Implementación de una red LoRa en el ámbito de la Universidad Nacional de San Martín

Lucas Bullian - Cristian Urbina

Lucas Bullian, Cristian Urbina: *Implementación de una red LoRa en el ámbito de la Universidad Nacional de San Martín,*
Proyecto Final Ingeniería en Telecomunicaciones, © Julio 2018

tutor:
Ing Marcelo E. Romeo

Índice

1. Introducción

El presente proyecto se basa en la instalación de una red con tecnología LoRa[1] para el ámbito de la Universidad Nacional de San Martín. Surge de la necesidad de realizar comunicaciones inalámbricas para diversos proyectos en los cuales no se dispone de una conexión WiFi o ésta no tiene el alcance esperado. este proyecto es parte de un proyecto mayor el cual tiene como objetivo el control, monitoreo y medición remota del consumo de un aire acondicionado y permitir actuar sobre él, apagandolo.

Aquí presentaremos todos los pasos realizados para la implementación como también posibles mejoras y aplicaciones de esta red para futuros usos.

2. Objetivos

Objetivo general:

- Implementación de una red LoRa en el ámbito de la Universidad Nacional de San Martín para el monitoreo de datos.

Objetivos específicos:

- Determinar los requerimientos tecnológicos que sean necesarios para la implementación del sistema de medición.
- Realizar la implementación de los módulos tecnológicos que sean necesarios para la implementación del sistema de medición.
- Llevar a cabo pruebas de rendimiento y alcance de la red para asegurar la cobertura de gran parte del campus miguelete.
- Realizar la infraestructura tecnológica para el acceso y lectura de los datos medidos.

3. Desarrollo

3.1. Teorema Shannon-Hartley

En teoría de la información, el teorema Shannon-Hartley[2] establece la tasa de transmisión máxima de información que se puede transmitir sobre un canal de comunicación de un determinado ancho de banda en presencia de ruido:

$$C = B * log_2 (1 + \frac{S}{N}) \qquad \textbf{Ecuación 3.1}$$

Donde:

C = Capacidad del canal (bit/s)
B = Ancho de banda del canal (Hz)
S = Señal recibida promedio (Watts)
N = Ruido recibido promedio (Watts)
S/N = Relación señal a ruido (SNR)

Reacomodando la ecuación 3.1 del Logaritmo en base 2 al natural, e, y notando que ln = $\log_e$ nos quedaría:

$$\frac{C}{B} = 1.433 * \frac{S}{N} \qquad \textbf{Ecuación 3.2}$$

Para aplicaciones de "Spread Spectrum" la relación señal a ruido es pequeña, ya que normalmente la señal está por debajo del nivel de ruido. Asumiendo un nivel de ruido S/N << 1, la ecuación 3.2 queda:

$$\frac{C}{B} \approx \frac{S}{N} \quad ó \quad \frac{N}{S} \approx \frac{B}{C} \qquad \textbf{Ecuación 3.3}$$

De esta ecuación podemos notar que para transmitir una señal libre de errores en un canal con relación señal a ruido fija, solo debemos aumentar el ancho de banda.

3.2. Spread Spectrum

Como vimos anteriormente, aumentando el ancho de banda de la señal podemos compensar la degradación de la relación señal a ruido de un canal inalámbrico.

En sistemas "Direct Sequence Spread Spectrum" (DSSS), la fase de la portadora del transmisor cambia de acuerdo a un código. Este proceso se consigue generalmente multiplicando la señal de datos con un código de expansión o secuencia de chips. Esta secuencia de chips ocurre a una velocidad mucho más alta que la señal y por lo tanto expande la señal utilizando un mayor ancho de banda que el original. Chip se denomina a bits codificados más cortos que los de la información original.

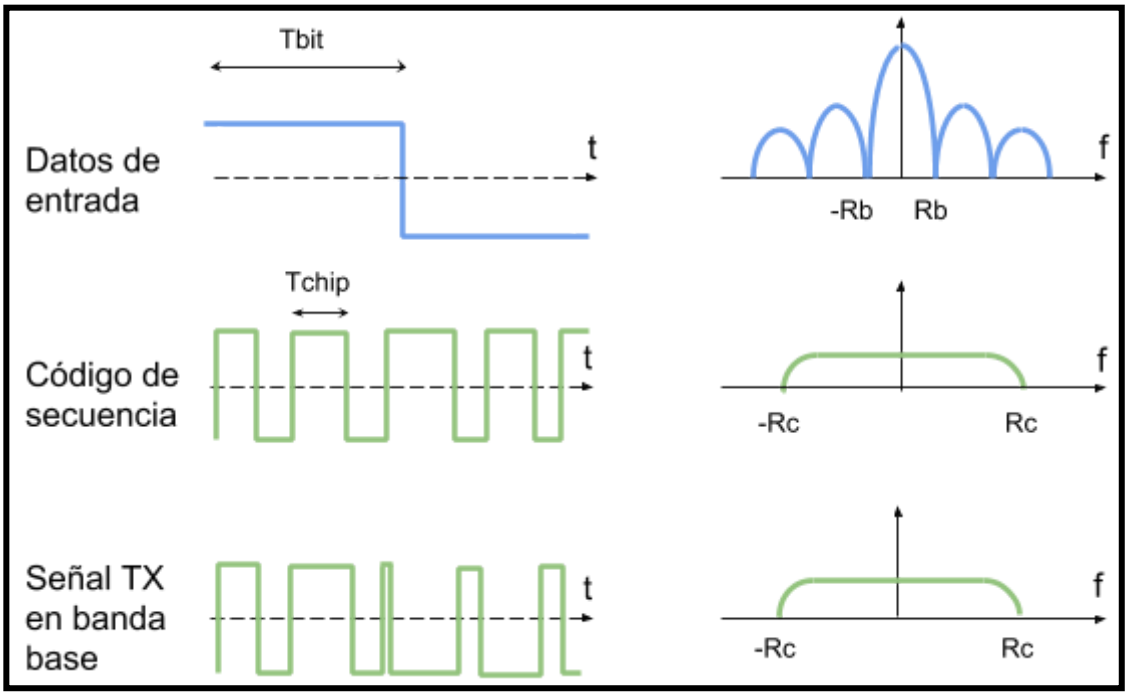


Figura 3.2.1. Modulación / Proceso de expansión

En el receptor, la señal buscada se recupera realizando una nueva multiplicación con una réplica local del código de expansión. Esta multiplicación realiza el proceso inverso en el ancho de banda y la señal recupera su ancho original como se ve en la figura 3.2.2. El código de expansión debe ser el mismo en ambos extremos para la correcta transmisión de la información.

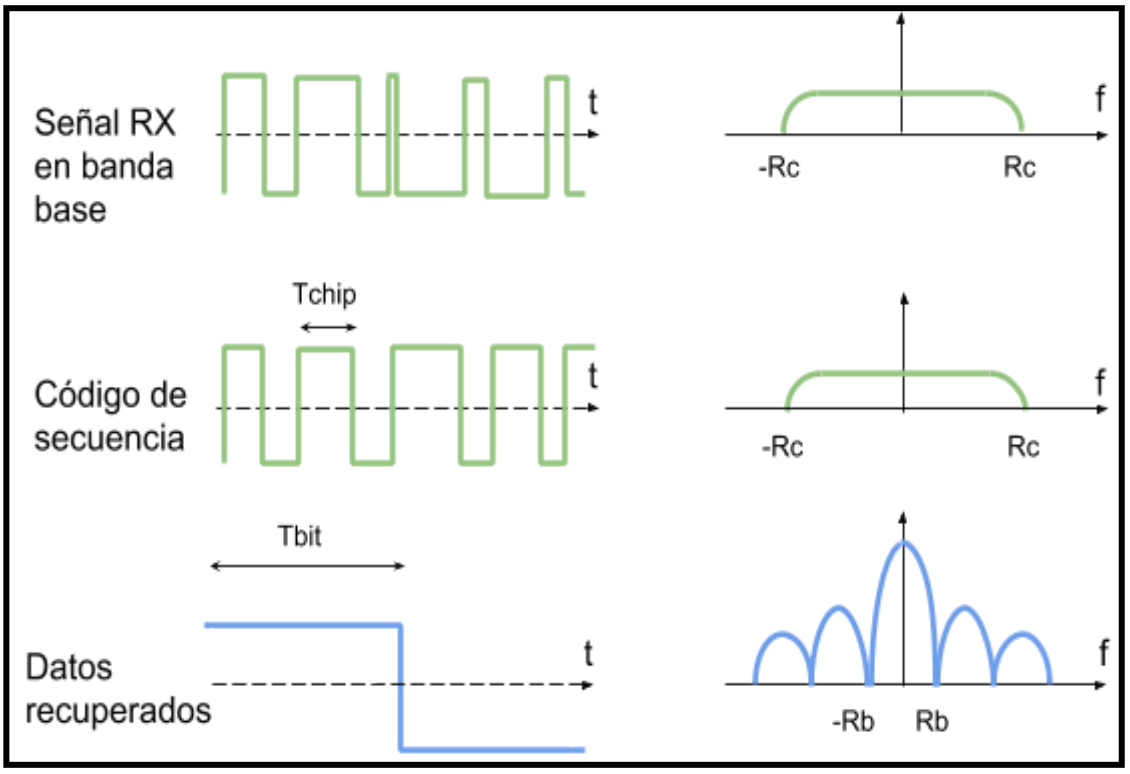


Figura 3.2.2. Demodulación / Proceso de contracción

El tamaño de expansión, para DSSS, depende de la relación chips por bit. La relación entre la secuencia de chips y la tasa de transmisión se denomina ganancia de proceso (G_P) expresada en dB.

$$Gp = 10 * log_{10}\left(\frac{R_C}{Rb}\right)(dB) \qquad \textbf{Ecuación 3.4}$$

Donde:

R_C = Tasa de chips (Chips/Segundo)
R_b = Tasa de bits (bits/Segundo)

Las interferencias son reducidas por la ganancia de procesamiento del receptor. Se extienden más allá del ancho de banda de la información y son fácilmente filtradas.

DSSS es usado extensamente en aplicaciones de comunicación. Sin embargo, existen inconvenientes para dispositivos y redes de bajo costo o con alimentación limitada.

Frecuentemente, como en el caso de GPS o PHY DSSS del estándar 802.15.4K[3] del IEEE, el sistema requiere un reloj muy preciso y costoso. Además, cuanto más extenso es el código de expansión, más largo es el tiempo requerido por el receptor para realizar la correlación de la secuencia.

Esto es un problema para los dispositivos de alimentación limitada, como mencionamos anteriormente, que no pueden estar todo el tiempo encendidos y necesitan realizar la sincronización reiteradas veces.

3.2.1. Chirp Spread Spectrum

Esta tecnología se desarrolló en 1940 para la utilización en radares. Tradicionalmente utilizado en aplicaciones de comunicaciones militares y seguras. Aunque en los últimos años esta técnica de modulación se adoptó en más aplicaciones debido a su relativo bajo consumo para una transmisión y robustez ante mecanismos de degradaciones del canal como multitrayecto, desvanecimiento, Doppler e interferencias.

Se adaptó un estándar 802.15.4 para el CSS PHY por parte del IEEE para Low-Rate Wireless Personal Area Networks (LR-WPANs) para aplicaciones que necesiten un mayor rango y movilidad que el conseguido por O-QPSK DSSS PHY.

3.2.2. LoRa Spread Spectrum

La modulación LoRa trabaja sobre todos estos inconvenientes asociados con sistemas DSSS para proveer una comunicación de bajo costo, bajo consumo, pero más robusta que el resto de las técnicas de Spread Spectrum.

En modulación LoRa, el ensanchamiento del espectro se consigue generando una señal chirp continua que varía con la frecuencia. Una ventaja de este método es que los desfases de tiempo y frecuencia entre Tx y Rx son los mismos, disminuyendo de manera notable la complejidad del receptor. El ancho de banda en frecuencia del chirp es equivalente al ancho de banda espectral de la señal.

A la señal deseada se le aumenta su tasa de transmisión y se la modula con el chirp.

La relación entre la tasa de transmisión de bits deseada, tasa de símbolos y tasa de chirps para una modulación LoRa se puede definir de la siguiente forma:

$$Rb = SF * \left(\frac{1}{\left[\frac{2^{SF}}{BW}\right]}\right) bits/seg \qquad \textbf{Ecuación 3.5}$$

Donde:

Rb = Tasa de modulación de bits
SF = Factor de ensanchamiento (7..12)
BW = Ancho de banda de modulación (Hz)

El perìodo del sìmbolo es:

$$Ts = \frac{2^{SF}}{BW}\ segs \qquad \textbf{Ecuación 3.6}$$

Por lo tanto, la tasa de símbolos, Rs es el recíproco de Ts:

$$Rs = \frac{1}{Ts} = \frac{BW}{2^{SF}}\ símbolos/seg \qquad \textbf{Ecuación 3.7}$$

Finalmente podemos definir la tasa de chips, Rc, como:

$$Rc = Rs * 2^{SF}\ Chips/seg \qquad \textbf{Ecuación 3.8}$$

Aquí se puede ver como se define en las hojas de datos “un chip se envía por segundo por Hz de ancho de banda”:

$$Rc = Rs * 2^{SF} \qquad \textbf{Ecuación 3.9}$$

$$Rc = \frac{BW}{2^{SF}} * 2^{SF}\ Chips/seg \qquad \textbf{Ecuación 3.10}$$

La modulación LoRa también incluye un esquema variable de corrección de errores que mejora la robustez de la señal transmitida al costo de redundancia.

Por ende, podemos definir la tasa de bits nominal de la señal como:

$$Rb = SF * \left(\frac{\left[\frac{4}{4+CR}\right]}{\left[\frac{2^{SF}}{BW}\right]}\right) \qquad \textbf{Ecuación 3.11}$$

Donde:

CR = Tasa de código (1..4)

3.3. Protocolo LoRaWAN

Las redes LoRaWAN[4] típicamente se despliegan en una topología estrella de estrellas, en donde los gateways (por gateway nos referimos a concentradores o estaciones base) envían mensajes entre los nodos y un servidor de red central. El gateway rutea los paquetes de cada dispositivo de la red a la aplicación de servidor asociada. Para asegurar la transmisión, el protocolo utiliza una criptografía simétrica usando llaves de sesión las cuales son derivadas de la llave de fabricación de cada dispositivo.

Los gateways se conectan al servidor de red a través de una conexión estándar IP mientras que los nodos usan un simple salto LoRa o una comunicación modulada por frecuencia (FSK) hacia uno o varios gateways. Toda la comunicación es bidireccional full duplex (si se utiliza la banda de frecuencia 902-928 MHz completa) o half-duplex (si sólo se utiliza la banda 915-928 MHz) aunque la comunicación nodo - gateway (a partir de ahora uplink) es el tráfico predominante.

La comunicación entre los nodos y gateways se reparte en diferentes canales de frecuencia y velocidades de datos, de manera tal que las comunicaciones con distintas velocidades de transmisión no interfieren unas con otras. Las velocidades de datos de LoRa varían desde 0.3 kbps hasta 50 kbps. Para maximizar tanto la duración de la batería de los nodos y el tráfico, la infraestructura de la red LoRa puede manejar la velocidad de datos y la potencia de cada nodo individualmente por medio de un esquema de velocidades de datos adaptativas (ADR).

Los nodos pueden transmitir por cualquier canal libre en cualquier velocidad, siempre y cuando se respeten las siguientes reglas:

- Los nodos cambien de canal de manera pseudoaleatoria por cada transmisión. Esta diversidad de frecuencia hace que el sistema sea más robusto ante las interferencias.

- Los nodos respeten la máxima duración de la transmisión dependiendo de la sub-banda usada y de las regulaciones locales.

- Los nodos respeten la máxima velocidad de datos dependiendo de la sub-banda usada y de las regulaciones locales.

3.3.1. Modos de operación

Existen tres modos de operación para los nodos, dependiendo de qué es lo que se quiera optimizar. El modo utilizado en la aplicación que se dió en este proyecto es el modo A, por lo que nos concentramos en los detalles del mismo y los demás serán explicados de manera muy breve.

