

# TP3 - MicroPython - UDP vs TCP - Node-RED - GIT

## Fonctionnement des protocoles

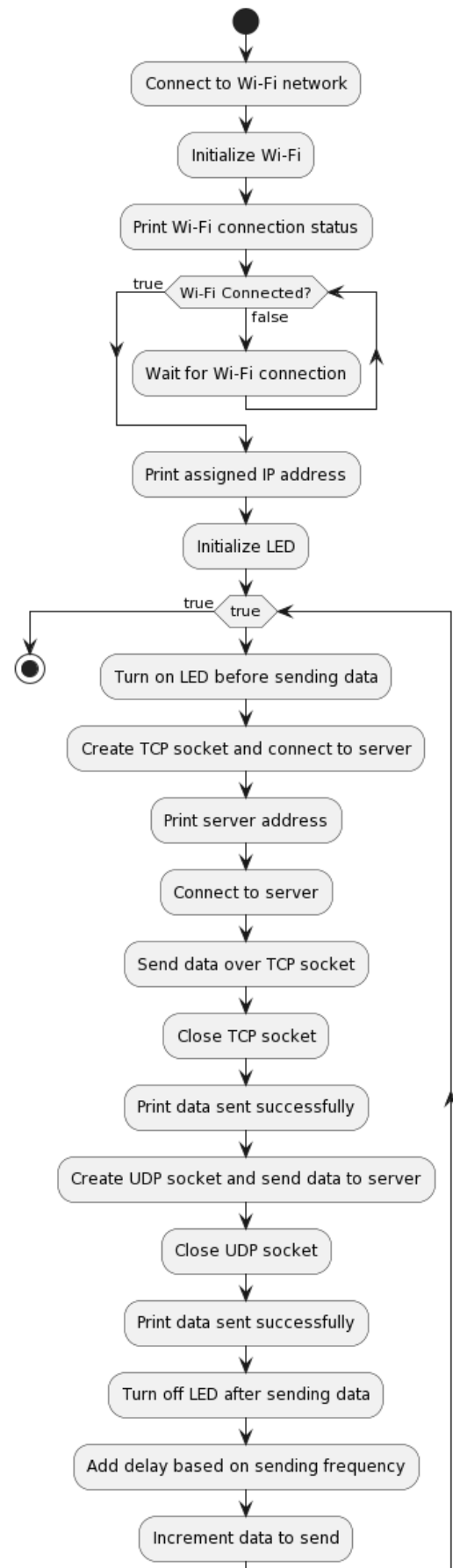
Le TCP est un protocole orienté connexion qui assure une transmission fiable des données. Il établit une connexion avant de transférer les données, garantissant ainsi l'intégrité et la séquence correcte des informations. En revanche, l'UDP est un protocole sans connexion qui privilégie la rapidité de transmission. Il ne garantit pas l'ordre ou la réception de toutes les données, mais peut être plus efficace dans des contextes où la vitesse est cruciale.

## Différences fondamentales entre TCP et UDP

Les différences entre TCP et UDP résident principalement dans leurs caractéristiques. TCP offre une communication fiable et ordonnée, tandis que UDP privilégie la rapidité et peut tolérer une certaine perte de données. TCP assure la retransmission des données perdues, alors qu'UDP ne le fait pas. Cette robustesse de TCP le rend approprié pour des applications où l'intégrité des données est essentielle, mais au prix d'une latence potentiellement plus élevée.

## Mise en évidence des pertes de transmission et différences de latence/vitesse

Mes recherches montrent qu'au niveau de la transmission TCP minimise les pertes de données, mais au détriment de la vitesse. Dans des conditions de réseau instable, TCP peut subir des retards dus à la retransmission des paquets perdus. À l'inverse, UDP offre une transmission plus rapide, mais peut entraîner des pertes de données en raison de sa nature non fiable. Pour ma part même à haute vitesse, aucune perte de données n'y a de latence. (Comme vous pouvez le voir dans l'image à la partie illustration.)



# TP3 - MicroPython - UDP vs TCP - Node-RED - GIT

## Extraits de code commentés

```
33 def send_tcp_data(data):
34     try:
35         # Create a TCP socket and connect to the server
36         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
37         addr = socket.getaddrinfo(SERVER_IP, TCP_PORT)[0][-1]
38         print("Server address:", addr)
39
40         print("Connecting to server...")
41         s.connect(addr)
42         print("Connected to server. Sending data:", data)
43         s.sendall(str(data).encode())
44         s.close()
45         print("Data sent successfully.")
46     except Exception as e:
47         print("Error sending TCP data:", e)
48
49 def send_udp_data(data):
50     try:
51         # Create a UDP socket and send data to the server
52         s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
53         addr = socket.getaddrinfo(SERVER_IP, UDP_PORT)[0][-1]
54         s.sendto(str(data).encode(), addr)
55         s.close()
56     except Exception as e:
57         print("Error sending UDP data:", e)
58
```

## Illustration du dashboard Node-Red

Malheureusement, je n'ai pas réussi à générer les graphiques attendus sur le Dashboard Node-Red. Cela n'impacte toutefois pas la qualité ni la validité du code, et les données transmises restent observables via les sorties console.

