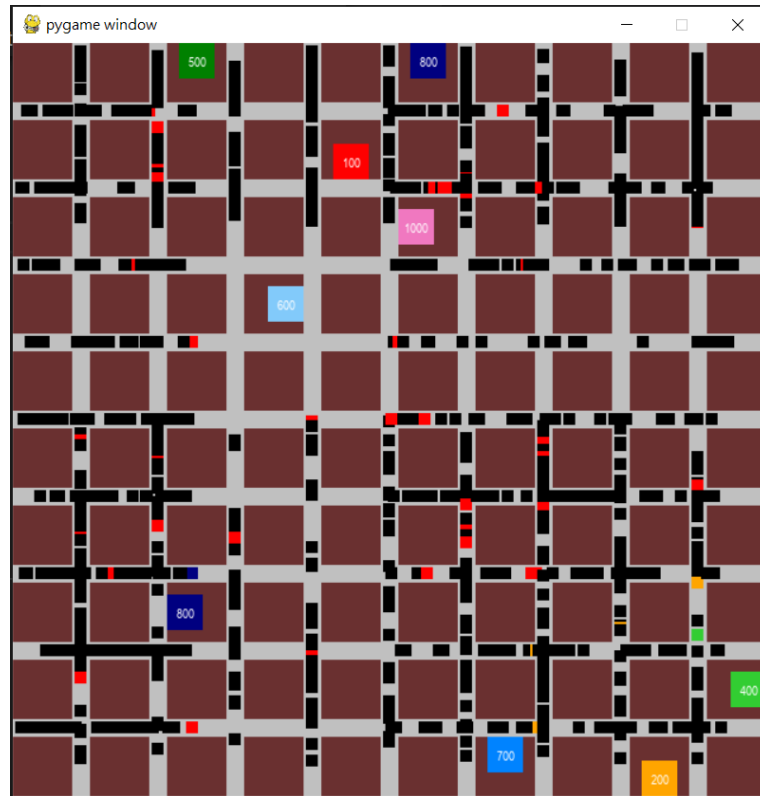


無線通訊網路程式模擬

使用 python pygame :



參數設定:

1.FPS = 100，模擬 1min。(FPS 設置 1/3 為真實情況(模擬起來太慢)，

因此加速 300 倍→FPS=1/3*300=100)

2.Lambda=1/4。(原本題目設定為 1/12，若 FPS=1/3 時(真實情況)，

表示每 3 秒有 1/4 的 arrival rate)

3.Speed=1.2。(原本題目設定為 0.02km/sec，但是我的視窗大小為

50*50，題目設定為 2.5km*2.5km，所以比例為 20 倍，因此

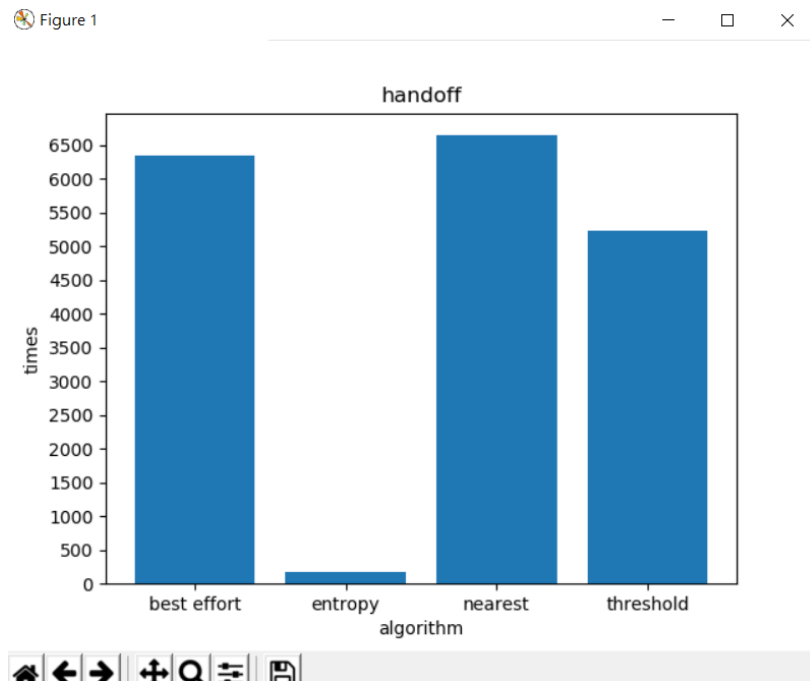
0.02*20=0.4，但是因為 FPS=1/3(真實情況)，所以速度要再乘 3 →

0.4*3=1.2)