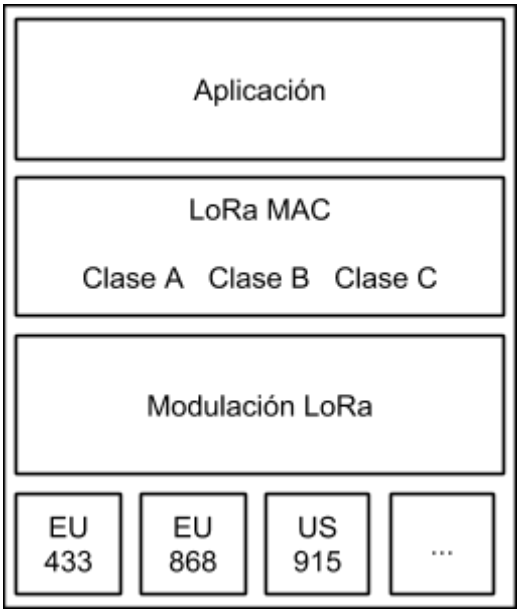


Figura 3.3.1.1. Arquitectura LoRa.

En la figura 3.3.1.2. distinguimos los diferentes modos de operación de los nodos, dependiendo de la cantidad de energía que utilizan y la demora que generan en la red.

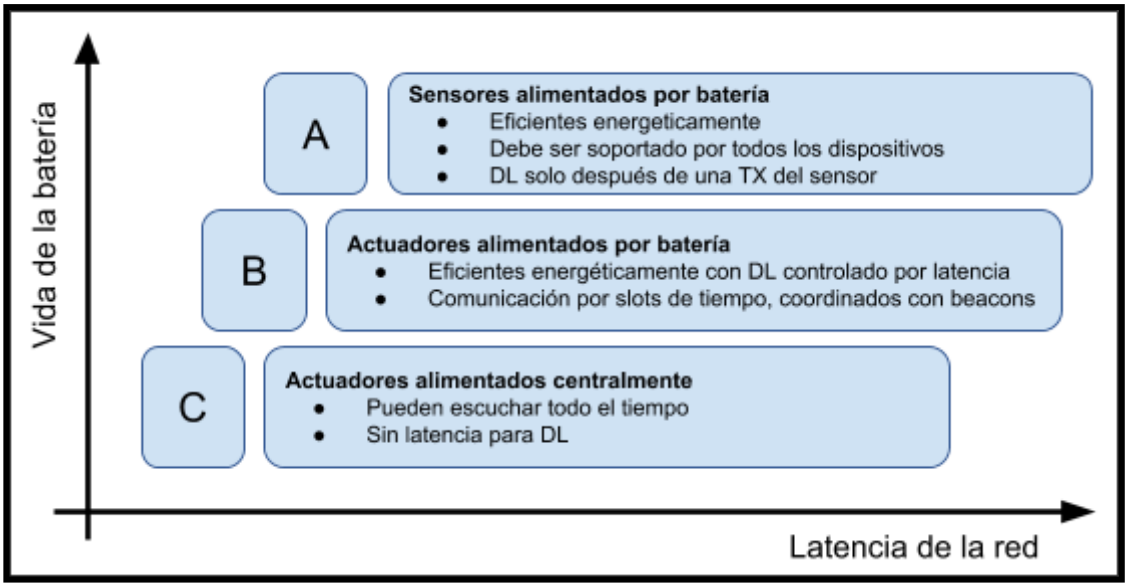


Figura 3.3.1.2. Comparación de los distintos dispositivos LoRa.

3.3.1.1. Clase A

La clase o modo A permite comunicaciones bidireccionales donde los mensajes son del tipo de transmisión Unicast, a un solo destino, con payload (carga útil, mensaje) pequeños e intervalos largos. Los nodos son los encargados de comenzar la comunicación y el gateway se comunica con los nodos durante la ventana predeterminada de respuesta. Cada transmisión uplink de un nodo es seguido por dos cortas ventanas de recepción downlink (comunicación gateway - nodo). El nodo es quien se encarga de variar los intervalos de transmisión, basándose en sus propias necesidades de comunicación junto con una pequeña variación azarosa para evitar colisiones. Esto se explicará más en detalle en la sección 3.3.2.3. Este tipo de clase es la que utiliza la menor energía en los nodos y solo se utiliza para aplicaciones que puedan esperar a las ventanas definidas de downlink. Si el gateway desea comunicarse con algún nodo no puede iniciar un enlace, debe esperar hasta que el nodo haga su transmisión uplink. Todos los nodos deben ser clase A y pueden tener características adicionales para ser B o C.

3.3.1.2. Clase B

El modo B establece una comunicación bidireccional con los nodos mediante el uso de slots de tiempo. Los nodos clase B permiten una mayor cantidad de ventanas de recepción. Además de la ventana azarosa de la clase A, los dispositivos de clase B abren una ventana extra de recepción en momentos planificados. Para que el nodo pueda abrir su ventana de recepción en un momento determinado, este recibe una señal de beacon (faro) del Gateway. Este modo permite señales del tipo Multicast y Unicast con payloads pequeños en intervalos largos. La señal beacon del gateway es transmitida de forma periódica y a las ventanas de recepción extras se las denomina ping slot, esto permite al Gateway iniciar una transmisión en intervalos fijos. El consumo de energía de esta clase es mayor que el A, pero al ser recepciones pactadas periódicas el nodo puede entrar en modo de suspensión entre ping slots y seguir siendo pequeño el consumo de energía.

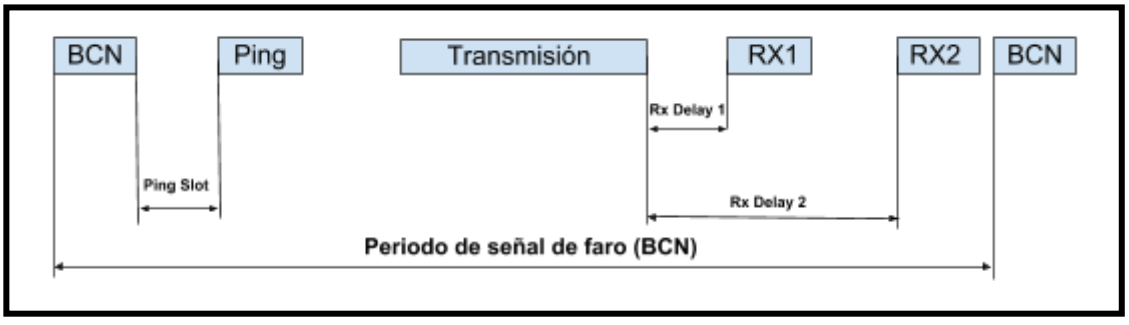


Figura 3.3.1.2.1. Tiempos de transmisión, recepción y señal de beacon del modo B.

3.3.1.3. Clase C

El modo C es también para nodos bidireccionales y se utiliza para minimizar la demora en la comunicación. Al igual que el modo B admite mensajes Unicast y Multicast, está diseñado para payloads pequeños pero con la notoria diferencia de que el servidor puede iniciar la comunicación en cualquier momento. Los nodos mantienen sus ventanas de recepción abiertas prácticamente de manera continua, cerrandolas solo cuando se requiere transmitir. Los nodos que permiten la operación en clase C utilizan más energía pero mimizan la demora en la comunicación gateway - nodo, por esa razón se conectan normalmente a una alimentación de 220V.

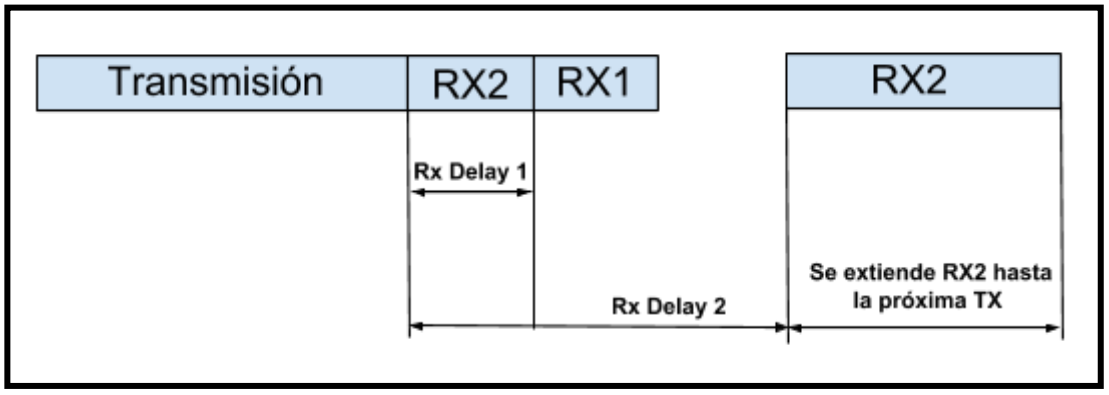


Figura 3.3.1.3.1 Tiempos de transmisión y recepción del modo C.

3.3.2. Formato de los mensajes de capa física

Hay dos tipos de mensajes; los mensajes tipo nodo - gateway llamados uplink y los tipo gateway - nodo llamados downlink. Antes de comenzar se explicarán algunos conceptos.

- **Preámbulo**: La descripción interna de los paquetes que viajan en una red.
- **Cabecera o Header**: Conjunto de datos que se encuentran antes de la carga útil o payload, que sirven para ordenar, entender o armar el mensaje.
- **Carga útil o Payload**: Es el conjunto de datos transmitidos o el mensaje enviado.
- **CRC o verificación por redundancia cíclica**: Es un código de detección de errores que se utiliza para detectar cambios accidentales en los datos.

- **ISM**: Bandas de frecuencias reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica.

3.3.2.1. Mensajes Uplink

Los mensajes de uplink incluyen un preámbulo, cabecera de la capa física (PHDR) además de un CRC de cabecera (PHDR_CRC) y un CRC para asegurar la integridad del payload. Tanto el PHDR, PHDR_CRC y el CRC del payload son insertados por el transmisor de radio.

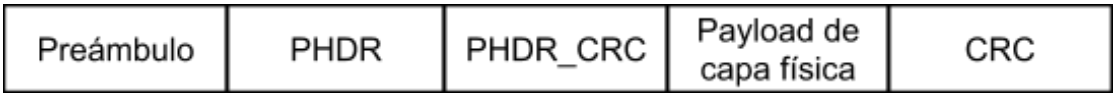


Figura 3.3.2.1.1. Estructura de mensaje Uplink de capa física

3.3.2.2. Mensajes Downlink

Cada mensaje de downlink se envía por el servidor de red hacia uno solo de los nodos y es retransmitido por un solo gateway. Los mensajes de downlink incluyen un preámbulo y una cabecera de la capa física (PHDR), además de un CRC de cabecera (PHDR_CRC)

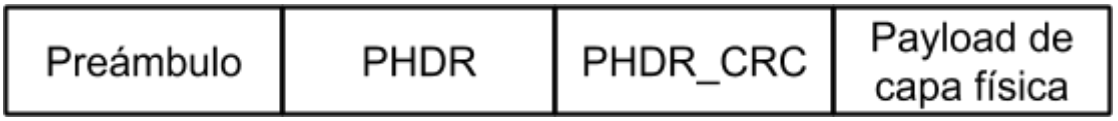


Figura 3.3.2.2.1. Estructura de mensaje Downlink de capa física

Cabe destacar que no se agrega un CRC para el Payload, de esta manera se mantiene el mensaje tan corto como sea posible para lograr un impacto mínimo en las limitaciones de ocupación de las bandas ISM usadas.

3.3.2.3. Ventanas de recepción

Luego de cada transmisión uplink el nodo debe abrir dos pequeñas ventanas de recepción. Los tiempos de comienzo de las ventanas de recepción se definen usando como referencia el fin de la transmisión.

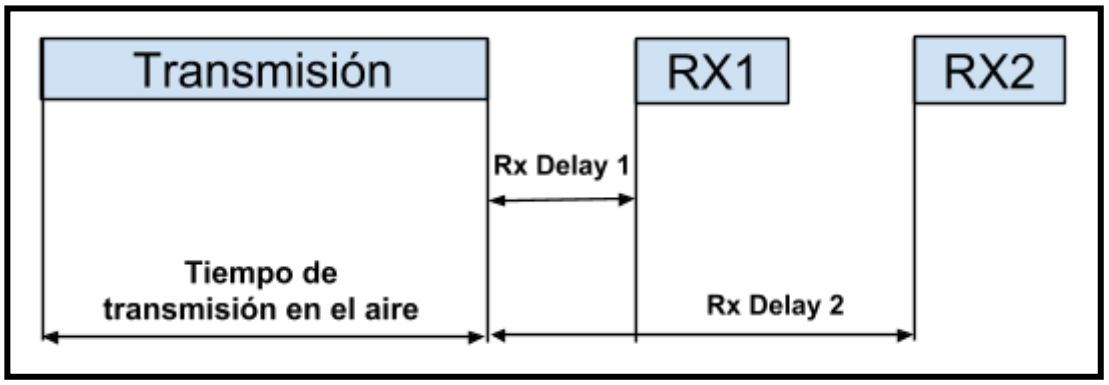


Figura 3.3.2.3.1. Tiempos de transmisión y recepción del nodo.

La primera ventana RX1 utilizará valores de frecuencia y tasa de transmisión dependiendo de los que haya utilizado el nodo para su transmisión uplink previa. Entre la transmisión uplink y ventana de recepción RX1 existe un tiempo llamado"Rx Delay 1" (el cual es una constante del protocolo junto con Rx Delay 2 que son alrededor de 20 microsegundos). La relación entre la tasa de transmisión de uplink y RX1 depende de la región y se especifican en la tabla 3.3.2.3.1, pero por defecto es la misma que el último uplink. El nodo no debe transmitir otro mensaje de uplink antes de que haya recibido un mensaje de downlink en la primera o segunda ventana de recepción de la transmisión previa o la segunda ventana de recepción del mensaje anterior haya caducado.

La segunda ventana de recepción RX2 usa una frecuencia y velocidad de datos fija configurable y abre "Rx Delay 2" segundos después de la finalización de la modulación del uplink. La frecuencia y velocidad de datos se puede modificar a través de comandos MAC (disponibles en la especificación LoRaWAN), pero por default dependen de la región y se especifican en la tabla 3.3.2.3.1. La duración de la ventana de recepción debe ser por lo menos el tiempo que le requiere al radio transmisor del nodo detectar un preámbulo de downlink.

Si se detecta un preámbulo durante una de las ventanas de recepción, el radio transmisor se mantiene activo hasta que el mensaje de downlink es demodulado. Si se detecta un mensaje y subsecuentemente se demodula durante la primera ventana de recepción y el mensaje era para el nodo que lo recibió (luego de chequear dirección y

MIC, integridad del código del mensaje) el nodo no debe abrir la segunda ventana de recepción.

País	*Banda de frecuencias*
Argentina	902 - 928 MHz (915-928 MHz es la utilizable)
Brasil	902 - 907.5 MHz; 915 - 928 MHz; 433 - 435 MHz
Alemania	433.05 - 434.79 MHz y 863 - 870 MHz
Corea del Sur	917 - 923.5 MHz
Estados Unidos	902 - 928 MHz

Tabla 3.3.2.3.1 Asignación de frecuencias por país.

Si se desea conocer algún país en particular se puede recurrir al anexo 9.5.

3.3.2.4. Envío de mensajes de la red hacia un nodo

Si la red intenta transmitir un mensaje de downlink a un nodo, esta debe iniciar una transmisión precisamente al principio de al menos una de las dos ventanas de recepción. Si un mensaje de downlink es transmitido durante ambas ventanas, se debe enviar el mismo mensaje exacto en ambas ventanas.

3.3.3. Formato de los mensajes de capa MAC

Todos los mensajes de uplink y downlink llevan un Payload de la capa física, el mismo contiene un encabezado MAC (MHDR) cuya longitud es un byte, seguido por un payload de MAC (MACPayload), y terminan con un mensaje de integridad de código (MIC) de 4 bytes de longitud.

- **Mensaje enviado por la capa física**:

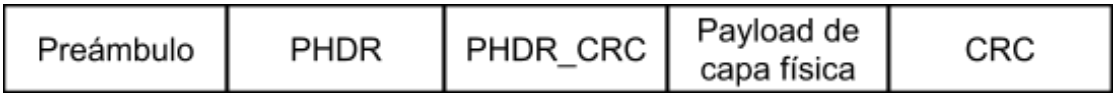
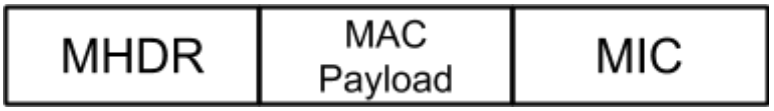
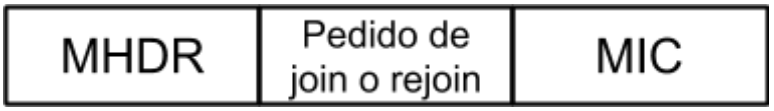


Figura 3.3.3.1. Mensaje enviado por la capa física (CRC es solo para mensajes uplink)

- **Payload de capa física**:



ó



ó

Figura 3.3.3.2. Estructura del Payload de la capa física. Nótese que para el mensaje de Join aceptado el campo MIC está encriptado con el Payload y no es un campo separado.

