



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Implementación de un transmisor AIS de bajo costo basado en SDR

TESIS PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE LA REPÚBLICA POR

Romina Belén García Camargo, Máximo Pirri Fernández

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERÍA EN SISTEMAS DE COMUNICACIÓN.

TUTOR

Gonzalo Belcredi Universidad de la República
Claudina Rattaro Universidad de la República

TRIBUNAL

Pablo Belzarena Universidad de la República
Federico La Rocca Universidad de la República
Florencia Blasina Universidad de la República
Nicolás Antoniello ICANN, miembro honorario de ADES
Favio Masson Universidad Nacional del Sur

Montevideo
lunes 20 mayo, 2024

Implementación de un transmisor AIS de bajo costo basado en SDR, Romina Belén
García Camargo, Máximo Pirri Fernández.

ISSN 1688-2806

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Contiene un total de 147 páginas.
Compilada el lunes 20 mayo, 2024.
<http://iie.fing.edu.uy/>

I am not afraid of storms, for I am learning how
to sail my ship.

LOUISA MAY ALCOTT

Esta página ha sido intencionalmente dejada en blanco.

Agradecimientos

En primer lugar, y en conjunto, queremos agradecer a nuestra casa de estudios, la Facultad de Ingeniería de la Universidad de la República. Vamos a llevar con nosotros los recuerdos de cinco años de desafíos y aprendizajes por el resto de nuestras vidas. También queremos expresar nuestra gratitud hacia el Instituto de Ingeniería Eléctrica que incluso en fechas festivas abrió sus puertas para permitir el avance en este proyecto. El IIE contribuyó con un papel protagónico a nuestra formación como estudiantes, docentes y personas. Agradecemos a todos y cada uno de los docentes y compañeros con los que nos cruzamos. Damos especiales gracias a Gonzalo Belcredi, nuestro tutor de proyecto, por sus aportes detallados y por su capacidad increíble de detectar problemas que nosotros mismos causamos. Damos gracias especiales también a Claudina Rattaro, nuestra tutora de proyecto y directora de carrera. En lo que respecta a su rol como tutora, damos gracias por su capacidad de abstracción de ideas difíciles, por sus correcciones a la última coma y por su mirada crítica a nuestro trabajo. En cuanto a su rol como directora, estamos convencidos de que su energía, apoyo y empatía fueron factores claves para lo mucho que disfrutamos el pasaje por facultad. Agradecemos también a ADES, la Institución Honoraria de Salvamento, por plantar como semilla la idea de este proyecto. Nos gustaría dar gracias al Servicio de Iluminación y Balizamiento de la Armada por facilitarnos el acceso a datos y a visitas de gran interés.

Pasando a agradecimientos individuales, yo, Romina, escribo. Dar gracias es una actividad que disfruto mucho e intento implementar a diario. En lo que respecta a los últimos cinco años de mi vida, tengo muchas gracias por dar.

Estoy firmemente convencida de que a la primera persona a la que quiero agradecer es a mí misma. No puedo evitar emocionarme al mirar hacia atrás y ver todo lo que logré. Al agradecerme a mí, tengo que mencionar a las personas que me hacen ser quien soy y que me permitieron llegar hasta acá. Doy gracias a mis padres, Analía y Eduardo, que me llevaron a facultad en días de lluvia, que me prepararon la merienda por veinte años y que confiaron en que iba a superar cada desafío. Agradezco a mi hermano, Gonzalo, por ser el principal referente en mi vida en lo que respecta a conocimientos y estudios. Agradezco a mi hermana, Evangelina, por inspirarme siempre a ser mi mejor versión, por escucharme repetir resúmenes durante toda la carrera y por dejarme compartir mi vida con ella.

A continuación quiero agradecerle a Máximo. No creo haberme cruzado nunca con alguien con quien aprender se me haga tan disfrutable y divertido. Como él me dijo una vez, tenemos la seguridad de que todos los problemas que afrontamos juntos van a ser solucionados. Gracias por compartir todos estos años de proyectos,

informes, llamadas de Zoom y reuniones en oficinas. Me alegra poder decir que somos amigos, además de ser compañeros de estudios, y que lo vamos a seguir siendo. Estoy segura de que voy a terminar volviendo a vos cada vez que necesite ayuda para solucionar problemas que sean solo míos.

Se me suele hacer difícil separar mi vida de mi carrera, porque hoy en día los considero casi que lo mismo. Cuando pierdo el foco, sé que cuento con amigos y amigas para volver a encontrarme. Quiero dar las gracias a Abril, Luhana y Hanna, por ser mis principales consejeras y por conocer a todas las personas de mi vida (aunque no las hayan visto nunca). Agradezco también a Maximiliano por sus mensajes ocurrentes y por su capacidad increíble para distraerme. Me siento muy agradecida de tenerlos a ellos y a muchas otras personas que siguen y apoyan mi progreso. Sin su confianza en el día a día ninguna proyección a cinco años hubiera sido posible.

Si la separación de mi vida y de mi carrera es difícil, no saben lo que implica introducir a mi trabajo en la cuestión. No creo que exista un lugar en el que haya aprendido tanto y de tantas formas distintas como en el Instituto de Ingeniería Eléctrica. Me permitió rodearme de personas diferentes a mí con las que encontré igualdades, y generé amistades que sé que van a durar mucho tiempo. Agradezco primero a quienes me posibilitaron entrar: a Claudina, por su apoyo incondicional, por contarme del llamado e insistir con que me presentara, y a Federico, por ser jefe de mi departamento y un gran docente, y enseñarme a respetarlo y admirarlo sin hacerme sentir varios grados por debajo de él. Doy gracias a mis guías desde el día 1: Martín, Lucas y Camilo. Me siento muy afortunada de haber conocido y de poder compartir el IIE con ustedes. Gracias a Juli, por acompañarme desde el liceo y ser una constante en mi vida en estos tiempos de variaciones. Está demás estudiar y trabajar contigo. Luego de casi tres años en el IIE, muchas otras personas se sumaron a mi equipo de trabajo y meriendas. Leo, Mari, Nacho, Emi, Diego, Emile, Gastón, Santi. Son un factor importantísimo en lo que respecta a cuanto me encanta mi trabajo. Experiencias en el exterior me permitieron sumar fuerzas en el último año de la carrera, como Samar y Nacho. Gracias por darme calma y por evitar que me pierda entre estrés y caos.

Finalmente, a quien sea que esté leyendo esto. Gracias por darle una oportunidad a mi trabajo. Espero también sientas que vale la pena.

Luego de un agradecimiento poético que no creo logre igualar me toca a mí, Máximo, escribir algunas palabras. Si bien no soy una persona que se caracterice por expresar sus sentimientos abiertamente, de vez en cuando toca devolver el cariño que la gente le da a uno de formas diversas.

Puede ser un poco cliché pero mi primer agradecimiento es para mi familia. No voy a mencionar a todos uno por uno porque de seguro alguno queda afuera y no sería justo pero creo que sí hay cuatro personas sin las que este trayecto no hubiera sido el mismo. En primer lugar a mis padres, que desde el día uno me dieron la libertad de elegir mi camino profesional. Desde que tengo memoria siempre están para mí, con su preocupación en exceso y chistes tontos pero nunca permitieron que me falte nada. Gracias a ellos tuve oportunidades increíbles que fueron parte fundamental de que hoy pueda estar escribiendo estos párrafos de

agradecimiento con un grado casi finalizado. No soy de expresarlo seguido pero nunca podré agradecerles lo suficiente ni devolver todo el amor que me dieron. No me alcanzan las palabras que conozco para darles las gracias pero sepan que son y serán un ejemplo como personas y padres.

En segundo lugar a mi hermana, Joaquina, el otro resultado de esa protección y bromas que tan bien se complementan y que nutrió nuestra infancia. Siempre fue un ejemplo de constancia, superación y vocación. Me enseñó que no hay nada más lindo que hacer lo que a uno le gusta. Siempre será mi primer opción cuando precise un compañero para molestar a mamá.

El otro miembro de la familia al que quiero agradecer es mi abuela Adela. Nunca dudó en dar una mano con lo que se necesite. Siempre tenía un lugar en su casa para ayudarme con literatura, filosofía o la materia del liceo que fuera. Como si fuera poco, sin entender mucho, se sentaba por horas para ayudarme a repasar teoremas y fórmulas, e incluso la pasaba bien.

Quiero agradecerle también a Tai, mi pareja, amiga, compañera de entrenamientos, y muchas otras cosas. A pesar de la distancia siempre estuvo para mí y me entendía y mimaba en mis momentos de estrés previo a los parciales. El mayor ejemplo de “perseguí tus sueños” y de perseverancia que tuve durante los 5 años de carrera. Siempre estuvo para escuchar mis problemas por más absurdos que fueran y para soportar mi bronca cada vez que perdía Peñarol. Ella me permitió mantenerme medianamente cuerdo y nunca bajar la cabeza, espero que estes en todos los agradecimientos que escriba en mi vida.

Quiero dar las gracias también a mis amigos que fueron parte fundamental de esta carrera. Con algunos compartí cursos e incluso grupo de trabajo. El agradecimiento es fundamentalmente por estar y permitir “desconectarme” aunque sea por la duración de un asado. Mantenemos nuestra amistad desde niños y sé que siempre voy a poder contar con ustedes para debatir, contarles problemas o comer carne.

Agradezco a la gente del IIE con la que trabajé y que todos los días ayudan a mi formación. También a aquellos con los que no trabajé pero siempre escucharon los problemas que surgieron en el proyecto y se preocuparon por su avance como Martín, Camilo, Lucas, Chino, Diego, Juli, los Emis y los que siempre están en la vuelta.

Por último quiero agradecerle a Romina, o Romi como ella prefiere. La otra mitad de este proyecto y me animo a decir que la mitad de mi título también se lo debo a ella. Si no fuera por Romi este y casi todos mis trabajos llevarían el doble de tiempo. Mirando hacia atrás me cuesta creer que antes de empezar la carrera éramos desconocidos. Nuestros métodos de trabajo se complementan tan bien que ya no recuerdo cómo era cursar una materia sin ella. No recuerdo cómo hacía para poder aprender sin alguien que organice “preguntómetros” para los parciales y que me agilice cuando me centro en detalles. Increíblemente es la misma persona que sabe estar en cada mínimo detalle de los trabajos y que tiene la capacidad de mantener a todos los grado 1 y 2 del IIE unidos como una familia. Gracias de nuevo por estos años de carrera. Espero que en el futuro podamos seguir aprendiendo, trabajando y enseñando juntos.

Esta página ha sido intencionalmente dejada en blanco.

A nosotros de hace cinco años.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

El Sistema de Identificación Automática (AIS, por sus siglas en inglés) es una tecnología de comunicaciones marítimas. A través de estaciones AIS se posibilita que embarcaciones se comuniquen entre sí o con tierra firme. Esto permite brindar seguridad a navegantes en alta mar y rápidas respuestas frente a accidentes. Se transmiten mensajes periódicos de tipo *broadcast*, conteniendo información de posición, velocidad y dirección de movimiento de la embarcación. Así, las otras embarcaciones planifican su movimiento en función de los vehículos cercanos.

Existen dos tipos de estaciones móviles AIS: las de clase A y las de clase B. Estas últimas están pensadas para embarcaciones pequeñas, de pesca o recreación, y su uso no es obligatorio en Uruguay. Debido al costo asociado, la mayoría de las embarcaciones no cuentan con una estación. Este riesgo de seguridad es la principal motivación del proyecto.

Las estaciones de clase B se dividen en dos subtipos según el mecanismo de acceso al medio utilizado. El medio de transmisión es el aire. Este es compartido por todas las estaciones AIS en los canales de frecuencia de la tecnología. Para que las embarcaciones puedan intercambiar información sin interferencias, se aplica una técnica de acceso al medio por división temporal (TDMA, por sus siglas en inglés). Se divide el tiempo en slots y se limita a que una sola embarcación transmita en cada uno. Existen dos variantes principales de TDMA utilizadas por las estaciones de clase B: TDMA auto-organizado (SOTDMA, por sus siglas en inglés) y TDMA de sensado de portadora (CSTDMA, por sus siglas en inglés). La diferencia entre ambas reside en cómo se elige qué estación hará uso del canal en cada slot de tiempo.

Por un lado, en SOTDMA se implementa un sistema de reserva de slots. Cada estación sabe quién hará uso del canal y cuándo. Por otro lado, en CSTDMA, se sensará el nivel de ruido del canal durante un tiempo acotado del slot en que se deba transmitir. Si en función de un umbral dinámico se detecta que nadie está transmitiendo, se envía el mensaje correspondiente. Sino, la estación deberá esperar al siguiente slot candidato y reintentar.

En este trabajo se plantea la implementación de una estación AIS de clase B. El mecanismo de acceso al medio utilizado es CSTDMA. Se discute la elección de dicho mecanismo y su implementación. Además, se profundiza en la generación de mensajes AIS interpretables. El desarrollo se realiza como bloques de la herramienta GNU Radio. Se realiza también un prototipo de hardware basado en Radio Definida por Software (SDR, por sus siglas en inglés), con el objetivo de que sea de bajo costo. Esto último busca revertir la situación en Uruguay.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Agradecimientos	III
Resumen	IX
1. Introducción	1
1.1. Motivación	1
1.2. Definición del problema	4
1.2.1. Objetivos generales	5
1.2.2. Objetivos y alcance	5
1.3. Antecedentes	6
1.3.1. Trabajos previos	6
1.3.2. Implementaciones de código libre	8
1.4. Contribuciones	10
1.5. Estructura general del documento	10
2. Sistema de Identificación Automática	11
2.1. Historia	11
2.2. Generalidades	12
2.2.1. Conceptos previos	12
2.2.2. Especificaciones	13
2.2.3. Tipos de estaciones	15
2.3. Funcionamiento a nivel de capa física	16
2.4. Funcionamiento a nivel de capa de enlace	18
2.4.1. Subcapa 1: Control de acceso al medio	19
2.4.2. Subcapa 2: Servicio de enlace de datos	21
2.4.3. Subcapa 3: Entidad manejadora de enlaces	22
2.5. Mensajes existentes	31
3. Mecanismo de acceso al medio	35
3.1. Comparación entre SOTDMA y CSTDMA	35
3.2. Simulador	36
3.2.1. Consideraciones previas	37
3.2.2. Esquema de simulación	38
3.2.3. Análisis de resultados	39

Tabla de contenidos

4. Implementación del sistema transmisor	45
4.1. Aspectos generales	45
4.2. Implementación de la recepción	46
4.2.1. Proyecto <i>gr-ais</i>	46
4.2.2. Cálculos de parámetros de funcionamiento	47
4.2.3. Modificaciones sobre el receptor de <i>gr-ais</i>	48
4.3. Herramientas de hardware utilizadas	52
4.3.1. Calibración del dispositivo SDR	53
4.4. Implementación de la transmisión	56
4.4.1. Proyecto <i>gr-aistx</i>	56
4.4.2. Transmisor	57
5. Generación y codificación de mensajes	65
5.1. Estructura general de las tramas	65
5.2. Mensajes considerados para el proyecto	66
5.2.1. Mensaje 18	66
5.2.2. Mensaje 24	67
5.3. Generación de mensajes	67
5.3.1. Obtención de datos	72
5.3.2. Armado del campo de datos del mensaje	72
5.3.3. Generación de la trama	72
5.3.4. Proceso de bit stuffing	72
5.3.5. Codificación en NRZI	72
5.3.6. Modulación GMSK	73
5.4. Ejemplo de mensaje AIS	73
6. Pruebas de funcionamiento y optimización	75
6.1. Proceso de optimización del sistema	75
6.1.1. Mejoras a nivel de bloques	76
6.1.2. Mejoras a nivel del sistema completo	80
6.2. Preparación del sistema para las pruebas	82
6.3. Pruebas de funcionamiento y análisis de resultados	83
6.3.1. Prueba 1: Potencia de transmisión	84
6.3.2. Prueba 2: Tiempo de cambio de canal	85
6.3.3. Prueba 3: Tiempo de actualización del umbral	86
6.3.4. Prueba 4: Intervalos de reporte	86
6.3.5. Prueba 5: Transmisión de mensajes interpretables	88
6.3.6. Prueba 6: Contenido de los mensajes	90
6.3.7. Prueba 7: Acceso al medio compartido	90
6.3.8. Prueba 8: Primer acercamiento al retardo de transmisión	91
6.3.9. Prueba 9: Retardo de transmisión	92
7. Conclusiones	97
7.1. Reflexiones	97
7.2. Trabajo a futuro	99
7.3. Contribuciones	102

A. Estudio del mercado	105
B. Radio Definida por Software	109
C. Configuración de un servidor NTP	111
C.1. Requerimientos previos	111
C.2. Configuración del servidor	112
C.2.1. Habilitación de pines de entrada/salida	112
C.2.2. Descarga de <i>gpsd</i>	112
C.2.3. Configuración de PPS	113
C.2.4. Configurando chrony como servidor de tiempo	113
D. Calibración con el analizador de espectro	115
Referencias	119
Índice de tablas	124
Índice de figuras	127

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 1

Introducción

En este capítulo se presenta el problema de estudio y su relevancia en Uruguay. Se realiza una breve introducción a conceptos importantes para el desarrollo del trabajo, que se profundizarán en los siguientes capítulos. Se mencionan los antecedentes más relevantes. En particular, se destacan algunos que funcionan como base para la implementación realizada. Se describen los objetivos considerados y se analiza su cumplimiento indicando las contribuciones realizadas.

1.1. Motivación

La comunicación ha sido siempre un factor de importancia en la vida humana. La necesidad de los seres humanos de vivir en sociedad implicó el desarrollo de diversos mecanismos para permitir el intercambio de ideas, desde pinturas en cavernas hasta el correo electrónico [1].

Las telecomunicaciones se basan en la tecnología para lograr el intercambio de información. Hacia finales del siglo XIX, las ecuaciones de Maxwell y la verificación de la existencia de radiaciones a cualquier frecuencia sirvieron como bases para la revolución digital [2]. Sus primeras etapas estuvieron fuertemente marcadas por el desarrollo de las comunicaciones inalámbricas, buscando realizar transmisiones entre lugares distantes. Una de las primeras aplicaciones de comunicaciones a larga distancia es el intercambio de información entre la costa y embarcaciones en alta mar.

Las comunicaciones marítimas han sido un tema de suma relevancia durante los últimos cien años. Uno de los sucesos más famosos del siglo pasado es el hundimiento del Titanic. La enorme embarcación contaba con un telégrafo a través del cual se le avisó de la presencia de hielo en las cercanías, pero el operador estaba ocupado enviando otros telegramas y no prestó atención al aviso [3]. La tragedia resultó en la primera convención sobre Seguridad de la Vida en el Mar (SOLAS, por sus siglas en inglés). En ella se definieron medidas para mayor protección en el transporte marítimo [4]. Por otro lado, los eventos de la Segunda Guerra Mundial resultaron en grandes avances en las comunicaciones marítimas. Las distintas flotas de los ejércitos buscaron intercambiar información confidencial, desarrollando

Capítulo 1. Introducción

técnicas de encriptación y posteriormente avanzando en mecanismos de detección y corrección de errores [5].

En cierta forma, la seguridad marítima siempre ha dependido de conexiones con otras personas, en tierra o en alta mar. Una de las primeras tecnologías definidas con el principal objetivo de asegurar el desarrollo de travesías marítimas es el radar [6]. Su principio de funcionamiento consiste en enviar señales de radiofrecuencia (RF) en todas las direcciones y detectar los rebotes de esas señales. En función de desde qué direcciones se obtiene una señal de rebote y su intensidad, es posible generar un mapeo de los alrededores, determinar si existen objetos cercanos y a qué distancia aproximada se encuentran. Estos sistemas permiten conocer la presencia de embarcaciones dentro de cierto rango, algunas veces previo a poder visualizar dichas naves en el horizonte. Sin embargo, el radar no indica qué objetos se identificaron y tampoco funciona de forma correcta si no se tiene línea de vista con el objeto. La búsqueda por desarrollar tecnologías que mejoraran la seguridad en el mar tuvo un hito significativo en el comienzo del siglo XXI, cuando se introdujo la tecnología del Sistema de Identificación Automático (AIS, por sus siglas en inglés).

La tecnología AIS permite a las embarcaciones comunicarse entre sí o con estaciones en tierra firme. Se trata de transmisiones periódicas de tipo *broadcast* en la banda de muy altas frecuencias (VHF, por sus siglas en inglés) [7]. El sistema transmite constantemente información de posición, velocidad y dirección de navegación. A partir de estos mensajes, todas las embarcaciones tienen acceso a un mapeo de las estaciones cercanas, su tipo e identificación y sus características de movimiento. El intercambio de información promueve el agregado de seguridad en la navegación, tal y como fue definido en el estándar de desempeño de la Organización Marítima Internacional (IMO, por sus siglas en inglés) [8].

Existen varias ventajas de la tecnología AIS por sobre el radar. La utilización de canales de radiofrecuencia en la banda VHF hace que AIS supere de forma significativa el rendimiento de los radares, ya que no es necesario tener línea de vista con las embarcaciones para tener conocimiento su presencia. Por otro lado, AIS permite recolectar una gran cantidad de información. Se logra tener una representación actualizada del estado del tráfico marítimo cercano [9], evitando colisiones y permitiendo cierto seguimiento de las embarcaciones. AIS es también más resistente a las inclemencias del tiempo [10] que los radares. Esto se debe a que trabaja en frecuencias que rondan los 162 MHz, mientras que los radares marítimos deben trabajar en las bandas de 3, 5 o 9 GHz [11].

Se tienen también algunas limitaciones al utilizar AIS. La interpretación de los datos recibidos como mensajes AIS presenta mayores dificultades que la interpretación de los mapas generados por los radares. Además, la información recolectada puede ser poco precisa si la estación no fue configurada de forma correcta [10], o si los sensores no han sido calibrados. Estos errores podrían corregirse manualmente. Sin embargo, esto indica otra desventaja de la tecnología: es sencillo manipularla. Basta con apagar la estación para dejar de transmitir información, haciendo que las embarcaciones pasen a tener cierto anonimato. Un mecanismo de manipulación sofisticado es el *spoofing*. Esta técnica consiste en modificar la información de

posición utilizada, con el fin de ocultar los movimientos reales de las embarcaciones [10].

Existen distintos tipos de sistemas AIS. Entre ellos se encuentran las estaciones de navegación marítima, es decir, aquellas que se utilizan para comunicaciones desde y hacia alta mar. A su vez, dentro de esta clasificación se tienen las estaciones móviles, utilizadas para embarcaciones, y las estaciones base, fijas en tierra firme. Las estaciones móviles se dividen entre aquellas de clase A y las de clase B. Una embarcación estará equipada con una estación AIS de clase A o clase B dependiendo de su tamaño o su función (comercial, recreacional). Las embarcaciones de gran porte suelen contener un sistema AIS de clase A, mientras que las embarcaciones pequeñas (usualmente de recreación) suelen tener un sistema de clase B. Estos sistemas se diferencian en la cantidad de especificaciones que deben cumplir, así como en parámetros de funcionamiento (potencia de transmisión, intervalos de reporte, mecanismos de acceso al medio, entre otros).

Desde hace poco más de veinte años, todas las embarcaciones de gran porte, por ejemplo barcos de carga o barcos de pasajeros, deben contar con una estación AIS. Esta obligatoriedad inició en 2002, como resultado de una convención SOLAS de la IMO [6]. En un principio, la regla se aplicó a embarcaciones con volúmenes mayores a 300 toneladas brutas, definiéndose primero el estándar para las estaciones AIS de clase A. En 2006 se estandarizó la utilización de estaciones de clase B [12]. Existió un período de adaptación/introducción a la tecnología AIS entre 2002 y 2008, durante el cual las penalizaciones por no contar con estaciones estaban mitigadas. Desde 2008 se encuentra en rigor la normativa internacional de contar con un sistema AIS, pero únicamente sobre las embarcaciones de gran porte.

La situación de Uruguay es la misma que se plantea a nivel internacional. La disposición marítima N^o 94/04, redactada en 2004, indica la obligatoriedad del uso de estaciones AIS de clase A para embarcaciones de pasajeros y buques tanque de cualquier tamaño, y para embarcaciones con volúmenes mayores a 300 toneladas brutas [13]. En lo que respecta a las embarcaciones pequeñas o que no sean de pasajeros, no es obligatorio en el territorio nacional el uso de un sistema AIS de clase B. La última noticia encontrada sobre esta reglamentación es del año 2014, y no se encontraron modificaciones al respecto [14].

Como se desarrolla en el apéndice A, el costo de los equipos disponibles en plaza resulta elevado para embarcaciones pequeñas. El costo mínimo de adquisición de una estación es aproximadamente 800 dólares americanos. A este precio se le debe sumar el costo del curso obligatorio para hacer uso del dispositivo, de 930 U.I.¹ [15]. El monto total necesario y la no obligatoriedad de tener sistemas AIS llevan a que pocas embarcaciones hagan uso de la tecnología. Esta situación ocurre a pesar de las grandes ventajas desde el punto de vista de la seguridad y de los esfuerzos del estado uruguayo para motivar el uso de estos sistemas [14].

¹Al día 27 de marzo del 2024, una unidad indexada tiene un valor de 5,99 pesos uruguayos.

1.2. Definición del problema

Se plantea un proyecto para implementar un sistema transceptor AIS de bajo costo basado en Radio Definida por Software (SDR, por sus siglas en inglés). En particular, teniendo en cuenta que la mayoría de las embarcaciones afectadas son pequeñas, es relevante centrarse en estaciones de clase B para el desarrollo del proyecto. Es decir, como usualmente las estaciones de clase A corresponden a barcos de gran porte o de carga, la inversión en el equipamiento AIS no suele ser significativa para las grandes compañías, además de ser obligatoria. Dada la necesidad presente en Uruguay, el foco principal del proyecto estará en la implementación de un transceptor AIS de clase B. Si bien los objetivos son planteados para el desarrollo de un transmisor, se hará uso de receptores de código libre. El sistema transmisor implementado será de uso y acceso libre.

El proyecto plantea la utilización de dispositivos de Radio Definida por Software (SDR). En el apéndice B se puede encontrar una breve introducción a la tecnología. La utilización de SDR implica costos considerables en el diseño. Sin embargo, también se tienen grandes ventajas al usar esta solución. La principal ventaja de utilizar SDR es que se evita el diseño de hardware específico, que es un proceso difícil y que implicaría varios meses de trabajo. Se utilizan dispositivos definidos por software que permiten hacer prototipos de forma rápida y aportan flexibilidad en el caso de que los estándares se actualicen o cambien. Con SDR es posible trabajar con las muestras de las señales de radiofrecuencia, aspecto que será de suma importancia para la implementación del transmisor. El proyecto requiere obtener valores del canal a través de algún módulo receptor para hacer cálculos de potencia de la señal recibida. Si se utilizara un receptor AIS adquirido en plaza, es poco probable tener acceso a las muestras, lo que implicaría un diseño de hardware específico para obtener muestras del canal. Existe en Internet una gran comunidad de profesionales y aficionados utilizando tecnología SDR. Esto posibilita el acceso a foros para realizar consultas o aportes, y a diversas herramientas de software de uso libre. Existen dispositivos de diferente gama y costo. Se trata de una tecnología moderna que ha demostrado buenos resultados de funcionamiento.

Como parte del trabajo se diseñará un prototipo de hardware. Esto implicará la elección de módulos GPS o placas de procesamiento a utilizar. Se tienen algunas restricciones con respecto a este diseño. El principal objetivo es un prototipo de bajo costo. Se buscará trabajar con elementos accesibles para el grupo de trabajo, minimizando la compra de nuevos dispositivos. En particular, está preestablecida la utilización de un SDR ADALM-PLUTO como principal transmisor. Esto se debe a que se tiene disponibilidad del equipo.

El desarrollo del proyecto tiene como principal referencia a la norma ITU-R M.1371-5 [16], de la Unión Internacional de las Telecomunicaciones (ITU, por sus siglas en inglés), la cual especifica el funcionamiento de las estaciones AIS. Para lograr el objetivo planteado, se estudiaron principalmente aspectos relacionados con funcionalidades a nivel de capa física y de capa de enlace.

A continuación se mencionarán los objetivos del proyecto, así como también el alcance del mismo.

1.2. Definición del problema

1.2.1. Objetivos generales

Se listan a continuación los objetivos generales del proyecto.

- Implementar un sistema transmisor AIS de clase B.
- Implementar uno de los mecanismos de acceso al medio descritos en la norma ITU-R M.1371-5.
- Generar y codificar mensajes AIS.
- Diseñar un prototipo de hardware de bajo costo y basado en SDR para el sistema transceptor.
- Probar el prototipo desarrollado en un escenario controlado y en un escenario real.

1.2.2. Objetivos y alcance

Se listan a continuación los objetivos del proyecto, detallando en los casos en los que corresponde con cuanta profundidad se planea trabajar.

- Comprender los puntos fundamentales de la norma ITU-R M.1371-5. El foco estará particularmente en las embarcaciones pequeñas, el acceso al medio y mensajes que estas utilizan. Se definirá de qué tipo de clase B será la estación a implementar. Se determinará cuáles de las funcionalidades opcionales van a ser implementadas. Se dejará constancia de los mecanismos o procedimientos que serán implementados. Se parte de la base de que no se implementará nada que no esté descrito en la norma, a no ser que se aclare específicamente lo contrario en este documento.
- Implementar un mecanismo de acceso al medio para el transmisor. Para esto se desarrollarán programas que permitan cumplir con las tareas. Se trabajará con especificaciones a nivel de capa de enlace en el modelo OSI². Se cuenta con dos posibles mecanismos a utilizar, ambos basadas en una estrategia de acceso múltiple por división de tiempo (TDMA, por sus siglas en inglés). El primero de ellos se basa en el sensado del canal (*carrier sense TDMA*, o CSTDMA, por sus siglas en inglés). En el segundo las embarcaciones se organizan entre sí, realizando reservas de intervalos de tiempo para las transmisiones (*self organized TDMA*, o SOTDMA, por sus siglas en inglés). Estos dos mecanismos serán abreviados como CS o SO, respectivamente, en lo que resta del documento.
- Lograr la capacidad de acceder al medio cuando este sea compartido. Asegurar que la estación diseñada detecte la presencia de otras estaciones (simuladas o reales) y que acceda o no al medio compartido, en función de lo

²Siglas en inglés para *open system interconnection*. Se trata de un modelo de capas para la estructura de redes de telecomunicaciones.

