113062647 資工碩
何定綸

無線網路 Lab4

- 實驗目的: 繪製各種 contention-based transmission protocol 的 throughput 圖
- 程式碼 & 解釋:
- 主程式:

```
1   import numpy as np
2   import matplotlib.pyplot as plt
3   import p_persistent_CSMA as P_CSMA
4   import math
```

引用函式庫，p-persistent CSMA 有另外寫成一個檔案 (p_persistent_CSMA)

```
6   # Define the range of offered load G
7   numG = 100
8   G = np.linspace(1e-5, 10, numG)
9
10  # Normalized propagation delay (τ/T)
11  alpha = 0.01
```

事先定義部分參數，G 是在 10^{-5} 到 10 之間平均取 100 個值當作當次實驗的 G 值。

113062647 資工碩
何定綸

```python
# ALOHA
throughput_pure_ALOHA = G * np.exp(-2 * G)  # Pure ALOHA
throughput_slotted_ALOHA = G * np.exp(-G)   # Slotted ALOHA

# CSMA
# Non-persistent CSMA
# slotted
throughput_non_persistent = alpha * (G * np.exp(-alpha * G)) / (1 - np.exp(-alpha * G) + alpha)

# 1-persistent CSMA
numer_1 = G * np.exp(-G * (1 + 2 * alpha)) * (1 + G + G * alpha * (1 + G + 0.5 * G * alpha))
denom_1 = G * (1 + 2 * alpha) - (1 - np.exp(-G * alpha)) + (1 + G * alpha) * np.exp(-G * (1 + alpha))
throughput_1persistent = numer_1 / denom_1
```

ALOHA, Slotted ALOHA, non-persistent CSMA, 1-persistent CSMA 在課程投影片中皆有公式，所以直接將變數對應到公式後寫出，其中 non-persistent 採用的是 slotted 的公式。

```python
# P-Persistent CSMA
N = int(1e4)

rng = np.random.default_rng()
throughput_01persistent = np.zeros(numG)
throughput_05persistent = np.zeros(numG)
throughput_001persistent = np.zeros(numG)

for i, lam in enumerate(G):
    # 隨機生成每兩封包的間隔時間
    inter_arrival_time = rng.exponential(1 / lam, N)
    arrival_times = np.array(inter_arrival_time)
    # 把每個封包的到達時間轉換成總累積時間 -> 表示每個封包在 T = ? 的時間到達
    # for j in range(1, len(arrival_times)):
    #     arrival_times[j] += arrival_times[j - 1]
    arrival_times = np.cumsum(arrival_times)

    throughput_01persistent[i] = P_CSMA.csma_generalized(arrival_times, 0.1, alpha) / math.ceil(arrival_times[-1])
    throughput_05persistent[i] = P_CSMA.csma_generalized(arrival_times, 0.5, alpha) / math.ceil(arrival_times[-1])
    throughput_001persistent[i] = P_CSMA.csma_generalized(arrival_times, 0.01, alpha) / math.ceil(arrival_times[-1])
```

此為 p-persistent CSMA 的主程式片段。N 為總共欲傳輸封包量，rng 是一個 random number generator。

根據事先定義好的 G 值，隨機生成每兩封包彼此的間隔時間(inter_arrival_time)，並將他們累加起來後變成每個封包應該會在什麼時間點送達接收端(arrival_times)。最後根據模擬 p-persistent CSMA 的檔案算出對應 G 值的 throughput。

113062647 資工碩
何定綸

- p_persistent_CSMA:

參考資料:

https://gist.github.com/arunsammit/da15a99164ac616c019fd92c290bcc27

```python
import matplotlib.pyplot as plt
import numpy as np
import math
from scipy.stats import binom
```

引用函式庫

```python
def csma_generalized(arrival_times: np.ndarray, p, cd: bool = False, alpha: float = 0.01) -> int:
    success = 0
    transmission_ends = 0
    collision_detect_possible = 0
    successfully_sending = False
    pending_packets = 0
```

變數、參數設定。以主程式決定的封包到達時間決定當下這個時間點應該做什麼事

```python
    for j, time in enumerate(arrival_times):
        if time < collision_detect_possible:
            # channel is busy, but detected as free due to alpha delay in detection of ongoing transmission
            if cd:
                transmission_ends = min(transmission_ends, time + alpha)
            else:
                transmission_ends = time + 1
            if successfully_sending:
                success -= 1
                successfully_sending = False
```

