

Universitat de Barcelona

FACULTAT DE MATEMÀTIQUES I INFORMÀTICA

INFORME PRÀCTICA 4: COMUNICACIONS A  
TRAVÉS DE LA PILA TCP/IP

*Xarxes*

Junjie Li i Manuel Liu Wang

Desembre 2022

## 1 Introducció

En aquesta pràctica, continuarem en concentra amb ESP8266 i el packet tracer, i entendrem la seva funcionalitat en la capa TCP/IP.

## 2 Creació del canal recol·lector de dades

A la primer exercici de pràctica vam aconseguir resultats inigualables utilitzant **ESP8266 + CCS811 + ThingSpeak** [1, 2].

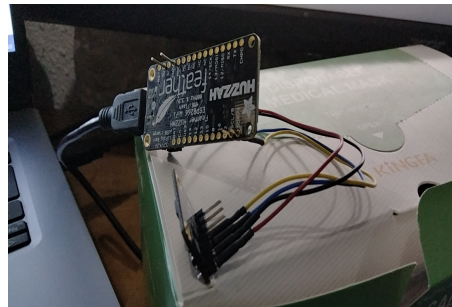


Figure 1: ESP8266 + CCS811 + ThingSpeak

Ja estem familiaritzats amb ESP8266, però què és CCS811?, El CCS811 Air Quality Breakout és un sensor de gas digital que detecta una àmplia gamma de compostos orgànics volàtils totals (TVOC), inclosos els nivells equivalents de diòxid de carboni (eCO<sub>2</sub>) i òxid metàl·lic (MOX) [3].

Per utilitzar ESP8266 i CCS811, hem de parar atenció a la connexió del pin. A CCS811 fem servir VCC, GND, SCL, SDA. En relativament, hauríem de connectar-nos a 3V, GND, SCL/5, SDA/4 pins del ESP8266, podeu veure les figures següents:

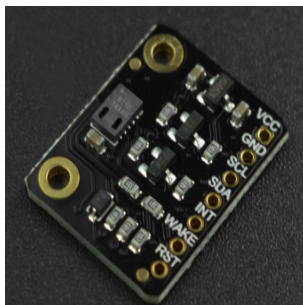


Figure 2: Placa CCS811 Air Quality pinout

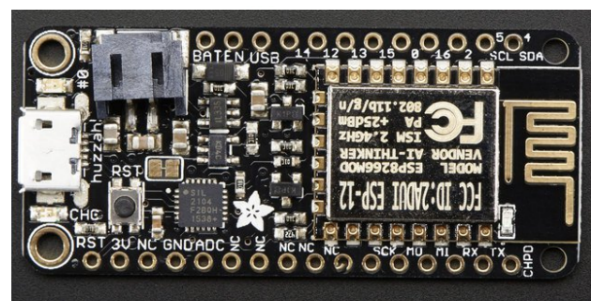


Figure 3: Placa ESP8266 pinout

Amb aquest sensor i ESP8266 no només podem mesurar el valor RSSI de la xarxa, sinó també podem mesurar la qualitat de l'aire, TVOC, eCO<sub>2</sub>, etc, al mateix temps. Podem veure a les dues imatges següents com canvien els seus valors quan el foc s'acosta al sensor:

