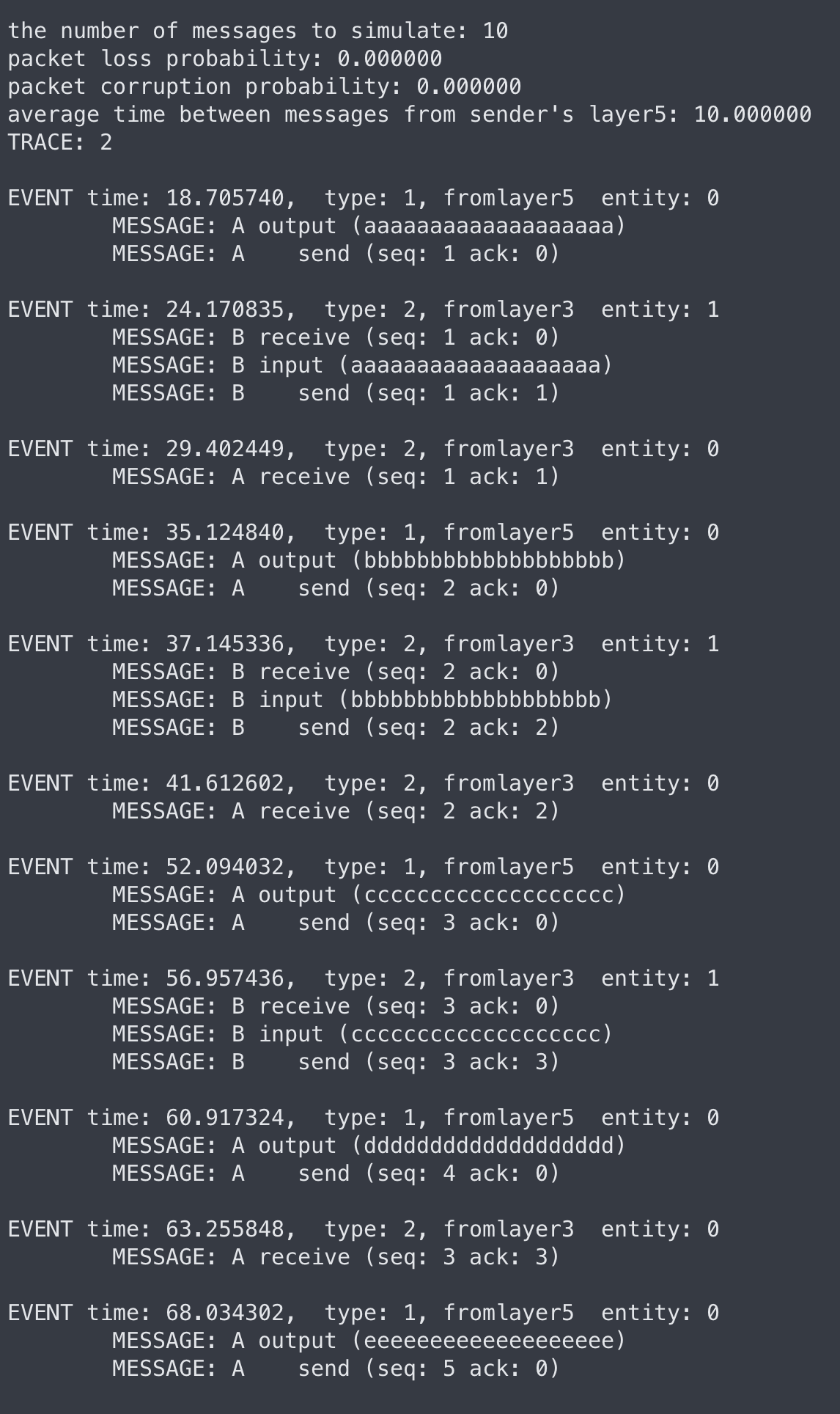
電腦網路導論 hw2 電機三 b08901048 陳宥辰

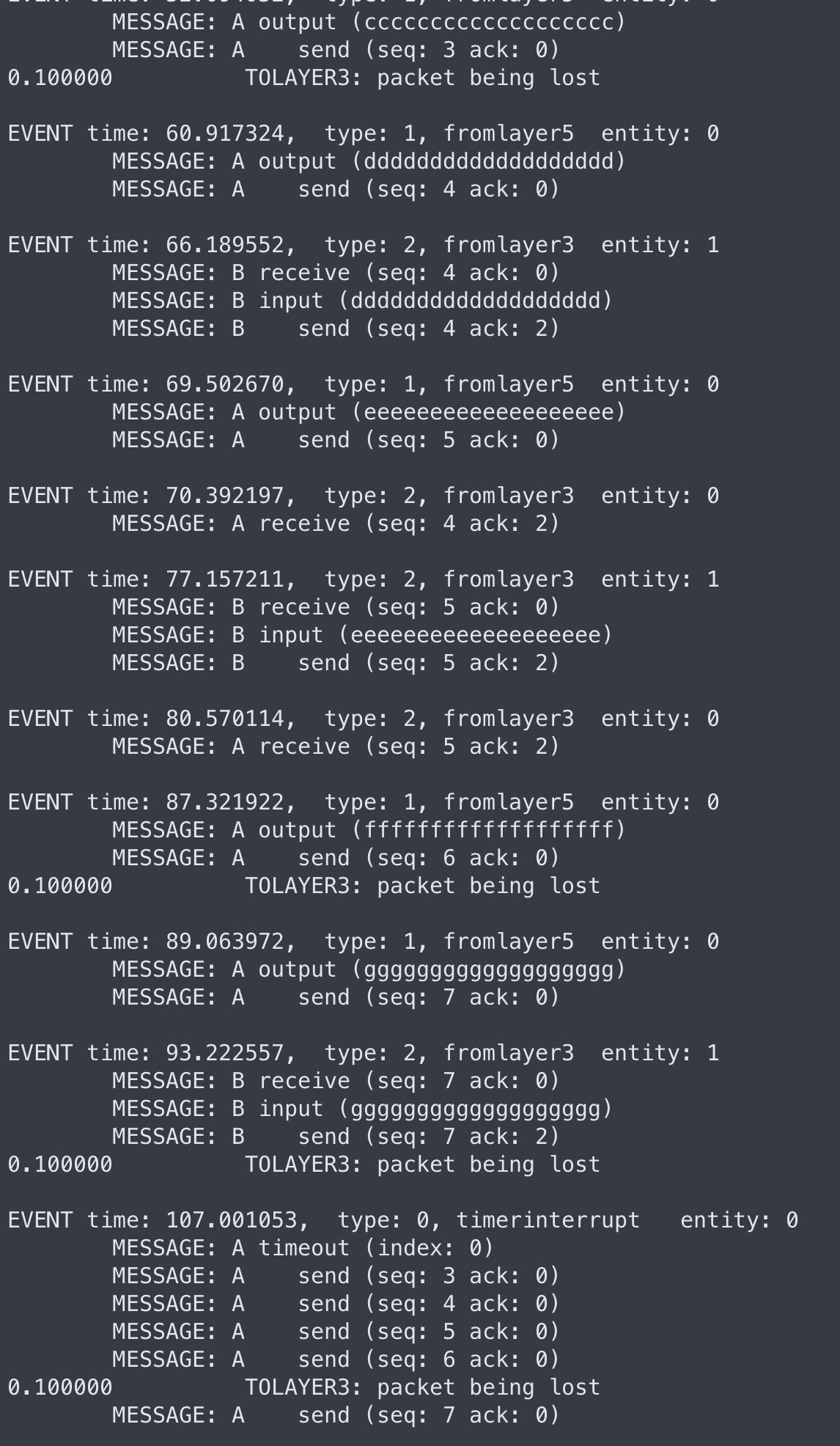
./b08901048 10 0 0 10 2

此為最正常，沒有loss pkt或corrupted的case，B收到應當收到的seqnum之後會回傳該seqnum作為acknum的pkt給A，而A會把A.base更新到pkt.acknum+1。



