

第 11 章

UDP 與 TCP

前言

- 11-1 UDP
- 11-2 TCP
- 11-3 TCP 傳送機制
- 11-4 TCP 連線
- 實作練習：SocketTest TCP/UDP 測試工具

UDP 與 TCP

- 在 DoD 模型中，傳輸層位於網路層與應用層之間，主要的功能是負責應用程式之間的通訊。



圖 11-1

TCP、UDP 對應於 OSI 模型中的傳輸層

11-1 UDP

UDP (User Datagram Protocol) 僅提供連接埠 (Port) 處理的功能。

- 記錄封包來源端與目的端的連接埠資訊, 正確地送達目的端的應用程式。
- 非連接式 (Connectionless) 的傳送。

使用 UDP 的考量：

- 為了要降低對電腦資源的需求。
- 應用程式本身已提供資料完整性的檢查機制

UDP

- 11-1-1 連接埠

11-1-1 連接埠

- UDP 最重要的功能和管理連接埠。
- 什麼是連接埠？
 - 連接埠的英文為 Port, 是一種邏輯上的概念。每一部使用 TCP/IP 的電腦, 都會有許多連接埠, 並使用編號加以區分。
 - 應用程式若經由 TCP/IP 存取資料, 就必須獨佔一個連接埠編號。
 - IP 位址與連接埠編號兩者合起來稱為 Socket Address, 可用來定義 IP 封包最後送達的終點, 亦即目的地應用程式。

連接埠

- 連接埠編號的原則

- 0~1023

此段的連接埠編號稱為『Well-Known』(廣為人知的)連接埠,主要給提供服務的伺服器應用程式使用。

- 1024~49151

此部份稱為『Registered』連接埠,保留給特定的應用程式或服務使用。

- 49152~65535

稱為『Dynamic』(動態)連接埠,由用戶端自行使用。

連接埠

表 11-1 常見的 Well-Known 連接埠

協定	連接埠編號	應用程式
UDP	53	DNS
UDP	68	DHCP Client
UDP	67	DHCP Server
UDP	520	RIP
TCP	19	NNTP
TCP	20	FTP Data
TCP	21	FTP Control
TCP	23	Telnet
TCP	25	SMTP
TCP	80	HTTP
TCP	110	POP3

用戶端有時也須要使用 Well-Known 連接埠。

有些伺服器需要 1 個以上的連接埠編號。例如：FTP 伺服器必須用到 2 個連接埠編號。

連接埠

- 使用自訂的伺服器連接埠編號
 - Well-Known 連接埠其實有點類似『約定俗成』的意思, 並不具有強制性

11-2 TCP

- TCP 為傳輸層的協定，也具備處理連接埠的功能。
 - 提供了一種『可靠』的傳送機制。
- TCP 『可靠』的傳輸方式，大致有以下幾種特性：
 - 資料確認與重送
 - 流量控制
 - 連線導向