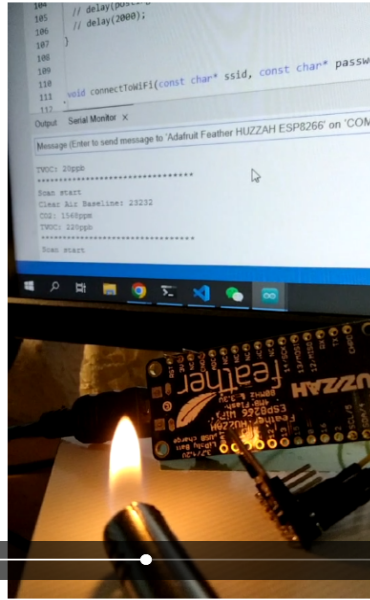


Figure 4: Prova 1 amb CCS811

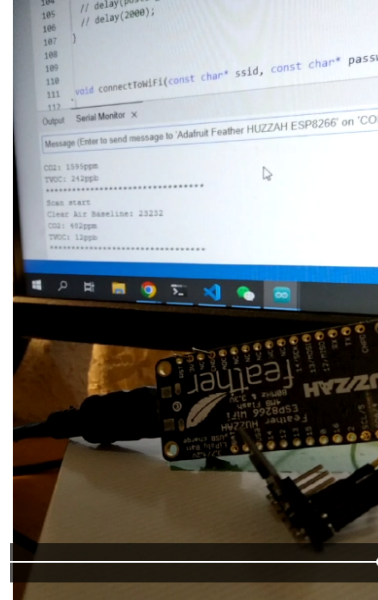


Figure 5: Prova 2 amb CCS811

A partir de les dues imatges anteriors, encara podem veure clarament que quan el foc està a prop del sensor, el canvi de CO2 i TVOC és molt gran, de 400 a 1600.

```
for (int i = 0; i < n; i++) {
  if (WiFi.SSID(i) == "Junjie_8266") {
    long rssi = WiFi.RSSI();
    ThingSpeak.setField(RSSIField, rssi);
  }
  delay(100);
}

if (CCS811.checkDataReady() == true) {
  long airclear = CCS811.readBaseline();
  ThingSpeak.setField(AirQualityField, airclear);

  long co2ppm = CCS811.getCO2PPM();
  ThingSpeak.setField(CO2Field, co2ppm);

  long tvocppb = CCS811.getTVOCPPB();
  ThingSpeak.setField(TVOCField, tvocppb);
} else {
  Serial.println("Data is not ready!");
}

writeTSDData(channelID);
```

Figure 6: codi principal

A la part del codi, la major part del codi es pot utilitzar directament al tutorial. Establiu sol·licituds http, connecteu-vos i envieu dades, cosa que és fàcil d'entendre i no requereix massa comentaris.

El codi més important que creiem es mostra a la figura 6. Utilitzem els mètodes proporcionats per ESP8266 i CCS811, podem obtenir fàcilment les dades que volem. I a través de ThingSpeak, podem mostrar les nostres dades al lloc web d'una manera bonica i concisa. El codi més important que creiem es mostra a la figura 6. Utilitzem els mètodes proporcionats per ESP8266 i CCS811, podem obtenir fàcilment les dades que volem. I a través de ThingSpeak, podem mostrar les nostres dades al lloc web d'una manera bonica i concisa[4, 5].

A la següent figura 7, podem veure clarament el RSSI de la xarxa, el valor estàndard d'aire net, els valors de CO2 i TVOC mesurats per ESP8266 i CCS811:

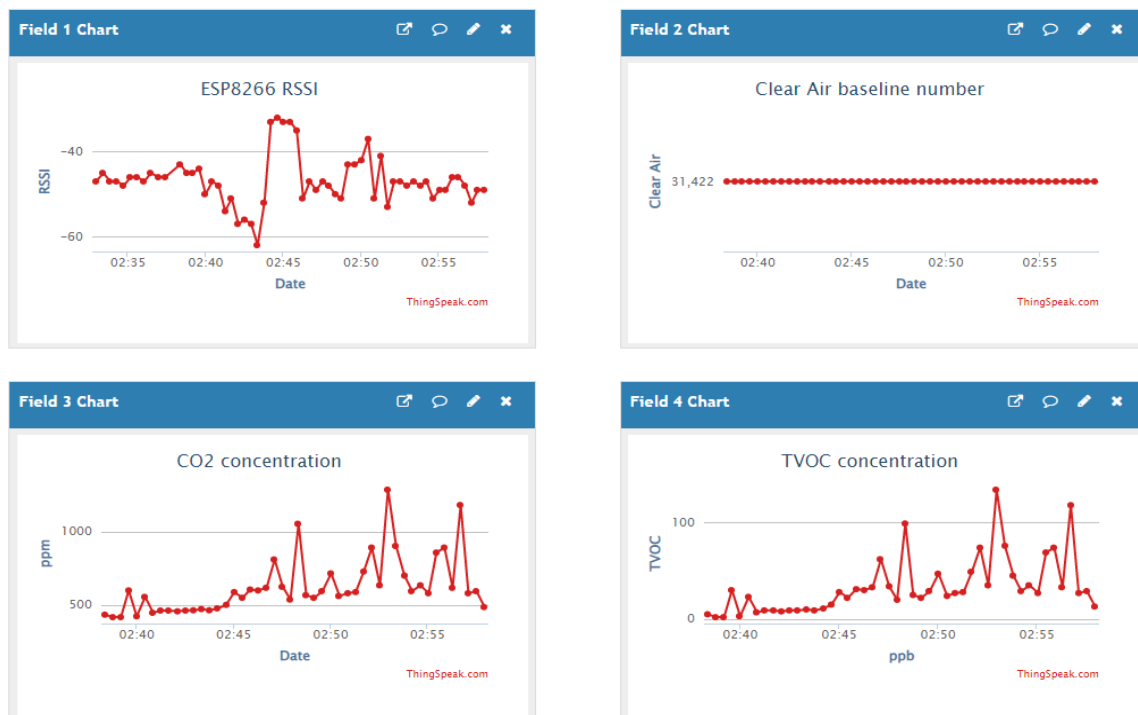


Figure 7: Channel ThingSpeak

Podeu trobar més detalls a la següent referència

### 3 Creació d'un Punt d'Accés

En el segon exercici d'aquesta pràctica, hem de convertir l'ESP8266 en un servidor,