3.3.3.1. Cabecera MAC (MHDR)

En la Cabecera de MAC la parte más importante es la que especifica el tipo de mensaje enviado. Los distintos tipos de mensajes se pueden ver en la siguiente tabla:

Tipo de Mensaje	*Descripción*
0	Join-request (pedido de unión)
1	Join-accept (pedido de union aceptado)
10	Datos sin confirmar Up
11	Datos sin confirmar Down
100	Datos confirmados Up
101	Datos confirmados Down
110	Rejoin-request
111	Mensaje Propietario

Tabla 3.3.3.1.1. Tipos de mensajes de MAC

- Join-request, Join accept y Rejoin-request: se utilizan en el proceso de activación sobre el aire o Over the Air (OTA) el cual será explicado más adelante en la sección 3.3.4.2.

- Los mensajes de datos se utilizan para transferir tanto comandos de MAC como datos de aplicaciones, los cuales pueden combinarse en un solo mensaje. Un mensaje de datos confirmado debe ser reconocido por el receptor pero los mensajes de datos sin confirmar no requieren un reconocimiento.

- Los mensajes propietarios se pueden usar para implementar formatos de mensajes personalizados pero estos no serán compatibles con los mensajes estándar. Se puede usar uno o el otro según la necesidad pero no ambos al mismo tiempo. Si un nodo o servidor recibe un mensaje que no es compatible con su estándar, debe descartarlo inmediatamente y sin avisar.

3.3.3.2. MAC Payload

Figura 3.3.3.2.1. Estructura de MACPayload

Contiene una cabecera de trama (FHDR) seguida de un campo opcional de puerto (FPort) y otro campo opcional de payload de trama (FRMPayload). El único valor obligatorio del MACPayload es el FHDR. FPort y FRMPayload se utilizan para indicar comandos MAC los cuales sirven para manejar y ordenar la red y todos estos comandos son invisibles a la aplicación. Por estas razones aquí se explica en detalle sólo el FHDR.

Figura 3.3.3.2.2. Estructura de cabecera de trama (FHDR)

El FHDR contiene 4 bits con la dirección de dispositivo del nodo (DevAddr), un bit de control de trama (FCtrl), dos bits de contador de trama (FCnt) y hasta 15 bits de opciones de trama (FOpts) los cuales se usan para transportar comandos de MAC. Si FOpts está presente, este mismo debe ser encriptado.

Figura 3.3.3.2.1. Campos del mensaje FCtrl de downlink

Figura 3.3.3.2.2. Campos del mensaje FCtrl de uplink

- **Velocidad de datos adaptativa en la cabecera de trama (ADR)**

La red LoRa le permite a los nodos el uso individual de cualquier velocidad posible de datos y potencia de transmisión. Esta característica se utiliza en el protocolo LoRaWAN para adaptar y optimizar la velocidad de datos y la potencia de transmisión de los nodos estáticos. A este sistema se lo denomina ADR y cuando este se habilita la red va a ser optimizada para utilizar la mayor velocidad de datos posible. El control ADR puede fallar cuando la atenuación del canal de radio cambia rápida y constantemente. Cuando el servidor de red no es capaz de controlar la velocidad de datos de un dispositivo, la capa de aplicación del dispositivo debería controlarla.

Si el bit de ADR de uplink se setea, la red va a controlar la velocidad de datos y la potencia de los nodos a través de los comandos MAC apropiados. Si el bit de ADR no se setea, la red no va a intentar controlar la velocidad de datos o la potencia de transmisión de los nodos independientemente de la calidad de la señal recibida. La red puede aún enviar comandos para cambiar el canal o los parámetros para reenvío de tramas.

Si el bit de ADR de downlink se setea, se informa al nodo que el servidor de red está en una posición para enviar comandos de ADR. El dispositivo puede setear o no el bit de ADR de uplink.
Cuando no se setea el bit ADR de downlink, este le avisa al nodo que dado a cambios rápidos en el canal de radio, la red temporalmente no puede estimar la mejor velocidad de datos. En este caso el dispositivo tiene la opción de:

- ✓ No setear el bit de ADR de uplink y controlar la velocidad de datos de uplink siguiendo su propia estrategia. Esta debería ser la estrategia utilizada por la mayoría de los nodos. Es decir que si la velocidad de datos no es controlada por el gateway, el nodo podrá elegir con qué tasa de transmisión trabajar en base a lo que dicte su código de aplicación y es totalmente independiente. Pero se recomienda que la aplicación siempre intente minimizar el tiempo de mensaje en el aire (tiempo que transcurre entre que el Tx envía el mensaje y el Rx realmente lo recibe).

- ✓ Ignorarlo manteniendo el bit ADR de uplink seteado, y aplicar el decaimiento normal de la velocidad de datos frente a la ausencia de comando ADR de downlink. Esta debería ser la estrategia utilizada por los nodos estacionarios.

El bit de ADR puede ser seteado o no por el nodo o la red en cualquier momento, Sin embargo, siempre que sea posible el sistema ADR debería estar habilitado para incrementar la duración de la batería del nodo y maximizar la capacidad de la red.

Si la velocidad de datos de un nodo es optimizada por la red para usar una velocidad de datos mayor que la estándar, necesita periódicamente validar que la red aún recibe las tramas de uplink. Cada vez que el contador de trama de uplink es incrementado (lo hace cada vez que se recibe un nuevo uplink, las transmisiones repetidas no se cuentan) el dispositivo incrementa un contador llamado ADR_ACK_CNT. Este se incrementa hasta un límite llamado ADR_ACK_LIMIT. Cuando el contador alcanza el límite de mensajes de uplink sin ninguna respuesta de downlink se setea el bit de pedido de confirmación llamado ADRACKReq. La red requiere responder con un mensaje de downlink dentro de un determinado tiempo (ADR_ACK_DELAY), cualquier mensaje de downlink recibido después de una trama de uplink resetea el contador. Si no se recibe respuesta después de ese tiempo (lo que tarda en llegar al límite ADR_ACK_LIMIT + ADR_ACK_DELAY) el nodo debe intentar reconectarse primero subiendo la potencia de transmisión y luego bajando la velocidad de datos paso a paso hacia una velocidad de datos que provea un mayor radio de distancia. El nodo debe seguir bajando su velocidad de datos cada vez que se llegue al límite de tiempo. Una vez que el nodo llega a su velocidad de datos mínima debe habilitar todos los canales de frecuencias de uplink.

- **Bit de ACK y FCtrl para la confirmación de mensajes**

Cuando se recibe un mensaje de datos confirmado, el receptor debe responder con una trama que tenga el bit de confirmación ACK seteado. Si el emisor es un nodo, la red debe intentar enviar la confirmación usando una de las ventanas de recepción abiertas por el nodo después del envío. Si el emisor es un gateway, el dispositivo debe transmitir una confirmación cuando lo vea propicio. El ACK solo se debe enviar en respuesta al último mensaje recibido y nunca es retransmitido.

- **Proceso de retransmisión**

Una trama de downlink confirmada o sin confirmar no debe ser re transmitida usando el mismo valor de contador de trama (FCnt).
Las tramas de uplink confirmadas y sin confirmar se transmiten “NbTrans” veces, a excepción de que se reciba un downlink válido luego de una transmisión. El nodo debe realizar un salto de frecuencias como es usual entre transmisiones repetidas, debe esperar hasta después de cada repetición hasta que la ventana de recepción haya expirado. La demora entre la retransmisión es a discreción del nodo y puede ser

diferente para cada nodo. Se deben parar futuras retransmisiones si se recibe la trama ACK de downlink correspondiente.

- **Bit FPending**

La trama de bit pendiente solo se usa en la comunicación downlink, indicando a la red que tiene más datos pendientes para ser enviados y por lo tanto le pide al nodo que abra otra ventana de recepción tan pronto como sea posible al enviar otro mensaje de uplink.

- **Contador de trama (FCnt)**

Cada nodo tiene tres contadores de trama para tener un control de la cantidad de mensajes de uplink que se envían al servidor de red y mensajes de downlink desde el servidor de red hacia el dispositivo.

- **Opciones de trama (FOptsLen dentro de FCtrl y Fopts)**

El campo de longitud de opciones de trama (FOptsLen) marca la longitud del campo de opciones de trama (FOpts) dentro de la misma. FOpts transporta comandos MAC de un máximo de 15 bits. Si FOptsLen es 0, el campo FOpts está ausente. Se pueden incluir comandos MAC simultáneamente en el payload como en FOpts y si esto ocurre el dispositivo debe ignorar la trama. Si la cabecera incluye FOpts, este mismo campo debe ser encriptado antes de que el mensaje de integridad de código (MIC) sea calculado. La encriptación usada es un algoritmo genérico descrito en IEEE 802.15.4/2006[5] Anexo B usando AES (para más información ver anexo 10.4) con una clave de 128 bits de longitud.

- **Bit Clase B**

El bit de clase B se setea en 1 si el mensaje de uplink le indica al servidor de red que el dispositivo pasó a la clase B.

- **Campo puerto (FPort)**

Si en la trama el campo de payload no está vacío, el campo puerto debe estar presente. Si está presente, un valor de 0 indica que el FRMPayload contiene solo mensajes de

MAC y cualquier trama con el valor de puerto 0 debe ser procesada por la implementación de LoRaWAN. Valores entre 1 y 223 son específicos de la aplicación y cualquier trama con estos valores se debe hacer disponible para la capa de aplicación de la implementación LoRaWAN.

- **Encriptación del payload de capa MAC (FRMPayload)**

Si la trama lleva un payload, FRMPayload debe estar encriptado antes de que se calcule el mensaje de integridad de código (MIC). El tipo de encriptación es el mismo que se utiliza para el campo FOpts.

3.3.3.3. Mensaje de integridad de código (MIC)

El mensaje de integridad de código[6] es una porción de información usada para confirmar que el mensaje proviene del emisor correcto y que no se ha modificado en el camino. El MIC protege tanto la integridad del dato como su autenticidad, permitiendo detectar cualquier cambio en el contenido del mensaje.

3.3.4. Activación de los nodos

Para participar en una red LoRaWAN, cada nodo debe ser personalizado y activado. La activación de un nodo se puede hacer de dos maneras: Activación sobre el aire (OTAA) o activación vía personalización (ABP). Como el modo elegido para este proyecto fue la activación sobre el aire se explicará el mismo en profundidad.

3.3.4.1. Datos guardados dentro del nodo

- **JoinEUI**

El JoinEUI es una identificación global de aplicación en el espacio de dirección IEEE EUI64[7] que identifica de manera única el servidor de Join que permite asistir el proceso de Join y de derivación de llaves de sesión. En los dispositivos OTAA el JoinEUI debe ser guardado por el nodo antes de ejecutar el proceso de Join.

- **DevEUI**

El DevEUI es una identificación global de nodo en el espacio de dirección IEEE EUI64 que identifica de manera única el nodo. El DevEUI es el identificador único de dispositivo para la servidor de red, sin importar el proceso de activación. En los dispositivos OTAA el DevEUI debe ser guardado por el nodo antes de ejecutar el proceso de Join.

- **Claves root de dispositivo (AppKey y NwkKey)**

El NwkKey y AppKey son claves del tipo AES-128 específicas para los nodos que se les asignan durante el proceso de fabricación. Cuando un nodo se una a una red por la activación por el aire el NwkKey es utilizado para derivar las llaves FNwkSIntKey, SNwkSIntKey y NwkSEncKey. La llave AppKey se utiliza para derivar la llave AppSKey. Ambas claves deben estar guardadas en el nodo antes de ejecutar el proceso de Join.

Luego de la activación , las siguientes informaciones adicionales son guardadas en el nodo: una dirección de dispositivo (DevAddr), un trío de llaves de sesión de red (NwkSEncKey / SNwkSIntKey / FNwkSIntKey) y una llave de sesión de aplicación (AppSKey)

- **Dirección de nodo (DevAddr)**

La dirección del nodo consiste en 32 bits que identifican el nodo dentro de la red actual. Está misma es brindada por el servidor de red del nodo

- **Llave de integridad de sesión red de reenvío (FNwkSIntKey)**

La llave de integridad de sesión de red de reenvío es una llave de sesión de red específica para los nodos. Es utilizada por los nodos para calcular el MIC o parte del MIC de todos los mensajes de uplink para asegurar la integridad de los datos.

- **Llave de integridad de sesión de red de envío (SNwkSIntKey)**

La llave de integridad de sesión de red de envío es una llave de sesión de red específica para los nodos. Es utilizada por el nodo para verificar el MIC de todos los

datos de los mensajes de downlink para asegurar la integridad de los datos y computar la mitad de los MIC de los mensajes de uplink.

- **Llave de encriptación de sesión de red (NwkSEncKey)**

La llave de encriptación de sesión de red es una llave de sesión de red específica del nodo, Se usa para encriptar y desencriptar los comandos MAC de uplink y downlink transmitidos como payload en el campo de FOpts.

- **Llave de aplicacion de sesión (AppSKey)**

La llave de aplicación de sesión es específica para el nodo. Es utilizada por aplicaciones de servidor y por el nodo para encriptar y desencriptar el campo payload de los mensajes de datos específicos para la aplicación. Los mensajes de aplicación están encriptados desde el nodo hasta el servidor de aplicación, es decir, de punta a punta.

3.3.4.2. Activación OTAA

Para el proceso de activación OTAA los nodos deben seguir el proceso de join para participar en el intercambio de datos de un servidor de red. Un nodo tiene que pasar por un nuevo proceso de join cada vez que pierde la información del contexto de la sesión.

- **Procedimiento de Join**

Para el punto de vista del nodo, el proceso de join consiste en un intercambio de Join o re Join-request y un Join-accept.

- **Join-request**

El procedimiento de Join siempre es iniciado por el nodo al enviar un mensaje de Join request

Figura 3.3.4.2.1. Campos del mensaje de Join-request

El mensaje de Join-request contiene JoinEUI y el DevEUI del nodo seguido por un nonce de dos bits. Un nonce es un número arbitrario que sólo puede utilizarse una vez, es un término muy utilizado en criptografía y seguridad informática. DevNonce es un contador que comienza desde 0 cuando el dispositivo es prendido y es incrementado con cada Join-request. El valor de DevNonce nunca puede ser reutilizado para un dado valor de JoinEUI. Resetear el DevNonce sin cambiar el JoinEUI va a causar que el servidor de red descarte el join-request del dispositivo. Por cada nodo el servidor de red mantiene un registro del último valor usado de DevNonce por el nodo e ignora el Join-request si este no es incrementado.

- **Join-accept**

El servidor de red va a responder al mensaje de Join o Join-request con un mensaje de Join-accept si se le permite al nodo unirse a la red. El mensaje de Join-accept se envía como un mensaje de downlink normal solo que utiliza delays diferentes. El canal de frecuencia y velocidad de datos utilizados para las ventanas de recepción son idénticas a las ventanas usuales. No se da respuesta al nodo si el Join-request no fue aceptado. El mensaje de join-accept contiene un nonce de servidor (JoinNonce) de 3 bits, un identificador de red (NetID), una dirección de nodo (DevAddr), un campo llamado (DLSettings) que provee algunos parámetros de downlink, el retraso entre Tx y Rx (RxDelay) y una lista opcional de los parámetros de red (CFList) para la red a la cual el nodo se está uniendo.

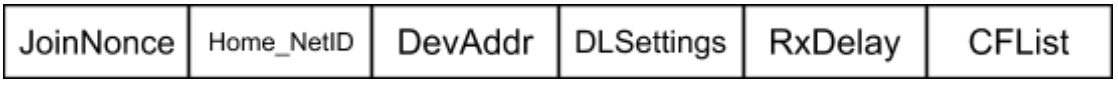


Figura 3.3.4.2.2. Campos del mensaje de Join-accept

El JoinNonce es un valor contador específico del dispositivo (el cual nunca se repite) provisto por el servidor de Join y utilizado por el nodo para derivar las llaves de sesión FNwkSIntKey, SNwkSIntKey, NwkSEncKey y la AppSKey. JoinNonce se incrementa con cada mensaje de Join accept.