11-3 TCP 傳送機制

- 11-3-1 確認與重送
- 11-3-2 雙向傳輸

11-3-1 確認與重送

- TCP 『可靠』的傳送機制簡而言之，就是『確認與重送』

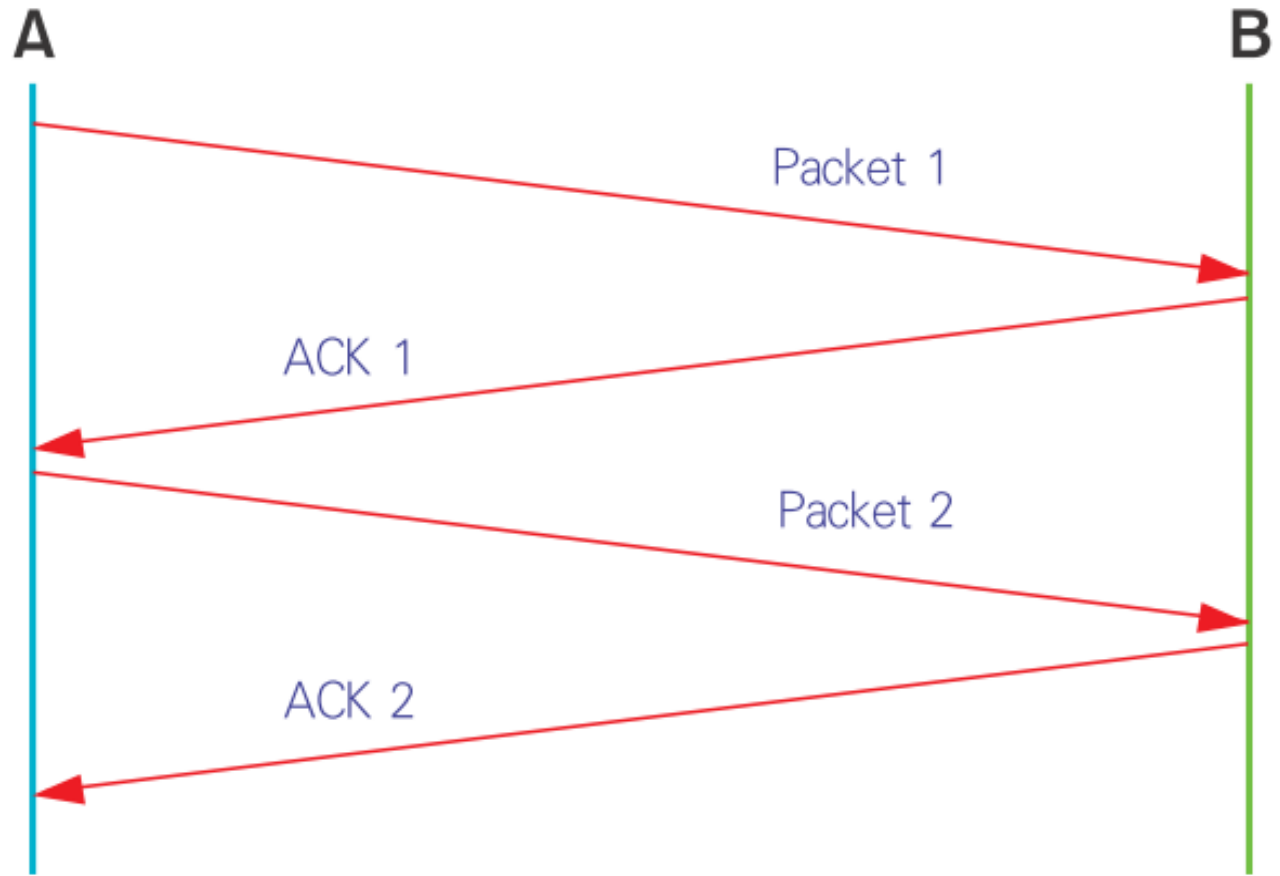


圖 11-3 利用確認機制可得知對方收到了封包

確認與重送

1. A 首先傳送 Packet 1 封包給 B
2. B 收到 Packet 1 封包後，傳送 ACK 1 封包給 A
3. A 收到 ACK 1 封包，接著即可傳送 Packet 2 封包給 B
4. B 收到 Packet 2 封包後，傳送 ACK 2 封包給 A