Capítulo 1. Introducción

que corresponda. La estación implementada deberá acceder al medio para transmitir en los momentos que sean adecuados: ya sea porque mediante el mecanismo SOTDMA se determinó que es el momento de transmitir, o porque se sensó el canal (mecanismo CSTDMA) y fue categorizado como libre. El acceso al medio incluirá entonces la correcta transmisión y la correcta recepción e interpretación de los mensajes externos.

- Implementar los mensajes a transmitir, con información de velocidad, dirección y ubicación. Escribir programas para la generación y la codificación de mensajes. Se trabajará con especificaciones a nivel de capa física en el modelo OSI. En particular, se aplican una codificación de no retorno a cero invertido (NRZI, por sus siglas en inglés) y un esquema de modulación digital por desplazamiento mínimo Gaussiano (GMSK, por sus siglas en inglés) para generar señales de voltaje a partir de bits. Se debe lograr que los mensajes puedan ser correctamente interpretados por otros dispositivos que utilicen esta tecnología.
- Enviar los mensajes a distintos intervalos de tiempo. Estos intervalos serán los establecidos por la norma ITU-R M.1371-5. La estación tendrá que comunicarse con los correspondientes sensores o equipos que funcionen como entradas. Esta información se utilizará para el armado de los mensajes y para la definición del intervalo de reporte.
- Armar un sistema que incorpore los dos módulos anteriores y permita que interactúen entre sí. Esto comprenderá todos los programas que permitan el funcionamiento de la estación a implementar.

1.3. Antecedentes

Como se mencionó en las secciones previas, el principal objetivo del proyecto es la implementación de un sistema transmisor AIS de clase B. Previo al comienzo del desarrollo, se realizó un estudio del estado del arte en el tema, repasando artículos, tesis de grado o de posgrado y repositorios de GitHub. Se destacan a continuación los trabajos previos analizados y considerados punto de inicio para el proyecto. Particularmente, en la sección 1.3.1 se analizan los trabajos previos cuya implementación no es de código libre. En la sección 1.3.2 se repasan los trabajos cuya implementación sí es *Open Source*.

1.3.1. Trabajos previos

En el artículo [17] se busca implementar las funcionalidades de una estación móvil AIS. Se consideran transmisores de clase A y de clase B, implementando los mecanismos de acceso al medio SOTDMA y CSTDMA. La implementación se realiza sobre un microprocesador específico para el procesamiento de señales digitales (conocido como DSP, por sus siglas en inglés). Se presentan diagramas de flujo para explicar el funcionamiento de sus estaciones, haciendo énfasis en su

1.3. Antecedentes

programación con hilos. Se tienen varios hilos que publican información a una base de datos y/o interactúan con otros hilos a través de semáforos. El código del proyecto no está disponible en la web. En el artículo únicamente se muestran esquemas del manejo de hilos, y se destaca cómo se relacionan entre sí y cómo fueron probados de forma independiente.

El objetivo del artículo [18] es la implementación de una estación AIS de bajo costo. Se propone la utilización de un circuito de radiofrecuencia y detectores de frecuencia para realizar la conversión analógica a digital de la señal. La digitalización permite procesar las muestras en una computadora y analizarlas con herramientas de software como AISMon [19]. Las únicas pruebas realizadas son simulaciones en Matlab y el código utilizado no está disponible en la web. El autor menciona que el funcionamiento es el deseado, y que los materiales utilizados son de bajo costo (destacando que los circuitos RF tienen costos menores a 100 dólares). Si bien los requisitos de bajo costo son similares a los del presente proyecto, el artículo no se consideró tan relevante para esta investigación debido a que su implementación no está basada en SDR y a que pasaron casi 20 años de su fecha de publicación.

En [20] se describe el diseño de un receptor AIS a ser agregado a un satélite sudafricano. Se propone realizar un seguimiento de embarcaciones en el mar usando el satélite, guardando la información recibida y reenviándola a una estación terrestre. Se utiliza el lenguaje C++ para la escritura del código de decodificación de mensajes. La solución se basa en tecnología SDR y fue simulada en un entorno de Matlab. Se destacan la descripción del esquema de modulación digital GMSK y los resultados de las simulaciones de un modulador GMSK. Ambos permiten un mejor entendimiento del tema. Por otro lado, no se describe de forma exhaustiva cómo fueron utilizados los dispositivos SDR para la obtención de muestras de las señales.

El trabajo [21] detalla el desarrollo de un prototipo de hardware para un transmisor y un receptor AIS. Se presenta el diseño y la implementación física de un prototipo para la realización de pruebas de funcionamiento del estándar AIS. Se trabaja con hardware específico en lugar de tecnología SDR. Además, la herramienta base de trabajo es un microcontrolador Arduino. Estas son diferencias importantes con el plan de trabajo para el proyecto del equipo.

En el artículo [22] se presenta la implementación de un transmisor de mensajes AIS. Se utiliza tecnología SDR y la herramienta de software GNU Radio. No se desarrolla un sistema transmisor propiamente dicho, ya que no se tiene manejo del acceso al medio compartido, ni se mantienen temporizadores. La implementación del artículo se limita a codificar y transmitir mensajes AIS cuando se le entrega información de GPS. A pesar de que el código del trabajo no está disponible, sí se referencia el proyecto tomado como base. Se trata de un módulo *out-of-tree* de GNU Radio llamado *gr-aistx*, disponible en GitHub, que será introducido en la siguiente sección.

1.3.2. Implementaciones de código libre

En esta sección se describen algunas implementaciones de código libre (también denominadas *open source*) de sistemas AIS para transmisión o recepción. Se introducen también módulos de GNU Radio. Se recuerda que se busca implementar un sistema transmisor AIS. Esto implica tener procesos de transmisión y de recepción funcionando de forma simultánea. Los trabajos analizados se encuentran en dos principales categorías: transmisores o receptores. Los transmisores servirán como referencia para el desarrollo. Los receptores podrían ser incorporados al sistema transmisor a implementar.

El primer proyecto revisado es el proyecto de MAIANA [23], comenzado en 2015. Su desarrollador, Peter Antypas, no estaba satisfecho con los transceptores AIS de clase B disponibles en el mercado. Los consideraba costosos, con poca potencia y de gran tamaño para embarcaciones pequeñas. Por estos motivos decide desarrollar un transmisor de clase B AIS para uso personal. Eventualmente decide comenzar a comercializar su diseño y para julio de 2023 ya se habían fabricado más de 300 unidades. El dispositivo de MAIANA es autocontenido, esto es, todas las antenas, la fuente de información del sistema global de navegación por satélite (GNSS, por sus siglas en inglés) y el circuito se alojan en la misma unidad. Se trata de un transmisor completamente funcional, que además es compatible con herramientas de software que permiten el manejo a través de una interfaz gráfica. Utiliza el mecanismo de acceso al medio SOTDMA. La principal ventaja de este proyecto es la disponibilidad de su código en Internet, desarrollado en el lenguaje C++. Se tiene acceso a una documentación completa y exhaustiva del proceso de diseño de los circuitos, así como los programas de código instalados en el microcontrolador del transmisor. Sin embargo, esta solución no utiliza tecnología SDR. Esto dificulta su adaptabilidad al proyecto, ya que el código está muy condicionado al hardware del dispositivo.

Se analizó un módulo *out-of-tree* de GNU Radio conocido como *gr-aistx* [24]. Se trata de una implementación de código libre de un transmisor de mensajes AIS a demanda. Esto es, se tienen los bloques para transmitir un cierto mensaje AIS, pero no se implementa la inteligencia asociada a cuándo se debe transmitir, o si el medio compartido está disponible para la transmisión. El repositorio está desarrollado con el objetivo de brindar herramientas para investigadores que deseen realizar simulaciones con un transmisor AIS y fue utilizado para algunos artículos académicos. El código fue escrito para la versión 3.7 de GNU Radio y utilizando los lenguajes C y C++ para los bloques. El repositorio cuenta con un script en Python para codificar mensajes hasta tener su representación en un string de bits. Además, es posible representarlos con caracteres ASCII en formato NMEA. El string de bits se utiliza como entrada para un esquema de GNU Radio definido como AiS.TX, con un archivo *.grc* y un archivo *.py*. El esquema cuenta con bloques que calculan el código de redundancia cíclica del mensaje, le agregan las banderas de inicio y fin, y luego lo codifican y modulan con NRZI y GMSK respectivamente. Los scripts y bloques implementados en el proyecto permiten comprender el proceso de generación y codificación de los mensajes AIS. Existen pedidos de *commits* en el repositorio de GitHub con versiones del código adaptadas para GNU Radio 3.8.

1.3. Antecedentes

Por otro lado, se estudió otro módulo *out-of-tree* de GNU Radio conocido como *gr-ais* [25]. Su implementación está disponible en un repositorio de GitHub. El módulo *gr-ais* define diagramas de flujo sin interfaz gráfica para la recepción de mensajes AIS en terminal. El código está desarrollado en el lenguaje Python, utilizando un entorno de GNU Radio 3.8. Se tiene el archivo *radio.py* como la definición de los bloques a aplicar a las muestras recibidas. Este proceso de recepción involucra una inversión de la modulación y codificación realizada en transmisión (demodulación GMSK/FM, decodificación NRZI). De esta forma, se tiene una referencia para implementar un transmisor en GNU Radio, considerando bloques análogos e inversos a los utilizados en recepción. Los bloques de *gr-ais* están diseñados para utilizar dispositivos receptores con tecnología SDR. Esto permite trabajar con las señales de radiofrecuencia muestra a muestra en bandabase. Para la implementación de un transmisor, es de suma importancia contar con procesos de recepción que aporten información sobre el estado del canal. El módulo *gr-ais* posibilita el análisis de las muestras a través de la utilización de otros bloques, desviando el flujo de datos para hacer cálculos de potencia recibida.

Otro receptor considerado fue dAISy [26]. Se trata de una herramienta de software libre que recibe y decodifica mensajes AIS. El código está implementado en el lenguaje C, y se puede encontrar en el repositorio de GitHub del proyecto. dAISy está diseñado para utilizarse con un dispositivo del mismo nombre desarrollado por el mismo grupo de trabajo. Este dispositivo no está basado en tecnología SDR, e implica un diseño de hardware complejo y específico. El proyecto está documentado en detalle, con descripciones de los materiales de hardware y la motivación para su uso. Sin embargo, la utilización de hardware específico dificulta la adaptación al proyecto del equipo.

El receptor AIS-catcher está implementado en el lenguaje C++ y también está disponible en GitHub [27]. Se trata de un receptor pensado para utilizar con dispositivos de tipo dongles RTL-SDR, que decodifica los mensajes y funciona como interfaz hacia otras aplicaciones. El proyecto cuenta con instrucciones claras de instalación y varios ejemplos de uso. Sin embargo, el código es poco claro y no cuenta casi con comentarios. Esto dificulta su utilización, y la posibilidad de desviar muestras del flujo recibido para determinar el estado del canal.

Se consideraron otros receptores, de los cuales no se existe demasiada documentación. AIS.Net.Receiver [28] está desarrollado en el lenguaje C# para ser utilizado con placas Raspberry Pi. Su documentación en el repositorio de GitHub asociado no tiene suficiente claridad, ni detalle. Por otro lado, se consideró Ships [29], que permite recibir información AIS utilizando dongles RTL-SDR y representarla en mapas. Se trata de una aplicación Android desarrollada en Java, lenguaje con el que el equipo no tiene experiencia.

Para el trabajo se tomó como base a los módulos *out-of-tree*³ de GNU Radio conocidos como *gr-aistx* y *gr-ais*.

³Módulos de GNU Radio que no vienen con la herramienta al descargarla. Suelen estar desarrollados por aficionados o profesionales a la tecnología SDR y son publicados en repositorios en línea.

1.4. Contribuciones

Se logró la implementación de un sistema transmisor para la tecnología AIS. Los aportes resultantes del proyecto se categorizan en tres áreas: aportes a la comunidad en línea, aportes a la comunidad científica y aportes a la comunidad local.

En primer lugar, se destaca que este proyecto resultó en la creación de un módulo *out-of-tree* de GNU Radio llamado *gr-itais*⁴. Este contiene bloques de proyectos previos, como *gr-ais* o *gr-aistx*, además de bloques desarrollados en el marco de este proyecto. A partir de este módulo, es sencillo clonar el código desarrollado y utilizarlo desde computadoras personales. La comunidad en línea de SDR tendrá acceso libre y completo al proyecto. Se podrán generar contribuciones y mejoras al trabajo.

En segundo lugar, se escribió y presentó un breve artículo en la XXX Jornada de Jóvenes Investigadores en Asunción, Paraguay. Se realizó una presentación oral y una presentación de póster, mencionando resultados iniciales del proyecto. En este evento se obtuvo una distinción por mejor presentación de póster. Además, como otro aporte a la comunidad científica, se escribió un artículo enviado a consideración para la XIII LANC (Latin American Networking Conference). En este último se aprecian resultados del proyecto completo.

Finalmente, los conocimientos adquiridos en este trabajo fueron utilizados para un proyecto del curso Taller de Introducción a la Ingeniería Eléctrica - Comunicaciones Inalámbricas. Así, se realizó un aporte a la comunidad local, aportando material de aprendizaje para estudiantes de primer año de la Facultad de Ingeniería de la Universidad de la República.

1.5. Estructura general del documento

Se presenta en este documento el proceso seguido para la implementación de un transmisor de clase B de la tecnología AIS, basado en SDR. El trabajo se estructura como se indica a continuación.

En el capítulo 2 se introduce el marco teórico de la tecnología AIS. En el capítulo 3 se mencionan las diferencias entre los distintos tipos de acceso al medio existentes. Se crea un simulador para justificar la elección de qué mecanismo utilizar. Más adelante, en el capítulo 4 se ve en detalle la implementación del sistema encargado de la transmisión así como también qué aspectos de otros módulos fueron utilizados. En el capítulo 5 se explica la generación y la codificación de los mensajes. Luego, en el capítulo 6 se ven algunas optimizaciones realizadas al sistema así como también pruebas de funcionamiento del mismo. Por último, en el capítulo 7 se concluye el trabajo, reflexionando sobre el cumplimiento de los objetivos y mencionando en qué se debe trabajar en el futuro.

⁴La versión final del repositorio está disponible en <https://gitlab.fing.edu.uy/rominag/itais>.

Capítulo 2

Sistema de Identificación Automática

En este capítulo se introduce el marco teórico en el cual se basa el proyecto. En particular, se describe el funcionamiento de la tecnología del Sistema de Identificación Automática (AIS, por sus siglas en inglés) siguiendo la norma ITU-R M.1371-5. Se realiza un análisis a nivel de capa física y de capa de enlace, siguiendo el modelo de capas OSI. El enfoque estará en las estaciones AIS de clase B, ya que son las estaciones consideradas para el objetivo del proyecto.

La tecnología AIS tiene como objetivo el intercambio de información entre estaciones marítimas de tipo estáticas o móviles. Se busca que cada embarcación pueda dar aviso a las embarcaciones cercanas sobre su ubicación y la dirección en la cual se estará moviendo en los instantes temporales siguientes. Además, se intercambia información de identidad y tipo de la nave [30]. La Asociación Internacional de Ayudas Marítimas (IALA, por sus siglas en inglés), junto con la IMO, estandarizaron el uso y funcionamiento de AIS. En particular, se fijó la mejora de la seguridad en la navegación como el objetivo principal de la tecnología [7].

2.1. Historia

Benny Pettersson es un capitán proveniente de Suecia y es también el principal impulsor y creador de AIS. En el año 1965, Pettersson se encontraba trabajando en Japón. Ante la llegada del tifón Jane, con vientos que alcanzaron los 185 km/h, su nave fue obligada a abandonar el puerto y anclar en las afueras [9]. En una entrevista, el capitán cuenta que el barco debió moverse a baja o media velocidad, y que el ancla debía ir arrastrándose por el suelo del mar, con el fin de evitar que la embarcación se moviera demasiado rápido. Pettersson contó que la visibilidad era muy reducida y que el radar no parecía ser de ayuda. La situación de desesperación a la que se sometió en ese momento lo llevó a desarrollar un sistema que pudiera ser utilizado incluso en condiciones climáticas extremas.

El proceso de pruebas y difusión de sus ideas le llevó a Pettersson varios años. En un inicio, únicamente propietarios de yates lujosos tenían disposición de utilizar su tecnología. Eventualmente, motivados por los ataques terroristas del 11

Capítulo 2. Sistema de Identificación Automática

de septiembre de 2001, organizaciones internacionales empezaron a prestar más atención a la tecnología AIS [9].

Durante los últimos veinte años ha sido obligatorio para las embarcaciones de gran porte contar con un sistema AIS. La obligatoriedad surge como resultado de la convención SOLAS de la IMO en 2002 [6]. La prioridad, en aquel entonces, era aplicar estos sistemas en embarcaciones de grandes volúmenes. Con este fin se definió una clase A de sistemas AIS para estaciones móviles, que involucran una gran variedad de mensajes a intercambiar, que deben cumplir parámetros de funcionamiento estrictos. En el año 2006 surgió la estandarización para estaciones AIS de clase B [12]. Con el objetivo de que fueran utilizadas por embarcaciones más pequeñas o de recreación, se trató de una regulación más flexible y con mecanismos de acceso al medio compartido más sencillos. Una vez las recomendaciones para la implementación de estaciones AIS de clase A y B fueron publicadas por la ITU, existió un período de adaptación o introducción a la tecnología AIS entre 2002 y 2008. Durante dicho período, las penalizaciones por no contar con estaciones estaban mitigadas. Desde 2008 se encuentra en rigor la obligatoriedad internacional de contar con un sistema AIS sobre las embarcaciones de gran porte, siendo el uso opcional para embarcaciones más pequeñas.

La tecnología AIS, desde su definición, busca asegurar la navegación marítima. Esto implica ciertos requerimientos funcionales básicos sobre las estaciones, con el fin principal de evitar colisiones entre embarcaciones [7]. Otro beneficio que se obtiene al usar AIS es la posibilidad de que autoridades costeras sepan qué embarcaciones se encuentran dentro de su territorio marítimo. Esto previene el uso de embarcaciones para propósitos ilegales [31]. Además, contar con información de la posición de embarcaciones permite tener respuestas rápidas en el caso de contaminación (pérdidas de petróleo, por ejemplo, pudiendo determinar con exactitud la zona afectada).

2.2. Generalidades

Se introducirán a continuación algunas generalidades sobre el funcionamiento y características de los transmisores AIS.

2.2.1. Conceptos previos

Tal como se mencionó en la introducción del trabajo, existen dos clases de equipos utilizados para transmitir, los de clase A y los de clase B. Los equipos de clase A suelen ser utilizados por embarcaciones de gran tamaño y tienen un alto consumo energético. Por otro lado, los equipos de clase B suelen utilizarse en embarcaciones pequeñas y son equipos más sencillos. Los equipos de clase B se clasifican en dos subtipos según el mecanismo de acceso al medio utilizado.

El medio por el cual se transmiten los mensajes es el aire. Este medio es compartido por todos los equipos. Para evitar colisiones entre los mensajes y permitir su correcta recepción, el medio debe utilizarse de forma organizada. Una de las formas en las que se puede organizar el acceso al medio es mediante una división

temporal, es decir, los equipos van tomando turnos para enviar mensajes. El acceso múltiple por división temporal (TDMA, por sus siglas en inglés) es la base de los mecanismos de acceso al medio utilizados por AIS. Todos ellos dividen el tiempo en frames y slots. En particular, los frames tienen una duración de un minuto y cada frame se divide en 2.250 slots. En cada slot de tiempo podrá transmitir un único equipo, evitando que se superpongan mensajes de estaciones distintas. Existen diversas formas de determinar a qué equipo le corresponde utilizar cada uno de los slots.

Las estaciones AIS de clase B pueden implementarse con dos tipos de mecanismos de acceso al medio: TDMA de sensado de portadora (CSTDMA, por sus siglas en inglés, que será también abreviado como CS) y TDMA auto-organizado (SOTDMA, por sus siglas en inglés, que será también abreviado como SO). Las estaciones de clase A están obligadas a usar el segundo mecanismo.

SOTDMA consiste en que cada equipo organiza sus propias transmisiones haciendo reservas de slots de tiempo. La estación escuchará las reservas del resto de los equipos y reservará los slots que requiera y no hayan sido reservados previamente, para luego transmitir en ellos. Por otro lado, en CSTDMA, cuando un equipo desea transmitir, este debe sensar el canal. Si el nivel de ruido es menor a cierto umbral se interpretará como que el canal está libre. En ese caso el equipo transmite y en caso contrario reintenta más adelante. El mecanismo CSTDMA es similar al mecanismo utilizado por el estándar 802.11 para redes WiFi. En el caso de AIS no se agregan mensajes específicos para solicitar el canal, a pesar de presentarse los mismos problemas de estación oculta. Cada método de acceso al medio tiene sus ventajas y desventajas. Tanto CSTDMA como SOTDMA serán descritos en detalle en las siguientes secciones.

2.2.2. Especificaciones

Independientemente de la clase de un equipo, es necesario poder diferenciarlo de otras embarcaciones. Para poder identificar a cada embarcación, los transmisores deben contar con un identificador móvil marítimo (MMSI, por sus siglas en inglés). Se trata de un número de 10 dígitos en total, de los cuales solo se consideran los primeros 9 dígitos (esto es, los más significativos). El MMSI viaja en los mensajes AIS en un campo de 30 bits. Para recuperar el número en notación decimal, se dividen los 30 bits en porciones de a 6 bits, y se decodifican utilizando el código ASCII para 6 bits. Se obtendrán cinco números decimales de dos cifras, que conforman los diez dígitos del MMSI. De acuerdo a la norma ITU-R M.1371-5 y a la reglamentación internacional, ninguna embarcación puede transmitir si no cuenta con un MMSI único programado. De los nueve dígitos relevantes, los tres más significativos informan sobre el país en el cual fue emitido el identificador [32]. Estos identificadores suelen ser asignados por la prefectura de cada país. Es posible consultar más información sobre el proceso de obtención del MMSI en Uruguay en [33].

Se tienen dos canales principales de funcionamiento: canal A (o AIS 1), centrado en 161,975 MHz, y canal B (o AIS 2), centrado en 162,025 MHz. Cada

Capítulo 2. Sistema de Identificación Automática

transmisor enviará mensajes de forma alternada entre los dos canales. Las transmisiones se harán en otras frecuencias solo en caso de que la embarcación reciba un pedido específico para ello (o casos extremos de congestión en donde el canal no esté disponible). Los otros dos canales utilizados son los llamados canales 75 y 76 (también en la banda VHF), centrados en 156,775 MHz y 156,825 MHz. Los mismos fueron reservados internacionalmente para el uso por parte de estaciones AIS. Toda transmisión se hará con un ancho de banda de 25 kHz. Los mensajes se codificarán con NRZI, para posteriormente modularse utilizando GMSK. Cada sistema AIS es capaz de transmitir en un canal a la vez, y de recibir en paralelo por dos canales (usando dos procesos receptores independientes).

La potencia de transmisión de los sistemas AIS debe ser de 1 W en su configuración para baja potencia. En su configuración de alta potencia, las transmisiones se harán con una potencia de 12,5 W para equipos de clase A, de 5 W para equipos de clase B SO o de 2 W para equipos de clase B CS. Los equipos que usualmente se utilizan en embarcaciones de gran porte cuentan con una mayor potencia de transmisión que implica un mayor alcance en las comunicaciones. Por defecto, las estaciones deben funcionar en el nivel alto de potencia, con una tolerancia aproximada de 1,5 dB. Todo sistema de transmisión AIS móvil cuenta con tres modos de operación diferentes: autónomo, asignado e interrogación. El modo por defecto debe ser el autónomo, en donde cada estación AIS administra sus propias transmisiones (reservar slots o elegir en cuáles se intentará transmitir) y resuelve potenciales conflictos con otras estaciones. En el modo asignado, las estaciones AIS recibirán mensajes de asignación que determinarán los intervalos de reporte o la agenda de transmisión (esto es, cuándo debe transmitir). El modo de interrogación permite operar sin conflicto en cualquiera de los otros dos modos. Dicho modo de operación consiste en que las estaciones respondan mensajes de interrogación en el mismo canal en donde estos fueron recibidos.

Existen varios tipos de mensajes, cada uno con funciones distintas, como será explicado más adelante. El intervalo de reporte de las estaciones AIS podrá variar según el mensaje que se deba transmitir. Esto es, puede ser que un mensaje 1 deba transmitirse cada 10 s, mientras que el mensaje 2 se transmite cada 30 s. Además, según la clase de la estación AIS (clase A o clase B), dichos intervalos también varían. En particular, la información estática y la relacionada con el viaje se debe enviar cada 6 minutos o cuando sea solicitado para ambos tipos de estaciones. Los mensajes de información estática contienen información como las dimensiones de la embarcación o su nombre. Por otro lado, el intervalo de reporte de los mensajes de información dinámica (como posición, velocidad, dirección de navegación) depende de la velocidad y alteraciones en el curso de la embarcación. Los distintos intervalos pueden observarse en las tablas 2.1 y 2.2, para estaciones de clase A y clase B, respectivamente. Observar que los intervalos cambian según el tipo de estación considerada. De aquí en adelante, las descripciones se centrarán en las embarcaciones de clase B, pues son el foco principal del proyecto.

2.2. Generalidades

Tabla 2.1: Intervalo de reporte para la información dinámica en embarcaciones de clase A. Tabla recuperada de la norma ITU-R M.1371-5 [16]. Se recuerda que un nudo es equivalente a 1,852 km/h.

Condiciones de la embarcación	Intervalo de reporte
Anclada o amarrada y moviéndose a una velocidad no mayor a 3 nudos.	3 minutos
Anclada o amarrada y moviéndose a una velocidad mayor a 3 nudos.	10 segundos
Velocidad entre 0 y 14 nudos.	10 segundos
Velocidad entre 0 y 14 nudos y cambiando el rumbo.	3,33 segundos
Velocidad entre 14 y 23 nudos.	6 segundos
Velocidad entre 14 y 23 nudos y cambiando el rumbo.	2 segundos
Velocidad mayor a 23 nudos.	2 segundos
Velocidad mayor a 23 nudos y cambiando el rumbo.	2 segundos

Tabla 2.2: Intervalo de reporte para la información dinámica en embarcaciones de clase B. Tabla recuperada de la norma ITU-R M.1371-5 [16].

Condiciones de la embarcación	Intervalo de reporte
Clase B SO a una velocidad no mayor a 2 nudos	3 minutos
Clase B SO a una velocidad entre 2 y 14 nudos	30 segundos
Clase B SO a una velocidad entre 14 y 23 nudos	15 segundos
Clase B SO a una velocidad mayor a 23 nudos	5 segundos
Clase B CS a una velocidad no mayor a 2 nudos	3 minutos
Clase B CS a una velocidad mayor a 2 nudos	30 segundos

2.2.3. Tipos de estaciones

Si bien el proyecto se centra en el uso de AIS para la seguridad marítima de embarcaciones, existen varios usos prácticos que se le dan a esta tecnología. Entre ellos se encuentra la búsqueda y rescate de aeronaves, u otros casos de riesgo como “hombre al agua” o posicionamiento de emergencia. Todos estos casos suelen estar asociados a situaciones límite, para los cuales existen mensajes reservados (como el 1 y el 14). Sobre estos usos no se profundizará a lo largo del proyecto pues escapan de sus objetivos.

Dentro del contexto del proyecto, orientado a la seguridad de navegación marítima, existen también distintos tipos de transmisores AIS que cumplen un rol en la comunicación. Entre ellos se encuentran las estaciones base y estaciones repetidoras.

Las estaciones base cumplen el rol fundamental de actuar como “semáforos”, es decir, actuar como fuente de sincronización para que las embarcaciones pue-

Capítulo 2. Sistema de Identificación Automática

dan tener acceso al tiempo UTC¹. Este aspecto es muy importante pues permite que todas las embarcaciones tengan la misma referencia temporal. Así, es posible determinar el comienzo de cada uno de los slots de transmisión. Las estaciones base también se pueden utilizar para asignar a embarcaciones qué slots específicos deben utilizar para su transmisión. También pueden enviar mensajes indicando el cambio de la frecuencia de transmisión (para transmitir en los canales 75 y 76, por ejemplo) y otros mensajes de manejo de canal².

Las estaciones repetidoras son utilizadas cuando se requiere cobertura extendida para los mensajes. Para esto se puede utilizar una o más estaciones repetidoras. Estos equipos no se analizan con profundidad a lo largo del proyecto pues no son parte del objetivo del mismo.

En lo que respecta a estaciones móviles, existen dos clases principales, las de clase A y las de clase B. Las estaciones móviles refieren a transmisores AIS que en general son utilizados en embarcaciones para enviar mensajes desde alta mar, aunque también deben hacerlo cuando están ancladas. Los mensajes que envían las estaciones móviles son principalmente de posición, velocidad y dirección, aunque existe una gran cantidad de mensajes cuyo contenido es variable. Se ahondará más adelante en los principales mensajes y su contenido.

Como ya fue mencionado, las estaciones móviles serán el foco principal del proyecto, particularmente las de clase B. Se considerarán dos tipos: clase B SO y clase B CS, según el mecanismo de acceso al medio compartido implementado.

2.3. Funcionamiento a nivel de capa física

La capa física tiene la función de transmitir una secuencia de bits desde la fuente de información hacia el enlace de datos. La transmisión se debe alternar entre los dos canales configurados. Por defecto, las transmisiones se efectúan en los canales A y B. Es posible especificar otras frecuencias de trabajo a partir de mensajes de control de canal. El tiempo máximo para pasar de un canal a otro es de 25 ms. En los slots de tiempo en los que se cambia de canal no se puede transmitir, principalmente porque el tiempo máximo de cambio de canal es casi la duración de un slot. Debe ser posible recibir información en los slots adyacentes a una transmisión, pero no se exige la capacidad de transmitir en slots adyacentes en dos canales distintos. Una vez el transmisor está sincronizado a cierta frecuencia central de un canal, su estabilidad frecuencial debe ser de ± 500 Hz o mejor. La tasa de transmisión de bits debe ser de $9.600 \text{ bits/s} \pm 50 \text{ ppm}$ ³.

