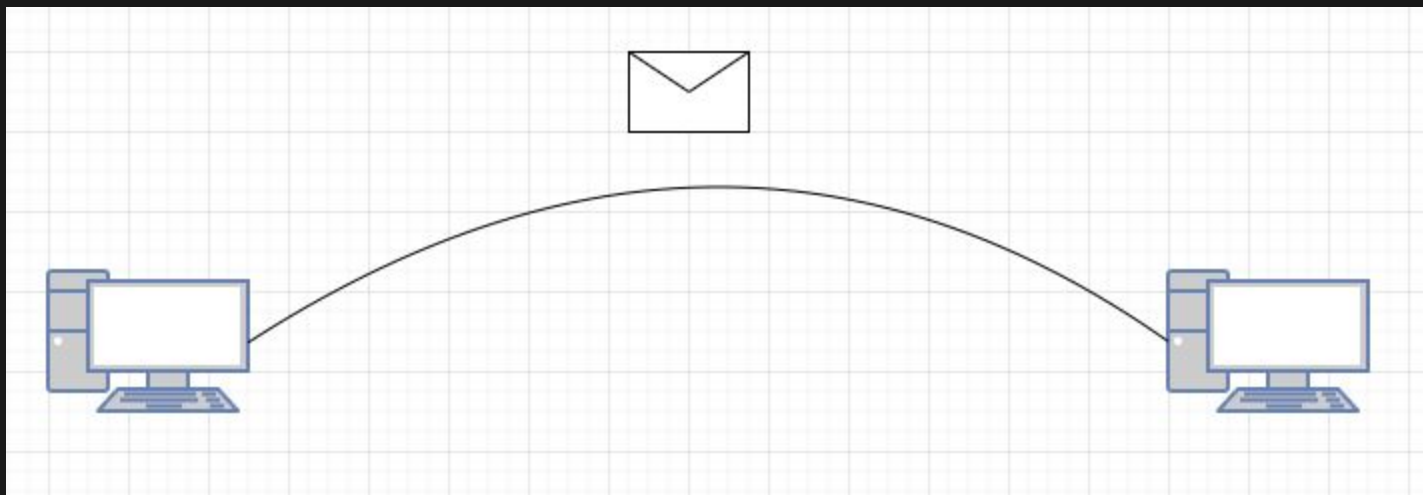
# 什麼是網路

事實上講白話一點，把很多台電腦用網路線或 WIFI 串聯在一起就是所謂的網路了。



# 什麼是網路

而在網路上傳輸資料可以比喻成送信。



# 什麼是網路

首先我們必須先撰寫我們信件的內容

嗨！最近如何？

# 什麼是網路

接這封進信封，並決定用掛號還是平信，寄給誰。

掛號信

王大壯 寄

李小美 收

丘如何？

# 什麼是網路

然後寫下寄件地址跟收件地址

台南市東區大學路1號

王大壯 寄

掛號信

台北市松山區南京東

路五段1號

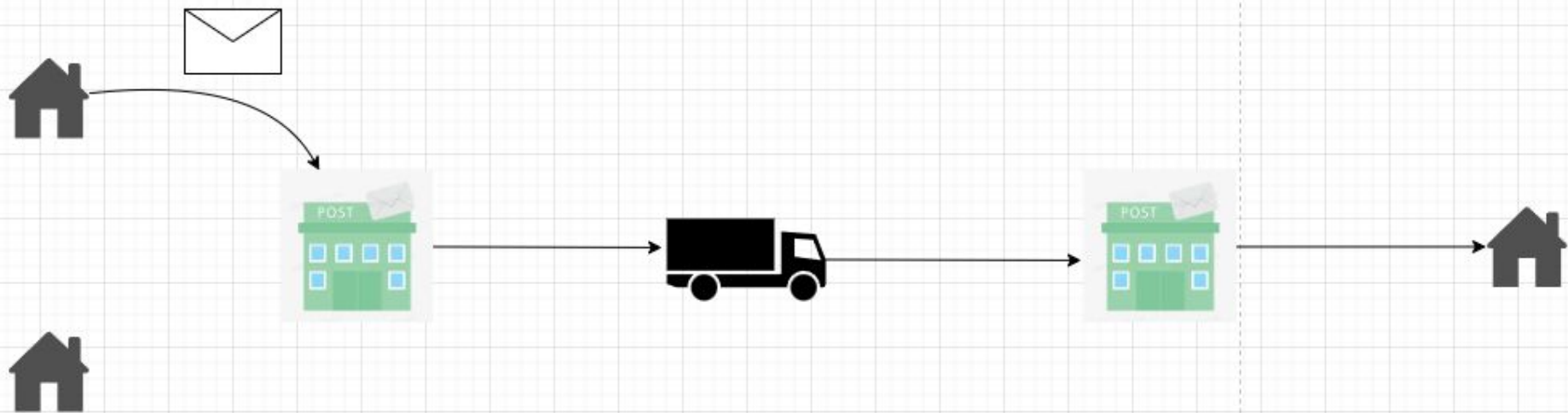
李小美 收

丘如何？

# 什麼是網路

接著將信交給郵局，由郵局轉送到目標地址區域的郵局再送給目標。

當然如果目標在你家隔壁，你可以直接送就好。





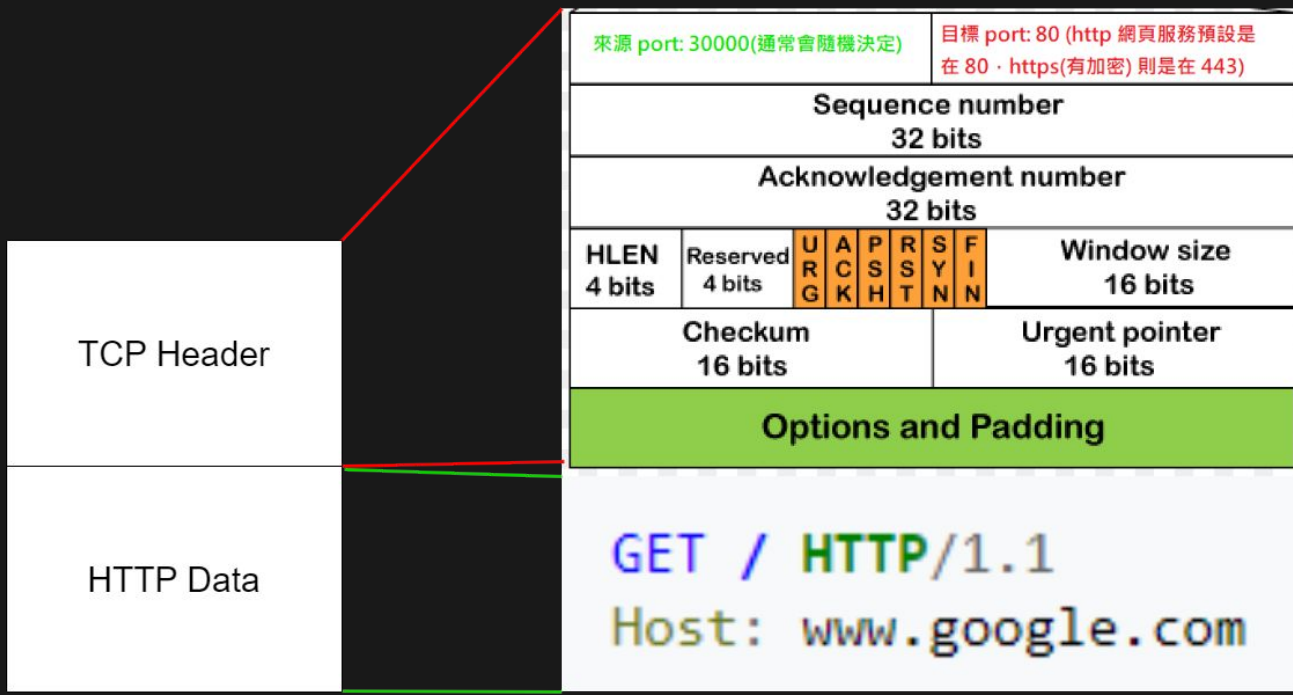
# 什麼是網路 -Application layer

而網路也是一樣的概念，以網頁為例，首先先撰寫內容

```
GET / HTTP/1.1  
Host: www.google.com
```

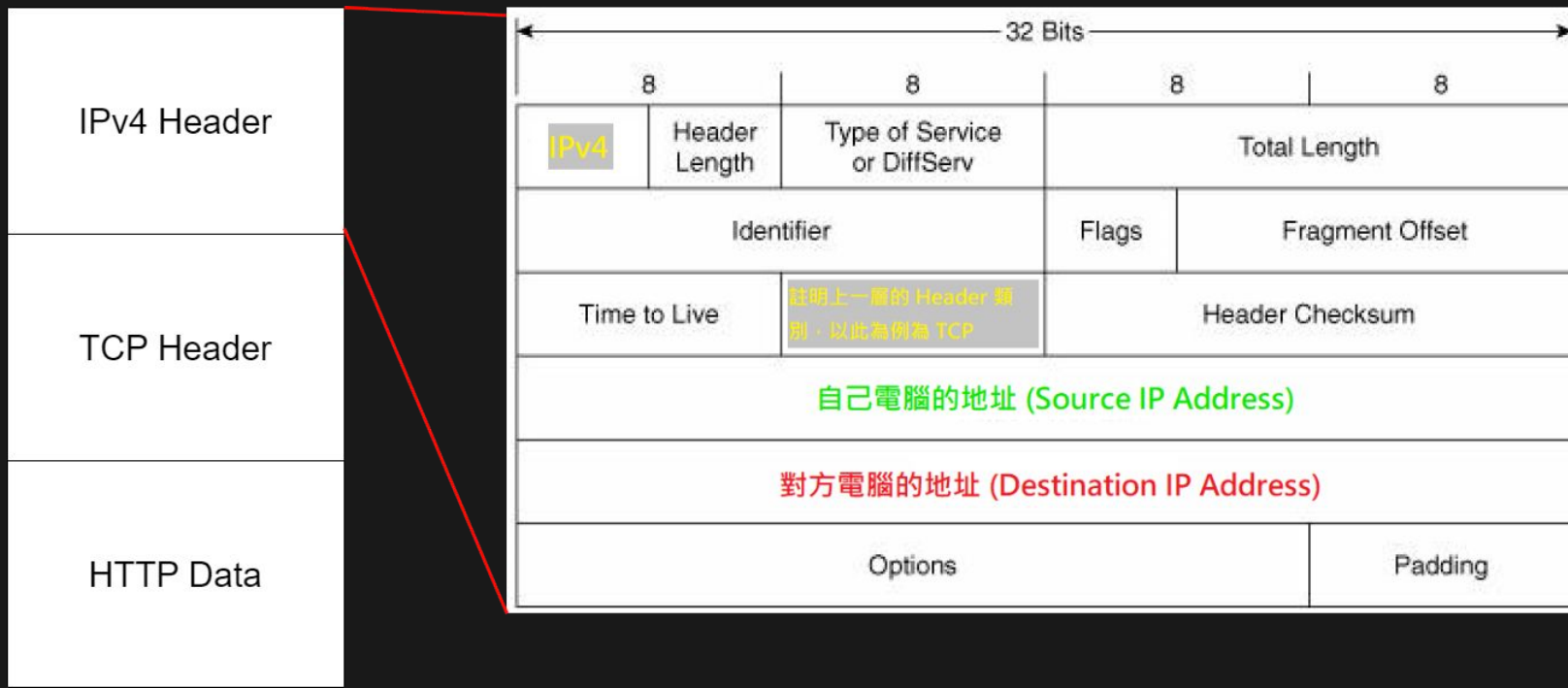
# 什麼是網路 -Transport layer

接著決定你現在要用 **TCP** 還是 UDP, 以及從自己電腦裡 **瀏覽器程式** 送到對方電腦裡的 **網頁伺服器程式**



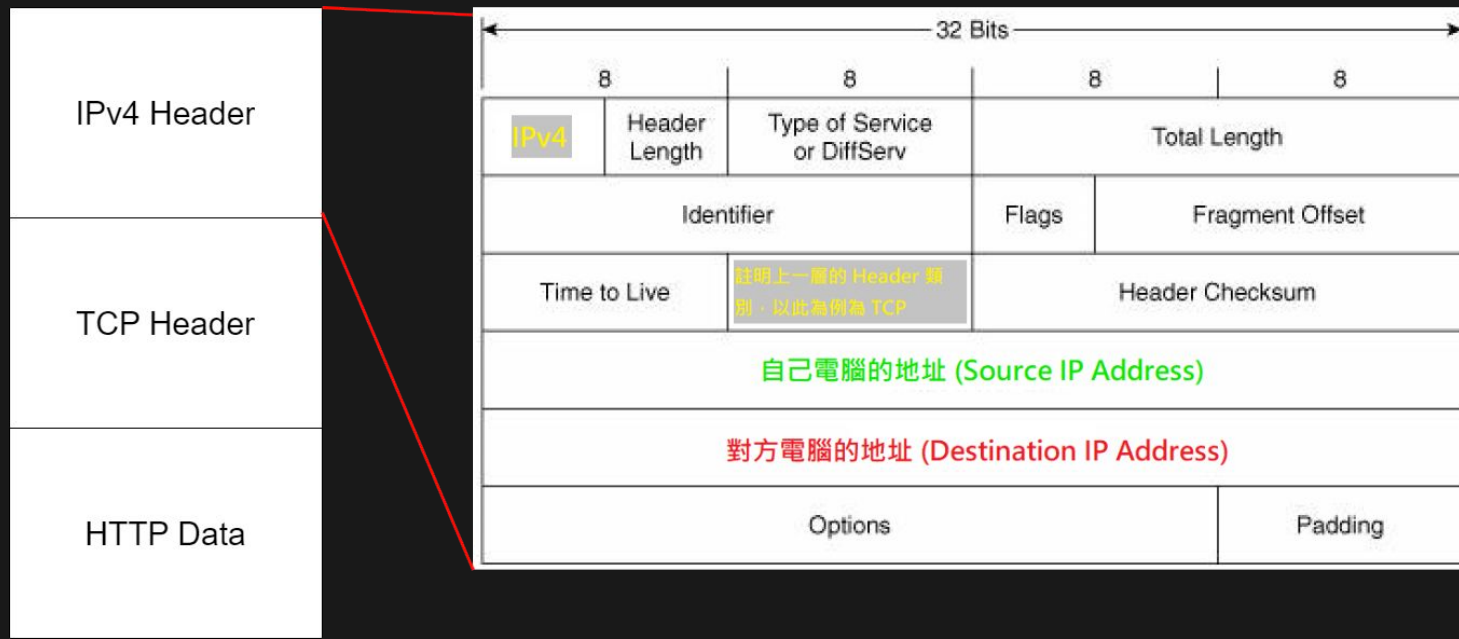
# 什麼是網路 -Network layer

接著決定地址的格式使用 **IPv4**，並寫下 **自己電腦的 IP Address** 與 **對方電腦的 IP Address**



# 什麼是網路 -Network layer

那在 **IPv4** 的地址格式中，也有一些特殊地址，如：**127.0.0.1** 這個地址對於任何電腦而言都**代表自己**。



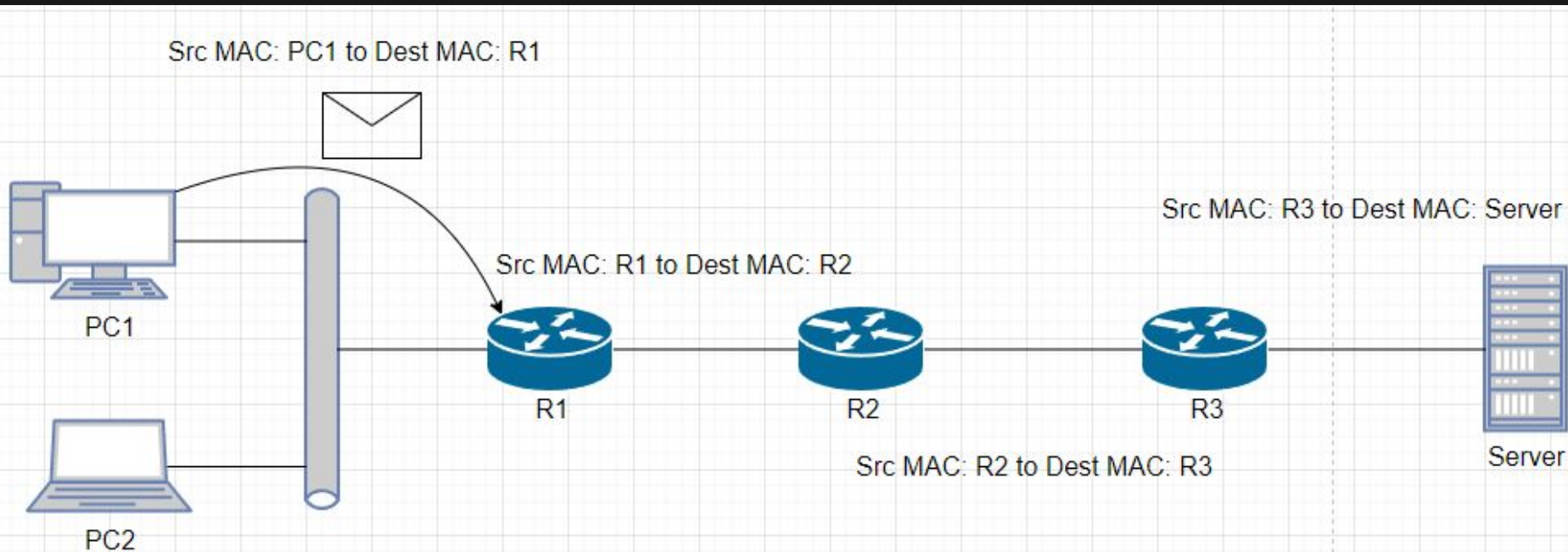
# 什麼是網路 -Link layer

最後決定要從哪一個 Router (可以想像成郵局) 寄信, 當然如果目標就在你目前的區域內, 就直接交給他就好。

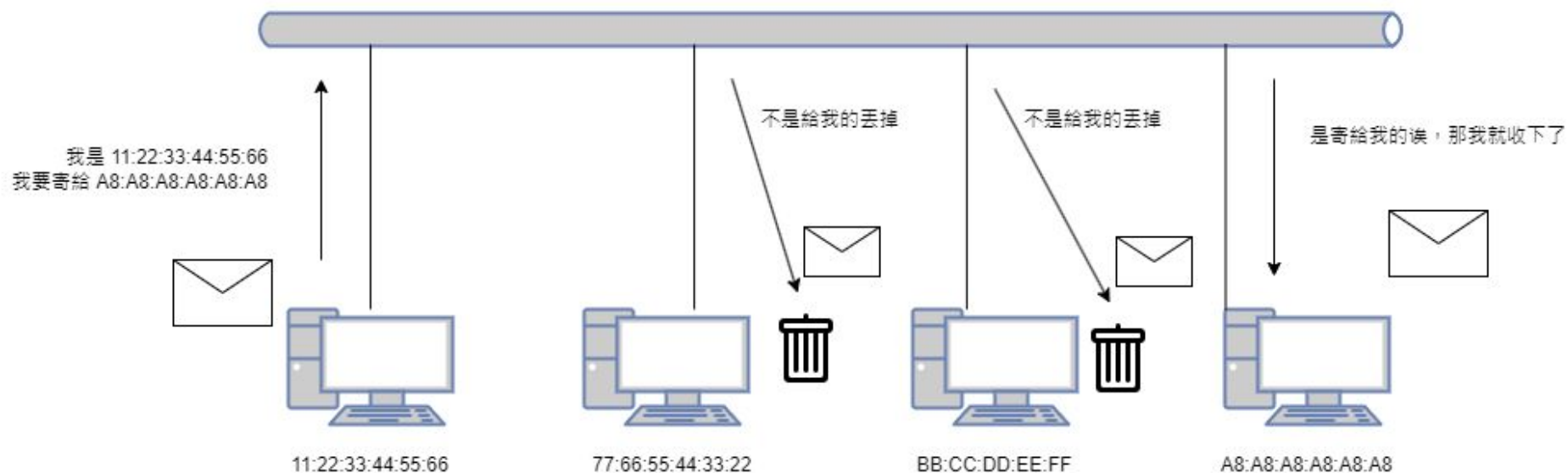


# 什麼是網路 -Link layer

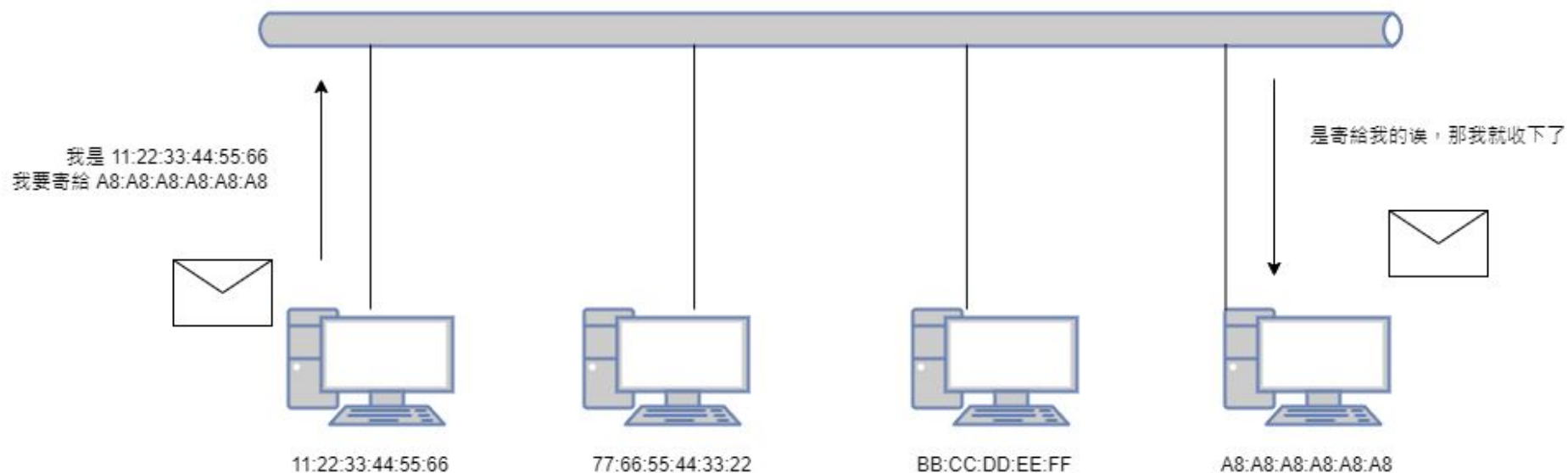
然後由 Router 轉送給其他 Router 直到到達目的地為止。



# 什麼是網路

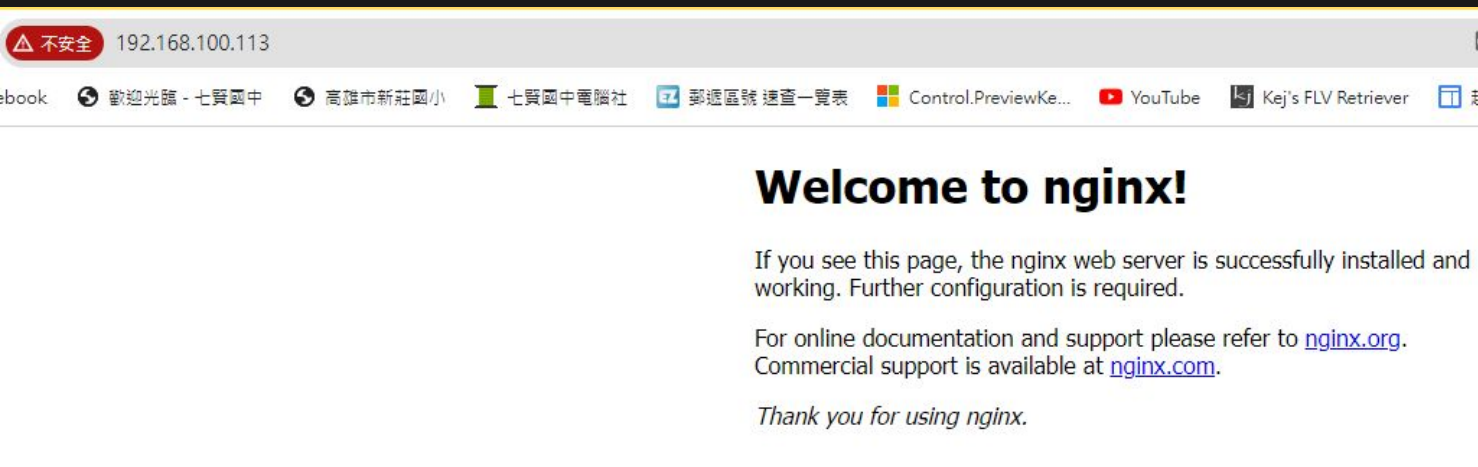


