

資安與數學的連結

TCP Reno 擁塞控制行為之模擬與視覺化

報告人: 簡偉恆

2025/12/21

學習背景與目的

目標：

- 重現程榮祥 (2001) 論文中描述的 **TCP Reno** 行為
- 將論文的理論與示意圖轉為可操作的程式模擬
- 觀察 cwnd 、 ssthresh 與 loss 的動態變化

定位

- 以 Python 重建論文中的 TCP Reno 模型
- 視覺化呈現四大階段：SS、CA、FR、Timeout
- 驗證行為是否與理論一致

著重重現，不涉新演算法。

主要成果

以 Python 建立 TCP 傳輸流程與擁塞控制模擬器，重現 TCP Reno 四大核心行為：

- (1) **Slow Start** : $cwnd$ 指數成長。
- (2) **Congestion Avoidance** : 超過 $ssthresh$ 後線性成長。
- (3) **Fast Retransmit** : 三次重複 ACK 偵測封包遺失。
- (4) **Fast Recovery** : $cwnd$ 部分下降、快速恢復。

Reno 行為成功被可視化，有助教學及後續比較 CUBIC、BBR 等演算法。

Slow Start : 指數成長

初始設定： $cwnd = 1$ 。

每收到一個 ACK：

$$cwnd \leftarrow cwnd + 1$$

每個 RTT：

$$cwnd \approx 2^k$$

呈現指數級膨脹，直到 $cwnd = ssthresh$ 為止。

Congestion Avoidance : 線性成長

當 $cwnd \geq ssthresh$:

$$cwnd \leftarrow cwnd + \frac{1}{cwnd}$$

每個 RTT 約成長 +1 MSS。

呈現穩定、線性的成長。

Fast Retransmit 與 Fast Recovery

若收到 三次重複 ACK :

$$ssthresh = \frac{cwnd}{2}, \quad cwnd = ssthresh$$

若發生 timeout :

