1. **實驗日期:** 2017/5/18

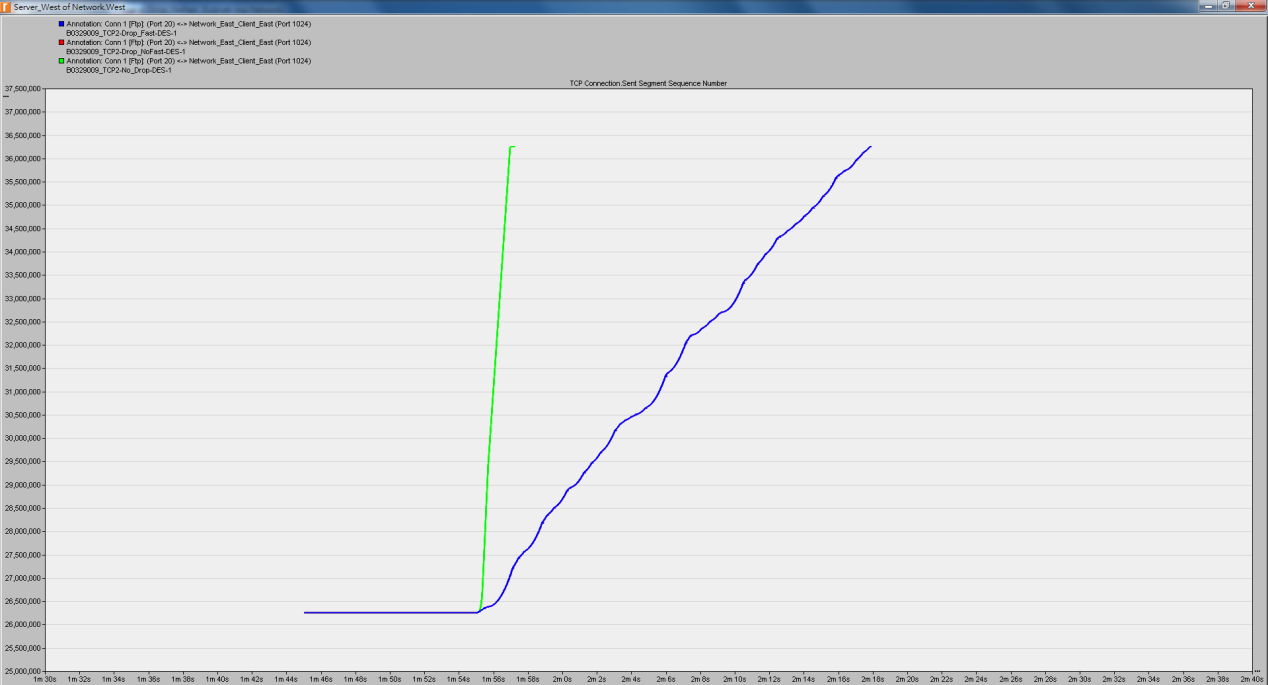
**2.**  **實驗名稱:**

Riverbed Modeler: TCP: Transmission Control Protocol

**3.  問題與答案:**

實驗一

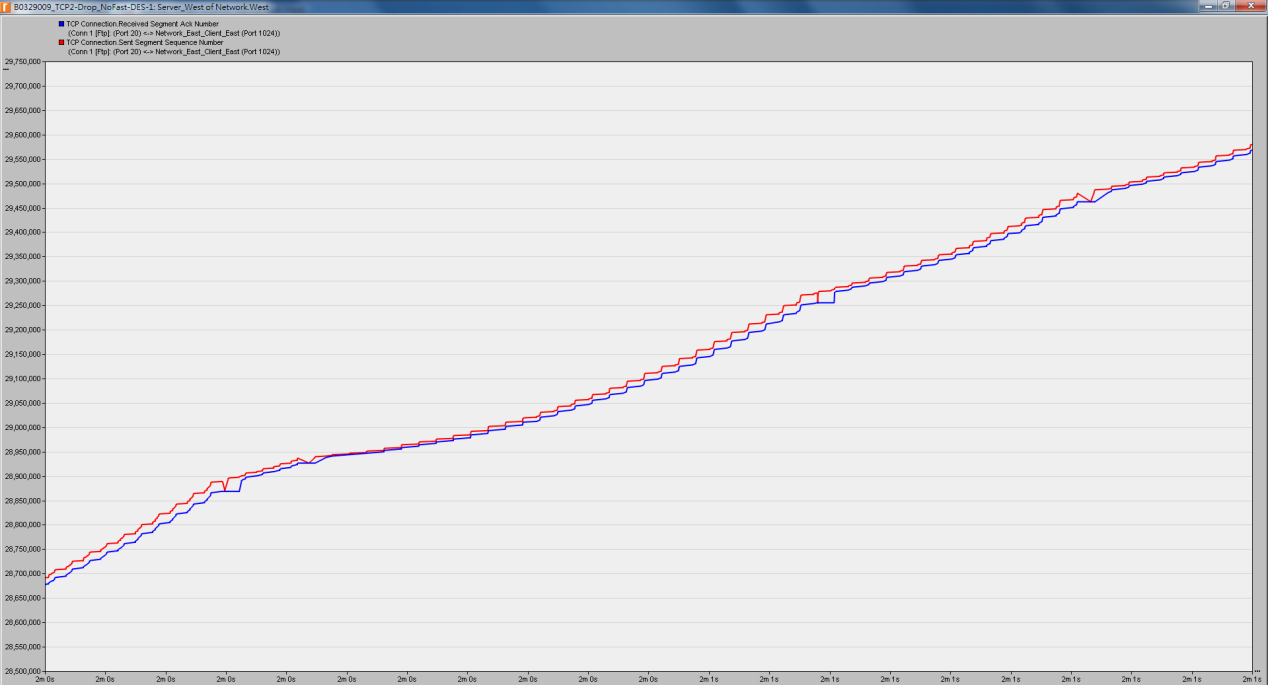
**1)** Why does the **Segment Sequence Number** remain unchanged (indicated by