確認與重送

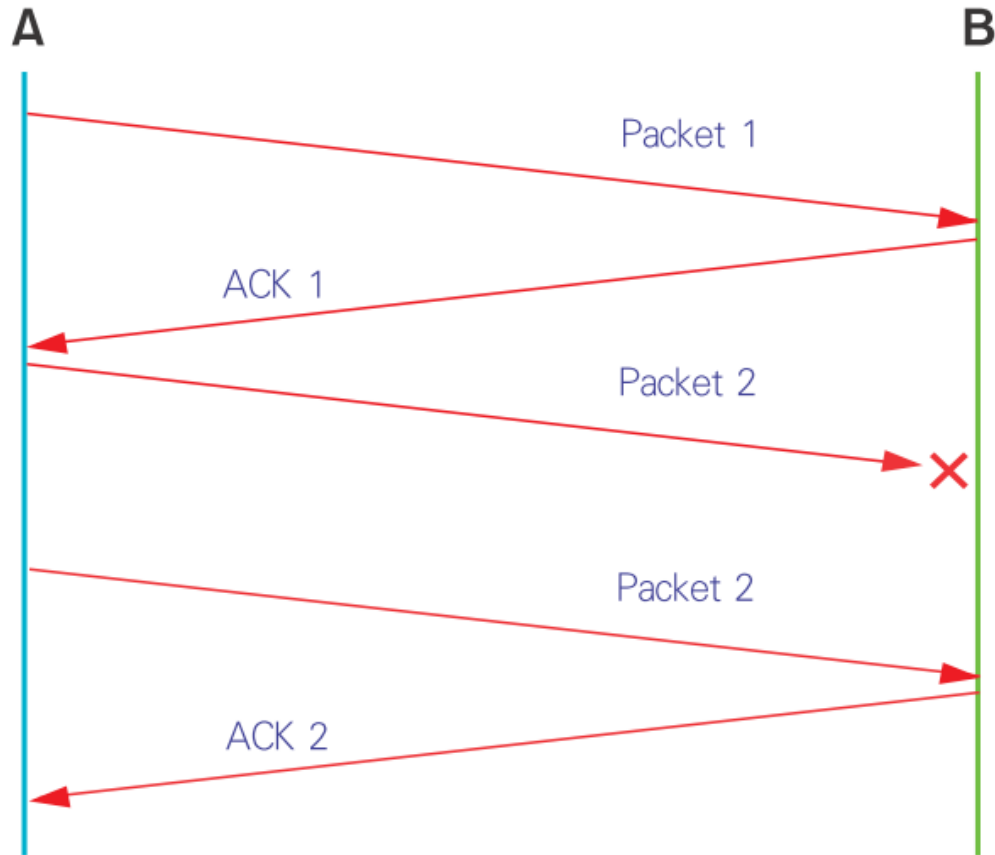


圖 11-4 利用重送機制來處理傳送過程中的錯誤

- 在 TCP 傳送過程中，即使發生錯誤，仍可藉由重送封包的方式來補救，維持資料的正確性與完整性。

11-3-2 雙向傳輸

- TCP 是一個雙向的協定,可想像成由兩條單向管道所構成的雙向傳輸

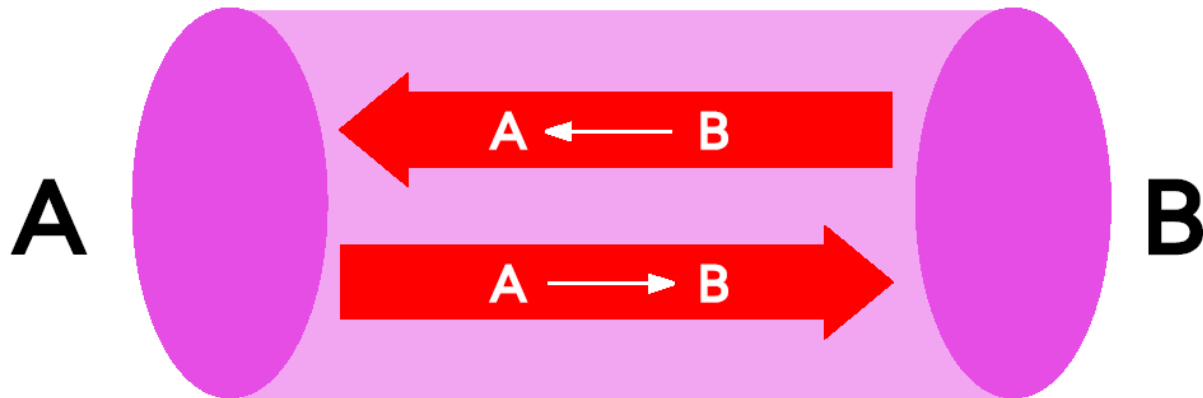


圖 11-18 TCP 連線是由兩條單向傳輸的管道結合而成

雙向傳輸

- A、B 之間互傳的封包,都扮演『傳送兼回應、回應兼傳送』雙重角色

