實驗日期:2020/4/16

實驗名稱:

1. ITS Exp. 5: 區域網路IP繞送
2. ITS Exp. 6: 網際網路IP繞送
3. ITS Exp. 7: IP繞送之TTL
4. ITS Exp. 8: IP繞送路徑之追蹤

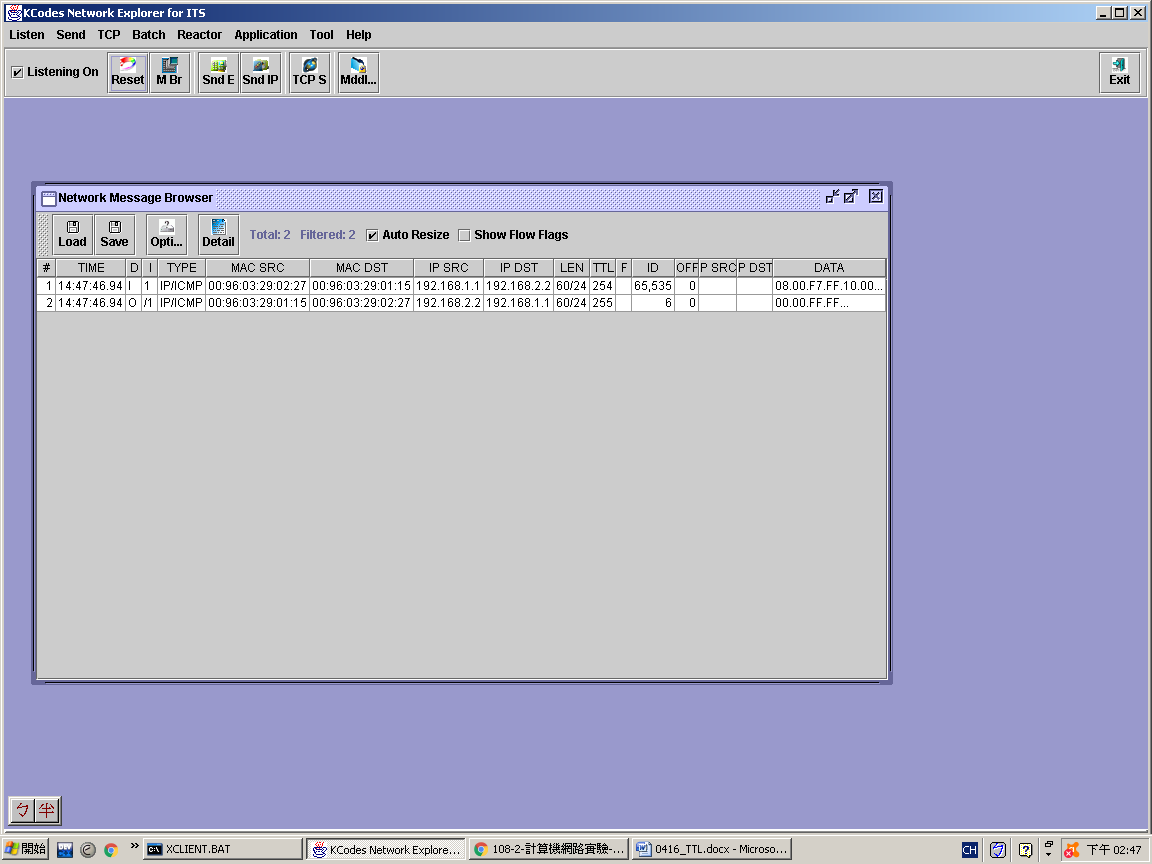
問題與答案:

**Exp5:區域網路IP繞送**

* **試著將ITS 1 的interface 1 的子網路遮罩數值改為”255.255.255.255”或”255.255.0.0”。這樣一來，ITS 1 是不是還可以成功的在這個網路拓樸中傳遞IP封包？**

