Chapter 09

ARP 與 ICMP

本章提要

- 9-1 ARP 簡介
- 9-2 ARP 封包格式
- 9-3 ARP 工具程式
- 9-4 ICMP 簡介
- 9-5 各類型的 ICMP 封包
- 9-6 ICMP 工具程式
- 9-7 擷取 ICMP 封包

ARP 與 ICMP

- 在 TCP / IP 協定組合中, 屬於網路層的協定有 IP、ARP 與 ICMP 等 3 種。其中最主要的當然是 IP, 至於 ARP 與 ICMP 一般皆視為輔助 IP 的協定。
- 本章將依序介紹 ARP 與 ICMP 這 2 個協定, 以及相關的應用。

9-1 ARP 簡介

- 鏈結層與網路層位址的特性：
  - 鏈結層在傳遞封包時, 必須利用鏈結層位址（例如：乙太網路 MAC 位址）來識別目的裝置。
  - 網路層在傳遞封包時, 必須利用網路層位址（例如：IP 位址）來識別目的裝置。
- 以 IP 為例, 便是將取得 MAC 位址的工作交由 ARP （Address Resolution Protocol, 位址解析協定）來執行。
- 若以 OSI 模型來說明 ARP 的功能, 便是利用網路層位址, 來取得對應的鏈結層位址。

9-1-1 ARP 運作方式

- ARP 運作的方式相當簡單，整個過程是由兩種封包所組成：
  - ARP 要求（ARP Request）
  - ARP 答覆（ARP Reply）

ARP 要求

- A 電腦廣播 ARP 要求封包給區域網路上所有的電腦。

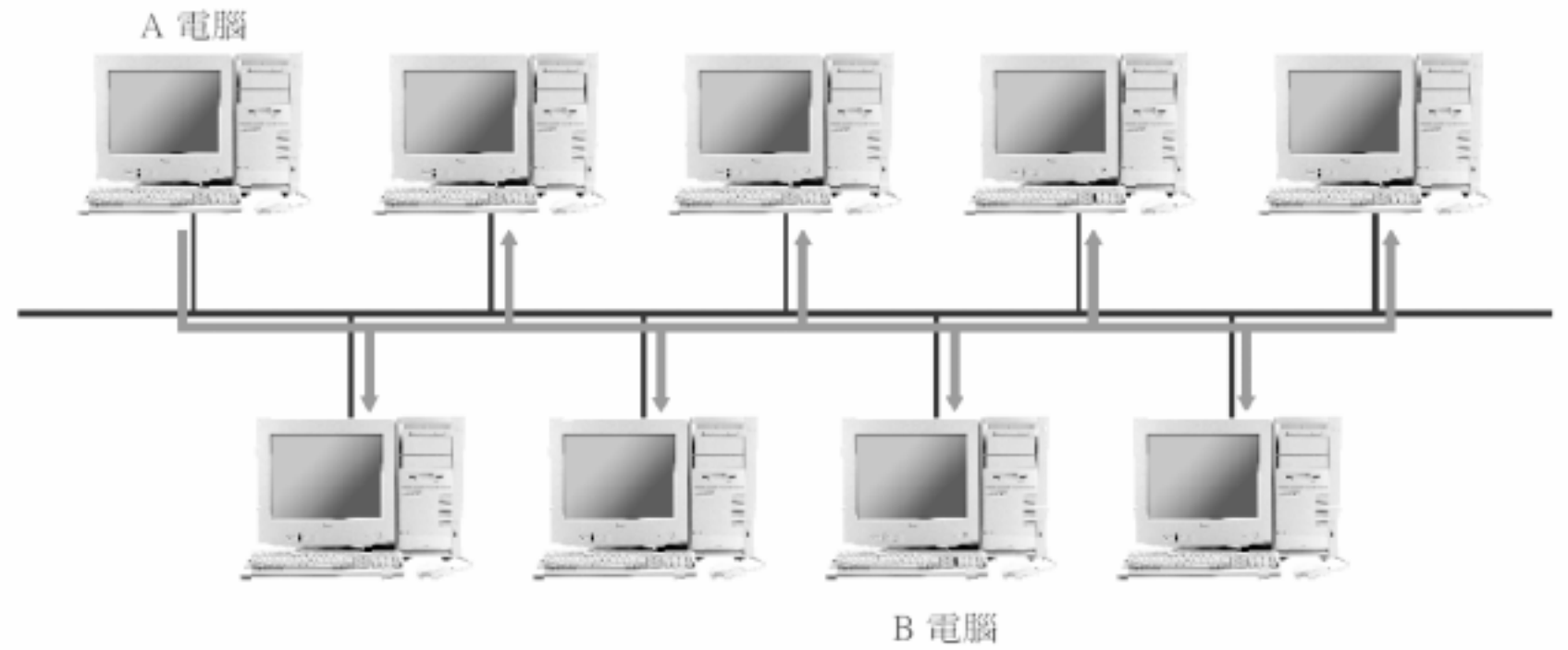


圖 9-01　ARP 要求封包會以廣播方式, 送至區域網路上的所有電腦

ARP 答覆

- 區域網路內的所有電腦都會收到 ARP 要求封包, 並與本身的 IP 位址比對, 判斷自己是否為要求解析的對象。
- 由於 B 電腦可從 ARP 要求封包中得知 A 電腦的 IP 位址與 MAC 位址, 因此 ARP 答覆封包不必再使用廣播的方式, 而是直接在乙太網路封包中, 指定 A 電腦的 MAC 位址為目的位址。
- ARP 答覆封包中最重要的內容當然就是 B 電腦的 MAC 位址。A 電腦收到此 ARP 答覆封包後, 即完成 MAC 位址解析的工作。

ARP 的解析範圍

- 路由器會阻擋乙太網路廣播封包，使該封包無法跨越到其它網路。因此 ARP 僅能解析同一網路內的 MAC 位址，無法解析其他網路的 MAC 位址。

9-1-2 ARP 快取

- 將常用的資料暫存在讀寫效率較佳的儲存區域，以加速存取的過程。
- ARP 快取所包含的紀錄，依產生的方式，可分為兩種紀錄。
  - 動態紀錄
  - 靜態紀錄
- 無論是動態或靜態紀錄，只要重新開機，全部都會消失

9-2 ARP 封包格式

- ARP 封包主要是記錄 IP 與 MAC 位址的相關資訊, 其包含的欄位如下：

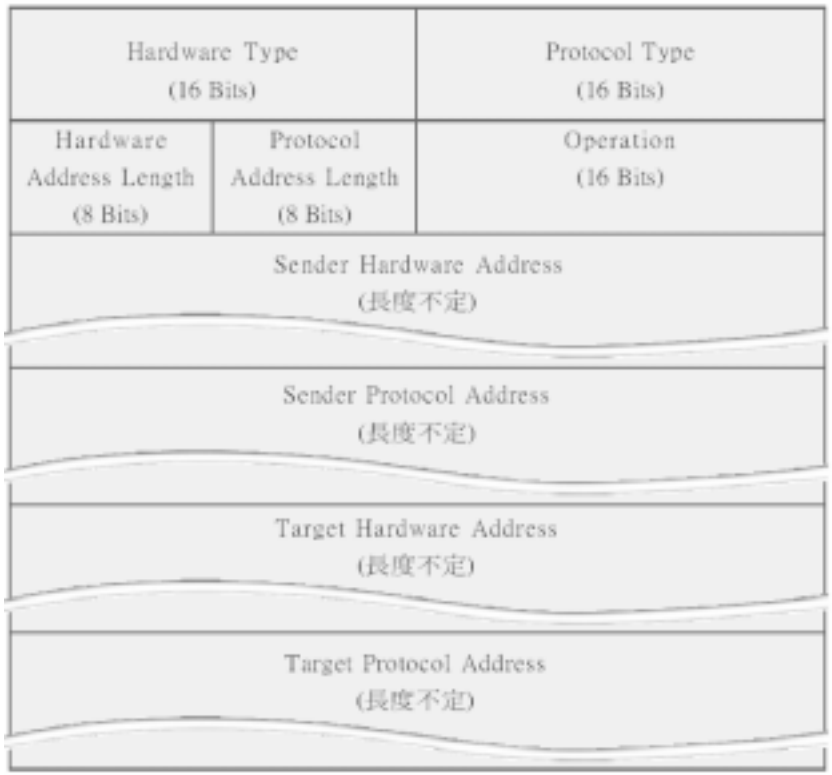


圖 9-02　ARP 封包結構

Hardware Type

- 長度為 2 Bytes, 指定硬體類型, 亦即資料連結層所用的技術。以乙太網路為例, Hardware Type 欄位值應為 1。下表列出一些常見的 Hardware Type 欄位值：

表 9-01　Hardware Type 欄位值代表的意義

Hardware Type 欄位值	技術
1	乙太網路
6	Token Ring
15	Frame Relay
16	ATM

Protocol Type

- 長度為 2 Bytes, 指定網路層所使用的協定, 一般即是 IP, 其欄位值為 2048, 若以 16 進位表示則為 0 x 0800。

Hardware Address Length

- 長度為 1 Byte, 指定 MAC 位址的長度。以乙太網路為例, 其 MAC 位址長度為 6 Bytes, 因此 Hardware Address Length 欄位值為 6。
- 此欄位值也決定了後續 Sender Hardware Address 與 Target Hardware Address 的長度。

Protocol Address Length

- 長度為 1 Byte, 網路層協定所用的位址長度。
- 若 Protocol Type 為 2048（IP），由於 IP 位址長度為 4 Bytes, 因此 Protocol Address Length 欄位值為 4。
- 此欄位值也決定了後續 Sender Protocol Address 與 Target Protocol Address 的長度。

