Exploració de l'estàndard IEEE802.11ah WiFi HaLow

Estudi de WiFi HaLow amb el mòdul AHPI7292S

V1.0 / Juliol 2025

Jaume Comas / IoT Research Group

i2CAT Foundation
The Internet Research Centre
C/ Gran Capità 2-4, Edifici Nexus I
2a planta, 08034 Barcelona

Taula de continguts

1. Marc teòric

WiFi HaLow, basat en l'estàndard IEEE 802.11ah, representa una extensió de la família WiFi dissenyada específicament per a Internet of Things (IoT). Aquesta tecnologia es distingeix per quatre característiques clau: opera en bandes de baixa freqüència (per sota d'1 GHz), cosa que li confereix un llarg abast significativament superior al WiFi convencional (aprox. 1km), incorpora mecanismes avançats d'estalvi d'energia per optimitzar la vida útil de la bateria dels dispositius, i és capaç de connectar un elevat nombre de dispositius. Aquestes propietats el fan ideal per a aplicacions que requereixen una àmplia cobertura i baix consum, com ara la domòtica, l'agricultura intel·ligent, sensors industrials i ciutats intel·ligents, facilitant la connectivitat eficient de milers de dispositius distribuïts en grans àrees.

1.1. Capa física

La capa física (PHY) de WiFi HaLow (IEEE 802.11ah) és una adaptació de l'OFDM de l'estàndard 802.11ac, dissenyada específicament per a les exigències de Internet of Things (IoT) en termes de llarg abast i eficiència energètica. La seva característica més distintiva és l'ús de bandes de freqüència per sota d'1 GHz, les quals permeten una penetració superior dels obstacles i un abast de senyal significativament major que el WiFi convencional, podent estendre's fins a 1 km en condicions ideals. Aquesta capa també ofereix flexibilitat en l'ample de banda del canal, suportant des d'estrets 1 MHz (ideals per a la robustesa del senyal i baix consum) fins a 16 MHz. Per optimitzar la transmissió, utilitza diversos esquemes de modulació i codificació (MCS), adaptant-se dinàmicament a les condicions del canal per equilibrar la velocitat i la fiabilitat, i incorpora mecanismes avançats d'estalvi d'energia que minimitzen el consum dels dispositius amb bateria. A més, la PHY de HaLow sovint fa ús d'un "down-clocking" de 10x respecte a la d'802.11ac, allargant la durada dels símbols OFDM i fent-la particularment apta per a la connectivitat a gran escala en xarxes IoT.

1.1.1. Gestió de Canals en WiFi HaLow

La capa física de WiFi HaLow opera en bandes de freqüència per sota d'1 GHz, com els 900 MHz (p. ex., 863-868 MHz a Europa), per garantir un abast estès i una millor penetració. Tot i que l'estàndard permet amples de banda fins a 16 MHz, les regulacions de la Unió Europea limiten freqüentment els canals a 1 MHz o 2 MHz. Aquesta restricció regulatòria es tradueix en taxes de dades màximes inferiors, però és el compromís per aprofitar els avantatges de l'abast i l'eficiència energètica que ofereixen aquestes freqüències baixes en entorns IoT.

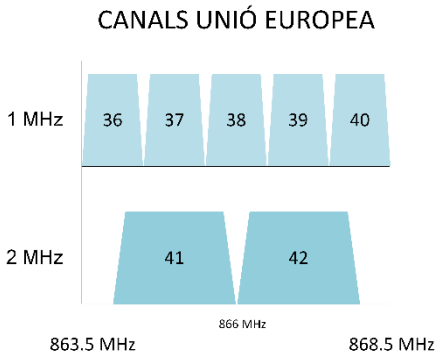


Figura 1: Representació freqüències que s'utilitzen a la unió europea per WiFi HaLow

Falta buscar canals exactes i tot

1.1.2. Taula MCS

A la Taula 1 podem observar les diferents modulacions i el throughput teòric màxim per a cada Modulation and Coding Scheme (MCS). Aquests valors varien en funció de l'ample de banda del canal (1 MHz o 2 MHz) i de l'interval de guarda (GI: curt o llarg).

Tot i que l'estàndard defineix MCS des del 0 fins al 9, el hardware amb el qual treballarem, l'AHPI7292S, només ofereix suport per als MCS del 0 al 7. A més, cal tenir en compte que considerem un únic spatial stream per a aquestes mesures.

MCS Index	Modulation type[1]	Coding rate[2]	Data rate (Mbit/s)			
			Canal 1MHz		Canal 2 MHz	
			Short GI (8µs)	Long GI (4µs)	Short GI (8µs)	Long GI (4µs)
0	BPSK	1/2	0.3	0.33	0.65	0.72
1	QPSK	1/2	0.6	0.67	1.3	1.44
2	QPSK	3/4	0.9	1.0	1.95	2.17
3	16-QAM	1/2	1.2	1.33	2.6	2.89
4	16-QAM	3/4	1.8	2.0	3.9	4.33
5	64-QAM	2/3	2.4	2.67	5.2	5.78
6	64-QAM	3/4	2.7	3.0	5.85	6.5
7	64-QAM	5/6	3.0	3.34	6.5	7.22

Taula 1: Modulation and Coding schemes

La Taula 2 detalla les taxes de dades màximes teòriques per als diversos MCS, considerant un únic flux espacial (single spatial stream) i un símbol OFDM de durada normal.

[1] **Modulation type:** es refereix a la tècnica utilitzada per codificar dades digitals en un senyal analògic apte per a la transmissió.
[2] **Coding rate:** Indica la proporció de bits de dades útils respecte al nombre total de bits transmesos, incloent-hi els bits redundants afegits per a la detecció i correcció d'errors.

1.2. Capa MAC (Control d’accés al medi)

La capa MAC (Medium Access Control) de WiFi HaLow (IEEE 802.11ah) és fonamental per a IoT, ja que aquesta es centra en un mecanisme clau per estalviar energia i augmentar la durabilitat de les bateries de milers de dispositius. Aquesta reorientació estratègica permet que dispositius amb alimentació limitada romanguin operatius durant anys, minimitzant els períodes d'activitat del receptor.

1.2.1. Creació de TIM Groups

Per gestionar eficientment un gran nombre de dispositius, la capa MAC de HaLow introdueix els Traffic Indication Map (TIM) Groups. Els dispositius s'assignen a grups més petits basats en el seu AID, permetent que l'AP només activi una part del TIM del Beacon per indicar dades pendents. Això significa que els dispositius que no pertanyen a un grup actiu poden romandre més temps en mode de baix consum, reduint significativament el seu consum d'energia en comparació amb el WiFi tradicional.

1.2.2. Mecanismes d’estalvi d’energia

La capa MAC d'IEEE 802.11ah incorpora diversos mecanismes innovadors d'estalvi d'energia. Un dels més destacats és el Target Wake Time (TWT), que permet a les estacions negociar amb l'AP un horari específic per despertar-se i comunicar-se. Això permet que els dispositius romanguin en un estat de son de molt baixa potència durant llargs intervals, despertant-se només en els moments acordats per rebre o enviar dades, ideal per a sensors amb transmissions esporàdiques. A més, es milloren les finestres d'accés restringit (Restricted Access Window - RAW), que permeten a l'AP segmentar el període de contenció en petites finestres, assignant-les a un subconjunt d'estacions. Això redueix les col·lisions i la necessitat de les estacions de mantenir-se actives esperant el seu torn, millorant l'eficiència energètica i la capacitat de la xarxa.

1.2.3. Format Beacons DTIM i TIM

El disseny dels Beacons de HaLow és crucial per als seus objectius d'estalvi d'energia. Es permeten intervals DTIM (Delivery Traffic Indication Map) molt més grans que en els estàndards Wi-Fi anteriors, fins a 255. Això significa que els dispositius poden romandre en mode de son durant períodes prolongats entre dos DTIM Beacons consecutius, ja que no necessiten despertar-se tan sovint per comprovar el tràfic broadcast o multicast. A més, el TIM (Traffic Indication Map) dins dels Beacons s'ha adaptat per suportar els TIM Groups, permetent que l'AP assenyali de manera més eficient quines estacions tenen dades pendents al buffer. Les estacions només necessiten llegir la porció del TIM rellevant per al seu grup, minimitzant el temps de processament i l'activitat de la ràdio, essencial per a la gestió energètica en xarxes massives de dispositius IoT.

1.3. Anàlisi mòdul WiFi HaLow AHPI7292S

El mòdul AHPI7292S de l'empresa Alfa Network és un Hardware Attached on Top (HAT) dissenyat per a Raspberry Pi, que integra el xipset Newracom NRC7292. Aquesta combinació proporciona una plataforma de hardware ideal per iniciar-se i aprofundir en la tecnologia WiFi HaLow. Gràcies al seu disseny i al xipset sub-GHz, el mòdul permet explorar des d'un nivell molt baix el funcionament dels protocols de WiFi HaLow (IEEE 802.11ah) i els seus diversos modes d'operació. Un dels recursos més valuosos que ofereix és el mode Sniffer, que facilita la captura i l'anàlisi de les trames a nivell físic de les comunicacions, cosa essencial per a la depuració i la comprensió del comportament del sistema. Aquesta eina esdevé, per tant, un punt de partida sòlid per a la investigació pràctica d'aquest estàndard emergent.