Perquè altres puguin accedir al nostre servidor ESP8266, primer hem d'habilitar el servei de servidor, podem triar una connexió de xarxa, de manera que quan altres utilitzin una LAN amb nosaltres, també puguin accedir al nostre servidor.

```
WiFiServer server(80);

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}
```

Figure 8: Inicialitzar el servidor

Després que el servidor s'hagi carregat correctament per primera vegada, retornarà una adreça IP. Introduïu l'adreça IP i el port 80 al navegador per accedir a la pàgina web del nostre servidor.

```
WiFi connected.
IP address:
192.168.1.136
```

Figure 9: IP i port de servidor

El codi següent ens permet enviar i rebre sol·licituds HTTP al servidor web, bàsicament, tret de la capçalera de la sol·licitud HTTP anterior, la resta són tots fitxers HTML.

```
client.println("HTTP/1.1 200 OK");
client.println("Content-type:text/html");
client.println("Connection: close");
client.println();

client.println("<!DOCTYPE html>");
client.println("<html lang=\"en\">");
client.println("<head>");
client.println("<meta charset=\"UTF-8\">");
client.println("<meta http-equiv=\" X-UA-Compatible \" content=\"IE=edge\">");
client.println("<meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\">");
client.println("<title>ESP8266_Junjie Web Server</title>");
```

Figure 10: Codi per sol·licitud HTTP

Per a més detalls, podeu veure el codi d'arduino codi\_2 o web.html a la mateixa carpeta.

Mitjançant la sol·licitud http anterior, el nostre servidor pot generar una pàgina web html segons la plantilla html després de rebre la sol·licitud. El patró específic de la pàgina web és el mateix que es mostra a la figura 10.

Els components de la pàgina web són molt senzills. Només hi ha 4 botons a tota la pàgina web amb els quals interactuar. Després d'utilitzar les dades detectades per ESP8266 i CCS811 respectivament, podem optar per enviar valors RSSI, indicador d'aire net, CO2 i TVOC al lloc web de ThingSpeak: <https://thingspeak.com/channels/1991175>, és una pàgina web de thinkspeak oberta, que es pot navegar a voluntat.

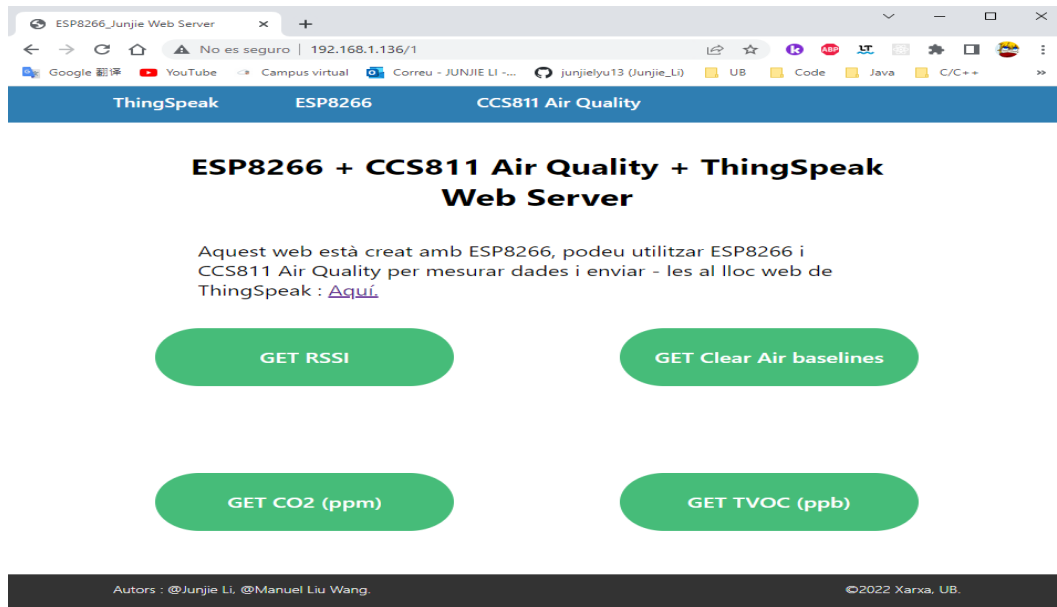


Figure 11: Servidor Web

```
if (header.indexOf("GET /1") ≥ 0) {
    int n = WiFi.scanNetworks();
    if (n == 0) {
        Serial.println("no networks found");
    } else {
        for (int i = 0; i < n; i++) {
            if (WiFi.SSID(i) == "MIWIFI_2G_bPkr") {
                long rssi = WiFi.RSSI(i);
                writeTSData(channelID, RSSIField, rssi);
            }
            delay(100);
        }
    }
}

} else if (header.indexOf("GET /2") ≥ 0) {
    long airclear = CCS811.readBaseLine();
    writeTSData(channelID, AirQualityField, airclear);
} else if (header.indexOf("GET /3") ≥ 0) {
    long co2ppm = CCS811.getCO2PPM();
    writeTSData(channelID, CO2Field, co2ppm);
} else if (header.indexOf("GET /4") ≥ 0) {
    long tvocppb = CCS811.getTVOCPPB();
    writeTSData(channelID, TVOCField, tvocppb);
}
```