**255.255.255.255:**

將subnet mask 改為255.255.255.255後，ITS 1還是可以在這個網路拓樸中正常運作傳遞封包。



**255.255.0.0:**

將subnet mask 改為255.255.255.255時，無法傳遞

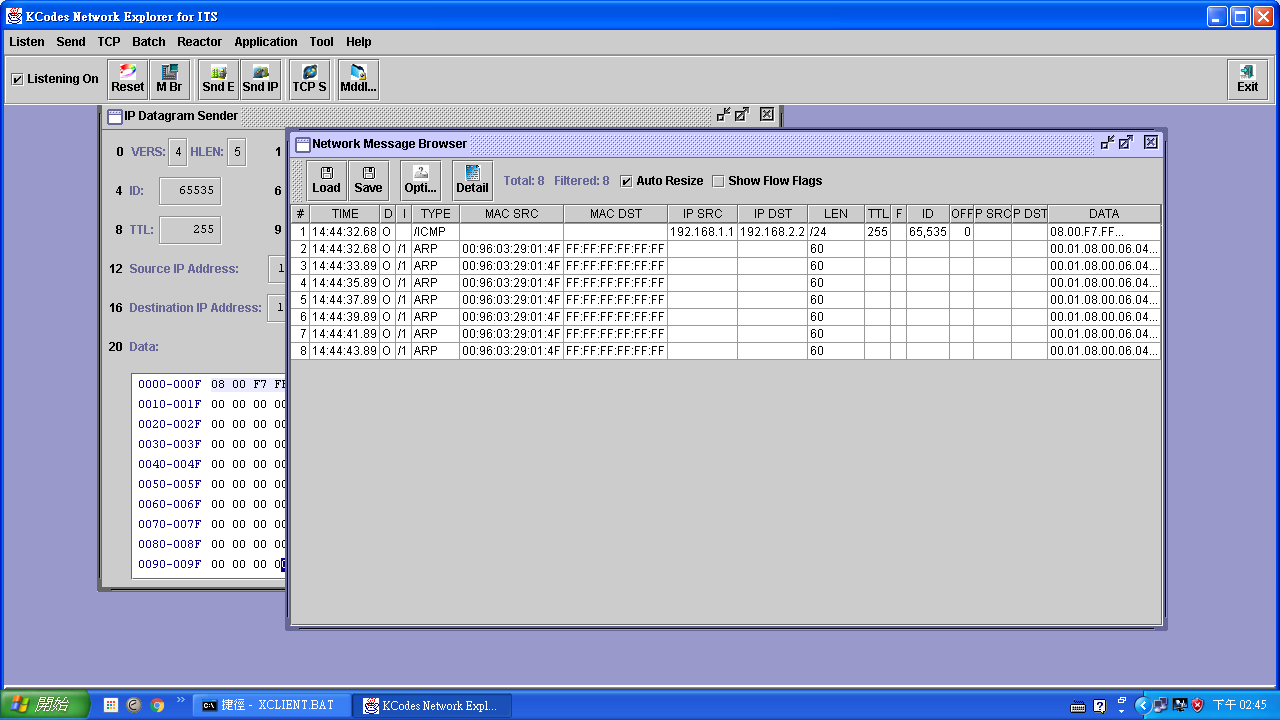
在切割子網後Router就無法經由Class判別Network ID及Host ID有多少位元。因此產生Subnet Mask機制，可以決定Network ID的bit數。

Class B Subnet Mask為11111111 11111111 00000000 00000000(255.255.0.0)

切割子網後Subnet Mask為11111111 11111111 11100000 00000000(255.255.224.0)

