NSCAP HW3 Report

109550073 陳宥安

1. Code Test

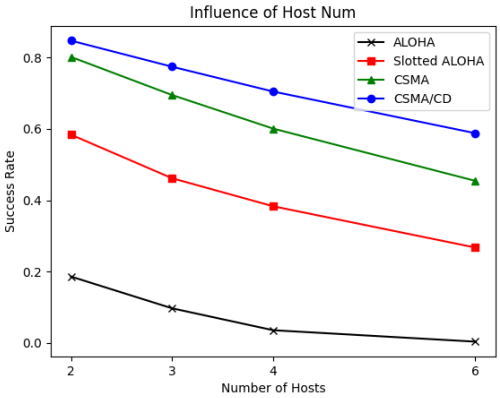
Setting(host_num=3, total_time=100, packet_num=4, max_colision_wait_time=20, p_resend=0.3, packet_size=3, link_delay=1, seed=109550073)

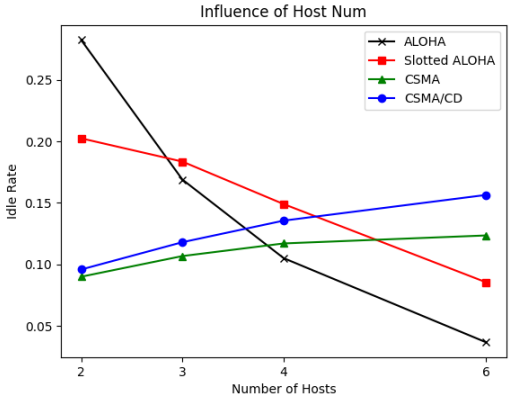
```
~/Desktop/nscap/hw3
❯ python3 main.py
aloha
                                                    V          V       V           V
h0: ...................................................<---|...................<---|.....<---|.......<
                      V       V              V    V
h1: ...................<---|......<---|....<---|.............<---|...................<---|.............
                      V           VV  V
h2: .................<---|......<---|......<---|...................<--->...<--->...<---|...........<---|.....
success rate: 0.1
idle rate: 0.42
collision rate: 0.48

slotted aloha
                                                    V          V       V           V
h0: ...................................................<---|.............................<--->...<---|
                      V       V              V    V
h1: ...................<---|.....<---|..............<--->...<---|..........<---|...................<---|
                      V           VV  V
h2: ...................<---|.....<---|.........................<--->...<--->...<---|.....<--->...<--->..........
success rate: 0.3
idle rate: 0.45
collision rate: 0.25
```

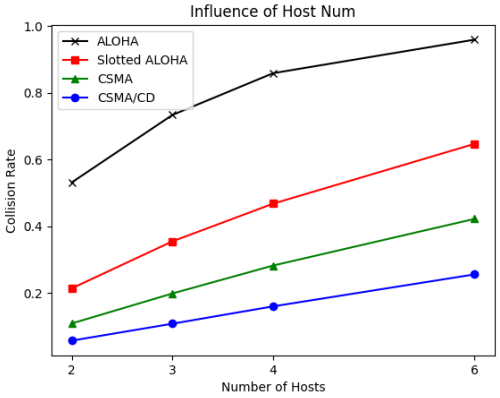
```
csma
                                                    V          V       V           V
h0: ...................................................<--->.......<---|.............................
                      V       V              V    V
h1: ...................<---|......<---|....<---|.................................<--->...<--->...<---|.......<
                      V           VV  V
h2: .................<---|......<---|......<---|...................<---|...................<---|......<
success rate: 0.15
idle rate: 0.54
collision rate: 0.31

csma cd
                                                    V          V       V           V
h0: ...................................................................<--->...<--->...<--->...<--->.......
                      V       V              V    V
h1: ...................<|......<--|....<--|.....................................................<---
                      V           VV  V
h2: .................<--|......<|......<|.............<--->...<--->...<--->...<--->.............................
success rate: 0.4
idle rate: 0.44
collision rate: 0.16
```

2. Questions

2.1. Setting(host_num=h, packet_num=p, max_colision_wait_time=20, p_resend=0.3) for h,p in zip(host_num_list, packet_num_list)