El dispositivo lleva un registro del último valor de JoinNonce utilizado en el último proceso exitoso de Join accept por lo tanto puede aceptar el Join accept solo si el campo MIC es correcto y el JoinNonce es estrictamente mayor que el valor guardado. En ese caso el nuevo valor de JoinNonce reemplaza el que estaba guardado previamente. La alianza LoRa provee NetID de 24 bits únicos para identificar las redes. El campo Home_NetID corresponde con el valor de NetID de la red del dispositivo.

- **Rejoin-request**

Una vez activado un dispositivo puede transmitir periódicamente un mensaje de Rejoin-request sumado a su tráfico habitual. Este mensaje le da al servidor de red la oportunidad de iniciar un nuevo contexto de sesión para el nodo. La red responde el Rejoin-request con un mensaje de Join-accept. El Rejoin-request puede ser utilizado para cambiar la llaves y/o cambiar el devAddr de un dispositivo de una red.

4. Descripción del modelo experimental

4.1. Hardware Utilizado

Para la realización de este proyecto se utilizó el siguiente hardware:

4.1.1 Multitech MultiConnect mCard MTAC-LORA-H-915
4.1.2 Multitech Multiconnect Conduit Gateway
4.1.3 Multitech Multiconnect xDot MTXDOT-915
4.1.4 Antenas LORA 915Mhz

4.1.1. Multitech MultiConnect mCard MTAC-LORA-H-915

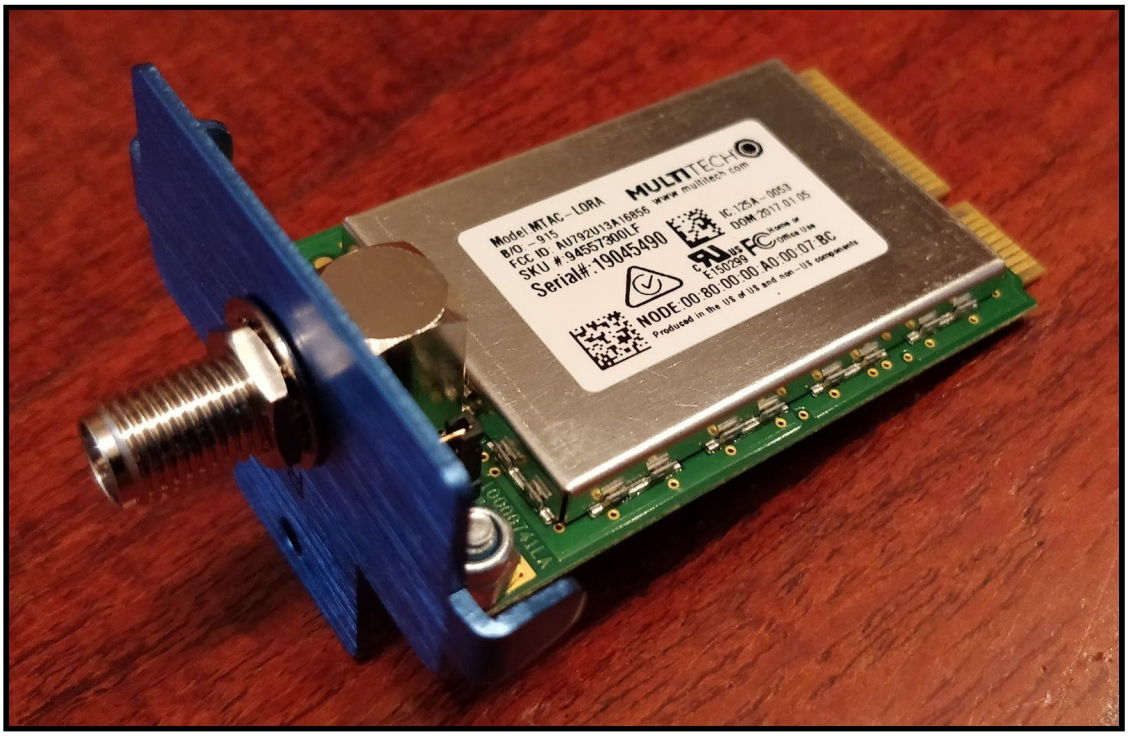


Figura 4.1.1.1. Fotografía del Módulo LoRa que se conecta al Gateway Conduit.

Este módulo simplemente se introdujo y atornilló a la parte trasera del gateway, posee una salida hembra RP-SMA para colocar la antena LoRa.

4.1.2. Multitech Multiconnect Conduit Gateway

Figura 4.1.2.1. Fotografía del Gateway Conduit empleado.

El gateway es la puerta de enlace de los distintos clientes, sirve para comunicar al mundo LoRa con el mundo TCP/IP y también para comunicar a los clientes entre sí. Es el punto clave de la red ya que de él dependen todos los módulos y es el encargado de procesar la información para mostrarla en un sitio web, el cual también es alojado en su sistema.

El sistema utilizado por el Gateway Conduit es una distribución de Linux de código abierto llamada mLinux. Se construyó en base al proyecto Yocto[8].

4.1.3. Multitech Multiconnect xDot MTXDOT-915

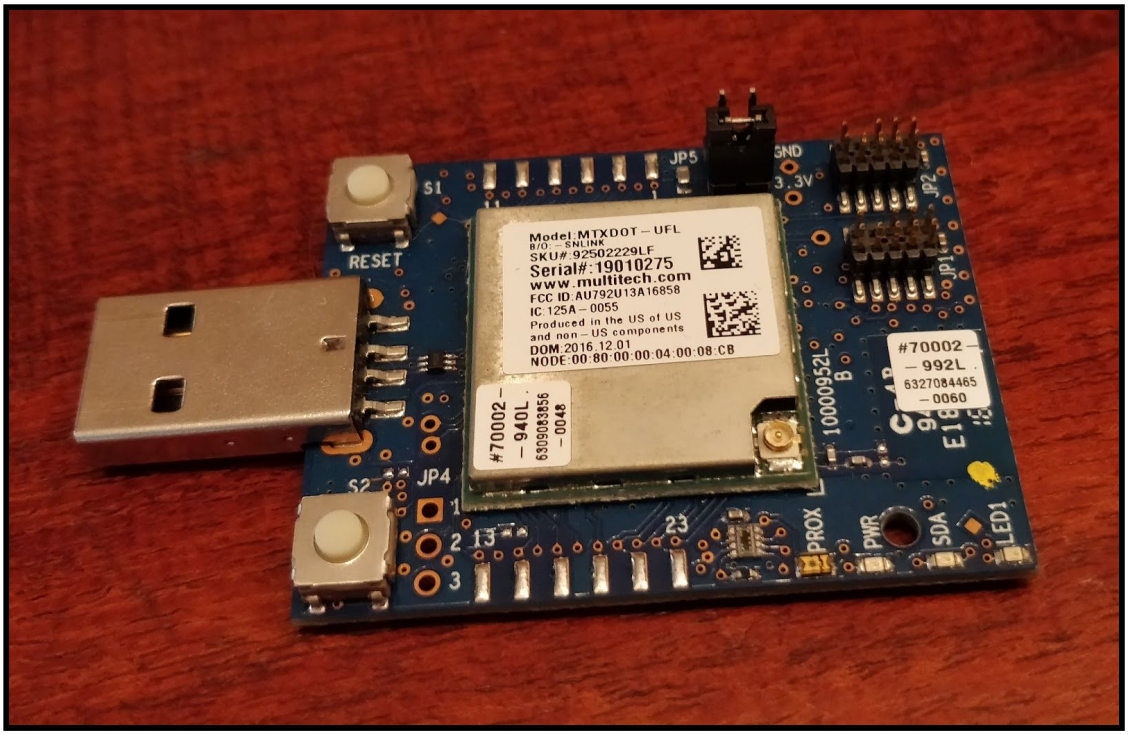


Figura 4.1.3.1. Fotografía del módulo xDot empleado.

El xDot es un equipo programable, de bajo consumo con comunicación por RF, que provee un largo alcance y bajo bit rate con conexión máquina a máquina a sensores y equipos. Es el dispositivo que tomará los datos de los sensores y los enviará mediante modulación LoRa al gateway Conduit para su procesado.

4.1.4. Antenas LORA 915Mhz

Figura 4.1.4.1. Fotografía de una antena LoRa.

Estas antenas se conectan a los módulos xDot y al gateway para permitirles un mayor alcance, la antena utilizada en nuestro caso para el módulo tiene una ganancia de 5 dBi y la del gateway una ganancia de 3 dBi, ambas poseen un conector RP-SMA macho y resonancia alrededor de las frecuencias de 868 y 915 Mhz. En nuestro caso nos interesan los 915 Mhz que se utilizan en la Argentina.

4.2. Modelo experimental

El primer paso proyecto fue configurar correctamente el gateway para la conexión con los distintos terminales.

El gateway posee dos conexiones físicas, una a la red eléctrica para su alimentación y una conexión RJ45 hembra para acceder a internet.

4.2.1. Configuración LoRa Network

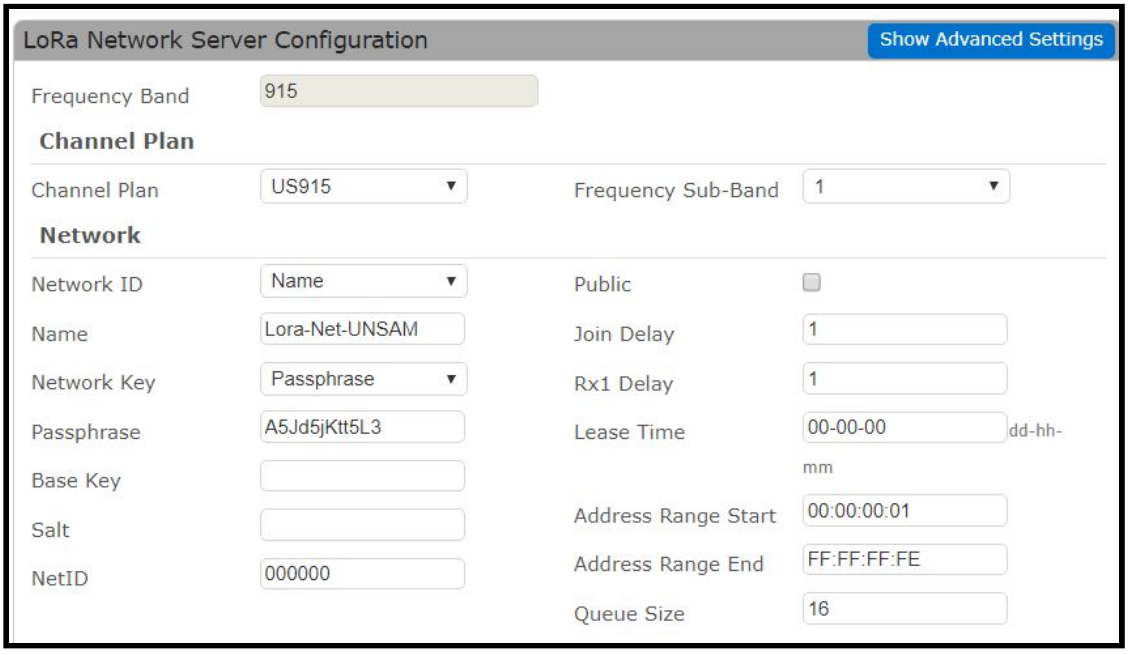


Figura 4.2.1.1. Screenshot de la pantalla de configuración del gateway.

Especificaciones:

- **Channel Plan: US915**: La tecnología LoRa utiliza la banda de frecuencias ISM (Industrial, Scientific and Medical), estas son bandas de frecuencias reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. El uso de estas bandas de frecuencia está abierto a todo el mundo sin necesidad de licencia,

respetando las regulaciones que limitan los niveles de potencia transmitida.[9] Sin embargo, estas frecuencias actualmente son privadas en Argentina, ya que fueron atribuidas a Nextel para el uso de transmisión de señales de LTE. [10][11]

- **Frequency Sub-Band**: 1, existen varios canales dentro de la banda de frecuencias 902-928 Mhz y se separan cada 500 Khz. Hay 64 + 8 canales de uplink encontrados entre las frecuencias de 902.3 Mhz y 914.2 Mhz. Los primeros 64 son de baja tasa de transmisión (0.5 a 1.7 Kbits/s) y los 8 restantes son de mayor velocidad (3 Kbits/s). Los canales a los cuales se refiere esta opción son los de downlink. Estos son 8 y se encuentran desde 923.3 Mhz hasta 927.5 Mhz.

- **Network ID+Name**: Nombre de la red LoRa a la que deben conectarse los nodos. Si se utiliza el Network ID se debe establecer el modo de contraseña de Network Key, y si se utiliza el Network name se debe establecer un Passphrase. No es posible utilizar Name e ID en simultáneo.

- **Network key+Passphrase**: Contraseña de la red LoRa, necesaria para la autenticación.

- **Public**: Habilita el modo de compatibilidad para redes públicas. Modifica la palabra de sincronización de 0x34 (público) a 0x12 (privado).

- **Lease Time**: Tiempo límite para un nodo antes de necesitar una nueva autenticación a la red. (00-00-00 significa ilimitado)

- **Address Range Start+End**: Direcciones que asigna el servidor LoRa para sus nodos.

- **Queue Size**: Delimita la cantidad de mensajes de Downlink a mantener en espera para cada nodo.

Posteriormente, se programó el software que controla al terminal xDot para poder enlazarse y enviar paquetes al gateway constantemente con el objetivo de comprobar su correcto funcionamiento, estos paquetes tienen la siguiente estructura:

{ "ack": false, "adr": true, "appeui": "f3-33-44-9e-59-c4-6f-c4", "chan": 8, "cls": 0, "codr": "4/5", "datr": "SF8BW500", "deveui": "00-80-00-00-04-00-08-cb", "freq": "903", "lsnr": "-4.2", "mhdr": "4001000000809d00", "modu": "LORA", "opts": "", "port": 1, "rfch": 0, "rssi": -90, "seqn": 157, "size": 4, "timestamp": "2018-05-03T21:18:07.342214Z", "tmst": 2775089653, "payload": "5", "eui": "00-80-00-00-04-00-08-cb", "_msgid": "c7f40f6c.380bf" }

Donde[26]:

- **ack:** boolean, si espero un ack posterior a la recepción del mensaje.
- **adr:** boolean, Tasa de transmisión adaptativa.
- **appeui:** Identificación única de la aplicación LoRa.
- **chan:** Canal de frecuencia intermedia utilizado por el gateway.
- **cls:** Clase
- **codr:** Tasa de codificación LoRa ECC.
- **datr:** Tasa de transmisión LoRa.
- **deveui:** Identificación única del dispositivo.
- **freq:** Frecuencia central de trabajo [MHz].
- **lsnr:** Relación señal a ruido LoRa.
- **mhdr:** MAC header.
- **modu:** Tipo de modulación utilizada.
- **opts:** Opciones de trama.
- **port:** Campo Puerto (FPort), depende de la aplicación.
- **rfch:** Canal de frecuencia final utilizado por el gateway.
- **rssi:** Nivel de potencia de la señal recibida [dBm].
- **seqn:** Número de secuencia del mensaje.
- **size:** Tamaño del payload [Bytes].
- **timestamp:** Horario y fecha del mensaje.
- **tmst:** Horario y fecha de RX interno.
- **payload:** Mensaje a transmitir.
- **eui:** Identificación única de la placa que envía el mensaje. (MAC Address)
- **_msgid:** Identificación única del mensaje.

Para la programación y compilación del software se utilizó tanto Eclipse[12] como el compilador online de Mbed[13], el cual consiste en un sitio web muy didáctico donde se puede escribir el programa, comparar con ejemplos de otros usuarios, agregar librerias y obviamente la compilación de los códigos, entre otras cosas.