a horizontal line in the graphs) with every drop in the congestion window?



**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?

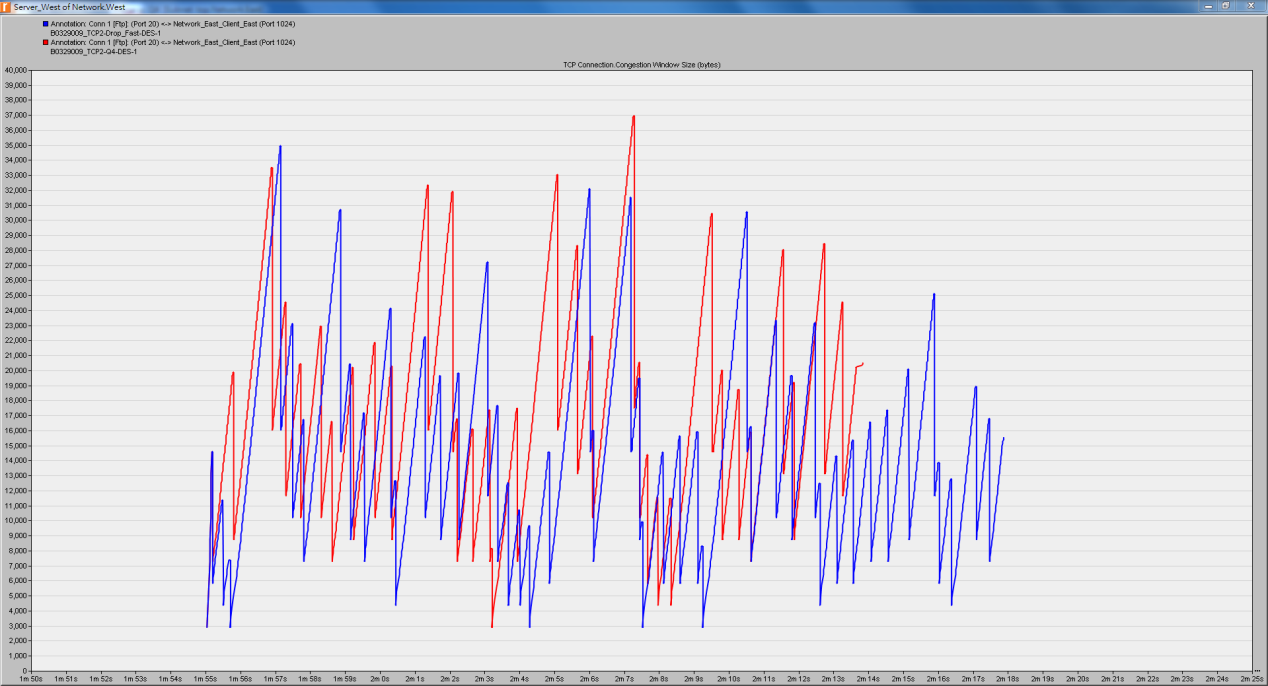
Drop\_NoFast會把5%的封包丟棄，並且在TCP的中retrainsmit option設為disable，代表每當timeout，TCP timeout期間的全部長度會在封包重新發送之前通過。