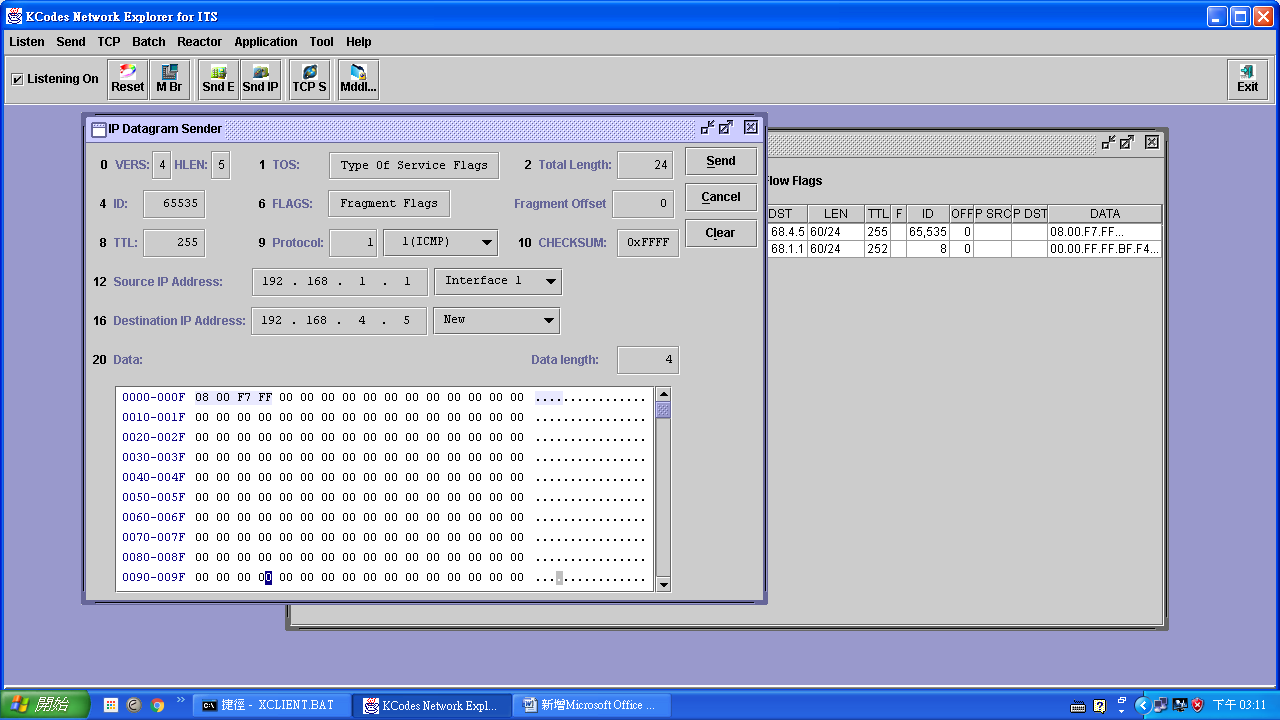
Subnet Mask的1對映Network ID，0對映Host ID

Subnet Mask及IP產生Network ID。Router就可以根據此Network ID決定要轉送至何處。



(當我們在做255.255.0.0時，因為沒有再清空routing table，所以實驗結果一直為錯的，然後我們就把第一台 設default gateway 0.0.0.0.0.0.0.0.192.168.1.2 結果才為正確的)