Operation

- 長度為 2 Bytes, 指定 ARP 封包的類型, 最常見的即是 request 與 reply 兩種類型：

表 9-02　Operation 欄位值所指的封包類型

Operation 欄位值	封包類型
1	ARP request
2	ARP reply

Sender Hardware Address

- ARP 封包來源端的 MAC 位址, 以乙太網路為例, 即為 6 Bytes 的 MAC 位址。

Sender Protocol Address

- ARP 封包來源端的所用協定的位址, 以 IP 為例, 即為 4 Bytes 的 IP 位址。

Target Hardware Address

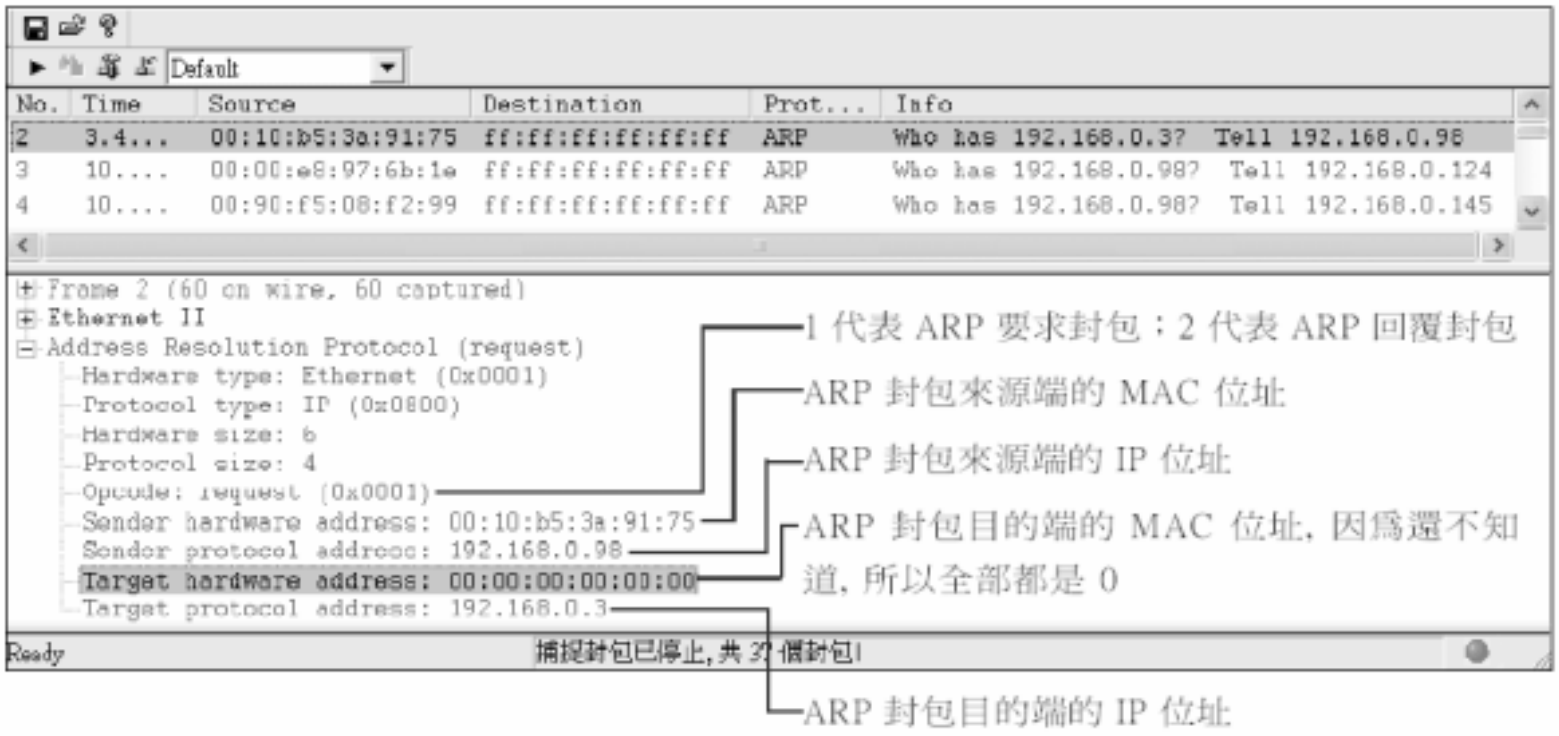
- ARP 封包目的端的 MAC 位址, 以乙太網路為例, 即為 6 Bytes 的 MAC 位址。若是 ARP request 封包, 因為還不知道目的端的 MAC 位址, 因此 Target Hardware Address 欄位址為 000000000000。

Target Protocol Address

- ARP 封包目的端所用協定的位址, 以 IP 為例, 即為 4 Bytes 的 IP 位址。

ARP 封包的長度

- 因為不同的網路層與鏈結層，所使用的位址長度都不同。
- 擷取一個 ARP 要求封包，用來驗證上述的說明：



9-3 ARP 工具程式

- 大部份作業系統都會提供 ARP 工具程式。以下將介紹 2 種 ARP 工具程式：
  - Windows 98 / XP 的 ARP.EXE
  - Linux 的 ARPWATCH。

9-3-1 ARP

- ARP.EXE 主要提供 3 項功能, 說明如下：
  - 檢視 ARP 快取中的紀錄
  - 刪除紀錄
  - 新增紀錄

檢視 ARP 快取中的紀錄

- 您可以利用 ARP.EXE 檢視 ARP 快取中目前的紀錄。
- 語法如下： `arp -a`

```
C:\>arp -a

Interface: 203.74.205.111 --- 0X10003
Internet Address      Physical Address      Type
203.74.205.1          00-10-7b-c1-ec-98     dynamic
203.74.205.3          00-10-b5-3a-91-75     dynamic
203.74.205.7          00-10-b5-3a-91-b8     dynamic
203.74.205.11         00-10-b5-3a-91-dc     dynamic
```

所解析的 IP 位址　　解析所得的 MAC 位址　　此紀錄產生的方式

```
C:\>
```

刪除紀錄

- 刪除 ARP 快取紀錄的語法如下：

```
arp -d [IP 位址]
```

- 請參考以下範例：

```
C:\>arp -a

Interface: 203.74.205.111 --- 0X10003
Internet Address      Physical Address      Type
203.74.205.1          00-10-7b-c1-ec-98     dynamic
203.74.205.3          00-10-b5-3a-91-75     dynamic
203.74.205.7          00-10-b5-3a-91-b8     dynamic
203.74.205.11         00-10-b5-3a-91-dc     dynamic

C:\>arp -d 203.74.205.11
```

原先有 4 筆紀錄

← 刪除 203.74.205.11 這筆紀錄

刪除紀錄

```
C:\>arp -a

Interface: 203.74.205.111 --- 0X10003
Internet Address      Physical Address      Type
203.74.205.1          00-10-7b-c1-ec-98     dynamic
203.74.205.3          00-10-b5-3a-91-75     dynamic
203.74.205.7          00-10-b5-3a-91-b8     dynamic
                ←── 少了 203.74.205.11 這筆紀錄
C:\>
```

新增紀錄

- 在 ARP 快取中新增一筆靜態紀錄的語法如下：`arp -s [IP 位址] [MAC 位址]`

```
C:\>arp -s 203.74.205.42 00-00-e8-97-73-86        ← 新增這筆紀錄

C:\>arp -a

Interface: 203.74.205.111 --- 0X10003
Internet Address      Physical Address      Type
203.74.205.1          00-10-7b-c1-ec-98     dynamic
203.74.205.3          00-10-b5-3a-91-75     dynamic
203.74.205.7          00-10-b5-3a-91-b8     dynamic
203.74.205.11         00-10-b5-3a-91-dc     dynamic
203.74.205.42         00-00-e8-97-73-86     static
                                              ↑
                                新增的紀錄, 請注意 Type 欄位值爲 static

C:\>
```

9-3-2 ARPWATCH

- Linux 的 ARPWATCH 可監聽與記錄區域網路中的 ARP 封包, 並透過電子郵件將結果報告給管理員, 或直接將結果顯示在螢幕上。

經由電子郵件

- 執行 arpwatch 後, 若偵測到新的 ARP 記錄, 即透過電子郵件來報告。以下為電子郵件的內容：

```
Date: Mon, 20 Mar 2006 16:24:00 +0800
From: Arpwatch <arpwatch@localhost.localdomain>
To: root@localhost.localdomain
Subject: new station

        hostname: <unknown>                          ← 主機名稱
      ip address: 203.74.205.96                      ← IP 位址
ethernet address: 0:0:e8:97:73:95                    ← 網路卡的硬體位址
 ethernet vendor: Accton Technology Corporation      ← 網路卡的製造商
       timestamp: Mon, Mar 20, 2006 16:23:25 +0800   ← 發生的時間
```

直接顯示在螢幕

- 若要直接在螢幕上顯示結果, 請執行：

```
arpwatch -d
```

- 若偵測到新的 ARP 記錄, 則螢幕上會顯示如下的內容：

```
# arpwatch -d

From: arpwatch (Arpwatch)
To: root
Subject: new station
```

直接顯示在螢幕

```
          hostname : <unknown>
        ip address : 203.74.205.22
  ethernet address : 0:0:e8:97:70:ea
   ethernet vendor : Accton Technology Corporation
         timestamp : Mon, Mar 20, 2006 16:28:59 +0800
...
```

9-4 ICMP 簡介

- IP 在傳送封包時, 只是單純的將 IP 封包送出即完成任務。
- 在傳送過程中若發生問題, 則是由上層的協定來負責確認、重送等工作。
- ICMP（Internet Control Message Protocol）這個協定在 IP 路由的過程中若發生問題則將此狀況通知 IP 封包的來源端。

ICMP 簡介

- ICMP
  - 在網路層運作的協定
  - 一般視為是 IP 的輔助協定, 常用來『報告錯誤』。
  - 網管人員也可利用適當的工具程式發出 ICMP 封包, 以便測試網路連線或排解問題等等。

9-4-1 ICMP 封包的封裝方式

- ICMP 封包實際上是以 IP 封包的形式在網路上傳送。因此，ICMP 封包的外層必須以如下方式來包裝：

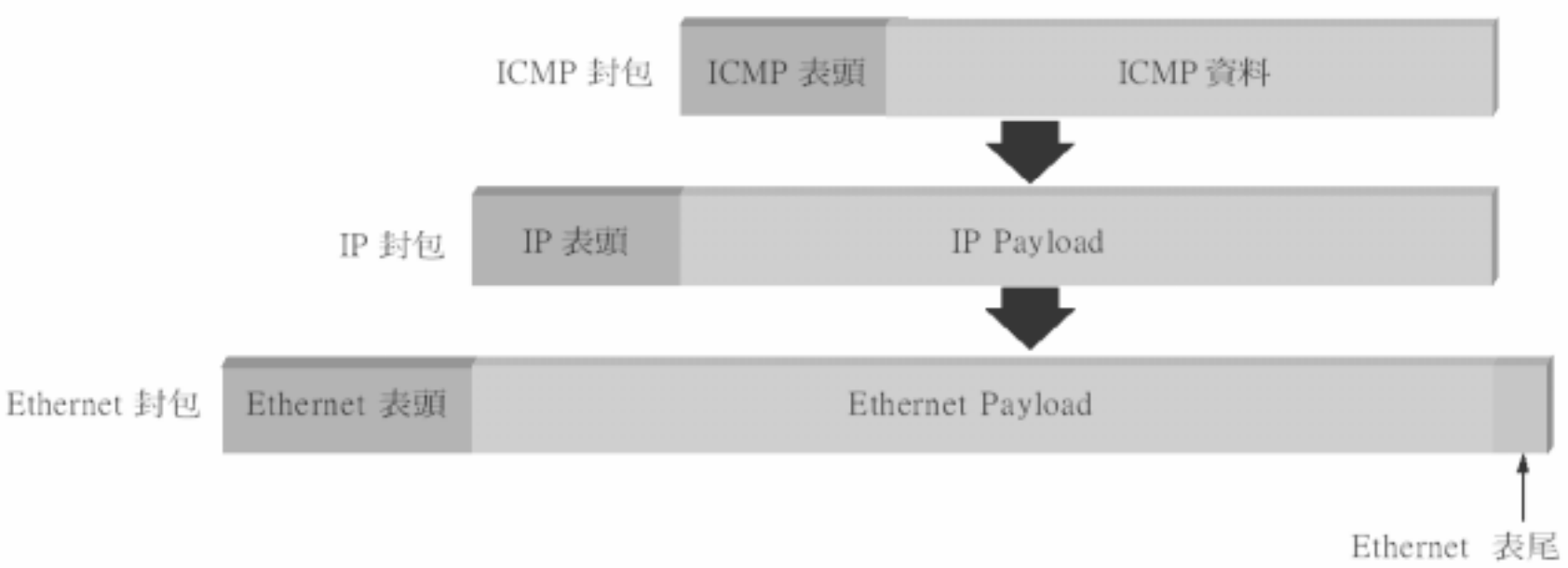


圖 9-04　ICMP 封包的封裝方式

9-4-2 ICMP 封包的欄位格式

- ICMP 封包可分為以下兩部份：

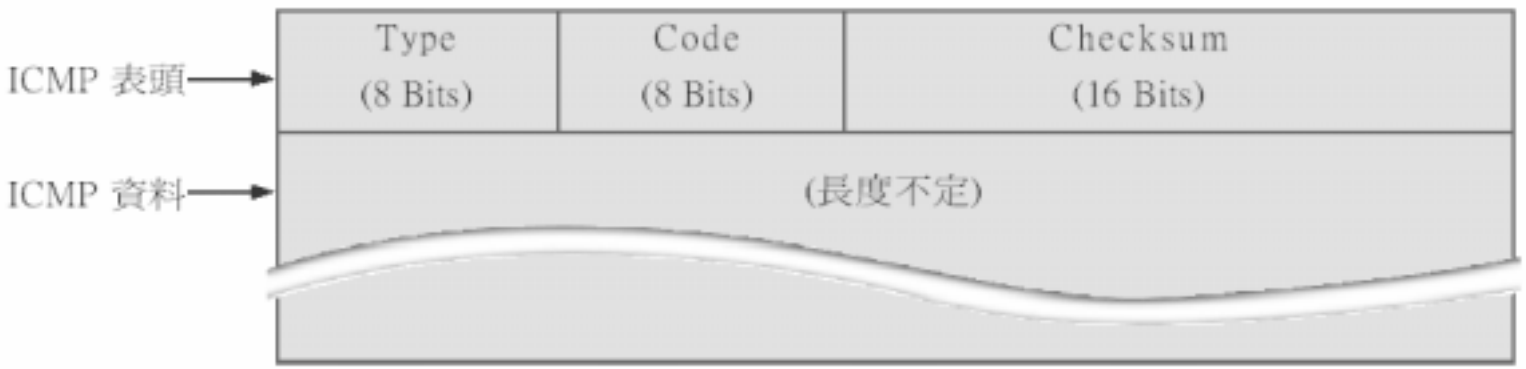


圖 9-05　ICMP 封包結構

- 每個 ICMP 封包都會有 ICMP 表頭, 其中包含了 3 個固定長度的欄位：Type、Code 與 Checksum。至於 ICMP 資料的部份, 則會隨著 ICMP 封包的類型而異。