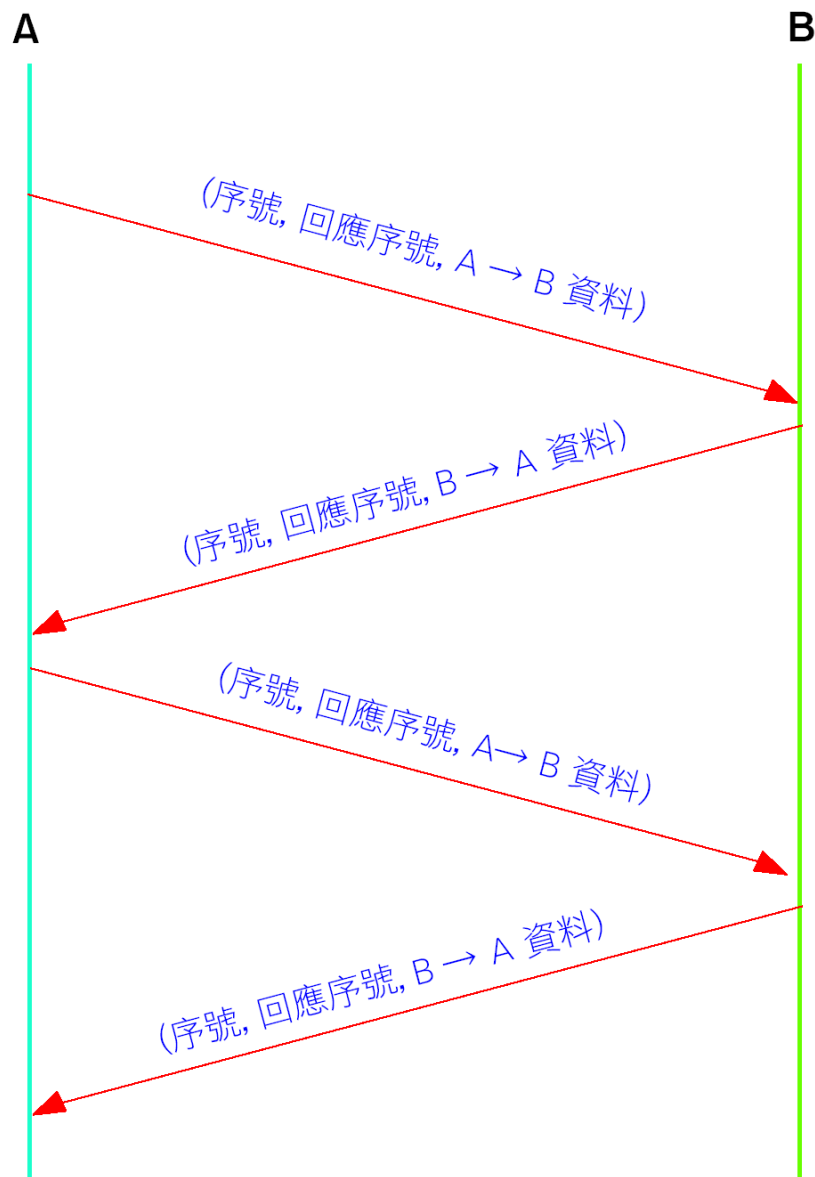


圖 11-19 每個 TCP 封包可能包含雙向傳輸的資訊

11-4 TCP 連線

- 11-4-1 識別連線
- 11-4-2 建立連線
- 11-4-3 中止連線

11-4-1 識別連線

- TCP 連線是由連線兩端的 IP 位址與連接埠編號所定義，伺服器可以和同一用戶端的不同連接埠建立多條連線，或是和不同的用戶端同時建立連線



圖 11-20 TCP 連線是由連線兩端的 IP 位址與連接埠編號所定義

識別連線

IP = 203.74.205.111
Port = 1800

A 1

IP = 203.74.205.112
Port = 1800