Figure 12: Codi de router

Per entendre la lògica interactiva de cada botó de la figura 10, hem de llegir el codi de la figura 11. El codi de la figura 11 és molt senzill. Quan interactuem amb un botó de la pàgina web del servidor, en realitat estem realitzant una operació de salta de pàgina. Sempre que el nostre servidor de backend controli a quina pàgina salten, podem fer les accions o operació corresponent.

1, 2, 3 i 4 corresponen als quatre botons d'esquerra a dreta i de dalt a baix.

## ESP8266\_Junjie Web Server

Channel ID: **1991175**  
Author: mwa0000028597376  
Access: Public

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Add Visualizations

Add Widgets

Export recent data

MATLAB Analysis

MATLAB Visualization

Channel 2 of 2 < >

### Channel Stats

Created: [about 3 hours ago](#)  
Last entry: [5 minutes ago](#)  
Entries: 31

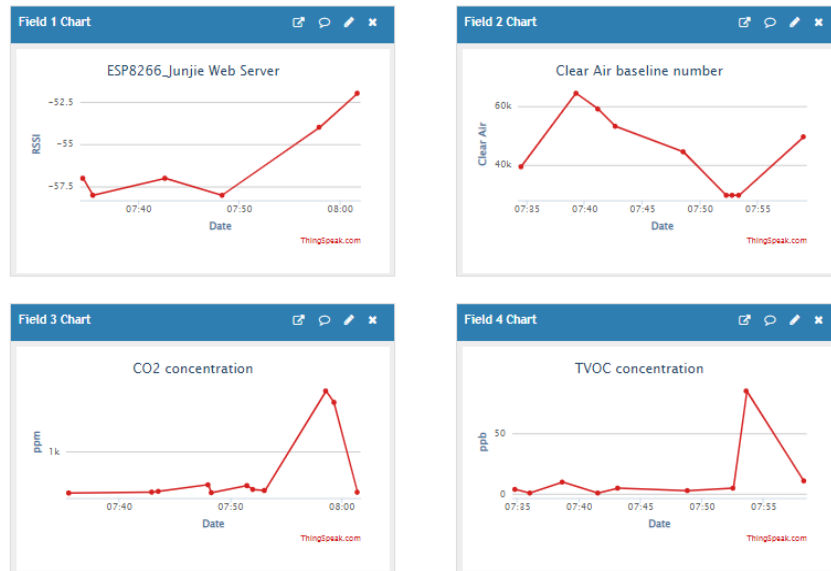


Figure 13: Servidor Web

Cada vegada que premem un botó a la pàgina web del nostre servidor, el nostre servidor llegeix les dades detectades per ESP8266 i CCS811 i les envia a la nostra pàgina web designada de thinkspeak. A la figura 12, podeu veure el nostre programa funcionant perfectament.