En la figura 2.1 se presenta un diagrama de transmisión para AIS, una vez que se tienen los bits que representan a los mensajes. Se describen a continuación los

¹El tiempo universal coordinado (UTC, por sus siglas en inglés) es una referencia internacional de tiempo, utilizada para sincronización en sistemas de comunicación inalámbricos o en redes de telecomunicaciones, entre otros usos.

²Las estaciones bases podrían ser quienes *asignan* en el caso de tener una estación de clase A o clase B funcionando en modo asignado.

³En este contexto, 50 partes por millón hace referencia a que la tasa de transmisión podrá variar entre $9.599,52 \text{ bit/s}$ y $9.600,48 \text{ bit/s}$.

2.3. Funcionamiento a nivel de capa física

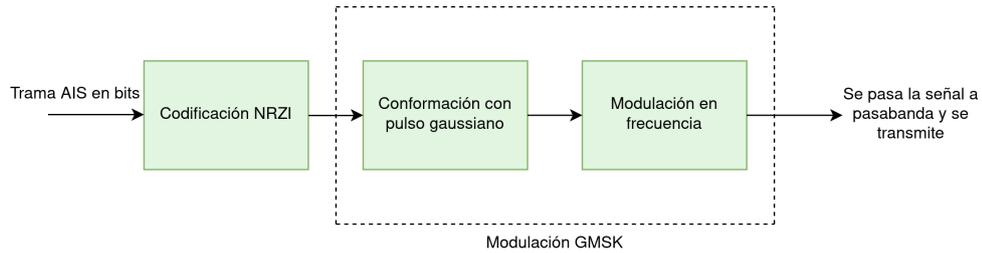


Figura 2.1: Diagrama de bloques para un transmisor AIS. Considerando que se cuenta con los bits asociados a una trama AIS, se procede a aplicar la codificación NRZI. Luego se conforman los bits con un pulso gaussiano y se modula la señal en frecuencia.

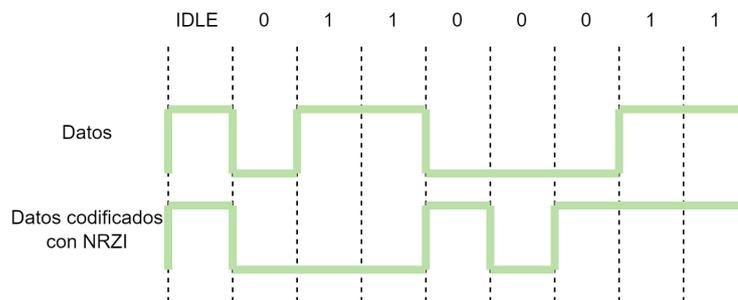


Figura 2.2: Ejemplo de funcionamiento de la codificación NRZI. Se observa que la señal codificada cambia de nivel cuando los datos tienen un bit en cero.

bloques involucrados.

El primer bloque corresponde a la codificación de los bits. Se utiliza NRZI para esta codificación, en una variante en que los cambios de nivel se hacen cuando se recibe un bit 0. Se mantiene el nivel hasta la llegada del próximo 0. Esto se ve ejemplificado en la figura 2.2 tomando como pulso conformador un pulso rectangular.

Una vez los bits han sido codificados se debe realizar la modulación GMSK. Se trata de un tipo de modulación de fase continua, que resulta en la transmisión de señales con una mejor eficiencia espectral. Para conformar la señal, será necesario un pulso conformador. Si se utilizara un pulso rectangular, se obtendría una señal con fase continua, pero la derivada de la fase no lo sería. De esta forma, se obtendrían señales que ocupan un mayor ancho de banda en frecuencia, resultando en una menor cantidad de canales a utilizar. Esto lleva a la utilización de un filtro gaussiano como pulso conformador, en lugar de un pulso rectangular, cuya expresión es

$$g(t) = \frac{Q\left(2\pi B\left(t - \frac{T}{2}\right)\right) - Q\left(2\pi B\left(t + \frac{T}{2}\right)\right)}{\sqrt{\ln 2}}. \quad (2.1)$$

En la ecuación 2.1 se tienen los parámetros B y T . El primer parámetro representa el punto con -3 dB de caída respecto a su pico. El segundo parámetro hace referencia a la duración del pulso en el tiempo. La función Q representa a la cola

Capítulo 2. Sistema de Identificación Automática

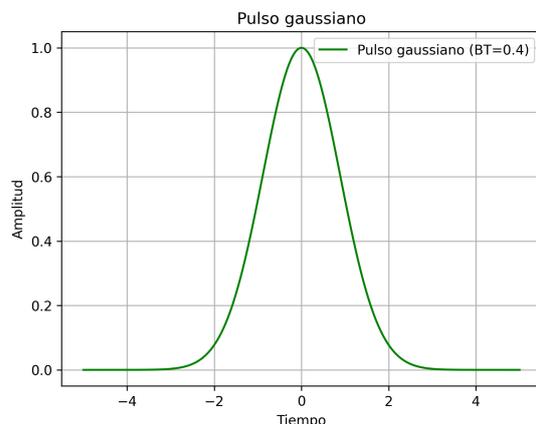


Figura 2.3: Gráfica de un pulso gaussiano en el tiempo. En particular, se tiene un parámetro BT de 0,4.

gaussiana [34], y está dada por la siguiente expresión

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt. \quad (2.2)$$

En la figura 2.3 se presenta el filtro gaussiano en el dominio del tiempo. Este filtro reduce el ancho de banda ocupado por la señal, pero aumenta la interferencia intersimbólica.

En la figura 2.4 se observan distintas representaciones del pulso al variar el producto entre B y la duración del pulso T . Dicho producto recibe el nombre de BT y es el parámetro que se fija en las aplicaciones. En el caso del transmisor AIS, BT tiene un valor máximo nominal de 0,4. En el caso del receptor, el valor máximo nominal es de 0,5. La figura 2.4 ilustra el pulso GMSK para valores de BT que van desde 0,1 hasta 1.

Los motivos para la utilización de GMSK son varios. Se logran señales sencillas de implementar con inmunidad a ruido o interferencias, y con envolvente constante. Esto último permite que el sistema siga funcionando incluso si los amplificadores de transmisión o recepción no se encuentran en zona lineal. Este esquema ha sido utilizado en comunicaciones inalámbricas como GSM y Bluetooth. Se cree que se propuso utilizar GMSK para AIS porque la tecnología fue desarrollada alrededor del mismo tiempo que GSM, por lo que se aprovechó el uso masivo de esta modulación. Finalmente, luego de conformar la señal con el pulso gaussiano, se aplica una modulación en frecuencia.

2.4. Funcionamiento a nivel de capa de enlace

La capa de enlace se divide en tres subcapas, cada una de ellas con funciones específicas. La primera es la subcapa de control de acceso al medio. Como el nombre lo indica, esta capa provee un mecanismo que garantice el acceso al canal para la transmisión de datos. Los mecanismos de acceso al medio están basados

2.4. Funcionamiento a nivel de capa de enlace

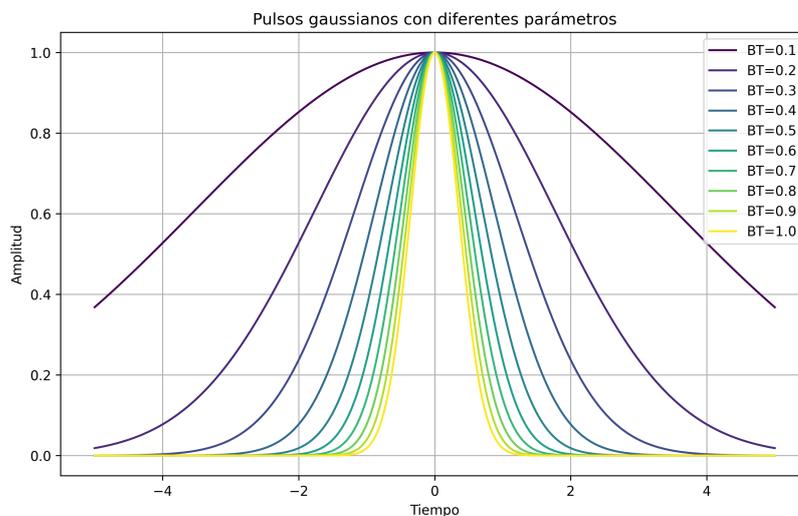


Figura 2.4: Gráfica de varios pulsos gaussianos en el dominio del tiempo, variando el parámetro BT . Se observa que, cuanto mayor es el parámetro BT , menor es el ancho de la gaussiana.

en TDMA. La subcapa de servicios del enlace de datos se involucra en el armado de los paquetes a transmitir. Se realizan conteos de control de integridad de los bits con verificaciones en las tramas (FCS, por sus siglas en inglés) y al contenido se le realiza bit stuffing⁴. La tercera subcapa es la entidad manejadora de enlaces, que se encarga de conformar bytes a partir de los bits de un mensaje AIS. De esta forma se ensamblan los paquetes de transmisión.

2.4.1. Subcapa 1: Control de acceso al medio

La subcapa de control de acceso al medio provee un mecanismo para que el dispositivo transmisor acceda al medio compartido. AIS es una tecnología de comunicaciones inalámbricas. El medio de transmisión es entonces el aire y debe ser compartido por diversos dispositivos al mismo tiempo. En particular, hay una porción del espectro radioeléctrico destinado para las comunicaciones marítimas al cual deben acceder de forma organizada. La estrategia utilizada para evitar que la información se entremezcle en el enlace de datos se basa en el acceso múltiple por división de tiempo (TDMA).

TDMA se basa en la división del tiempo en intervalos de corta duración. En cada uno de estos intervalos puede transmitir como máximo un solo dispositivo. En el caso de la tecnología AIS, el tiempo se divide en cuadros o frames, cada uno de duración de un minuto. Luego, cada cuadro se divide en un total de 2.250 slots de tiempo, cada uno con una duración cercana a 27 ms. Los slots se identifican con el índice correspondiente, empezando por el cero. Un transmisor debe iniciar su transmisión al comienzo del slot y debe apagarse luego de que el último bit

⁴Se agrega un cero luego de tener cinco unos seguidos. Esto se hace para limitar la cantidad de bits con valor 1 seguidos a enviar y para reservar banderas de comienzo y fin de paquetes.

Capítulo 2. Sistema de Identificación Automática

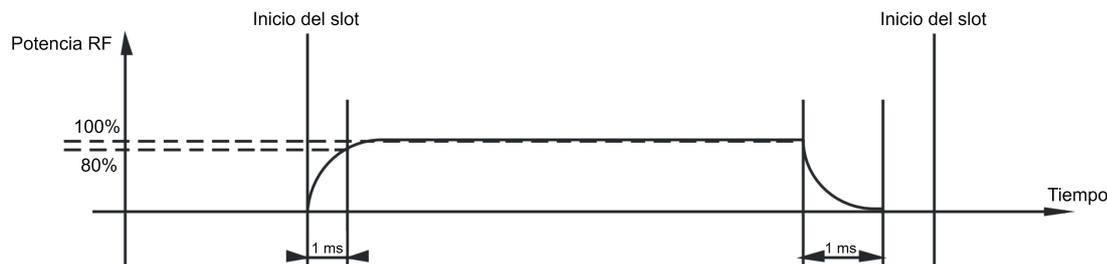


Figura 2.5: Diagrama en el tiempo de una transmisión AIS en un slot. Se observa que se tiene un período de 1 ms al inicio del slot para encender el transmisor. Además, se tiene que terminar la transmisión antes de comenzar el siguiente slot. Esta figura fue adaptada de la figura 5 de la norma ITU-R M.1371-5 [16].

del paquete sea enviado. Este proceso debe ocurrir por completo dentro del slot reservado para la transmisión, como se observa en la figura 2.5. En el caso de que una transmisión se retrase y se siga transmitiendo cuando llega el final del slot, la información debe dejar de modularse. Esto evita confundir dicha información con el mensaje del siguiente slot. Es interesante notar que los procesos de recepción no tienen por qué estar sincronizados a los límites temporales de los slots.

Para los transmisores se vuelve importante la sincronización con otras estaciones. Como referencia común de tiempo se utiliza UTC, una referencia universal que las estaciones pueden obtener a través de satélites (junto con la información de GPS). El comienzo de un cuadro AIS está sincronizado con el comienzo de un minuto UTC. Así, todas las estaciones pueden saber la distribución de los slots en el tiempo. La sincronización con los slots se vuelve más fina a través del intercambio de mensajes, ya que las estaciones que implementan SOTDMA deben indicar el número del slot actual. Esto permite que estaciones receptoras determinen el índice asignado al slot actual, o mejoren su sincronización. Cada un cierto intervalo de actualización (de 2 s en casos de estaciones móviles, y 3 s para estaciones base) es necesario resincronizarse con la fuente UTC considerada.

Sincronización para estaciones de clase B SO

Los sistemas AIS de clase B SO deben reportar de qué forma están accediendo a la información de UTC. Esto se indica agregando una bandera de estado de sincronización (*sync state*) en sus mensajes.

Una estación que tenga acceso directo a UTC tendrá un estado *UTC direct*. Podría ocurrir que algunas estaciones no tengan acceso a UTC. Deberán entonces sincronizarse a la referencia indicada por otros sistemas AIS cercanos, y fijar su estado de sincronización en *UTC indirect*. Solo es posible un nivel de sincronización UTC, por lo que una estación no podría sincronizarse a otra que indique UTC indirecto. Si no se puede acceder a UTC, ni a estaciones móviles directamente sincronizadas, es posible sincronizarse a estaciones base que estén dentro del rango de alcance. Estas estaciones fijarán su estado en *base station*, indicando si se recibe de forma directa o indirecta. El restante caso posible es no recibir UTC directo, ni indirecto, de estaciones móviles o estaciones base. Como último recurso, los

2.4. Funcionamiento a nivel de capa de enlace

dispositivos podrán sincronizarse a la estación cercana que indique recibir la mayor cantidad de otras estaciones. El estado se fijará en *number of received stations*.

Los estados de sincronización previamente descritos se mapean a valores numéricos para ser transmitidos. La bandera *sync state* mencionada tomará los siguientes valores. El estado *UTC direct* se asocia al valor 0. El estado *UTC indirect* se representa con el valor 1. El estado *Base direct* tiene el valor 2. Si se tiene un estado *Base indirect* o de *Number of received stations*, la bandera de *sync state* se fijará en 3.

Sincronización para estaciones de clase B CS

En el caso de los sistemas AIS de clase B CS se tienen dos modos posibles de sincronización. No se transmite en los mensajes el modo en que se encuentre la estación, sino que es simplemente una variable interna. La bandera *sync state* previamente mencionada sí se agrega en los mensajes transmitidos. Debe estar fija en un valor de 3, ya que estas estaciones no pueden ser utilizadas como una fuente de sincronización.

El modo de sincronización 1 se utiliza cuando se recibe alguna estación AIS que no sea de clase B CS. En este caso, se deben sincronizar los slots de tiempo a los reportes de posición que realiza esa otra estación. Será necesario considerar los retardos de propagación de las distintas estaciones recibidas. En el caso de que pasen 30 s sin recibirse ninguna estación de clase A o de clase B SO, la estación debe cambiarse al modo de sincronización 2.

El modo de sincronización 2 es utilizado cuando únicamente se reciben estaciones de clase B CS. Dado que no se cuenta con estaciones externas que se puedan usar como fuentes de sincronización, se deberá determinar el inicio de los períodos de tiempo según su reloj interno. En cuanto se comiencen a recibir estaciones de clase A o de clase B SO, se pasa al modo de sincronización 1.

2.4.2. Subcapa 2: Servicio de enlace de datos

La subcapa de servicio de enlace de datos permite la activación y liberación del enlace de datos para la transferencia de datos. Esto debe realizarse en los slots en que deba transmitirse. Es necesario seguir el comportamiento presentado en la figura 2.5. En el caso de estar en los slots marcados como libres o reservados por otras estaciones, cada equipo deberá estar en el modo de recepción.

Para las transmisiones se utiliza un protocolo orientado a bits basado en el protocolo HDLC⁵. En la figura 2.6 se presenta un esquema del formato del paquete, con un largo total de 256 bits, equivalentes a un slot de transmisión. Los paquetes transmitidos tendrán un cierto encabezado y una carga útil. En el encabezado se agrega un valor de FCS para permitir la detección de errores en recepción. Se describen brevemente los campos del paquete.

⁵Protocolo sincrónico orientado a bits en donde la información se organiza en cuadros, se envía a través de la red y se verifica su llegada exitosa [35].

Capítulo 2. Sistema de Identificación Automática

Secuencia de entrenamiento	Bandera de inicio	Campo de datos	CRC	Bandera de finalización	Buffer
----------------------------	-------------------	----------------	-----	-------------------------	--------

Figura 2.6: Trama de un mensaje AIS. Se observan los seis campos principales, resultando en un total de 256 bits de largo. Esta figura fue adaptada de la figura 6 de la norma ITU-R M.1371-5 [16].

- **Secuencia de entrenamiento.** Consiste de 24 bits alternando entre ceros y unos. Dado que este patrón será modificado al utilizar la codificación NRZI, la secuencia puede comenzar en cero o en uno.
- **Bandera de comienzo.** Se trata de 8 bits utilizados para detectar el comienzo de un paquete: 01111110.
- **Datos.** Campo de 168 bits de largo por defecto. En el caso de que fuera necesario transmitir más información que eso, es posible que una estación transmita por hasta cinco slots consecutivos, agregando una sola vez el preámbulo, las banderas, el FCS y el buffer.
- **Frame check sequence (FCS).** Se utiliza una verificación por redundancia cíclica (CRC) con un polinomio de 16 bits para calcular la suma de verificación. Solo debe considerarse el campo de datos para dicha suma.
- **Bandera de finalización.** Se trata de una bandera idéntica a la bandera de comienzo.
- **Buffer.** Campo de 24 bits o menos. Se reservan los primeros 4 bits para el agregado de bit stuffing, y en el caso de mensajes de largo variable se pueden necesitar bits extra. Los siguientes 14 bits consideran retardos introducidos debido a la distancia, que equivalen a casi 236 millas náuticas de protección. Los bits restantes se utilizan como medida ante fluctuaciones en la sincronización. En el caso de las estaciones de clase B CS, se considera además un buffer inicial previo a la secuencia de entrenamiento, de 23 bits de duración. Este buffer agrega bits para tener en cuenta ciertos retardos⁶.

A todos los bits entre las banderas de comienzo y final se les aplicará bit stuffing. Esto hace referencia al agregado de un cero luego de tener cinco unos seguidos. En recepción se invierte este proceso como corresponde.

2.4.3. Subcapa 3: Entidad manejadora de enlaces

La subcapa de entidad manejadora de enlaces regula el funcionamiento de las otras dos subcapas de la capa de enlace y de la capa física. Las estaciones AIS de clase B pueden basar su acceso al medio compartido en el mecanismo SOTDMA

⁶Retardos de recepción, retardos asociados a sincronización propia, retardo asociado a la ventana de detección de CS, entre otros.

2.4. Funcionamiento a nivel de capa de enlace

o en el mecanismo CSTDMA [36]. En el caso de utilizar SOTDMA, se deben utilizar además los mecanismos de TDMA de acceso aleatorio (RATDMA, por sus siglas en inglés) y TDMA incremental (ITDMA, por sus siglas en inglés). Estos mecanismos se utilizan para la elección de los primeros slots. Una vez obtenidos, las transmisiones siguientes se harán en base a su intervalo de reporte. Los esquemas SOTDMA y CSTDMA operan de forma continua y en paralelo sobre el medio compartido.

Acceso al enlace de datos para estaciones de clase B SO

El acceso al medio por parte de las estaciones de clase B SO considera tres mecanismos: RATDMA, ITDMA y SOTDMA. El funcionamiento luego del encendido de la estación es el siguiente. Se selecciona un primer slot de transmisión de forma aleatoria utilizando el mecanismo RATDMA. La transmisión en ese primer slot se realiza utilizando ITDMA, y se continúa utilizando este mecanismo durante un minuto completo. Una vez completada esta etapa de ingreso a la red se pasa al funcionamiento en régimen, utilizando SOTDMA.

Se describen algunos conceptos previos y luego se procede a describir cómo se realiza el acceso al medio compartido en estas estaciones AIS.

Selección de slots de tiempo Los mecanismos de acceso al medio compartido basados en TDMA dividen el eje temporal en cuadros y slots de tiempo. SOTDMA buscará implementar un esquema de reservas de los slots. Se guardará en un mapa o agenda local la disponibilidad de los slots, asignando a cada uno alguno de los siguientes estados.

- **Libre:** Se consideran como libres aquellos slots que no estén reservados, o que lo estén y no hayan sido utilizados en los últimos tres cuadros. Este tipo de slot puede considerarse un slot candidato para reservas o transmisiones.
- **Reserva interna:** El slot está reservado por la propia estación y puede ser utilizado para transmitir.
- **Reserva externa:** El slot está reservado por una estación externa y no puede ser utilizado para transmitir.
- **Disponible:** El slot está reservado por una estación externa. Sin embargo, es un posible candidato para reutilización de slots (que se explicará más adelante).
- **No disponible:** El slot está reservado por una estación externa y no es un posible candidato para reutilización de slots. Suelen tratarse de slots reservados por una estación base.

Para cada transmisión es necesario determinar qué slot será utilizado. Se definen slots candidatos como aquellos que podrán ser considerados para reservas o transmisiones. Los slots candidatos serán los que estén libres en ambos canales de

Capítulo 2. Sistema de Identificación Automática

operación. Se pueden utilizar slots reservados de forma externa como slots candidatos si la estación que los reservó no está anclada y no se ha recibido información de ella en los últimos tres minutos. Esto es equivalente a asumir que la estación se está moviendo y se alejó lo suficiente como para que no se generen interferencias al utilizar los slots que reservó. Podrían también utilizarse slots disponibles como slots candidatos, utilizando los mecanismos de reutilización de slots que se describen en esta sección. Dado el tiempo de cambio de canal previamente mencionado, los slots adyacentes a un slot utilizado por la propia estación no podrán ser considerados como candidatos en el otro canal⁷. En el caso de las estaciones de clase B SO móviles, los slots candidatos para transmitir ciertos mensajes (los mensajes 6, 8, 12 y 14) deben estar libres.

Se introduce el concepto de intervalo de selección (SI, por sus siglas en inglés) como un intervalo de tiempo conformado por slots. Considerando el intervalo de reporte actual de la estación, se define un SI centrado en cada momento en el que corresponda transmitir. Aquellos slots candidatos incluidos en el SI serán los potenciales slots de transmisión. Una vez definido el intervalo de selección, y considerando los slots dentro del mismo, todos los slots candidatos tienen la misma probabilidad de resultar elegidos.

En casos en los cuales se dificulta encontrar slots candidatos, es posible reutilizar slots. La reutilización de slots consiste en tomar slots reservados por estaciones lejanas, cuya interferencia sea despreciable, y agregarlos al conjunto de slots candidatos. Para reutilizar slots, la propia estación debe tener acceso a su posición por GPS, y no podrá reutilizar slots de estaciones que indiquen no tener referencia de posición disponible.

En principio, solo se pueden reutilizar slots de estaciones móviles, aunque se podría considerar reutilizar slots de una estación base si está ubicada a más de 120 millas náuticas (NM, por sus siglas en inglés). Si se reutiliza el slot de una cierta estación, entonces la misma deberá ser excluida de los mecanismos de reutilización de slots durante un período de un cuadro (un minuto). La reutilización de slots resulta en slots candidatos para la selección aleatoria. Se busca que siempre haya cuatro slots de entre los que elegir, pero podría ser que se consigan menos. Los slots candidatos podrán ser seleccionados utilizando reutilización de slots según las siguientes reglas.

Se agregarán al conjunto de slots libres, en este orden, todos los slots que:

1. estén clasificados como libres en el canal de selección, y como disponibles en el otro canal.
2. estén clasificados como disponibles en el canal de selección, y como libres en el otro canal.
3. estén clasificados como disponibles en ambos canales.

⁷Es decir, si en el canal A se transmite en el slot 32, en el canal B los slots 31 y 33 no podrán ser considerados como candidatos (ya que se requiere un tiempo de aproximadamente un slot de duración para realizar el cambio de canal).

2.4. Funcionamiento a nivel de capa de enlace

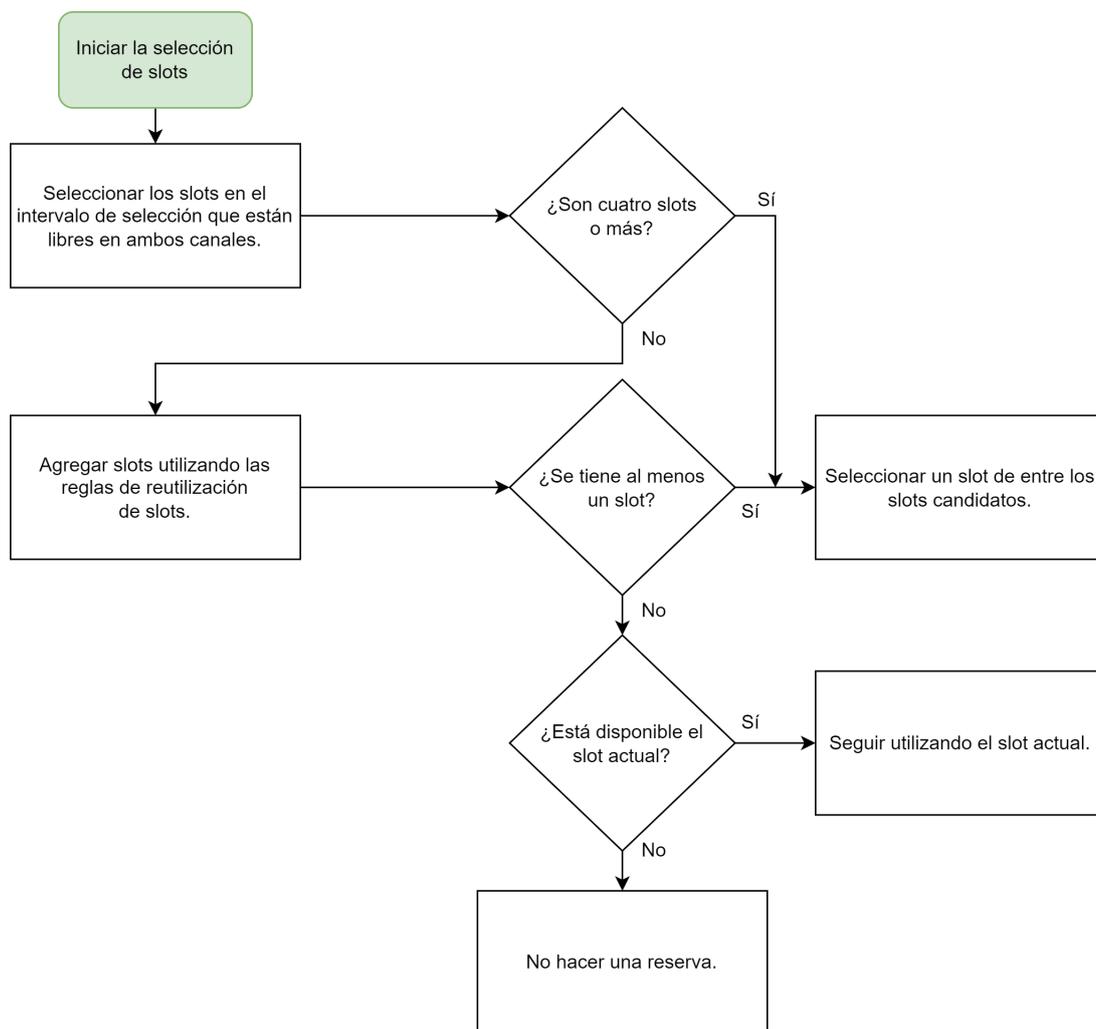


Figura 2.7: Diagrama de flujo asociado al proceso de selección de un slot para transmitir. Se destaca el mecanismo de reutilización necesario cuando no hay suficientes slots libres. Esta figura está basada en la figura 9 de la norma ITU-R M.1371-5 [16].

- estén clasificados como libres en el canal de selección y como no disponibles en el otro canal.
- estén clasificados como disponibles en el canal de selección y como no disponibles en el otro canal.

En la figura 2.7 se presenta el proceso de selección entre los slots candidatos para la transmisión. En el mismo se define que, en el caso de que una primera preselección de slots libres no resulte en al menos cuatro slots candidatos, se deben aplicar mecanismos de reutilización de slots. Si esto no resultara en encontrar slots candidatos, se podría pedir mantener la reserva del slot actual por un cuadro de tiempo más.

Capítulo 2. Sistema de Identificación Automática

Inicialización de la estación Al encenderse, la estación deberá escuchar y monitorear los dos canales por un minuto. Durante este tiempo se obtiene información del medio, como la actividad en él, los identificadores de otras estaciones y las posiciones reportadas, además de ir generando el mapa interno de asignaciones de los slots. Una vez se completa el período de inicialización, la estación ingresa al modo de operación que corresponda (autónomo, asignado, interrogación) y comienza a transmitir.

RATDMA El mecanismo RATDMA es utilizado cuando una estación debe reservar un slot sin poder anunciar la reserva previamente. Esto se aplica al ingresar al enlace de datos o para mensajes no periódicos. Por defecto, se debe evitar el uso de RATDMA. Debería utilizarse siempre un mensaje que anuncie una futura transmisión.

Para determinar los slots a utilizar, RATDMA utiliza un algoritmo p-persistente. El algoritmo consiste en, al detectar un slot candidato, seleccionar un valor entre 0 y 100, y compararlo con un parámetro de transmisión actual. Si se cumple que la probabilidad es menor o igual al parámetro de transmisión actual, la transmisión deberá ocurrir en el slot candidato detectado. En caso contrario, se deberá incrementar el parámetro de transmisión actual y la estación deberá esperar al próximo slot candidato del cuadro.

ITDMA El mecanismo ITDMA permite que la estación anuncie previamente el slot a utilizar en la próxima transmisión. Se seleccionan slots de entre slots candidatos.

ITDMA se utiliza en tres ocasiones: durante el ingreso al enlace de datos (luego de un reinicio de la estación), para cambios o transiciones temporales en intervalos de reporte periódicos y para el anuncio previo a mensajes relacionados a seguridad.