A 2

IP = 203.74.205.113
Port = 1900

A 3

IP = 203.74.205.113
Port = 1901

B

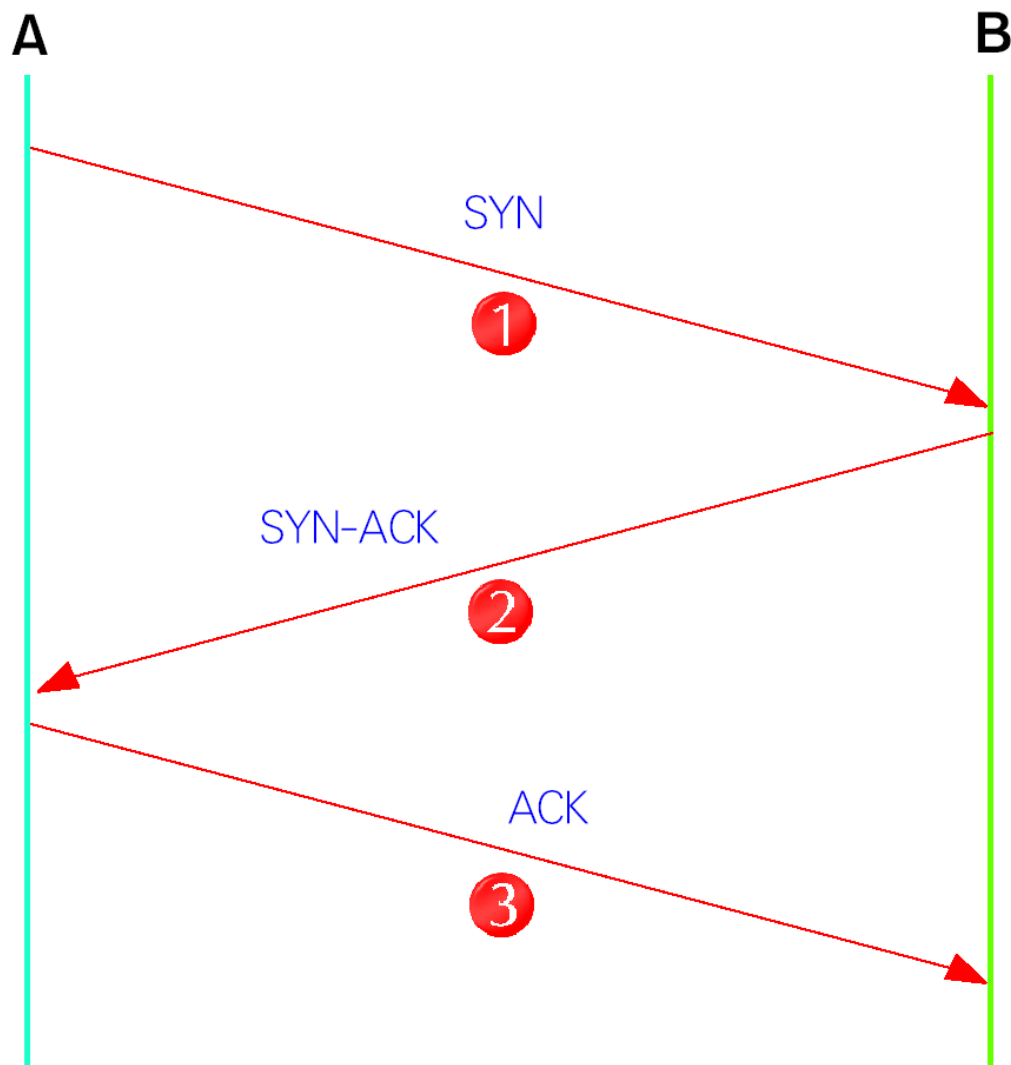
IP = 168.95.1.83
Port = 80

圖 11-21 伺服器可以和多個用戶端, 或同一用戶端的不同連接埠建立多條連線

11-4-2 建立連線

- 開始建立連線時，一定會有一方為主動端 (Active)，另一方為被動端 (Passive)。
- 在建立連線時，必須交換以下資訊：
 - 雙方的 ISN (初始序號)
 - 雙方的 Window Size