**Exp6:網際網路IP繞送**



* **在繞送規則中，Destination欄位與Mask欄位如果都設為”0.0.0.0”的話，代表著甚麼意義?**

0.0.0.0為預設路徑，只要路由表找不到傳送路徑的封包，最後都會交由預設路徑傳送，任何IP目的地都可以是預此路徑。

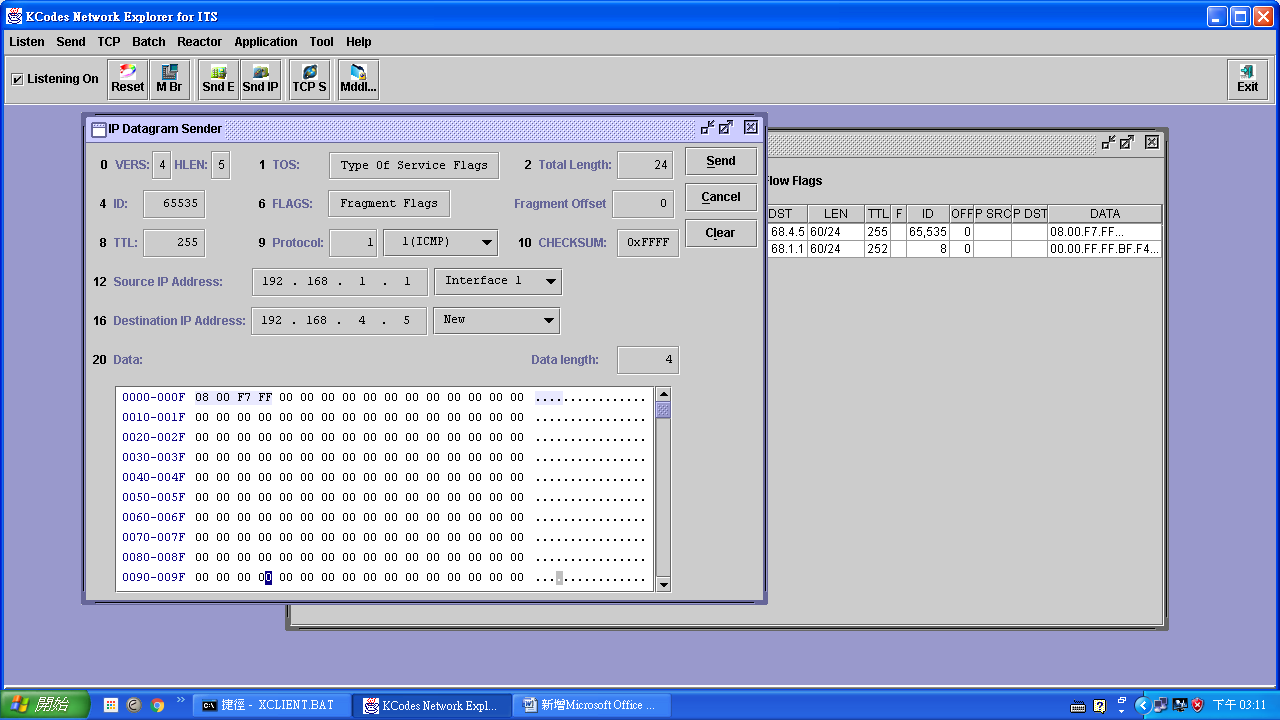
* **如果一個ISP(Internet Services Provider)業者要提供服務給數十萬的網路使用者，靜態繞送(static routing)的方式是否可以滿足需求？**

無法滿足需求，靜態繞送為路由器之間並不會交換訊息來探討網路狀態而隨時變更路由表。靜態路由是在路由器中設置的固定的路由表。除非網路管理員干預，否則靜態路由不會發生變化。由於靜態路由不能對網路的改變作出反映，一般用於網路規模不大、拓撲結構固定的網路中。它的優點是簡單、高效率及可靠。在所有的路由中，靜態路由優先級最高。當動態路由與靜態路由發生衝突時，以靜態路由為準。雖然靜態路由的優點很多，但它還是有缺陷。主要由於網路是一個很容易中斷的一個環境，時常會因為一個裝置或是一條線中斷而停止，這時如果剛好壞在那個路由節點的話，會因為路由器不再尋找新的路由器節點，所以使用舊的而導致完全中止。維護靜態路由是一件非常大的負擔，只要網路上有任何的改變，無論是新增或減少任何一部路由器，網路管理者都要隨時的去更改路由表的設定。

