

Universitat de Barcelona

FACULTAT DE MATEMÀTIQUES I INFORMÀTICA

INFORME PRÀCTICA 2: COMUNICACIONS DE DADES. LA CAPA FÍSICA

Xarxes

Junjie Li i Manuel Liu Wang

Novembre 2022

1 Introducció

En aquesta pràctica, amb la mota i el codi proporcionat en l'enunciat, hem de detectar les diferents xarxes wifi, representacions gràfiques amb el RSSI i el Prx basant-se en diferents distàncies de l'ordinador/router, connectar-nos a una xarxa wifi.

2 Treball realitzat

2.1 Anàlisi de la potència rebuda.

Què fa aquest codi? Expliqueu detalladament el que veieu.

```
1 int n = WiFi.scanNetworks();
2 if(n==0)
3     Serial.println("no networks found");
4 else{
5     for(int i = 0; i<n; i++){
6         Serial.print(i+1);
7         Serial.print(WiFi.SSID(i));
8         Serial.print("(");
9         Serial.print(WiFi.RSSI(i));
10        Serial.print(")");
11        Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE) ? " " : "*");
12        delay(10);
13    }
14 }
```

Figure 1: codi principal

Creiem que la part més important del codi de la primera pregunta és el codi d'aquesta secció. La primera és `scanNetworks()`, que s'utilitza per escanejar les xarxes WiFi disponibles i retornar el nombre de xarxes WiFi trobades. Per a l'anàlisi, utilitzem `SSID()`, `RSSI()` i `encryptionType()`, tornen respectivament el nom de WiFi, la potència del senyal rebut i el tipus de xifratge de la xarxa actual.

.....

Què és la RSSI? Expliqueu detalladament perquè podeu fer servir la RSSI i com es relaciona amb la qualitat del senyal.

RSSI (Received Signal Strength Indicator), és una escala de referència per mesurar el nivell de potència dels senyals rebuts per un dispositiu a les xarxes sense fil (típicament WIFI o telefonia mòbil).

L'RSSI indica intensitat rebuda, l'escala tendeix al valor 0 (zero) com a centre. Generalment l'escala s'expressa dins de valors negatius; com més negatiu, més pèrdua de senyal.

ESP8266 us en **dBm** en unitat.

Taula d'equivalències aproximada per esbrinar el nivell de cobertura en funció dels **dBm** en aire rebuts:

- Més de -76 dBm (números més propers a 0) = Excel·lent
- Entre -89 i -77 := Molt bona
- Entre -97 i -90 := Bona/Mitjana
- Entre -103 i -98 := Baixa cobertura
- Entre -112 i -104 := Baixíssima cobertura (problemes per establir trucades)
- Entre -113 i -132 := Molt poca cobertura (problemes per establir trucades i rendiment baixíssim)

- A partir de -135 := Sense cobertura

Treball de camp.

Primer, utilitzem la línia de la comnada de Windows **Net Wlan show interface** per obtenir el **SSID** WiFi al qual estem connectats ara:

```
JUNJIE_LI@DESKTOP-D024E0C MINGW64 /c/ub/3/tardo (XARXA-P2)
$ Netsh WLAN show interface

Hay 1 interfaz en el sistema:

Nombre                : Wi-Fi
Descripción           : Intel(R) Wi-Fi 6 AX201 160MHz
GUID                  : caa3a248-12d1-411b-9251-9f6882f5eb05
Dirección física      : e4:5e:37:9b:f5:eb
Estado                : conectado
SSID                  : MIWIFI_2G_bPkr
BSSID                 : 2a:d3:f2:c0:19:df
Tipo de red           : Infraestructura
Tipo de radio         : 802.11ac
Autenticación         : WPA2-Personal
Cifrado               : CCMP
Modo de conexión      : Perfil
Canal                 : 36
Velocidad de recepción (Mbps) : 866.7
Velocidad de transmisión (Mbps) : 866.7
Señal                 : 81%
Perfil                : MIWIFI_2G_bPkr

Estado de la red hospedada: No disponible
```

Figure 2: el meu SSID

I fem una petita modificació al codi de l'ESP8266 per localitzar-lo amb més claredat i només mostrar el nostre WiFi.

```
1 if (WiFi.SSID(i) == "MIWIFI_2G_bPkr"){
2   Serial.print(i+1);
3   Serial.print(WiFi.SSID(i));
4   Serial.print("(");
5   Serial.print(WiFi.RSSI(i));
6   Serial.print(")");
7   Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE) ? " " : "*");
8 }
```

Figure 3: codi modificat (exercici_1.ino)

```

Scan start
Distancia: 0m :MIWIFI_2G_bPbR(-33)*
*****
Scan start
Distancia: 1m :MIWIFI_2G_bPbR(-55)*
*****
Scan start
Distancia: 2m :MIWIFI_2G_bPbR(-63)*
*****
Scan start
Distancia: 3m :MIWIFI_2G_bPbR(-66)*
*****
Scan start
Distancia: 4m :MIWIFI_2G_bPbR(-79)*
*****
Scan start
Distancia: 5m :MIWIFI_2G_bPbR(-72)*
*****
Scan start
Distancia: 6m :MIWIFI_2G_bPbR(-70)*
*****
Scan start
Distancia: 7m :MIWIFI_2G_bPbR(-73)*
*****