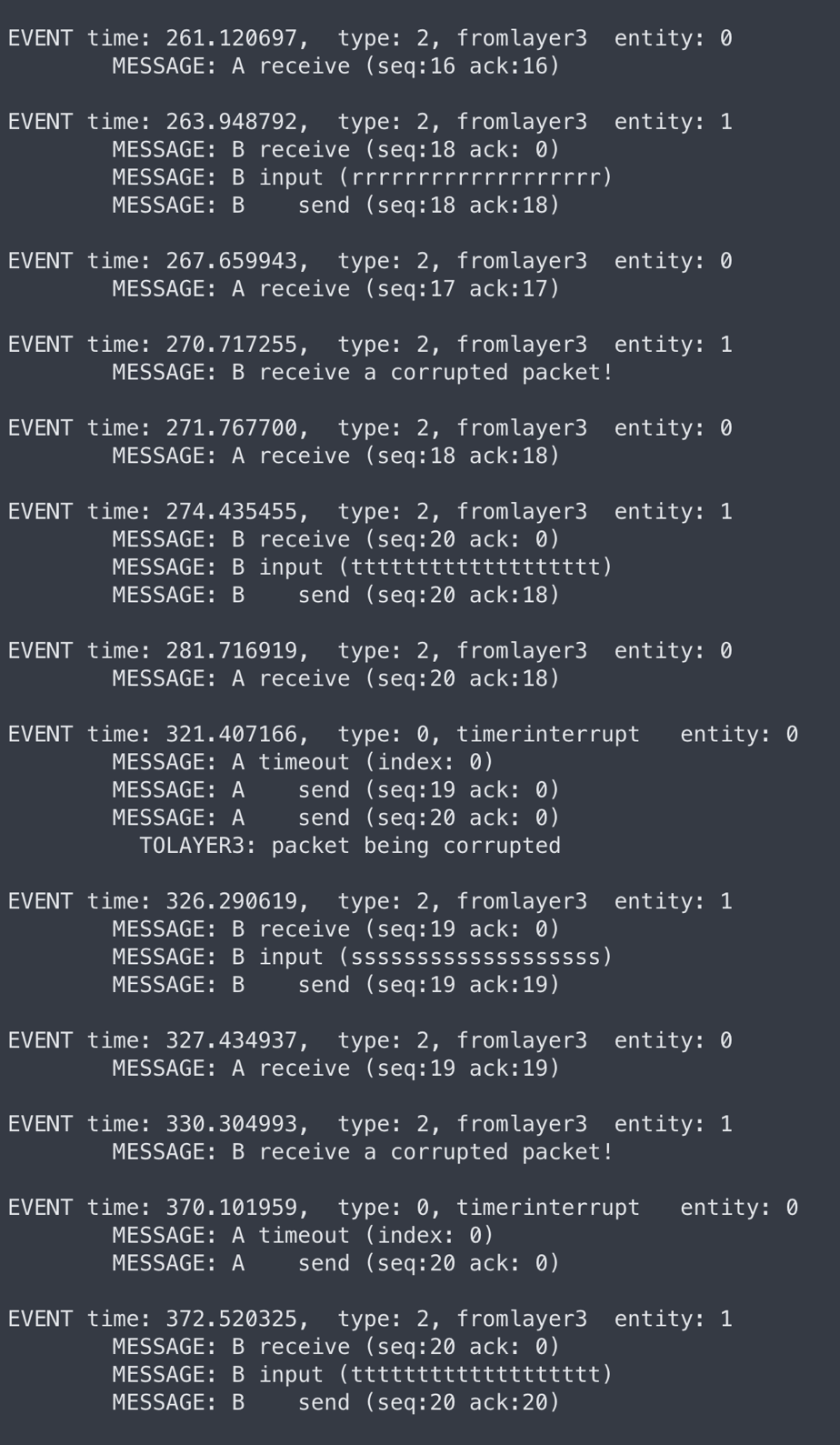
./b08901048 20 0.1 0 10 2

此為可能出現loss的case，一旦loss在A send pkt的過程中發生，B就不會收到順序正確的pkt，因此會回傳和上一個pkt一樣的ack2的pkt給A，A會重複收到ack2，直到timeout，timeout發生時，A會send buffer中所有的pkt，從pkt3到pkt7，然後B會收到pkt3並回傳ack3，A的base會被更新到4。