# 什麼是網路





# 什麼是網路



```
C:\Users\jimmy>ping 192.168.100.110
```

```
Ping 192.168.100.110 (使用 32 位元組的資料):  
回 覆 自 192.168.100.110: 位元組=32 時間<1ms TTL=64  
回 覆 自 192.168.100.110: 位元組=32 時間<1ms TTL=64  
回 覆 自 192.168.100.110: 位元組=32 時間<1ms TTL=64  
回 覆 自 192.168.100.110: 位元組=32 時間<1ms TTL=64
```

# 什麼是網路

1	0.000000	7e:dd:d6:94:09:6d	Broadcast	ARP	42 Who has 192.168.100.113? Tell 192.168.100.100
2	0.000150	e6:82:a0:97:43:45	7e:dd:d6:94:09:6d	ARP	60 192.168.100.113 is at e6:82:a0:97:43:45
3	0.000480	192.168.100.100	192.168.100.113	HTTP	1... GET / HTTP/1.1
4	0.000710	192.168.100.113	192.168.100.100	HTTP	9... HTTP/1.1 200 OK (text/html)

> Frame 3: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface \Device\NPF\_{D50E8A4D-2461-42C7-8F...}

> Ethernet II, Src: 7e:dd:d6:94:09:6d (7e:dd:d6:94:09:6d), Dst: e6:82:a0:97:43:45 (e6:82:a0:97:43:45)

> Internet Protocol Version 4, Src: 192.168.100.100, Dst: 192.168.100.113

> Transmission Control Protocol, Src Port: 22475, Dst Port: 80, Seq: 1, Ack: 1, Len: 78

▼ Hypertext Transfer Protocol

> GET / HTTP/1.1\r\n

Host: 192.168.100.113\r\n

User-Agent: curl/8.0.1\r\n

Accept: \*/\*\r\n

\r\n

[Full request URI: <http://192.168.100.113/>]