Type（類型）

- 長度為 1 Byte。本欄位定義了 ICMP 封包的類型，每一種類型會有不同的功能。

表 9-03　各 Type 欄位值所對應的封包類型

Type 欄位值	ICMP 封包類型
0	Echo Reply*
3	Destination Unreachable*
4	Source Quench*
5	Redirect*
8	Echo Request*
9	Router Advertisement
10	Router Solicitation
11	Time Exceeded for a Datagram*
12	Parameter Problem on a Datagram
13	Timestamp Request
14	Timestamp Reply
17	Address Mask Request
18	Address Mask Reply

Code（代碼）

- 長度為 1 Byte。
- 每種 ICMP 封包類型可再根據 Code 欄位細分為各種不同的用途。例如：
  Destination Unreachable 類型的 ICMP 封包便利用 Code 欄位值來區分無法傳遞 IP 封包的各種情況。
- 不過, 大部份 ICMP 封包類型（Type）只定義了一種 Code 欄位值。

Checksum（錯誤檢查碼）

- 長度為 2 Bytes, 記錄了 ICMP 封包的錯誤檢查碼。

ICMP 資料

- ICMP 資料會隨著 Type 欄位值, 定義不同的欄位。

9-5 各類型的 ICMP 封包

- 9-5-1 回應要求與回應答覆
- 9-5-2 無法送達目的
- 9-5-3 降低來源端傳送速度
- 9-5-4 重新導向
- 9-5-5 傳送逾時

9-5-1 回應要求與回應答覆

- 回應要求與回應答覆（Echo Request / Echo Reply）可說是最常見的 ICMP 封包類型，可用來排解網路問題，包括 IP 路由的設定、網路連線等等。
- 回應要求與回應答覆必須以配對的方式來運作：

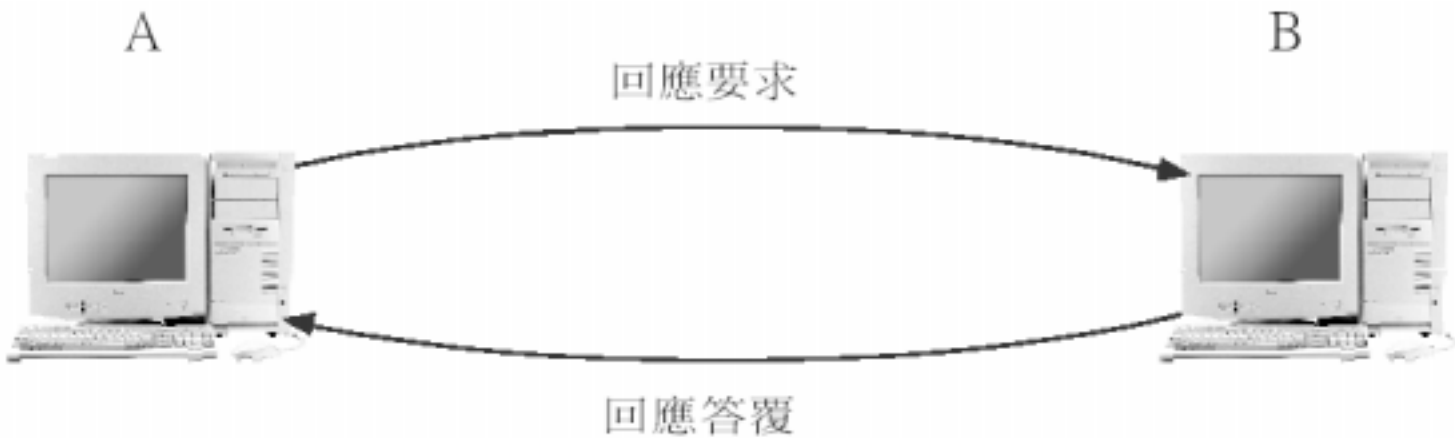


圖 9-06　回應要求與回應答覆的運作方式

回應要求與回應答覆

- 1. A 主動發出回應要求封包給 B。
- 2. B 收到回應要求後, 被動發出回應答覆封包給 A。
- 由於 ICMP 封包都是包裝成 IP 封包的形式來傳送, 因此, 若能完成上述步驟, A 便能確認以下事項：
  - B 裝置存在, 且運作正常。
  - A、B 之間的網路連線狀況正常。
  - A、B 之間的 IP 路由正常。

回應要求與回應答覆

- 回應要求與回應答覆的 ICMP 表頭為：

表 9-04　回應要求與回應答覆封包的表頭欄位值

ICMP 封包名稱	Type	Code	Checksum
Echo Request	8	0	計算值
Echo Reply	0	0	計算值

回應要求與回應答覆

- 回應要求與回應答覆的 ICMP 資料包含了 Identifier、Sequence Number、Optional Data 等 3 個欄位：

圖 9-07　回應要求與回應答覆封包中的 ICMP 資料之結構

Identifier

- 長度為 2 Bytes, 作為識別之用。
- Identifier 欄位由 Echo Request 來源端裝置的程式。

Sequence Number

- 長度為 2 Bytes, 用來記錄序號。
- Sequence Number 欄位由 Echo Request 來源端裝置的程式所決定。
- 由於 Echo Request 與 Echo Reply 有配對的關係, Identifier 與 Sequence Number 兩個欄位合起來可識別特定配對的 Echo Request 與 Echo Reply。

Optional Data

- 本欄位由 Echo Request 來源端裝置的程式所決定, 可記錄一些選擇性的資料。
- 當裝置收到 Echo Request 後, 所發出的 Echo Reply 的 Optional Data 欄位值必須與收到的 Echo Request 相同。
- 來源端收到 Echo Reply 後, 會讀取 Optional Data 欄位, 確認此為配對的 Echo Reply。

9-5-2 無法送達目的

- 無法送達目的（Destination Unreachable）也是常見的 ICMP 封包類型。
- 在路由過程中若出現下列問題, 路由器或目的裝置便會發出此類型的 ICMP 封包, 通知 IP 封包的來源端。
  - 路由器無法將 IP 封包傳送出去。例如：在路由表中找不到合適的路徑, 或是連線中斷而無法將封包從合適的路徑傳出。
  - 目的裝置無法處理收到的 IP 封包。例如：目的裝置無法處理 IP 封包內所裝載的傳輸層協定。

無法送達目的

- 無法送達目的的 ICMP 表頭為：

表 9-05　無法送達目的封包的表頭欄位值

ICMP 封包名稱	Type	Code	Checksum
Destination Unreachable	3	0-12	計算值

無法送達目的

- Code 欄位值可從 0 至 12, 以下僅舉例說明 3 個較常見的值：
  - 0, Network Unreachable
  - 1, Host unreachable
  - 2, Protocol unreachable

無法送達目的

- 無法送達目的的 ICMP 資料欄位有 Unused 和 IP 表頭與 Payload 兩個欄位：

圖 9-08　無法送達目的封包中的 ICMP 資料之結構

無法送達目的欄位

- Unused:長度為 4 Bytes。本欄位未定義用途, 欄位內容必須為 0。
- IP表頭與Payload
  - 將問題封包的 IP 表頭（長度不定）, 以及 IP Payload 的前 8 Bytes 寫入本欄位。
  - IP 封包來源端, 可根據本欄位的資訊, 得知是哪一個 IP 封包有問題, 並據此決定因應的措施

9-5-3 降低來源端傳送速度

- 當路由器因為來往 的 IP 封包太多, 以致於來不及處理時, 便會發出降低來源端傳送速度（Source Quench）的 ICMP 封包給 IP 封包的來源端裝置。
- 在實作時, 廠商通常是以路由器的 CPU 或緩衝區的負荷作為衡量標準。
- 降低來源端傳輸速度的 ICMP 表頭為：

表 9-06　降低來源端傳輸速度的封包表頭欄位值

ICMP 封包名稱	Type	Code	Checksum
Source Quench	4	0	計算值

降低來源端傳送速度

- 降低來源端傳輸速度的 ICMP 資料包含了 Unused 和 IP 表頭與 Payload 等 2 個欄位：

圖 9-09　降低來源傳輸速度的封包中的 ICMP 資料之結構

- 這兩個欄位的內容與長度與無法送達目的相同。

9-5-4 重新導向

- 當路由器發現主機所選的路徑並非最佳路徑時，會送出 ICMP 重新導向（Redirect）封包，通知主機較佳的路徑。以下圖為例：

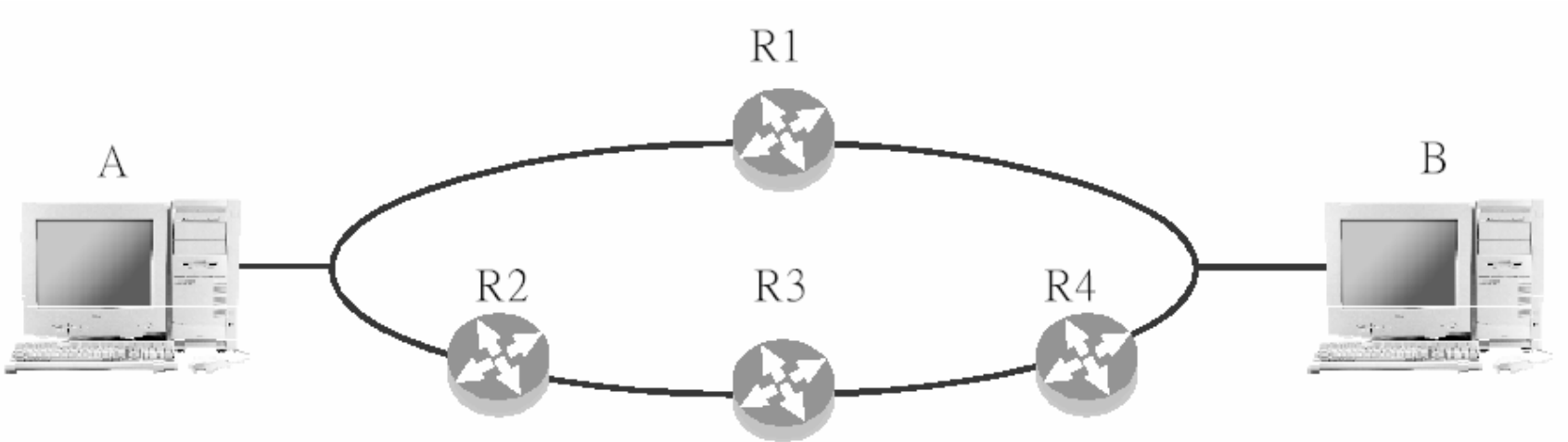


圖 9-10　可能產生 ICMP 重新導向的網路環境

重新導向

- 重新導向封包的表頭為：

表 9-07　重新導向封包的表頭欄位值

ICMP 封包名稱	Type	Code	Checksum
Redirect	5	0-3	計算值

重新導向

- Code 欄位值可從 0 至 3, 以下僅說明較常見的值：
  - 1, Host：路由器收到 IP 封包後, 若在路由表中找到更合適的路徑, 便會發出此訊息給 IP 封包的來源端裝置。
  - 3, TOS and Host：路由器收到 IP 封包後, 若在路由表中找到更符合 IP 封包要求的 TOS 路徑, 便會發出本訊息給 IP 封包的來源端裝置。