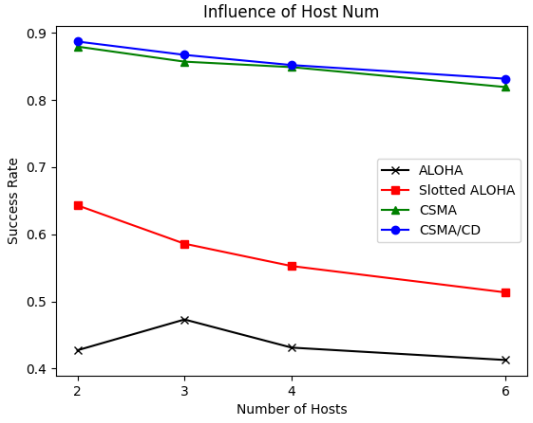
2.2. max_colision_wait_time = ? * c (Hint: Two parameters)
p_resend = ? / c (Hint: One parameter)

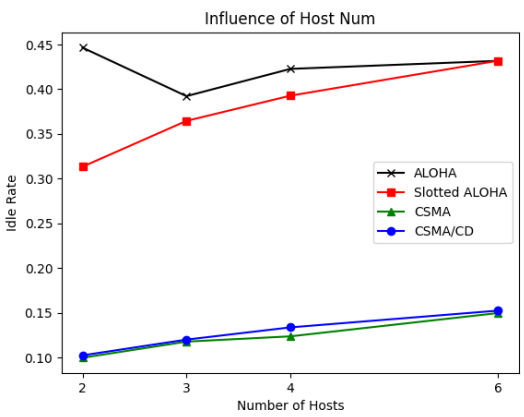
```python
if max_colision_wait_time is None:
    self.max_colision_wait_time = host_num * self.packet_time * coefficient# your answer
else:
    self.max_colision_wait_time = max_colision_wait_time
if p_resend is None:
    self.p_resend = (1 / (host_num * coefficient))# your answer
else:
    self.p_resend = p_resend
```

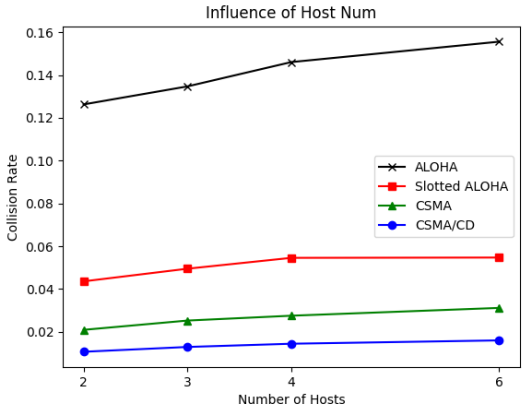
max_colision_wait_time = host_num * self.packet_time * coefficient
第一個參數會選擇host_num主要是因為Collision跟host的數量一定有關西，因為host越多的情況下，等於越多人在固定大小的頻寬下試著送packet，越容易碰撞，所以collision time要長一點才會減低碰撞機率。第二個參數是packet_time，因為packet_time越大的話代表一個packet要完整傳送的時間就會越長，等於在固定大小的頻寬下嘗試送越長的packet，那當然越容易碰撞，所以collision time要長才可以避免碰撞。

p_resend = 1/(host_num * coefficient)
會選擇host_num跟第一題很像，是因為 host越多的情況下，等於越多人在固定大小的頻寬下試著送packet，越容易碰撞，所以p_resend應該要小一點才可以減少碰撞機率。