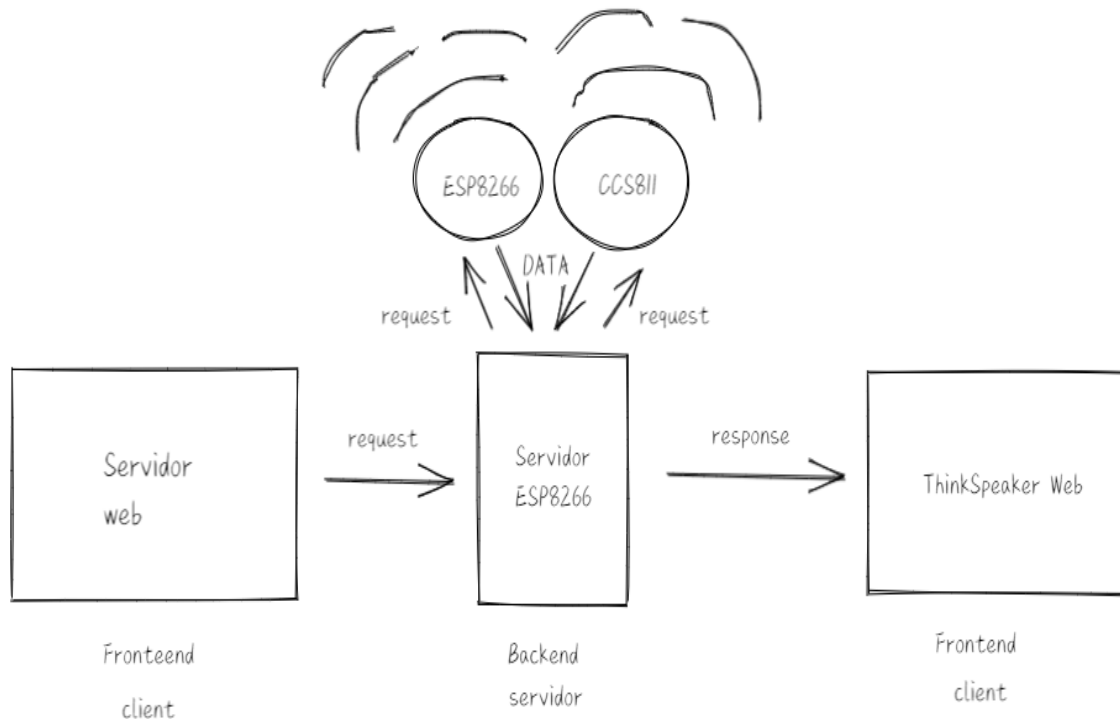


Figure 14: Estructura de client-servidor

Per tal d'explicar bé la nostra estructura client-servidor, hem fet figura13, que explica perfectament la nostra pràctica. El primer que ens crida l'atenció és el servidor web, que és una pàgina web retornada pel nostre accés al port 80 del servidor esp8266. Com a backend, esp8266 s'encarrega de processar la sol·licitud del servidor web, i obté les dades requerides per la sol·licitud d'ESP8266 i CCS811Z segons la sol·licitud, i finalment les envia a la pàgina web de ThinkSpeak.

## 4 Simulació amb Packet Tracer

A la pràctica del segon exercici, hem d'implementar una petita xarxa, que consta de 2 routers, dues petites xarxes d'àrea local(LAN), hem d'establir una connexió entre elles i garantir l'èxit de la seva comunicació.

A la figura 15, podem veure la topologia de la nostra xarxa. Podem veure que hi ha 2 routers, un s'encarrega de connectar un switcher i 4 ordinadors, i l'altre router s'encarrega de connectar un servidor i el núvol.

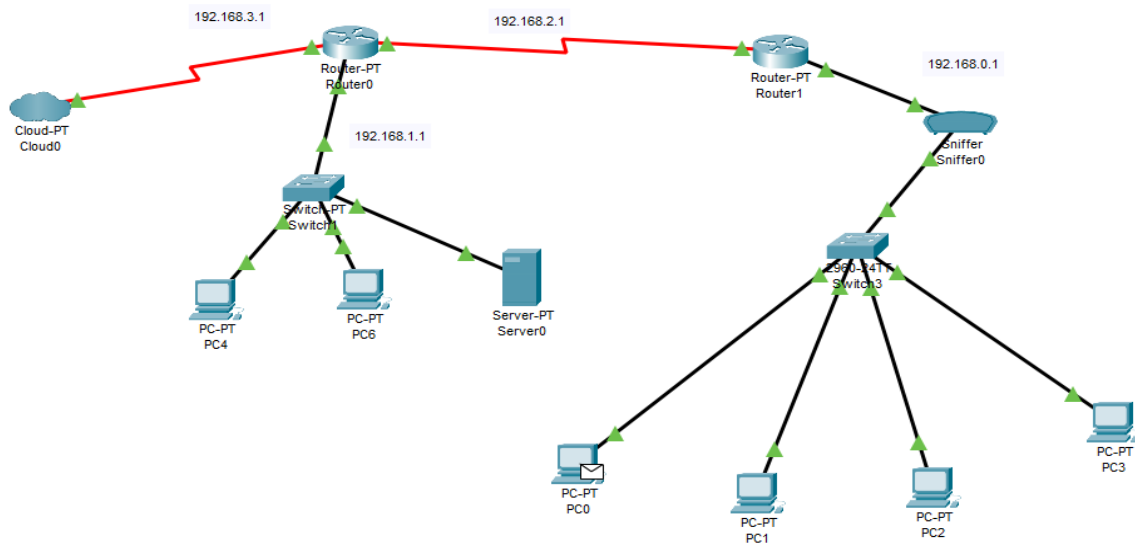


Figure 15: Estructura topologica

La configuració de l'adreça IP i la configuració de router s'introduiran més endavant, i es mostraran més detalls al fitxer Packet tracer. A continuació mostrarem els resultats del seu funcionament:

A la Figura 16, podem trobar a través de la funció de depuració del Packet tracer que hem enviat amb èxit un missatge de PC0 a PC4, és a dir, hem realitzat la comunicació entre dues LAN.