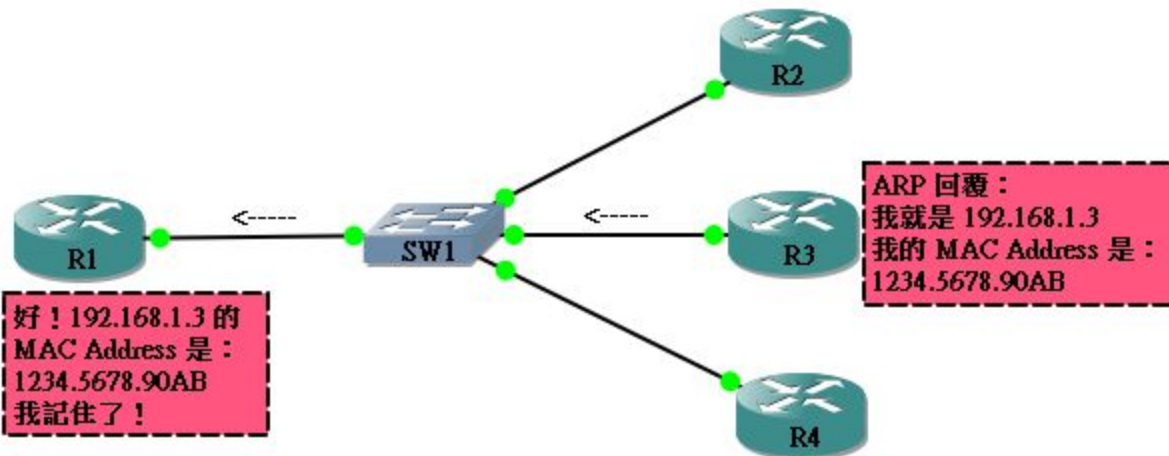
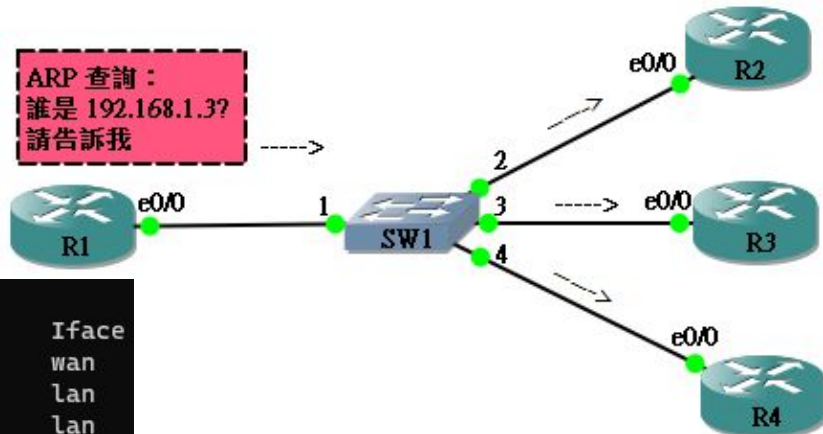
[HTTP request 1/1]

[Response in frame: 4]

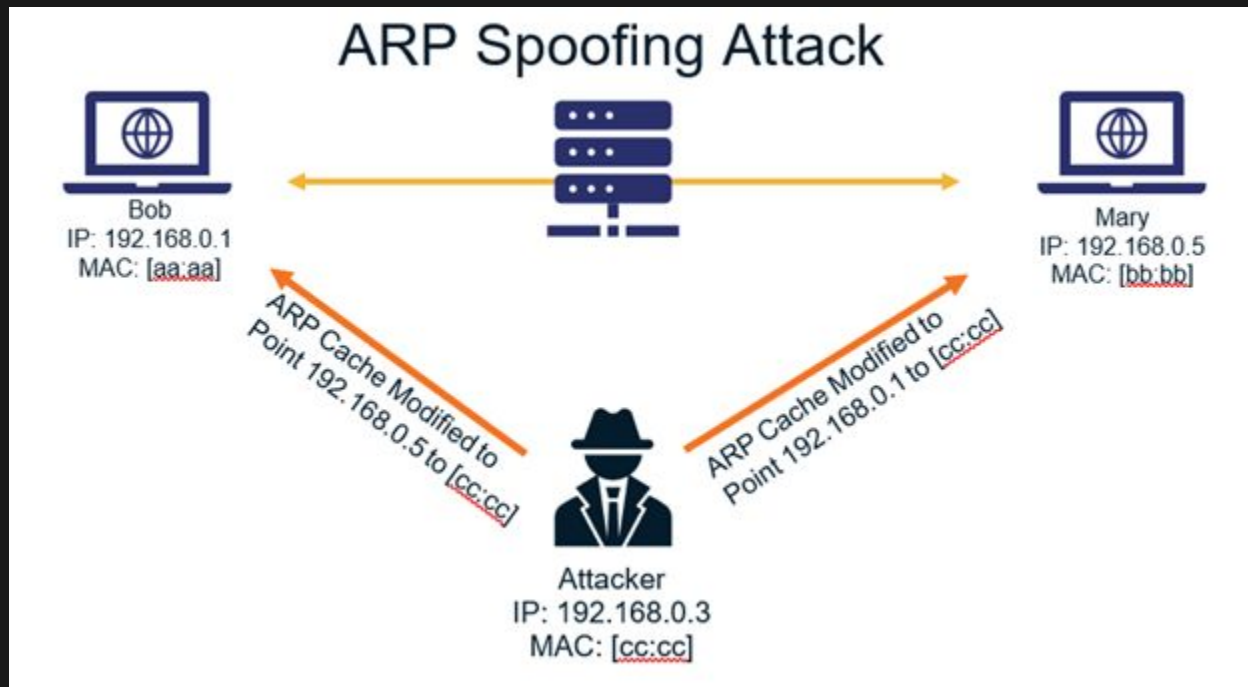
# 什麼是網路-ARP

```
root@node1-router:~# arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
router.asus.com	ether	f0:79:59:78:22:5c	C		wan
192.168.100.179	ether	0a:74:3a:55:c8:67	C		lan
192.168.100.103		(incomplete)			lan
192.168.100.177	ether	0a:74:3a:55:c8:67	C		lan
192.168.100.101	ether	8c:85:90:0f:1c:76	C		lan
192.168.100.107	ether	02:c1:94:91:f9:f1	C		lan
192.168.245.164	ether	74:a6:cd:93:e2:6b	C		wan



# 什麼是網路 -ARP Spoofing



# 什麼是網路 -Lab1

練習 ARP Spoofing

連線靶機 kali 密碼:nckuctf

```
ssh 7c87f3b2@140.116.246.59 -p 2253
```

or

```
ssh 7c87f3b2@140.116.246.59 -p 2254
```

or

```
ssh 7c87f3b2@140.116.246.59 -p 2255
```

# 什麼是網路 -Lab1

下指令：

```
sudo tcpdump -Ai eth1
```

你會看到甚麼都沒有

接著開第二個 ssh console (同一台機器) 下：

```
sudo arpspoof -i eth1 -t 192.168.111.1 192.168.111.2
```

然後再回去看第一個 console，你應該可以看到 Flag

# 什麼是網路 - 網段

使用 mac address 直接 packet 給目標是區域網路內的事。當今天要送到區域網路外的時候, 我們會送給 router 然後再由 router 幫我們轉送出去。



# 什麼是網路-網段

1	0.000000	7e:dd:d6:94:09:6d	Broadcast	ARP	42 Who has 192.168.100.113? Tell 192.168.100.100
2	0.000150	e6:82:a0:97:43:45	7e:dd:d6:94:09:6d	ARP	60 192.168.100.113 is at e6:82:a0:97:43:45
3	0.000480	192.168.100.100	192.168.100.113	HTTP	1... GET / HTTP/1.1
4	0.000710	192.168.100.113	192.168.100.100	HTTP	9... HTTP/1.1 200 OK (text/html)

> Frame 3: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface \Device\NPF\_{D50E8A4D-2461-42C7-8F...}

> Ethernet II, Src: 7e:dd:d6:94:09:6d (7e:dd:d6:94:09:6d), Dst: e6:82:a0:97:43:45 (e6:82:a0:97:43:45)

> Internet Protocol Version 4, Src: 192.168.100.100, Dst: 192.168.100.113

> Transmission Control Protocol, Src Port: 22475, Dst Port: 80, Seq: 1, Ack: 1, Len: 78

1	0.000000	7e:dd:d6:94:09:6d	Broadcast	ARP	42 Who has 192.168.100.254? Tell 192.168.100.100
2	0.000272	36:fa:a7:55:ff:3f	7e:dd:d6:94:09:6d	ARP	60 192.168.100.254 is at 36:fa:a7:55:ff:3f
3	0.542885	192.168.100.100	93.184.216.34	HTTP	1... GET / HTTP/1.1
4	0.682994	93.184.216.34	192.168.100.100	TCP	1... 80 → 25166 [ACK] Seq=1 Ack=75 Win=128 Len=1452
5	0.682994	93.184.216.34	192.168.100.100	HTTP	2... HTTP/1.1 200 OK (text/html)

> Frame 3: 128 bytes on wire (1024 bits), 128 bytes captured (1024 bits) on interface \Device\NPF\_{D50E8A4D-2461-42C7-8F...}

> Ethernet II, Src: 7e:dd:d6:94:09:6d (7e:dd:d6:94:09:6d), Dst: 36:fa:a7:55:ff:3f (36:fa:a7:55:ff:3f)

> Internet Protocol Version 4, Src: 192.168.100.100, Dst: 93.184.216.34

> Transmission Control Protocol, Src Port: 25166, Dst Port: 80, Seq: 1, Ack: 1, Len: 74



# 什麼是網路-網段

本地地址：台南市東區大學路1號

區域定義：同區就是同區域

目標地址：台南市東區東寧路 77 號

動作：騎機車直接送到目標

# 什麼是網路-網段

本地地址：台南市東區大學路1號

區域定義：同區就是同區域

目標地址：台南市北區公園南路206號

動作：騎機車送到成功大學郵局讓郵差幫我送信

# 什麼是網路-網段

來源 ip: 192.168.100.100

netmask: 255.255.255.0

目標 ip: 192.168.100.113

動作: arp 取得 192.168.100.113 的 mac, 然後送到 192.168.100.113 的 mac address 的電腦

# 什麼是網路-網段

來源 ip: 192.168.100.100

netmask: 255.255.255.0

目標 ip: 93.184.216.34

動作: arp 取得 192.168.100.254 (default gateway 或叫 預設出口) 的 mac, 然後送到 192.168.100.254 的 mac address 的電腦

# 什麼是網路-網段