Router IP Address

- Redirect 的 ICMP 資料包含了 Router IP Address 和 IP 表頭與 Payload 等 2 個欄位。

圖 9-11　重新導向封包中的 ICMP 資料之結構

- 長度為 4 Bytes, 用來通知主機較佳的路由器。

IP 表頭與 Payload

- 將 IP 封包的 IP 表頭, 以及 IP Payload 的前 8 Bytes 寫入本欄位, 提供來源端 IP 封包額外的資訊。

9-5-5 傳送逾時

- 當路由器收到存活時間為 1 的 IP 封包時, 會將此 IP 封包丟棄, 然後送出傳送逾時（Time Exceeded）的 ICMP 封包給 IP 封包的來源裝置。
- 當 IP 封包在傳送過程中發生切割時, 必須在目的裝置重組切割後的 IP 封包。重組的過程中若在指定的時間內未收到全部切割後的 IP 封包, 目的裝置也會發出傳送逾時的 ICMP 封包給 IP 封包的來源裝置。

傳送逾時

- 傳送逾時的 ICMP 表頭為：

表 9-08　傳送逾時的封包表頭欄位值

ICMP 封包名稱	Type	Code	Checksum
Time Exceeded	11	0-1	計算值

傳送逾時

- Code 欄位值可為 0 或 1, 請參考以下說明：
  - 0, TTL count exceeded：當路由器收到 TTL 值為 1 的 IP 封包時, 便會發出此訊息給 IP 封包的來源端裝置。
  - 1, Fragment reassembly time exceeded：IP 封包目的裝置重組 IP Fragment 時, 若在指定的時間內未收到全部的 IP Fragment, 便會發出本訊息給 IP 封包的來源端裝置。

傳送逾時

- 傳送逾時的 ICMP 資料包含了 Unused 與 IP 表頭與 Payload 等 2 個欄位：

圖 9-12　傳送逾時封包中的 ICMP 資料之結構

Unused

- 長度為 4 Bytes。本欄位未定義用途, 欄位內容必須為 0。

IP 表頭與 Payload

- 當路由器或目的裝置要發出傳送逾時的封包時，會將問題封包的 IP 表頭，以及 IP Payload 的前 8 Bytes 寫入本欄位，提供 IP 封包來源端額外的資訊。

9-6 ICMP 工具程式

- 大部份作業系統都會提供一些 ICMP 工具程式, 方便使用者測試網路連線狀況。以下便以 Windows 作業系統為例, 介紹數種常見的 ICMP 工具程式。

9-6-1 PING

- PING 工具程式可用來發出 ICMP 回應要求封包。網管人員可利用 PING 工具程式，發出回應要求給特定的主機或路由器，以診斷網路的問題。

利用 PING 來診斷網路問題

- 當您發現網路連線異常時, 可參考下列步驟, 利用 PING 工具程式, 由近而遠逐步鎖定問題所在。
  - 1. ping 127.0.0.1
  - 2. ping 本機 IP 位址
  - 3. ping 對外連線的路由器
  - 4. ping 網際網路上電腦的 IP 位址
  - 5. ping 網際網路上電腦的網址

1. ping 127.0.0.1

- 127.0.0.1 是所謂的 Loopback 位址（請參考第 8 章）。
- 目的位址為 127.0.0.1 的封包不會送到網路上, 而是送至本機的 Loopback 驅動程式。
- 此一動作主要是用來測試本機的 TCP / IP 協定是否正常運作。

2. ping 本機 IP 位址

- 若步驟 1 中本機 TCP / IP 設定正確, 接下來可試試看網路裝置是否正常。若網路裝置有問題（例如：舊型網路卡的 IRQ 設定有誤）, 則不會回應。

3. ping 對外連線的路由器

- 也就是 PING『預設閘道』（在Windows 98 稱為『通訊閘』）的 IP 位址。若成功, 代表內部網路與對外連線的路由器正常。

4. ping 網際網路上電腦的 IP 位址

- 您可以隨便找一台網際網路上的電腦, PING 它的 IP 位址。如果有回應, 代表 IP 設定全部正常。

5. ping 網際網路上電腦的網址

- 您可以隨便找一台網際網路上的電腦, PING 它的網址, 例如：www.hinet.net.（Hinet 的 WWW 伺服器）。如果有回應, 代表 DNS 設定無誤。

PING 的語法與參數

- PING 的語法如下：

  PING [參數] [網址或 IP 位址]

- PING 的參數相當多, 以下僅說明較常用的參數：

表 9-09 PING 的參數

參數	意義
-a	執行 DNS 反向查詢 (由 IP 位址查出 FQDN, 詳見第 13 章), 預設不會執行此查詢。
-i <存活時間>	設定 IP 封包的存活時間, 預設爲 128 。
-n <封包數量>	每次執行時, 發出回應要求封包的數量, 預設爲 4 。
-t	持續發出回應要求封包, 直到按 Ctrl + C 才停止。
-w <等待時間>	等待回應答覆的時間。<等待時間> 的單位爲千分之一秒, 預設值爲 1000, 亦即 1 秒。

PING 範例

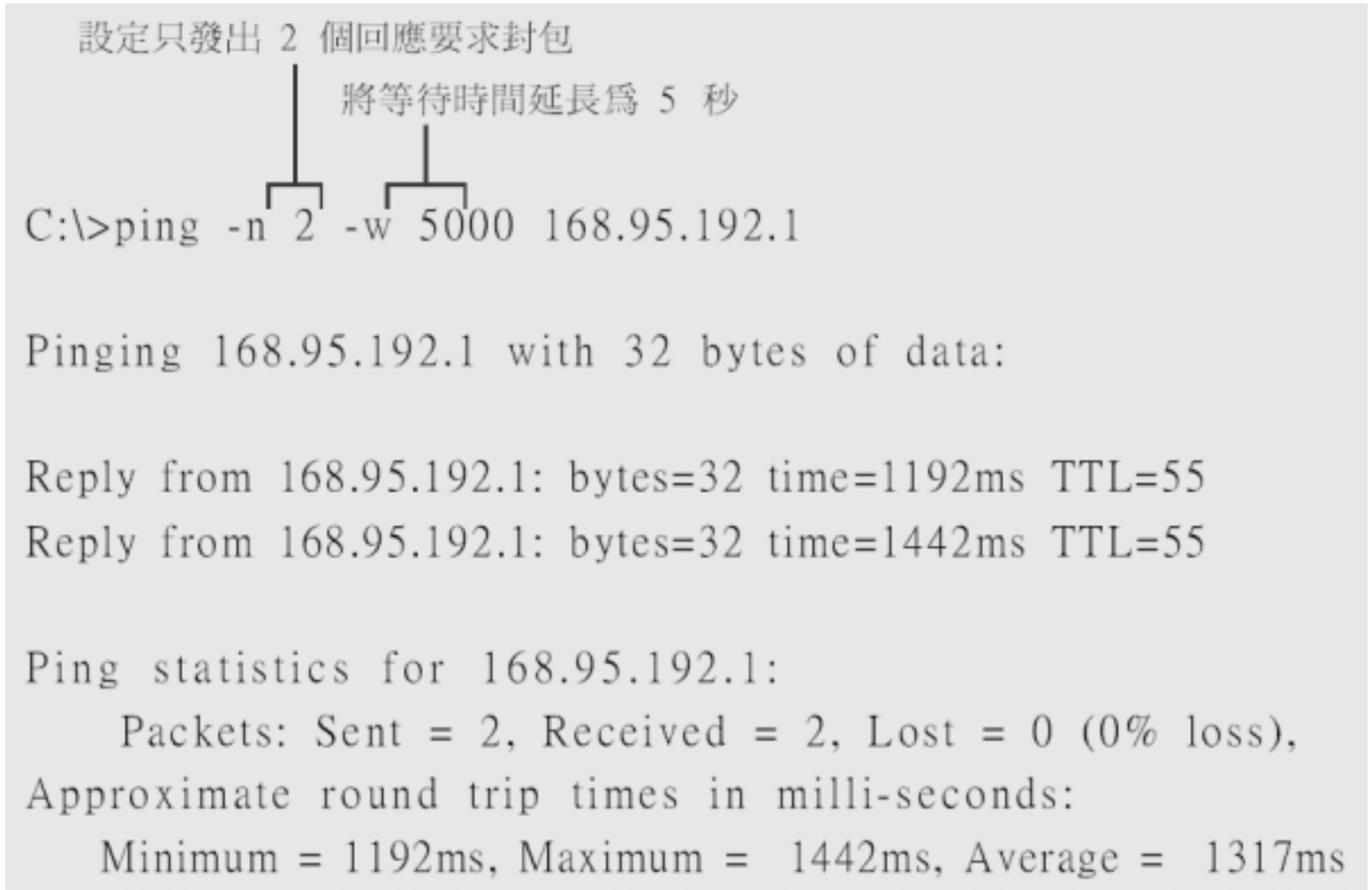
- 若要讓 PING 執行 DNS 反向查詢：

```
C:\>ping -a 168.95.192.1
                  ┌──反向查詢所得的名稱
Pinging hntp1.hinet.net [168.95.192.1] with 32 bytes of data:

Reply from 168.95.192.1: bytes=32 time=1292ms TTL=55
Request timed out.──┐
Request timed out.  ├──超過預設的等待時間未獲回應，便會出現此種訊息
Request timed out.──┘

Ping statistics for 168.95.192.1:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
Approximate round trip times in milli-seconds:
   Minimum = 1292ms, Maximum =  1292ms, Average =  323ms
```

PING 範例

- 利用 -w 參數, 可以延長等待回應答覆的時間。
- 此外, 也可以結合多個參數一起使用, 例如以下的例子同時用 -n 參數設定只發出 2 個回應要求封包：

PING 範例


設定只發出 2 個回應要求封包
將等待時間延長爲 5 秒
C:\>ping -n 2 -w 5000 168.95.192.1
Pinging 168.95.192.1 with 32 bytes of data:
Reply from 168.95.192.1: bytes=32 time=1192ms TTL=55
Reply from 168.95.192.1: bytes=32 time=1442ms TTL=55
Ping statistics for 168.95.192.1:
Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1192ms, Maximum = 1442ms, Average = 1317ms

9-6-2 TRACERT

- TRACERT（Trace Route）工具程式可找出本機電腦至目的 IP 位址所經過的路由器。
  - TRACERT 原理
  - TRACERT 的語法與參數
  - TRACERT 範例