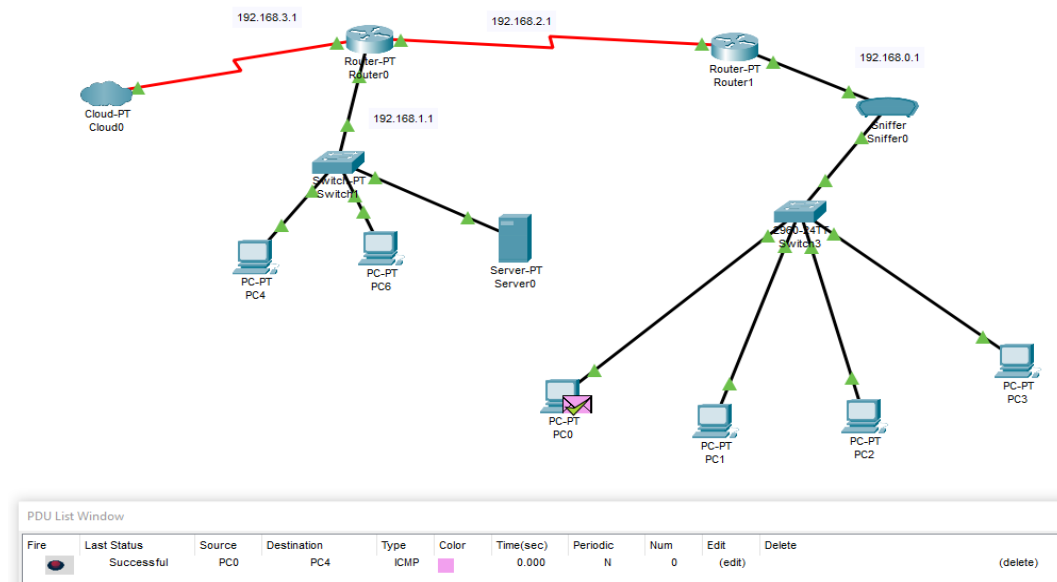


Figure 16: PC0 envia a PC4

De la mateixa manera, podem enviar informació de PC4 de nou a PC0:

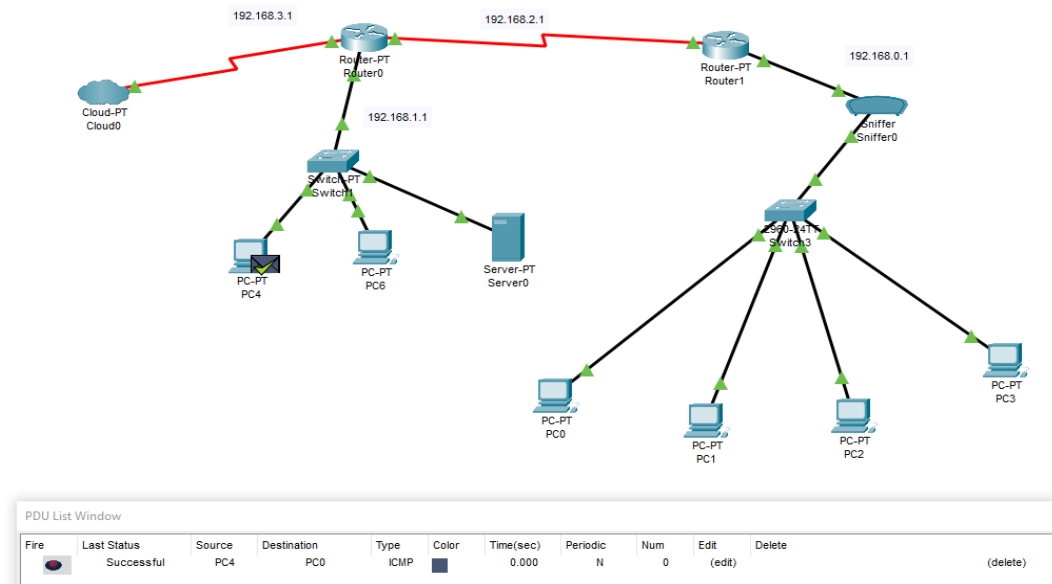


Figure 17: PC4 envia a PC0

No només això, no limitat a la comunicació entre ordinadors, en figura18,19, també podem realitzar l'intercanvi d'informació entre el servidor i l'ordinador.