```

Figure 4: resultat

La figura anterior és el resultat que hem mesurat amb la placa **ESP8266**, a continuació, podem calcular quin és el valor de rssi en l'estat ideal mitjançant la fórmula següent:

$$P_{RX} = P_{TX} \cdot G_{TX} \cdot G_{RX} \cdot \left(\frac{\lambda}{4\pi d} \right)^2 \cdot \eta \quad (1)$$

Distància	RSSI	P _{RX}
0	-33dBm	0.0dBm
1	-55dBm	-70.052dBm
2	-63dBm	-76.072dBm
3	-66dBm	-79.594dBm
4	-79dBm	-82.093dBm
5	-72dBm	-84.031dBm
6	-70dBm	-85.615dBm
7	-73dBm	-86.953dBm

```

1 P_t_x = 1 / 1000
2 G_t_x = 1
3 G_r_x = 1
4 _eta = 1
5 _C = 299792458
6 f = 2.4
7 GHzToHz = 1e9
8 pi = math.pi
9 _lambda = _C / (f * GHzToHz)
10
11 def mWattTodBm(mWatt):
12     return 10 * math.log(mWatt,10) + 30 if mWatt else 0.0
13
14 res = dict()
15 for dist in range(8):
16     Watt = P_t_x * G_t_x * G_r_x * math.pow(( _lambda / (4 * pi * dist) ), 2) *_eta if dist else 0.0
17     mWatt = Watt / 1000
18     dBm = mWattTodBm(mWatt)
19     print(f"mWatt {mWatt} dBm : {dBm}")

```

Figure 5: codi de python (analysis.py)

Podem escriure fàcilment un script de càlcul en Python, La imatge de dalt està en forma de codi de fórmula (1). El que potser haurem de prestar atenció és que quan fem servir la fórmula (1), el P_{RX} que calculem és en **Watt**. Per ser coherents amb el valor experimental, hem de convertir **Watt** a **dBm**. Per tant hem escrit un mètode **mWattTodBm()** per convertir unitats, la fórmula utilitzada és la següent fórmula (2):

$$P_{(dBm)} = 10 \cdot \lg(mWatt) + 30 \quad (2)$$

Després d'utilitzar l'script per calcular el valor **RSSI** en l'estat ideal, podem dibuixar un gràfic per comparar-lo (El codi per dibuixar el gràfic és relativament divers i no es mostrarem, però al fitxer **analisi.py**)

A la **figura 6**, podem veure clarament que la tendència dels resultats de la mesura RSSI i l'estat ideal és molt similar, tot i que pot haver-hi alguns errors de precisió, però els resultats segueixen sent molt satisfactoris.

Quan la distància és 0, la bretxa és relativament gran, perquè 0 és l'estat ideal, és gairebé impossible que el RSSI sigui 0 a l'experiment.

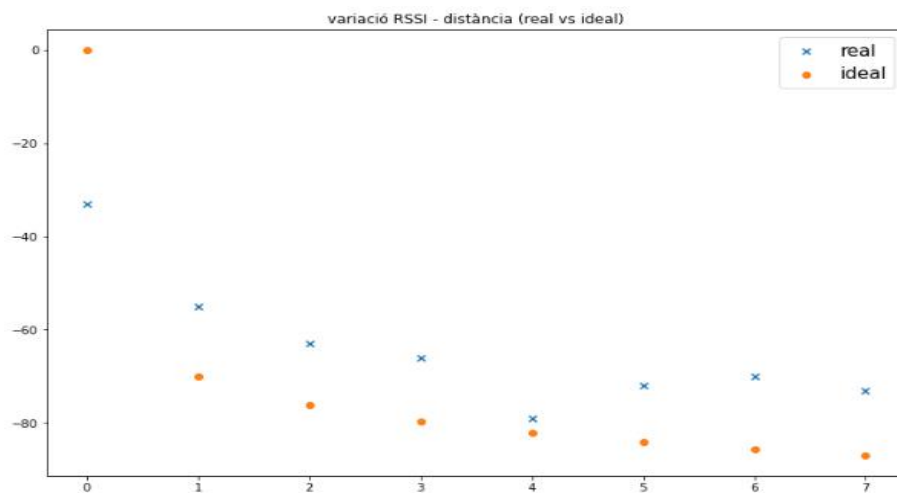


Figure 6: Gràfica de resultat

Pel que fa al problema de la zona morta, estem utilitzant WiFi de 2,4 GHz, la banda de 2,4 GHz proporciona més cobertura i l'abast sense fil disminueix a mesura que augmenta la freqüència, ja que les freqüències més baixes són més fàcils de penetrar en objectes sòlids com ara parets i sòls (excepte alguns materials aïllants característics). Per tant, per a nosaltres, sempre que estigui dins d'un rang raonable i sense alguns materials aïllants específics, no hi ha zona morta per als senyals sense fil a la banda de 2,4 GHz.

