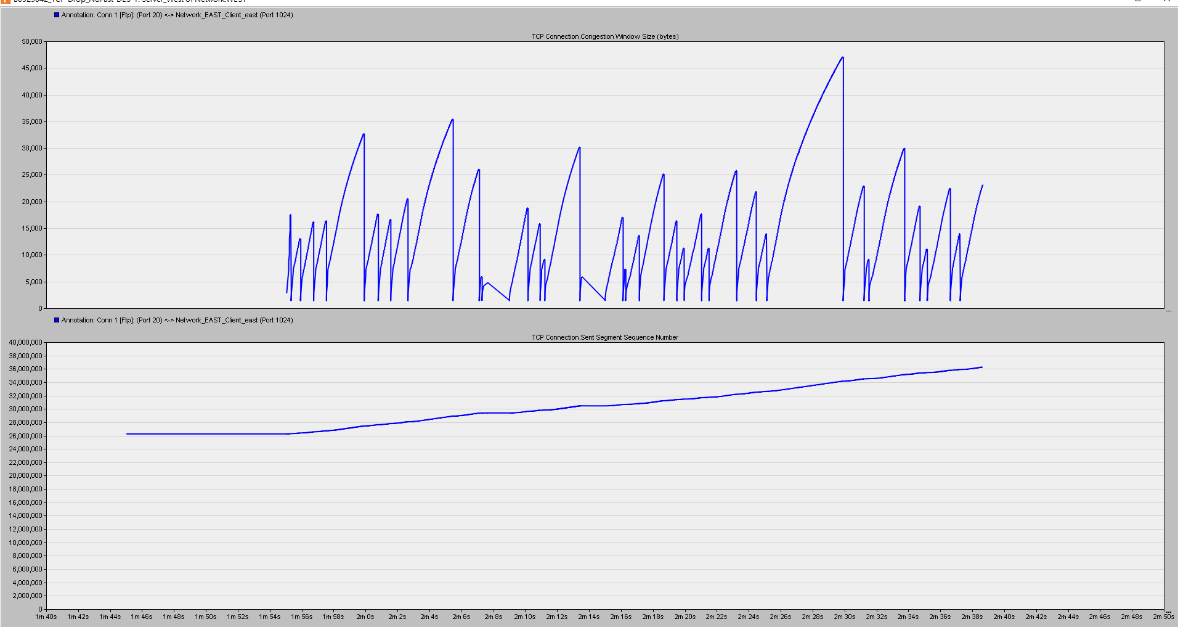
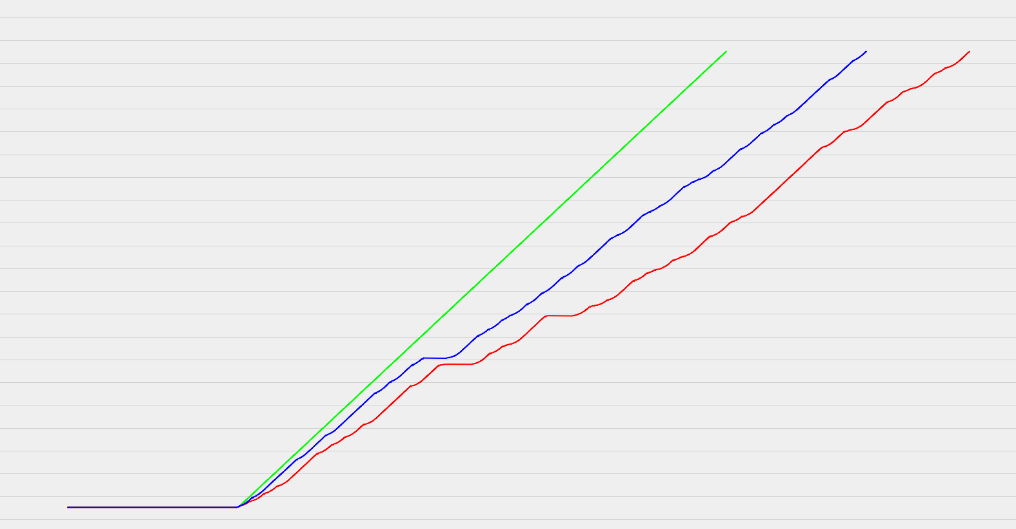