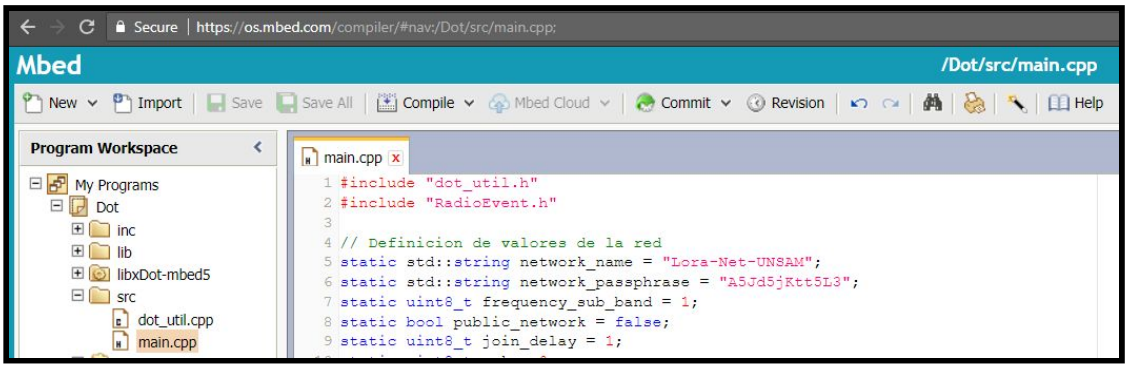


Figura 4.2.1.2. Screenshot del compilador online Mbed.

En este punto se tuvo que definir de qué modo los terminales harían el envío de datos. Debido a que quisimos un consumo bajo para evitar un cambio continuo de baterías, se eligió la clase A.

Para la correcta compilación de un código para las placas xDot, se deben importar ciertas librerías. Estas librerías se encuentran recopiladas en el siguiente link[21]:

https://os.mbed.com/users/unsamdev/code/Dot/

Para utilizarlas, se debe clickear el botón amarillo "Import into compiler" y una vez importadas modificar el archivo "main.cpp" el cual contiene el código principal. Para poder cargar el código ya compilado dentro del nodo xDot, solo basta con arrastrar el archivo que se descarga al compilar hasta el almacenamiento USB que figura en los exploradores de archivos del sistema operativo como "XDOT" y reiniciar el nodo.

Una vez que los terminales se enlazaron correctamente, lo cual se puede visualizar en la sección de estadísticas como se ve en la figura 4.2.1.2. se buscó poder visualizar la información que envían los nodos de una manera más sencilla y amigable al usuario final. La solución a este problema fué la introducción de un sitio web el cual presentaría, mediante un navegador web genérico, la información solicitada.

---

*Desde una conexión SSH al gateway (la cual se explica en la sección 4.2.2) el equivalente a ingresar en la sección estadísticas es ingresar el comando "**lora-query -n**"*

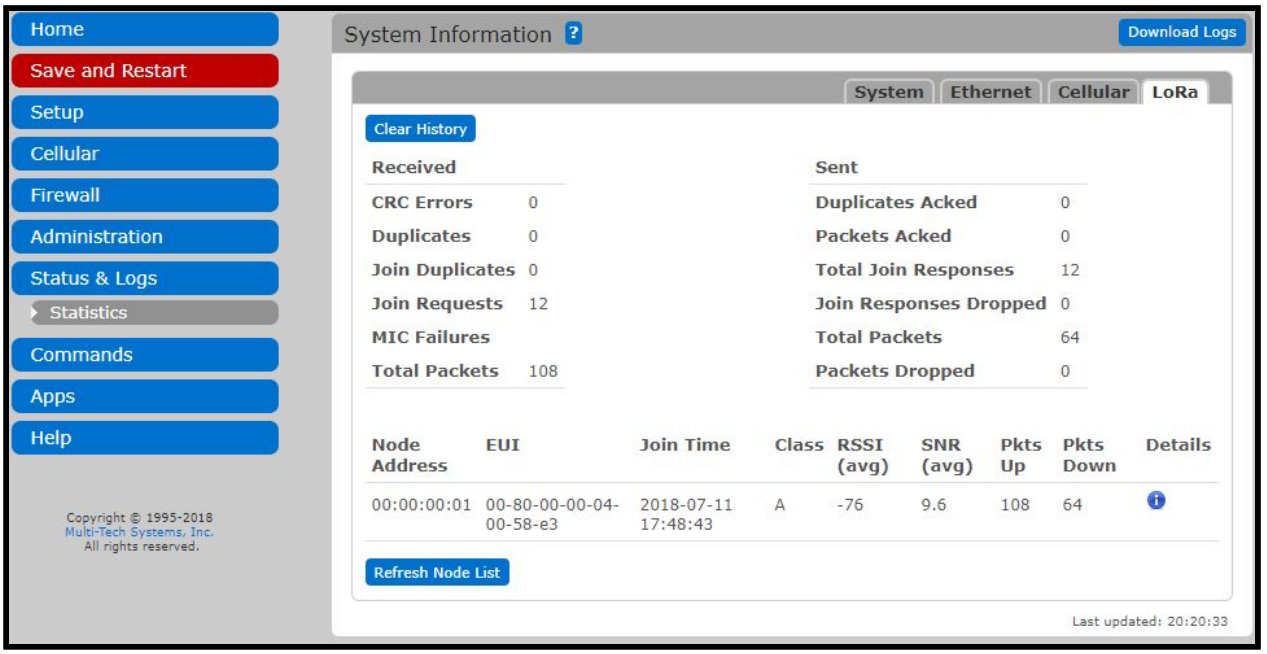


Figura 4.2.1.3. Screenshot de la sección estadísticas del gateway Conduit.

En una primera instancia se utilizó The Things Network (TTN)[14], haciendo que el gateway Conduit reenvíe los datos recibidos a una cuenta en dicha plataforma, la cual a su vez conecta con una aplicación llamada Cayenne[15] para visualizarlo de manera gráfica. Pero posteriormente se encontró la posibilidad de montar un sitio web en el mismo gateway Conduit, el cual hace que no dependamos de la disponibilidad de servidores de terceros para alojar nuestra información y eliminar un punto de falla. Además, esto nos permitió posicionar accionadores o botones en el sitio web para que un humano pueda realizar acciones en base a lo que visualiza en la pantalla, como por ejemplo, si encuentra que el nivel de presión de un tanque es elevado, puede enviarle una señal a su módulo LoRa para accionar el mecanismo que libere la presión acumulada.

---

*Si se desea reiniciar la red LoRa por alguna razón sin reiniciar el gateway, mediante conexión SSH se puede utilizar el comando "**/etc/init.d/lora-network-server stop**" y luego "**/etc/init.d/lora-network-server start**"*

4.2.2. Configuración Node-Red

Para obtener un sitio web alojado en el gateway se instaló un paquete en el sistema mLinux el cual se denomina node-red-contrib-ui[16]. Este paquete se asocia al software Node-Red[17] que ya se encuentra instalado en el Gateway de fábrica y permite la creación del sitio web. Node-Red es una interfaz de programación por bloques que facilita la interacción entre el humano y el dispositivo. La instalación se realizó de la siguiente manera:

- Conectarse por SSH al gateway
  - *ssh admin@10.1.100.130*
  - Ingresar la contraseña correspondiente

- *cd /var/config/app/install/development* (para situarse en el directorio de instalación de nodos)

- *npm install node-red-contrib-ui* (para instalar dicho paquete mediante npm[18], una librería de paquetes para Javascript)

- */etc/init.d/node-red restart* (reinicia la aplicación para refrescar cambios)

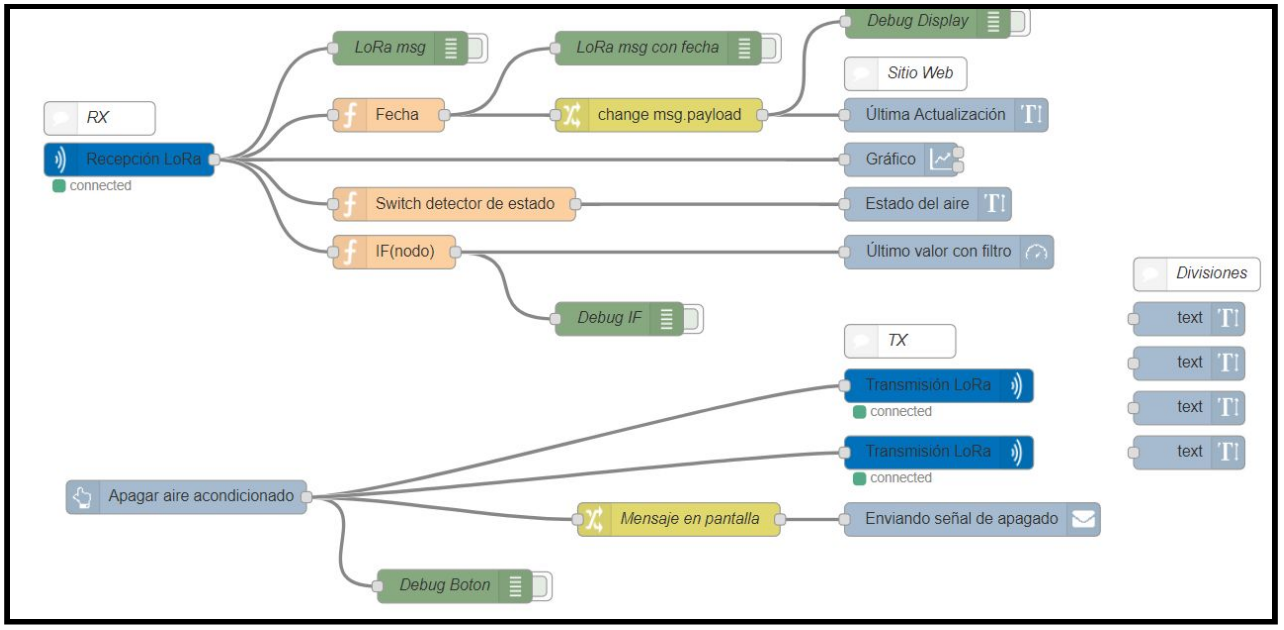


Figura 4.2.2.1. Screenshot de la programación en bloques en Node-Red

Para conectarse al Node-Red se debe ingresar en la barra de direcciones de un navegador web la dirección IP del gateway con el puerto correspondiente, siempre y cuando estemos en la misma subred, de la siguiente manera:

- https://10.1.100.130:1880

Y luego ingresar los datos para iniciar sesión igual que en la administración del gateway.

Los nodos utilizados en el Node-Red fueron:

- **Input**:
  - **Lora**: nodo de ingreso de paquetes desde módulos enlazados con el gateway, acepta la totalidad de envíos de todos los módulos asociados, se puede filtrar luego si decido aceptar solo algunos.

- **Output**:
  - **Debug**: nodo de testeo, muestra por pantalla todo lo que recibe.
  - **Lora**: nodo de salida de paquetes hacia módulos, se define un solo módulo por nodo, al contrario de la recepción.

- **Function**:
  - **Function**: nodo para realizar funciones en Javascript, lo utilizamos para agregar la fecha de recepción, y detectar el estado del aire.
  - **Comment**: nodo para agregar un comentario en la pantalla de programación.
  - **Change**: nodo para modificar cierta parte del mensaje en base a patrones. Lo utilizamos para quitar la zona horaria.

- **ui**:
  - **Button**: nodo para agregar un botón en el sitio web creado por ui.
  - **Switch**: nodo para agregar un switch en el sitio web creado por ui.
  - **Text**: nodo para agregar texto a un cuadro en el sitio web creado por ui.
  - **Chart**: nodo para agregar un gráfico en el sitio web creado por ui.

Para ingresar inicialmente al sitio web se debía colocar la dirección:

- https://10.1.100.130:1880/ui

Para facilitar el uso de esta herramienta, se solicitó a la Gerencia de Informática la posibilidad de publicar a internet dicho sitio web mediante la siguiente URL, para conseguir esto se debe asignar una IP pública al gateway (o mediante un proxy pero no fue este el caso) y generar un registro DNS que traduzca esa IP a nuestro dominio:

- https://lora.unsam.edu.ar:1880/ui

*De la misma manera que para reiniciar la red LoRa, si se desea reiniciar el Node-Red y el sitio web se puede utilizar mediante conexión SSH el comando "**/etc/init.d/node-red stop**" y luego "**/etc/init.d/node-red start**"*

Luego, se procedió a agregar distintas funcionalidades al sitio web para lograr nuestro objetivo de mostrar eficientemente la información para un proyecto de ejemplo, el cual consta de medir el consumo de dispositivos eléctricos y poder apagarlos remotamente en caso de ser necesario. Para esto se agregaron gráficos que representan el consumo de energía respecto al tiempo y accionadores tipo botón para poder apagar el dispositivo que está consumiendo en caso que el humano lo considere correcto.

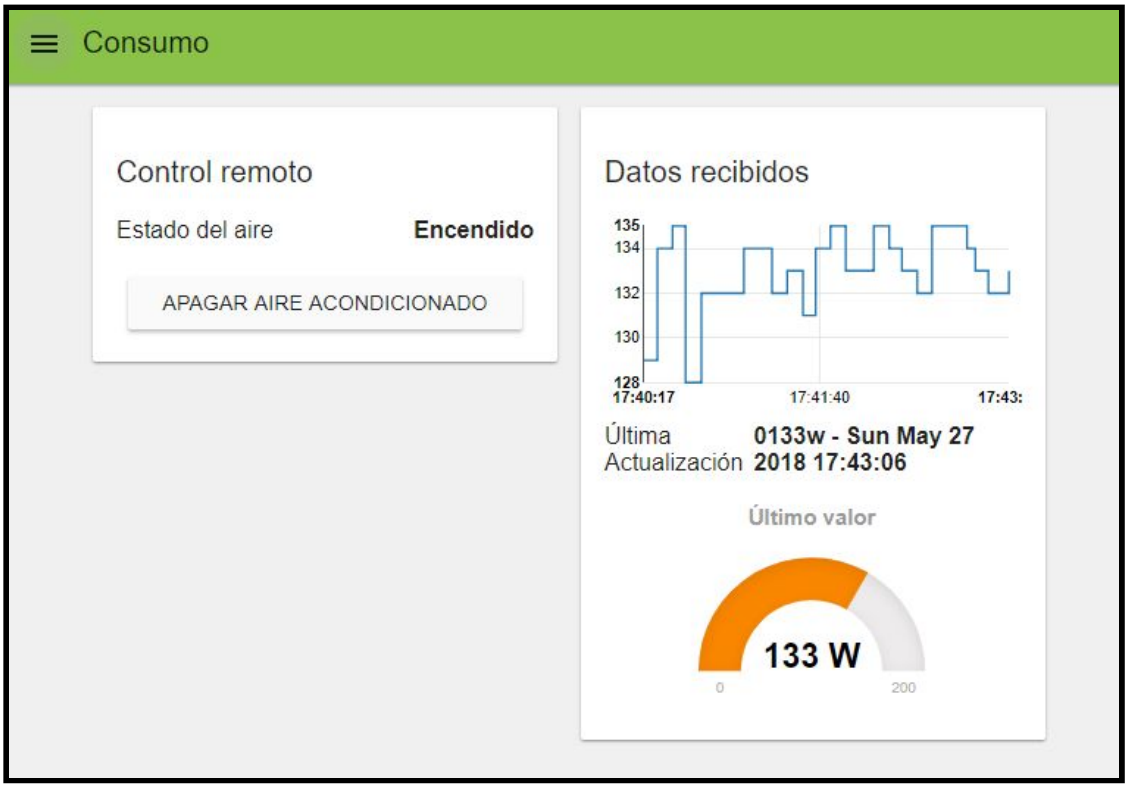


Figura 4.2.2.2. Screenshot del sitio web con valores ejemplo donde se visualizan los resultados.

En la pantalla se encuentra un estimador del estado del aire en base al valor que recibe de la medición inicial de consumo; Un botón de apagar aire acondicionado, el cual al presionarlo envía un caracter "j" al nodo configurado que debe realizar el apagado cuando detecta un payload con este contenido; Dos gráficos, uno del tipo línea y otro de tipo aguja; Y un texto que muestra el horario y valor del último dato recibido.

En la esquina superior izquierda se encuentra un botón de menú el cual sirve para navegar por los distintos proyectos que se puedan ir creando en el uso de la red LoRa.

4.2.3. Configuraciones extra

El sitio web está configurado para recibir con un nodo específico y enviar a otro específico también, en este caso los EUI son 00-80-00-00-04-00-08-cb y 00-80-00-00-00-00-d9-34 respectivamente, los cuales fueron nuestros nodos de prueba. Si se desea utilizar otros nodos se deben modificar en Node-Red:

- EUI del nodo "IF(nodo)" para filtrar correctamente el dato recibido.
- EUI del nodo "Transmisión LoRa" para decidir a quién enviar el apagado del aire.