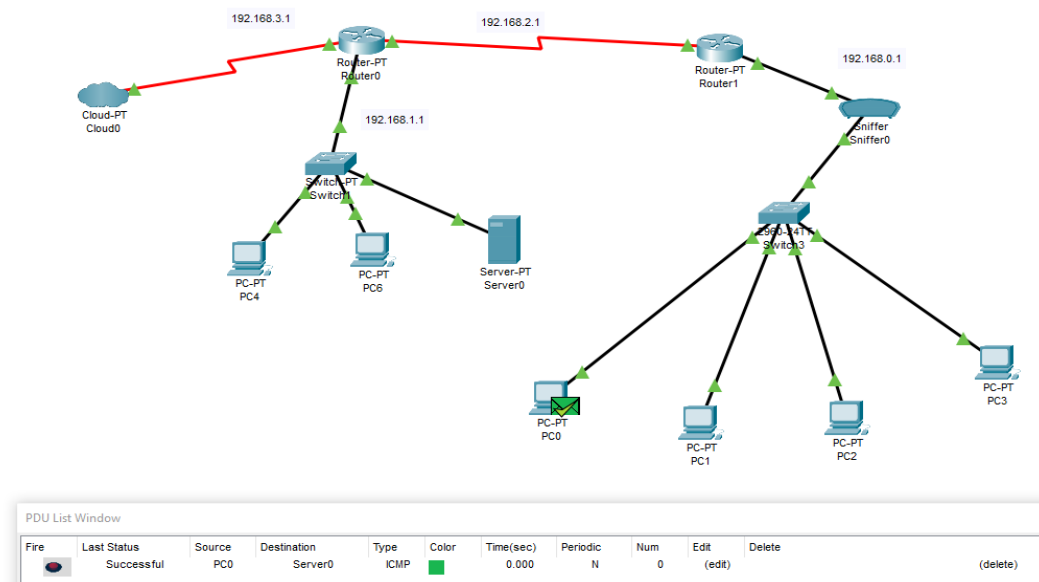


Figure 18: PC a Servidor

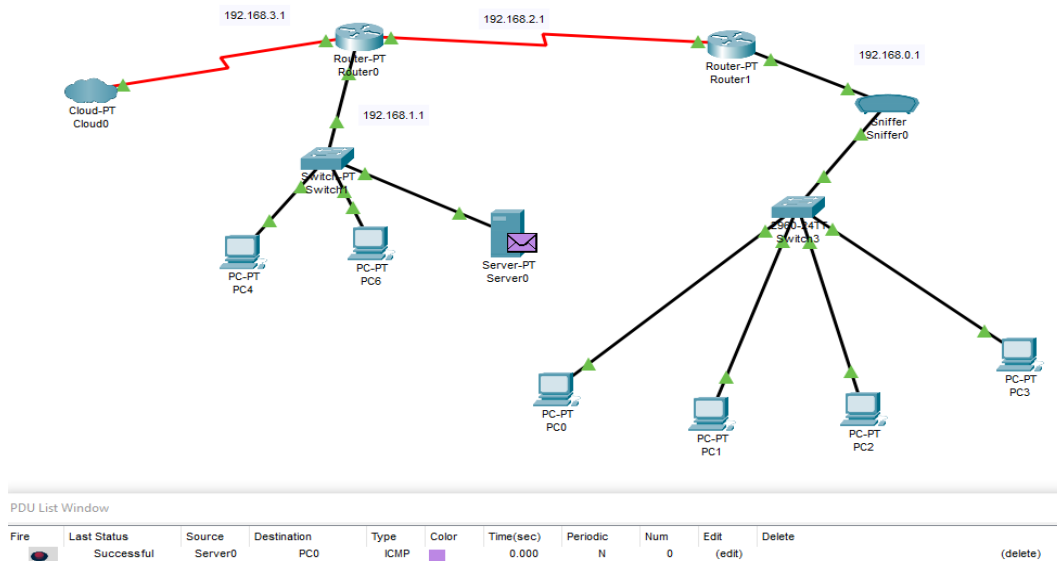


Figure 19: Servidor a PC

És possible que també hagi descobert que en el primer router també hi posem un sniffer, mitjançant el sniffer podem comprovar l'estat de la informació.

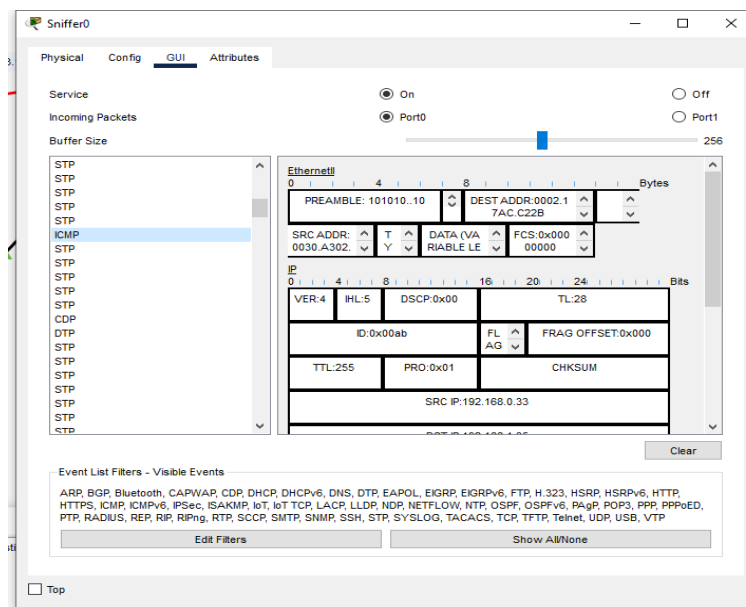


Figure 20: Sniffer

Mitjançant sniffer, podem entendre l'encapsulació de la informació sota diferents protocols de xarxa informàtica.