```
IPv4 位址 . . . . . : 192.168.100.100
子網路遮罩 . . . . . : 255.255.255.0
預設閘道 . . . . . : fe80::34fa:a7ff:fe55:ff3f%19
                        192.168.100.254
```

```
chummy@ad143142:~$ ip a show eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 8a:c9:72:22:da:6d brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.110.100/24 metric 100 brd 192.168.110.255 scope global dynamic eth0
        valid_lft 408sec preferred_lft 408sec
    inet6 fe80::88c9:72ff:fe22:da6d/64 scope link
        valid_lft forever preferred_lft forever
```

```
chummy@ad143142:~$ ip route
```

```
default via 192.168.110.254 dev eth0 proto dhcp src 192.168.110.100 metric 100
192.168.5.123 via 192.168.110.254 dev eth0 proto dhcp src 192.168.110.100 metric 100
192.168.100.146 via 192.168.110.254 dev eth0 proto dhcp src 192.168.110.100 metric 100
192.168.110.0/24 dev eth0 proto kernel scope link src 192.168.110.100 metric 100
192.168.110.254 dev eth0 proto dhcp scope link src 192.168.110.100 metric 100
```

# 什麼是網路-網段

Address and Netmask = Network

```
root@node1-router:~# ipcalc 192.168.100.100/24
Address:    192.168.100.100    11000000.10101000.01100100. 01100100
Netmask:    255.255.255.0 = 24  11111111.11111111.11111111. 00000000
Wildcard:   0.0.0.255          00000000.00000000.00000000. 11111111
=>
Network:    192.168.100.0/24    11000000.10101000.01100100. 00000000
HostMin:    192.168.100.1       11000000.10101000.01100100. 00000001
HostMax:    192.168.100.254     11000000.10101000.01100100. 11111110
Broadcast:  192.168.100.255     11000000.10101000.01100100. 11111111
Hosts/Net:  254                  Class C, Private Internet
```

# 什麼是網路-網段

Address and Netmask = Network

```
root@node1-router:~# ipcalc 192.168.100.100/20
```

```
Address:    192.168.100.100    11000000.10101000.0110 0100.01100100
```

```
Netmask:    255.255.240.0 = 20  11111111.11111111.1111 0000.00000000
```

```
Wildcard:   0.0.15.255         00000000.00000000.0000 1111.11111111
```

```
=>
```

```
Network:    192.168.96.0/20     11000000.10101000.0110 0000.00000000
```

```
HostMin:    192.168.96.1       11000000.10101000.0110 0000.00000001
```

```
HostMax:    192.168.111.254    11000000.10101000.0110 1111.11111110
```

```
Broadcast:  192.168.111.255    11000000.10101000.0110 1111.11111111
```

```
Hosts/Net:  4094               Class C, Private Internet
```

# 什麼是網路 -路由

資料送到 router 時, router 會根據他上面的路由表去轉送 packet

```
node1-router (192.168.245.100) -> 1.1.1.1 (1.1.1.1)
Keys:  Help    Display mode    Restart statistics    Order of fields    quit

Host
1. 192.168.245.254
2. 172.16.1.1
3. 168.95.98.254
4. 168.95.54.30
5. (waiting for reply)
6. 220.128.26.10
7. 220.128.1.105
8. 203.75.230.125
9. 1.1.1.1
```



# 什麼是網路 -路由

```
jimmy (172.16.0.5) -> 192.168.100.100 (192.168.100.100)  
Keys:  Help    Display mode    Restart statistics    Order
```

Host

1. 172.16.3.7
2. 172.16.0.6
3. 192.168.100.100