TRACERT 原理

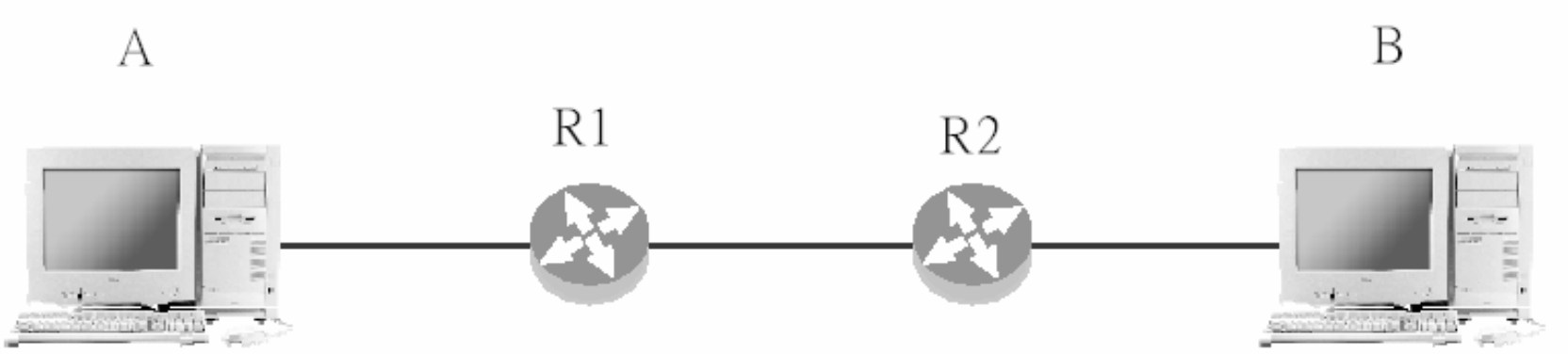


圖 9-13　說明 TRACERT 原理的網路環境

TRACERT 原理

- 首先假設如下的網路環境：
- 若從 A 主機執行 TRACERT, 並將目的地設為 B 主機, 則 TRACERT 會利用以下步驟, 找出沿途所經過的路由器：
  - 1. 發出回應要求封包, 該封包的目的地設為 B, 存活時間設為 1。為了方便說明, 我們將所有封包都加以命名, 此封包命名為『回應要求 1』。

TRACERT 原理

- 2. R1 路由器收到『回應要求 1』後, 因為存活時間為 1, 因此會丟棄此封包, 然後發出『傳送逾時 1』給 A。
- 3. A 收到『傳送逾時 1』之後, 便可得知到 R1 為路由過程中的第一部路由器。接著, A 再發出『回應要求 2』, 目的地設為 B, 存活時間設為 2。

TRACERT 原理

- 4.『回應要求 2』會先送到 R1, 然後再轉送至 R2。到達 R2 時, 『回應要求 2』的存活時間為 1, 因此, R2 會丟棄此封包, 然後傳送『傳送逾時 2』給 A。
- 5. A 收到『傳送逾時 2』之後, 便可得知到 R2 為路由過程中的第二部路由器。接著, A 再發出『回應要求 3』, 目的地設為 B, 存活時間設為 3。

TRACERT 原理

- 6.『回應要求 3』會經由 R1、R2 然後轉送至 B。B 收到此封包後便會回應『回應答覆 1』給 A。
- 7. A 收到『回應答覆 1』之後便大功告成。

TRACERT 原理

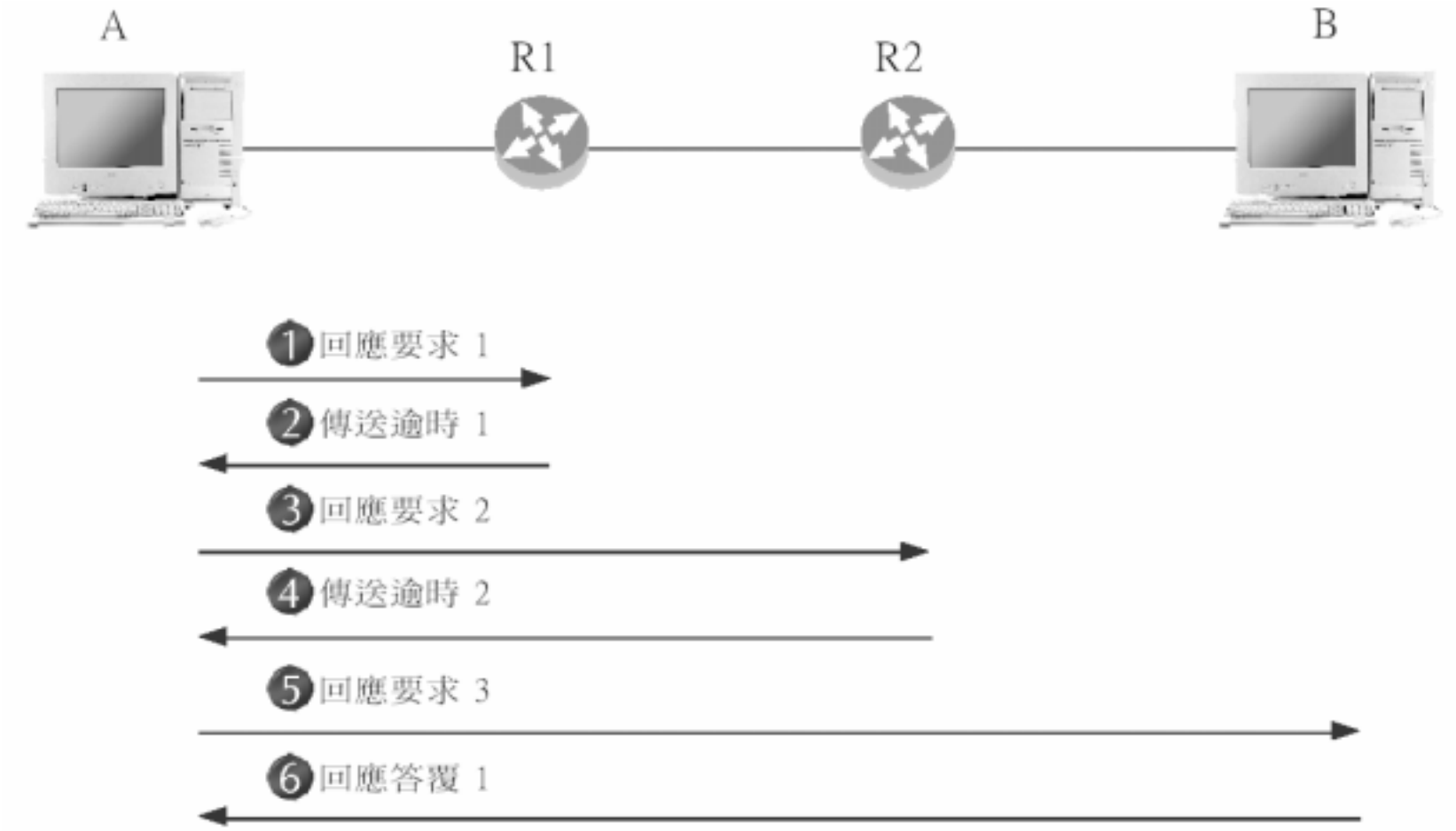


圖 9-14　TRACERT 的過程

TRACERT 原理

- 以上的工作原理說明係以微軟的 Windows 9x / 2000 / XP / 2003 為範例。若是 Unix、Linux 作業系統的 traceroute, 則不使用 ICMP 封包, 而是用 UDP 封包配合特定的連接埠。
- 由於第 12 章才會介紹 UDP 的特性, 因此在本章不予說明。有興趣研究的讀者不妨瀏覽 http://www.develcon.com/kb/i014.htm 網頁。

TRACERT 的語法與參數

- TRACERT 的語法如下：

```
TRACERT [參數] [網址或 IP 位址]
```

- 以下為 TRACERT 常用的參數：

表 9-10　TRACERT 的參數

參數	意義
-d	TRACERT 預設會執行 DNS 反向查詢。若不要反向查詢，請使用此參數。
-h <存活時間>	TRACERT 每次發出回應要求時存活時間會加 1 。本參數可設定存活時間最大值，預設爲 30 。
-w <等待時間>	等待傳送逾時或回應答覆的時間。<等待時間> 的單位爲千分之一秒，預設值爲 1000，亦即 1 秒。

TRACERT 範例

- 以下我們不使用任何參數，利用 TRACERT 找出至目的主機沿途所經的路由器。

```
C:\>tracert 168.95.192.1

Tracing route to hntp1.hinet.net [168.95.192.1]
over a maximum of 30 hops:

  1  <10 ms  <10 ms  <10 ms  203.74.205.3
  2  <10 ms  <10 ms  <10 ms  c137.h203149174.is.net.tw [203.149.174.137]
  3   50 ms   60 ms   60 ms  10.1.1.70
  4   60 ms   60 ms   60 ms  c248.h202052070.is.net.tw [202.52.70.248]
  5  290 ms   60 ms   60 ms  ISNet-PC-TWIX-T3.rt.is.net.tw [210.62.131.225]
  6   70 ms   50 ms   70 ms  210.62.255.5
  7   50 ms   70 ms   51 ms  210.65.161.126
  8   51 ms   50 ms   50 ms  168.95.207.21
  9   60 ms   50 ms   50 ms  hntp1.hinet.net [168.95.192.1]

Trace complete.
```

TRACERT 範例

- TRACERT 的結果顯示了以下資訊：
  - 由近到遠, 顯示沿途所經的每部路由器。以上範例顯示, 從來源端主機至 168.95.192.1 主機必須經過 8 部路由器。
  - 顯示每部路由器回應的時間。由於 TRACERT 會傳送 3 個回應要求封包給每部路由器, 因此會有 3 個回應時間。

9-7 擷取 ICMP 封包

- 接下來我們將利用 NetAnalyzer, 在執行 PING 與 TRACERT 時, 擷取過程中所有的封包。

9-7-1 PING

- 首先我們在 192.168.0.40 這部主機上開始擷取封包, 然後執行 PING 工具程式, 目的主機為 192.168.0.140。
- 在 NetAnalyzer 中利用設定過濾器功能, 只擷取 ICMP 的封包。以下依序說明檢視前兩對 Echo Request / Echo Reply 封包的情形。