**3)** In the **Drop\_NoFast** scenario, obtain the overlaid graph that compares **Sent Segment Sequence Number** with **Received Segment ACK Number** for **Server\_West**. Explain the graph.



紅線部分(Sent Segment Sequence Number)高於藍線部分(Received Segment ACK Number)，表示會等到有接收到封包後再傳遞下一個封包，若沒有接收到就會重傳封包(紅線有下降的部分)。

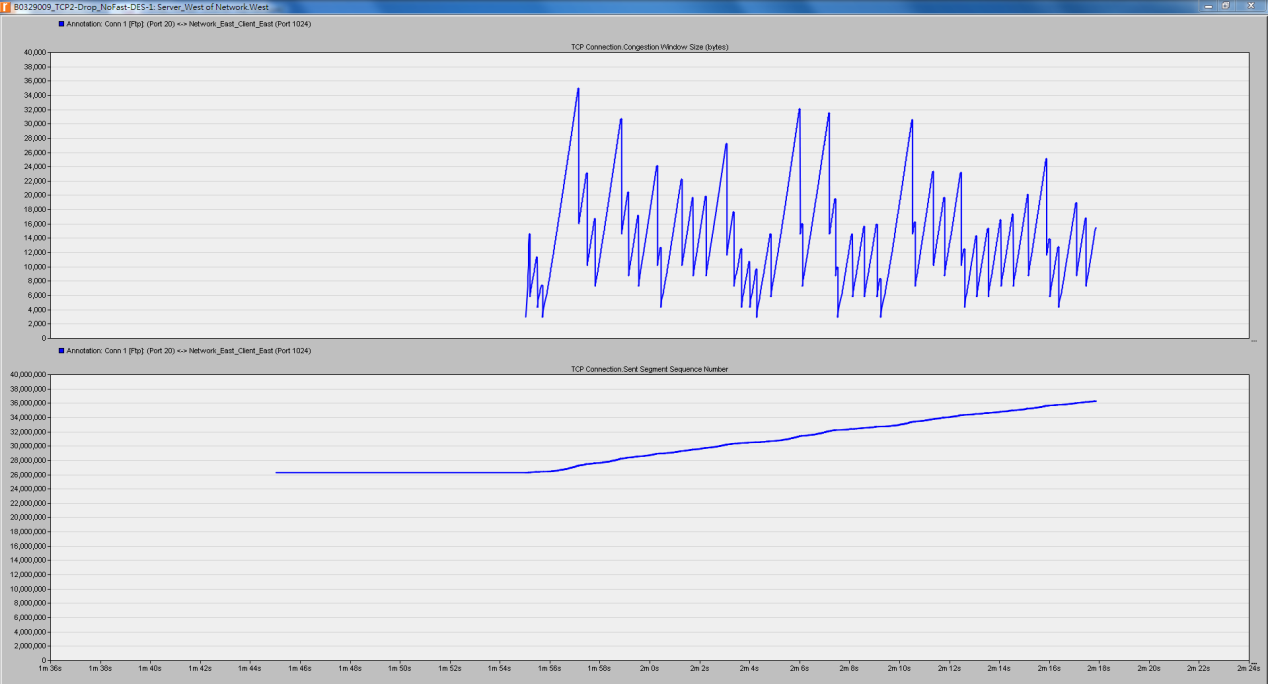
**4)** Create another scenario as a duplicate of the **Drop\_Fast** scenario. Name the new scenario **Q4\_Drop\_Fast\_Buffer**. In the new scenario, edit the attributes of the **Client\_East** node and assign 65535 to its **Receiver Buffer (bytes)** attribute (one of the **TCP Parameters**). Generate 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.)