```
root@jimmy:/etc/wireguard/VpnGenerator# ip route  
default dev wan scope link  
140.116.154.66 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
140.116.165.7 dev wan scope link  
140.116.246.59 dev wan scope link  
168.95.98.254 dev wan proto kernel scope link src 220.134.202.36  
172.16.0.0 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
172.16.0.2 nhid 177 via 172.16.3.7 dev toNCKUbone proto ospf metric 20 onlink  
172.16.0.3 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
172.16.0.6 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
172.16.3.7 dev toNCKUbone proto kernel scope link src 172.16.3.6  
172.16.3.9 dev tostuix proto kernel scope link src 172.16.3.8  
172.16.3.11 dev tovultr proto kernel scope link src 172.16.3.10  
192.168.1.0/24 dev pppbridge proto kernel scope link src 192.168.1.10  
192.168.10.0/24 dev LAN proto kernel scope link src 192.168.10.254  
192.168.20.0/24 via 192.168.20.2 dev vpnin443  
192.168.20.2 dev vpnin443 proto kernel scope link src 192.168.20.1  
192.168.21.0/24 via 192.168.21.2 dev vpnin444  
192.168.21.2 dev vpnin444 proto kernel scope link src 192.168.21.1  
192.168.30.0/24 dev wgserver proto kernel scope link src 192.168.30.254  
192.168.100.0/24 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
192.168.110.0/24 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
