

## Punto 3 Parcial / Simulación en Python

3.

### Simulación

```
1 import random # Importamos la libreria random que despues usamos para simular la probabilidad de exito de exito o perdida de transmi
2
3 def simulate_tcp_cwnd(transmissions=20, success_prob=0.8): # Definimos la funcion simulate_tcp_cwnd con dos parametros sencillos, un
4     cwnd = 1 # Inicializamos la ventana de congest
5
6     for i in range(1, transmissions + 1): #Bucle que simula las transmisiones
7         if random.random() < success_prob: # Si el numero aleatorio es menor a la probabilidad de exito
8             cwnd += 1 # Aumenta la ventana en caso de éxito
9             print(f"Transmisión {i}: Éxito - cwnd = {cwnd}") #Simplemete imprimimos el mensaje de exito
10        else:
11            cwnd = 1 # Si la transmisión falla, la ventana se reinicia
12            print(f"Transmisión {i}: Pérdida - cwnd = {cwnd}")
13
14 simulate_tcp_cwnd() # Aca Llamamos a la funcion para que se ejecute
15
```

Ejecución del código:

```
Transmisión 1: Éxito - cwnd = 2
Transmisión 2: Éxito - cwnd = 3
Transmisión 3: Éxito - cwnd = 4
Transmisión 4: Éxito - cwnd = 5
Transmisión 5: Éxito - cwnd = 6
Transmisión 6: Éxito - cwnd = 7
Transmisión 7: Éxito - cwnd = 8
Transmisión 8: Éxito - cwnd = 9
Transmisión 9: Éxito - cwnd = 10
Transmisión 10: Éxito - cwnd = 11
Transmisión 11: Pérdida - cwnd = 1
Transmisión 12: Éxito - cwnd = 2
Transmisión 13: Éxito - cwnd = 3
Transmisión 14: Éxito - cwnd = 4
Transmisión 15: Éxito - cwnd = 5
Transmisión 16: Éxito - cwnd = 6
Transmisión 17: Pérdida - cwnd = 1
Transmisión 18: Éxito - cwnd = 2
Transmisión 19: Éxito - cwnd = 3
Transmisión 20: Éxito - cwnd = 4
```

### Resumen del comportamiento

La ventana de congestión aumenta en cada transmisión exitosa.

Si ocurre una pérdida, la ventana de congestión se reinicia a 1.

Este proceso se repite durante 20 transmisiones.