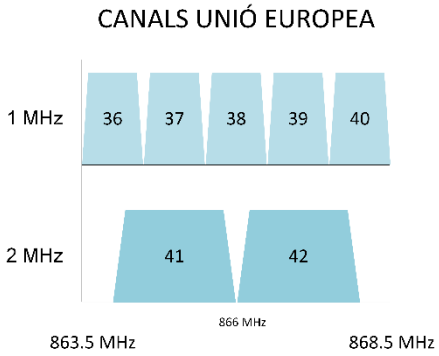
1.3.1. Característiques tècniques

El mòdul AHPI7292S és un HAT (Hardware Attached on Top) per a Raspberry Pi que integra el xipset Newracom NRC7292, un SoC (System-on-Chip) compatible amb l'estàndard IEEE 802.11ah. Aquesta plataforma opera en la banda sub-1GHz, amb suport per a freqüències com 866 MHz (UE) o 915 MHz (EUA), i utilitza modulació OFDM (Orthogonal Frequency-Division Multiplexing) amb esquemes com BPSK, QPSK, 16QAM i 64QAM. Ofereix amples de banda de canal d'1, 2 i 4 MHz, aconseguint taxes de dades PHY entre 150 Kbps i 15 Mbps. El NRC7292 incorpora una doble CPU ARM Cortex-M3 i Cortex-M0, juntament amb mecanismes d'estalvi d'energia avançats com el TWT (Target Wake Time). Disposa d'una potència de sortida lineal de 0 dBm amb un guany de transmissió de 30 dB, i la capacitat de suportar external FEM devices (Front-End Modules) per a una potència de transmissió final més elevada. El xipset suporta diversos modes d'operació, incloent-hi Estació (STA), Punt d'Accés (AP), Sniffer i Mesh, tot i que aquest últim no s'utilitzarà en aquesta memòria. A més, proporciona interfícies com SPI, UART i I2C per a les comunicacions, especialment quan el mòdul funciona en mode standalone, és a dir, quan opera de forma independent sense estar connectat a una Raspberry Pi.

Característiques tècniques AHPI7292S	
Xipset	Newracom NRC7292 (IEEE 802.11ah SoC)
Banda de Freqüència	Sub-1GHz (847 MHz, 866 MHz (EU), 915 MHz (US), 922 MHz, 924 MHz)
Modulació	OFDM (BPSK, QPSK, 16QAM, 64QAM)
Ample de Banda Canal	1 MHz, 2 MHz, 4 MHz
Taxes de Dades PHY	150 Kbps - 15 Mbps
CPU Integrada	Dual ARM Cortex-M3 (Wi-Fi/aplicació) i Cortex-M0 (WLAN)

Mecanismes d'Estalvi	TWT (Target Wake Time)
Interfícies	SPI, UART, I2C, GPIO, ADC
Modes Suportats	Estació (STA), Punt d'Accés (AP), Sniffer, Relay, Mesh
Potència TX (Lineal)	0 dBm
Guany TX Range	30 dB
Potència TX Màxima	0 dBm (Xipset) + 30 dB (HAT) = 30 dBm
Connector Antena	1 × IPEX/U.FL
Factor de Forma	Raspberry Pi HAT
RF Transceptor	750-950MHz, RX noise figure: 4dB

Un cop analitzades les especificacions, observem que la potència de sortida lineal del xipset és de 0 dBm. No obstant això, el mòdul està dissenyat per suportar mòduls front-end externs (FEMs) que permeten assolir potències finals superiors (30 dBm). A la Unió Europea, l'operació d'HaLow es centra en la banda de 866 MHz, i les regulacions solen limitar l'ús de canals a 1 o 2 MHz, tot i que el xipset suporta 4 MHz. Aquestes restriccions influeixen en les taxes de dades màximes assolibles en la pràctica dins del context regulatori europeu.



Canals que suporta aquest mòdul dins la regulació europea

1.3.2. Modulacions que suporta

L'estàndard IEEE 802.11ah defineix un conjunt d'índexs MCS (Modulation and Coding Scheme). Cadascun d'aquests índexs determina la combinació del tipus de modulació (o constel·lació), la taxa de codificació (Coding Rate) i l'interval de guarda (Guard Interval - GI), essent paràmetres clau per obtenir un valor de throughput teòric. Les modulacions suportades inclouen BPSK, QPSK, 16QAM i 64QAM, les quals, combinades amb la taxa de codificació, influeixen en la densitat de

dades per símbol. El Guard Interval, que pot ser normal o curt (SGI), és el temps d'espera afegit entre la transmissió de símbols OFDM per mitigar la interferència intersímbol (ISI). Totes aquestes combinacions s'han dissenyat per optimitzar el rendiment i la fiabilitat de la comunicació en les condicions de llarg abast i baix consum que caracteritzen Wi-Fi HaLow.

1.3.3. Tipus de modes que ofereix

Com ja hem esmentat, en aquesta memòria farem servir quatre modes d'operació principals: Estació, Punt d'Accés, Sniffer i Relay.

- Mode Punt d'Accés (AP):
  Quan el mòdul opera en mode Punt d'Accés (AP), actua com el concentrador central de la xarxa, al qual es connecten totes les estacions. Gràcies a les millores de l'estàndard 802.11ah en la gestió dels AIDs (Association IDs), permet la vinculació d'un gran nombre de dispositius, aproximadament 8000.
- Mode Estació (STA):
  En mode Estació (STA), el mòdul funciona com un client, connectant-se a un Punt d'Accés. A cada Estació se li assigna un AID únic, que a més la vincula a un grup TIM per optimitzar els mecanismes d'estalvi d'energia.
- Mode Sniffer:
  Aquest mode permet capturar els paquets sense fil que es transmeten per un canal radioelèctric específic. Un cop configurat amb eines com Wireshark, el mode Sniffer facilita l'observació detallada de les trames intercanviades entre un AP i una Estació, essent fonamental per a l'anàlisi i la depuració de les comunicacions.
- Mode Relay:
  El mode Relay és una funcionalitat dissenyada per estendre l'abast d'un Punt d'Accés. Actua com un híbrid entre una Estació i un AP: funciona com una Estació respecte a l'AP i com un AP respecte a altres Estacions, fent de salt de comunicació. Atès l'èmfasi de Wi-Fi HaLow en l'estalvi d'energia, les implementacions de Relay solen limitar-se a configuracions d'un sol salt (one-hop), evitant cadenes de salts múltiples que podrien incrementar el consum i la complexitat.

1.3.4. Interfície que ofereix

Per poder utilitzar correctament el HAT de AHPI7292S s'ha d'instal·lar la imatge proporcionada per alfa network, la quan ja te instal·lat el packatge nrc_pkg i també té habilitades les configuracions perquè funcioni correctament, com la deshablitació del wifi i el bluethoothi altres.

2. Realització de proves

2.1. Configuracions amb AHPI7292S

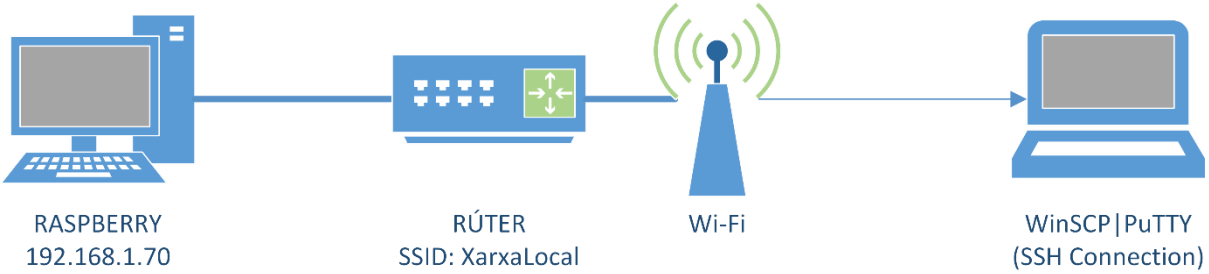
(INSTAL·LACIÓ IMATGE I PROGRAMARI), VERSIÓ IMATGE I VERSIÓ DRIVER NRC

POSAR PDF DE NOVA INSTAL·LACIÓ

2.1.1. Configuració accés remot (SSH)

Per facilitar el treball remot amb les Raspberry Pi, s'ha implementat una connexió mitjançant SSH (Secure Shell). Per a aquesta configuració, hem utilitzat les eines WinSCP (per a la transferència de fitxers i connexions gràfiques) i PuTTY (per a l'accés a la terminal de Linux remota).