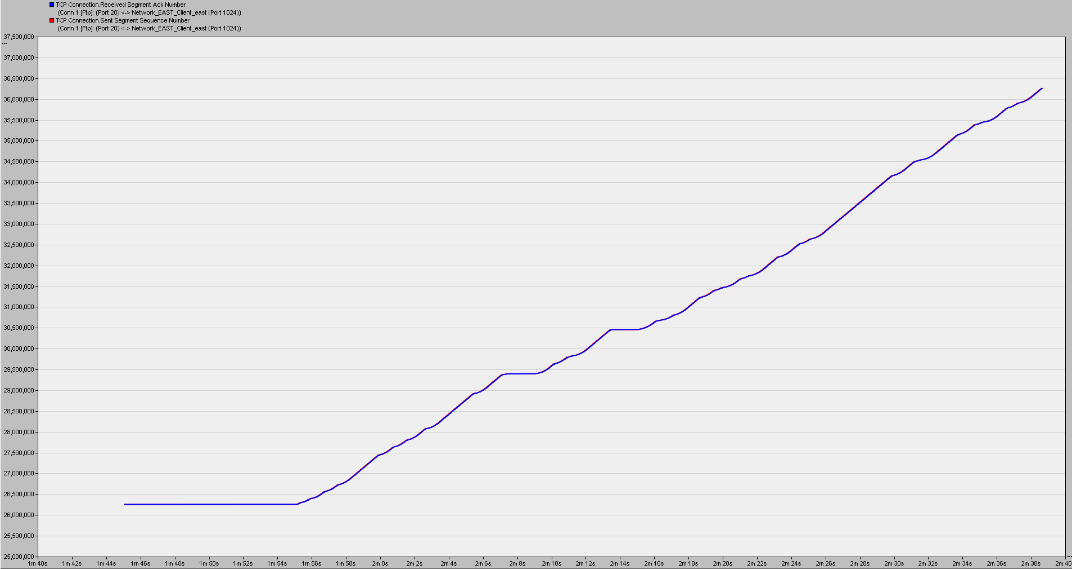
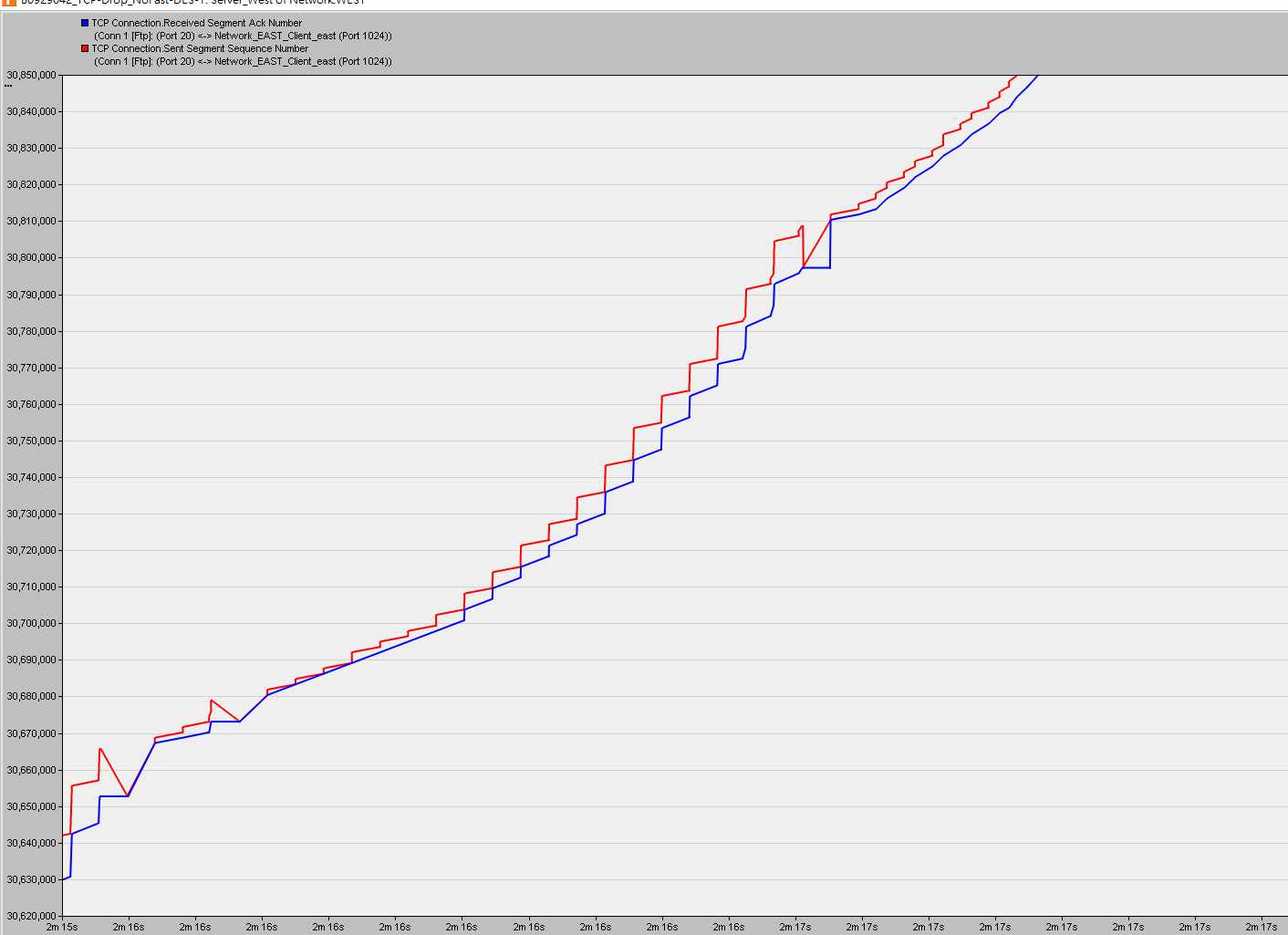