Durante el ingreso al enlace de datos, el funcionamiento consiste en determinar un primer slot con RATDMA y realizar esa transmisión con ITDMA. Esa transmisión indicará dos aspectos: el próximo slot de transmisión a utilizar (indicando un *offset* con la diferencia entre el slot actual y el próximo), y además reserva el slot actual para ser utilizado en el próximo cuadro (con una bandera de mantener, *keep*). Recordar que en ese próximo cuadro comienza la operación autónoma y continua, por lo que se pasará al uso de SOTDMA.

SOTDMA El mecanismo SOTDMA es el utilizado por las estaciones móviles que operan en modo autónomo o asignado. Se trata de una estrategia que resuelve los conflictos entre las agendas de diversas estaciones de forma rápida y sin necesidad de intervención de una estación controladora. SOTDMA se aplica en el funcionamiento en régimen de la estación AIS al realizar transmisiones de mensajes periódicos. La idea básica de funcionamiento es que cada vez que una estación transmite avisa en su mensaje cuál será el próximo slot que utilizará para transmitir y reserva el slot actual para varios cuadros en el futuro. De esta forma, el resto de las estaciones saben cuándo estará ocupado el canal, y pueden determinar su propia agenda.

2.4. Funcionamiento a nivel de capa de enlace

Modo de operación autónomo El modo de operación autónomo comienza con una fase de inicialización, en la cual se espera un minuto antes de ingresar al enlace de datos. Durante ese intervalo de tiempo se monitorea y se lleva un registro de la actividad del medio compartido y de qué estaciones forman parte de él. Se construye el mapa interno con las reservas externas para considerar como referencia en los siguientes cuadros.

Una vez transcurre el primer minuto, la estación debe ingresar a la red y transmitir siguiendo su propia agenda (determinada a partir del intervalo de reporte que corresponda). Sigue entonces una fase de ingreso a la red, durante la cual se selecciona el primer slot de transmisión, utilizando la estrategia RATDMA. Este primer slot de un cuadro se denomina slot nominal de inicio (NSS, por sus siglas en inglés), y se selecciona entre el slot actual y los siguientes incremento nominal (NI, por sus siglas en inglés) slots. La cantidad de slots NI depende de la tasa de reporte (Rr), de forma que $NI = 2,250/Rr$. Las siguientes transmisiones que sean repetidas suelen tomar como referencia al NSS. Si se están usando ambos canales para las transmisiones, se selecciona el NSS para un primer canal A, y el NSS del canal B seguirá la ecuación $NSS_B = NSS_A + NI$. El slot nominal (NS) será el slot usado como centro para los intervalos de selección de slots. En la primera transmisión de un cuadro, el NSS y el NS son el mismo. Se tiene que NSS es el primer slot en el que se transmite en un cuadro, y se llamará NS a todos aquellos slots en los que se centrarán los intervalos de selección de slots candidatos para transmitir. Los índices de slots que serán considerados NS dependen de la ecuación

$$NS = NSS + (n \times NI), \quad (2.3)$$

en el caso de utilizar un solo canal para las transmisiones. El parámetro n se encuentra entre 0 y Rr , marcando la dependencia de la tasa de reporte. Como es de esperar, al utilizar dos canales alternados para las transmisiones, en cada uno se tendrán la mitad de los NS necesarios. En el canal A se expresa con la ecuación

$$NS_A = NSS_A + (n \times 2 \times NI), \quad (2.4)$$

mientras que en el canal B se expresa con

$$NS_B = NSS_A + NI + (n \times 2 \times NI), \quad (2.5)$$

donde n varía entre 0 y $Rr/2$.

En la fase de ingreso a la red se define entonces el $NSS = NS$ del primer cuadro, se determina alrededor de ese slot central un intervalo de selección (SI), que se define a partir de

$$SI = [NS - 0,1.NI; NS + 0,1.NI]. \quad (2.6)$$

Dentro del SI se determinan los slots candidatos, y se elige de forma aleatoria uno para que sea el slot nominal de transmisión (NTS). Este NTS deberá ser marcado como reserva interna, y se le debe asociar un *time out* de entre dos constantes fijas TMO-MIN y TMO-MAX. Este valor de *time out* indicará durante

Capítulo 2. Sistema de Identificación Automática

cuántos cuadros se mantendrá esta reserva como válida. Al haber seleccionado el NTS, resta esperar a que llegue ese slot de tiempo para pasar a la siguiente fase.

La fase del primer cuadro consiste en pasar un minuto entero determinando los próximos NS y NTS, y reservando los slots fijos como NTS para los siguientes cuadros. Se transmiten reportes especiales de posición (mensaje 3) utilizando el mecanismo ITDMA. Una vez pasó un minuto completo, el *offset* debería colocarse en cero en la última transmisión, para indicar que no se harán más reservas. Al terminar el primer minuto, las reservas iniciales de slots deberían haberse realizado, y debería comenzar la operación normal.

Finalmente se llega a la fase de operación continua, en la que la estación se mantiene hasta que sea apagada, se le cambie a modo de operación asignado o cuando cambie su intervalo de reporte. El funcionamiento en régimen de esta fase consiste en esperar a que llegue el NTS previamente reservado para la transmisión. Al llegar, se disminuye en uno su *time out*, que se incluye en las transmisiones e indica por cuántos cuadros está reservado el slot. En el caso de que el *time out* de un cierto slot llegue a cero, será necesario seleccionar un nuevo NTS, ya que expiró la reserva realizada. Nuevamente se define un SI alrededor del NS, en el cual se buscan slots candidatos y se selecciona uno de forma aleatoria. También se incluye en la transmisión el *offset*, que en los casos en que el *time out* expire, indica cuántos slots faltarían para el NTS reservado para el siguiente cuadro, y en caso contrario será fijo en cero. Teniendo el *time out* y el *offset*, se transmite en el slot definido como NTS, y se procede a esperar a que llegue el siguiente NTS.

En el caso de que sea necesario cambiar el intervalo de reporte, la estación debe ingresar a la fase de cambio de intervalo de reporte⁸. Se deberán reagendar las transmisiones periódicas de forma que funcionen para el nuevo intervalo de reporte. El cambio implica cambiar los parámetros NS, NI y SI previamente definidos y hacer nuevas reservas utilizando ITDMA, o aprovechando slots que ya pudiera tener reservados.

Modo de operación asignado Es posible que se le indique a una embarcación operar en modo asignado. Es decir, deberá operar de acuerdo a un cronograma de transmisión específico (asignado por otra estación, usualmente estaciones base) en ambos canales de funcionamiento. El modo asignado afecta únicamente las transmisiones de reporte de posición y se mantienen los criterios para el cambio del intervalo de reporte.

Las asignaciones recibidas tienen una duración limitada en el tiempo. En el caso de que se reciban nuevas instrucciones, se pasa a funcionar según la asignación más reciente. Si se recibe un nuevo valor para el intervalo de reporte la planificación de las transmisiones se realiza de la misma forma que en modo autónomo, siguiendo un esquema SOTDMA. Es posible también recibir una cierta asignación de los slots a utilizar para las transmisiones periódicas.

⁸Si el nuevo intervalo se mantiene por al menos dos minutos, se reagendan las transmisiones. Si el cambio de intervalo de reporte durara menos que ese tiempo, simplemente se insertan algunas transmisiones ITDMA entre las transmisiones SOTDMA ya planificadas.

2.4. Funcionamiento a nivel de capa de enlace

Si la asignación actual se termina, la embarcación debe volver al modo autónomo. El cambio comienza al detectar un slot asignado con *time out* nulo. Se comienza el funcionamiento desde la fase de ingreso a la red.

Acceso al enlace de datos para estaciones de clase B CS

El acceso al medio por parte de las estaciones de clase B CS se realiza con el mecanismo CSTDMA.

Slots de tiempo El funcionamiento de las estaciones de clase B CS se basa en los slots de tiempo previamente definidos, cada uno con una duración aproximada de 27 ms. En este caso no se cuenta con un mapa de slots que indique si los mismos han sido o no reservados. Cada uno tendrá asociado uno de los siguientes estados, como resultado del proceso de sensado del canal.

- **Libre:** El slot está disponible y no se identificó a ninguna otra estación utilizándolo.
- **Utilizado:** El canal está siendo utilizado por otra estación.
- **No disponible:** Se indican los slots como no disponibles si son reservados por estaciones base utilizando el mensaje 20. Estos no deberán ser considerados como candidatos para uso de la propia estación hasta que pase un cierto *time out* (indicado en el mensaje 20 o con un máximo de 3 minutos).

CSTDMA El mecanismo CSTDMA consiste en sensar el canal al inicio de los slots que, según el intervalo de reporte, podrían utilizarse para transmitir. El proceso a describir a continuación se ilustra en la figura 2.8. Como primer paso, se mide el nivel de ruido del canal dentro de una ventana de tiempo de $1.146 \mu\text{s}$ dentro del slot de interés, empezando en $833 \mu\text{s}$ y terminando $1.979 \mu\text{s}$ después del comienzo del slot. El hecho de ignorar los primeros $833 \mu\text{s}$ del slot (que corresponden a 8 bits de información) permite considerar retardos de propagación y tiempos de apagado de transmisiones externas. Las embarcaciones de clase B CS no deben transmitir en ningún período de tiempo en el que, durante la detección de la ventana, se detectó una señal con un nivel mayor que el umbral.

El umbral de detección de CS se determina a partir de intervalos de 60 s en cada canal de recepción por separado. El valor del umbral se conforma por el nivel mínimo de potencia medido en ese intervalo, más un *offset* de 10 dB. El umbral mínimo de detección debe ser -107 dBm y el ruido de fondo debe ser medido en un rango de al menos 30 dB. Esto resulta en un umbral máximo de -7 dBm.

Una vez la estación determinó que puede transmitir, inicia su transmisión encendiendo la potencia RF como se indica en la figura 2.5. Luego de transmitir deberá apagarse hasta que le corresponda transmitir nuevamente. En el caso que finalice el slot y la transmisión no haya terminado, entonces se debe dejar de modular la información en ese instante. Esto se realiza con el objetivo de no interferir con las transmisiones de otras estaciones.

El mecanismo CSTDMA tiene los siguientes pasos asociados.

Capítulo 2. Sistema de Identificación Automática

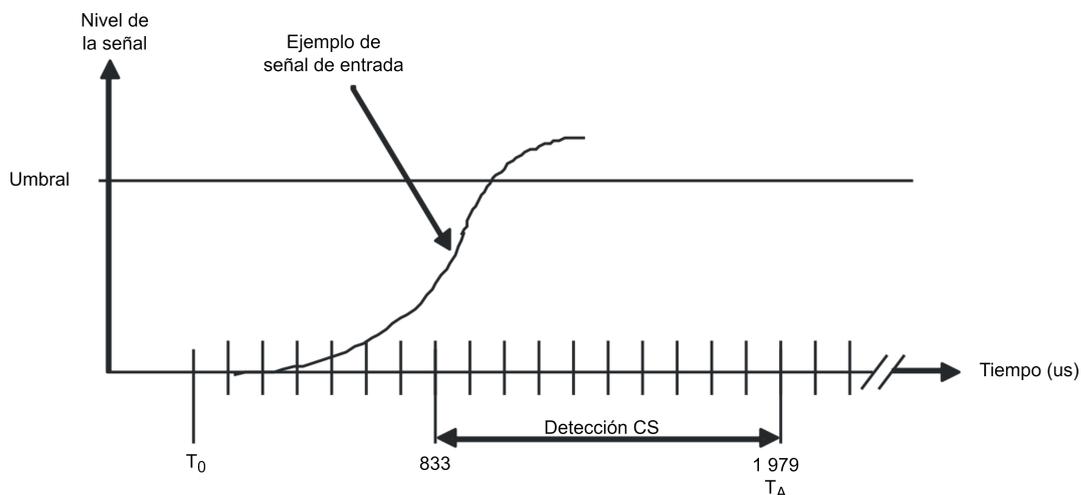


Figura 2.8: Esquema en el tiempo del sensado del nivel de ruido del canal realizado por una estación CS. Se destaca el monitoreo del canal solamente al inicio del slot, entre los instantes de $833 \mu s$ y $1.979 \mu s$.

1. En función del intervalo de reporte actual, identificar el momento exacto en el cual se debe transmitir la información propia. El slot de ese instante de tiempo se llamará tiempo de transmisión nominal (NTT, por sus siglas en inglés). Definir el intervalo de transmisión (TI, por sus siglas en inglés) como un conjunto de slots centrado en el slot NTT. El largo total del TI será el mínimo entre 10 s y el resultado de dividir el intervalo de reporte entre tres.
2. Definir aleatoriamente 10 slots candidatos dentro del TI.
3. Empezando del primer slot candidato, medir el nivel de ruido en el canal durante la ventana previamente mencionada. Si el slot está libre, transmitir. En caso contrario, esperar al siguiente de los 10 slots candidatos seleccionados.
4. Si ninguno de los 10 slots permitió transmitir, abandonar la transmisión.

Modo de operación autónomo Al operar en modo autónomo, la estación determina su propio cronograma de transmisión. En la fase de inicialización, cuando la estación se enciende, es necesario monitorear los canales durante un minuto. Esto permite la sincronización con las transmisiones del enlace de datos y determinar el umbral de detección de CS. Una vez pasa este tiempo inicial, se comienza el funcionamiento en régimen. La primera transmisión autónoma es el reporte de posición asociado al mensaje 18.

Modo de operación asignado El modo asignado implica que la estación utilice un cronograma de transmisión asignado por una estación base para los reportes de posición. Se pasa a estar en este modo al recibir un mensaje 22 de asignación individual, o con un mensaje 23 de asignación de grupo. Si la estación pertenece

2.5. Mensajes existentes

al grupo al que se le envió el mensaje, debe ingresar en modo asignado. Al asignar una agenda de transmisión a una estación, esta deberá enviar las transmisiones con un *time out* seleccionado de forma aleatoria entre 4 y 8 minutos después de la primera transmisión. Después del tiempo de *time out*, la estación debe volver al modo autónomo. Si se asigna además una tasa de reporte específica, el primer reporte de posición se realizará después de un intervalo de tiempo aleatorio entre el momento en que llegó el mensaje 23, y el intervalo asignado.

Receptor interno de GNSS para el reporte de la posición La estación de clase B CS deberá tener un receptor de GNSS interno, que obtenga información de posición, velocidad en tierra (SOG, speed over ground) y dirección en tierra (COG, course over ground). En el caso de que el receptor GNSS deje de funcionar, la estación no debe transmitir mensajes 18 y 24 a menos que sea interrogada por una estación base.

2.5. Mensajes existentes

En cuanto al contenido de la información a enviar, existen distintos tipos de mensajes. Existen mensajes que son enviados de forma periódica según un intervalo de reporte. Como ya fue mencionado, este intervalo de reporte depende de qué tipo de estación AIS sea el transmisor, la velocidad de la embarcación y su alteración de curso. Hay mensajes cortos relacionados con la seguridad que contienen información relevante en cuanto a la navegación o aspectos meteorológicos. Existen mensajes direccionados, es decir, mensajes que son dirigidos específicamente a un usuario destino. El destinatario se identifica según el MMSI, pero en general los mensajes enviados por estaciones móviles son de difusión. En otras palabras, aquellos que estén dentro del alcance de la estación emisora recibirán los mensajes.

La norma ITU-R M.1371-5 define un total de 27 mensajes para la tecnología AIS. Algunos de estos pueden ser enviados únicamente por estaciones base, otros solo por estaciones AIS de clase A y otros son obligatorios para todas las estaciones. Se analizan aquellos mensajes que deben ser interpretados o transmitidos por estaciones AIS de clase B. Estos mensajes se encuentran enumerados en las tablas 2.3 y 2.4, donde se presenta una breve descripción de cada uno de ellos. Además, se indica cuáles mensajes pueden o deben ser transmitidos y/o recibidos por estaciones AIS de clase B SO o CS. Si bien en la tabla se ordenan según el número de mensaje, se mencionarán según su funcionamiento y cómo se relacionan entre sí.

Desde el punto de vista de la recepción, las estaciones de clase B deben poder recibir y procesar los reportes de estaciones base. Estos son los mensajes que se identifican con el número 4 y las estaciones de clase B CS los utilizan principalmente para saber si están en un rango de 120 millas náuticas de la estación base. En caso afirmativo, deberán considerarse los mensajes 20 y 23 (que se explicarán más adelante). También se debe poder recibir y procesar los mensajes 15 y 22. El mensaje 15 corresponde a interrogaciones. Las estaciones de clase B CS responden a interrogaciones por los mensajes 18 y 24 que serán descritos en esta sección. El

Capítulo 2. Sistema de Identificación Automática

Tabla 2.3: Mensajes AIS con su número identificador. Se detalla cuáles deben ser transmitidos o recibidos por estaciones AIS de clase B SO y CS. La abreviación TX hace referencia a la transmisión, mientras que la abreviación RX hace referencia a la recepción.

ID	Nombre del mensaje	Clase B SO		Clase B CS		Comentarios
		TX	RX	TX	RX	
1	Reporte de posición (periódico)	Sí	Sí	No	Opcional	
2	Reporte de posición (asignado)	Sí	Sí	No	Opcional	
3	Reporte de posición (interrogación)	Sí	Sí	No	Opcional	
4	Reporte de estación base	No	Sí	No	Sí	Las estaciones de clase B CS deben obedecer la regla de las 120 NM.
5	Datos de estadísticas y de la expedición	Sí	Sí	No	Opcional	
6	Mensaje binario dirigido	Sí	Sí	No	No	
7	Mensaje binario de reconocimiento	Sí	Sí	No	No	
8	Mensaje binario de difusión	Sí	Sí	No	Opcional	
9	Reporte de posición estándar para búsquedas y rescates por aeronaves	Sí	Sí	No	Opcional	
10	Consulta de tiempo UTC y fecha	Sí	Sí	No	No	
11	Respuesta de tiempo UTC y fecha	Sí	Sí	No	Opcional	
12	Mensaje dirigido asociado a la seguridad	Sí	Sí	No	Opcional	Las estaciones de clase B CS podrían transmitir esta información con el mensaje 14.
13	Mensaje de reconocimiento asociado a la seguridad	Sí	Sí	Opcional	No	Las estaciones de clase B CS deben transmitir si se implementa el mensaje 12.
14	Mensaje de difusión asociado a la seguridad	Sí	Sí	Opcional	Opcional	Las estaciones de clase B CS deben transmitir únicamente con texto predefinido.

mensaje 20 se corresponde con reserva de slots por parte de una (o más) estación base. En caso de que una embarcación de clase B CS se encuentre dentro de un rango de 120 millas náuticas (lo cual es determinado con el mensaje 4) de la estación base, deberá marcar esos slots como ocupados para evitar utilizarlos.

El mensaje 22 es enviado por estaciones base para indicar los parámetros del enlace VHF para el área geográfica que le fue designada. Esto será adoptado por aquellos que, con el mensaje 4, determinen estar en un rango de 120 millas náuticas de la estación base. Este mensaje también puede ser utilizado por la estación base para indicarle a una estación AIS específica que use determinados parámetros del enlace. Por último, el mensaje 23 es un comando de asignación de grupo y

2.5. Mensajes existentes

Tabla 2.4: Mensajes AIS con su número identificador. Se detalla cuáles deben ser transmitidos o recibidos por estaciones AIS de clase B SO y CS. La abreviación TX hace referencia a la transmisión, mientras que la abreviación RX hace referencia a la recepción.

ID	Nombre del mensaje	Clase B SO		Clase B CS		Comentarios
		TX	RX	TX	RX	
15	Interrogación	Sí	Sí	No	Sí	Las estaciones de clase B CS deben responder eventuales interrogaciones con los mensajes 18 y 24.
16	Comando de modo de asignación	No	Sí	No	No	El mensaje 23 podría ser utilizado por una estación de clase B CS.
17	Mensaje binario de difusión de GNSS diferencial (DGNSS)	No	Sí	No	Opcional	
18	Reporte de posición estándar de equipamiento de clase B	No	Sí	Sí	Opcional	Las estaciones de clase B CS deben encender la bandera CS en el bit 143.
19	Reporte de posición extendido de equipamiento de clase B	No	Sí	Sí	Opcional	Se debe transmitir por la clase B CS únicamente en respuesta a una interrogación de una estación base.
20	Mensaje de administración del enlace de datos	No	Sí	No	Sí	Las estaciones de clase B CS deben recibir y evaluar si se cumple la regla de las 120 NM previo a contestar un mensaje 4.
21	Reporte de ayuda a la navegación	Sí	Sí	No	Opcional	
22	Mensaje de administración del canal	No	Sí	No	Sí	La respuesta de estaciones de clase B CS depende de las capacidades de ciertas regiones.
23	Asignación de grupo	No	Sí	No	Sí	Las estaciones de clase B CS deben recibir y evaluar si se cumple la regla de las 120 NM previo a contestar un mensaje 4.
24	Reporte de datos estáticos	Sí	Sí	Sí	Opcional	Existen parte A y parte B.
25	Mensaje binario de un solo slot	Sí	Sí	No	Opcional	
26	Mensaje binario de múltiples slots con estado de la comunicación	Sí	Sí	No	No	
27	Reporte de posición para aplicaciones de largo alcance	Sí	Sí	No	No	
28-63	No definidos	-	-	-	-	Reservados para uso futuro.

Capítulo 2. Sistema de Identificación Automática

es transmitido por una estación base cuando opera como entidad controladora. Este mensaje se aplicará a las estaciones de clase B CS que se encuentren dentro del rango de 120 millas náuticas y controla el modo de transmisión y recepción, el intervalo de reporte y los tiempos de silencio. Si a una estación se le solicita tiempo de silencio deberá suspender por el tiempo indicado sus reportes periódicos, contestando únicamente interrogaciones.

Desde el punto de vista de la transmisión, las embarcaciones de clase B CS deben poder transmitir los mensajes 18 y 24. El mensaje 18 es el mensaje utilizado por las estaciones de clase B para el reporte de posición y debe enviarse de forma periódica según la tabla 2.2 (excepto que se especifique algún otro intervalo con un mensaje 23, por ejemplo). El mensaje 24 tiene una parte A y una parte B. A grandes rasgos, la parte A de este mensaje asocia un MMSI con un nombre de embarcación. La parte B incluye también información de la identificación de la unidad provista por el vendedor, dimensiones de la embarcación, el tipo de embarcación, entre otros. Ambas partes del mensaje 24 deben ser enviadas cada 6 minutos por una estación AIS de clase B CS. Se debe enviar primero el mensaje 24A y un minuto después el 24B. Si a una estación AIS de clase B se le solicita el envío del mensaje 24, esta debe enviar ambas partes del mensaje.

En este capítulo se introdujo la tecnología AIS. Se describió su historia, que destaca la necesidad de este tipo de sistemas en las embarcaciones. Se indicaron sus generalidades y se diferenciaron los tipos de estaciones AIS existentes, haciendo énfasis en las estaciones de clase B. El funcionamiento de las estaciones se desarrolló siguiendo la norma ITU-R M.1371-5, organizando la estructura como el modelo de capas OSI. Se detallaron los mecanismos de acceso al medio disponibles y las diferencias de funcionamiento para ambos. Además, se introdujeron los procesos de armado, codificación y modulación de mensajes.

Capítulo 3

Mecanismo de acceso al medio

El aire como medio físico es compartido por todos los equipos que realicen transmisiones inalámbricas. En particular, en la tecnología AIS, las estaciones base, repetidores y embarcaciones compartirán un mismo rango de frecuencias. Se buscará un canal libre de colisiones, que podrá ser utilizado por un solo equipo a la vez para que los mensajes lleguen a destino correctamente. Para lograr esto, es necesario que todos los equipos accedan a él de forma organizada. En la tecnología AIS existen distintos mecanismos que permiten cumplir este objetivo.

En este capítulo se discutirá qué método de acceso al medio es de mayor conveniencia en el proyecto. En particular, se comparan los mecanismos SOTDMA (Self Organized TDMA, SO) y CSTDMA (Carrier Sense TDMA, CS). Para las comparaciones, se tienen en cuenta los efectos de la introducción de un nuevo equipo al canal sobre el resto de las embarcaciones y la congestión que se genera. Se argumentarán las decisiones tomadas en base a resultados obtenidos mediante simulaciones.

3.1. Comparación entre SOTDMA y CSTDMA

Para comprender las decisiones tomadas en cuanto al módulo de acceso al medio, es importante comprender las diferencias principales entre ambos mecanismos. Se recuerda que tanto SOTDMA como CSTDMA son utilizados en las estaciones de clase B.

Ambos se basan en técnicas de división temporal para acceder al medio. Es decir, se divide el tiempo en slots y cuadros, teniendo cada cuadro una duración fija de un minuto y conteniendo 2.250 slots. El inicio y final de cada cuadro de tiempo está sincronizado con el minuto UTC.

- **SOTDMA:** En términos generales, se basa en que cada estación AIS pueda reservar ciertos slots de tiempo en los que va a transmitir en el futuro. De esta manera, se logra organizar el medio compartido pues las estaciones solo podrán transmitir en los slots que tengan reservados.

Capítulo 3. Mecanismo de acceso al medio

- **CSTDMA:** El funcionamiento se basa en elegir 10 slots candidatos (slots que no se encuentren clasificados como no disponibles) y en cada uno de ellos se intentará transmitir. La embarcación sensa el canal durante cierto período de tiempo previo a la transmisión, y en función del nivel de ruido medido (y un umbral calculado), determina si el medio compartido está siendo utilizado. En caso de que el medio compartido esté disponible se transmite, en caso contrario se intentará con los otros candidatos.

A partir de un rápido análisis en ambos tipos de acceso al medio, se logra ver una desventaja para los equipos que utilizan CSTDMA. Estos equipos no transmiten en caso de detectar una señal mayor a cierto umbral, por lo que si el canal está ocupado, no se transmitirá en ese slot candidato. En otras palabras, los equipos que utilizan CSTDMA le dan prioridad a los de clase SOTDMA, o respetan las reservas que estos hicieron. Esto podría provocar que una embarcación que utiliza un equipo de clase CSTDMA no transmita nunca. Por ejemplo, si siempre se eligen slots candidatos ya reservados por estaciones SO.

En lo que respecta al funcionamiento, se puede deducir un grado de complejidad mayor en las estaciones de clase SOTDMA. En particular, estas estaciones deben poder recibir y recordar las reservas de todos los otros equipos, implicando gran coordinación y cooperación con otras estaciones. Además, deben tener la capacidad de enviar las reservas propias en los slots disponibles.

El mecanismo SOTDMA tiene intervalos de reporte menores que CSTDMA. Esto implica que las estaciones SO transmiten más mensajes en un cierto período de tiempo, generando mayor congestión en el canal. Además, requieren un mayor procesamiento y un mayor almacenamiento de la información.

Habiendo evaluado ventajas y desventajas de ambos mecanismos disponibles, se planteó la construcción de simulaciones simples para apoyar la decisión sobre cuál mecanismo a implementar. Se diseñó un simulador, dado que no se encontró ninguno con las características deseadas [37] [38]. En principio las ventajas de la simplicidad de CSTDMA parecen prometedoras. Además, se tiene que los dispositivos de esta clase requieren una menor potencia de operación, lo cual se traduce en un menor costo y se alinea con los objetivos del proyecto. Dado que todo apunta a utilizar CSTDMA como acceso al medio, se busca evaluar su desempeño en un canal con presencia de equipos que utilizan SOTDMA. Se pretende determinar qué tan significativas son las desventajas de los equipos CSTDMA y cuánto impacto tiene la introducción de un nuevo equipo al canal sobre el resto de las estaciones. Esto permitirá evaluar si es desventajoso el uso de estaciones CS en un canal con presencia de estaciones SOTDMA.

3.2. Simulador

Se procede a describir el funcionamiento del simulador en sí, teniendo en cuenta consideraciones sobre el intervalo de transmisión y algunas simplificaciones.

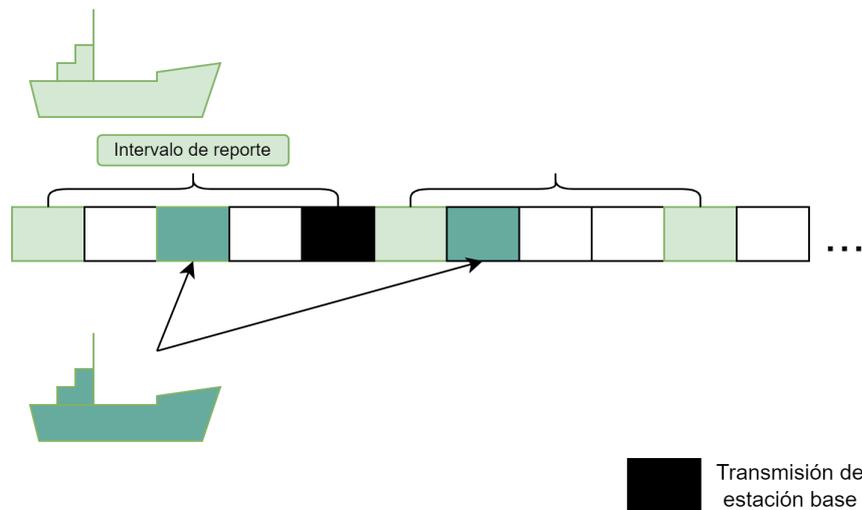


Figura 3.1: Representación de reservas de slots de tiempo por dos estaciones que utilizan el mecanismo SOTDMA para el acceso al medio compartido. Se indica también una reserva de una estación base que obstaculiza la reserva de una estación SO.

3.2.1. Consideraciones previas

Para la construcción del simulador que permitiera evaluar el desempeño de una estación de clase CS, se tomaron algunas decisiones.

Las estaciones de clase SOTDMA eligen los slots a transmitir en base a aquellos que estén libres o disponibles en ambos canales. Sin embargo, esto no fue modelado exactamente así. Lo que se desea evaluar es la capacidad para transmitir de las estaciones CSTDMA. Se tomó la decisión de que las estaciones de clase SO elijan de forma aleatoria el primer slot para transmitir y el siguiente slot es tanto tiempo después como lo indique su intervalo de reporte. Esto se ilustra en la figura 3.1. Se observa que la embarcación de color verde oscuro reserva sus slots cada cierto tiempo constante, indicado por su intervalo de reporte.