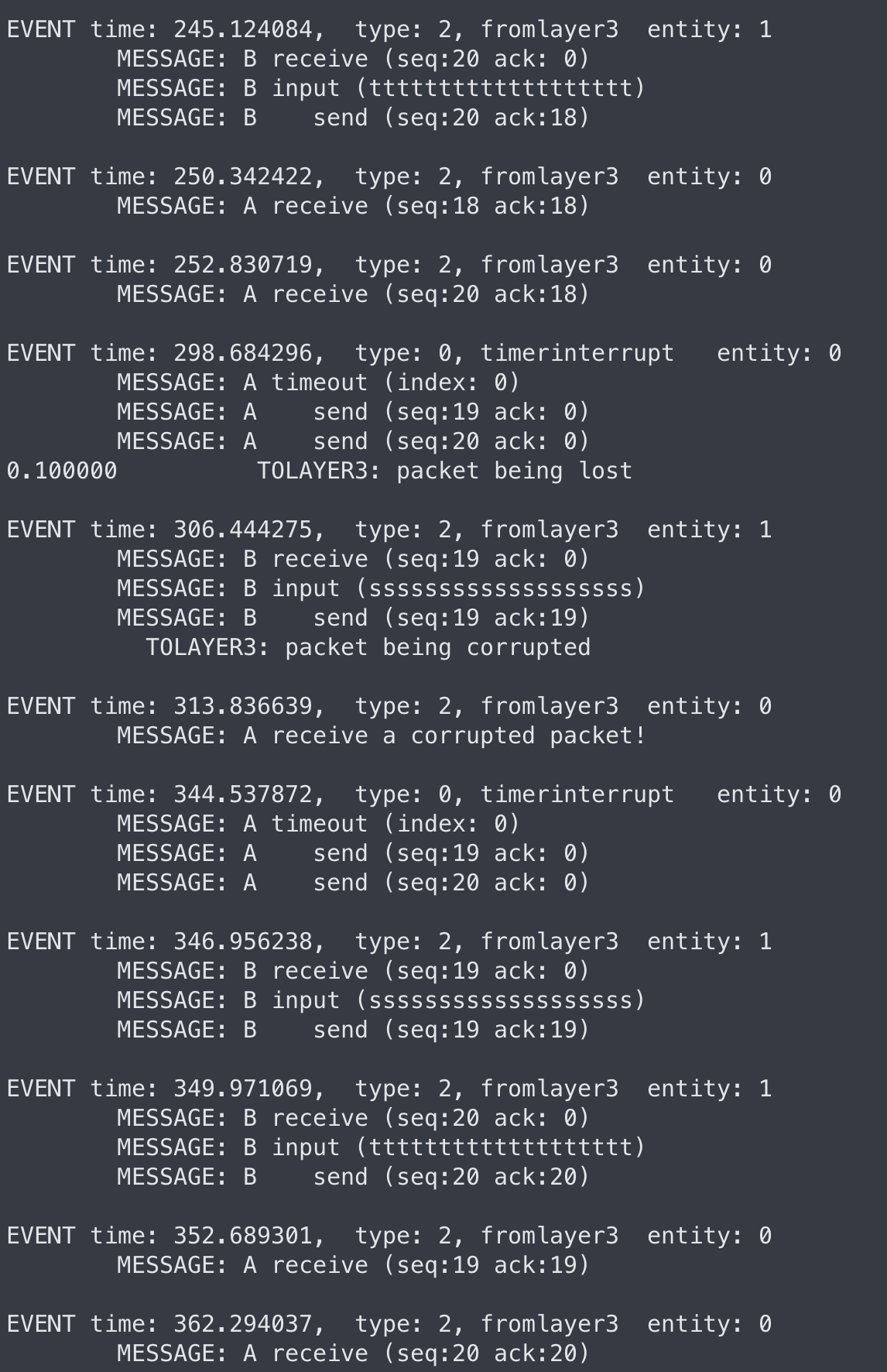
./b08901048 20 0 0.1 10 2

B在收seq19的時候收到corrupted pkt，所以B會回傳ack18給A，等到timeout時，A重傳seq19,seq20，B收到seq19回傳ack19，但seq20又corrupted，所以再等到timeout時，A回傳seq20給B，B收到seq20後，回傳ack20。



./b08901048 20 0.1 0.1 10 2

A send pkt20會lost，然後B回傳ack19的時候會corrupted，因此A沒收到ack19，等到timeout後A重傳seq19,seq20，B收到seq19回傳ack19，B收到seq20回傳ack20，A收到ack19,ack20。



2.我的timeout interval採用2\*EstimatedRTT，以下的圖橘色的是SampleRTT，藍色的則是EstimatedRTT，我在執行程式時有發現一件事，在我的電腦上執行時，time有時候會變成亂數（或是很長的數）而我完全沒有改動到time的值，我在同學的電腦可以正常使用，因此我在實作這部分時，SampleRTT有時會因為time值變成亂數而爆掉，所以我這邊做了一個判斷是如果SampleRTT過大的時候會讓他回到1.6倍的EstimatedRTT，太小的話則是變成0.8倍的EstimateRTT，我知道這樣的做法和課本上或課堂上有些出入，但我目前能做的處置是這樣，然後調整好的SampleRTT再利用EWMA的公式生成EstimatedRTT。因為這個公式的用意應該是讓time interval可以隨著EstimatedRTT進行彈性修正，我的作法應該也可以達到類似的效果，還請助教斟酌。