Per tal de realitzar la comunicació entre routers, així com la comunicació entre router i

servidor, núvol, hem de configurar routers.

Les dues imatges següents són les configuracions del router0 a la figura 15, però les configuracions dels dos encaminadors són gairebé les mateixes.

En una xarxa d'àrea local, totes les adreces IP són desconegudes a l'exterior i tot el que sabem és la informació d'un altre router. Així, quan el nostre ordinador o servidor vol enviar informació a una altra LAN, hem de passar pel router. Configurant el router, quan el nostre router rep una sol·licitud d'un ordinador o servidor, podem localitzar ràpidament la informació que volem en un altre router, i establir una connexió de comunicació bidireccional.

La configuració de RIP permet a l'encaminador saber a quines xarxes estem connectats.

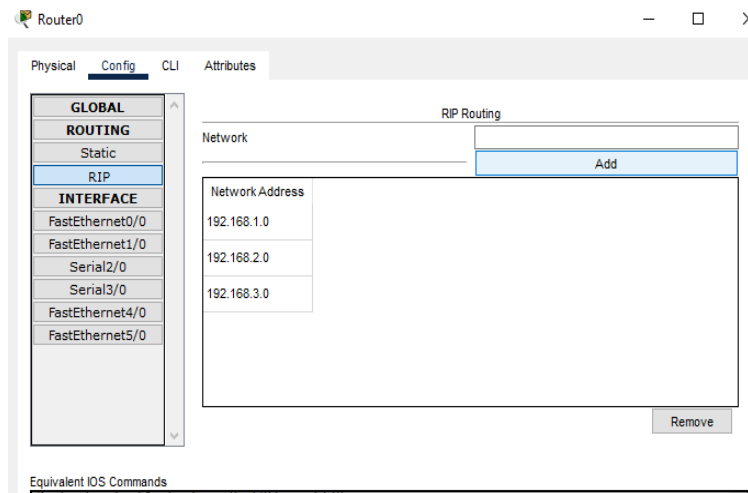


Figure 21: Configuració de router

El propòsit de configurar estàtica és triar quina connexió transmetre quan sabem la destinació que estem enviant.

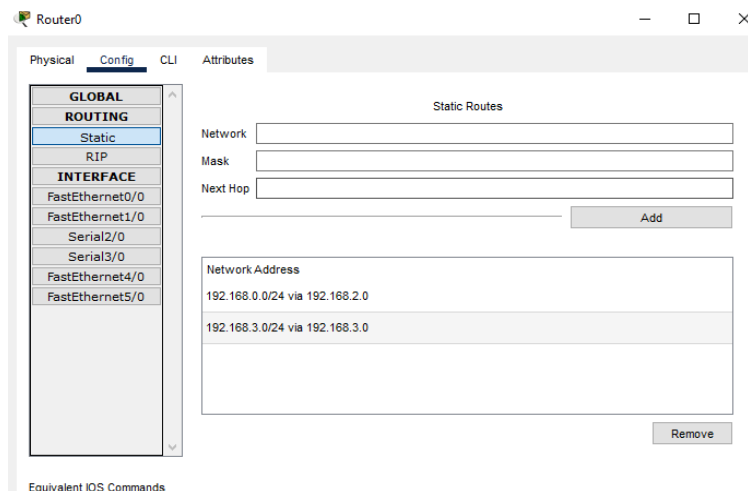


Figure 22: Configuració de router

## 5 Conclusió

En aquesta pràctica, hem entrat en contacte amb diferents usos de l'ESP8266 i el packet tracer, a partir de les classes teòriques que hem après i d'aquesta secció de pràctica, tenim una comprensió més profunda del seu paper a la capa TCP/IP.

## References

- [1] ThingSpeak Arduino: <https://github.com/mathworks/thingspeak-arduino>
- [2] Adafruit Feather HUZZAH ESP8266 <https://www.generationrobots.com/media/adafruit-feather-huzzah-esp8266-user-guide-tutorial.pdf>
- [3] CCS811 Air Quality Sensor <https://www.dfrobot.com/product-1981.html>
- [4] CCS811 Air Quality Sensor Example [https://wiki.dfrobot.com/CCS811\\_Air\\_Quality\\_Sensor\\_SKU\\_SEN0339](https://wiki.dfrobot.com/CCS811_Air_Quality_Sensor_SKU_SEN0339)
- [5] CCS811 Air Quality Sensor Libray [https://github.com/DFRobot/DFRobot\\_CCS811](https://github.com/DFRobot/DFRobot_CCS811)