192.168.130.0/24 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
192.168.200.0/24 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
192.168.245.0/24 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink  
192.168.246.0/24 nhid 223 via 172.16.3.7 dev toNCKUbone proto ospf src 172.16.0.5 metric 20 onlink
```

# TCP vs UDP

**TCP**

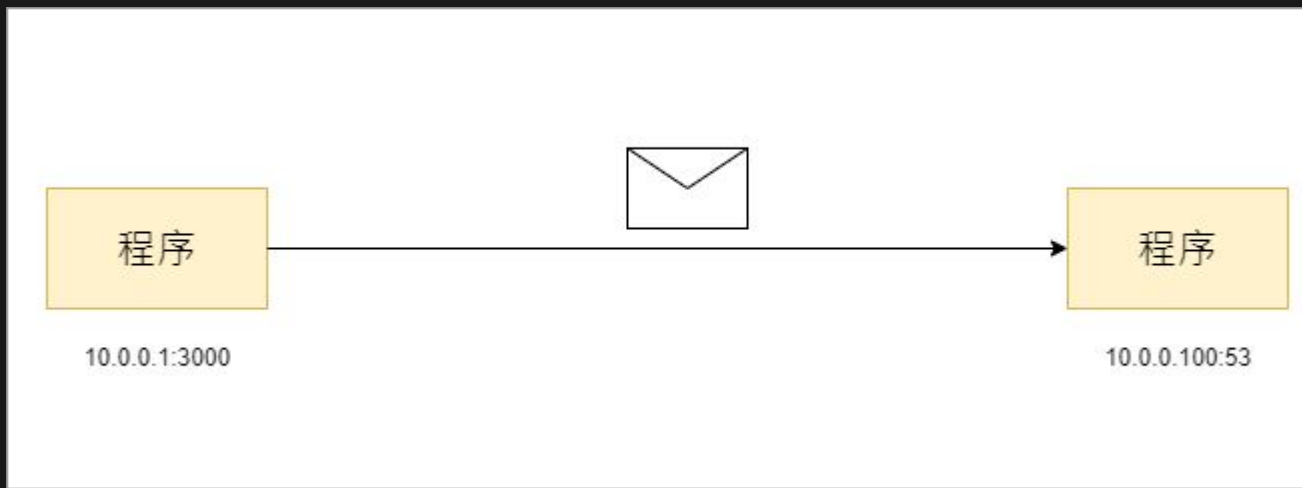


**UDP**



# TCP vs UDP-UDP

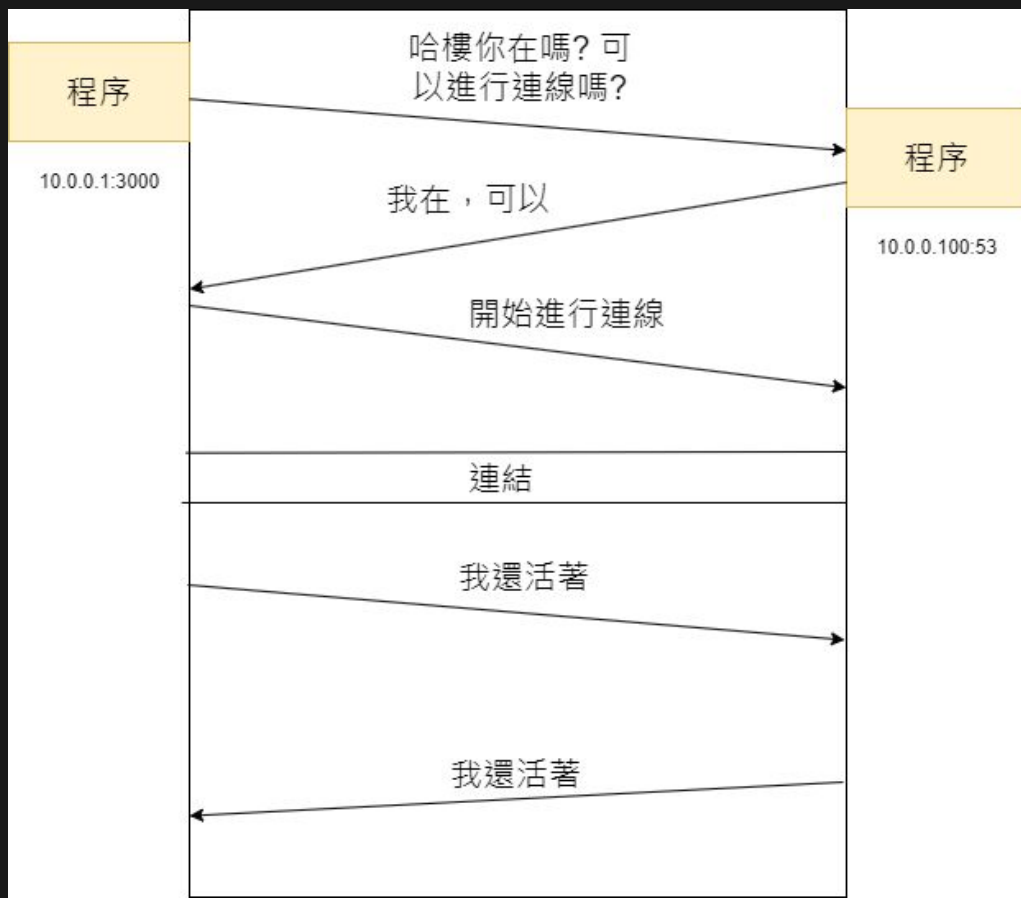
當資料以 UDP 傳輸時，傳送方就只會單純地把資料丟給接收方就不管了。不管接收方有沒有收到。



# TCP vs UDP-TCP

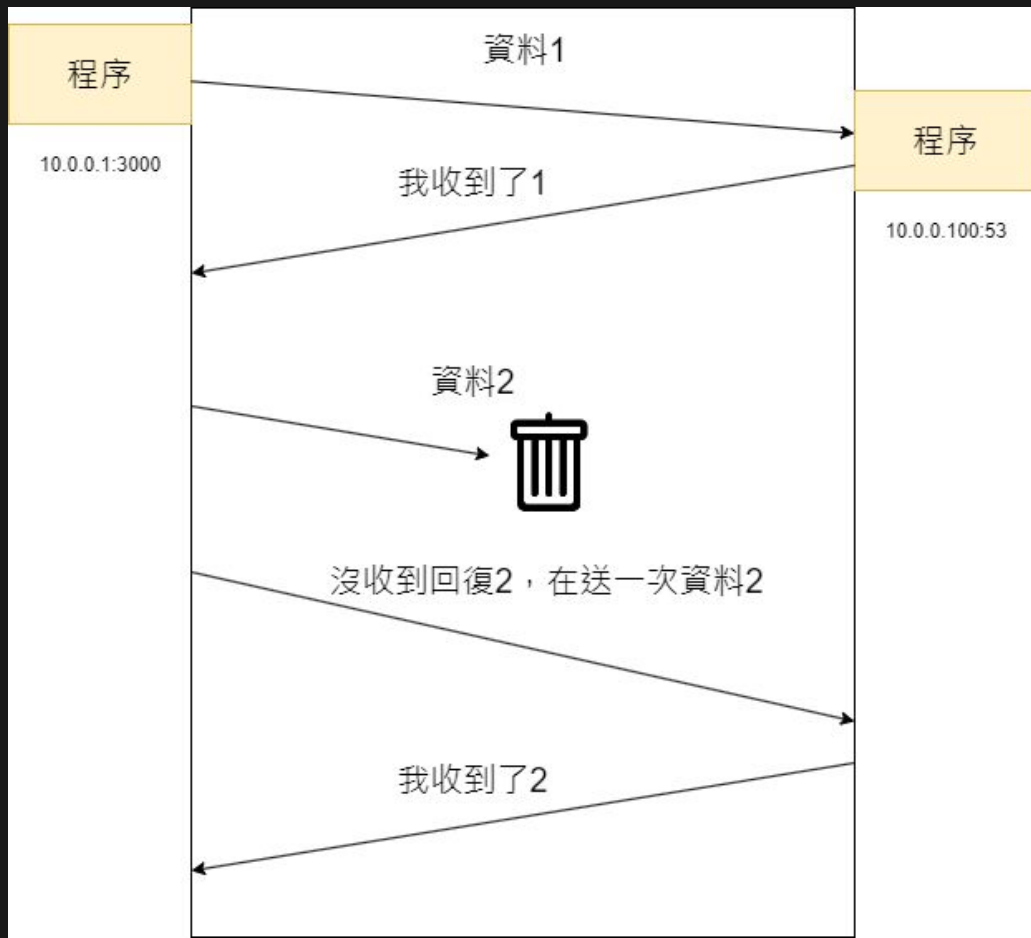
當資料以 TCP 傳輸時，傳送方與接收方必須要先三向交握以建立連線(確保兩邊都能正常傳送與接收訊息)。

連線中必須隨時發出 keepalive 訊息讓對方知道自己還在。



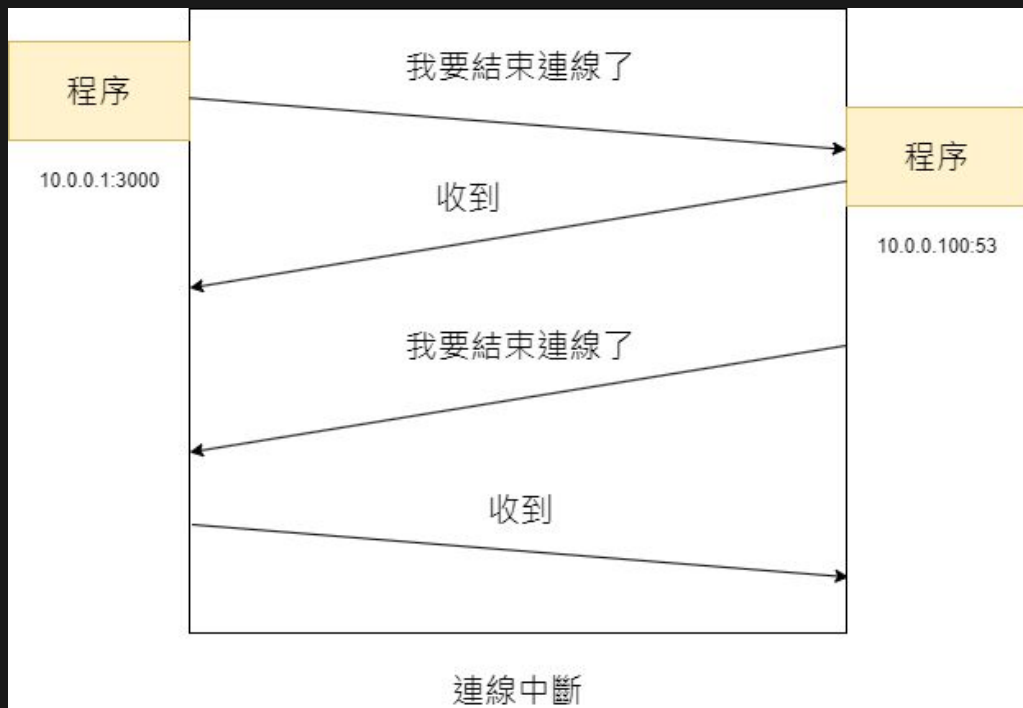
# TCP vs UDP-TCP

之後每次傳送方發送資料給接收方時，接收方都必須要回傳一個「訊息已收到(ACK)」回來，如果傳送方在一定時間內沒收到確認，就會進行重傳的動作，同時接收方在接收資料時，也會確保資料是按造正確的順序收到。

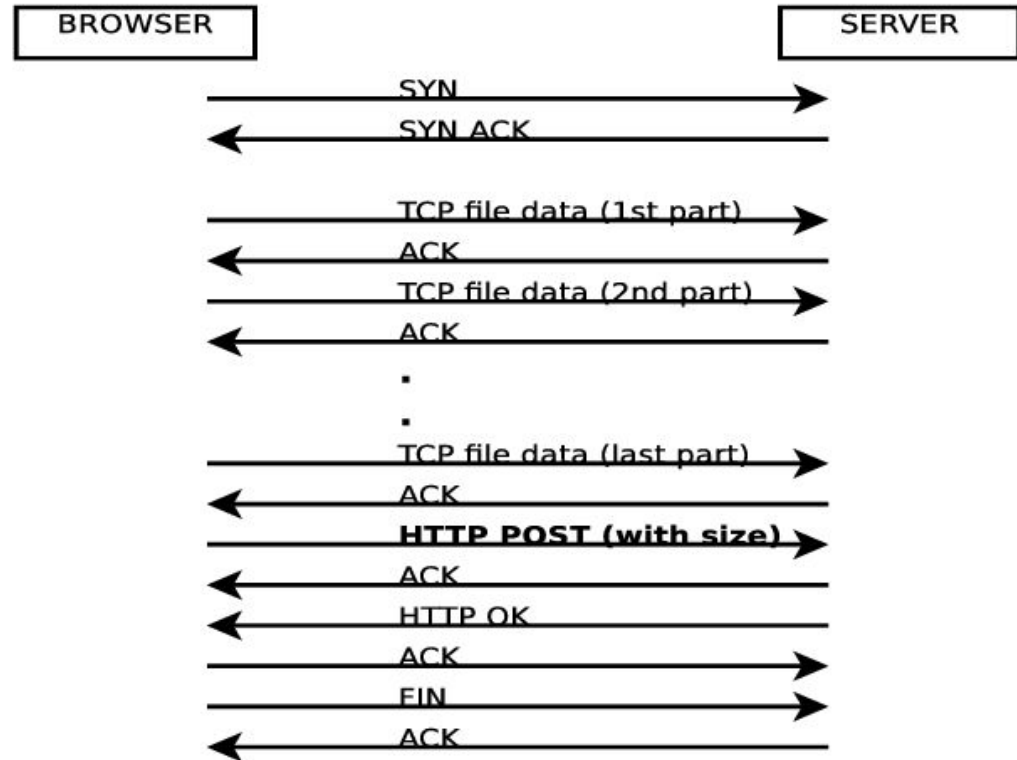


# TCP vs UDP-TCP

最後，如果超過一定時間沒收到  
keepalive，或者對方發送**結束連線**的  
要求時就會結束連線。



# TCP vs UDP-TCP



# NAT

現今 IPv4 網路有一個很嚴重的問題，那就是 IP 不夠用，



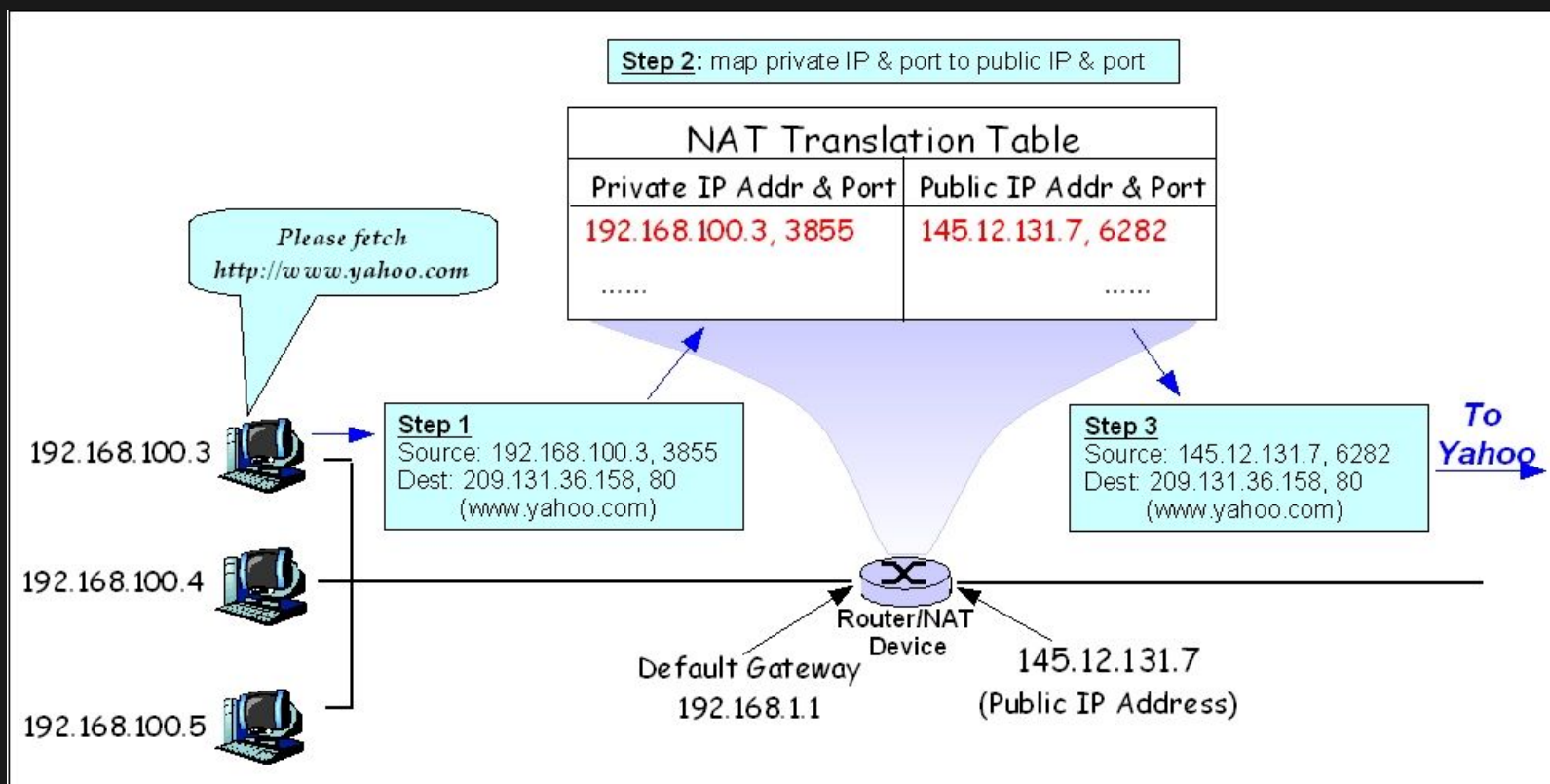
# NAT

- Private Address

- RFC1918
- 使用 private address 的 Packets 出去外部網路時會被改成外部網路IP

IP Class	From	To	CIDR notation
Class A	10.0.0.0	10.255.255.255	10.0.0.0/8
Class B	172.16.0.0	172.31.255.255	172.16.0.0/12
Class C	192.168.0.0	192.168.255.255	192.168.0.0/16

# NAT



# NAT-Lab1

下指令 `curl -L ifconfig.me`