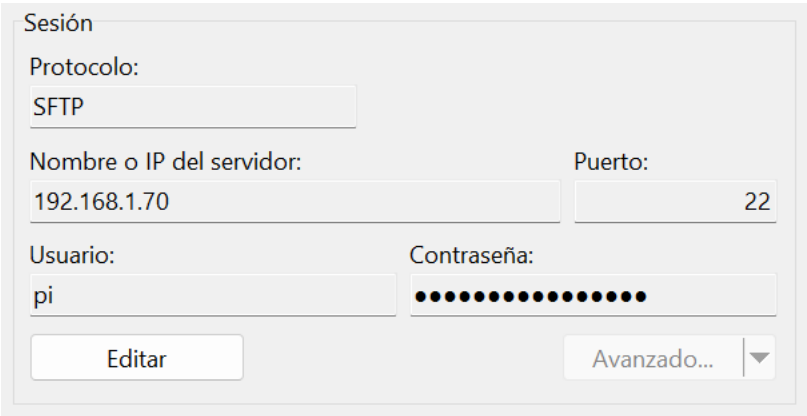
2.1.1.1. Accés des d'una xarxa local



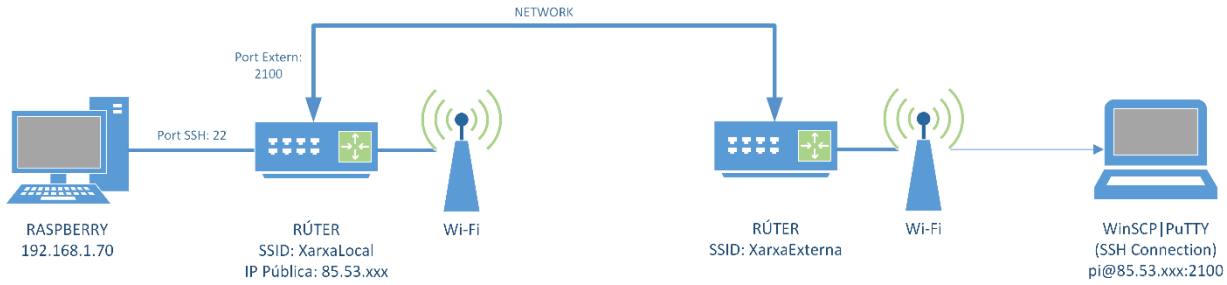
Il·lustració 1: Esquema connexió local amb SSH

Si l'objectiu és que la Raspberry Pi sigui accessible només dins d'una xarxa local, el procediment és el següent:

1. **Connexió Física:** Connecta la Raspberry Pi directament al router mitjançant un cable Ethernet. És important destacar que la imatge del sistema operatiu proporcionada per Alfa Networks desactiva les interfícies Bluetooth i Wi-Fi per garantir el correcte funcionament de la interfície Wi-Fi HaLow, fent que la connexió per cable sigui l'única opció en aquest cas.
2. **Configuració de WinSCP:** Configura WinSCP utilitzant l'adreça IP local de la Raspberry Pi (per exemple, 192.168.1.70) i el port 22, que és el port per defecte per a SSH.



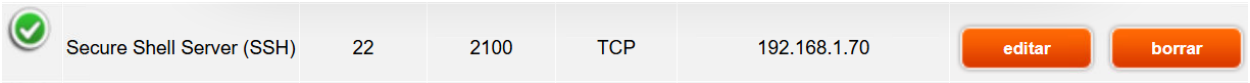
2.1.1.2. Accés des de l'exterior (Xarxa Remota)



Il·lustració 2: Esquema connexió remota amb SSH

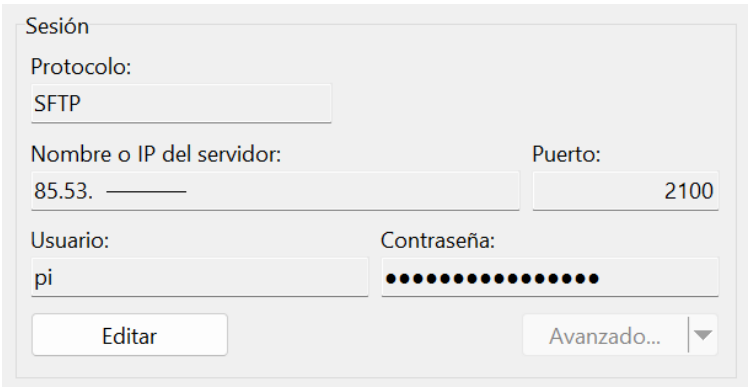
Per accedir a la Raspberry Pi des d'una xarxa externa a la xarxa local, cal configurar el reenviament de ports (port forwarding) al router. Abans d'això, és crucial fixar l'adreça IP de la Raspberry Pi dins de la xarxa local (les indicacions poden variar depenent del model del router):

1. Fixar l'IP de la Raspberry Pi:
   - Accedeix a la configuració del router des de la xarxa local. Normalment, això es fa introduint 192.168.0.1 o 192.168.1.1 al navegador.
   - Busca la secció de configuració avançada de DHCP (el nom pot variar segons el model del router).
   - Localitza la Raspberry Pi per la seva adreça IP dinàmica (per exemple, 192.168.1.70) i converteix-la en estàtica.
2. Configuració del Reenviament de Ports (Port Forwarding):
   - Un cop l'IP de la Raspberry Pi és estàtica, navega a la secció NAT, Port Forwarding o similar del router.
   - Crea una nova regla que permeti l'accés al port 22 de l'adreça IP de la Raspberry Pi (192.168.1.70) des de l'exterior de la xarxa. En aquest pas, haurem d'escollir el port que ens connecta amb l'exterior, per exemple 2100 (port extern)



A l'exemple de la imatge, s'observen el port intern 22 i el port extern 2100. Això significa que es podrà accedir al port 22 de la Raspberry Pi (IP 192.168.1.70) a través del port 2100 de la IP pública del router.

Per connectar-se des de fora, primer cal conèixer la IP pública de la xarxa d'origen. Pots obtenir-la visitant un lloc web com *www.cual-es-mi-ip.net* (connectat al router al que vols accedir) Un cop obtinguda, introdueix-la a WinSCP d'aquesta manera:



Per connectar més dispositius, simplement repeteix aquest procés per a cadascun.

Un problema comú que sorgeix quan es treballa amb el mòdul AHPI7292S en mode STA (i altres modes, excepte AP) és la creació d'una ruta a la taula d'encaminament que té prioritat sobre la interfície eth0. Això provoca que, tot i que la Raspberry Pi sigui localitzable des de l'exterior, no rebi informació de retorn, ja que aquesta s'envia a través de la interfície wlan0 (WiFi HaLow), per tant la connexió amb una xarxa remota no funciona.

Per verificar-ho, executa *ip r* a la terminal de Linux. Veurem una sortida similar a aquesta:

```
default via 192.168.200.1 dev wlan0 proto dhcp src 192.168.200.31 metric 100
default via 192.168.1.1 dev eth0 proto dhcp src 192.168.1.71 metric 202
192.168.1.0/24 dev eth0 proto dhcp scope link src 192.168.1.71 metric 202
192.168.200.0/24 dev wlan0 proto dhcp scope link src 192.168.200.31 metric 100
```

Per resoldre aquest conflicte, cal eliminar la primera ruta per defecte que interfereix amb la ruta eth0. Per fer-ho, executa la següent comanda, substituint els valors pels que apareguin a la teva configuració:

```
sudo ip r del default via <AP_IP> dev wlan0 proto dhcp src <STA_IP> metric 100
```

En essència, per eliminar una ruta, s'utilitza *sudo ip r del* seguit de la descripció exacta de la ruta que es vol eliminar.