PING

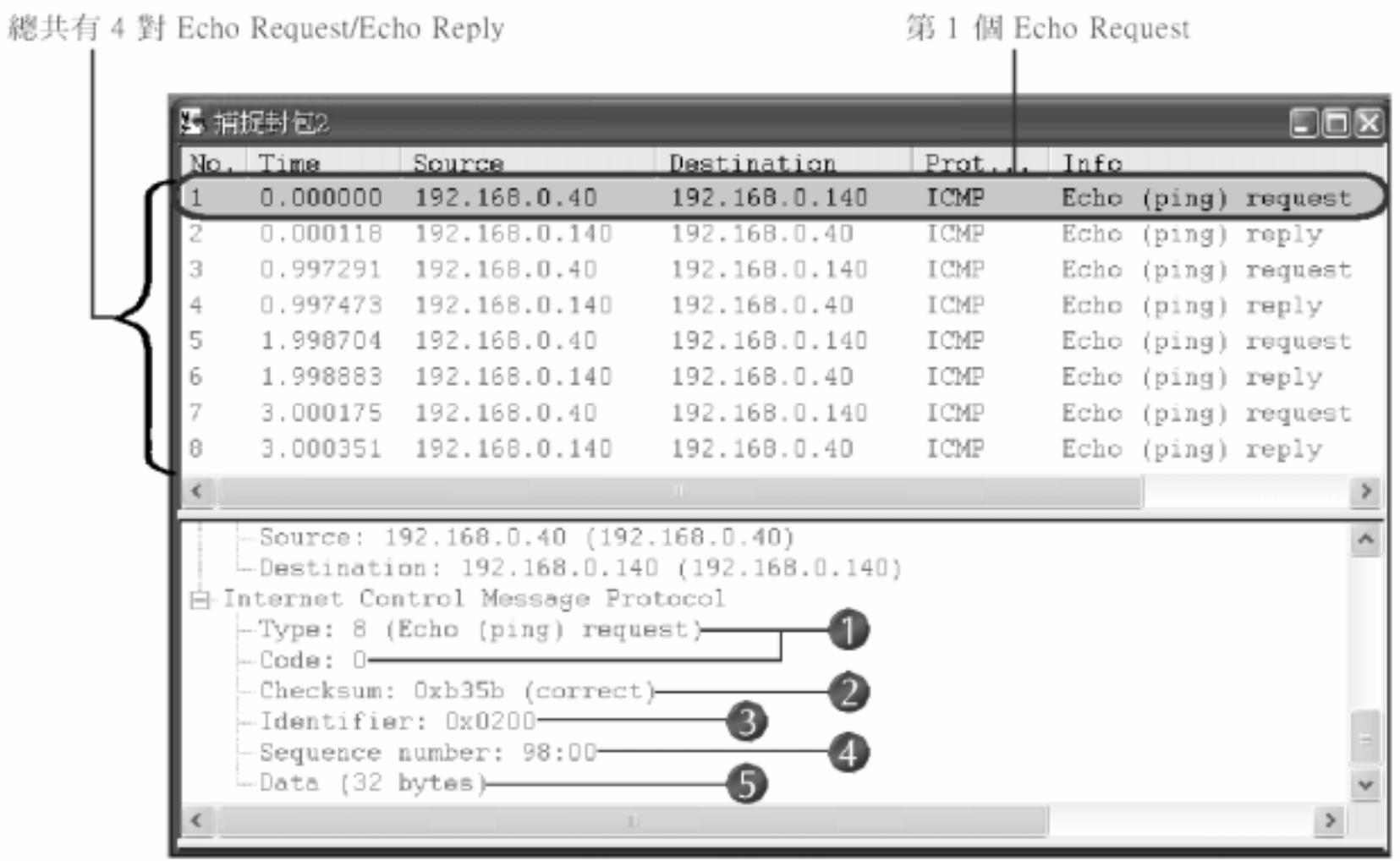


圖 9-15 Echo Request 封包內容

PING

❶Type = 8、Code = 0, 代表此為 Echo Request 封包。

❷錯誤檢查碼。

❸Windows 2000 的 PING 工具程式, Identifier 欄位皆為 512。

❹每個 Echo Request 封包的 Sequence Number 都不一樣, 每次遞增 256。

❺32 Bytes 的 Optional Data 欄位。

PING

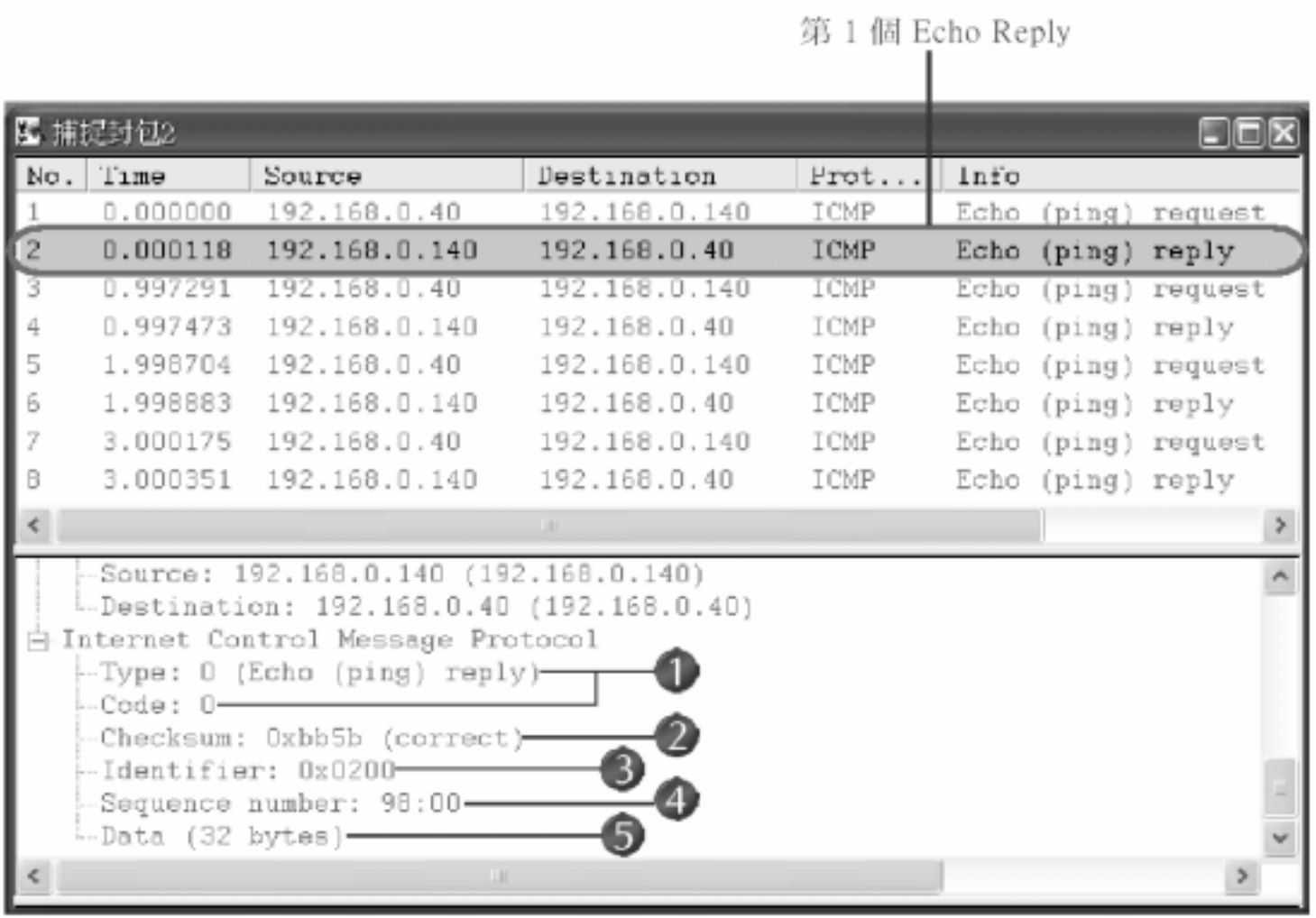


圖 9-16　Echo Reply 封包內容

PING

❶Type = 0、Code = 0, 代表此為 Echo Reply 封包。

❷錯誤檢查碼。

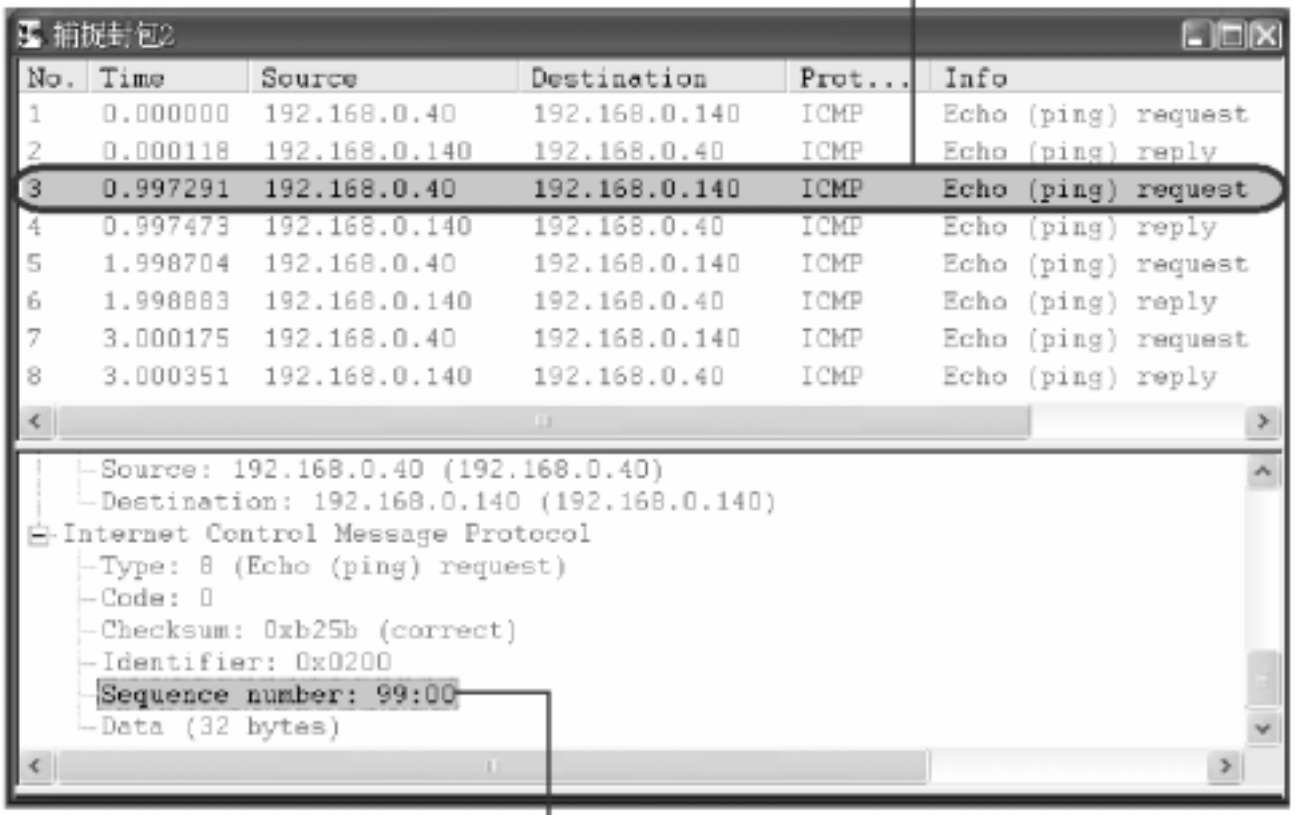
❸Echo Reply 的 Identifier 欄位必須與配對的 Echo Request 相同, 所以也是 512。

❹Echo Reply 的 Sequence Number 欄位必須與配對的 Echo Request 相同, 所以也是 0 x 9800 。

❺Echo Reply 的 Optional Data 欄位必須與配對的 Echo Request 相同。

PING

第 2 個 Echo Request



Sequence Number 值爲 0x9900, 減去 256 等於 0x9800,
亦即第 1 個 Echo Request 的 Sequence Number 值