$$cwnd = 1, \quad ssthresh = \frac{ssthresh}{2}$$

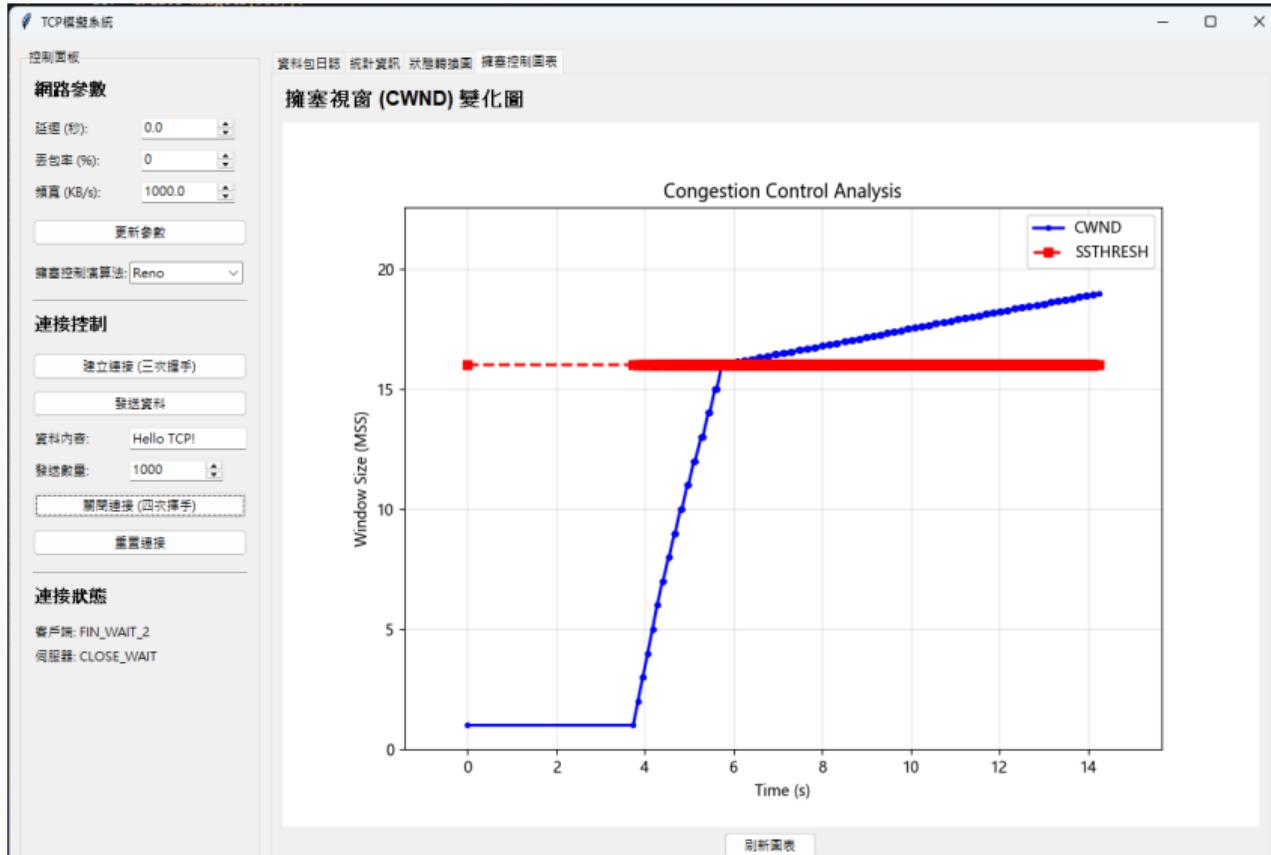
Timeout 的懲罰比 Fast Retransmit 更重。

實作 TCP 的核心流程：

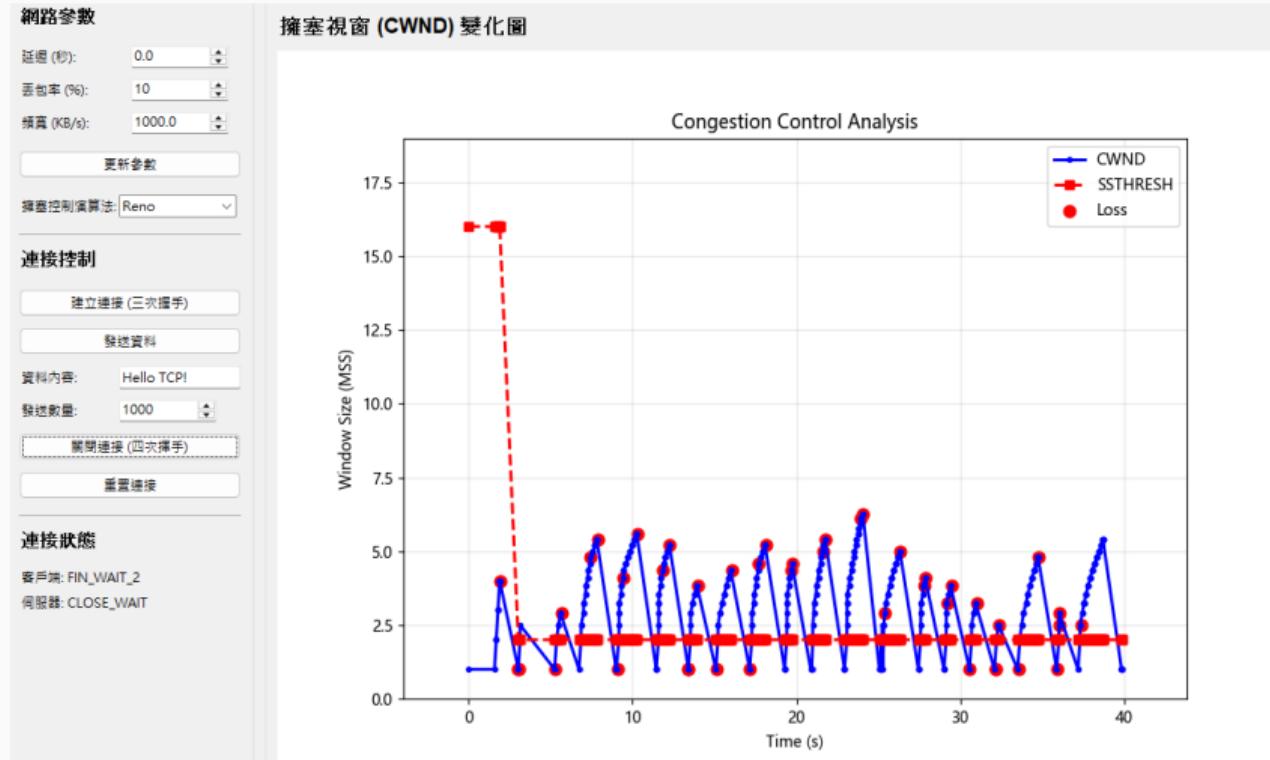
- 三次握手、封包傳輸、ACK 機制
- 可調整延遲 (latency)、丟包率 (loss)、頻寬 (bandwidth)
- 即時顯示：*cwnd*、*ssthresh*、封包事件

模組	功能描述
Connection	SEQ、ACK、封包收發
Simulator	時間軸事件、延遲、丟包
Reno Engine	Slow Start、CA、Fast Retransmit、Timeout
GUI	即時曲線視覺化

無丟包情況



有丟包情況 (10%)



數學觀點：離散動力系統

TCP Reno 可視為狀態轉換系統：

$$S \xrightarrow{f} S'$$

其中 f 為封包事件 (ACK / loss / timeout)。

其結構對應：

- $cwnd$ ：控制變數
- $ssthresh$ ：分段條件 (piecewise condition)
- duplicate ACK：局部 reset
- timeout：全域 reset

呈現與離散動力系統相近的遞迴行為。

- 多數教材以靜態圖示呈現 TCP，難以理解動態邏輯；
- 希望重建真實的 $cwnd$ 與 $ssthresh$ 時間演化；
- 探索「數學模型」與「網路行為」的結構相似性；
- 建立未來比較 CUBIC、BBR 等演算法的平台。

總結

- (1) 成功以模擬器重現 TCP Reno 的動態行為；
- (2) 可即時觀察 $cwnd$ 、 $ssthresh$ 、loss 時間點；
- (3) 模型與理論高度一致；
- (4) 可延伸至 CUBIC / BBR 等現代演算法研究。

References

- [1] Jacobson, V. (1988). *Congestion Avoidance and Control*. ACM SIGCOMM Computer Communication Review.
- [2] 程榮祥 (2001)。《TCP擁塞控制之效能改進》。國立成功大學資訊工程研究所碩士論文。
- [3] Jerry Banks、John S. Carson II、Barry L. Nelson、David M. Nicol 著，肖田元、范文慧譯 (2008)。《離散事件系統仿真 (原書第4版)'》。北京：機械工業出版社。
- [4] Postel, J. (1981). *RFC 793: Transmission Control Protocol*. Internet Engineering Task Force (IETF).