建立連線



- 1 序號 = ISN (A → B),
回應序號 = 0,
Window Size = Window (A ← B)
- 2 序號 = ISN (A ← B),
回應序號 = ISN (A → B) + 1,
Window Size = Window (A → B)
- 3 序號 = ISN (A → B) + 1,
回應序號 = ISN (A ← B) + 1,
Window Size = Window (A ← B)

圖 11-22 建立 TCP 連線的 3 個步驟

建立連線

- 第 1 步驟

- 序號

- 指定 $A \rightarrow B$ 的 ISN

- 回應序號

- 指定 $A \leftarrow B$ 的回應序號。

- SYN Flag

- 用來表示這是同步封包

- Window Size

- 代表A預設Receive Window的大小

建立連線

- 第 2 步驟

- 序號

代表 $A \leftarrow B$ 的 ISN

- 回應序號

SYN-ACK 封包的回應序號等於 ISN ($A \rightarrow B$) 再加上 1

- SYN-ACK Flag

表示此封包是回應先前收到的同步封包，所以它的序號就是 ISN ($A \leftarrow B$)。

- Window Size

代表 B 預設 Receive Window 的大小

建立連線

- 第 3 步驟

- 序號

也就是 SYN-ACK 封包的回應序號, 等於 $ISN(A \rightarrow B) + 1$ 。

- 回應序號

此處的序號等於 $ISN(A \leftarrow B)$ 再加上 1, 表示 A 期望 B 下次送來的資料, 是以 $ISN(A \leftarrow B) + 1$ 為第 1 Byte 的編號。