圖 9-17 第 2 個 Echo Request

PING

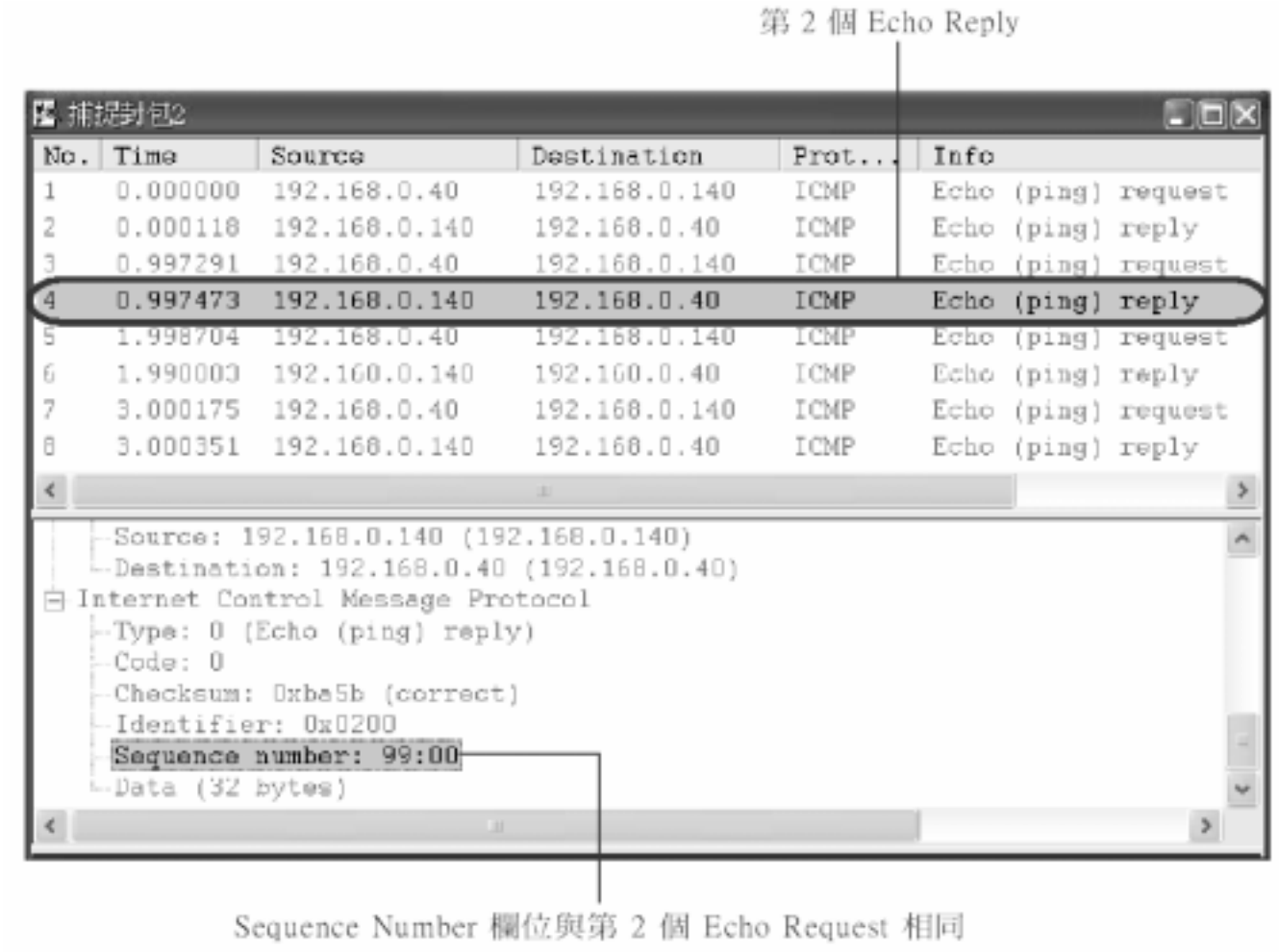


圖 9-18　第 2 個 Echo Reply

9-7-2 TRACERT

- 我們仍舊在 192.168.0.40 這部主機上擷取封包, 然後執行 TRACERT 工具程式, 目的主機為 www.hinet.net（IP 位址為 168.95.1.88）：

TRACERT

```
C:\>tracert www.hinet.net

Tracing route to WWW.HINET.NET [168.95.1.88]
over a maximum of 30 hops:

 1    <1 ms   <1 ms   <1 ms  192.168.0.3
 2     2 ms    1 ms    1 ms  c29.h061013225.is.net.tw [61.13.225.29]
 3   103 ms  210 ms  282 ms  10.1.1.70
 4   183 ms   66 ms  144 ms  c230.h202052070.is.net.tw [202.52.70.230]
 5   160 ms  194 ms   93 ms  c49.h210062131.is.net.tw [210.62.131.49]
 6   206 ms   81 ms  173 ms  211.78.150.69
 7     *       *       *    Request timed out.
 8   256 ms  256 ms  245 ms  211.22.41.238
 9   285 ms  322 ms  339 ms  tp-s2-c12r2.router.hinet.net [211.22.32.202]
10   294 ms  206 ms  267 ms  tp-b-c6r2.router.hinet.net [168.95.2.49]
11   307 ms  312 ms  481 ms  www.hinet.net [168.95.1.88]

Trace complete.
```

TRACERT

- 由以上結果可以得知：
  - 封包從 192.168.0.40 出發到達 168.95.1.88, 總共經過 10 部路由器（編號 1　10 為路由器, 編號 11 為目的主機）。
  - 第 7 部路由器收到 Echo Request 封包後, 在 TRACERT 的等待期限內沒有回應, 因此以 * 代表回應時間。

TRACERT

- tracert 對每個節點發出 3 個 Echo Request 封包, 每個節點應該回覆 3 個封包, 總計有 11 個節點（10 部路由器 + 目的主機）發出封包, 所以總共應擷取到 66個封包。但是因為第 7 部路由器沒有回覆, 少了 3 個封包, 所以擷取的封包總數為 63 個。

- 我們將檢視第 1 個 Echo Request 與第 1 個 Time Exceeded 封包, 以及最後 1 個 Echo Request 與 Echo Reply 封包。