En caso de que el slot que se debe seleccionar ya esté reservado por otra estación SO o una estación base, se avanza de a un slot hasta encontrar uno libre en el frame actual. Esto se ilustra con la embarcación de color verde claro en la figura 3.1. Al intentar hacer su segunda reserva, se encuentra con que su slot óptimo de transmisión no está disponible, por lo que transmite en el siguiente. A su vez, se retrasa su próxima transmisión. Esto no se corresponde con el comportamiento real del SOTDMA. Si bien el slot inicial es aleatorio, no se mueve el slot de su siguiente transmisión de a uno hasta encontrar uno libre. Estos cambios se realizaron para simplificar la simulación sin afectar su objetivo.

Otro aspecto que no se tuvo en cuenta en la simulación fue la existencia de dos canales. Esto no tiene un efecto demasiado significativo a la hora de organizar la transmisión de estaciones CS. Si bien sí puede afectar la elección de slots por parte de embarcaciones SO, el efecto no será grande. Si se tuvieran dos canales, habría más slots disponibles que son más potenciales elecciones para la reserva por parte de los equipos SOTDMA. Esto a su vez se traduce en más slots disponibles a

Capítulo 3. Mecanismo de acceso al medio

partir de los cuales un equipo CSTDMA puede elegir, haciendo aún más difícil que el canal se sature por completo. En otras palabras, la congestión en dos canales no será peor que en un único canal.

Previo a ver qué tan limitadas están las transmisiones de clase CS, se debe agregar el comportamiento de las estaciones base. Esto es porque los slots que las estaciones CS no toman como posibles slots candidatos son aquellos que estén reservados por las estaciones base. Se agregan entonces reservas de dos estaciones base, número que pareció adecuado para un lugar que no tenga tanto tráfico como es el caso de Montevideo¹.

Como se quiere evaluar qué tanto puede transmitir una estación de clase CS, se tomó un canal lo más congestionado posible. Se optó por tomar los intervalos de transmisión más cortos tanto en embarcaciones CS como en embarcaciones SO. Es decir, todas las embarcaciones transmiten cada poco tiempo para que el canal esté congestionado. Por este motivo, las estaciones de clase CS tienen un intervalo de reporte de 30 s mientras que las estaciones de clase SO el intervalo es de 5 s. En el caso de las estaciones base, se tomó un intervalo de reservas de 10 s.

3.2.2. Esquema de simulación

El simulador toma como base un arreglo con $2,250 \times 5$ lugares, que representa los slots de tiempo en un intervalo de cinco minutos. Este arreglo estará inicializado con ceros. Para cada estación base y estación SO se fijan sus reservas colocando un valor 1 en el lugar del arreglo que corresponda. Cada estación CS, en un slot candidato para transmisión, mide el nivel de ruido del canal. En la simulación, esto se corresponderá con verificar si el valor de ese slot en el arreglo es o no igual a 1. Dependiendo de si existe una reserva, la estación CS transmitirá o no. Se describe a continuación más en detalle este funcionamiento.

En primer lugar, se define el arreglo de slots y se fijan las reservas de las estaciones base. Luego, con las consideraciones descritas en la sección anterior, se definen las reservas de las estaciones SO. A continuación se procede al intento de transmisión de las estaciones CS.

Como se describió en el capítulo 2, ninguna estación podrá utilizar un slot que reservó una estación base. Se eliminan entonces estos slots como posibles slots candidatos para los equipos CS. Se elegirán 10 slots candidatos que no hayan sido reservados por las estaciones base, pero que permitan cumplir con el intervalo de reporte correspondiente (ver tabla 2.2). Para el caso de la simulación, el slot donde estará centrado el intervalo en la primera transmisión se elige de forma aleatoria y se van eligiendo los posteriores según el intervalo de reporte. En caso de haber menos de 10 slots disponibles en el intervalo, se tomarán todos los slots que haya

¹Se compara al puerto de Montevideo con un puerto de gran tamaño como es el puerto de Hamburgo, en Alemania. Durante el año 2019, ingresaron en el puerto de Montevideo 543 buques de contenedores [39]. En el mismo año ingresaron al puerto de Hamburgo un total de 3.500 buques de contenedores [40]. Estas estadísticas permiten diferenciar la cantidad de tráfico de ambos puertos. Además, observando rankings internacionales, Montevideo no es un puerto destacado, a pesar de serlo en la región.

como candidatos.

Luego de obtenidos los 10 slots candidatos, las estaciones de clase CS sensarán el canal para determinar si pueden transmitir o no. Si una estación SO ya había reservado el slot que se está evaluando (el valor del arreglo es 1), la embarcación CS no transmitirá y se pasará a evaluar el siguiente candidato. Si la embarcación CS no puede transmitir en ninguno de los 10 candidatos elegidos, se considerará un mensaje fallido. Existe otro caso en el que la transmisión fallará. Esto se da cuando dos o más estaciones de clase CS sensan el canal en un mismo slot y transmiten en él. Así, los mensajes serán transmitidos pero se generará una colisión en el canal, y no podrán ser interpretados. Este caso se tuvo en cuenta a la hora de determinar la cantidad de transmisiones exitosas en cada slot.

3.2.3. Análisis de resultados

Se evaluó el desempeño de las estaciones de clase CS en distintos casos de estudio. Se describe cada uno de los casos y los resultados obtenidos.

Previo a la presentación de los resultados de simulaciones es necesario introducir el concepto de éxitos en la transmisión. La definición de transmisiones exitosas depende del tipo de estación considerado.

Por el lado de estaciones base o estaciones SO, basta con que reserven un slot para decir que su transmisión fue exitosa. Esto se debe a que otros dispositivos utilizando SO verán la reserva y no utilizarán ese slot. Además, los dispositivos CS cercanos sensarán el canal y detectarán a alguien transmitiendo, por lo que tampoco harán uso de ese slot. Se asume entonces que todas las transmisiones SO son exitosas.

Para las estaciones CS, se consideran transmisiones exitosas cuando se transmite en un slot no reservado y no existe otra embarcación CS que transmita en el mismo slot.

Primer caso: cantidad de estaciones SO que saturan al canal

En un primer caso de simulación se fijó un único equipo de clase CS, aumentando la cantidad de estaciones SO. Se buscó determinar a partir de cuántas estaciones SO se dificulta la transmisión del equipo CS. Los resultados de la simulación se ilustran en la figura 3.2.

En este primer caso de estudio, dado que se tiene una única estación de tipo CS, las transmisiones no exitosas dependen de las reservas de dispositivos SO. En la figura 3.2 la proporción de éxitos de la estación CS disminuye a medida que aumenta la cantidad de estaciones SO. Esto se debe a que las estaciones SO comienzan a reservar casi todos los slots del canal, y pasa a ser difícil para la estación CS hallar alguno en el que transmitir. El comportamiento de la gráfica es, a grandes rasgos, el esperado (decreciente).

Los resultados de simulación muestran que las pérdidas se observan a partir de unas 130 estaciones de clase SO, considerando 2 estaciones base. Dichas pérdidas son más significativas (cerca de un 30%) con alrededor de 160 estaciones SO.

Capítulo 3. Mecanismo de acceso al medio

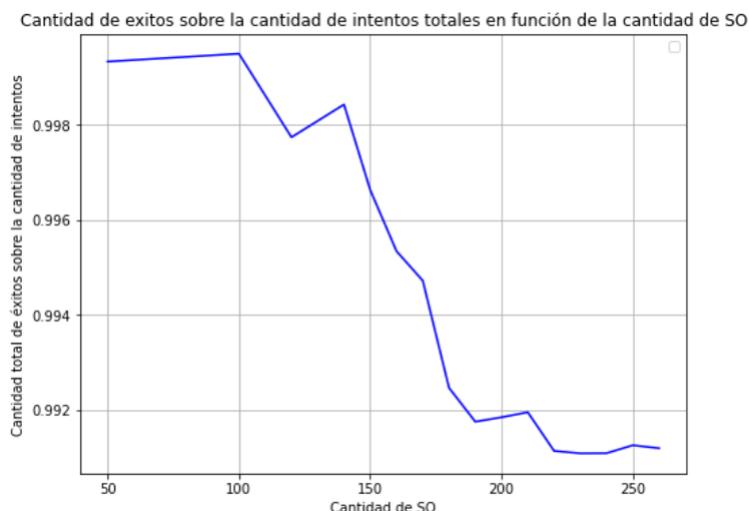


Figura 3.2: Gráfica de la cantidad de intentos sobre éxitos totales en función de la cantidad de embarcaciones SO con una única embarcación CS. Se observa un decrecimiento significativo cerca de las 150 estaciones, que indica que el canal comienza a saturar. El comportamiento no es monótono decreciente debido a componentes aleatorios en la simulación.

Así, se toma 160 como la cota superior de estaciones de clase SO que pueden existir en el canal para que una estación de clase CS funcione de forma correcta.

Se debe tener en cuenta también que cada estación SO transmite cada 5 segundos, resultando en 12 transmisiones en un minuto. En el caso de que todos los slots se ocupen en orden, se podría tener hasta $\frac{2 \cdot 250}{12} = 187,5$ estaciones. Si solo hubiera estaciones de clase SO el canal estaría saturado con 188. En un caso real, las pérdidas empezarán a ser significativas con menos embarcaciones. Como en la simulación realizada las estaciones SO hacen una búsqueda más ordenada, el número real de saturación completa es cercano a 188. Por esta razón, la cota obtenida de 160 estaciones es coherente en un caso real.

Es interesante notar que en la figura 3.2 se presentan algunos picos cerca de 140 estaciones SO que no parecen tener sentido, ya que el canal se va congestionando cada vez más. Se esperaría un comportamiento decreciente en todo momento. El hecho de que esto no ocurra se debe a que el simulador tiene componentes aleatorios, que pueden generar comportamientos inesperados en algunas ejecuciones.

Se presenta entonces en la figura 3.3 el promedio de la proporción de éxitos para una única estación CS en 20 realizaciones. De esta forma se disminuyen los efectos de los componentes aleatorios en la simulación. Se obtiene una gráfica más similar al comportamiento esperado, que decrece de forma monótona hasta que el canal se satura por completo (cerca de las 180 estaciones SO). Se observa en esta gráfica la caída significativa que comienza apenas luego de las 150 estaciones SO. Se recuerda que se había definido un valor máximo de 160 estaciones SO.

En el eje vertical de las figuras 3.2 y 3.3 se observa una proporción de éxitos sobre intentos cercana a uno en todo momento. Lo que se pretende mostrar es el comportamiento y la saturación del canal más allá de los valores en el eje de

3.2. Simulador

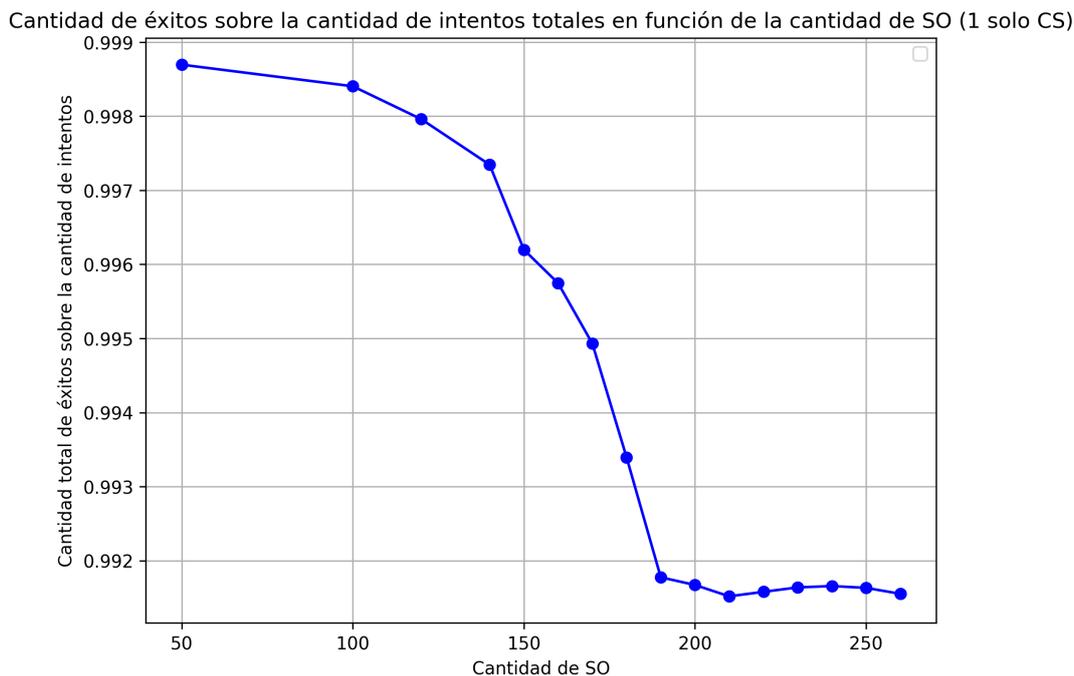


Figura 3.3: Gráfica del promedio de la cantidad de intentos sobre éxitos totales en función de la cantidad de embarcaciones SO con una única embarcación CS. Se consideran 20 realizaciones para el promedio. La simulación se realizó para valores más cercanos de cantidades de estaciones SO a medida que el total se acercaba a 200. Esto permitió un mejor estudio del momento de congestión del canal.

las ordenadas. A grandes rasgos, se ve con claridad que disminuye el desempeño general y como a partir de cierta cantidad de estaciones SO la proporción de éxitos se mantiene aproximadamente constante. Sin embargo, los valores podrían parecer extraños en una primera lectura. Se explica a continuación por qué tienen sentido, dada la métrica utilizada.

Si se tienen N estaciones de clase SO, las mismas aportan un total de $N \times 12 \times 5$ intentos de transmisiones y la misma cantidad de transmisiones exitosas². Por otro lado, una estación de clase CS aportará como máximo $2 \times 5 \times 10$ intentos de transmisión, ya que transmite 2 veces en cada uno de los 5 minutos y en cada transmisión puede tener 10 intentos. Los intentos de las estaciones de clase CS pueden ser todos fallidos pero serán como máximo 100, independientemente de la cantidad de estaciones de clase SO presentes. Al aumentar la cantidad de estaciones de clase SO, la cantidad de éxitos aumenta (mientras el canal no esté saturado) y en consecuencia la proporción de la cantidad de fallos será aún menor. Por ejemplo, en la figura 3.3 cuando se tienen 180 estaciones SO, habrá $180 \times 12 \times 5 = 10,800$ intentos de transmisión exitosos. Como mucho habrá 100 intentos por parte de las estaciones de clase CS, dando como total 10.900 intentos. Si se considera que los

²El factor de 5 en la multiplicación se debe a que se consideran cinco minutos para la simulación. El factor de 12 ya fue mencionado, y se debe a que transmitirán cada 5 s.

Capítulo 3. Mecanismo de acceso al medio

100 intentos de las estaciones CS fueron fallidos, la proporción de éxitos será, en el peor caso, de $\frac{10,800}{10,900} = 0,991$, valor muy similar al obtenido en la figura.

Se obtuvo entonces el dato de que el límite de cantidad de estaciones SO que permiten un funcionamiento adecuado de una estación CS es cercano a 160. Se realizó una investigación para hallar la cantidad de embarcaciones con equipos de clase SO que hay, en un día cualquiera y considerando promedios, en aguas uruguayas. Se realizaron intercambios con Prefectura Nacional Naval, así como una visita al Servicio de Iluminación y Balizamiento de la Armada. Según estimaciones de prefectura, se tienen entre 30 y 40 embarcaciones de gran porte en territorio nacional por día. Si se agregan embarcaciones extras que pueden no estar registradas como de gran porte (ya que embarcaciones más pequeñas pueden utilizar también un esquema SO), o embarcaciones dirigidas a puertos cercanos (como Buenos Aires), una posible cota es 100 embarcaciones diarias. Considerando este valor, las pérdidas de un equipo de clase CS serían menores al 10% de acuerdo al simulador desarrollado. Así, se puede asumir que utilizar el mecanismo CSTDMA para el acceso al medio compartido es factible para el presente proyecto.

Segundo caso: efectos de estaciones de cada clase sobre el canal

En un segundo caso de prueba se fijó una cantidad de estaciones SO, aumentando la cantidad de estaciones CS presentes en el canal. Se observó que, para una estación CS, el aumento en cantidad de estaciones de esa misma clase afecta muy poco su desempeño. Más allá de que se podría provocar una disminución en la cantidad de transmisiones exitosas, ya que aumenta la probabilidad de colisiones, el efecto no es notorio. Aquí se observa una ventaja de las estaciones CS por sobre las estaciones SO: dado un canal con una cantidad determinada de equipos de cualquier clase, el efecto del ingreso de una estación CS es menor que el de una estación SO.

Se procedió a probar con el simulador la intuición de que la introducción de una estación de clase SOTDMA genera más congestión en el canal establecido que una de clase CSTDMA. Para el caso de una única estación de clase CS, se graficó la cantidad de éxitos totales (de clases CS y SO) en función de la cantidad de equipos de clase SO. Esto se puede ver en la figura 3.4. Se observa que la cantidad de éxitos aumenta a medida que aumenta la cantidad de estaciones en el canal. Además, se muestra la congestión que se genera en el canal con las estaciones de clase SO, al llegar a 180 estaciones. En otras palabras, a partir de una cantidad elevada de estaciones de clase SO, aumentar este tipo de equipos no generará más transmisiones exitosas pues todos los slots estarán ocupados.

Una estación CS casi no cambia el desempeño conjunto de todos los equipos, pero una estación SO puede afectar de forma significativa la congestión del canal. Esto se debe principalmente a los intervalos de reporte pequeños definidos para las estaciones SO, que hacen que deba transmitir muchas veces por minuto. Al introducir un nuevo dispositivo al canal, si es de clase CS se adaptará a los slots disponibles. Por otro lado, si es de clase SO definirá parte de la agenda del canal, afectando a todos los equipos presentes.

Otra ventaja importante de las estaciones CS es que no mantienen las reservas

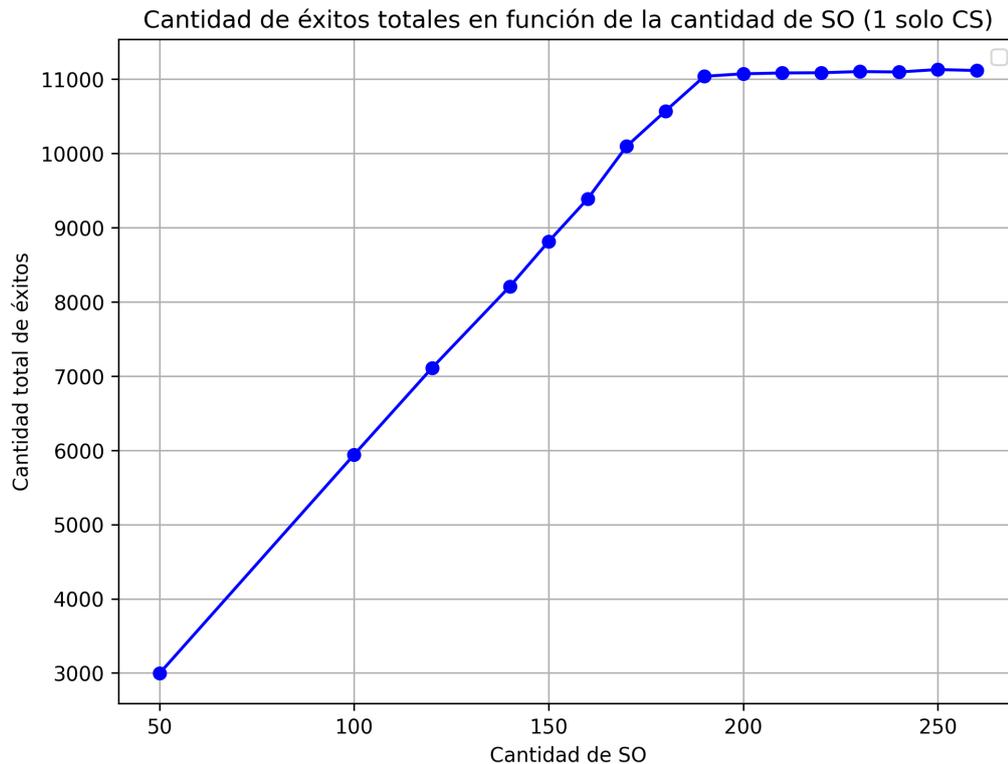


Figura 3.4: Gráfica de la cantidad de éxitos totales en función de la cantidad de embarcaciones SO con una única embarcación CS. La simulación se realizó para valores más cercanos de cantidades de estaciones SO a medida que el total se acercaba a 200. Esto permitió un mejor estudio del momento de congestión del canal.

de otras estaciones, lo que permite reutilizar slots reservados por equipos a grandes distancias. Esto se puede visualizar con el siguiente ejemplo. Sean dos estaciones SO que se mueven en direcciones opuestas, SO 1 y SO 2, y una estación CS que se mueve en la misma dirección que SO 1. Cuando las embarcaciones que se mueven en direcciones opuestas estén a una distancia suficientemente grande, la estación que utiliza CSTDMA no sentirá a la embarcación lejana en sus slots y podrá reutilizarlos. En cambio, la embarcación SO 1 tendrá registradas las reservas de SO 2, y evitará usar esos slots reservados al menos por algunos minutos más.

Se realizó una última prueba para comprobar que agregar más estaciones de clase SO en situaciones de mucha congestión empeora el problema. Se presenta en la figura 3.5 la cantidad de aciertos en función de la cantidad de intentos. Las distintas curvas fueron hechas para distinta proporción de estaciones de clase CS del total de embarcaciones. Para cada una de ellas la cantidad de equipos total aumenta de 100 a 400 y se mantiene la proporción que se indica en la curva.

Como es de esperar, cuantas más estaciones de clase SO hay, aumenta la cantidad de aciertos debido al aumento de la cantidad de equipos totales. Esto tiene que ver principalmente con el hecho de que el intervalo de reporte en las estaciones de clase SO es menor que el de las de clase CS. Sin embargo, se confirma que

Capítulo 3. Mecanismo de acceso al medio

Cantidad de aciertos en función de la cantidad de intentos para distintas proporciones de embarcaciones CS

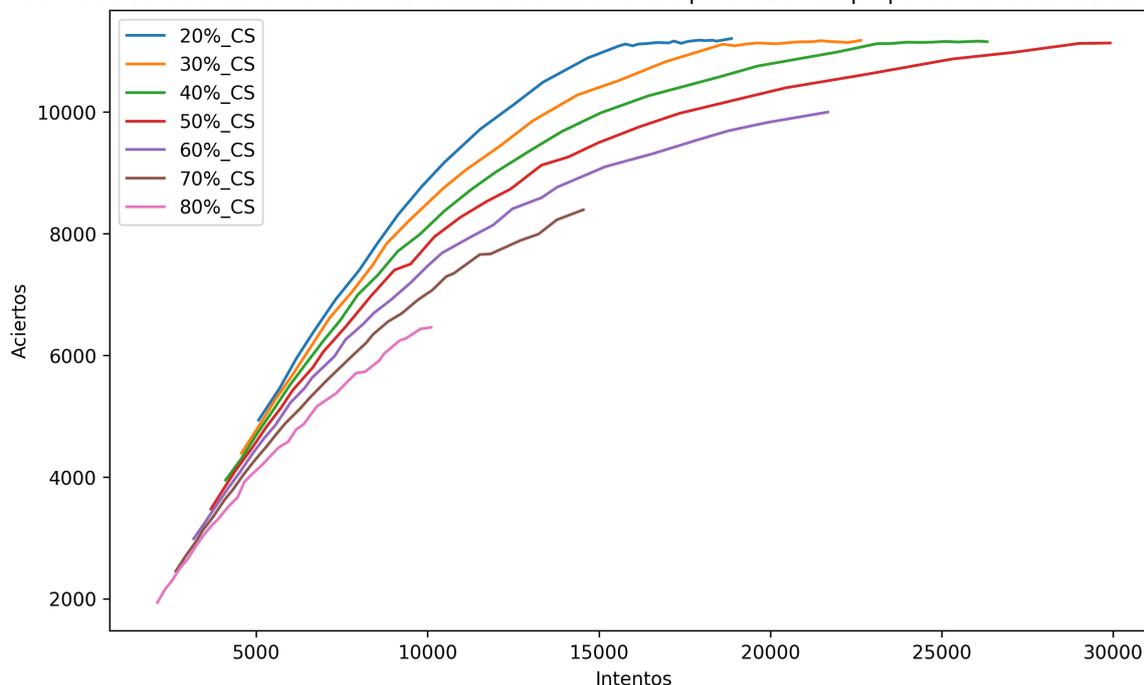


Figura 3.5: Gráfica de la cantidad de éxitos totales en función de la cantidad de intentos para distintas proporciones de embarcaciones de clase CS. Observar que cuanto menor es la proporción de estaciones CS (aumentando la cantidad de estaciones SO) el canal es más propenso a saturarse.

agregar estaciones de clase SO a un canal dado congestionará mucho más el canal que agregar de clase CS. Se puede ver que cuanto más estaciones de clase SO hay, la cantidad de aciertos tiende a estar limitada, alcanzando un máximo de aciertos. Por otro lado, con una misma cantidad de embarcaciones pero con un porcentaje más alto de equipos de clase CS, el canal estará mucho más liberado, permitiendo lugar a aún más estaciones para su uso .

En este capítulo se consideraron los mecanismos de acceso al medio disponibles: SOTDMA y CSTDMA. Se evaluaron ventajas y desventajas a nivel de funcionamiento, apoyadas en la norma que define la tecnología. Se realizaron suposiciones sobre el desempeño de las estaciones y se comprobaron a través de simulaciones desarrolladas. Se tuvo en cuenta que el caso de uso de esta investigación es en Uruguay. Las zonas más congestionadas del área marítima uruguaya no cuentan con una cantidad de estaciones SO que sobrepasen la cota establecida para la saturación del canal. Considerando los resultados y las conclusiones desarrolladas, se decidió implementar una estación que utilice CSTDMA.

A lo largo de este capítulo se obtuvieron diversos resultados, algunos de los cuales fueron presentados en la 30° Jornada de Jóvenes Investigadores, en Asunción, Paraguay, en octubre de 2023.

Capítulo 4

Implementación del sistema transmisor

En el capítulo 3 se fundamentó una decisión respecto al mecanismo de acceso al medio a utilizar en el proyecto. Se procede entonces a desarrollar el módulo de acceso al medio para el sistema transmisor. Se ahondará en su implementación, las características que debe tener y cómo se sortearon algunas dificultades. El objetivo del proyecto es desarrollar un sistema transmisor de clase B de la tecnología AIS considerando un mecanismo de acceso al medio CSTDMA.

4.1. Aspectos generales

El esquema del sistema transmisor implementado se presenta en la figura 4.1. Se observan los módulos de transmisión y recepción, así como los intercambios que se dan entre ellos. Observar que este diagrama presenta únicamente las componentes asociadas a la implementación del acceso al medio de la estación.

Una estación deberá contar con un proceso transmisor y dos procesos receptores (uno para cada canal de frecuencia). Dado que el mecanismo de acceso al medio a utilizar debe sensar el canal se tomó como primer paso la implementación de un receptor AIS. Al lograr adquirir mediciones del canal, se desarrolló un proceso transmisor que considera esas medidas para determinar si transmitir o no. El sistema transmisor fue implementado en un entorno de GNU Radio en su versión 3.8. Esto se debe a que uno de los proyectos de referencia, *gr-ais*, está implementado en dicha versión.

Las secciones de este capítulo se estructuran de la siguiente manera. Se describe primero el desarrollo asociado a la recepción, mencionando el proyecto que se tomó como base y haciendo énfasis en su influencia en el trabajo. A continuación se detalla el desarrollo de la transmisión, además de introducir el proyecto *gr-aistx*. Se presentan tanto la implementación de software realizada, como los dispositivos de hardware utilizados y su adecuación para el uso. En particular, se presentan los esquemas finales para el transmisor y el receptor. En el capítulo 5 se completa la descripción del sistema transmisor, describiendo la generación y codificación de mensajes. En el capítulo 6 se indicarán modificaciones realizadas sobre los diseños iniciales, con el fin de optimizar el sistema implementado.

Capítulo 4. Implementación del sistema transmisor

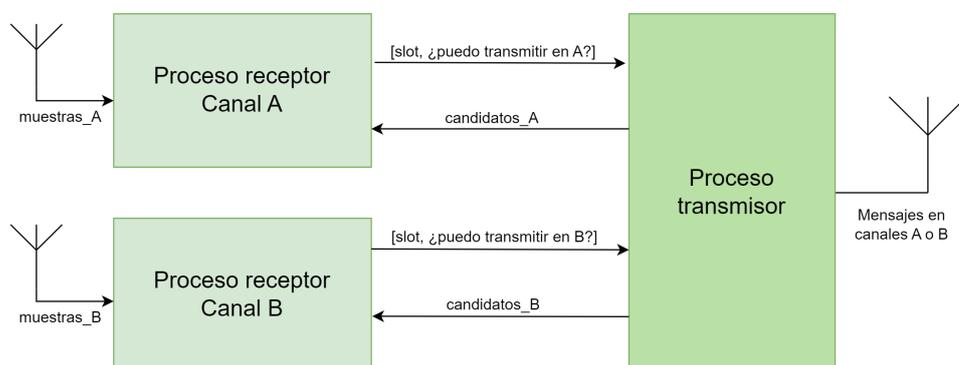


Figura 4.1: Esquema a alto nivel del sistema transmisor implementado, con los procesos de recepción y el proceso de transmisión. Se agrega la información intercambiada entre ambos para el correcto funcionamiento del sistema completo.

4.2. Implementación de la recepción

El receptor implementado permite medir el nivel de potencia del canal y mantener umbrales de ruido. El procesamiento se realiza de forma independiente en ambos canales de frecuencia. Los valores medidos se envían hacia el proceso transmisor para realizar el acceso al medio.

Para lograr el objetivo se hace uso del lenguaje de programación Python, implementando varios scripts en conjunto con la herramienta de software GNU Radio. Como base para el desarrollo se utilizó *gr-ais*, un proyecto de un receptor AIS de código libre disponible en GitHub [25].