此為 collision detection

113062647 資工碩
何定綸

```
elif time < transmission_ends:
    # channel is busy, do nothing
    pending_packets += 1
    if (j + 1 < len(arrival_times) and arrival_times[j + 1] >= transmission_ends) \
            or j + 1 == len(arrival_times):
        if p == 1:
            x = pending_packets
        elif p == 0:
            x = 0
        else:
            x = binom(pending_packets, p).rvs()
        if x == 1:
            success += 1
            successfully_sending = True
            collision_detect_possible = transmission_ends + alpha
            transmission_ends = transmission_ends + 1
        elif x > 1:
            collision_detect_possible = transmission_ends + alpha
            transmission_ends = transmission_ends + 1
        pending_packets = 0
```

當封包抵達接收端時，如果還沒到 transmission_ends，表示 channel busy，欲傳輸封包量+1。

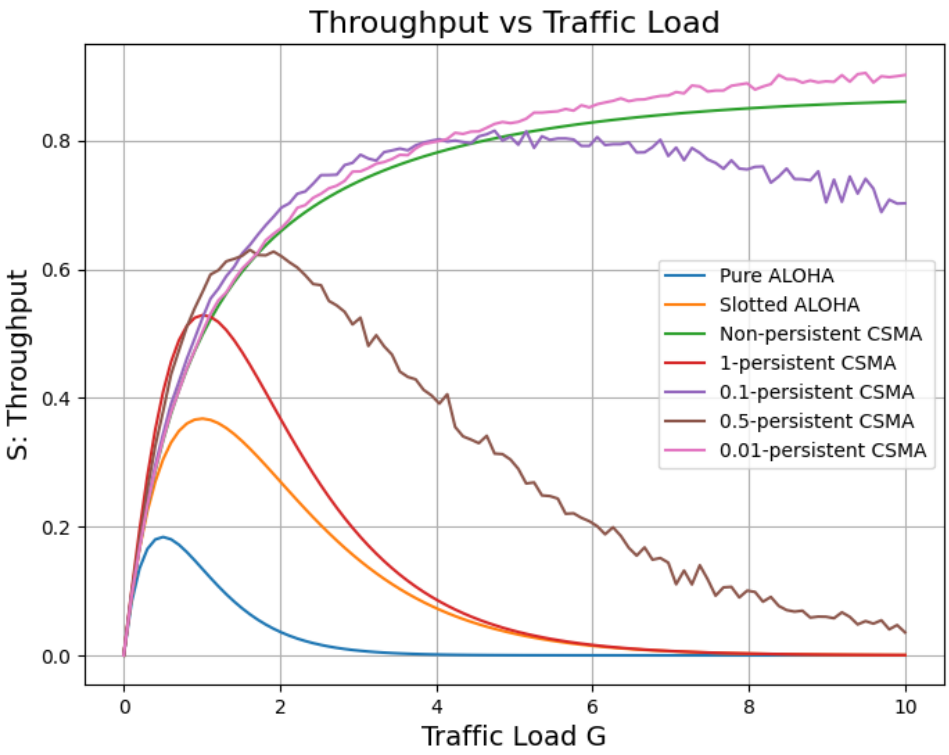
一到傳輸時間結束後根據 p-persistent protocol 決定下個封包的傳輸，依照 binomial(n, p)去決定有幾個封包要傳。如果最後只有一個，則成功傳輸(success + 1)，反之 collision。

113062647 資工碩
何定綸

```python
# Plotting
plt.figure(figsize=(8, 6))
plt.plot(G, throughput_pure_ALOHA,        label='Pure ALOHA')
plt.plot(G, throughput_slotted_ALOHA,     label='Slotted ALOHA')
plt.plot(G, throughput_non_persistent,    label='Non-persistent CSMA')
plt.plot(G, throughput_1persistent,       label='1-persistent CSMA')
plt.plot(G, throughput_01persistent,      label='0.1-persistent CSMA')
plt.plot(G, throughput_05persistent,      label='0.5-persistent CSMA')
plt.plot(G, throughput_001persistent,     label='0.01-persistent CSMA')

plt.xlabel('Traffic Load G', fontsize=14)
plt.ylabel('S: Throughput', fontsize=14)
plt.title('Throughput vs Traffic Load', fontsize=16)
plt.legend()
plt.grid(True)
plt.show()
```

結果圖呈現



最終結果