2.3. Setting(host_num=h, packet_num=p, coefficient=1) for h,p in zip(host_num_list, packet_num_list)

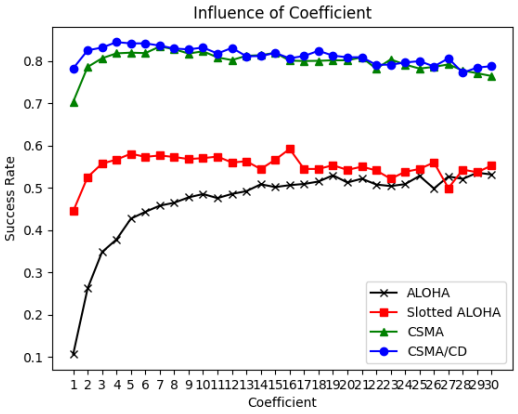


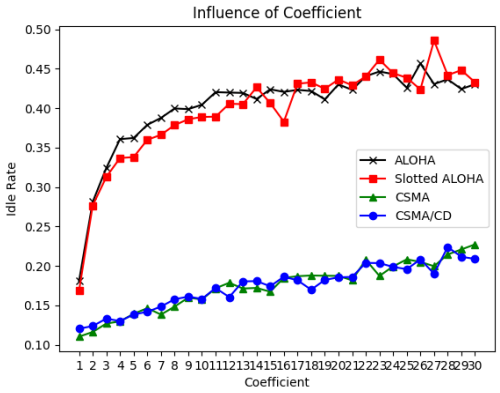


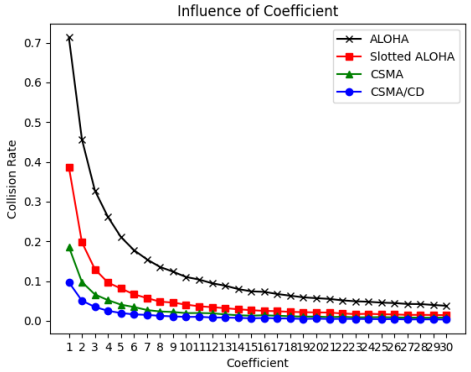


這個問題是Q1的重作板，所以基本上趨勢跟Q1很像，而protocol中值得一說的是slotted aloha的p_resend在納入hostnum作為計算參數後，slotted aloha要馬拿頻寬去成功送packet要馬保持idle(idle rate的上升趨勢則是因為上升的p_resend)而避免了collision造成額外浪費的時間，aloha也因為新公式得到較高的max_colision_wait_time而降低了collision，提高了success，idle也趨於更平緩。總體來說，success rate相較於Q1來說更平穩且平均更高了，collision rate則更平穩且平均更低，這樣表現了我們Q2得出的公式有隨著條件的不同而有最大化success rate/最小化collision rate，且讓每個protocol在不同hostnum的情況下表現更平均。

2.4. Setting(coefficient=c) for c in range(start=1, stop=31, step=1)

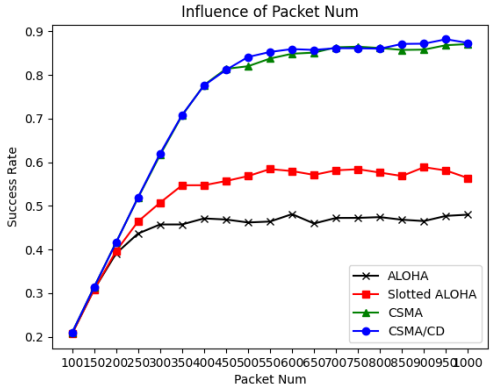


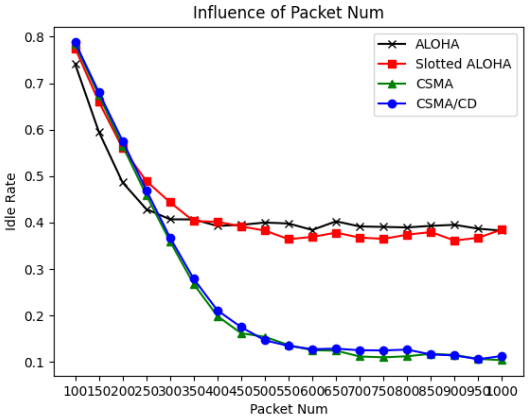


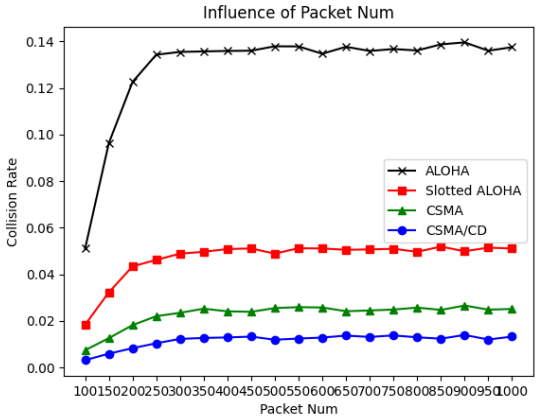


這部份可以看出隨著coefficient的增加，success rate增加，idle rate增加，collision rate降低，而success_rate/collision rate增加/減少到最後都趨於平穩，推測是雖然透過增加coefficient來拉長max_colision_wait_time和降低p_resend以降低碰撞的風險，但因為每個protocol都有他們的極限在(畢竟拉長max_colision_time/拉低p_resend也可能讓channel處於idle，這解釋了idle_rate穩定增加的原因，且網路資源固定)，所以最後才會趨於定值。