4.2.1. Proyecto *gr-ais*

gr-ais es un módulo *out-of-tree* de GNU Radio disponible en GitHub [25]. La última versión del código está diseñada para su utilización con GNU Radio 3.8, y se clonó la rama *master* del repositorio. El proyecto permite la recepción de mensajes AIS, realizando su detección, demodulación y decodificación.

Su uso se basa en la terminal de comandos, sin tener interfaz gráfica de GNU Radio para la definición de diagramas de flujo. En particular, para ejecutar el receptor se debe correr un script llamado *ais_rx* con ciertos parámetros de funcionamiento. El siguiente comando permite la ejecución del receptor.

```
./gr-ais/apps/ais_rx
```

El archivo *ais_rx* se limita a tomar las opciones que se puedan agregar al comando y llamar a la clase *ais_radio* (definida en el script *radio.py*) con ellas. Se describen a continuación las opciones más relevantes.

- **-s** Indica el dispositivo receptor que se utilizará para tomar las muestras. Las opciones son *uhd*, *osmocom*, la ruta a un archivo o una dirección IP

4.2. Implementación de la recepción

seguida de un puerto. En función de qué dispositivo se utiliza se podrían tener opciones extra.

- **-g** Fija la ganancia de entrada del SDR.
- **-r** Fija la tasa de muestreo a utilizar por el SDR. El valor por defecto es de 250.000 muestras por segundo.
- **-S** Indica si se utilizará únicamente el canal de frecuencia A (161,975 MHz) o ambos canales (canal A, en 161,975 MHz, y canal B, en 162,025 MHz).

Una vez se ejecuta el comando con las opciones deseadas, *ais_rx* resulta en el llamado al script *radio.py*. Este fija los parámetros de recepción en el dispositivo a utilizar. Además, instancia uno o dos procesos receptores, en función de si se quiere recibir en uno o dos canales.

Un proceso receptor contiene los bloques de procesamiento necesarios para demodular y decodificar los mensajes y presentarlos en formato NMEA¹. En el script *radio.py*, el flujo de datos está definido con cuatro bloques que pueden observarse dentro del cuadro punteado en la figura 4.2. En primer lugar se aplica un filtro pasabajos de decimación, que disminuye la tasa de muestreo en un factor de 5. Luego se utiliza un bloque llamado *ais_demod*, definido en el proyecto y detallado más adelante. Finalmente se aplica un deframer para generar Unidades de Datos Empaquetados (PDU, por sus siglas en inglés), y un conversor PDU a NMEA para imprimir los mensajes en terminal.

El bloque *ais_demod* se define en el script de Python del mismo nombre. Contiene las inicializaciones y conexiones de bloques de GNU Radio para lograr la demodulación de los mensajes de AIS a PDUs. En primer lugar se realiza una sincronización en frecuencia. Se agrega un bloque de control automático de ganancia (AGC, por sus siglas en inglés). Este bloque mantiene fija en su salida la potencia de la señal de entrada, incluso si esta última presenta variaciones. Se realiza a continuación la detección del preámbulo o secuencia de entrenamiento agregada al inicio de los mensajes AIS. Se aplica un bloque para sincronización temporal previo a la demodulación GMSK. Una vez las muestras están demoduladas, se dividen para que sean procesadas de a un bit. Para la decodificación NRZI, en primer lugar se realiza una decodificación diferencial, y luego se invierte su resultado. De esta forma se obtienen los mensajes AIS en formato PDU.

4.2.2. Cálculos de parámetros de funcionamiento

En primer lugar se buscó obtener la frecuencia de muestreo con la que se trabaja dentro del esquema de recepción. El módulo *gr-ais* utiliza como frecuencia de muestreo 250.000 muestras por segundo, tasa a la que debe tomar muestras el dispositivo SDR de entrada. Luego, para el procesamiento de la señal, se aplica un filtro decimador con un factor de 5. Esto lleva a que la tasa de muestreo utilizada sea 50.000 muestras por segundo.

¹El formato NMEA se explica en el capítulo 5, al desarrollar el proceso de generación de mensajes.

Capítulo 4. Implementación del sistema transmisor

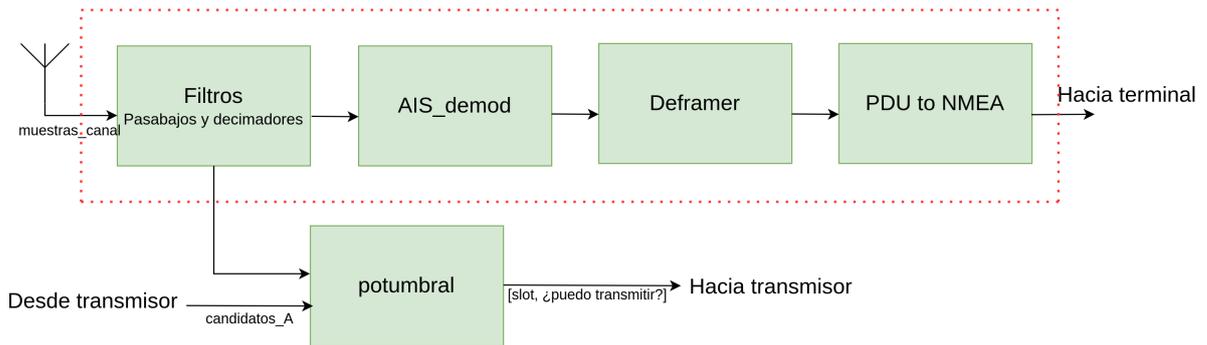


Figura 4.2: Diagrama de bloques del proceso receptor. Se observan en la línea superior los bloques preexistentes en *gr-ais*, y en la línea inferior los bloques nuevos.

El dispositivo ADALM-PLUTO utilizado no permite una tasa de 250.000 muestras por segundo. Por este motivo, se fija su tasa de muestreo en 750.000 muestras por segundo. Para poder utilizar y modificar de forma sencilla el receptor del proyecto *gr-ais*, se agrega un filtro decimador con factor de 3 para bajar la frecuencia de muestreo a 250.000 muestras por segundo. Una vez hecho esto, se prosigue con el funcionamiento convencional, bajando la tasa a 50.000 muestras por segundo.

Por definición de la norma, cada cuadro tiene una duración de 60 segundos, que se divide en 2.250 slots. En consecuencia, cada slot tendrá una duración de $\frac{60}{2.250} \approx 26.67$ ms. Al tener 50.000 muestras en un segundo, se tendrá que en la duración de un slot habrá $\frac{60}{2.250} \cdot 50.000 \approx 1.333$ muestras.

En la norma se especifica que la medición de los slots para determinar su potencia (y si está libre) se debe hacer en un tiempo específico. En particular, el sensado se debe hacer desde los 833 μs hasta los 1.979 μs desde el comienzo del slot, tal como se ilustra en la figura 2.8. Teniendo en cuenta que un segundo equivale a 50.000 muestras, se llegó a la conclusión de que se debe sensar desde la muestra 41 hasta la 98 en cada slot.

4.2.3. Modificaciones sobre el receptor de *gr-ais*

En una primera instancia, las modificaciones fueron realizadas en *radio.py*. Se cambió el esquema de conexión de GNU Radio. Se puede observar el diagrama de bloques final en la figura 4.2. Notar que se tiene una instancia del diagrama para cada canal. En esta sección se detallan los cambios efectuados, asociados al bloque de la línea inferior de la figura 4.2.

El gran cambio es la presencia de `Potumbral`. Este es un bloque agregado a *gr-ais* exactamente después del filtro. El filtro, así como otros bloques mencionados en este capítulo, es parte de los bloques que se encuentran por defecto en GNU Radio. Por otro lado, `Potumbral` es un bloque agregado por el equipo de trabajo, con la función de medir la potencia y el umbral actual en el canal. Su código fue realizado siguiendo las especificaciones de la norma. La salida de este bloque es una tupla. Para cada slot candidato recibido desde el proceso transmisor, se genera una salida de dos dimensiones. Esta contiene el número del slot actual y un 1 o un

4.2. Implementación de la recepción

0, según si es posible transmitir en ese slot o si está ocupado.

La presencia de `Potumbral` es imprescindible para lograr el correcto funcionamiento de un transmisor que basa su acceso al medio en CSTDMA. Para determinar si el canal está ocupado o libre, se debe conocer el umbral y la potencia del canal para saber cual es mayor y en base a eso transmitir (o no). El pseudocódigo de `Potumbral` se adjunta en el algoritmo 1 y se describirá su funcionamiento a continuación. Tal como se especifica en la norma, el valor del umbral no puede ser menor a -107 dBm, valor con el que es inicializado el umbral para tener valores conocidos².

La norma especifica que la estación no puede transmitir en el primer minuto desde su encendido. Esto se hace para que las mediciones del umbral sean fieles al canal donde se encuentra y lograr la sincronización con UTC. Luego de unos segundos, la potencia del canal pasa a tener el valor de la potencia medida. Dado que el transmisor no transmite en esos segundos, que los valores no sean verídicos no tiene efecto sobre el desempeño del sistema. Para el cálculo de la potencia del canal, se van procesando las muestras complejas de a diez a medida que ingresan al bloque. Para cada una de ellas se calcula su módulo y se toma como un valor de potencia representativo de las nueve muestras siguientes. De esta forma el procesamiento es más manejable y dicho valor de potencia es tomado como la potencia actual del canal. Como cada muestra equivale a 20 μ s, la potencia actual del canal se actualiza cada 200 μ s.

El cálculo del umbral se explica a continuación, ilustrando el proceso con el diagrama de la figura 4.3. En primer lugar se acumulan 100 muestras, que equivalen a 20 ms (pues cada una se toma cada 200 μ s). Cada uno de los intervalos de 20 ms es representado con el promedio de las 100 muestras que lo componen. Luego, se acumulan 200 representantes de 20 ms, que equivalen a 4 s. De igual forma que antes, cada intervalo de 4 s es representado por el promedio de las 200 muestras que lo componen. Por último, se acumulan 15 representantes de 4 s que equivalen a 60 s de muestras. Ahora, para obtener un valor de umbral, se toma el mínimo de esos 15 representantes y su valor en dBm será el umbral actual del canal. De esta forma, el umbral del canal se calcula teniendo en cuenta el último minuto. Luego de 4 s, el representante más viejo de los 15 será sustituido por el nuevo representante calculado y el mínimo de estos 15 será el umbral actual, que podría coincidir (o no) con el umbral anterior. Si dicho valor es menor al umbral mínimo permitido, el umbral se fija en el valor mínimo. Lo mismo sucede si el umbral es mayor al máximo permitido. Todos los cálculos se repiten para ambos canales. Para cada canal se mantienen sus propios valores de potencia y de umbral.

En el diagrama de bloques de la figura 4.2 se ven también las conexiones de `Potumbral` con el proceso transmisor. Estas conexiones se realizan con entradas y salidas de GNU Radio de tipo mensaje. Se reciben del proceso transmisor los slots candidatos para la próxima transmisión. En cada uno de esos slots, el bloque `Potumbral` evaluará la potencia y el umbral del canal entre las muestras 41 y 98.

²Se realizó una calibración del ADALM-PLUTO para que los valores de potencia medidos sean comparables con los valores máximos y mínimos. Este proceso se describirá más adelante.

Capítulo 4. Implementación del sistema transmisor

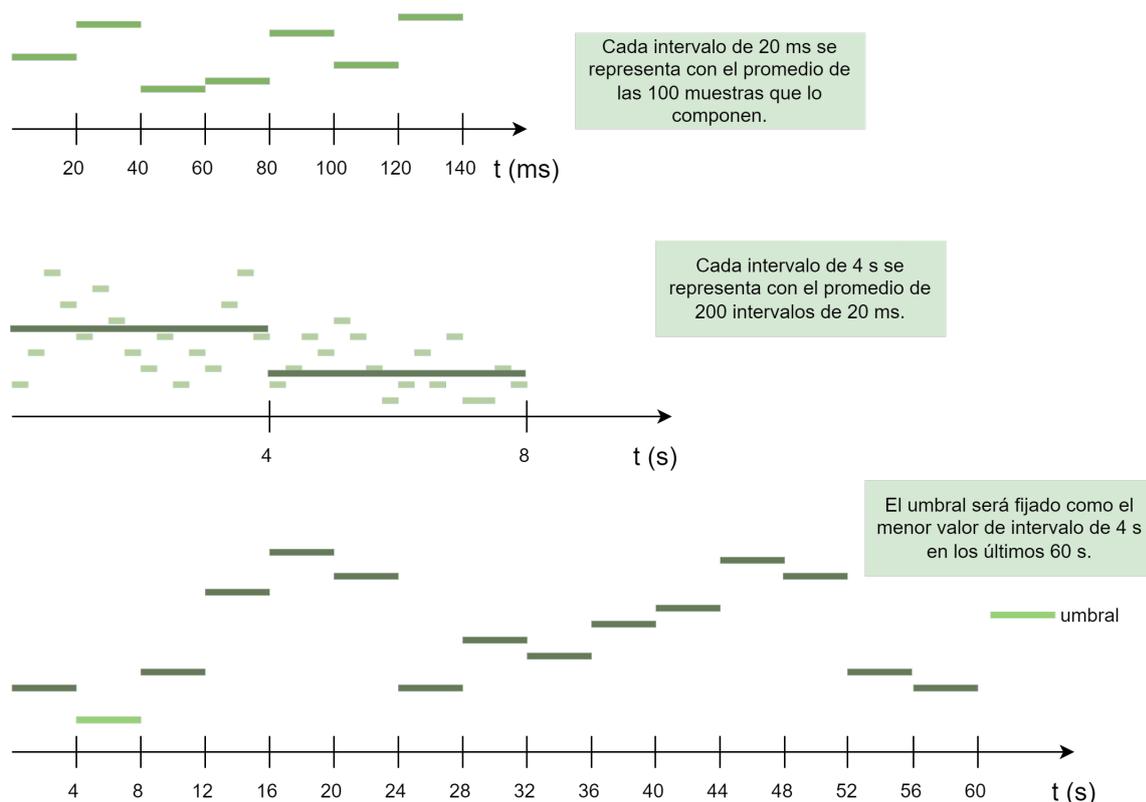


Figura 4.3: Ilustración del cálculo del umbral dinámico en función de los 60 s anteriores. Se describe el procedimiento desde intervalos de 20 ms, conformando intervalos de 4 s. Se ilustra cómo se elige el mínimo de los últimos 15 intervalos de 4 s como el umbral actual.

Se enviará por una salida del bloque hacia el proceso transmisor si en cierto slot se puede o no transmitir.

Para saber con precisión cuál de todas las muestras procesadas es la 41 y cuál es la 98, es necesario saber con exactitud cuándo comienza un minuto. Esta sincronización se hace a través de una señal recibida desde el módulo GPS, que recibe la referencia universal UTC. Una vez identificado el comienzo de un minuto UTC, es posible calcular el slot y la muestra actuales.

La sincronización con UTC se realiza utilizando la librería *datetime* de Python. En particular, la función *utcnow()* devuelve la referencia UTC de la computadora local. Para asegurar que esta referencia es confiable se implementó un servidor del NTP local que tiene como entrada la salida PPS del módulo GPS.

Además, dentro de *radio.py* se agregó otra funcionalidad que permitió adaptar *gr-ais* a las necesidades del presente proyecto. Dado que las recepciones y transmisiones se harán con un ADALM-PLUTO, se agregó la opción de “pluto” a la hora de invocar el receptor. En *radio.py* se hicieron las configuraciones necesarias para que el receptor pueda utilizar este equipo de hardware.

Como se ha descrito hasta ahora, la implementación del proceso de recepción tomó como referencia al proyecto *gr-ais*. Se utilizó la misma estructura para es-

Algoritmo 1 Pseudocódigo para el bloque Potumbral.

Potumbral() :

Considerar los slots candidatos para una cierta transmisión, recibidos desde el proceso transmisor.

Sea *puedoUsar* la variable en la que se guarda si un slot puede ser utilizado para la transmisión, luego de sensarlo.

Mientras queden muestras por considerar y *puedoUsar* != 1:

 Considerar una de cada 10 muestras de entrada.

 Calcular la potencia actual del intervalo de 10 muestras y guardar el valor.

Si se completó un intervalo equivalente a 20 ms de potencias guardadas **entonces**

 Tomar el promedio de esos 20 ms y guardar el valor.

Si se completó un intervalo equivalente a 4 s de promedios guardados *entonces*

 Tomar el mínimo de esos promedios y guardarlo junto con los 14 promedios anteriores.

 Fijar el umbral actual del canal con el mínimo de los 15 intervalos.

Si se están considerando muestras entre la 41 y la 99 de uno de los slots candidatos **entonces**

 Comparar la potencia actual medida con el umbral actual del canal.

Si se consideraron todas las muestras entre la 41 y la 99 **entonces**

Si la potencia actual es menor al umbral actual **entonces**

 Indicar que se puede transmitir en ese slot, fijando *puedoUsar* en 1.

Sino

 Indicar que no se puede transmitir en ese slot, fijando *puedoUsar* en 0.

 Esperar al siguiente slot candidato de esta transmisión.

Capítulo 4. Implementación del sistema transmisor

quemados de GNU Radio sin interfaz gráfica, así como los mismos parámetros de funcionamiento. Sin embargo, los bloques de *gr-ais* no fueron incorporados a la versión final del sistema transmisor implementado. Esto se explicará con mayor detalle en el capítulo 6. Al esquema del receptor presentado en la figura 4.2 le quedarán únicamente los filtros y el bloque `Potumbral`. La recepción de mensajes pasará a implementarse de forma opcional con un dispositivo SDR extra, en el que se ejecutará la versión original de *gr-ais*.

4.3. Herramientas de hardware utilizadas

El correcto funcionamiento del sistema transmisor depende de la utilización de ciertas piezas de hardware. Estas fueron de importancia para el desarrollo del módulo de recepción y para la planificación del módulo de transmisión. En esta sección se describen las piezas elegidas y la eventual configuración para incorporarlas al proyecto.

El desarrollo inicial se realizó sobre una placa Intel NUC [41] con un procesador i3, debido a que se contaba previamente con ella. Se utilizó un sistema operativo Ubuntu 20.04, con un total de 16 GB de RAM disponibles y 512 GB de disco. Dado que existen placas con precios más económicos, se buscó otra opción para el proyecto. El código fue integrado también a una placa Raspberry Pi 4 modelo B [42], a la que se instaló un sistema operativo Raspberry OS Bullseye para procesador ARM de 64 bits. La versión del sistema operativo instalada es del 22 de septiembre de 2022. Esta placa contó con una tarjeta SD de 32 GB y una memoria RAM de 4 GB. La utilización de una computadora Raspberry Pi estuvo motivada por su precio económico (que ronda los 200 dólares en el mercado local y se puede conseguir por la mitad de ese precio en Estados Unidos) y por la posibilidad de contar con pines de entrada/salida (pines I/O).

Para la recepción de la señal de GPS se utilizó un módulo UBlox NEO-7M [43] que cuenta con un conector para la antena receptora y cinco pines. Se tienen pines de TX y RX, que transmiten y envían información hacia los pines I/O de la Raspberry Pi. Se cuenta con un pin para alimentación VCC a 3.3 V y un pin de tierra GND. Finalmente, se tiene un pin llamado PPS. Este genera una señal de un pulso por segundo sincronizada a nivel de nanosegundos con la referencia temporal UTC. La señal de PPS fue utilizada para la configuración de un servidor del protocolo de tiempo de red (NTP, por sus siglas en inglés) que estuviera sincronizado con UTC. De esta forma, constantemente se corrige el reloj interno de la Raspberry Pi, que es usado desde el código implementado para determinar el slot actual. El detalle de la configuración del servidor NTP puede encontrarse en el apéndice C. Esta configuración no fue realizada en el NUC debido a que no cuenta con pines de entrada/salida³. Se conectó una antena activa GPS de la marca Bingfu [44] al módulo GPS. Esto permitió tener mejor recepción de satélites GNSS.

Resta introducir el dispositivo usado para la transmisión y recepción de men-

³Esto implica que la sincronización de la placa NUC no será tan precisa. En particular, estará sincronizado a UTC a través de Internet, con una precisión del nivel de milisegundos.

4.3. Herramientas de hardware utilizadas

sajes AIS. Se utilizó un ADALM-PLUTO Spec/Rev C02716D [45], un dispositivo SDR diseñado para la enseñanza de temas de comunicaciones inalámbricas analógicas y/o digitales. Cuenta con una entrada de alimentación de tipo micro USB y entradas para antenas receptora y transmisora. La hoja de datos del dispositivo indica que su funcionamiento es entre los 325 y los 3.800 MHz [45]. Este rango de frecuencia no permitiría su uso para la tecnología AIS. Sin embargo, existen varios tutoriales en Internet para modificar los límites de funcionamiento del equipo, indicando que funciona de manera correcta entre 70 y 6.000 MHz. Así, tomando como referencia a [46], se modificó el firmware del ADALM-PLUTO para poder utilizarlo en los canales de frecuencia definidos por la norma⁴.

Dado que se decidió utilizar un mecanismo de acceso al medio basado en el sensado del canal, fue de importancia calibrar la potencia recibida por el dispositivo SDR. Esto permite comparar valores de potencia medidos con valores reales de potencia, de forma de ajustar el sistema transmisor a las especificaciones de la norma. La calibración del ADALM-PLUTO se realizó utilizando un analizador de espectro como se describe en la siguiente subsección.

4.3.1. Calibración del dispositivo SDR

La calibración del dispositivo ADALM-PLUTO se realizó utilizando un analizador de espectro. A partir de este se determinará la potencia enviada por un transmisor y se comparará con la potencia medida por el receptor.

En primer lugar, vale la pena resaltar la necesidad de calibrar el dispositivo que se utilizará para la transmisión. Más allá de que la norma especifique cotas para la potencia a la hora de transmitir, la necesidad surge por el hecho de que el transmisor AIS necesita recibir para poder también enviar. De esta forma se podrá determinar si la potencia actual es inferior o superior al umbral del canal y tomar la decisión de transmitir o no. Sin embargo, el receptor a utilizar medirá la potencia del canal con un error que está asociado al instrumento en sí mismo. Es conveniente que la potencia utilizada para determinar el umbral esté corregida (sin el error del instrumento), especialmente por el hecho de que dicho umbral tiene un valor máximo y un valor mínimo determinado por la norma. Si bien el receptor estimará la potencia actual y el umbral con los mismos errores, los valores deberán estar normalizados para que todos los equipos tengan una misma noción de cuál es el umbral más grande y más pequeño posible.

El esquema utilizado para determinar el valor de potencia recibida se muestra en la figura 4.4. El receptor utilizado es el ADALM-PLUTO que se utilizará también para el desarrollo del transmisor. Como transmisor de la señal de referencia se utilizó un HackRF One conectado a una computadora. Dicho transmisor se conectó a un divisor de potencia o splitter a través de un atenuador de 30 dB. El splitter divide entre dos la potencia que recibe como entrada y la envía por sus dos salidas. Una de las salidas se conectó al analizador de espectro⁵ y otra al ADALM-PLUTO, ambas de forma cableada. Se presenta este esquema en la

⁴Se recuerda que el canal A está centrado en 161,975 MHz y el canal B en 162,025MHz.

⁵Se utilizó el analizador de espectro Siglent SSA3021X.

Capítulo 4. Implementación del sistema transmisor

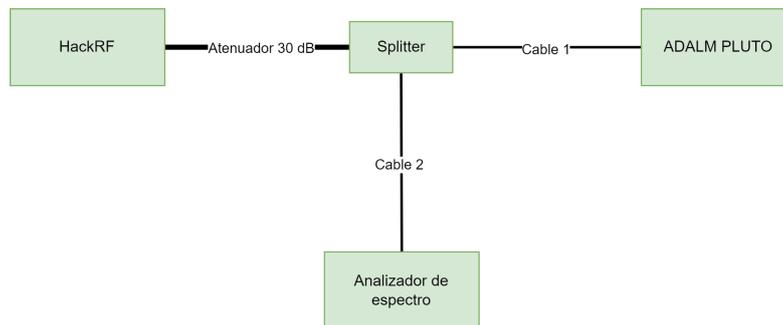


Figura 4.4: Diagrama con las conexiones para la normalización de la potencia. Se busca calibrar el dispositivo ADALM-PLUTO en función del analizador de espectro.

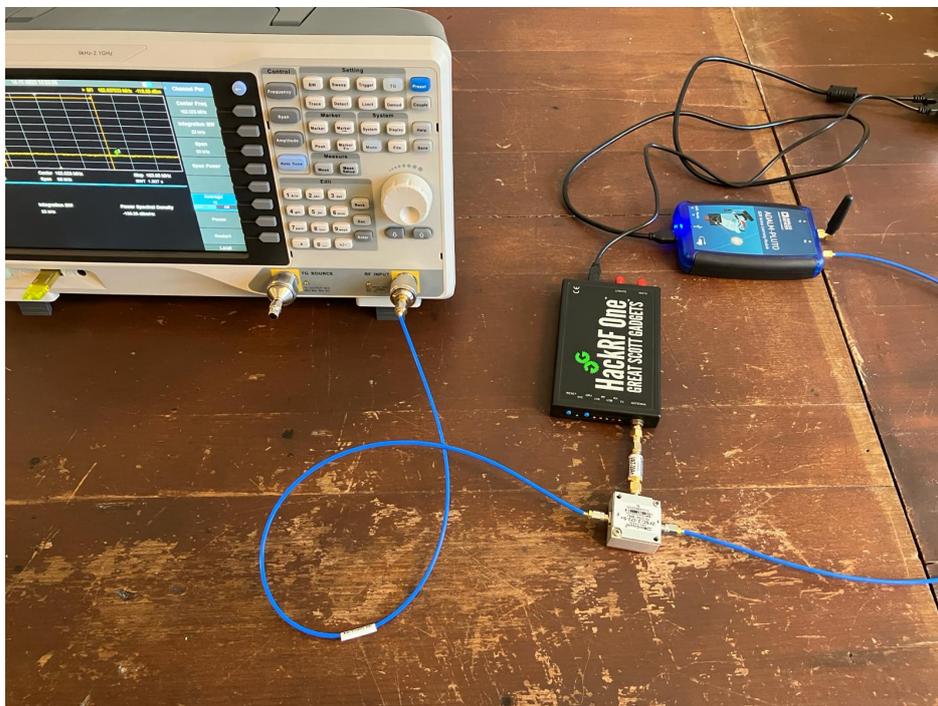


Figura 4.5: Sistema utilizado para la calibración del dispositivo ADALM-PLUTO. Se presenta la versión real del esquema de la figura 4.4.

figura 4.5, en este caso con una imagen real de los dispositivos utilizados con sus conexiones.

La idea del experimento fue generar una señal de un tono desde la computadora, utilizando un esquema de GNU Radio, y transmitirla a través del HackRF. La amplitud del tono se fue variando y se registraron los valores de potencia medidos en el analizador de espectro y en el dispositivo SDR. El objetivo fue determinar los factores multiplicativos que se debían agregar en el ADALM-PLUTO para que la potencia recibida fuera igual a la potencia recibida en el analizador de espectro.

Se tuvieron algunas consideraciones previas para asegurar que el procedimiento fuera correcto. En primer lugar, se asume que el analizador de espectro es un

4.3. Herramientas de hardware utilizadas

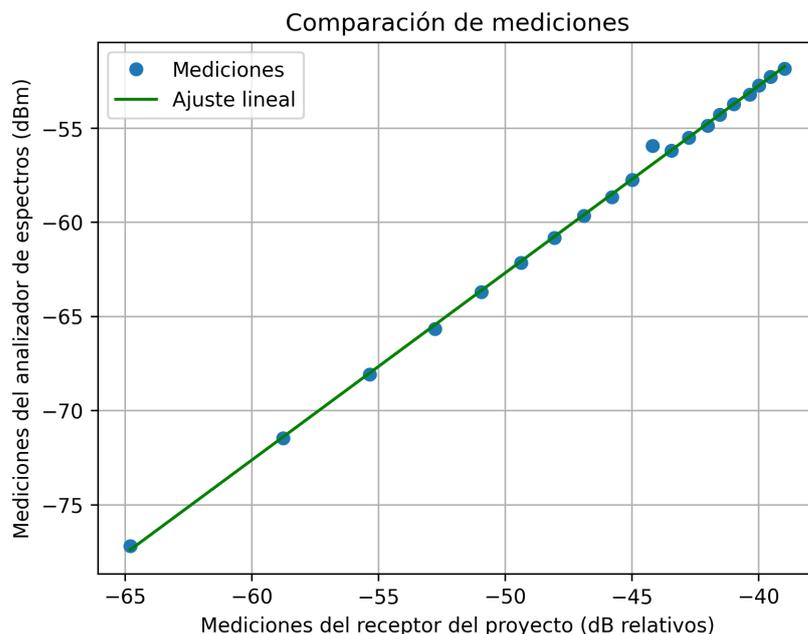


Figura 4.6: Ajuste lineal del transmisor ADALM-PLUTO.

dispositivo que mide potencia de forma adecuada. Se consideró que la atenuación provocada por el atenuador de 30 dB al ingreso del splitter afecta a ambos equipos de igual forma. La presencia del splitter en sí también afecta a ambos equipos de la misma manera ya que ambas salidas tienen una potencia equivalente a la mitad de la potencia de entrada. La única diferencia de los caminos desde el transmisor hasta cada receptor son los cables utilizados. Los cables, en principio, pueden tener atenuaciones distintas y esto afectaría a cada uno de forma diferente. Por esto, fue necesario analizar primero la atenuación de un cable con respecto al otro y después invertir los roles para ver los efectos provocados.

La medición de la atenuación de los cables implicó numerar ambos cables como “cable 1” y “cable 2”. Se obtuvo que el cable 1 tiene un corrimiento de +0,03 dB con respecto al cable 2 y que el cable 2 tiene un corrimiento de +0,01 dB con respecto al cable 1. Se concluyó entonces que a efectos prácticos los cables son iguales y no se considerará el efecto de ellos a la hora de hacer las medidas de la potencia.

En el apéndice D se presenta la elección de los parámetros de funcionamiento, así como el detalle del procedimiento seguido con mayor profundidad. Una vez todos los instrumentos estuvieron conectados de forma correcta y sus parámetros fijados adecuadamente, se tomaron varias medidas. El tono transmitido fue variando su amplitud desde 0,05 hasta 1, aumentando 0,05 entre cada medida. Esto resultó en un total de 20 mediciones, presentadas también en el apéndice D.