2.1.2. Configuració modes d'us AHPI7292S

Dins del codi ./start.py, a la part superior, hi trobem uns requadres de configuració. Aquests serveixen per establir diverses configuracions del mòdul, com ara l'estalvi d'energia, el canal, etc. A continuació, es detallen les configuracions que es poden modificar a ./start.py.

```
# Default Configuration (you can change value you want here)
##############################################################################################
# Raspbery Pi Conf.
max_cpuclock     = 1        # Set Max CPU Clock : 0(off) or 1(on)
##############################################################################################
# Firmware Conf.
model           = 7292     # 7292 or 7192
fw_download      = 1        # 0(FW Download off) or 1(FW Download on)
fw_name          = 'uni_s1g.bin'
##############################################################################################
# DEBUG Conf.
# NRC Driver log
driver_debug     = 0        # NRC Driver debug option : 0(off) or 1(on)
#-----------------------------------------------------------------------#
# WPA Supplicant Log (STA Only)
supplicant_debug  = 0        # WPA Supplicant debug option : 0(off) or 1(on)
#-----------------------------------------------------------------------#
# HOSTAPD Log (AP Only)
hostapd_debug     = 0        # Hostapd debug option    : 0(off) or 1(on)
##############################################################################################
# CSPI Conf. (Default)
spi_clock    = 20000000      # SPI Master Clock Frequency
spi_bus_num  = 0             # SPI Master Bus Number
spi_cs_num   = 0             # SPI Master Chipselect Number
spi_gpio_irq = 5            # NRC-CSPI EIRQ GPIO Number
                    # BBB is 60 recommanded.
spi_polling_interval = 0     # NRC-CSPI Polling Interval (msec)
```

```
#
# NOTE:
#  - NRC-CSPI EIRQ Input Interrupt: spi_gpio_irq >= 0 and spi_polling_interval <= 0
#  - NRC-CSPI EIRQ Input Polling  : spi_gpio_irq >= 0 and spi_polling_interval > 0
#  - NRC-CSPI Registers Polling   : spi_gpio_irq < 0 and spi_polling_interval > 0
#
#-------------------------------------------------------------------------#
# FT232H USB-SPI Conf. (FT232H CSPI Conf)
ft232h_usb_spi = 0          # FTDI FT232H USB-SPI bridge
                        # 0 : Unused
                        # 1 : NRC-CSPI_EIRQ Input Polling
                        # 2 : NRC-CSPI Registers Polling
###########################################################################################
# RF Conf.
# Board Data includes TX Power per MCS and CH
txpwr_val        = 30      # TX Power
txpwr_max_default = 24     # Board Data Max TX Power
bd_download      = 0       # 0(Board Data Download off) or 1(Board Data Download on)
bd_name          = 'nrc7292_bd.dat'
#-------------------------------------------------------------------------#
# Calibration usage option
#  If this value is changed, the device should be restarted for applying the value
cal_use          = 1       # 0(disable) or 1(enable)
###########################################################################################
# PHY Conf.
guard_int        = 'long'   # Guard Interval ('long'(LGI) or 'short'(SGI))
###########################################################################################
# MAC Conf.
# AMPDU (Aggregated MPDU)
#  Enable AMPDU for full channel utilization and throughput enhancement
ampdu_enable     = 1       # 0 (disable) or 1 (enable)
#-------------------------------------------------------------------------#
# 1M NDP (Block) ACK (AP Only)
#  Enable 1M NDP ACK on 2/4MHz BW for robustness (default: 2M NDP ACK on 2/4MH BW)
#  STA should follow, if enabled on AP
#  Note: if enabled, max # of mpdu in ampdu is restricted to 8 (default: max 16)
ndp_ack_1m       = 0       # 0 (disable) or 1 (enable)
#-------------------------------------------------------------------------#
# NDP Probe Request
#  For STA, "scan_ssid=1" in wpa_supplicant's conf should be set to use
ndp_preq         = 0       # 0 (Legacy Probe Req) 1 (NDP Probe Req)
#-------------------------------------------------------------------------#
# CQM (Channel Quality Manager) (STA Only)
#  STA can disconnect according to Channel Quality (Beacon Loss or Poor Signal)
#  Note: if disabled, STA keeps connection regardless of Channel Quality
cqm_enable       = 1       # 0 (disable) or 1 (enable)
#-------------------------------------------------------------------------#
# RELAY (Do NOT use! it will be deprecated)
relay_type       = 1       # 0 (wlan0: STA, wlan1: AP) 1 (wlan0: AP, wlan1: STA)
#-------------------------------------------------------------------------#
# Power Save (STA Only)
#  4-types PS: (0)Always on (1)Modem_Sleep (2)Deep_Sleep(TIM) (3)Deep_Sleep(nonTIM
#  Modem Sleep : turn off only RF while PS (Fast wake-up but less power save)
#   Deep Sleep : turn off almost all power (Slow wake-up but much more power save)
#     TIM Mode : check beacons during PS to receive BU from AP
#  nonTIM Mode : Not check beacons during PS (just wake up by TX or EXT INT)
power_save       = 2       # STA (power save type 0~3)
ps_timeout       = '3s'    # STA (timeout before going to sleep) (min:1000ms)
sleep_duration   = '3s'    # STA (sleep duration only for nonTIM deepsleep) (min:1000ms)
listen_interval  = 1000    # STA (listen interval in BI unit) (max:65535)
#-------------------------------------------------------------------------#
# BSS MAX IDLE PERIOD (aka. keep alive) (AP Only)
#  STA should follow (i.e STA should send any frame before period),if enabled on AP
#  Period is in unit of 1000TU(1024ms, 1TU=1024us)
```

```
#  Note: if disabled, AP removes STAs' info only with explicit disconnection like deauth
bss_max_idle_enable = 1      # 0 (disable) or 1 (enable)
bss_max_idle        = 180    # time interval (e.g. 60: 614400ms) (1 ~ 65535)
#-----------------------------------------------------------------------#
# Mesh Options (Mesh Only)
#  SW encryption by MAC80211 for Mesh Point
sw_enc              = 0     # 0 (disable), 1 (enable)
#  Manual Peering & Static IP
peer                = 0     # 0 (disable) or Peer MAC Address
static_ip           = 0     # 0 (disable) or Static IP Address
#-----------------------------------------------------------------------#
# Self configuration (AP Only)
#  AP scans the clearest CH and then starts with it
self_config       = 0       # 0 (disable)  or 1 (enable)
prefer_bw         = 0       # 0: no preferred bandwidth, 1: 1M, 2: 2M, 4: 4M
dwell_time        = 100     # max dwell is 1000 (ms), min: 10ms, default: 100ms
#-----------------------------------------------------------------------#
# Credit num of AC_BE for flow control between host and target (Internal use only)
credit_ac_be      = 40       # number of buffer (min: 40, max: 120)
##############################################################################
```

Podem modificar el guard interval, si volem que sigui short (4 us) o si volem que sigui long (8 us), el guard interval es el temps que passa en pausa entre símbol i símbol. Per altre banda, també es pot modificar la potencia del senyal transmès, modificant el valor d'aquesta variable txpwr_val que esta en dBm, la máxima potencia que pot oferir el modul HAT es de 30 dBm, per tant qualsevol valor superior a aquest no funcionarà.

Nomes fent referencia a l'estacio STA es poden posar diferents modes d'estalvi d'energia, per exemple 2 seria com si STA fos una estacio TIM, nomes enviant trames en el RAW, també hi ha un altre método el numero 3 que seria una estacio nonTIM, la qual nomes enviaría trames en l'espai PRAW, en aquest cas també es poden

2.1.2.1. Mode AP (Acces Point)

El mode acces point (AP) es el mode a on es conectaran les diferents STA o estacions en el protocol de wifi haLow o IEEE802.11ah. El programari del xip Newracom ens permet modificar algunes configuracions concretes nomes per AP

Configuracions només de AP

SELF CONFIGURATION (Dona problemes de python)

```
# Self configuration (AP Only)
#  AP scans the clearest CH and then starts with it
self_config       = 0       # 0 (disable)  or 1 (enable)
prefer_bw         = 0       # 0: no preferred bandwidth, 1: 1M, 2: 2M, 4: 4M
dwell_time        = 100     # max dwell is 1000 (ms), min: 10ms, default: 100ms
```

El que ens permet aquesta configuració es que l'AP configuri de forma automàtica el canal de comunicacions pel qual s'establirà la comunicació entre AP i STAs. Si activem aquesta configuració (self_config = 1) l'AP escanejarà tots els canals disponibles i escollirà el mes apte per la comunicació, també, podem escollir un ample de banda preferent, es a dir, agafarà preferentment els bandwth especificat , tal i com posa a la llegenda, posem el digitl de la banda que volguem (1M, 2M, 4M) a la variable prefer_bw. Per últim, el dwell_time es el temps que l'AP es quedarà escoltant a cada canal per després escollit el millor, per defecte es de 100ms.

1M NDP (Null data packet)

```
# 1M NDP (Block) ACK (AP Only)
#  Enable 1M NDP ACK on 2/4MHz BW for robustness (default: 2M NDP ACK on 2/4MH BW)
#  STA should follow, if enabled on AP
#  Note: if enabled, max # of mpdu in ampdu is restricted to 8 (default: max 16)
ndp_ack_1m      = 0      # 0 (disable) or 1 (enable)
```

El paràmetre ndp_ack_1m en Wi-Fi HaLow permet que el punt d'accés (AP) utilitzi un mecanisme ACK més robust a 1 Mbps en lloc de la configuració predeterminada de 2 Mbps quan opera en amples de banda de 2 o 4 MHz. Aquesta configuració està dissenyada per millorar la fiabilitat de la connexió en entorns amb senyals febles o interferències, ja que la taxa de dades més baixa fa que l'ACK sigui menys propens a errors. No obstant això, activar-lo té una contrapartida: redueix el nombre màxim de paquets que es poden agrupar en una única transmissió (A-MPDU) de 16 a 8, la qual cosa pot disminuir lleugerament l'eficiència general de la transmissió de dades.

Per últim també tenim una configuració que ens permet depurar la sortida de l'arxiu start, per tant poder observar de forma mes clara quan tenim qualsevol problema, o podem activar d'aquesta manera: (HOSTAPD = 1)

Per altra banda, a part de les configuracions internes del script, a l'hora d'iniciar l'AP, podem escollir diferents paràmetres. Primer de tot el paràmetre de mode de treball del mòdul el fixarem a 1, es a dir a mode AP, després, al ser mode AP, escollim el protocol de seguretat amb el que volem treballar, el qual podem escollir d'entre tots aquests (security_mode [0:Open | 1:WPA2-PSK | 2:WPA3-OWE | 3:WPA3-SAE | 4:WPS-PBC]), per últim hem d'especificar el país on ens trobem i per tant el qual funcionarà el mòdul AHPI7292S d'acord amb les normes d'aquest país.

Un exemple d'execusió del codi start.py seria el següent:

$ ./start.py 1 0 EU

Es a dir, iniciem el mòdul en mode AP, sense protocol de seguretat associat (Open) i estem a la unió europea, per tant només podrà accedir a canals de fins a 2 MHz.