```
C:\Users\jimmy>curl ifconfig.me  
59.127.61.82
```

# NAT

那如果有 server 架在內部網路。當外部網路上的主機要對內部網路上的主機發起主動連線時。會有入站 NAT 的問題

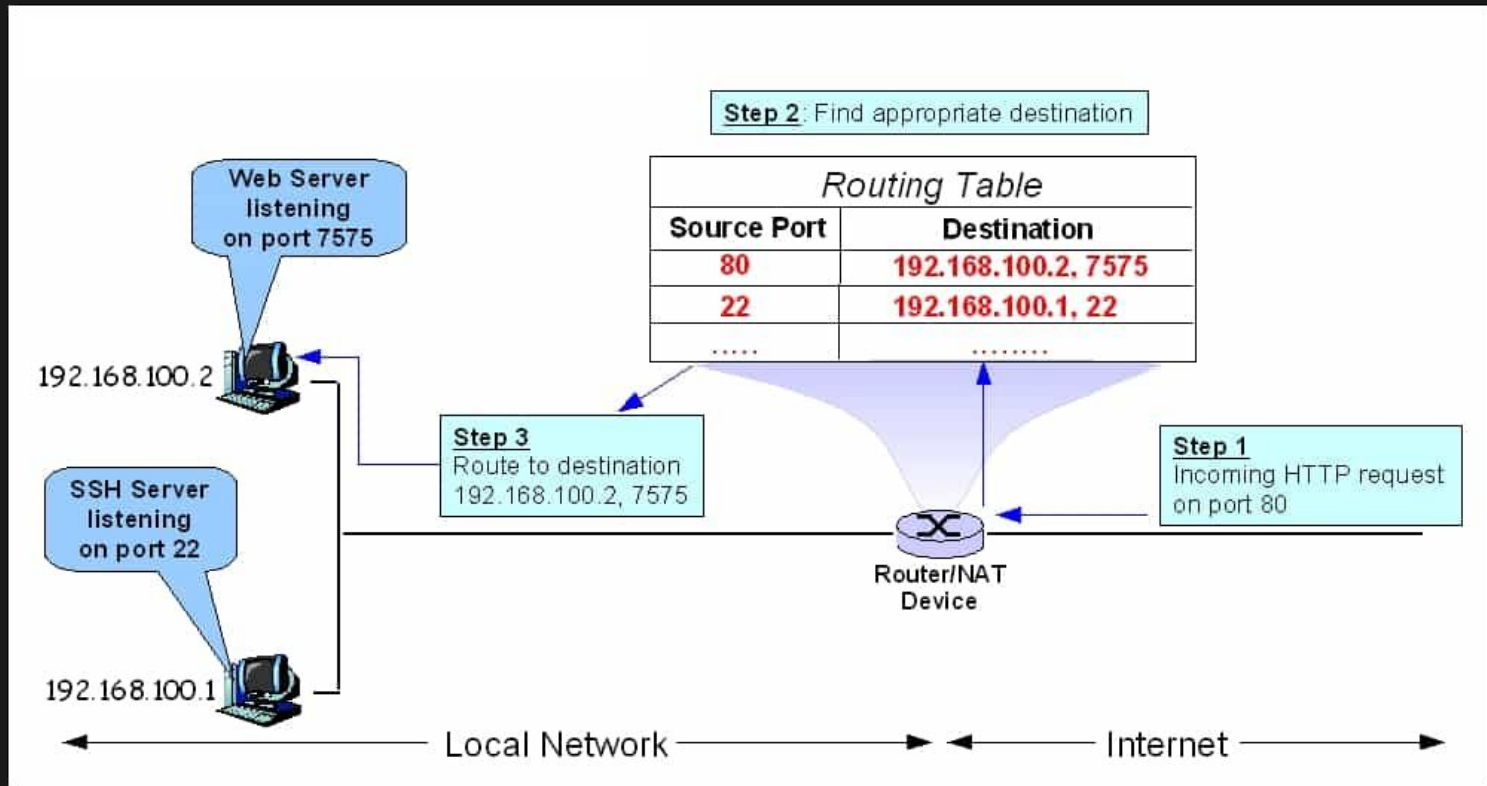
外部網路不知道內部網路的私有位址

隱藏地址不可路由

解決方案：

port forwarding

# NAT



# NAT

How does NAT work

# 什麼是協定

協定是指網路中的兩個程式間溝通的規範，可以想像成語言。

比如說瀏覽器對網頁伺服器之間的溝通方式，就必須要按照網頁的協定(HTTP)的規範去溝通。

對於使用者來說最常見的協定有：

1. 網頁服務: HTTP
2. 檔案分享服務: FTP or SMB or NFS
3. 玩遊戲(由遊戲開發商自行定義)

# HTTP

HTTP/2 之前的版本是一種建立在 TCP 上的傳輸協定。(HTTP/3 例外, 今天不提)  
為何需要使用 TCP 是因為網頁他會需要可靠性的傳輸。



# HTTP