Se graficaron los valores del analizador de espectro en función de los valores del ADALM-PLUTO. Se presenta la gráfica en la figura 4.6, observando una relación lineal entre las mediciones del analizador de espectro y las mediciones del ADALM-

Capítulo 4. Implementación del sistema transmisor

PLUTO. Se realizó un ajuste lineal por el método de mínimos cuadrados. Los coeficientes obtenidos para el ajuste, considerando una recta de la forma $y = a.x + b$, son $a = 0,9941$, $b = -12,99$. La ecuación de esta recta mapea los valores de potencia que mide el receptor en los valores que obtendría el analizador de espectro. Los factores determinados por dicha ecuación fueron considerados en el script de recepción que determina los valores de potencia y umbral actuales en el canal.

4.4. Implementación de la transmisión

El transmisor implementado envía mensajes hacia otras estaciones considerando los intervalos de reporte de la estación y el estado de ocupación del canal. Se transmite alternando el canal de frecuencia y se intercambia información con el proceso receptor. En particular, el proceso transmisor envía los slots candidatos para cada transmisión. El proceso receptor envía, para cada slot candidato, si se puede transmitir en él o no.

Para lograr el funcionamiento deseado se hace uso del lenguaje de programación Python, implementando algunos scripts en conjunto con la herramienta de software GNU Radio. El desarrollo tomó como base a *gr-aistx*, un proyecto de un transmisor AIS de código libre disponible en GitHub. Este software se utilizó para comprender el proceso de generación y codificación de mensajes, contrastando la implementación con la norma de referencia. Además, se agregaron bloques de *gr-aistx* al transmisor diseñado.

4.4.1. Proyecto *gr-aistx*

El proyecto *gr-aistx* es un módulo *out-of-tree* de GNU Radio disponible en GitHub [24]. El proyecto permite la generación, codificación y transmisión de mensajes AIS. En el capítulo 5 se detalla el proceso de generación y codificación de mensajes, y se describe cómo resuelve *gr-aistx* estos procesos. Es importante destacar que este módulo no implementa el funcionamiento completo de un transmisor. Por ejemplo, no determina los mensajes a enviar, ni obtiene información de un receptor de GPS para armarlos. No tiene sincronización con ninguna referencia temporal, ni considera los slots definidos por la norma para las transmisiones.

gr-aistx define bloques de GNU Radio con interfaz gráfica. Esto permite su utilización desde la terminal de comandos o a través de diagramas de flujo. El proyecto cuenta con el archivo *AiS_TX.py* que define un diagrama de GNU Radio con el transmisor. Para que funcione es necesario indicarle un mensaje AIS como entrada a dicho archivo. El repositorio cuenta con un archivo para codificar mensajes AIS a partir de parámetros de entrada, llamado *AIVDM.Encoder.py*. Un ejemplo de utilización de ambos scripts para enviar un mensaje es el siguiente comando.

```
./AIVDM.Encoder.py --type=1 --mmsi=970010000 --lat=45.6910  
--long=9.7235 | xargs -IX ./AiS_TX.py --payload=X --channel=A
```

4.4. Implementación de la transmisión

El comando anterior concatena la salida del script generador de mensajes con el esquema de transmisión definido. Se agregan algunos parámetros de funcionamiento para ambos scripts. El archivo *AIVDM_Encoder.py* se describirá con mayor detalle en el capítulo 5. El archivo *AiS_TX.py* tiene varias opciones de entrada. Se describen a continuación las opciones más relevantes.

- - - **payload** Indica la carga útil del mensaje a transmitir.
- - - **channel** Indica el canal de frecuencia a utilizar (A o B).
- - - **sampling_rate** Fija la tasa de muestreo a utilizar.
- - - **power** Indica la potencia de transmisión (entre -12 dB y 12 dB).

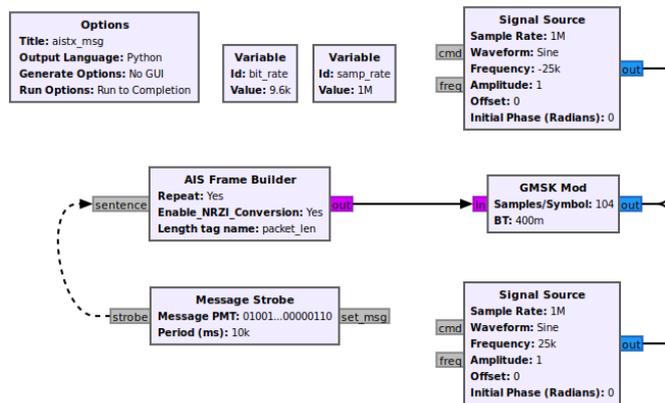
La versión original del código de *gr-aistx* está diseñada para su utilización con GNU Radio 3.7. Debido a esto se consideró una versión modificada para GNU Radio 3.8 [47]. En particular, se utiliza la rama *src-formatting* del repositorio. El funcionamiento es análogo. Se tiene un esquema de GNU Radio dentro del directorio *examples* que define el transmisor de mensajes y puede verse en la figura 4.7. El bloque que construye la trama, `Build AIS Frame` (que en la figura se llama `AIS Frame Builder`), recibe un flujo de bits de un bloque `Message Strobe` con un mensaje AIS por defecto. El script *AIVDM_Encoder.py* permite obtener el flujo de bits asociado a un mensaje, como el utilizado en `Message Strobe`. Luego de que la trama está preparada para transmitirse, el bloque mencionado la codifica en NRZI. Un siguiente bloque se encarga de la modulación GMSK y el último bloque se utiliza para la transmisión con un dispositivo SDR.

4.4.2. Transmisor

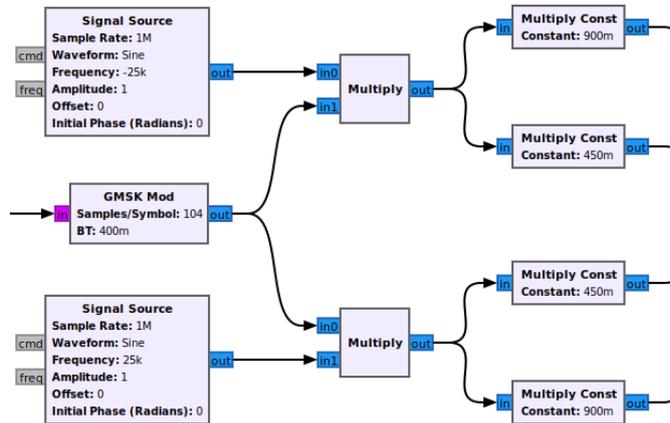
En la figura 4.8 se presenta parte del diagrama de bloques del transmisor diseñado. Se tienen tres bloques desarrollados y se indican las interconexiones con los procesos receptores. Los bloques asociados a la modulación y transmisión en la banda de frecuencias correspondiente se detallan más adelante.

En las siguientes secciones se describirá en detalle el funcionamiento de cada bloque implementado. En conjunto permiten obtener los resultados deseados. Se tiene un bloque central, `Transmitter`, que implementa la lógica principal. Este bloque avisa al proceso receptor cuáles son los slots candidatos para la siguiente transmisión. En función de lo que recibe desde el proceso receptor, determina si puede transmitir o no en un cierto slot. Si la respuesta es afirmativa, entonces invoca al bloque que genera los mensajes, `Messages`. Este bloque arma los mensajes con información recibida desde el módulo GPS conectado al sistema, a través de la salida del bloque `Sub_gps`. Finalmente, los bits del mensaje son enviados al bloque `Build AIS Frame` para terminar de conformar la trama a transmitir y codificarla. La señal se modula en GMSK y se transmite en el canal de frecuencia correspondiente. Las interconexiones asociadas a la codificación y modulación se presentarán junto con el esquema de transmisión completo.

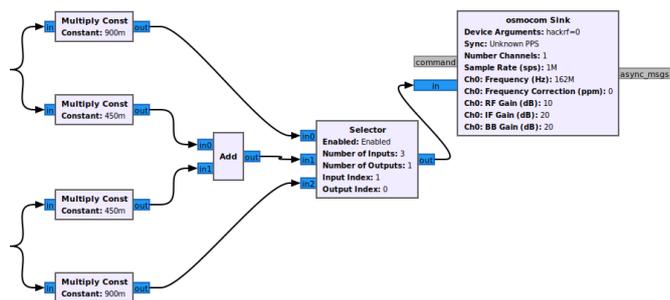
Capítulo 4. Implementación del sistema transmisor



(a) Primera parte del diagrama ejemplos.



(b) Segunda parte del diagrama ejemplos.



(c) Tercera parte del diagrama ejemplos.

Figura 4.7: Esquema de *examples* que se encuentra en el repositorio de *gr-aistx*. Se cuenta con los bloques necesarios para generar la trama a partir de los bits del campo de datos, codificar en NRZI, modular en GMSK y luego enviar la señal al canal correspondiente.

4.4. Implementación de la transmisión

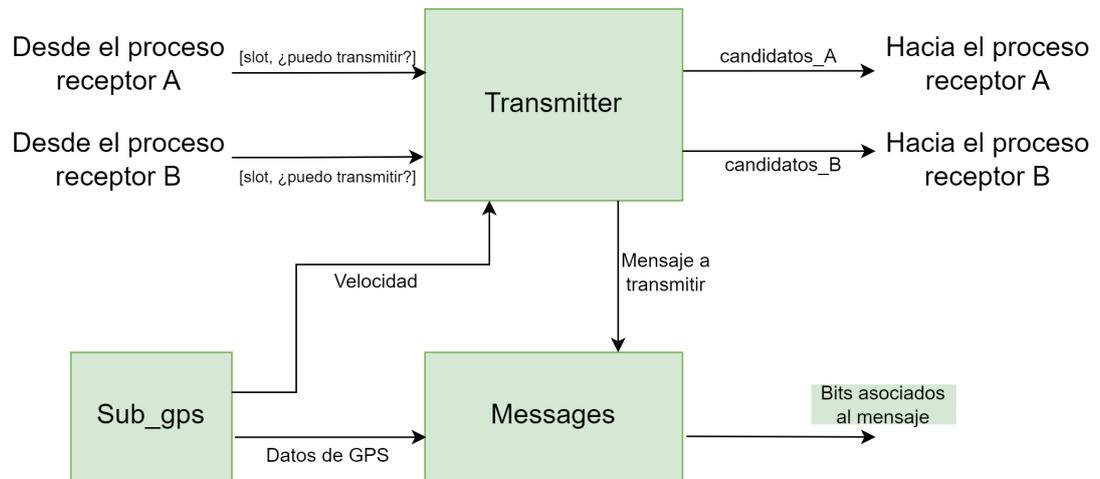


Figura 4.8: Diagrama de bloques del transmisor implementado, hasta obtener los mensajes a transmitir como un string de bits. Se destacan los intercambios de información entre los bloques.

Bloque Transmitter

El bloque `Transmitter` se encarga de la lógica principal del proceso transmisor. Se presenta en el algoritmo 2 su pseudocódigo y se describe su funcionamiento.

Al encender el transmisor, se debe esperar un minuto entero antes de comenzar a transmitir mensajes. El proceso receptor requiere de ese minuto para obtener valores correctos de los umbrales de funcionamiento. Durante los primeros sesenta segundos, el transmisor está suspendido, mientras que el receptor funciona de manera normal. Al terminar el tiempo de inicialización, se comienzan los temporizadores para los mensajes a transmitir. Un tiempo antes de que expire el temporizador de cierto mensaje se envían al proceso receptor los slots candidatos para ese mensaje. A medida que llegan dichos slots, el receptor realiza el sensado del canal en cada uno de ellos y le indica al transmisor si un slot puede o no ser utilizado. En caso de que un slot no esté disponible, se espera al siguiente slot candidato. En caso de que sí esté disponible, se envía un número al bloque `Transmitter` indicando qué mensaje debe transmitirse.

En el caso particular de estudio, se implementó la transmisión de dos mensajes: el mensaje 18 para transmisión de información dinámica de la embarcación, y el mensaje 24 para transmisión de información estática de la embarcación.

Bloque Sub_gps

La obtención de información de posicionamiento y velocidad se realiza a través del bloque `Sub_gps`. Por fuera del entorno de GNU Radio existirá un script que se comunica con el módulo GPS⁶. Dicho script recibe información haciendo uso del

⁶En el repositorio del proyecto se tendrán varios scripts. Esto se debe a que el código depende de la placa utilizada con el módulo GPS. Se incluirá un script de señal de GPS falsa, para utilizar si no se cuenta con un módulo GPS.

Algoritmo 2 Pseudocódigo para el bloque Transmitter.

Transmitter(*velocidad*) :

Esperar 60 s para que el proceso receptor pueda obtener un valor correcto del umbral de ruido del canal.

Inicializar un temporizador para el mensaje 18 y otro para el mensaje 24.

Si faltan 400 slots para que expire el temporizador del mensaje X

entonces

Indicar que se está intentando de transmitir el mensaje X

Determinar los slots candidatos para la transmisión

Enviar al receptor los slots candidatos.

Si pasan 200 slots desde el slot central de la próxima transmisión y aún no transmitió **entonces**

Reiniciar el temporizador según el mensaje fallido, desde el slot actual.

Si llega uno de los slots candidatos para el mensaje X **entonces**

Recibir si se puede o no transmitir en dicho slot.

Si se puede transmitir en dicho slot **entonces**

Enviar a *Messages* el mensaje a transmitir.

Reiniciar el temporizador según el mensaje transmitido y la *velocidad*.

Si no

Esperar al próximo slot candidato.

protocolo uART y genera un diccionario con los datos relevantes para el armado de los mensajes. El diccionario es enviado a un socket TCP y será recibido por el bloque de GNU Radio. Se presenta en el algoritmo 3 el pseudocódigo del bloque.

El bloque establece una conexión al socket TCP previamente mencionado y recibe el diccionario armado. Se organiza esta información en arreglos para un procesamiento más sencillo y se envía por las salidas del bloque.

Bloque Messages

Para la generación de los mensajes que se desean enviar, se tomó como base la generación de mensajes implementada en gr-aistx. En dicho repositorio se encontraba la codificación para varios mensajes, desde donde fueron tomadas las funciones que se encargaban de los mensajes 18 y 24.

Algoritmo 3 Pseudocódigo para el bloque Sub_gps.

Sub_gps() :

Inicializar un socket ZMQ al puerto TCP para recibir información del módulo GPS.

Con la información recibida en formato de diccionario, armar un arreglo de la forma: [*velocidad, longitud, latitud, curso, segundo UTC*].

Enviar por una salida el valor de velocidad (hacia el bloque *Transmitter*).

Enviar por otra salida el arreglo (hacia el bloque *Messages*).

El bloque *Messages* se conecta a *Sub_gps* recibiendo como entradas la latitud, longitud, velocidad, curso y segundo UTC. *Messages* se conecta también con *Transmitter* desde donde recibe qué mensaje es el que corresponde transmitirse (si existe alguno). A partir de las entradas desde el GPS, cuando se indica que se debe transmitir el mensaje 18, 24 A o 24 B, *Messages* llama a la función generadora de mensajes que corresponda con los parámetros que esta requiera. El pseudocódigo de *Messages* se presenta en el algoritmo 4.

Algoritmo 4 Pseudocódigo para el bloque Messages.

Messages(*mensaje, velocidad, longitud, latitud, curso, tiempoSec*) :

Si *mensaje* = 18 **entonces**

 Generar el mensaje 18 con los datos de entrada.

 Enviar el mensaje 18 por la salida.

Si *mensaje* = 240 **entonces**

 Generar el mensaje 24 A con los parámetros de la embarcación.

 Enviar el mensaje 24 parte A por la salida.

Si *mensaje* = 241 **entonces**

 Generar el mensaje 24 B con los parámetros de la embarcación.

 Enviar el mensaje 24 parte B por la salida, considerando el mismo canal que fue utilizado para la transmisión de la parte A del mensaje.

La salida del bloque tendrá la carga útil del mensaje en un flujo de bits. Este flujo se utiliza como entrada para el bloque *Build AIS Frame*. Se explicará con más detalle la comunicación entre los bloques en el capítulo 5.

Capítulo 4. Implementación del sistema transmisor

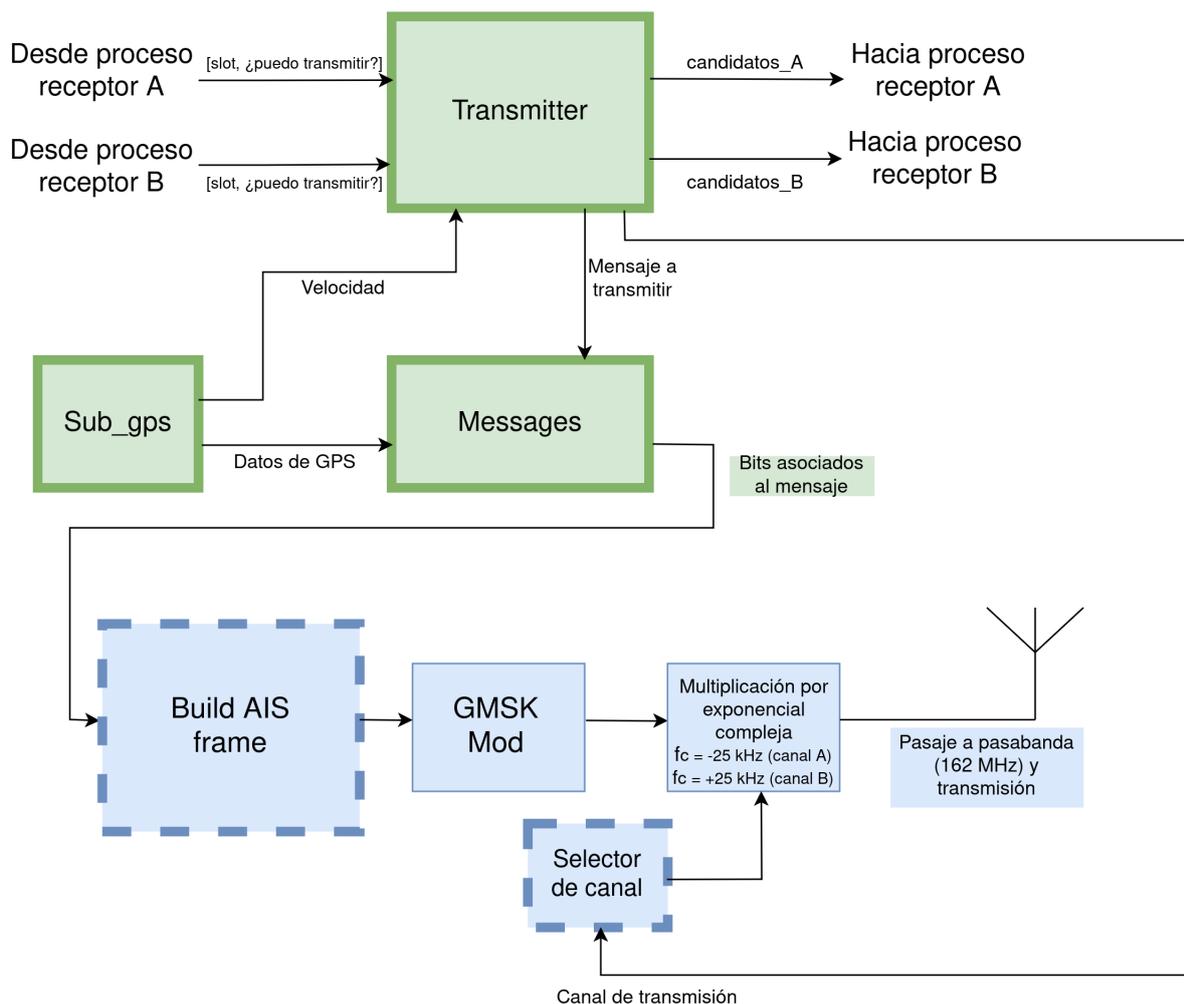


Figura 4.9: Diagrama de bloques completo del transmisor. Se indican en color verde los bloques centrales de este proyecto. Se presentan con azul los bloques que realizan el procesamiento de los bits hasta que se tiene una señal para enviar al canal. Los bloques con línea gruesa continua fueron creados en el marco de este trabajo. Los bloques con línea gruesa punteada fueron modificados por el grupo. El resto de los bloques no presenta cambios.

Esquema de transmisión

En la figura 4.9 se muestra el esquema del transmisor implementado. Se señalan en color verde los bloques que representan el diagrama de bloques de la figura 4.8. En azul se muestran los bloques que realizan el resto del procesamiento previo a la transmisión. Los bloques con línea gruesa continua fueron creados desde cero en el marco de este proyecto. Los bloques con línea gruesa punteada fueron modificados en el marco de este proyecto. Los bloques con línea sencilla no fueron cambiados. Se describe a continuación lo que hace cada uno de los bloques, aunque el detalle de la generación y codificación de mensajes se aclara en el próximo capítulo.

En primer lugar se tiene un bloque `Build AIS Frame`. Este bloque conforma

4.4. Implementación de la transmisión

la trama AIS, recibe los bits de la carga útil en su entrada y calcula el código de redundancia cíclica (CRC). En este bloque se termina de conformar la trama AIS en bits, agregando el CRC calculado y las banderas de inicio y fin del paquete. También se hace el bit stuffing, previniendo largas secuencias de bits seguidos en 1. Se agrega el preámbulo al paquete (24 bits en 0 y 1 de forma alternada) y el buffer, como se menciona en el capítulo 2. Este bloque también realiza la codificación NRZI. Tal como se ve en el ejemplo de la figura 2.2, los bits son codificados de acuerdo a la lógica de que con cada 0 recibido se cambia el nivel del pulso. Dado que el sistema está pensado para funcionar en estaciones de clase B CS, el preámbulo de los mensajes debe comenzar con un 0. Esto tuvo que ser modificado del bloque `Build AIS Frame` tomado como referencia.

El siguiente bloque realiza la modulación GMSK. Se tiene la conformación con un pulso gaussiano y luego una modulación en frecuencia (FM).

Por último, la señal es pasada a pasabanda y transmitida. Para alternar la transmisión entre ambos canales, como especifica la norma que se debe hacer, se creó un bloque `Selector`. Este bloque recibe desde el bloque `Transmitter` un mensaje indicando el canal a utilizar, y se limita a seleccionar cuáles muestras debe enviar hacia el SDR.

En este capítulo se detallaron las implementaciones de los procesos de transmisión y recepción, relacionándolos con los repositorios preexistentes que se usaron como base. Habiendo desarrollado el acceso al medio compartido, el siguiente capítulo se enfoca en la generación y codificación de los mensajes AIS a transmitir.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Generación y codificación de mensajes

El objetivo de cualquier tecnología de comunicaciones es el intercambio de información. En el caso de las embarcaciones es de relevancia compartir características de movimiento o de identificación. Los mensajes de la tecnología AIS deben seguir cierto formato para que sea posible interpretarlos. En particular, la norma ITU-R M.1371-5 especifica una estructura para los mensajes y mecanismos para su codificación.

En este capítulo se retoma el estudio de los mensajes disponibles en sistemas AIS realizado en el capítulo 2. Se describe la estructura general de un mensaje y el contenido de sus campos. Dado que se definió la implementación de una estación de clase B CS, se especifican los mensajes que transmitirá dicha estación. Además, se indica el proceso que sigue un mensaje una vez ha sido conformado hasta que es transmitido por la antena al canal de radiofrecuencia.

5.1. Estructura general de las tramas

Las tramas AIS tienen la estructura que se ilustra en la figura 5.1, que ya fue presentada en el capítulo 2. Esta estructura es idéntica a la estructura utilizada por el protocolo de capa de enlace HDLC [35], a excepción de la secuencia de entrenamiento. La trama AIS tendrá un largo por defecto de 256 bits, que es la máxima cantidad de bits que se pueden enviar en un slot de tiempo ¹.

En primer lugar se tiene la secuencia de entrenamiento, que también se denomina preámbulo y sirve para sincronización. Se trata de 24 bits alternando entre 0 y 1, comenzando con un bit en 0 para las estaciones de tipo CS. Se agrega una bandera de inicio de 8 bits, con el valor por defecto de 01111110. A pesar de tener varios bits en 1 seguidos, no se aplicará bit stuffing a esta bandera. El campo de datos tiene un largo de 168 bits, y se describe en la sección 5.2 cómo se completa en función del mensaje AIS a transmitir. Luego del campo de datos se tiene un campo de 16 bits de largo en el que se agrega un código de redundancia cíclica (CRC, por sus siglas en inglés). Se calcula una suma de comprobación sobre el campo de

¹Observar que se tienen 37,5 slots en cada segundo, y al dividir la tasa de bits (de 9.600 bits/s) entre 37,5 se obtiene que cada slot tendrá 256 bits como máximo.

Capítulo 5. Generación y codificación de mensajes

Secuencia de entrenamiento	Bandera de inicio	Campo de datos	CRC	Bandera de finalización	Buffer
----------------------------	-------------------	----------------	-----	-------------------------	--------

Figura 5.1: Trama de un mensaje AIS. Se observan los seis campos principales, resultando en un total de 256 bits de largo. Esta figura fue adaptada de la figura 6 de la norma ITU-R M.1371-5 [16].

datos únicamente, siguiendo el mecanismo definido en ISO/IEC 13239:2002. Para finalizar, se agrega una bandera de finalización de 8 bits idéntica a la bandera de inicio.

Se tiene un buffer de 24 bits de largo al final de la trama. Los primeros 4 bits son reservados para bit stuffing. Se reservan también 14 bits para retardos de distancias entre transmisor y receptor. Finalmente, los últimos 6 bits son reservados para variación de retardos en la sincronización.

5.2. Mensajes considerados para el proyecto

En las tablas 2.3 y 2.4 se presentan los mensajes existentes en sistemas AIS. Además de indicar el nombre de cada uno y su número identificador, se detalla cuáles deben ser transmitidos o recibidos por estaciones de clase B SO y CS. Como se argumentó en el capítulo 3, se definió implementar un sistema transmisor de clase B CS. Así, se deben transmitir de forma obligatoria los mensajes 18 y 24, y de forma opcional los mensajes 13 y 14.

Los mensajes considerados para la transmisión en el proyecto son el mensaje 18, de reporte de información dinámica, y el mensaje 24, de reporte de información estática.

5.2.1. Mensaje 18

El mensaje 18 corresponde al reporte de información dinámica por parte de estaciones de clase B. Este mensaje deberá ser enviado de forma automática y periódica, dependiendo de la velocidad de la embarcación. Más precisamente, si la velocidad de la embarcación es superior a los 2 nudos, el mensaje se debe enviar cada 30 segundos. De lo contrario, el mensaje se debe enviar cada 3 minutos, en ambos casos de forma automática (sin intervención por parte del usuario).

Los datos dinámicos que son enviados por este mensaje son la latitud, longitud, dirección en tierra, velocidad en tierra y el segundo UTC en el que se generó el mensaje. La cantidad de bits necesarios para cada dato se indica en las tablas 5.1 y 5.2. Como se ve en las mencionadas tablas, también se envían otros datos sobre la embarcación que son relevantes para las otras embarcaciones pero a diferencia de los anteriores, no son dinámicos. Por el contrario, valores como el identificador de usuario y las banderas de clase B están fijos en un determinado valor en todas las transmisiones de una misma embarcación. En el caso del transmisor implementado, el valor de MMSI (identificador de usuario) que se tomó para las pruebas fue de

5.3. Generación de mensajes

0123456789 debido a que no se contaba con un identificador real por no poseer una embarcación física. Luego, en los campos de banderas de la tabla 5.2 (los campos de un solo bit), se utilizaron valores de 0 en todos excepto en la bandera de clase B de unidad y en la bandera de selección de estado de comunicación. En la primera bandera, se fijó el valor en 1 ya que se implementó una estación de clase B CS. En la última bandera se fija el valor en 1 ya que el estado de la comunicación no se corresponde con SOTDMA. Las otras banderas se fijaron en 0 ya que corresponde a mecanismos o utilidades no implementadas en este transceptor de bajo costo.

Una vez que todos estos bits tienen un valor fijo, se posicionan los bits de cada parámetro uno a continuación de los otros (siguiendo el orden de las tablas 5.1 y 5.2). Esta cadena da como resultado a los bits del campo de datos.

5.2.2. Mensaje 24

El mensaje 24 corresponde al reporte de información estática por parte de estaciones de clase B. Se divide en dos partes, el mensaje 24 parte A (el cual será referido como mensaje 24 A) y el mensaje 24 parte B (el cual será referido como mensaje 24 B).

El mensaje 24 A es transmitido cada 6 minutos y se utiliza principalmente para asociar el MMSI de una embarcación con el nombre de la misma. En la tabla 5.3 se encuentran todos los parámetros que lo componen. El campo que indica la parte del mensaje se fija como 0 cuando se trata del mensaje 24 A. Para las pruebas, el valor del MMSI se fijó en 0123456789 al igual que en el mensaje 18 y como nombre de la embarcación se utilizó el nombre ficticio “ROMA”.

El mensaje 24 B es transmitido 1 minuto después de transmitir la parte A y en él se envía información de las proporciones de la embarcación y otros identificadores. En la tabla 5.4 se pueden ver todos los parámetros que lo componen. Para las pruebas de funcionamiento, se fijó el tipo de embarcación con un valor de 30 que se corresponde con un barco pesquero. Para el identificador del vendedor y la señal de llamada se utilizó el valor de “@@@@@” pues estas no están disponibles. Las dimensiones de la embarcación se fijaron en 8 m de largo y 4 m de ancho.

Al igual que en el mensaje 18, para ambas partes del mensaje 24 se tienen los 168 bits de carga útil. Es decir, se posicionan los bits de cada parámetro (siguiendo el orden de las tablas 5.3 y 5.4). Estas cadenas dan como resultado a los bits del campo de datos de cada mensaje.

5.3. Generación de mensajes

Se presenta en la figura 5.2 un diagrama de bloques con los pasos que se siguen para generar un mensaje. A lo largo de esta sección se explicarán uno a uno. La idea principal consiste en generar el mensaje a partir de la información de GPS o la información estática, y generar la trama AIS a partir de ese campo de datos. Luego se aplica bit stuffing, se codifica en NRZI y se modula en GMSK.