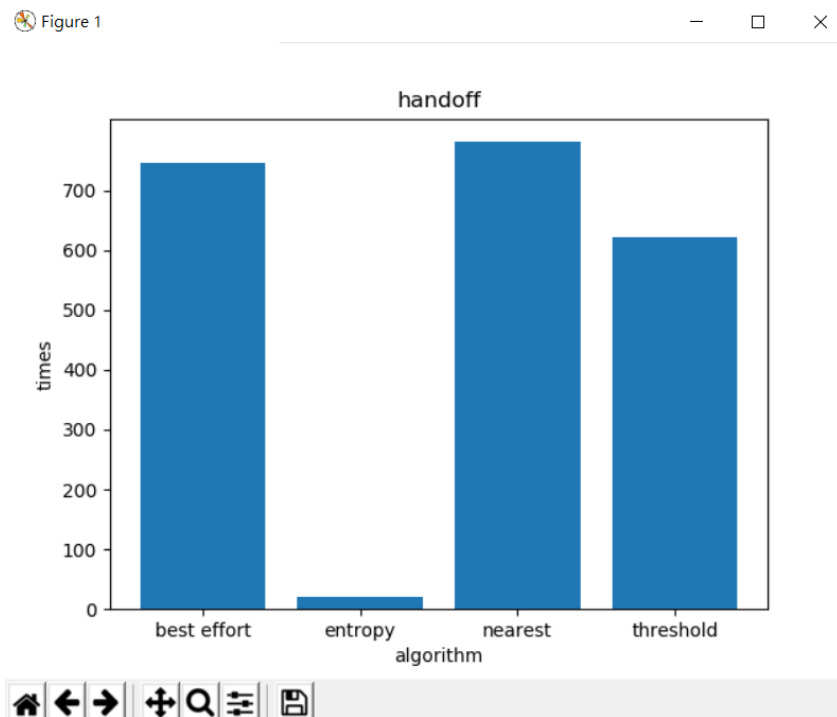
Threshold、entropy、Ptransmit 與題目設定相同。

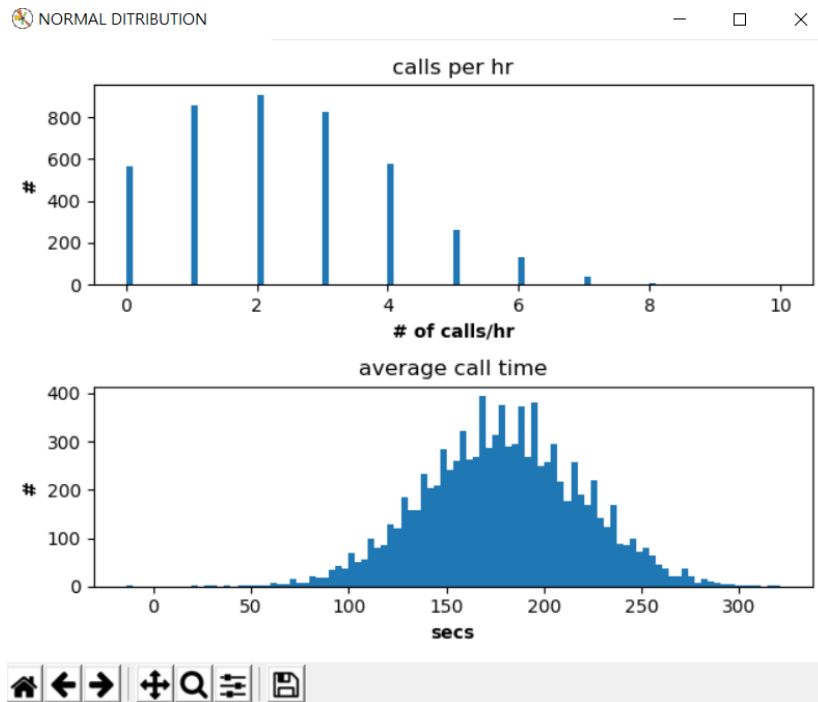
4 種 algorithm 分析:

Q1:



Q2:





1. Minium(Threshold):強度夠的時候不切換($P_{min}=100\text{dB}$)
2. Best effort:訊號強者提供訊號
3. Entropy:兩基地台訊號落差值超過閾值時候切換(閾值 $E=25\text{dB}$)
4. Nearest:選最近的 base station

解析：

兩題中做最少交換次數的 algorithm 都是 entropy，因為他的交換條件是兩基地台訊號落差超過 25db(蠻大的)才會交換，條件最嚴苛；而 threshold 為交換次數第 2 少的原因是因為只要訊號強度夠就不切換(符合條件才切換)，所以會比 best effort 交換次數少(比 best effort 保守派)；Best effort 是訊號強的基地台提供訊號，所以只要車子發現訊號比原本基地台提供的訊號強就會切換，所以會常常 handoff；

而我自己設定的 **nearest** 方法是類似 **best effort**，不過條件變成距離不是訊號：永遠選最近的基地台，不管訊號強度如何，所以只要車子開到哪裡，就會連最近的基地台，最常切換。我當初會設計連最近的基地台是因為我以為這是事實(哈哈，原來不是阿)，雖然交換次數最多，但是因為老師說過電信業者在架設基地台的時後統計過近距離的人會比遠距離的人更容易通話(譬如說都在成大校園裡的兩個人打電話機率 > 台南東區打電話給北區)，所以如果都連距離最近的，兩電話間的資料傳輸就不用轉換基地台，會省一些轉換的損耗，因此我認為也是不錯的 **algorithm**。

3 個 algo 的好壞:

	Best effort	Threshold	Entropy
Switch cause overhead	worse	Medium	better
訊號品質	best	Medium?(需要看臨界值是多少)	Worse?(需要看臨界值是多少)