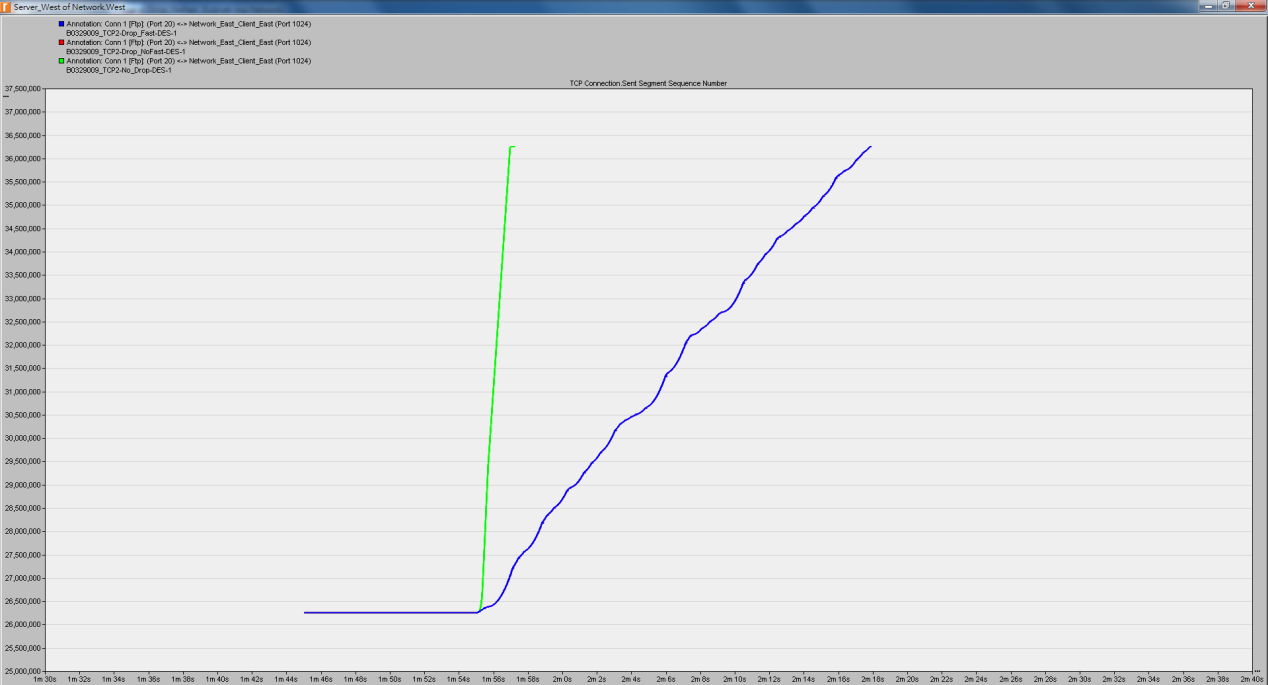
**4.  結果討論:**

實驗一

Congestion Window Size (bytes) and Sent Segment Sequence Number



Sent Segment Sequence Number(No\_Drop綠、Drop\_NoFast紅、Drop\_Fast藍)



這次實驗也是砍掉重練ing哈哈，不過我做第二次就成功了，可能是其中一個地方忽略掉了，才會導致錯誤。整體來說算OK，了解封包傳遞和接收的過程，Window Size及Receiver Buffer (bytes)的大小都會影響到封包傳遞的快慢。