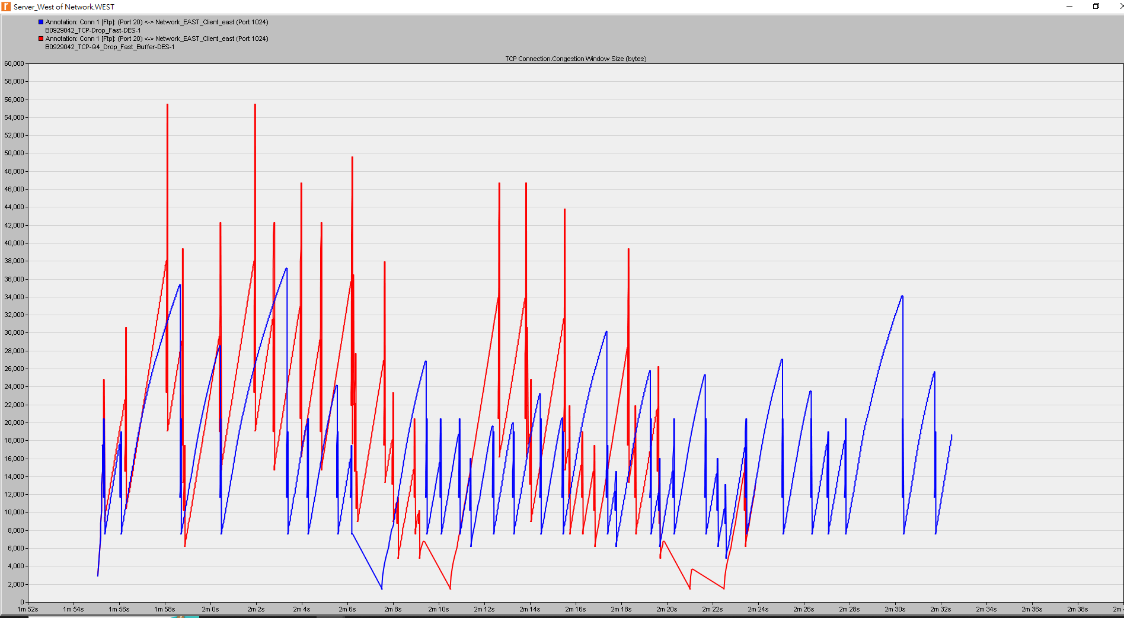
**實驗日期：**