2.5. Setting(packet_num=p) for p in range(start=100, stop=1050, step=50)

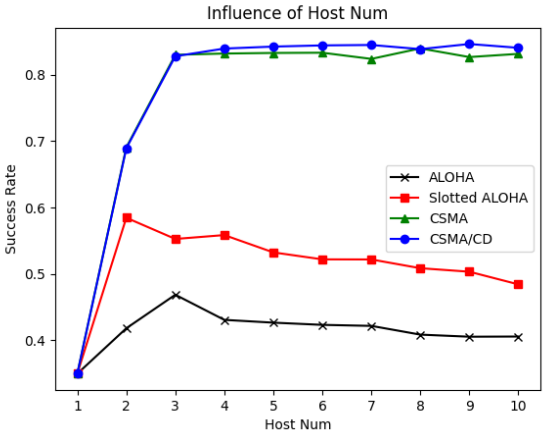


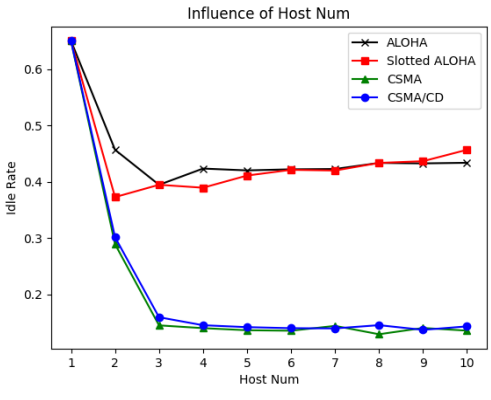


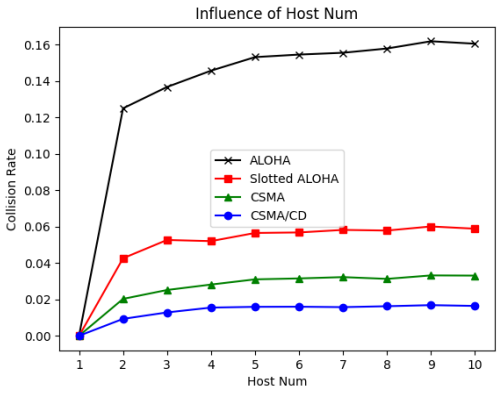


這部份可以看出隨著packet_num增加，success rate和collision rate都增加並趨於定值，idle rate則是降低並趨於定值，這樣是因為隨著packet_num增加，這些封包慢慢塞滿原本處於idle的時間段，而另一方面，隨著塞的packet越多，collision的機會也會越大，因此success rate和collision rate增加，idle rate降低，而最後者都趨於定值是由於各個protocol都有其極限(在固定的網路資源下)，不可能無止境的上升/下降。

2.6. Setting(host_num=h) for h in range(start=1, stop=11, step=1)

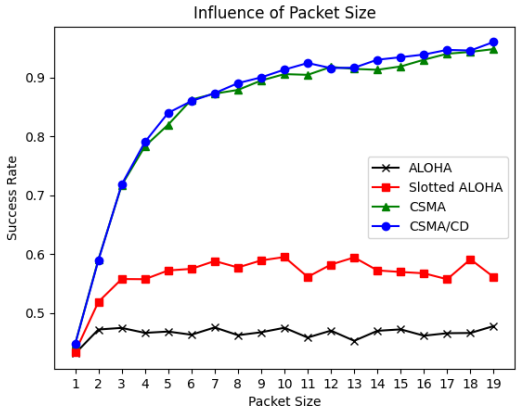


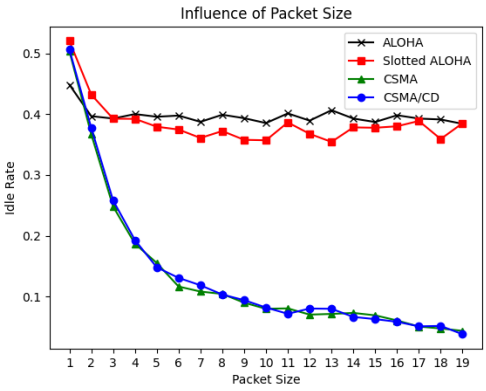


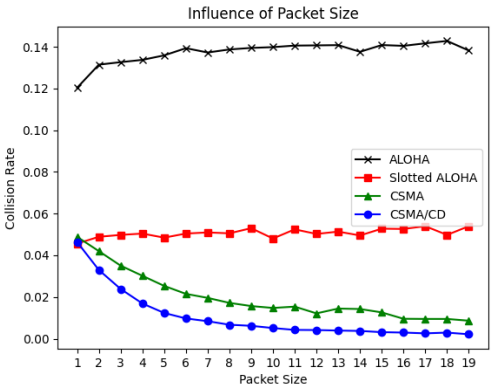


這部份可以看出隨著host_num增加，success rate和collision rate都增加並趨於定值，idle rate則是降低並趨於定值，與第五題非常類似，一開始頻寬沒有充分被利用而幾乎都呈現idle，直到隨著host_num增加，這些多的host送的packet開始利用那些idle的時間來送packet，且配合上我們計算max_colision_wait_time和p_resend的公式，各個protocol合理安排了送packet的時機，讓success_rate上升且idle rate下降，而host增加也會讓collision的機會變大(送packet的人變多)，因此collision rate上升，而趨於定值的原因是各個protocol都有其極限(在固定的網路資源下)，不可能無止境的上升/下降。