```
root@pve-node01:~# nc example.com 80
GET / HTTP/1.1
Host: example.com
User-Agent: curl/7.88.1
Accept: */*
```

```
1 GET / HTTP/1.1
2 Host: example.com
3 User-Agent: curl/7.88.1
4 Accept: */*
5
6
```

這是 HTTP request

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 114204
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Sat, 24 Jun 2023 19:42:13 GMT
Etag: "3147526947"
Expires: Sat, 01 Jul 2023 19:42:13 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (oxr/830C)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256
```

```
<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
```

這是 HTTP response

# HTTP-Request

	指令 Method	路徑 Path	協定版本 Version	
標頭資訊 Header	GET	/index.html	HTTP/1.1	\r\n 換行
	Host: example.com			\r\n
	User-Agent: curl/7.88.1			\r\n
	Accept: */*			\r\n
				\r\n

# HTTP-Response

詳細 status: <https://http.cat/>

	協定版本	狀態
	Version	Status
	HTTP/1.1	200 OK
標頭資訊 Header	Accept-Ranges: bytes\r\nAge: 114204\r\nCache-Control: max-age=604800\r\nContent-Type: text/html; charset=UTF-8\r\nDate: Sat, 24 Jun 2023 19:42:13 GMT\r\nEtag: "3147526947"\r\nExpires: Sat, 01 Jul 2023 19:42:13 GMT\r\nLast-Modified: Thu, 17 Oct 2019 07:18:26 GMT\r\nServer: ECS (oxr/830C)\r\nVary: Accept-Encoding\r\nX-Cache: HIT\r\nContent-Length: 1256\r\n\r\n	
內容 Content	<!doctype html>\r\n<html>\r\n<head>\r\n<title>Example Domain</title>\r\n....	

# HTTP-Lab1

請大家用 linux, 開啟 command line, 然後下這個指令。

```
> nc example.com 80
```

## 接著輸入這些東西

GET / HTTP/1.1

Host: example.com

```
User-Agent: nc
```

Accept: \*/\*

```
1 GET / HTTP/1.1 CR LF
2 Host: example.com CR LF
3 User-Agent: nc CR LF
4 Accept: */* CR LF
5 CR LF
6
```

# HTTP-Lab1

接著打開瀏覽器按 F12 到 network,

然後網址輸入 <http://example.com>

點開第一個資料看看

他的 Header 跟  
Response

The screenshot shows a web browser window with the address bar displaying `example.com`. The page content shows "Example Domain" with a brief description and a link to "More information...".

On the right, the Chrome DevTools Network panel is open, showing a list of requests. The first request, `example.com`, is selected. The right-hand pane displays the details of this request:

- General:**
  - Request URL: `https://example.com/`
  - Request Method: `GET`
  - Status Code: `304`
  - Remote Address: `[2606:2800:220:1:248:1893:25c2:1946]:443`
  - Referrer Policy: `strict-origin-when-cross-origin`
- Response Headers (10):**
- Request Headers:**
  - Authority: `example.com`
  - Method: `GET`
  - Path: `/`
  - Scheme: `https`
  - Accept: `text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7`
  - Accept-Encoding: `gzip, deflate, br`
  - Accept-Language: `zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7`
  - Cache-Control: `max-age=0`
  - If-Modified-Since: `Thu, 17 Oct 2019 07:18:26 GMT`
  - If-None-Match: `"3147526947"`
  - Sec-Ch-Ua: `"NotA/Brand";v="8", "Chromium";v="114", "Google Chrome";v="114"`

At the bottom of the panel, it shows "4 requests" and "4.7 kB transferred".

# HTTP-Lab2

請大家用 linux, 開啟 command line, 然後下這個指令。

```
> nc -lp 80
```

開啟同一台 linux 上的瀏覽器瀏覽 <http://127.0.0.1>

接著馬上到剛剛下指令的 command line 輸入下面的資料。

```
HTTP/1.1 200 OK
```

```
Content-Type: text/html
```

```
Content-Length: 11
```

```
Hello World
```

# HTTP-method

<https://developer.mozilla.org/zh-TW/docs/Web/HTTP/Methods>

# HTTP-curl

用 nc 構建 http request 太慢了，所以我們可以用 curl

```
root@node1-router:~# curl http://192.168.100.113
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```



# DNS

全名叫做 Domain Name System, 是一種建立在 UDP 的協定。

主要用來查詢 Domain Name 的 IP Address 或是查詢 IP Address 對應的 Domain Name。

那什麼是 Domain Name 呢？

# DNS

facebook.com

google.com.tw

hackmd.io/

github.com

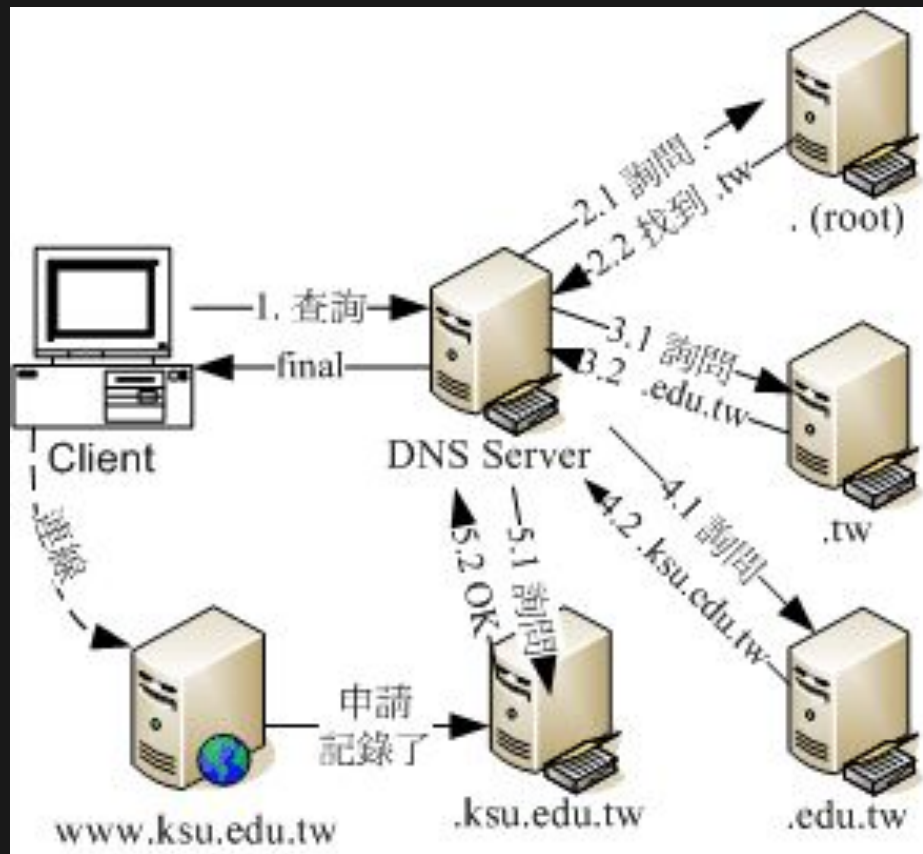
example.com

# DNS

頂級域  
主機名稱 三級域 Top Level Domain  
www.csie.ncku.edu.tw.  
四級域 二級域

# DNS




DNS 用的是一種樹狀的查詢架構，這樣的優點是上層 Domain 的管理者可以自由地將下層 Domain 的管理權發給別人。




ref: [link](#)


# DNS


網路上有很多的域名註冊商，通常你可以用大約 NTD 300 /年，的價格購買到你喜歡的二級域名，網路上也有一些域名託管服務，讓你可以不用架設實體的 DNS Server。

+ 新增網站  支援 ▼ 中文 (台灣) ▼ 

Jimmy012403976@...

 網站


 Discover


 網域註冊


管理網域

移轉網域


註冊網域


 分析與記錄 ▼


 資訊安全中心 ▼


 Turnstile


測試版

 Zero Trust

 Area 1

 Workers 和 Pages ▼

 R2 ▼

 Stream

◀ 摺疊側邊欄

搜尋網域名稱

搜尋

按一下[這裡](#)以查看支援的延伸項目清單。

letmeseesee.com 已註冊，無法使用。如果您已擁有此網域，並且希望將其移轉至 Cloudflare，請至儀表板中的[移轉網域](#)區段。

以下是一些可能可用的類似網域。

網域	價格	
letmeseesee.io	US\$36.00	<a href="#">購買</a>
letmehearsee.com	US\$9.15	<a href="#">購買</a>
letmehearsee.net	US\$10.10	<a href="#">購買</a>
letmeseesee.me	US\$12.85	<a href="#">購買</a>
letmeseesee.org	US\$10.11	<a href="#">購買</a>
letmeseesee.blog	US\$20.18	<a href="#">購買</a>
letmeseesee.site	US\$20.18	<a href="#">購買</a>

# DNS-Lab1

請大家用 linux, 開啟 command line, 然後下這個指令。

```
> dig +trace www.google.com
```