TRACERT

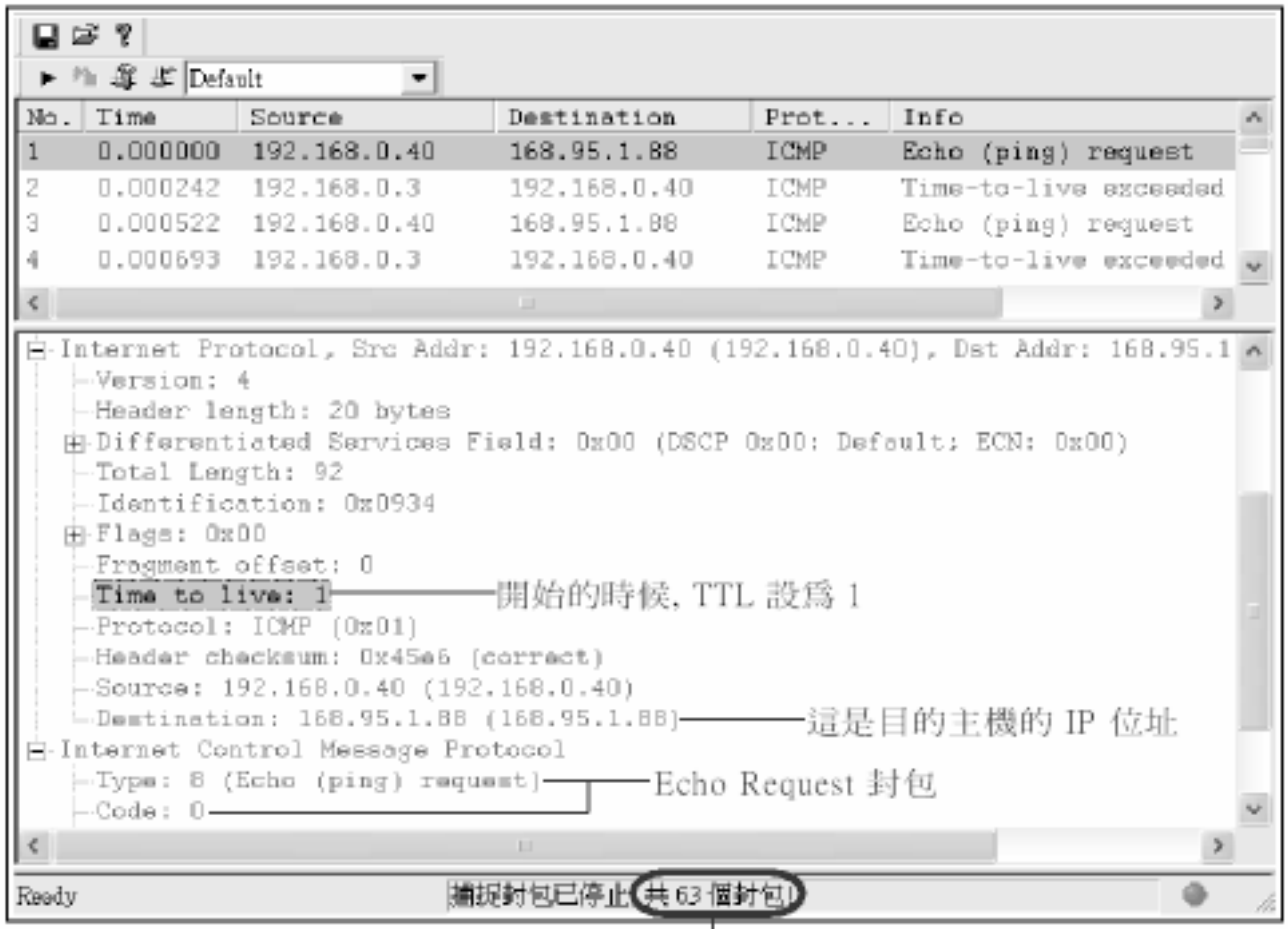


圖 9-19　第 1 個 Echo Request 封包

TRACERT

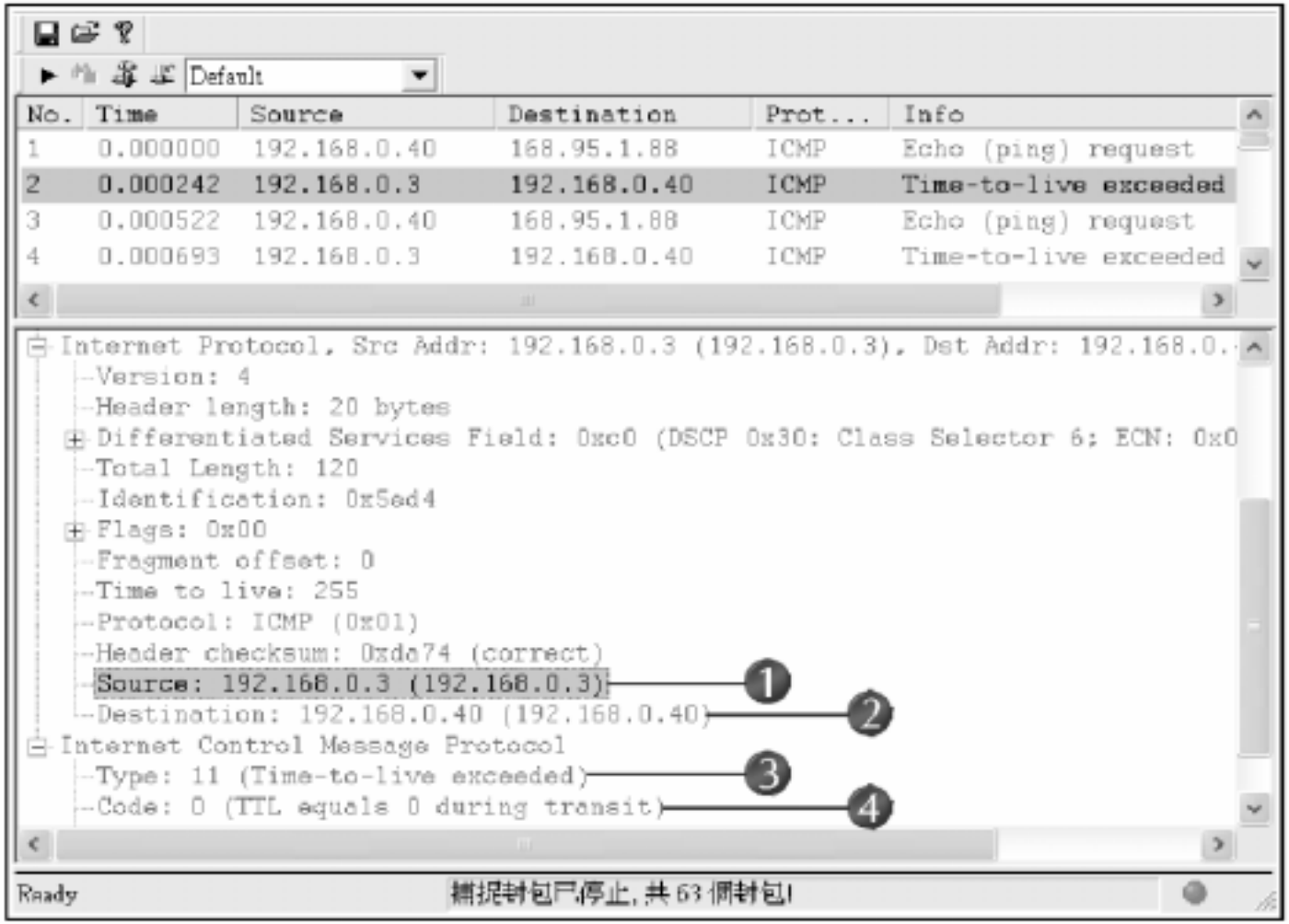


圖 9-20　第 1 個 Time Exceeded 封包

TRACERT

❶此封包是由第 1 部路由器發出, 因此 Source Address 記錄的是該路由器的 IP 位址。

❷這是我們執行 TRACERT 的主機 IP 位址。

❸ICMP 的 Type 欄位顯示此為 Time Exceeded 封包。

❹路由器會將收到的封包的 TTL 值減 1, 若所得的結果為 0, 便回覆此 Time Exceeded 封包。

TRACERT

圖 9-21　最後 1 個 Echo Request 封包

TRACERT

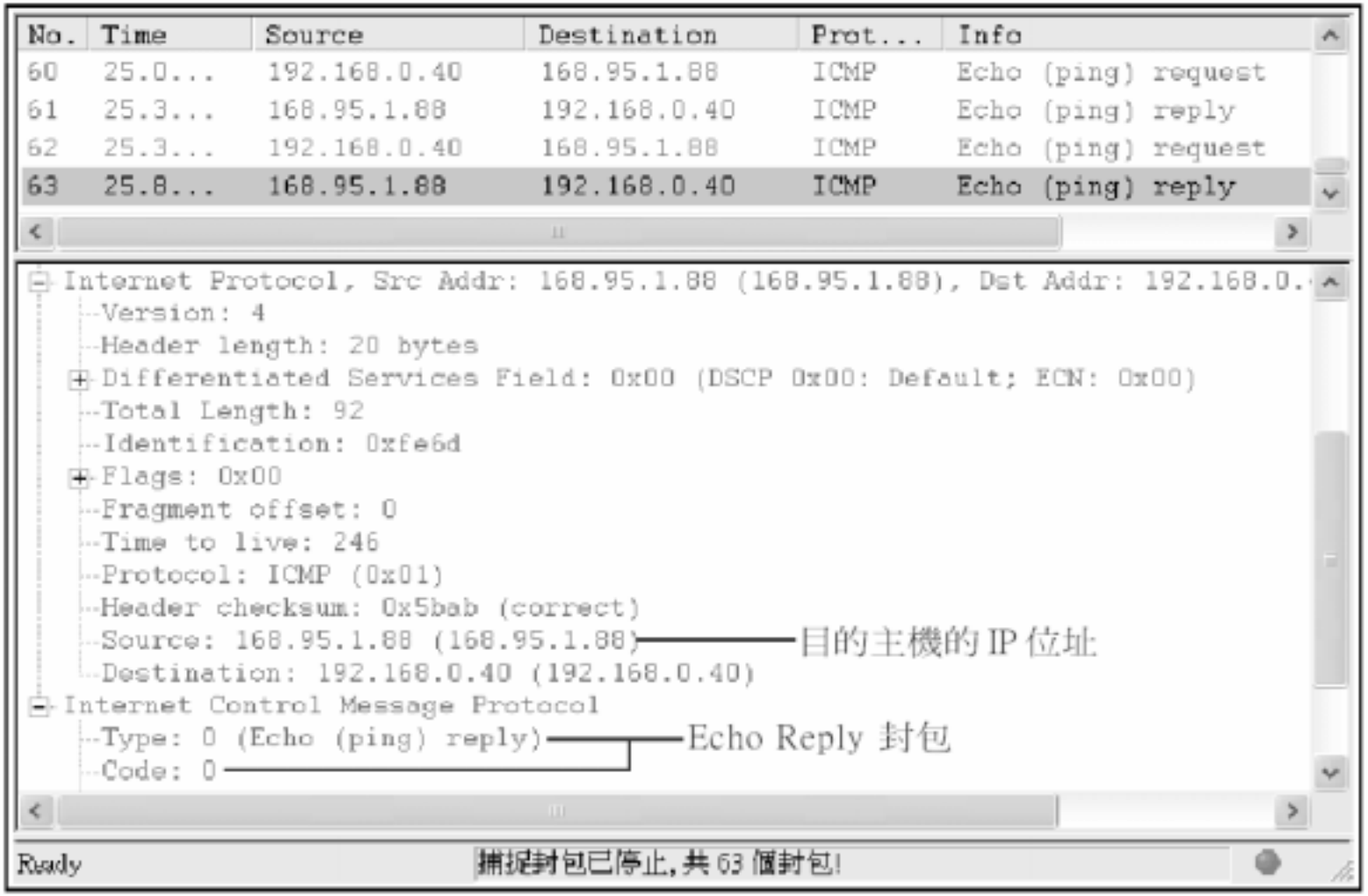


圖 9-22 Echo Reply 封包

TRACERT

- 在 TRACERT 的過程中, Echo Request 封包的 TTL 值漸增, 一直增加到可傳至目的主機, 此時目的主機便會傳回正常的 Echo Reply 封包。

Windows XP SP2 預設會擋下 Echo Request 類型的 ICMP 封包

- 基於安全上的考量, Windows XP SP2 作業系統預設會啟用它所內建的防火牆功能。
- 若您使用 PING 或 TRACERT 程式傳送 Echo Request 類型的 ICMP 封包到 XP SP2 電腦, 在預設的情況下該 ICMP 封包就會被 XP SP2 內建的防火牆擋掉, 因而得不到這台電腦的回應。

Windows XP SP2 預設會擋下 Echo Request 類型的 ICMP 封包

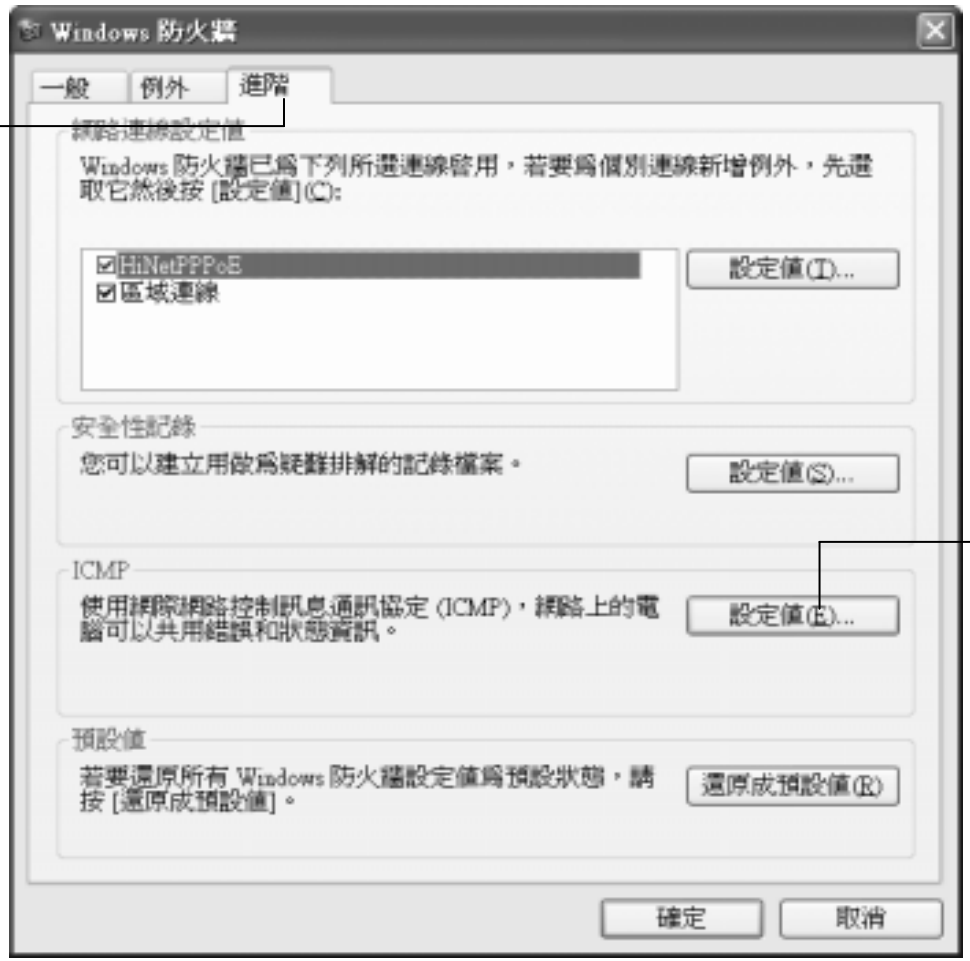
- 若要讓 XP SP2 的防火牆別擋掉外界傳入的 Echo Request 類型 ICMP 封包, 請執行『開始 / 控制台』命令, 在控制台視窗裡雙按資訊安全中心圖示, 開啟 Windows 資訊安全中心交談窗：

Windows XP SP2 預設會擋下 Echo Request 類型的 ICMP 封包

1 點選此項

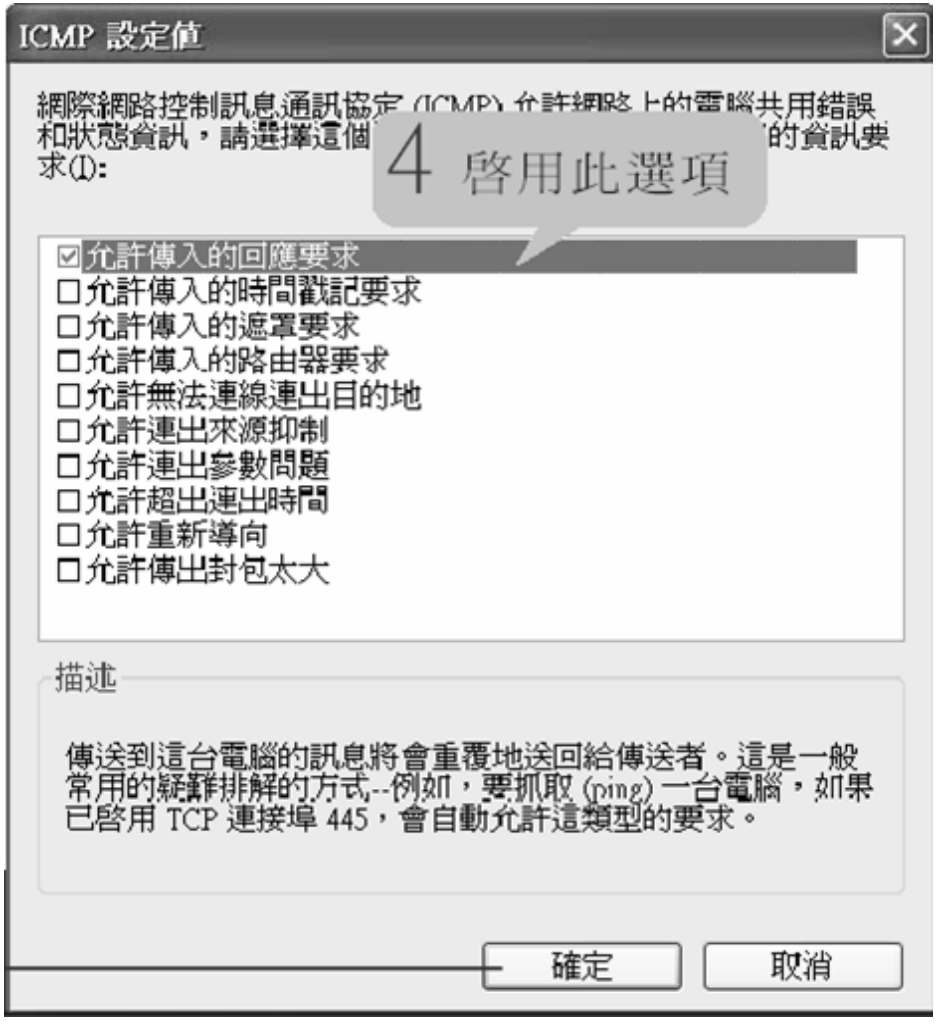
Windows XP SP2 預設會擋下 Echo Request 類型的 ICMP 封包

2 選取進階頁次



3 按下此鈕

Windows XP SP2 預設會擋下 Echo Request 類型的 ICMP 封包



5 按此確定鈕回到上一個交談窗後再按下確定鈕即可完成設定