Exemple de linea de comandos

```
pi@raspberrypi:~/nrc_pkg/script $ ./start.py 1 0 EU
Done.
Done.
------------------------------
Model            : 7292
STA Type         : AP
Country          : EU
Security Mode    : OPEN
CAL. USE         : OFF
AMPDU            : ON
Download FW      : uni_s1g.bin
TX Power         : 30
------------------------------
NRC AP setting for HaLow...
[*] Set Max CPU Clock on RPi
1400000
1400000
1400000
1400000
Done
[0] Clear
hostapd: no process found
wireshark-gtk: no process found
rmmod: ERROR: Module nrc is not currently loaded
```

```
rm: cannot remove '/home/pi/nrc_pkg/script/conf/temp_self_config.conf': No such file
or directory
[1] Copy
total 336
drwxr-xr-x 2 pi pi   4096 Apr 18  2022 .
drwxr-xr-x 4 pi pi   4096 Apr 18  2022 ..
-rwxr-xr-x 1 pi pi    342 Apr 14  2022 copy
-rwxr-xr-x 1 pi pi    562 Jan  8  2022 nrc7292_bd.dat
-rwxr-xr-x 1 pi pi 325628 Apr 14  2022 nrc7292_cspi.bin
-rwxr-xr-x 1 root root 325628 Apr 15  2022 /lib/firmware/uni_s1g.bin
====================================
 AP INTERFACE     : wlan0
 AP STATIC IP     : 192.168.200.1
 NET MASK NUM     : 24
====================================
Config for AP is done!
IP and DHCP config done
[2] Set Module Parameters
Error: AP interface does not support power save: 2
[3] Loading module
sudo modprobe nrc  hifspeed=20000000 spi_bus_num=0 spi_cs_num=0 spi_gpio_irq=5
spi_polling_interval=0 fw_name=uni_s1g.bin power_save=0 auto_ba=1
listen_interval=1000 credit_ac_be=40
[4] Set tx power
FAIL
Board Data use                  : off
OK
[5] Set guard interval
guard interval : long
OK
[6] Set cal_use
Calibration_use : off                   Country  : US
OK
[*] Start DHCPCD and DNSMASQ
[*] Self configuration off
[6] Start hostapd on wlan0
Configuration file: /home/pi/nrc_pkg/script/conf/EU/ap_halow_open.conf
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
Using interface wlan0 with hwaddr 00:c0:ca:b6:89:cb and ssid "halow_demo"
wlan0: interface state COUNTRY_UPDATE->ENABLED
wlan0: AP-ENABLED
[7] Start NAT
[8] ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.71  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a5a2:f0a8:1d28:20af  prefixlen 64  scopeid 0x20<link>
        ether d8:3a:dd:0d:b4:df  txqueuelen 1000  (Ethernet)
        RX packets 92082  bytes 6260047 (5.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12743  bytes 580014 (566.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 681  bytes 51149 (49.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 681  bytes 51149 (49.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.1  netmask 255.255.255.0  broadcast 192.168.200.255
        inet6 fe80::b3a2:928a:772:2c83  prefixlen 64  scopeid 0x20<link>
        ether 00:c0:ca:b6:89:cb  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 19  bytes 2609 (2.5 KiB)
```

```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

HaLow AP ready
------------------------------------------------------------------
Done.
```

Si fem la comanda ifconfig tenim això

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.200.1  netmask 255.255.255.0  broadcast 192.168.200.255
        inet6 fe80::b3a2:928a:772:2c83  prefixlen 64  scopeid 0x20<link>
        ether 00:c0:ca:b6:89:cb  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 257  bytes 34293 (33.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Si fem iwconfig

```
wlan0     IEEE 802.11  Mode:Master  Tx-Power=23 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
```

2.1.2.2. Mode STA (Station)

./start 0 0 EU

Exemple de sortida (Linea de comandos)

Que passa si surt waiting for IP

TIPUS DE MODES TIM, NON TIM

2.1.2.3. Mode Sniffer

El mode Sniffer ens serveix per poder observar els paqueshow ttemp

Configuració amb el canal 41, 42

Maneres de veure wireshark

./start 2 0 EU 41 0

Que es mode remot

Que es mode local

Exemple sortida linea de comandos

Quin canal s'ha de posar

Problemes amb la visualització del canal a 1 MHz

./change chanel (Crec que hi ha problemes)

2.1.2.4. Mode Relay

Sentència start

Problemes de coneció que hi ha

2.1.3. FAQ

Problemes en general que van sorgint

2.2. Proves amb AHPI7292S

2.2.1. Modificació de configuracions amb AHPI7292S

Configuracions canal fitxer EU/ap_config

COnfuguracions Start

2.2.2. Configuracions iPerf3 per la simulació de trànsit

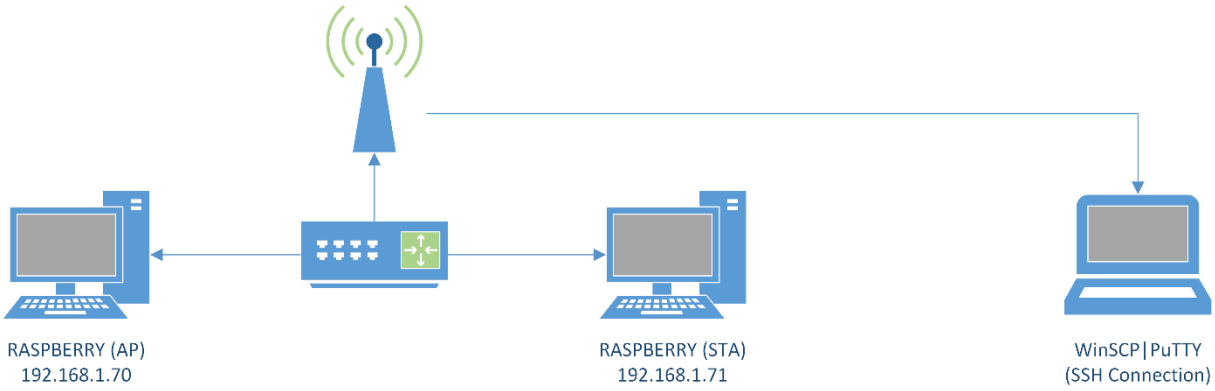
Instal·lació de iperf

Una mica de resum memòria iperf toni

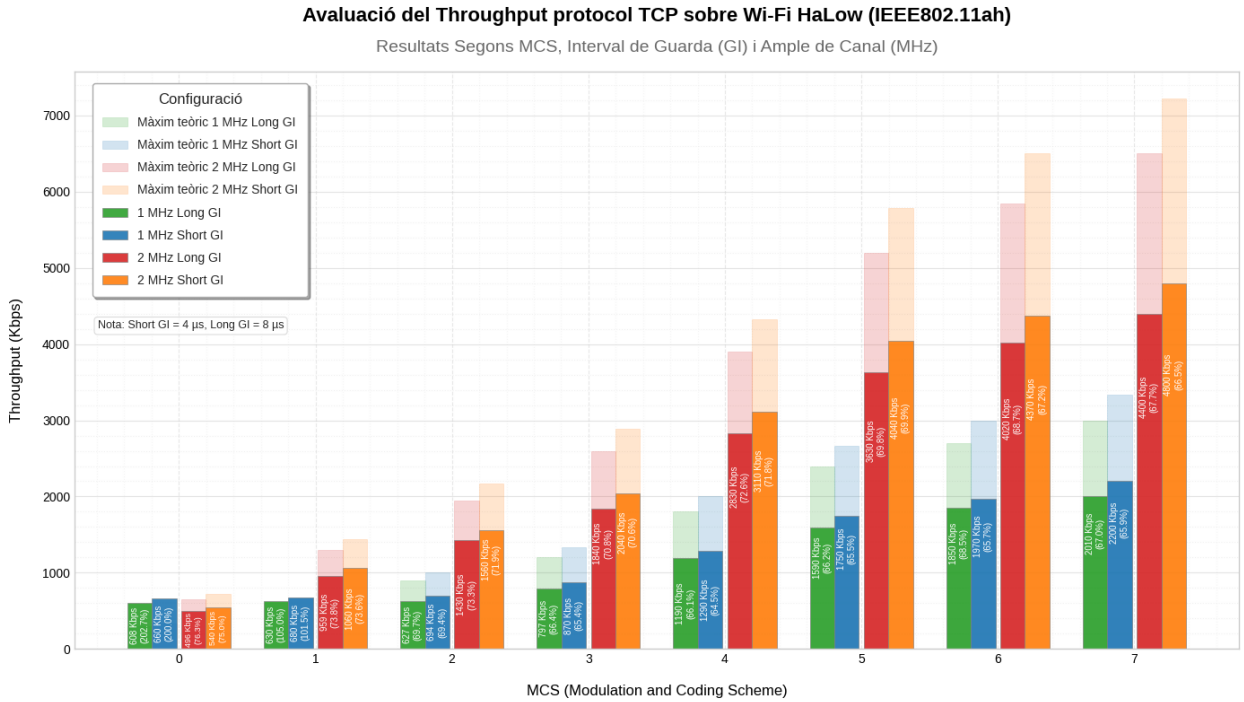
2.2.3. Automatització recull de dades

Script explicació i funcionalitats

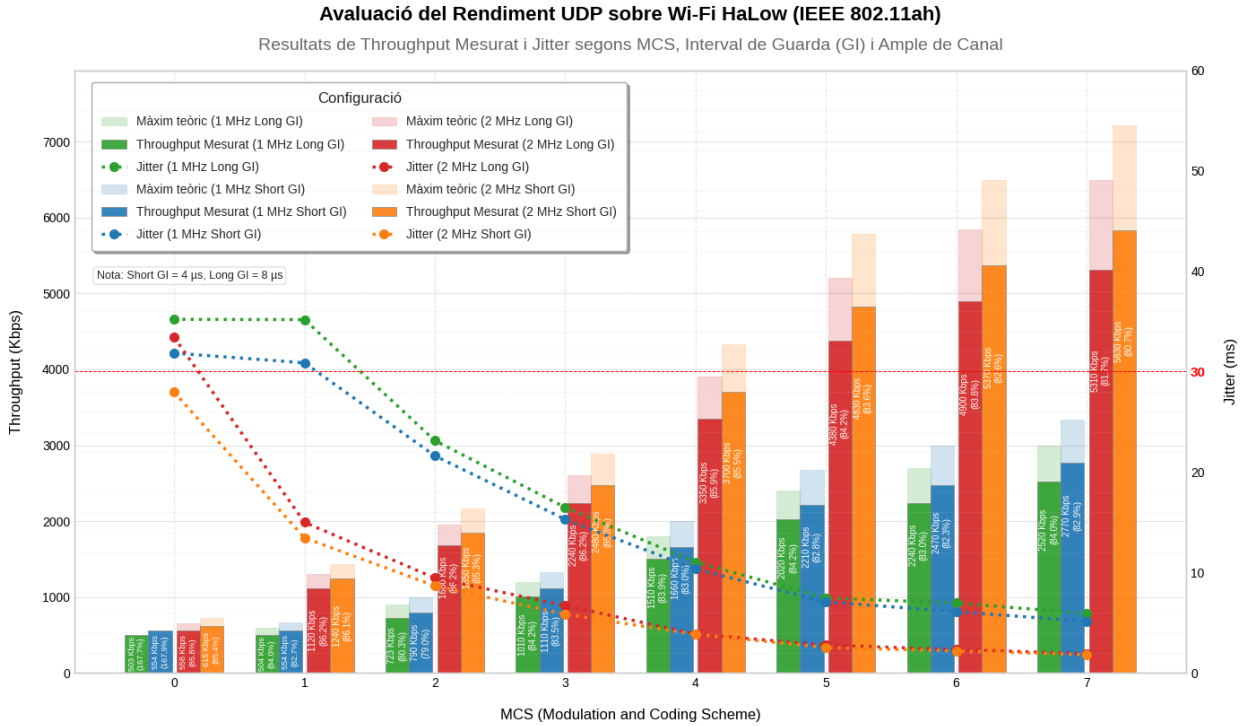
2.2.4. Resultats (AP – STA)