Otro detalle a tener en cuenta es el tiempo durante el cual el dispositivo permanece en modo de hibernación, esto ocurre siempre luego de un envío para ahorrar energía. Dicho de otra manera significa cada cuanto tiempo debe enviar datos un nodo al gateway. Este es un factor clave para la larga duración de la batería y la elección de esta tecnología y no debe ser tomado a la ligera. Actualmente el código cargado en el xDot cuenta con un tiempo entre envíos de 10 segundos, el cual es muy corto, y se configuró de esta manera para realizar las pruebas sin intervalos de espera largos. Este tiempo está definido en la función "sleep_wake_rtc_or_interrupt" :

```
void sleep_wake_rtc_or_interrupt(bool deepsleep) {
  // in some frequency bands we need to wait until another channel is available before transmitting again
  // wait at least 10s between transmissions
  uint32_t delay_s = dot->getNextTxMs() / 1000;
  if (delay_s < 10) {
    delay_s = 10;
```

*Extraido del archivo "**dot_util.cpp**" incluido en las librerías del repositorio [21] https://os.mbed.com/users/unsamdev/code/Dot*

Modificando estos dos números marcados en rojo y subrayados podemos obtener distintos tiempos de hibernación del dispositivo. Sugerimos utilizar valores del orden de minutos o hasta horas dependiendo del uso que se quiera dar. Para el proyecto de visualización de consumo de energía sería correcto hacer una medición cada 10 o 15 minutos.

4.2.4. Seguridad

Como el gateway tiene una IP pública asignada para poder visualizar el sitio web desde internet, es importante habilitar también funciones de seguridad que se detallan a continuación:

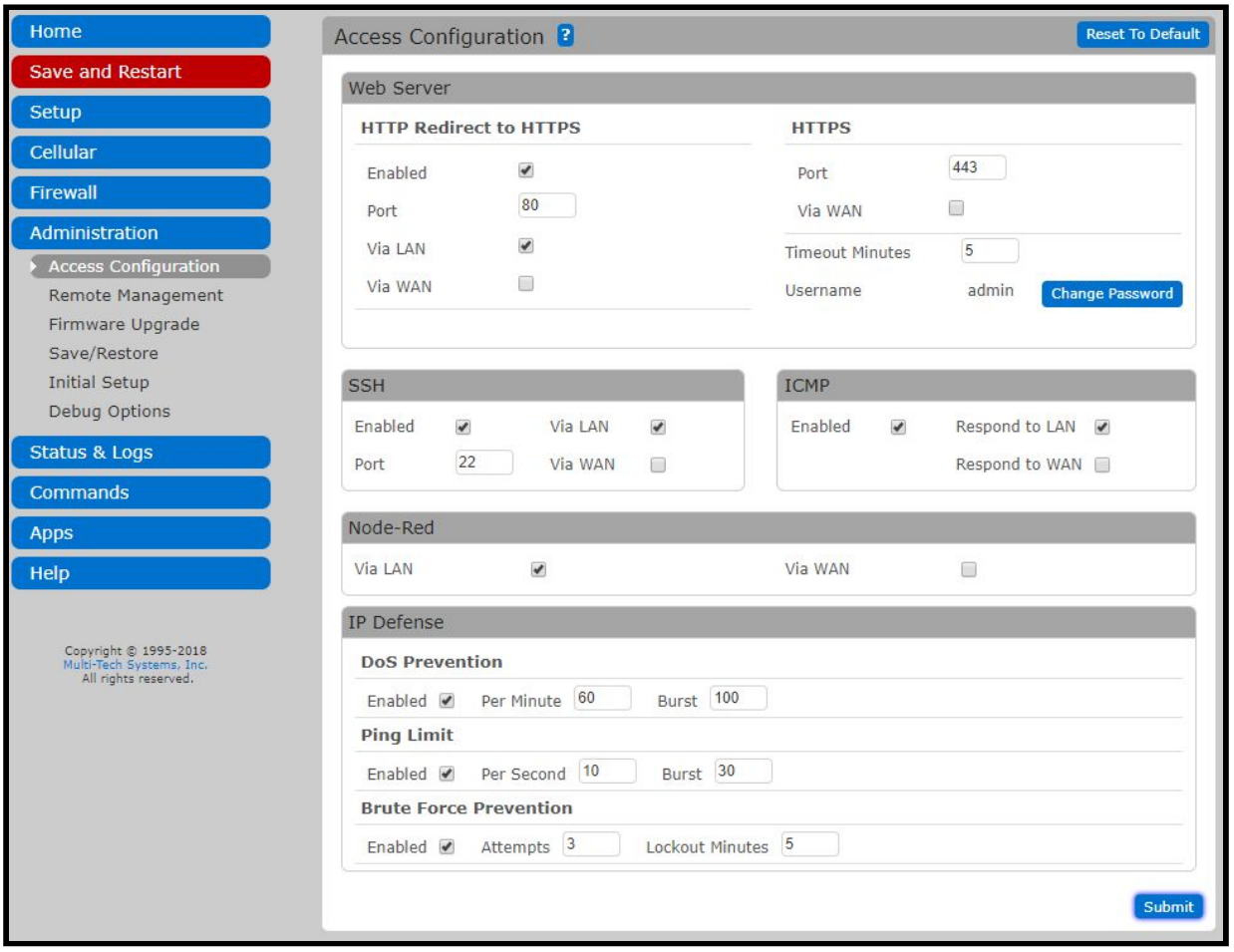


Figura 4.2.2.3. Screenshot de la configuración de acceso al gateway Conduit.

- **Redirección HTTP a HTTPS**: el tráfico que se genera al visualizar la web y/o configurar el gateway se encripta mediante llaves criptográficas por protocolo TLS.

- **Acceso via WAN**: permitir el acceso vía WAN, es decir desde una red no cableada, deshabilitada. Significa poder ingresar a la configuración desde una IP que no sea de la red interna de la universidad.

- **Acceso SSH**: permite el acceso mediante la consola serie, siempre por vía LAN.

- **ICMP**: permite el uso de ping al gateway, siempre vía LAN.

- **IP Defense**:

    - **DoS Prevention**: Prevención de denegación de servicio, bloquea los ataques de denegación de servicio, es decir, ataques cuyo funcionamiento es la continua petición de descargas legítimas, pero que al ser tantas y en tan poco tiempo saturan el procesamiento y/o la red del dispositivo. Habilitado

    - **Ping Limit**: Limita la cantidad de pings respondidos por segundo y por ráfagas. Habilitado.

    - **Brute Force Prevention**: Prevención de fuerza bruta, evita que se puedan probar combinaciones de contraseñas al azar o mediante el uso de diccionarios bloqueando la IP del emisor por un tiempo, luego de una cierta cantidad de fallos. Habilitado.

Una vez que tuvimos todo esto funcionando correctamente y segurizado, se procedió al análisis del alcance de la red y mediciones varias de nuestro proyecto.

4.3. Mediciones Realizadas

El proceso de medición se llevó adelante mediante el uso de una PC portátil a la cual se le anexó un nodo xDot el cual fue programado para enviar un valor constante al gateway y este se encarga de subirlo a la página web. Se trazó en un mapa de la Universidad de San Martín varias circunferencias concéntricas al tornavías de la misma. Una vez trazadas las circunferencias procedimos a elegir los puntos claves para realizar la medición, los mismos se pueden observar en la figura 4.3.2.

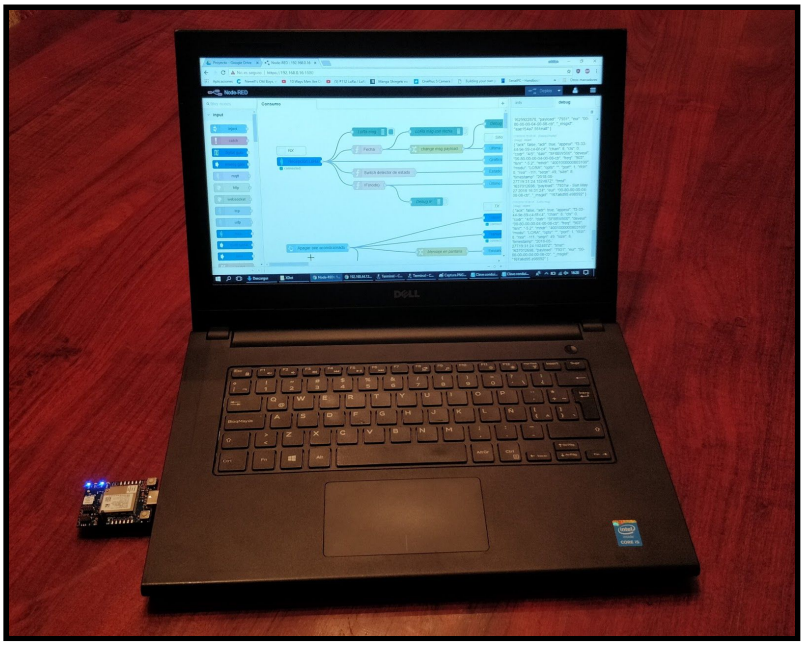


Figura 4.3.1. Fotografía de PC portátil utilizada para la medición con su nodo xDot conectado vía USB.

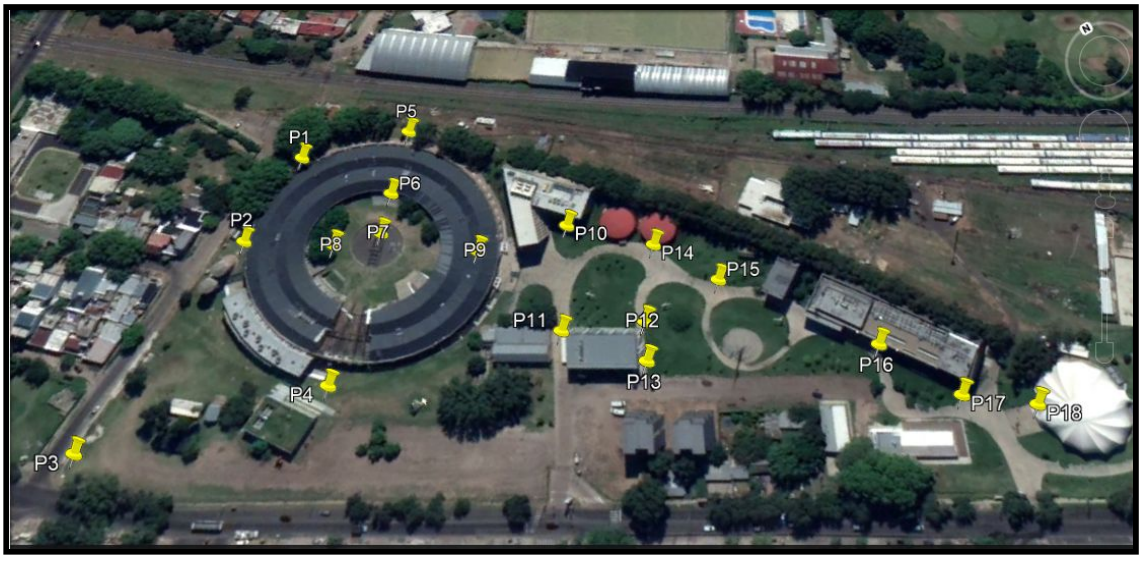


Figura 4.3.2 Mapa de los radios trazados y puntos de medición realizados.

El proceso de medición se realizó tomando el valor de RSSI de la señal en cada punto seleccionado tres veces y se calculó la media para cada uno. El RSSI (received signal strength indicator) es una escala de referencia (en relación a 1 mW) para medir el nivel de potencia de las señales recibidas por un dispositivo en las redes inalámbricas. La escala tiene al valor 0 como centro; representa 0 RSSI, o 0 dBm. La escala se expresa dentro de valores negativos; cuanto más negativo, mayor pérdida de señal.
En la tabla 4.3.1 se muestra cómo se interpretan los valores de RSSI. Cada fabricante establece una escala diferente para su chipset ya que el estándar 802.11 no posee ninguna especificación. Para el caso de la placa xDot que utiliza ARM los valores van desde 0 hasta -127.

Valor de la medición de RSSI	Interpretación
0	Señal ideal
-40 a -60	Señal idónea con tasas de transferencia estables
-60	Enlace bueno, con el cual se puede lograr una conexión estable al 80%.
-70	Enlace normal a bajo, es una señal medianamente buena, se puede sufrir problemas con lluvia y viento
-80 a -110	Señal mínima para el enlace. Puede ocurrir caídas, que se traducen en corte de comunicación.

Tabla 4.3.1 Referencia de valores de RSSI[19]

Se realizaron dos mediciones una con el nodo sin la antena y otra con la antena de 5 dBi. En la medición con la antena se realizó una mayor cantidad de mediciones ya que el alcance fue superior. Tal es el caso que en los puntos 3, 4, 13, 16, 17 y 18 solo se pudo medir con la antena de 5 dBi.

4.4. Ensayos con el proyecto de consumo energético

5. Resultados

5.1. Mediciones

Los resultados de la medición de potencia recibida, mencionada en la sección 4.3, se presentan en la tabla 5.1.1 a continuación:

Puntos	RSSI sin antena	RSSI con antena
1	-93,66	-65
2	-102,5	-88,66
3	-	-98,66
4	-	-79,33
5	-76,66	-50,33
6	-96,33	-68
7	-98,33	-59,66
8	-97	-71,33
9	-101,33	-69,66
10	-95	-81,33
11	-106	-94
12	-107	-92,66
13	-	-99,33
14	-105	-95,33
15	-107	-102,66
16	-	-96,33
17	-	-102,66
18	-	-101,33

Tabla 5.1.1. Valores de las mediciones realizadas dentra de la UNSAM relacionadas con la figura 4.3.2.

Como era de esperarse se observa una amplia variación del valor de señal con y sin antena, una diferencia promedio de 17,7 para cada punto de medición.

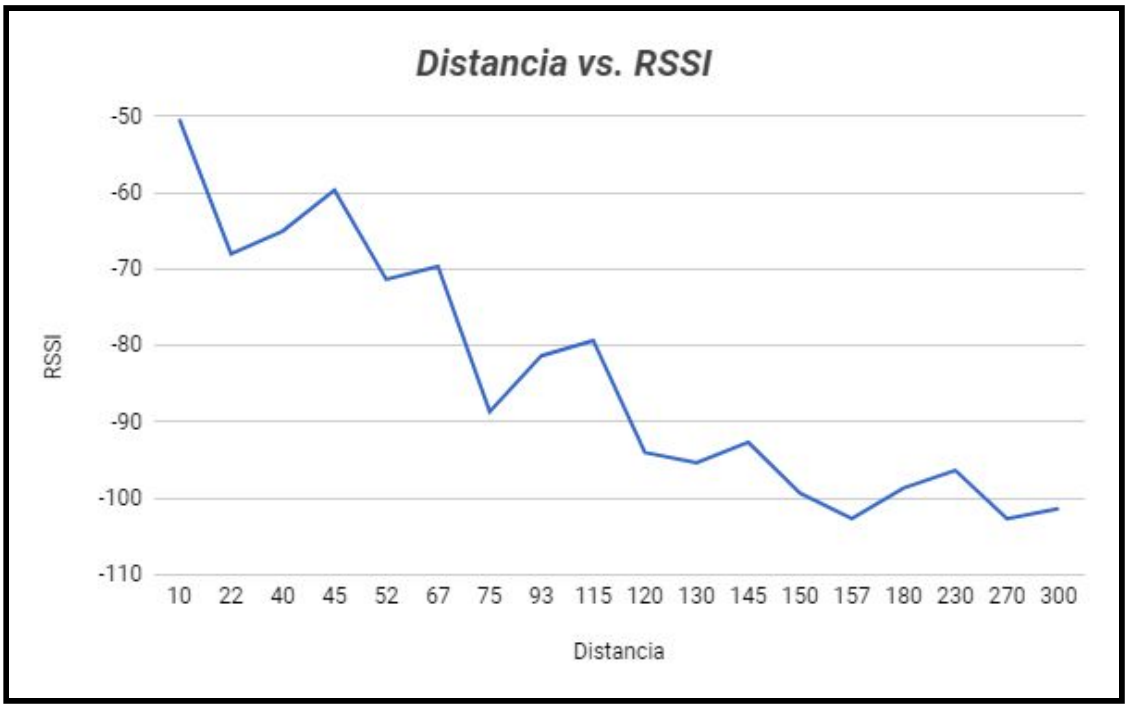


Figura 5.1.1. Gráfico de análisis distancia vs señal recibida.