2.7. Setting(packet_size=p) for p in range(start=1, stop=20, step=1)



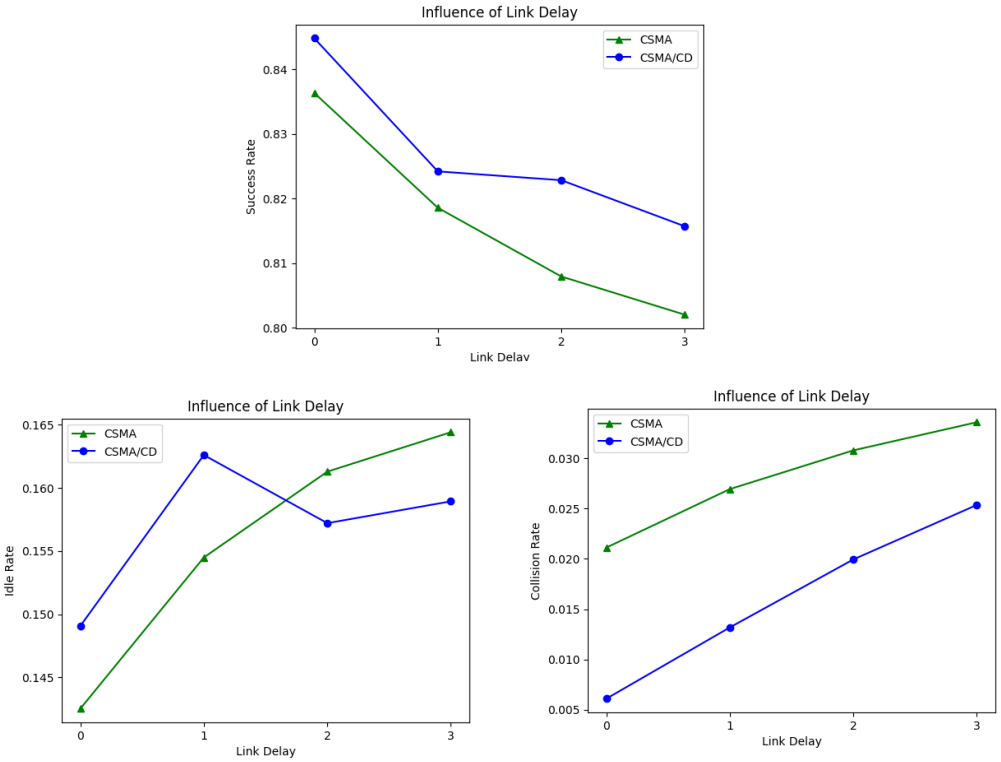




這部份可以看出當packet_size增加時，整體來看success rate上升，idle rate和collision rate下降，其實這題跟前兩題很像，但是collision的趨勢不同，我認為原因是首先aloha系列的機制是想送就送，因此就算packet_size增加，會collision的封包就是會collision，因此collision的趨勢很平(甚至有點上升的感覺)，而csma系列的機制會先檢查channel的狀況，host在嘗試傳輸自己的

封包之前，有更多時間來感知channel是否繁忙，且我們max_colision_wati_time的定義與packet size成正比，host們不太可能同時傳輸。，因此collision rate下降，而趨於定值的原因同樣是因為各個protocol都有其極限(在固定的網路資源下)，不可能無止境的上升/下降。

2.8. Setting(link_delay=l, packet_siz=p) for l,p in zip(link_delay_list, packet_size_list)



這個部份可以看出當link_delay增加，success_rate下降，idle_rate和collision rate 上升，我推測原因是因為越長的link_delay意味著host有更高的機會不知道channel是被佔用的，而導致packet可以送出，卻以collision收場的狀況，因此success rate下降，collision rate上升，而idle rate上升的原因是每個host需要等更久才知道channel沒被佔用，而保留了很多channel屬於idle的時間，因此idle rate上升。