2.2.4.1. Resultats protocol TCP



2.2.4.2. Resultats protocol UDP



2.2.4.3. Resultats variant la distància

Tema throughput en mode Relay

2.2.4.4. Resultats variant el mode d'estalvi d'energia

Sembla que no canvien, potser variant el tema de la fusió de paquets

- Canviant potència tampoc varia, miraré el rang canviant potència a veure si canvia

2.2.5. Resultats (AP – RELAY – STA)

(Diagrama muntatge, taula de variables controlades, etc.)

2.2.5.1. Resultats protocol TCP

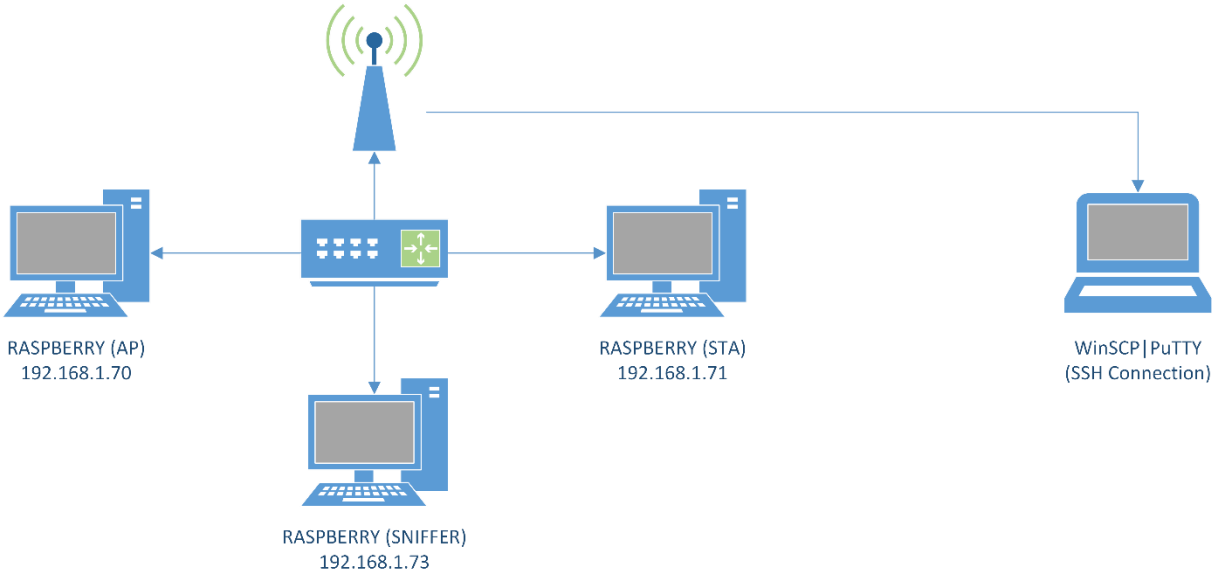
2.2.5.2. Resultats protocol UDP

2.2.5.3. Resultats variant la distància

2.2.5.4. Resultats variant el mode d'estalvi d'energia

2.3. Interpretació de trames capturades amb WireShark

Diagrama captura trames de la capa física entre AP – STA



2.3.1. Inici AP

Context: Iniciem AP amb la comanda start ./start.py 1 0 EU i surten les següents trames abans d'iniciar cap estació STA, per tant, només agafa el primer tros després d'inicial l'AP

Foto: CNX_AP_reset.pcap

- Configuració: BEACON INTERVAL es el tems entre Beacons TIM

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Alfa_b6:8a:17	Broadcast	802.11	64	Deauthentication, SN=0, FN=0, Flags=........C
2	0.033965	Alfa_b6:8a:17		802.11	135	S1G Beacon, Flags=....R...C, SSID="halow_demo"
3	0.129959	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
4	0.231966	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
5	0.334971	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
6	0.436976	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
7	0.538978	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
8	0.641979	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
9	0.743983	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
10	0.846986	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
11	0.948989	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
12	1.054992	Alfa_b6:8a:17		802.11	135	S1G Beacon, Flags=....R...C, SSID="halow_demo"
13	1.153995	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
14	1.256007	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
15	1.359002	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
16	1.461003	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
17	1.563010	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
18	1.666011	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
19	1.768014	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
20	1.871019	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
21	1.973022	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
22	2.079025	Alfa_b6:8a:17		802.11	135	S1G Beacon, Flags=....R...C, SSID="halow_demo"
23	2.178029	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
24	2.280030	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
25	2.383034	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
26	2.485035	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
27	2.587043	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
28	2.690047	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
29	2.792073	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
30	2.895054	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
31	2.997072	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
32	3.103059	Alfa_b6:8a:17		802.11	135	S1G Beacon, Flags=....R...C, SSID="halow_demo"

Coses importants a tenir en compte:

1. La trama nº 1

No.	Time	Source	Destination	Protocol	Bytes	Informació
1	0.000000	Alfa_b6:8a:17	Broadcast	802.11	64	Deauthentication, SN=0, FN=0, Flags=.....C

Aquesta trama sempre s'envia quan comença la conexio d'un AP, com una mena de reset.

2. La trama nº 2

No.	Time	Source	Destination	Protocol	Bytes	Informació
2	0.055959	Alfa_b6:8a:17		802.11	135	S1G Beacon, Flags=....R...C, SSID="halow_demo"

Es una trama DTIM i s'envia periòdicament cada 9 TIMs intermitjos mes el TIM que porta incorporat el DTIM

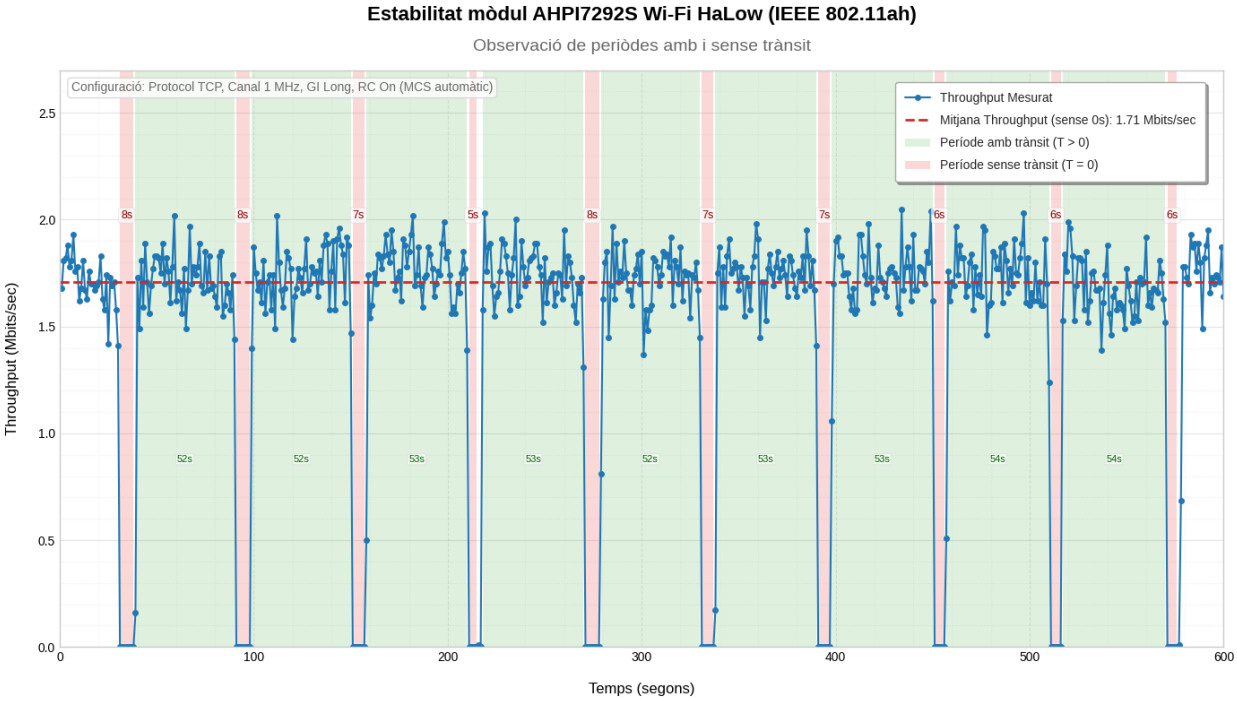
Això perquè s'ha configurat així a la configuració, es adir, hi ha una cosa que posa DTIM interval que es lo que fa que s'enviin els DTIM periodicamtne, no es amb els grups que hi ha

2.3.2. Autenticació i handshake amb l'estació

2.3.3. Throughput 0 bps periòdic (Trames wireshark)

Durant les proves de throughput amb diferents paràmetres (com ara MCS i GI), apartat 2.2, vam detectar una caiguda a 0 del rendiment entre el punt d'accés (AP) i l'estació (STA). Aquest fenomen no era un problema de maquinari aïllat, ja que es produïa de forma periòdica durant uns 7 segons, cada 60 segons, independentment del trànsit generat. Aquesta interrupció del trànsit es correlaciona directament amb l'enviament de certes trames de control, com ARP, ICMP, mDNS i IGMP, com es pot observar a les imatges del wireshark figura x

Aquesta figura mostra clarament que cada 52 segons (període entre la finalització d'una caiguda i l'inici de la següent) es produeix un nou període de 0 throughput, que coincideix amb l'enviament d'aquestes trames. Això ens porta a preguntar-nos quins són els motius darrere d'aquestes trames de control i per què interrompen temporalment el trànsit de dades.