May 11, 2023

**實驗名稱：**

1. Riverbed Modeler: TCP: Transmission Control Protocol

**問題與答案：**

1. Why does the **Segment Sequence Number** remain unchanged with every drop in the congestion window?  
     
   在Drop\_NoFast中是使TCP Tahoe演算法，在TCP Tahoe演算法中，如果丟失了封包接收方就會回傳上一個順利收到的封包號碼三次(3 duplicate ACKs)。因此在此圖中當CWND減少時代表有封包遺失，遺失就必須要重傳相同號碼的封包，因此封包號碼不會增長，而是保持不變。
2. Analyze the graph that compares the **Segment Sequence** numbers of the three scenarios. Why does the **Drop\_NoFast** scenario have the slowest growth in sequence numbers?  
   綠線表示N0\_Drop，在這個scenario中沒有封包被丟失，因此線非常平滑，也最快將所有封包傳送完畢。  
   藍線表示Drop\_Fast，這個scenario應用TCP Reno演算法，當封包遺失時會實施快速恢復機制，讓CWND不會降至1再爬升，因此在遺失封包之後仍可以用較快的速度繼續傳資料，與TCP Tohoe相比能較快將所有資料傳完。  
   紅線為Drop\_NoFast。
3. In the **Drop\_NoFast** scenario, obtain the overload graph that compares **Sent Segment Sequence Number** with **Received Segment ACK Number** for **Server\_West** explain the graph.  
     
   下圖為上圖的部分放大。  
     
   紅線為送出的封包號碼，藍線代表收到的封包號碼。  
   上圖可見紅線稍高於藍線，因為必須先送出封包而後才收到該封包的ACK，除此之外當有封包遺失須重送時，紅線會先下降之後再爬升，代表重送之前送過的封包號碼，如黃框所示。  
   而重送之後的直線看起來較平滑，如綠框所示，因為CWND降至1的緣故。
4. Create another scenario as a duplicate of the **Drop\_Fast** scenario. Name the new scenario **Q4\_Drop\_Fast\_Buffer**. In the new scenario, edit the attributes of the **Client\_East** node and assign 65535 to its **Receiver Buffer (bytes)** attribute (one of the **TCP Parameters**). Generate a graph that shows how the **Congestion Window Size (bytes**) of **Server\_West** gets affected by the increase in the receiver buffer (compare the congestion window size graph from the Drop\_Fast scenario with the corresponding graph from the **Q4\_Drop\_Fast\_Buffer scenario**.)  
     
   由上圖可見，紅線為Q4\_Drop\_Fast\_Buffer，接收端的buffer為最大大小，因此由於buffer過小導致的封包遺失機率減少，CWND就有較大機會增長到非常大才導致倍減，所以傳輸速度快，比藍線還快將所有封包傳完。

**討論：**我在操作Riverbed Modeler玩TCP實驗時，真的對上學期學的TCP傳輸協議印象深刻啊！這次實驗讓我更加理解改進TCP的重要性，也親眼見證了它的好處。透過這次實驗，我更深入地瞭解了TCP的運作方式，我觀察了封包傳輸的過程，搞懂了TCP如何利用序列號、確認號和窗口大小來保證封包傳輸的可靠性。真的讓我更明白TCP在網際網路通信中的關鍵作用。實驗讓我意識到改進TCP可以大幅提升網絡性能和吞吐量，通過這些改進，我發現網絡的延遲減少了，封包傳輸速度也提高了。這進一步證實了改進TCP的重要性，尤其是在高負載的網絡環境中。

**補充資料：無**