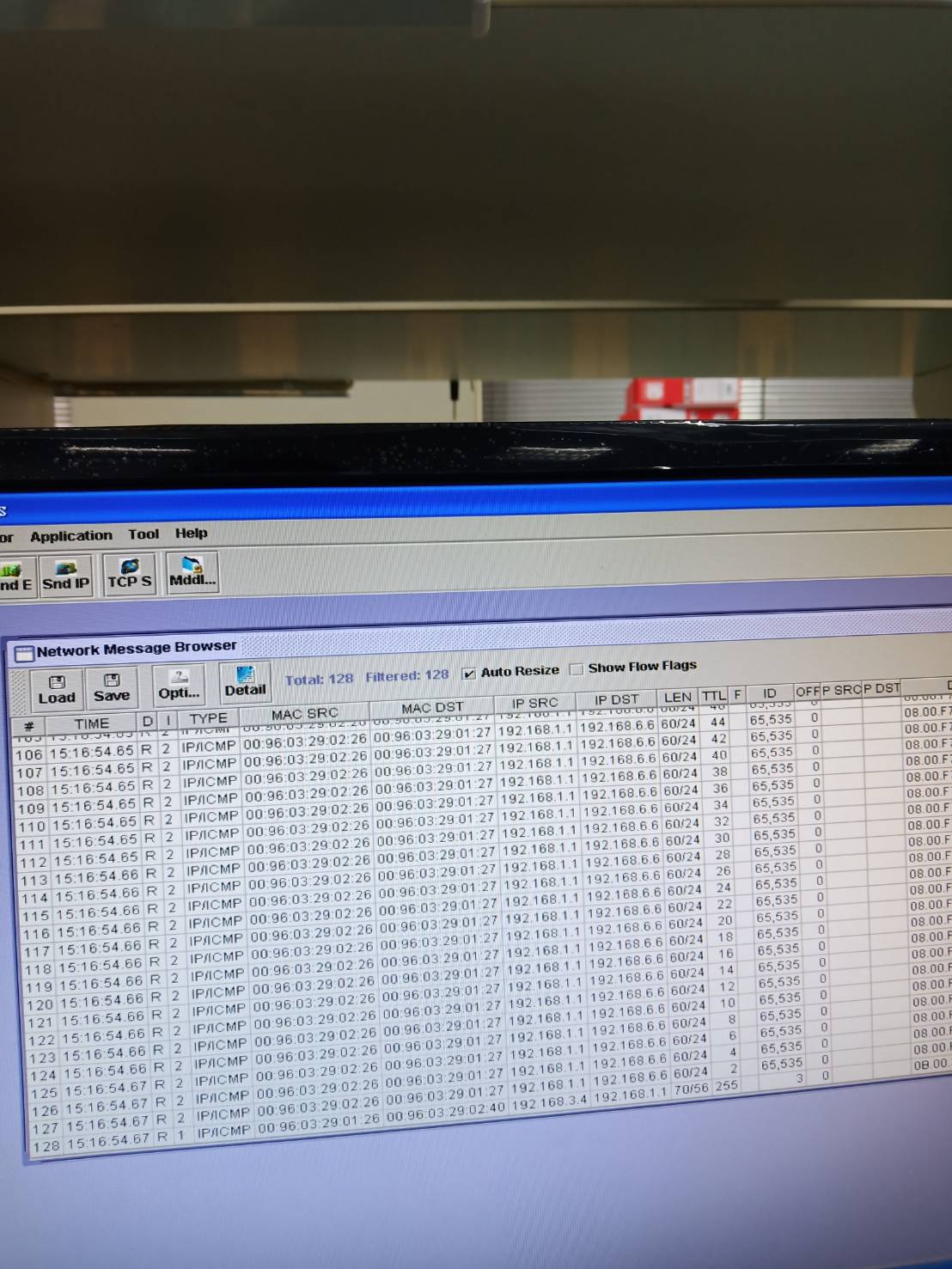
* **是否有其他的不同類型的繞送方式？**

有，繞送協定的目的是為了要自動的維護路由器的路徑表，為了達到這個目的，每台路由器都必需將自已的路徑資訊分享出來給彼此，也就是交換路由器特有的封包，這稱為路徑更新（routing update）。每一種繞送的協定都有不同的路徑更新方式，所以在內容、大小、頻率上都有差異。假設有一條線路故障時，和這條線路直接相連的路由器會自動偵測到目前有一條線路沒有回應，當偵測到沒有回應時，路由器會修改路徑表，並移除所有之前透過這條線路轉送的路徑記錄，然後散播路徑更新紀錄給所有仍然相連的網路區段，通知它們這條線路已經不通了，而那些被通知的路由器會根據收到的資訊來修改它們的路徑表，然後再轉送到它們附近的路由器，並一直轉送下去。