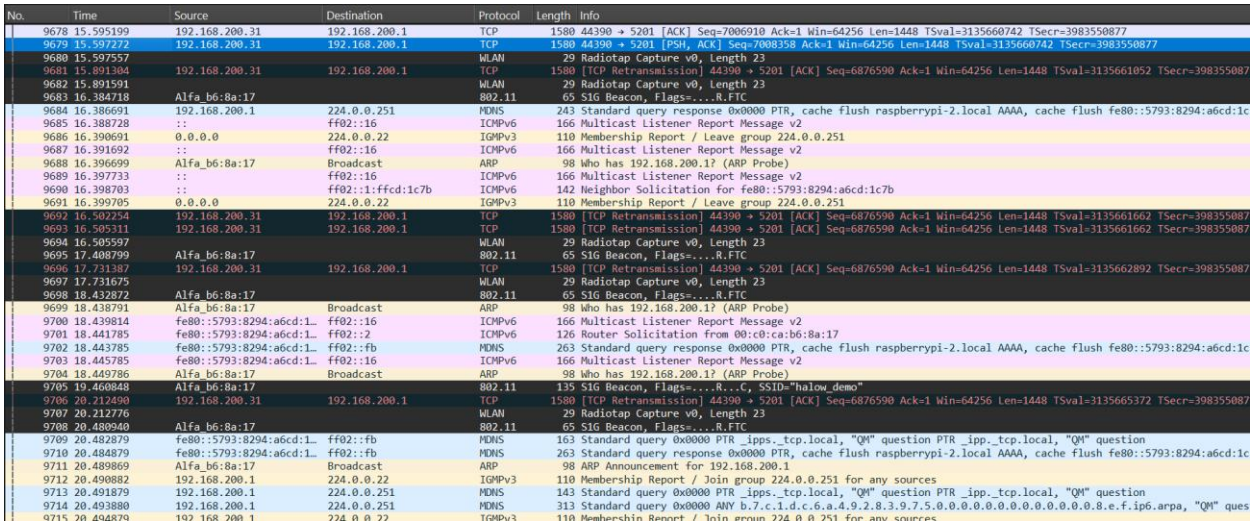


Les trames que observes cada 60 segons, com ara mDNS, ICMPv6, IGMPv3 i ARP, formen part del manteniment rutinari de la xarxa. Aquestes trames són fonamentals per assegurar-ne la connectivitat i el bon funcionament, i coexisteixen amb qualsevol trànsit d'aplicació, com el que es genera amb iperf3.

L'interval d'aproximadament un minut és un compromís ideal entre mantenir la xarxa actualitzada i evitar-ne una sobrecàrrega. Si un dispositiu canvia, s'apaga o hi ha algun canvi en la topologia, aquestes trames periòdiques asseguren que els altres dispositius se n'assabentin ràpidament. Aquest valor predeterminat és comú en molts protocols de xarxa per a tasques de manteniment.

A continuació, es detallen les funcions de cadascun d'aquests protocols:

- mDNS (Multicast DNS): Permet que els dispositius es trobin i es comuniquin per nom en una xarxa local sense necessitat d'un servidor DNS centralitzat.
- ICMPv6 (Internet Control Message Protocol for IPv6): S'utilitza per a la gestió d'errors i per al "descobriment de veïns" (Neighbor Discovery Protocol), que és l'equivalent de l'ARP per a IPv6.
- IGMPv3 (Internet Group Management Protocol): Aquest protocol serveix per subscriure's i donar-se de baixa de grups de trànsit multicast, que és un trànsit de dades que va a diversos destinataris alhora.
- ARP (Address Resolution Protocol): Tradueix adreces IP a adreces MAC en xarxes IPv4. Les seves peticions periòdiques ajuden a mantenir la taula d'adreces actualitzada.



No.	Time	Source	Destination	Protocol	Length	Info
9678	15.595199	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=7006910 Ack=1 Win=64256 Len=1448 TSval=3135660742 TSecr=3983550877
9679	15.597272	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [PSH, ACK] Seq=7008358 Ack=1 Win=64256 Len=1448 TSval=3135660742 TSecr=3983550877
9680	15.597557			WLAN	29	Radiotap Capture v0, Length 23
9681	15.891304	192.168.200.31	192.168.200.1	TCP	1580	[TCP Retransmission] 44390 → 5201 [ACK] Seq=6876590 Ack=1 Win=64256 Len=1448 TSval=3135661052 TSecr=398355087
9682	15.891591			WLAN	29	Radiotap Capture v0, Length 23
9683	16.384718	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
9684	16.386691	192.168.200.1	224.0.0.251	MDNS	243	Standard query response 0x0000 PTR, cache flush raspberrypi-2.local AAAA, cache flush fe80::5793:8294:a6cd:1c
9685	16.388728	::	ff02::16	ICMPv6	166	Multicast Listener Report Message v2
9686	16.390691	0.0.0.0	224.0.0.22	IGMPv3	110	Membership Report / Leave group 224.0.0.251
9687	16.391692	::	ff02::16	ICMPv6	166	Multicast Listener Report Message v2
9688	16.396699	Alfa_b6:8a:17	Broadcast	ARP	98	Who has 192.168.200.1? (ARP Probe)
9689	16.397733	::	ff02::16	ICMPv6	166	Multicast Listener Report Message v2
9690	16.398703	::	ff02::1:ffcd:1c7b	ICMPv6	142	Neighbor Solicitation for fe80::5793:8294:a6cd:1c7b
9691	16.399705	0.0.0.0	224.0.0.22	IGMPv3	110	Membership Report / Leave group 224.0.0.251
9692	16.502254	192.168.200.31	192.168.200.1	TCP	1580	[TCP Retransmission] 44390 → 5201 [ACK] Seq=6876590 Ack=1 Win=64256 Len=1448 TSval=3135661662 TSecr=398355087
9693	16.505311	192.168.200.31	192.168.200.1	TCP	1580	[TCP Retransmission] 44390 → 5201 [ACK] Seq=6876590 Ack=1 Win=64256 Len=1448 TSval=3135661662 TSecr=398355087
9694	16.505597			WLAN	29	Radiotap Capture v0, Length 23
9695	17.408799	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
9696	17.731387	192.168.200.31	192.168.200.1	TCP	1580	[TCP Retransmission] 44390 → 5201 [ACK] Seq=6876590 Ack=1 Win=64256 Len=1448 TSval=3135662892 TSecr=398355087
9697	17.731675			WLAN	29	Radiotap Capture v0, Length 23
9698	18.432872	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
9699	18.438791	Alfa_b6:8a:17	Broadcast	ARP	98	Who has 192.168.200.1? (ARP Probe)
9700	18.439814	fe80::5793:8294:a6cd:1…	ff02::16	ICMPv6	166	Multicast Listener Report Message v2
9701	18.441785	fe80::5793:8294:a6cd:1…	ff02::2	ICMPv6	126	Router Solicitation from 00:c0:ca:b6:8a:17
9702	18.443785	fe80::5793:8294:a6cd:1…	ff02::fb	MDNS	263	Standard query response 0x0000 PTR, cache flush raspberrypi-2.local AAAA, cache flush fe80::5793:8294:a6cd:1c
9703	18.445785	fe80::5793:8294:a6cd:1…	ff02::16	ICMPv6	166	Multicast Listener Report Message v2
9704	18.449786	Alfa_b6:8a:17	Broadcast	ARP	98	Who has 192.168.200.1? (ARP Probe)
9705	19.460848	Alfa_b6:8a:17		802.11	135	S1G Beacon, Flags=....R...C, SSID="halow_demo"
9706	20.212490	192.168.200.31	192.168.200.1	TCP	1580	[TCP Retransmission] 44390 → 5201 [ACK] Seq=6876590 Ack=1 Win=64256 Len=1448 TSval=3135665372 TSecr=398355087
9707	20.212776			WLAN	29	Radiotap Capture v0, Length 23
9708	20.480940	Alfa_b6:8a:17		802.11	65	S1G Beacon, Flags=....R.FTC
9709	20.482879	fe80::5793:8294:a6cd:1…	ff02::fb	MDNS	163	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
9710	20.484879	fe80::5793:8294:a6cd:1…	ff02::fb	MDNS	263	Standard query response 0x0000 PTR, cache flush raspberrypi-2.local AAAA, cache flush fe80::5793:8294:a6cd:1c
9711	20.489869	Alfa_b6:8a:17	Broadcast	ARP	98	ARP Announcement for 192.168.200.1
9712	20.490882	192.168.200.1	224.0.0.22	IGMPv3	110	Membership Report / Join group 224.0.0.251 for any sources
9713	20.491879	192.168.200.1	224.0.0.251	MDNS	143	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
9714	20.493880	192.168.200.1	224.0.0.251	MDNS	313	Standard query 0x0000 ANY b.7.c.1.d.c.6.a.4.9.2.8.3.9.7.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa, "QM" ques
9715	20.494879	192.168.200.1	224.0.0.22	IGMPv3	110	Membership Report / Join group 224.0.0.251 for any sources

A continuació, una descripció de les trames que s'observen a la captura i l'ordre en què apareixen:

Trànsit Normal (TCP): Al principi i final de la captura, es veu trànsit TCP. Això és el trànsit d'iperf3. La majoria d'aquest trànsit és entre l'adreça IP 192.168.200.31 i 192.168.200.1. On 192.168.200.31 es la STA, es a dir el que envia les trames i 192.168.200.1 es la AP i rep les trames i envia ACK cap a .31.