A partir de l'expressió teòrica i les dades experimentals, fem també la regressió per tal d'obtenir el millor valor possible de eta. Eta varia entre 0 i 1:

El procés d'implementació es troba a l'script **analysis.py**. A la figura següent, podem veure clarament que quan la distància és propera i quan eta és igual a 1, la nostra resposta és la millor, és a dir, quan l'eta és més gran, la resposta és la millor, i quan l'eta és menor, la resposta és pitjor que ideal. Quan la distància és llunyana, la resposta amb una eta menor pot ser el millor resultat.

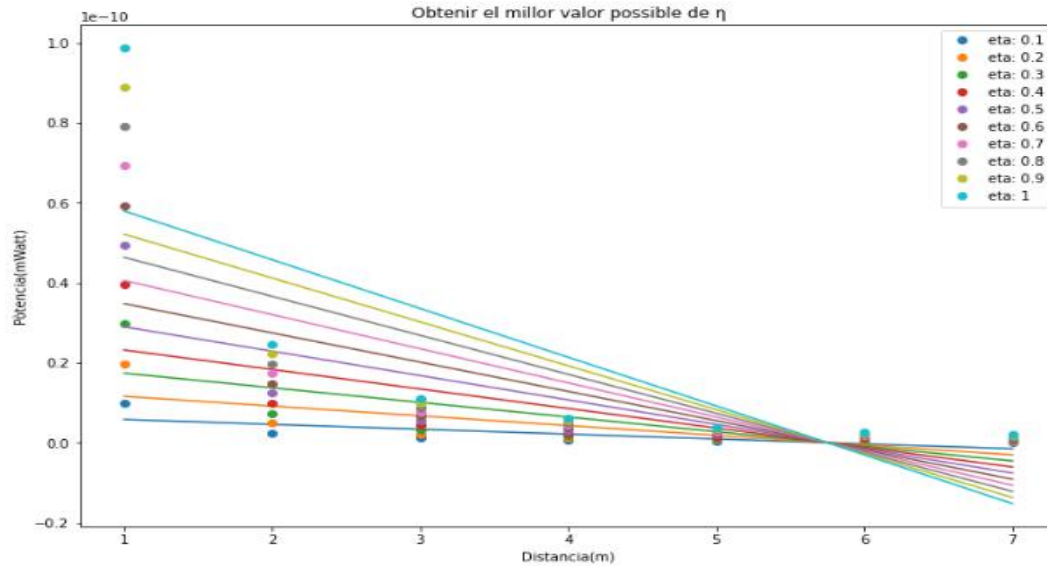


Figure 7: Potència amb diferents η

2.2 Connectar WiFi.

Per connectar correctament l'ESP8266 al wifi del nostre telèfon mòbil, hem de fer algunes modificacions al codi(exercici_2.ino). La següent figura és la part principal del codi ESP8266. Abans d'això, hem d'encendre la xarxa del telèfon mòbil. A la figura 8, podem veure la nostra configuració de SSID i contrasenya de hostpot.

```

1 if (WiFi.SSID(i) == "Junjie_8266") {
2   Serial.print(WiFi.SSID(i));
3   Serial.print("(");
4   Serial.print(WiFi.RSSI(i));
5   Serial.print(")");
6   Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE) ? " " : "*");
7
8   wifiNetworkSelection();
9
10  str_ssid.trim();
11  ssid = str_ssid.c_str();
12
13  str_password.trim();
14  password = str_password.c_str();
15
16  connectToWifi(ssid, password);
17 }

```

Figure 8: codi principal per connectar WiFi (exercici_2.ino)

Al codi de la figura 7, el que hem de fer és, quan supervisem el nostre propi hostpot, cridar a **wifiNetworkSelection()** per seleccionar SSID i introduir la contrasenya, i després canviar el SSID i la contrasenya del format de string a char, al final fem servir **connectToWifi()** per connectar-se a la al wifi del nostre telèfon mòbil.

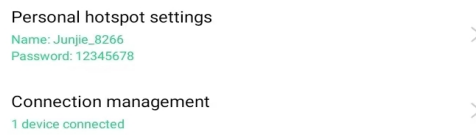


Figure 9: Personal hotspot

A la figura següent, podeu veure el flux d'execució i el resultat del nostre codi. Quan introduïm el SSID i la contrasenya corresponents, ens podem connectar al nostre propi hotspot.

```
Setup done
start connecting...
Junjie_8266(-34) *
SSID? >
SSID selected > Junjie_8266

Password? >
PWD > 12345678

Try to connecting...
Connecting...
Connecting...
Connecting...
Connecting...
Connecting...
Connecting...
Connecting...
*****
Junjie_8266(-32) *
WiFi connecting successful
192.168.114.242
*****
```

Figure 10: Connectar WiFi resultat

3 Conclusió

En conclusió, en aquesta pràctica hem après a identificar el RSSI i el Prx de la xarxa wifi connectada, hem entès el que fa el codi donat, la relació entre el resultat experimental i ideal i après alguns conceptes nous.