建立連線

- 第 3 步驟

- ACK Flag

表示回應序號包含了確認收到的資訊。

- Window Size

A 的 Receive Window 大小, 亦即 Window (A ← B)。

11-4-3 中止連線

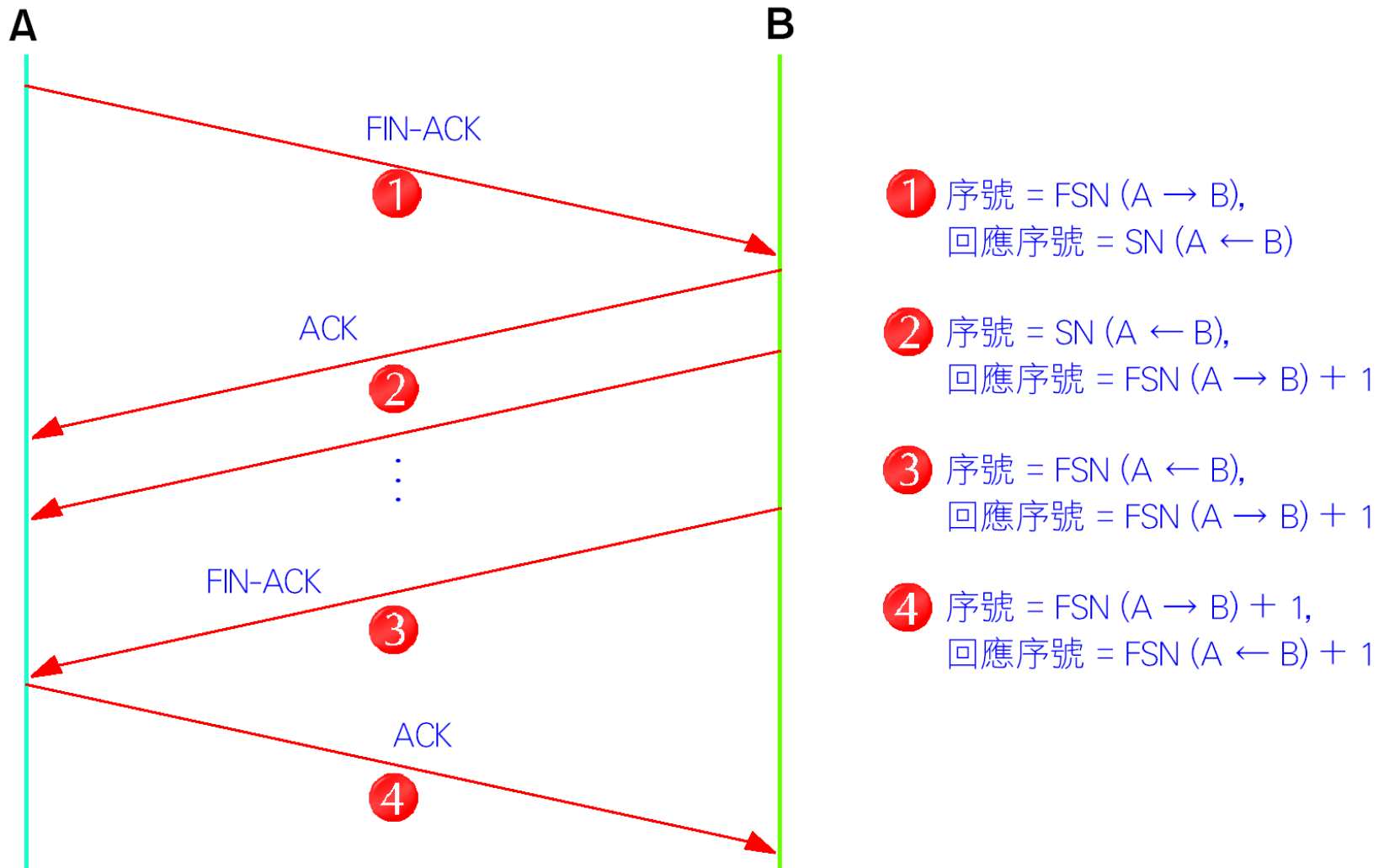


圖 11-23 結束 TCP 連線的 4 個步驟

11-4-3 中止連線

- 第 1 步驟
 - 序號
指定 $A \rightarrow B$ 的序號
 - 回應序號
指定 $A \leftarrow B$ 的回應序號。

11-4-3 中止連線

- FIN-ACK Flag

A → B 已經傳輸完畢, ACK 表示回應序號包含了確認收到的資訊。

- Window Size

與一般相同。

中止連線

- 第 2 步驟

- 序號

指定 $A \leftarrow B$ 的序號。

- 回應序號

此處回應序號等於第 1 步驟 FIN-ACK 封包的 FSN ($A \rightarrow B$) 再加上 1。

- ACK Flag

表示回應序號包含了確認收到的資訊。

中止連線

- 第 3 步驟

- 序號

指定 $A \leftarrow B$ 的序號

- 回應序號

指定 $A \rightarrow B$ 的回應序號。此處的回應序號與第 2 步驟的回應序號相同，皆為 $FSN(A \rightarrow B) + 1$ 。

- FIN-ACK Flag

代表 $A \leftarrow B$ 已經傳輸完畢

中止連線

- 第 4 步驟

- 序號

指定 $A \rightarrow B$ 的序號, 此處序號等於第 1 步驟 FIN-ACK 封包的 FSN ($A \rightarrow B$) 再加上 1。

- 回應序號

指定 $A \leftarrow B$ 的回應序號, 此處回應序號等於第 3 步驟 FIN-ACK 封包的 FSN ($A \leftarrow B$) 再加上 1。

- ACK Flag

表示回應序號包含了確認收到的資訊。