Trames de Manteniment :

IGMPv3 - Aquesta és una trama "Host Membership Report" (informe de pertinença de l'amfitrió) per a un grup de multicast. La direcció 224.0.0.251 és una adreça de multicast.

MDNS - Una trama "Standard query" (consulta estàndard) que busca el nom d'un host i altres serveis a la xarxa.

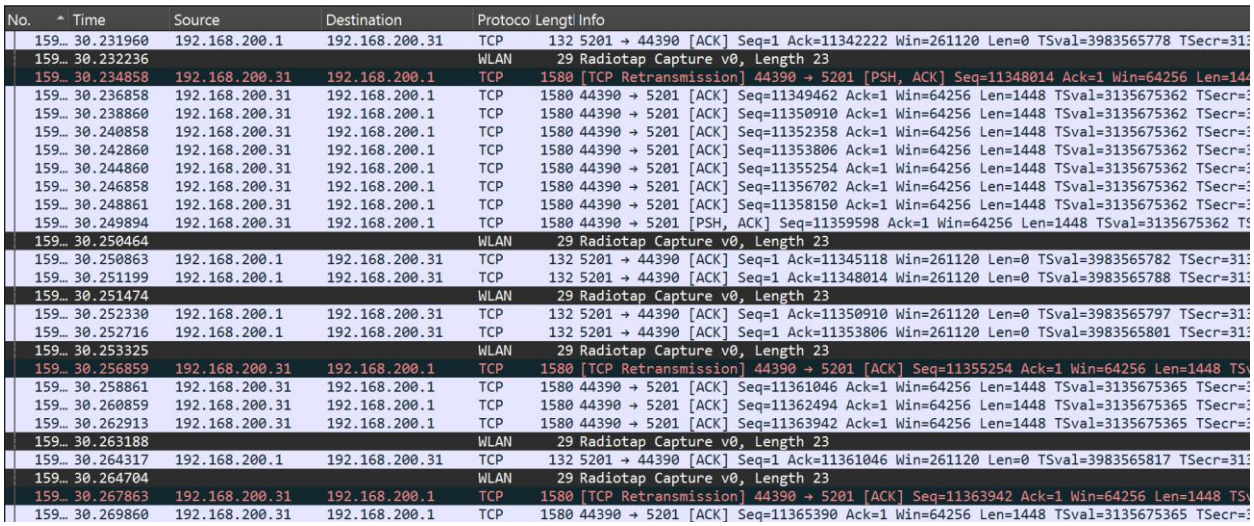
IGMPv3 - Una altra trama "Host Membership Report" per a un grup de multicast diferent (224.0.0.252).

ARP - Una trama "ARP Announcement" (anunci d'ARP) des de l'adreça 192.168.200.31. Aquesta trama anuncia que aquesta adreça IP pertany a la MAC f6:b6:6e:17.

ICMPv6 - Una trama "Neighbor Solicitation" (sol·licitud de veí) que busca l'adreça MAC d'un altre host IPv6.

Es pot veure clarament que les trames de manteniment (MDNS, IGMP, ARP, ICMPv6) s'envien en ràpids successos, i aquests blocs de comunicació es repeteixen de manera gairebé idèntica a intervals d'aproximadament un minut. Aquest patró és completament normal per al funcionament d'un dispositiu en una xarxa IP moderna i concretament amb xarxes IoT.

2.3.4. Transmissió dades mitjançant TCP



No.	Time	Source	Destination	Protocol	Length	Info
159…	30.231960	192.168.200.1	192.168.200.31	TCP	132	5201 → 44390 [ACK] Seq=1 Ack=11342222 Win=261120 Len=0 TSval=3983565778 TSecr=313
159…	30.232236			WLAN	29	Radiotap Capture v0, Length 23
159…	30.234858	192.168.200.31	192.168.200.1	TCP	1580	[TCP Retransmission] 44390 → 5201 [PSH, ACK] Seq=11348014 Ack=1 Win=64256 Len=144
159…	30.236858	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11349462 Ack=1 Win=64256 Len=1448 TSval=3135675362 TSecr=3
159…	30.238860	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11350910 Ack=1 Win=64256 Len=1448 TSval=3135675362 TSecr=3
159…	30.240858	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11352358 Ack=1 Win=64256 Len=1448 TSval=3135675362 TSecr=3
159…	30.242860	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11353806 Ack=1 Win=64256 Len=1448 TSval=3135675362 TSecr=3
159…	30.244860	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11355254 Ack=1 Win=64256 Len=1448 TSval=3135675362 TSecr=3
159…	30.246858	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11356702 Ack=1 Win=64256 Len=1448 TSval=3135675362 TSecr=3
159…	30.248861	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11358150 Ack=1 Win=64256 Len=1448 TSval=3135675362 TSecr=3
159…	30.249894	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [PSH, ACK] Seq=11359598 Ack=1 Win=64256 Len=1448 TSval=3135675362 TS
159…	30.250464			WLAN	29	Radiotap Capture v0, Length 23
159…	30.250863	192.168.200.1	192.168.200.31	TCP	132	5201 → 44390 [ACK] Seq=1 Ack=11345118 Win=261120 Len=0 TSval=3983565782 TSecr=313
159…	30.251199	192.168.200.1	192.168.200.31	TCP	132	5201 → 44390 [ACK] Seq=1 Ack=11348014 Win=261120 Len=0 TSval=3983565788 TSecr=313
159…	30.251474			WLAN	29	Radiotap Capture v0, Length 23
159…	30.252330	192.168.200.1	192.168.200.31	TCP	132	5201 → 44390 [ACK] Seq=1 Ack=11350910 Win=261120 Len=0 TSval=3983565797 TSecr=313
159…	30.252716	192.168.200.1	192.168.200.31	TCP	132	5201 → 44390 [ACK] Seq=1 Ack=11353806 Win=261120 Len=0 TSval=3983565801 TSecr=313
159…	30.253325			WLAN	29	Radiotap Capture v0, Length 23
159…	30.256859	192.168.200.31	192.168.200.1	TCP	1580	[TCP Retransmission] 44390 → 5201 [ACK] Seq=11355254 Ack=1 Win=64256 Len=1448 TSv
159…	30.258861	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11361046 Ack=1 Win=64256 Len=1448 TSval=3135675365 TSecr=3
159…	30.260859	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11362494 Ack=1 Win=64256 Len=1448 TSval=3135675365 TSecr=3
159…	30.262913	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11363942 Ack=1 Win=64256 Len=1448 TSval=3135675365 TSecr=3
159…	30.263188			WLAN	29	Radiotap Capture v0, Length 23
159…	30.264317	192.168.200.1	192.168.200.31	TCP	132	5201 → 44390 [ACK] Seq=1 Ack=11361046 Win=261120 Len=0 TSval=3983565817 TSecr=313
159…	30.264704			WLAN	29	Radiotap Capture v0, Length 23
159…	30.267863	192.168.200.31	192.168.200.1	TCP	1580	[TCP Retransmission] 44390 → 5201 [ACK] Seq=11363942 Ack=1 Win=64256 Len=1448 TSv
159…	30.269860	192.168.200.31	192.168.200.1	TCP	1580	44390 → 5201 [ACK] Seq=11365390 Ack=1 Win=64256 Len=1448 TSval=3135675365 TSecr=3

Les trames WLAN les afegeix Wireshark com a meta informació de la capa física que posa wireshark per poder observar diferentes dades

Podem veure també trames TCP PSH les quals tenen una longitud de 1580 bytes i després tenim els ACK amb una de 132, veient això també es veu clarament que qui envia la informació es la IP (192.168.200.31) i qui la rep i pertant respon amb ACK es la (192.168.200.1) com que estem fent uplink son STA i AP respectivament.

Per últim podem veure trames TCP de retransmissió, paquets que no s'ha rebut correctament i es tornen a enviar despres d'un timeout.

3. Conclusions

4. Referències

https://ahfibit.com/my-first-802-11ah-frames/

https://ahfibit.com/halow-association-explained/

https://ahfibit.com/halow-beacon-frame-explained/

5. Tables

5.1. Table 1

Field 1	Field 2	Field 3	Field 4
Content 1	Information		
Content 2			
Content 3			

5.2. Table 2

	Field 1	Field 2	Field 3	Field 4

	Content 1	Information		
	Content 2			
	Content 3			

6. Next section