Como podemos ver en la figura 5.1.1. la señal recibida se deteriora a medida que nos alejamos del dispositivo emisor como es de esperarse. Las variaciones encontradas se deben a que en los distintos puntos no tuvimos una línea de visión directa entre transmisor y receptor, por lo tanto la señal se ve afectada o beneficiada por los efectos de multitrayecto - es decir la propagación de una onda por distintos caminos y su posterior interferencia constructiva o destructiva - y desvanecimiento - el efecto de atenuación de una onda debido a condiciones climáticas, geográficas, obstáculos, etc.

5.2. Integración al proyecto de consumo energético

6. Conclusiones

Se logró realizar con éxito el despliegue y funcionamiento de la red LoRa dentro de la Universidad Nacional de San Martín. La cobertura no es total pero se consigue un rango aceptable para su uso, especialmente dentro del tornavías. Para mejorar este aspecto entendemos que la posición física del gateway no es la óptima y debería colocarse en una zona más central. Estableciendo el gateway dentro del edificio de rectorado se podría conseguir una mayor cobertura y posiblemente total dentro del predio de la universidad permitiendo así el uso de la red para futuros proyectos en todo el campus miguelete. Siendo más exigentes se podría colocar una antena exterior que permita eliminar los principales obstáculos que lo son hoy las paredes del tornavías donde se encuentra el gateway y su antena. También debemos tener en cuenta que dentro del edificio tornavías se realizan la mayor cantidad de actividades relacionadas a Electrónica y Telecomunicaciones (a pesar de que el uso de la red puede extenderse a cualquier actividad), por lo tanto dejar el gateway en el lugar que se encuentra actualmente también es una decisión correcta.

Además pudimos probar el uso de esta red en un proyecto en concreto, el cual es el de la medición de consumo eléctrico y el apagado de un aire acondicionado remotamente y estas pruebas fueron satisfactorias. Sin embargo este es sólo un ejemplo de las grandes posibilidades que el protocolo LoRa presenta, el cual esperamos que siga creciendo y utilizándose en la UNSAM.

Para poder realizar este proyecto aplicamos conocimientos adquiridos en diversas materias de nuestra carrera, y sin ellos no podríamos haber comprendido y solucionado los distintos problemas que surgieron a lo largo de este trabajo.

7. Agradecimientos

8. Glosario

ABP	Activación por personalización
ADR	Adaptative Data Rate
ADR_ACK_CNT	Adaptative Data Rate Acknowledge Count
ADR_ACK_DELAY	Adaptative Data Rate Delay
ADR_ACK_LIMIT	Adaptative Data Rate Limit
ADRACKReq	Adaptative Data Rate Acknowledge Request
AES	Advanced Encryption Standard
AppKey	Application Key
AppSKey	Application Session Key
ARM	Advanced RISC Machine
Beacon	Señal de faro
BW	Bandwidth o Ancho de banda
CFList	Channel Frequency List
CR	Tasa de código
CRC	Cyclic redundancy check
dBi	Decibel isotropic
dBm	Decibel milliwatt
DNS	Domain Name Server
DevAddr	Device Address
DevEUI	End Device Unique Identifier
DevNonce	Device Nonce
DL	Downlink
DLSettings	Downlink Settings
DoS	Denial of Service
DSSS	Direct Sequence Spread Spectrum
EUI	End-Device Unique Identifier
FCnt	Frame Counter
FCtrl	Frame Control
FHDR	Frame Header
FNwkSIntKey	Forward Network Session Key
FOpts	Frame Options
FOptsLen	Frame Options Length
FPending	Frame Pending
FPort	Frame Port
FRMPayload	Frame Payload Field
FSK	Frequency Shift Keying
Gateway	Puerta de enlace, concentrador, enrutador
GP	Ganancia de proceso
GPS	Global Positioning System
HTTPS	Hypertext Transfer Protocol with SSL (SSL es el precursor de TLS)
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISM	Industrial, Scientific and Medical Band
JoinEUI	Join EUI
JoinNonce	Join Nonce
Kbits	Kilobits
Kbps	Kilobits per Second

Khz	Kilohertz
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LR-WPAN	Low Rate Wide Personal Area Network
MAC	Media Access Control
MACPayload	Media Access Control Payload o carga útil
MHDR	MAC Header
Mhz	Megahertz
MIC	Mensaje de integridad de código
Multicast	Mensaje con múltiples destinos
mW	milliwatt
NbTrans	Number of Transmissions
NetID	Identificación de red
Nonce	Número que no se repite
NwkKey	Network Key
NwkSEncKey	Network Session Encryption Key
O-QPSK	Offset Quadrature Phase Shift Keying
OTA	Over The Air
OTAA	Over The Air Activation
PHDR	Physical Layer Header
PHDR_CRC	Physical Layer Header CRC
PHY	Physical Layer (OSI Model)
Ping slot	Ventana de recepción extra
Proxy	Servidor intermediario
Rb	Tasa de bits
R_C	Tasa de chips
RF	Radiofrequency
RP-SMA	Reverse Polarity - SubMiniature version A
Rs	Tasa de símbolos
Rx	Receiver / Reception
RxDelay	Reception Delay
S/N	Signal to Noise Ratio
SF	Spreading Factor
SNwkSIntKey	Serving Network Session Integrity Key
SSH	Secure Shell
SSL	Secure Socket Layer
Sub-Band	Sub-Banda
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
Ts	Período del símbolo
TTN	The Things Network
Tx	Transmitter / Transmission
UL	Uplink
Unicast	Mensaje con un único destino
UNSAM	Universidad Nacional de San Martín
WiFi	Wireless Fidelity

9. Bibliografía

[1] https://www.semtech.com/technology/lora, Noviembre 2017.
[2] https://en.wikipedia.org/wiki/Shannon%E2%80%93Hartley_theorem, Mayo 2018.
[3] https://standards.ieee.org/findstds/standard/802.15.4k-2013.html, Mayo 2018.
[4]https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf, Abril 2018.
[5] https://standards.ieee.org/findstds/standard/802.15.4-2006.html, Abril 2018.
[6] https://en.wikipedia.org/wiki/Message_authentication_code, Mayo 2018.
[7] http://standards.ieee.org/develop/regauth/tut/eui.pdf
[8] https://www.yoctoproject.org, Noviembre 2017.
[9] https://www.enacom.gob.ar/bandas-no-licenciadas_p680, Abril 2018.
[10]http://www.enacom.gob.ar/cuadro-de-atribucion-de-bandas-de-frecuencias-de-la-republica-argentina-cabfra-_p1588, Abril 2018.
[11]http://www.telesemana.com/blog/2017/03/07/enacom-impone-devolucion-de-espectro-y-metas-de-cobertura-para-refarming-de-nextel/, Abril 2018.
[12] https://os.mbed.com/compiler/, Noviembre 2017.
[13]https://os.mbed.com/teams/MultiTech/wiki/Using-Eclipse-and-mbed-5-for-MTS-Devices, Noviembre 2017.
[14] https://www.thethingsnetwork.org/, Noviembre 2017.
[15] https://mydevices.com/cayenne/features/, Diciembre 2017.
[16] https://www.npmjs.com/package/node-red-contrib-ui, Febrero 2018.
[17] https://nodered.org/, Febrero 2018.
[18] https://www.npmjs.com/, Diciembre 2017.
[19] https://es.wikipedia.org/wiki/Indicador_de_fuerza_de_la_se%C3%B1al_recibida, Marzo 2018.
[20] Sistemas de comunicaciones electrónicas, Tomasi, Cuarta Edición.
[21] https://os.mbed.com/users/unsamdev/code/Dot/, Junio 2018.
[22] https://es.wikipedia.org/wiki/Advanced_Encryption_Standard
[23] https://www.multitech.com/documents/publications/manuals/s000645.pdf, XDot Dev guide, Agosto 2017.
[24] http://www.multitech.net/developer/software/lora/glossary/, Mayo 2018.
[25]http://www.multitech.net/developer/software/lora/lora-network-server/mqtt-messages/, Junio 2018.
[26]https://www.lora-alliance.org/sites/default/files/2018-04/lorawantm_regional_parameters_v1.1rb_-_final.pdf, Julio 2018.

Anexos

9.1. Código utilizado para xDot (Disponible en fuente [21])

Escrito en C++, para el envío de datos desde el nodo, el cual permitió el testeo del correcto funcionamiento de la red y sentar la base para agregar cualquier código de envío de datos mediante red LoRa.

```
#include "dot_util.h"
#include "RadioEvent.h"

// Definicion de valores de la red
static std::string network_name = "Lora-Net-UNSAM";
static std::string network_passphrase = "A5Jd5jKtt5L3";
static uint8_t frequency_sub_band = 1;
static bool public_network = false;
static uint8_t join_delay = 1;
static uint8_t ack = 0;
static bool adr = true;

static bool deep_sleep = true;

// Definicion de variables
std::vector<uint8_t> recibido;
std::vector<uint8_t> dato;
std::vector<uint8_t> hexa;
int32_t ret = mDot::MDOT_ERROR;

mDot* dot = NULL;  // agregamos el NULL (porque sino no andan varias
funciones de los modulos)
lora::ChannelPlan* plan = NULL;

Serial pc(USBTX, USBRX); // tx, rx

// Definiciones de la placa
AnalogIn corriente(GPIO2);
DigitalOut dout(LED1);

int main() {

    RadioEvent eventos; // Si lo comentamos podemos borrar el include?
(Custom event handler for automatically displaying RX data)

    pc.baud(115200);

//  mts::MTSLog::setLogLevel(mts::MTSLog::TRACE_LEVEL); // Log level,
descomentar si es necesario
```

```
    plan = new lora::ChannelPlan_US915();
    assert(plan); // Chequea que el plan sea el correcto

    dot = mDot::getInstance(plan);
    assert(dot); // chequea que la placa sea la correcta

    dot->setEvents(&eventos); // (attach the custom events handler)

    if (!dot->getStandbyFlag()) {

        ///start from a well-known state // Verificar si es necesario
        ///logInfo("defaulting Dot configuration");
        ///dot->resetConfig();
        ///dot->resetNetworkSession();
        if (dot->getJoinMode() != mDot::OTA) {   // Configuramos el modo
de unirse a la red LoRa
            logInfo("Cambiando el modo de conexion a la red OTA\n");
            if (dot->setJoinMode(mDot::OTA) != mDot::MDOT_OK) {
            logInfo("Error al cambiar el modo de conexion a la red
OTA\n");
            }
        }

        update_ota_config_name_phrase(network_name, network_passphrase,
frequency_sub_band, public_network, ack); // Pasamos los datos al modo
OTA
        update_network_link_check_config(3, 5); // Pido un chequeo de
link cada 5 paquetes, si fallan 3 respuestas = link caido
        dot->setAdr(adr); // Habilito la adaptacion de tasa de
transmision
        dot->setJoinDelay(join_delay); // Configuro el delay del join

        if (!dot->saveConfig()) {
            logInfo("Error al guardar la configuracion");
        }

        ///display_config();
    }
    else {
        logInfo("Recuperando datos de la red de la memoria");
        dot->restoreNetworkSession();
    }

    while (true) {
        if (!dot->getNetworkJoinStatus()) {
            ret = dot->joinNetwork(); // Nos unimos a la red LoRa
            if (ret != mDot::MDOT_OK) logInfo("Error al unirse a la red
LoRa\r\n");
            else logInfo("Se ha unido a la red LoRa\r\n");
```

```
        }

// Aqui se pega el programa a ejecutar en el nodo xDot para la medicion
de consumo
        int i = 0;
        int k = 0;
        float samples = 0;
        float tension = 220;
        float potencia = 0;
        float patron = 0.1;
        uint16_t Psend = 0;

        for(i=0; i<500; i++) {
            samples += corriente.read();
        }
        samples /= 500; // Escalado
        samples *= 3.3; // Convertido a corriente
        samples *= 2;
        potencia = samples*tension;
        if(samples<patron) {
            samples = 0;
            tension = 0;
            dout = 0;
            Psend = 0;
        }
        else {
            tension = 220;
            dout = 1;
        }
        Psend = potencia*100;
// Aqui finaliza el programa a ejecutar en el nodo xDot para la medicion
de consumo

        for(k=1000; k>0; k/=10) {
            dato.push_back((Psend/k) + 48);
            Psend -= (Psend/k)*k;
        }
        send_data(dato);
        dato.clear();

        if (dot->recv(recibido) == mDot::MDOT_OK) {
            if (recibido[0]=='j') logInfo("Dato correcto\n"); // Si
recibo 'j' el dato es correcto
            else logInfo("Dato erroneo\n");
            recibido.clear();
        }
        else logInfo("Nada Recibido\n\r");
            logInfo("*****************************\r\nCORRIENTE: %.2f
[A]\r\nTENSION: %.2f [V]\r\nPOTENCIA: %.2f
[W]\r\n*************************************\r\n",samples,tension,potenci
```

```
a);
            if (deep_sleep) {
                logInfo("Guardando datos de la red a la memoria");
                dot->saveNetworkSession();
            }
        sleep_wake_rtc_or_interrupt(deep_sleep); // Paso a estado de
hibernacion durante x tiempo o boton wake
    }
}
```

9.2. Código utilizado para Node-Red

[{"id":"c36e1664.7cdd28","type":"ui_tab","name":"Consumo","icon":"settings_power","order":"1"},{"id":"c575e6b6.a1b9e8","type":"ui_button","tab":"c36e1664.7cdd28","name":"Apagar aire acondicionado","payload":"j","topic":"Boton presionado","group":"Control remoto","order":"4","x":188,"y":480,"z":"91d7d5b3.1bed08","wires":[["bc2c5675.580a98","1395a8d6.3f7837","68c0c797.b475c8","135ab579.009ffb"]]},{"id":"39406071.ab331","type":"lora in","name":"Recepción LoRa","datatype":"utf8","x":136.50000762939453,"y":177,"z":"91d7d5b3.1bed08","wires":[["8c894089.c1f7e","96bda23e.341fa","3a6cd332.8f337c","d5e5dbe8.786498","51c505d7.933a2c"]]},{"id":"bc2c5675.580a98","type":"lora out","name":"Transmisión LoRa","eui":"00800000040008cb","payload":"msg.payload","ack":false,"port":"","x":865,"y":380,"z":"91d7d5b3.1bed08","wires":[]},{"id":"be3a568a.9ed218","type":"comment","name":"Sitio Web","info":"https://192.168.44.12:1880/ui","x":835,"y":100,"z":"91d7d5b3.1bed08","wires":[]},{"id":"70610b1f.d392d4","type":"ui_text","tab":"c36e1664.7cdd28","name":"Última Actualización","group":"Datos recibidos","order":"3","format":"{{msg.payload}}","x":871.5000076293945,"y":137,"z":"91d7d5b3.1bed08","wires":[]},{"id":"8c894089.c1f7e","type":"function","name":"Fecha","func":"msg.payload = msg.payload + \"w - \" + new Date().toString();\nreturn msg;","outputs":1,"noerr":0,"x":370.00000762939453,"y":137,"z":"91d7d5b3.1bed08","wires":[["d0d8f006.108e6","5f88d03d.dcfa"]]},{"id":"a6aeb4c9.03f268","type":"debug","name":"Debug Display","active":false,"console":"false","complete":"true","x":855,"y":52.99999809265137,"z":"91d7d5b3.1bed08","wires":[]},{"id":"d0d8f006.108e6","type":"change","name":"","rules":[{"t":"change","p":"payload","from":" GMT-0300 (ART)","to":"","re":false}],"action":"","property":"","from":"","to":"","reg":false,"x":613.0000076293945,"y":137,"z":"91d7d5b3.1bed08","wires":[["70610b1f.d392d4","a6aeb4c9.03f268"]]},{"id":"96bda23e.341fa","type":"function","name":"Switch detector de estado","func":"var valor = msg.payload;\nvalor = parseInt(msg.payload);\n\nif(valor > 10) { msg.payload = \"Encendido\"; }\nelse { msg.payload = \"Apagado\"; }\n//if(valor < 10) { msg.payload = \"Apagado\"; }\nreturn msg;","outputs":1,"noerr":0,"x":429.00000762939453,"y":217,"z":"91d7d5b3.1bed08","wires":[["21750e8c.693d42"]]},{"id":"3a6cd332.8f337c","type":"debug","name":"LoRa msg","active":false,"console":"false","complete":"true","x":380.50000762939453,"y":77,"z":"91d7d5b3.1bed08","wires":[]},{"id":"1395a8d6.3f7837","type":"debug","name":"Debug Boton","active":false,"console":"false","complete":"true","x":429.5,"y":560,"z":"91d7d5b3.1bed08","wires":[]},{"id":"d5e5dbe8.786498","type":"ui_chart","tab":"c36e1664.7cdd28","name":"Gráfico","group":"Datos recibidos","order":1,"interpolate":"step-after","nodata":"Sin Conexión","removeOlder":"30","removeOlderUnit":"60","x":831.0000076293945,"y":177,"z":"91d7d5b3.1bed08","wires":[[],[]]},{"id":"5f88d03d.dcfa","type":"debug","name":"LoRa msg con