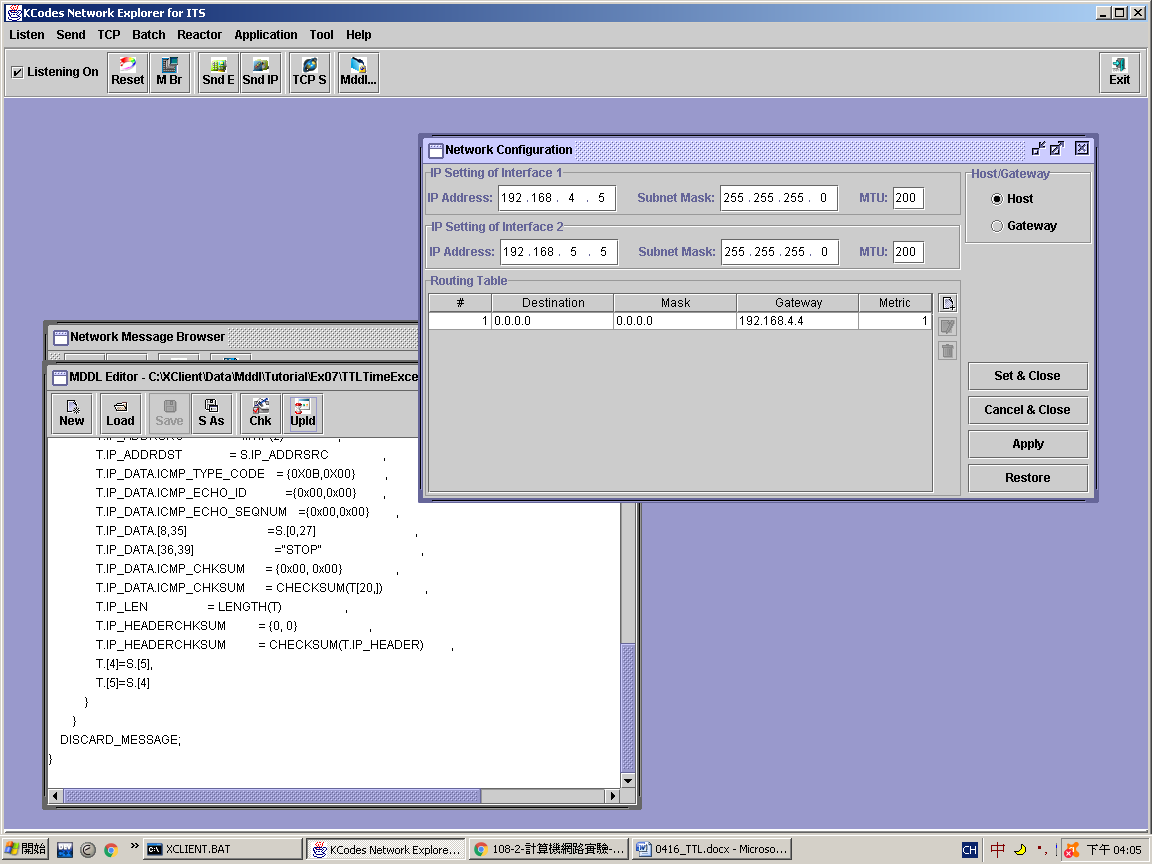
**Exp7:IP繞送之TTL**



**host(ITS 1) 所設定的圖**



**上圖為不斷出現迴圈的樣子**



**設置完，再LOAD 程式**

* **擷取執行MDDL時的畫面，說明結果與實驗步驟7、8有何不同，並解釋原因**

兩者差異在於checksum的部分，因為MDDL封包會在TTL=0時刪除，所以讓checksum=type的1補數值。

* **在此實驗案例裡，ITS的網路繞送表是否有錯，又或者有缺陷？**

沒有錯，缺陷在當發送不到的位址，便會不停的出現迴圈。

* **如何解決步驟7與8中，封包會陷入無窮迴圈的問題？**

可以加入TTL，當TTL=0時，則會刪除封包，得以結束無窮的迴圈。

結果討論:

這次的實驗還蠻需要每個組員的配合，各自知道自己在做甚麼，這樣才能快而有效率的完成。了解每個角色HOST GATEWAY，所扮演的功能。

參考資料:

<http://pollucite.blog126.fc2.com/blog-entry-604.html>