fecha","active":false,"console":"false","complete":"true","x":612.5000076293945,"y":77,"z":"91d7d5b3.1bed08","wires":[]},{"id":"f3c147ab.f66888","type":"ui_toast","name":"Enviando señal de apagado","x":895,"y":500,"z":"91d7d5b3.1bed08","wires":[]},{"id":"68c0c797.b475c8","type":"change","name":"Mensaje en pantalla","rules":[{"t":"set","p":"payload","to":"Enviando señal de apagado"}],"action":"","property":"","from":"","to":"","reg":false,"x":631,"y":500,"z":"91d7d5b3.1bed08","wires":[["f3c147ab.f66888"]]},{"id":"21750e8c.693d42","type":"ui_text","tab":"c36e1664.7cdd28","name":"Estado del aire","group":"Control remoto","order":"1","format":"{{msg.payload}}","x":855.0000076293945,"y":217,"z":"91d7d5b3.1bed08","wires":[]},{"id":"1c0807a4.ec4a58","type":"ui_text","tab":"c36e1664.7cdd28","name":"","group":"Datos recibidos","order":"4","format":"","x":1090,"y":360,"z":"91d7d5b3.1bed08","wires":[]},{"id":"a3a7527d.a123b","type":"ui_text","tab":"c36e1664.7cdd28","name":"","group":"Datos recibidos","order":"2","format":"","x":1090,"y":320,"z":"91d7d5b3.1bed08","wires":[]},{"id":"98ea7418.3004f8","type":"ui_text","tab":"c36e1664.7cdd28","name":"","group":"Control remoto","order":"2","format":"","x":1090,"y":400,"z":"91d7d5b3.1bed08","wires":[]},{"id":"51c505d7.933a2c","type":"function","name":"IF(nodo)","func":"if (msg.eui==\"00-80-00-00-04-00-08-cb\"){return msg}\nelse{}","outputs":1,"noerr":0,"x":375,"y":260,"z":"91d7d5b3.1bed08","wires":[["fb2a9b68.25b3b8","d6ab835.ad6f58"]]},{"id":"fb2a9b68.25b3b8","type":"debug","name":"Debug IF","active":false,"console":"false","complete":"true","x":577.5,"y":320,"z":"91d7d5b3.1bed08","wires":[]},{"id":"d6ab835.ad6f58","type":"ui_gauge","tab":"c36e1664.7cdd28","name":"Último valor","group":"Datos recibidos","order":"6","format":"{{value}} W","min":0,"max":"200","x":845.5,"y":260,"z":"91d7d5b3.1bed08","wires":[]},{"id":"e7aa9beb.9ae718","type":"comment","name":"Divisiones","info":"Los nodos que estan aca abajo son\nespacios en blanco para separar los graficos","x":1097,"y":280,"z":"91d7d5b3.1bed08","wires":[]},{"id":"135ab579.009ffb","type":"lora out","name":"Transmisión LoRa","eui":"008000000000d934","payload":"msg.payload","ack":false,"port":"","x":865,"y":440,"z":"91d7d5b3.1bed08","wires":[]},{"id":"b9a9b686.6d35d8","type":"comment","name":"TX","info":"","x":830,"y":340,"z":"91d7d5b3.1bed08","wires":[]},{"id":"4ebfaff8.45a9a","type":"comment","name":"RX","info":"","x":110,"y":140,"z":"91d7d5b3.1bed08","wires":[]},{"id":"1d17eeac.0dc541","type":"ui_text","tab":"c36e1664.7cdd28","name":"","group":"Control remoto","order":"3","format":"","x":1090,"y":440,"z":"91d7d5b3.1bed08","wires":[]}]

9.3. Actualización Firmware a versión AEP 1.4.16

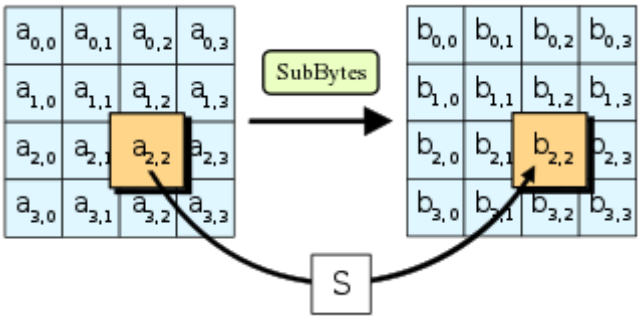
9.4. Encriptación AES [22]

Es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) como FIPS PUB 197 de los Estados Unidos (FIPS 197) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Se transformó en un estándar efectivo el 26 de mayo de 2002. Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica.

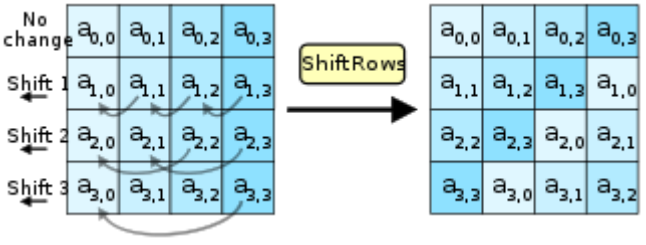
Pseudocódigo

- Expansión de la clave usando el esquema de claves de Rijndael.
- Etapa inicial:
  1. AddRoundKey

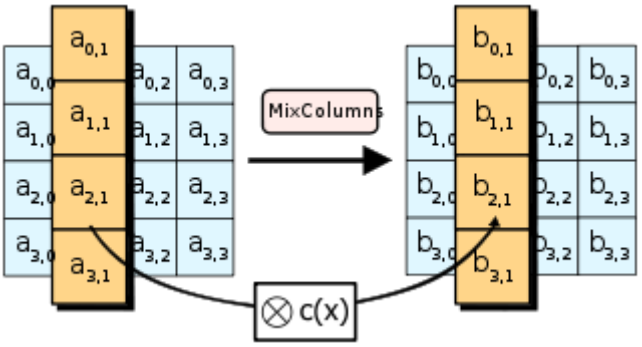
- Rondas:
  1. SubBytes — en este paso se realiza una sustitución no lineal donde cada byte es reemplazado con otro de acuerdo a una tabla de búsqueda.



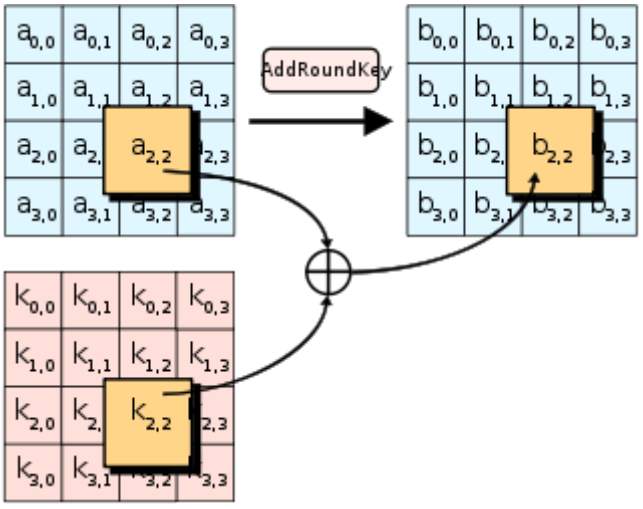
  2. ShiftRows — en este paso se realiza una transposición donde cada fila del «state» es rotada de manera cíclica un número determinado de veces.



3. MixColumns — operación de mezclado que opera en las columnas del «state», combinando los cuatro bytes en cada columna usando una transformación lineal.



4. AddRoundKey — cada byte del «state» es combinado con la clave «round»; cada clave «round» se deriva de la clave de cifrado usando una iteración de la clave.



- Etapa final:
    1. SubBytes
    2. ShiftRows
    3. AddRoundKey

9.5. Asignación de frecuencias por país [26]

País	*Banda de frecuencias*
Afghanistan	None
Albania	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Algeria	433.05 - 434.79 MHz
	870-876MHz
	880-885MHz
	915 - 921 MHz
	925 - 926 MHz
Andorra	433.05 - 434.79 MHz
	863 - 870 MHz
Armenia	863 - 870 MHz
	433.05 - 434.79 MHz
Argentina	902 - 928 MHz (915-928 MHz usable)
Austria	433.05 - 434.79 MHz
	863 - 870 MHz
Australia	915 - 928 MHz
Azerbaijan	433.05 - 434.79 MHz
	863 - 868 MHz
Bahrain	862 - 870MHz
Bangladesh	433.05 - 434.79 MHz
	818 - 824 MHz
	863 - 869 MHz
	925.0 - 927.0 MHz
Belarus	433.05 - 434.79 MHz
	864.4 - 868.6 MHz
	869-869.2MHz
Belgium	433.05 - 434.79 MHz
	863 - 870 MHz
Burma (Myanmar)	433 - 435 MHz
	866 - 869MHz
	919 - 923 MHz
Bolivia	915 - 930 MHz
Bosnia and Herzegovina	433.05 - 434.79 MHz

	863 - 870 MHz
Botswana	None
Brazil	902 - 907.5 MHz
	915 - 928 MHz
	433 - 435 MHz
Brunei Darussalam	866 - 870 MHz
	920 - 925 MHz
Bulgaria	433.05 - 434.79 MHz
	863 - 870 MHz
Cambodia	866 - 869 MHz
	923 - 925 MHz
Cameroon	None
Canada	902 - 928 MHz
Chile	902 - 928 MHz (915-928MHz usable)
China	920.5 - 924.5 MHz
	779 - 787 MHz
	470 - 510 MHz
	433.05 - 434.79 MHz
	314-316 MHz
	430 - 432 MHz
	840 - 845 MHz
Colombia	902 - 928 MHz
Congo Rep.	None
Costa Rica	920.5 - 928 MHz
Croatia	433.05 - 434.79 MHz
	863 - 870 MHz
Cuba	433.05 - 434.79 MHz
	915 - 921 MHz
Cyprus	433.05 - 434.79 MHz
	863 - 870 MHz
Czech Republic	433.05 - 434.79 MHz
	863 - 870 MHz
Denmark	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Dominican Republic	915 - 928 MHz
Ecuador	902 - 928 MHz
Egypt	433.05 - 434.79 MHz

	863 - 876 MHz
Estonia	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Finland	433.05 - 434.79 MHz
	863 - 873 MHz
France	433.05 - 434.79 MHz
	863 - 870 MHz
Georgia	None
Germany	433.05 - 434.79 MHz
	863 - 870 MHz
Ghana	None
Greece	433.05 - 434.79 MHz
	868 - 870 MHz
Guatemala	902 - 928 MHz (915-928 MHz usable)
Haiti	None
Honduras	915-928 MHz
Hong Kong	433.05 - 434.79 MHz
	865 - 868 MHz
	920 - 925 MHz
Hungary	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Iceland	433.05 - 434.79 MHz
	863 - 873 MHz
India	865 - 867 MHz
Indonesia	923 - 925 MHz
Iraq	None
Iran	433.05 - 434.79 MHz
	863 - 873 MHz
	915 - 918 MHz
Ireland	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Israel	433.05 - 434.79 MHz
	915 - 917 MHz
Italy	433.05 - 434.79 MHz
	863 - 870 MHz
Ivory Coast	None

Jamaica	902 - 928 MHz (915-928 MHz usable)
Japan	920.6 - 928.0 MHz (steps of 200kHz)
	920.8 - 927.8 MHz (steps of 600kHz)
Jordan	865 - 868 MHz
Kazakhstan	433.05 - 434.79 MHz
Kenya	None
Korea (DPR)	None
Kuwait	433.05 - 434.79 MHz
Kyrgyz Republic	None
Laos	433 - 435 MHz
	862 - 875 MHz
	923 - 925 MHz
Latvia	433.05 - 434.79 MHz
	863 - 870 MHz
Lebanon	433 - 435 MHz
	862 - 870 MHz
Liechtenstein	433.05 - 434.79 MHz
	863 - 873 MHz
Libya	None
Lithuania	433.05 - 434.79 MHz
	863 - 870 MHz
Luxembourg	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Macao	None
Macedonia, FYR	433.05 - 434.79 MHz
	863 - 870 MHz
Malaysia	433 - 435 MHz
	919 – 924 MHz
Maldives	None
Malta	433.05 - 434.79 MHz
	863 - 870 MHz
Mauritius	None
Mexico	902 - 928 MHz
Moldova	433.05 - 434.79 MHz
	863 - 870 MHz
Mongolia	None
Montenegro	433.05 - 434.79 MHz
	863 - 870 MHz
Morocco	433.05 - 434.79 MHz

	867.6 - 869 MHz
Netherlands	433.05 - 434.79 MHz
	863 - 870 MHz
New-Zealand	915 - 928 MHz
	819 - 824 MHz
	864 - 870MHz
	433.05 - 434.79 MHz
Nicaragua	902 - 928 MHz (915-928 MHz usable)
Nigeria	863 - 870 MHz
Norway	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Oman	433.05 - 434.79 MHz
	863 - 870 MHz
Pakistan	433.05 - 434.79 MHz
	865 - 869 MHz
	900 - 925 MHz
Panama	902 - 928 MHz
Paraguay	433.05 - 434.79 MHz
	915 - 928 MHz
Peru	915 - 928 MHz
Papua New Guinea	915 - 925 MHz
Philippines	915 - 918 MHz
	868 – 869.2 MHz
	869.7 - 870 MHz
	433.05 - 434.79 MHz
Poland	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Portugal	433.05 - 434.79 MHz
	863 - 870 MHz
Qatar	433.05 - 434.79 MHz
	868 - 868.6 MHz
	868.7 - 869.2 MHz
	869.4 - 869.65 MHz
	869.7 - 870 MHz
Romania	433.05 - 434.79 MHz
	863 - 870 MHz
Russian federation	866 - 868 MHz (Licensed)

	864 - 865 MHz
	868.7 - 869.2 MHz
	433.075 - 434.75 MHz
	916 - 921 MHz (Licensed)
Salvador	915-928
Saudi Arabia	863 - 870 MHz
	433.05 - 434.79 MHz
Senegal	None
Serbia	433.05 - 434.79 MHz
	863 - 870 MHz
Singapore	920 - 925 MHz
	433.05 - 434.79 MHz
	866 - 869 MHz
Slovak Republic	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
Slovenia	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
South Africa	433.05 - 434.79 MHz
	865 – 868.6 MHz
	868.7 – 869.2 MHz
	869.4 – 869.65 MHz
	869.7 – 870 MHz
	915 - 921 MHz
South Korea	917 - 923.5 MHz
Spain	433.05 - 434.79 MHz
	863 - 870 MHz
Sri Lanka	433.05 - 434.79 MHz
Sudan	None
Sweden	433.05 - 434.79 MHz
	868 - 870 MHz
Switzerland	433.05 - 434.79 MHz
	863 - 873 MHz
Syrian Arab Rep.	None
Taiwan	920 - 925 MHz
Tajikistan	None
Tanzania	None
Thailand	433.05 - 434.79 MHz

	920 - 925 MHz
Trinidad and Tobago	None
Tunisia	433.05 - 434.79 MHz
	868 – 868.6 MHz
	868.7 – 869.2 MHz
	869.4 – 869.65 MHz
	869.7 – 870 MHz
Turkey	433.05 - 434.79 MHz
	863 - 870 MHz
Turkmenistan	None
Uganda	433.05 - 434.79 MHz
	865 - 867.6 MHz
	869.25 - 869.7 MHz
	923 - 925 MHz
Ukraine	433.05 - 434.79 MHz
	863 - 865 MHz
	868 - 868.6 MHz
United Arab Emirates	433.05 - 434.79 MHz
	863 - 870 MHz
	870 - 875.8 MHz
	915 - 921 MHz
United Kingdom	433.05 - 434.79 MHz
	863 - 873 MHz
	918 - 921 MHz
United States	902 - 928 MHz
Uruguay	902 - 928 MHz (915 - 928 MHz usable)
Uzbekistan	433.05 - 434.79 MHz
Venezuela	922 - 928 MHz
Vietnam	433.05 - 434.79 MHz
	863 - 870 MHz
	918 - 923 MHz
Yemen, Rep.	None
Zimbabwe	None