La implementación de la primera parte de este proceso se realizó en el bloque Messages, descrito en la sección 4.4.2 del capítulo 4. La salida de este bloque es un

Capítulo 5. Generación y codificación de mensajes

Tabla 5.1: Campos del mensaje 18 con sus respectivos largos en bits y descripciones.

Parámetro	Cantidad de bits	Descripción
Identificador de mensaje	6	Identificador para el mensaje 18. Siempre es 18.
Indicador de repetición	2	Utilizado por una estación repetidora para indicar cuántas veces el mensaje se repitió. El valor por defecto es 0, y se utilizan valores entre 0 y 3. El valor 3 indica que no se debe volver a repetir.
Identificador de usuario	30	Identificador MMSI.
Extra	8	No deben ser utilizados. Deben fijarse en cero.
Velocidad en tierra	10	Velocidad en tierra en incrementos de 1/10 de nudo. Se tienen valores entre 0 y 1.021 para indicar décimos de nudos. El valor 1.022 indica una velocidad de 102.2 nudos o mayor. El valor 1.023 indica que la velocidad no está disponible.
Precisión en la posición	1	Vale 0 para indicar una precisión baja (mayor a 10 m), y 1 para indicar una precisión alta (menor o igual a 10 m). Por defecto fijo en cero.
Longitud	28	Longitud en 1/10.000 min (va entre -180° y $+180^\circ$, con Este positivo y Oeste negativo). El valor 181° (6791AC0h) es el valor por defecto e indica que la longitud no está disponible.
Latitud	27	Latitud en 1/10.000 min (va entre -90° y $+90^\circ$, con Norte positivo y Sur negativo). El valor 91° (3412140h) es el valor por defecto e indica que la latitud no está disponible.
Dirección en tierra	12	Dirección en tierra en décimos, con valores entre 0 y 3.599. El valor 3.600 es el valor por defecto e indica que la dirección en tierra no está disponible. Los valores entre 3.601 y 4.095 no deben ser utilizados.
Dirección verdadera	9	Grados entre 0 y 359, con el valor 511 como valor por defecto, indicando que la dirección no está disponible.
Marca de tiempo	6	Segundo UTC en el que se generó el mensaje. Va entre 0 y 59, tomando el valor por defecto 60 si la marca de tiempo no está disponible. Los valores 61, 62 y 63 no son utilizados por las estaciones de clase CS.
Extra	2	Reservados para uso futuro. Deben estar fijos en cero.

5.3. Generación de mensajes

Tabla 5.2: Campos del mensaje 18 con sus respectivos largos en bits y descripciones.

Parámetro	Cantidad de bits	Descripción
Bandera de clase B de unidad	1	Vale 0 si es una estación de clase B SO, y vale 1 si es una estación de clase B CS.
Bandera de clase B de display	1	Vale 0 si no hay un display disponible (lo que implica que no se pueden mostrar los mensajes 12 y 14), y vale 1 si la unidad está equipada con un display integrado.
Bandera de clase B de DSC	1	Vale 1 si se tiene una función DSC y vale 0 si no.
Bandera de clase B de banda	1	Vale 0 si la estación es capaz de operar en la banda superior marítima de 525 kHz y vale 1 si es capaz de operar en toda la banda marítima. Este campo es irrelevante si la bandera de clase B de mensaje 22 es 0.
Bandera de clase B de mensaje 22	1	Vale 0 si no se puede cambiar la frecuencia de trabajo con el mensaje 22 y se opera únicamente en los canales AIS 1 y AIS 2. Vale 1 si se puede cambiar la frecuencia de trabajo con un mensaje 22.
Bandera de modo	1	Vale 0 si la estación está operando en modo autónomo y continuo y vale 1 si se está operando en modo asignado.
Bandera de RAIM	1	Bandera de monitoreo de integridad de receptor autónomo (RAIM, por sus siglas en inglés). Aporta información sobre el dispositivo electrónico de fijación de posición. Vale 1 si se utiliza RAIM, y vale 0 si no.
Bandera de selección de estado de comunicación	1	Vale 0 si a continuación se indica el estado de comunicación SOTDMA y vale 1 si sigue un estado de comunicación ITDMA. Está fija en 1 para estaciones de clase B CS.
Estado de comunicación	19	Estado de comunicación SOTDMA si la bandera de estado de comunicación es un 0, o estado de comunicación ITDMA si la bandera de estado de comunicación es un 1. En las estaciones de clase B CS este debe tener siempre el siguiente valor: 1100000000000000110.
Cantidad de bits	168	Se ocupa un slot.

Capítulo 5. Generación y codificación de mensajes

Tabla 5.3: Campos del mensaje 24 parte A, con sus respectivos largos en bits y descripciones.

Parámetro	Cantidad de bits	Descripción
Identificador de mensaje	6	Identificador para el mensaje 24. Siempre es 24.
Indicador de repetición	2	Utilizado por una estación repetidora para indicar cuántas veces el mensaje se repitió. El valor por defecto es 0, y se utilizan valores entre 0 y 3. El valor 3 indica que no se debe volver a repetir el mensaje.
Identificador de usuario	30	Identificador MMSI.
Parte del mensaje	2	Identificador de la parte del mensaje 24. Para la parte A, es siempre 0.
Nombre	120	Nombre de la embarcación registrada con el MMSI indicado. Se tienen como máximo 20 caracteres de ASCII de 6 bits.
Cantidad de bits	160	Se ocupa un slot.

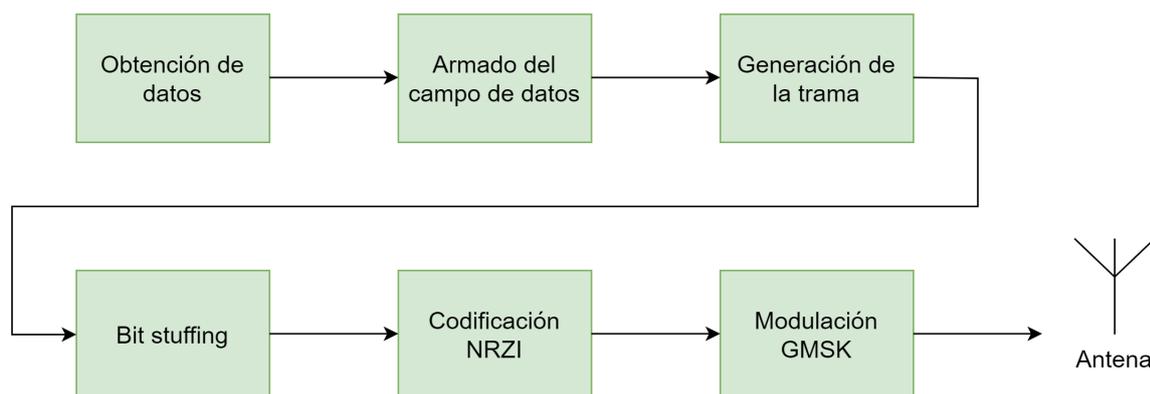


Figura 5.2: Diagrama de bloques para la generación de un mensaje. Se deben obtener los datos del módulo GPS previo al armado de la carga útil del mensaje. Se conforma la trama y se aplica el mecanismo de bit stuffing sobre la cadena de bits resultante. Luego se codifica en NRZI y se modula en GMSK, previo a la transmisión en el canal correspondiente.

5.3. Generación de mensajes

Tabla 5.4: Campos del mensaje 24 parte B, con sus respectivos largos en bits y descripciones.

Parámetro	Cantidad de bits	Descripción
Identificador de mensaje	6	Identificador para el mensaje 24. Siempre es 24.
Indicador de repetición	2	Utilizado por una estación repetidora para indicar cuántas veces el mensaje se repitió. El valor por defecto es 0, y se utilizan valores entre 0 y 3. El valor 3 indica que no se debe volver a repetir el mensaje.
Identificador de usuario	30	Identificador MMSI.
Parte del mensaje	2	Identificador de la parte del mensaje 24. Para la parte B, es siempre 1.
Tipo de embarcación	8	Vale 0 si el tipo de embarcación no está disponible. Los números entre 1 y 99 se definen en la sección 3.3.2 de la norma ITU-R M.1371-5. Los números entre 100 y 199 se reservan para uso regional, y los números mayores a 200 se reservan para uso futuro
Identificador del vendedor	42	Identificación única de la embarcación dada por el fabricante. Si no está disponible, se utiliza “@@@@@”.
Señal de llamada	42	Señal de llamada de la embarcación. Si no está disponible, se utiliza “@@@@@”.
Dimensiones de la embarcación	30	Dimensiones de la embarcación en metros. Además, se puede indicar la ubicación del equipo con el cual se reporta la posición.
Tipo de dispositivo electrónico para obtener posicionamiento	4	0 = No definido (valor por defecto); 1 = GPS, 2 = GLONASS, 3 = Combinación de GPS/GLONASS, 4 = Loran-C, 5 = Chayka, 6 = Sistema de navegación integrado, 7 = Encuestado, 8 = Galileo, 9-14 = No utilizados, 15 = GNSS interno
Extra	2	No utilizados, fijos en cero.
Cantidad de bits	168	Se ocupa un slot.

Capítulo 5. Generación y codificación de mensajes

string de bits que representa el campo de datos de la trama AIS. El bloque `Build AIS Frame` se encarga de conformar la trama y prepararla para la transmisión.

5.3.1. Obtención de datos

El primer paso consiste en obtener los datos necesarios para armar el campo de datos de una trama AIS. En el caso del mensaje 18, se debe recibir información desde el módulo GPS para saber la posición de la embarcación, la velocidad de la misma y otras características de su movimiento. En el caso del mensaje 24, la información transmitida es estática. Tanto la parte A como la parte B pueden generarse a partir de información que debe estar almacenada en la memoria interna del sistema transmisor.

5.3.2. Armado del campo de datos del mensaje

Se tomó como base la generación de mensajes de *gr-aistx*, en particular el archivo *AIVDM_Encoder.py*. Este script se encuentra escrito en lenguaje Python y se tomaron las funciones generadoras de los mensajes que se transmitirán (18, 24 A y 24 B). Se fijaron los campos estáticos para los mensajes en los valores correspondientes a la estación AIS implementada. Los valores utilizados para los campos estáticos de los mensajes fueron descritos en las secciones 5.2.1 y 5.2.2.

El funcionamiento de la generación de mensajes es el siguiente. Se consideran los parámetros descritos en la tabla correspondiente (tablas 5.1 y 5.2 para el mensaje 18, tablas 5.3 y 5.4 para el mensaje 24). El proceso consiste en concatenar los bits de cada uno de los parámetros, utilizando el valor por defecto considerado para el proyecto, o el valor obtenido del módulo GPS. El resultado de estas funciones es un string de bits con el campo de datos de un mensaje AIS.

5.3.3. Generación de la trama

Se genera la trama a partir del campo de datos que se tiene. En primer lugar se calcula el CRC de 16 bits sobre los datos y se concatena al final de estos.

5.3.4. Proceso de bit stuffing

Se aplica bit stuffing sobre el campo de datos y el valor de CRC calculado. Esto implica que se agregará un bit en 0 inmediatamente después de cinco bits seguidos en 1. Hecho esto, se agregan las banderas de inicio y finalización, ya que no están sujetas a bit stuffing. Se adjunta también la secuencia de entrenamiento.

5.3.5. Codificación en NRZI

Se codifica la trama obtenida en NRZI, como fue descrito en el capítulo 2. Se recuerda que los cambios de nivel se dan al recibir bits en 0.

5.3.6. Modulación GMSK

Se aplica modulación GMSK sobre la trama codificada en NRZI. Para esta modulación se tiene el producto $B.T = 0,4$, donde B es el punto con -3 dB de caída y T hace referencia a la duración del pulso en el tiempo.

5.4. Ejemplo de mensaje AIS

Si bien la información que se envía es la que está en los campos de la figura 5.1 (luego de los procesos descritos en 5.3), se suele utilizar otra representación para los mensajes. Se utiliza una representación denominada sentencia NMEA. Esto se realiza para que el usuario pueda determinar de forma más sencilla si efectivamente está recibiendo un mensaje AIS y para que cierto tipo de información (como el canal usado) esté más accesible. Estos mensajes consisten de la secuencia binaria del campo de datos en su representación ASCII (agrupando bits de a seis) precedida por algunos caracteres de comienzo (!AIVDM) que se mantienen invariados para todos los mensajes.

Se presenta a continuación un ejemplo de la representación NMEA de un mensaje AIS. Se explicarán sus campos para comprender su estructura, y se realiza un análisis de cómo obtener información AIS de dicha sentencia. Los mensajes de ejemplo fueron recuperados de [48].

```
!AIVDM,1,1,,A,14eGL:@000o8oQ`LMjOchmG@08HK,0*40
```

El ejemplo comienza con el caracter “!”, lo que indica que el mensaje a continuación fue encapsulado [49]. Se tienen los caracteres “AI”, que hacen referencia a que el mensaje fue transmitido por una estación AIS móvil. Los siguientes tres caracteres, “VDM”, culminan el encabezado de la sentencia NMEA, y refieren al tipo del mensaje. En particular, se utiliza una sigla para indicar “Mensaje de Enlace de Datos de Muy Alta Frecuencia” (VDM, por sus siglas en inglés). A continuación se tienen los campos de la sentencia NMEA.

Los campos se delimitan con comas, y quedarán vacíos aquellos que no contengan información. Los caracteres “1, 1, , A,” indican, respectivamente la cantidad de sentencias correspondientes a este mensaje, el número de la sentencia actual y el canal a través del cual se recibió el mensaje [48]. El campo vacío se utiliza para mensajes de más de una sentencia de largo. El siguiente campo es el de mayor longitud, y contiene el campo de datos del mensaje AIS. Observar que al recibir un mensaje AIS, los receptores eliminan las banderas de inicio y fin, invierten el bit stuffing y corroboran el código CRC. Al hacer el pasaje a NMEA, únicamente se deja el campo de datos del mensaje AIS original.

En el ejemplo, se tiene que la carga útil del mensaje se codificó como “14eGL:@000o8oQ`LMjOchmG@08HK”. Para esta codificación, se parte de los bits y se agrupan de a seis pero se utiliza la tabla ASCII de 8 bits. Para hacer la adaptación desde los seis bits y utilizar la tabla se hacen algunas operaciones. En particular como no se utilizan los caracteres x,y,z, ni X,Y,Z (ningún valor ASCII

Capítulo 5. Generación y codificación de mensajes

desde 88 a 95); si el valor de los bits es mayor a 39 se debe sumar 8. Luego, en cualquier caso (se le haya sumado 8 a los seis bits o no), se le debe sumar 48 y utilizar la conversión de bits a carácter ASCII [50]. De esta forma, el mensaje utilizado como ejemplo tiene los primeros bits de la forma siguiente: “000001 000100 101101 010111 011100 001010”. Para una mejor interpretación, los bits ya fueron separados de a seis. Se puede ver que el valor decimal de los bits agrupados de a seis es “1 4 45 23 28 10”. Haciendo las operaciones mencionadas, el valor en ASCII de los números es “49 52 93 71 76 58” y si se utiliza una tabla ASCII, los caracteres coinciden con los primeros seis de la carga útil del mensaje “1 4 e G L :”.

En este capítulo, se describió la generación y la codificación de mensajes. Se definió qué mensajes envía el transceptor implementado en base al receptor utilizado, las exigencias de la norma y con el objetivo de mantener los costos acotados. Para cada uno de los mensajes utilizados, se describió su utilidad y contenido, y se detalló el proceso de generación. Se especificó la codificación y la modulación, finalizando con un ejemplo de un mensaje AIS en NMEA y su interpretación.

Capítulo 6

Pruebas de funcionamiento y optimización

El desarrollo de prototipos en cualquier área de ingeniería debe estar acompañado por un posterior proceso de pruebas de funcionamiento. Es de importancia caracterizar el producto obtenido. Esto permite comunicar las capacidades del dispositivo de forma más transparente a potenciales usuarios.

A lo largo de este trabajo se ha descrito la implementación de un sistema transmisor de la tecnología AIS. Se indicó cómo se realiza el acceso al medio y cómo se generan y codifican los mensajes a transmitir. El desarrollo tomó como base a la norma ITU-R M.1371-5 que describe las estaciones AIS [16]. La caracterización del prototipo obtenido consistirá en analizar el cumplimiento de los requerimientos de la norma.

En este capítulo se estudia el sistema transmisor implementado. La estructura del código y de los bloques es la de un módulo *out-of-tree* de GNU Radio, llamado *gr-itais*. En primer lugar se detalla el proceso de optimización de los bloques desarrollados. Se mencionan mejoras implementadas con el principal fin de cumplir con restricciones de tiempos. Una vez se describieron las consideraciones para obtener las versiones finales de los códigos, se pasa a las pruebas de funcionamiento. Se realizan pruebas generales, para verificar que se tiene el comportamiento usual de un transmisor, y también pruebas específicas, contrastadas con lo indicado en la norma de referencia. Se detallan procedimientos seguidos y se analizan los resultados obtenidos.

6.1. Proceso de optimización del sistema

La metodología de trabajo seguida para la implementación del sistema transmisor es la siguiente. Se desarrollaron módulos independientes, cuyo funcionamiento se verificó en ambientes controlados. Esto implicó la generación de esquemas de GNU Radio simplificados. Se desarrollaron bloques extras para simular la interacción del bloque a probar con otros bloques. Este proceso resultó en correcciones de código y cambios en las estructuras de datos utilizadas para guardar o intercambiar

Capítulo 6. Pruebas de funcionamiento y optimización

biar valores. Una vez todos los bloques fueron probados por separado, se realizaron pruebas del sistema completo. Esto llevó a modificaciones dentro de bloques o a nivel del sistema, cambiando interconexiones entre bloques.

Se detallan a continuación las mejoras y optimizaciones mencionadas. El principal desafío en la mayoría de los casos fue el cumplimiento con restricciones de tiempo impuestas por la norma de referencia. En las optimizaciones se colocó un límite de tiempo de trabajo para cada bloque o problema particular y se buscó avanzar lo más posible hacia la solución del mismo. En algunos casos fue posible cumplir el objetivo y en otros no. Se priorizó la terminación de un sistema completo funcional y no el cumplimiento de todos los detalles descritos en la norma. Más adelante en este capítulo, al describir las pruebas de funcionamiento, se detallarán las restricciones que se cumplieron y sobre aquellas en las que se debería seguir trabajando a futuro.

6.1.1. Mejoras a nivel de bloques

Se presentan mejoras realizadas a nivel de bloques, algunos de ellos desarrollados en el marco de este trabajo y otros adaptados de bloques ya existentes en GNU Radio o en módulos *out-of-tree*.

Bloque Potumbral

En la implementación de un sistema transmisor AIS de clase B CS, es de vital importancia conocer la potencia actual en cada canal y también su umbral. Sin esta información, los sistemas no sabrán si pueden transmitir cuando desean hacerlo. Para posibilitar el cálculo de estos valores en cada uno de los canales, el bloque `Potumbral` recibe muestras complejas del canal.

Inicialmente, con el objetivo de tener una alta precisión en los valores de potencia y umbral, este bloque procesaba cada muestra de forma individual, manteniendo separados los valores para el canal A y el canal B. Algunas ejecuciones de este bloque recibiendo muestras y calculando el umbral fueron suficientes para notar que la alta precisión generaba retardos importantes. Estos retardos se volvían más significativos al ejecutar el bloque en los equipos de menor costo (donde pretende utilizarse el sistema). Por esta razón, `Potumbral` debió ser modificado.

En lugar de contar con el procesamiento para los dos canales, el bloque procesa las muestras que recibe manteniendo un único valor de umbral y potencia. Para poder manejar ambos canales, se hacen dos instancias de este bloque, teniendo un camino distinto para cada uno (una instancia de la figura 4.2 para cada canal). Además, para tener un procesamiento más rápido y eficiente, no se procesan todas las muestras que ingresan al canal. Por el contrario, se toma una muestra de cada diez y la potencia de dicha muestra es la potencia actual del canal, asumiendo que su valor es representativo de las siguientes nueve muestras. Si bien esto no es siempre cierto y es menos preciso que tomar las muestras de a una, se observó una mejora en el desempeño y que el cálculo del umbral y la potencia seguía siendo bueno.

6.1. Proceso de optimización del sistema

Se modificó también la conexión de `Potumbral` con el resto del sistema. La idea inicial era que este bloque envíe como salida la potencia y el umbral actual del canal para que la decisión de si es posible transmitir o no sea tomada por el bloque `Transmitter`. Como esto se pretendía hacer con cada muestra procesada, `Transmitter` precisaba saber si el valor de potencia actual y el umbral pertenecían a muestras dentro del rango donde el canal debe ser sensado. Esto se pretendía lograr etiquetando los valores de relevancia.

El etiquetado se pretendía hacer dentro un bloque `ZMQ PUB` que enviaría las muestras hacia el bloque `Transmitter`¹. No se logró un buen funcionamiento al intercambiar los mensajes con etiquetas. En esa etapa del proyecto, se priorizaba el uso de los bloques `ZMQ` como bloques de interconexión. Por este motivo se tomó la decisión de una modificación en la información procesada y enviada por el bloque `Potumbral`. La salida del bloque pasó a ser el número del slot actual y un 0 o un 1 para indicar si se puede transmitir (1) o no (0) en dicho slot. Pasa a ser `Potumbral` el bloque en el que se toma la decisión de poder transmitir o no en cada slot. Para evitar enviar esta información en todos los slots y así ahorrar procesamiento, `Potumbral` recibe desde `Transmitter` los slots candidatos donde se desea hacer la transmisión. De esta forma, se optimiza la toma de decisión, que recae en `Potumbral` y `Transmitter` solo mira si los slots de interés tiene un 1 o un 0.

Se realizaron otros pequeños ajustes para intentar optimizar aún más el procesamiento del sistema completo. En particular, en lugar de calcular la potencia de las muestras complejas dentro de `Potumbral`, se agregó un bloque de `GNU Radio` que calcula el módulo al cuadrado de todas las muestras. De esta forma, `Potumbral` ya recibe los valores de potencia que puede utilizar para comparar con el umbral o actualizar el mismo. Si bien las mejoras con esta modificación no fueron significativas, se mantuvo este bloque pues el procesamiento de bloques predefinidos de `GNU Radio` suele ser más eficiente.

Luego de todas estas modificaciones, el desempeño de `Potumbral` mejoró de forma significativa. Su optimización permitió un funcionamiento del sistema completo más cercano a lo que se especifica en la norma. Esto se verá en mayor detalle en la sección 6.3.

Bloque Transmitter

El bloque `Transmitter` es el que implementa la inteligencia principal del esquema de bloques. Se realizaron pequeñas modificaciones sobre este bloque para asegurar un correcto funcionamiento de todo el sistema.

Debido a las restricciones de tiempo existentes, se tienen momentos críticos en el sistema. Estos son los momentos de la transmisión de un mensaje. Dentro de un slot de 27 ms se debe recibir desde el bloque `Potumbral` la habilitación para transmitir, se debe indicar al bloque `Messages` que se genere un mensaje y desen-

¹Se desarrollará más adelante en este capítulo en qué consistió el uso de los bloques de tipo `ZMQ`. Además, se explicará por qué no fueron considerados para el modelo final del sistema.

Capítulo 6. Pruebas de funcionamiento y optimización

cadena los siguientes pasos necesarios. Por este motivo, se buscó descongestionar esos períodos de tiempo. La principal estrategia utilizada es la siguiente. Cuando faltan 400 slots de tiempo para el slot central de una transmisión (que se corresponden con unos 10 s), el bloque `Transmitter` envía al bloque `Potumbral` cuáles son los slots candidatos para dicha transmisión. Esto evita tener que enviarlos en el momento crítico. El proceso receptor ya sabrá cuáles slots deberá sensar. Una vez se transmitió, se envía al proceso receptor un arreglo con valores -1 en todas las entradas, para evitar que siga sensando slots hasta la próxima transmisión.

En el momento en que se envían los slots candidatos para la transmisión se envía también el canal de transmisión al bloque selector de canal. Esto asegura que estará sintonizado a la frecuencia de interés al momento de transmitir.

Los cambios previos están asociados a cumplir restricciones de tiempo de la mejor forma posible.

Bloque Messages

La correcta generación del campo de datos de los mensajes AIS es de gran importancia. En un comienzo se consideró utilizar la librería de Python `pyais` para la generación de mensajes [51]. Sin embargo, esta generación resultaba en secuencias NMEA y se necesitaba tener el mensaje a nivel de bits. Se utilizó la generación de mensajes del módulo `gr-aistx` como referencia, que devuelve como resultado el campo de datos de un mensaje como un string de bits.

La norma indica que no deben transmitirse los mensajes 18 o 24 (en cualquiera de sus dos partes) si no se cuenta con información de GPS confiable. Para evitar complejizar aún más el bloque de transmisión, el descarte de los mensajes se realiza en el bloque `Messages`.

Bloque Sub_gps

Al probar algunos bloques del proyecto se detectó que un bloque de GNU Radio no comienza a funcionar hasta que tiene valores en todas sus entradas. El bloque problemático en este caso es `Transmitter`, cuya entrada está asociada a `Sub_gps`. Por fuera de GNU Radio se ejecuta un script de Python que se comunica con el módulo GPS y envía la información recibida a un socket. `Sub_gps` tiene en su código una conexión a dicho socket, y de esta forma inyecta la información de GPS al esquema de GNU Radio.

El script de Python obtenía información del módulo GPS cada un segundo, por lo que cada ese intervalo de tiempo se enviaba información al socket. El bloque `Transmitter` recibía entonces información cada un segundo. Este hecho enlentecía el sistema entero, ya que el transmisor realizaba cierto procesamiento solo una vez por segundo.

Se realizaron modificaciones en el script de Python que se comunica con el módulo GPS. La información se guarda y envía al socket en formato de diccionario. Inicialmente se contaba con código que se ejecutaba únicamente al recibir información del módulo GPS. El principal cambio consiste en que mientras no se

6.1. Proceso de optimización del sistema

recibe nueva información del GPS, se continúa enviando al socket el último diccionario obtenido. Estas mejoras permitieron un correcto funcionamiento del bloque `Sub_gps`, y del posterior `Transmitter`.

Se hizo otra corrección sobre el script de Python. Cuando no se cuente con información GPS confiable (esto es, no se tenga al menos 5 o 6 satélites), el diccionario a enviar tendrá todos los valores fijos en cero. Así, el bloque `Messages` podrá identificar cuando no se tiene GPS *fix*, y descartará los mensajes. Esto se debe a que la norma indica que no se deben transmitir mensajes 18, ni 24, cuando no se cuenta con información GPS confiable.

Bloque Selector

En la tecnología AIS es necesario alternar el canal de transmisión entre mensajes. Se recuerda que se cuenta con dos canales de frecuencia: el canal A, centrado en 161,975 MHz, y el canal B, centrado en 162,025 MHz.

En un principio se consideró un esquema de GNU Radio del módulo *gr-aistx* como referencia para la transmisión. En particular, se tomaron los bloques y conexiones para la conformación de la trama, la codificación NRZI, la modulación GMSK y el pasaje a la antena, tal como se ve en la figura 4.7. Previamente a la transmisión en RF del mensaje, el esquema cuenta con un bloque `Selector` de GNU Radio. Las entradas del mismo eran la señal a transmitir transportada a -25 kHz y +25 kHz, ya que la transmisión se centra en 162 MHz. Se asumió que el bloque funcionaba como un multiplexor, alternando la transmisión entre los dos canales. Sin embargo, al verificar las transmisiones en una computadora con la herramienta de software SDR++ [52] se observó que se estaba transmitiendo en ambos canales a la vez.

Se decidió implementar un bloque `Selector` propio. La idea se trató de un bloque que reciba un mensaje desde `Transmitter` indicando el canal de transmisión. En función del canal, se coloca en su única salida la entrada correspondiente (la señal corrida -25 kHz para el canal A, la señal corrida +25 kHz para el canal B). No se obtuvieron los resultados esperados, a pesar de que el bloque implementado era muy sencillo y fue revisado en detalle.

Se realizó un estudio en mayor profundidad del bloque `Selector` provisto por GNU Radio. Existe la posibilidad de activar entradas de tipo mensaje al bloque. La entrada de mensajes disponible en GNU Radio 3.8 no era de utilidad para el equipo. Se precisaba el funcionamiento que el bloque `Selector` tenía implementado para GNU Radio 3.9. Por este motivo, se generó un bloque `Selector_39` en lenguaje C++ dentro del proyecto. Se adaptó el código del bloque `Selector` de GNU Radio 3.9 para poder utilizarlo con GNU Radio 3.8. De esta forma se consiguió el funcionamiento esperado.

Es importante notar que no es usual el pasaje de bloques de versiones más nuevas de GNU Radio a versiones anteriores, y en muchos casos puede no ser recomendable. El bloque considerado es muy simple. La migración de una versión a otra no implicó cambios significativos.

Al medir tiempos de las transmisiones, se observaron retardos cercanos a 20 s. Un análisis de los bloques utilizados permitió concluir que el bloque culpable de

Capítulo 6. Pruebas de funcionamiento y optimización

estos retardos era el bloque `Selector_39`. La presencia de ese bloque generaba que la mitad de los mensajes AIS tuvieran un retardo de 20 s, y la otra mitad de 2 s. Se realizó una prueba quitando dicho bloque, transmitiendo en un solo canal todos los mensajes, y el retardo era siempre de unos 2 s. A través del estudio del código, se determinó que el valor de muestras de entrada al bloque no era inicialmente constante. Por este motivo, se fijó la variable asociada en 4.096. Esto eliminó los retardos de 20 s. En las secciones 6.3.8 y 6.3.9 se realiza un estudio más en detalle sobre el retardo asociado a las transmisiones realizadas.

Bloque `Build_AIS_Frame`

Se realizó principalmente una modificación al bloque, debido a la detección de un funcionamiento inesperado del sistema. Se observó que el segundo UTC indicado en los mensajes transmitidos no coincidía con el segundo UTC en que se generaba y enviaba el mensaje. Al analizar los tiempos, se pudo ver que el receptor detecta un mensaje X justo cuando se manda un mensaje Y, siendo el mensaje X previo al mensaje Y. Luego, el mensaje Y es detectado por el receptor al enviar el mensaje Z. Este comportamiento se repetía de forma sucesiva. Para evitar este desfase, se decidió utilizar mensajes “Dummies” compuestos de 155 bits de ceros. De esta forma, luego de cada mensaje que se envía desde el bloque `Messages`, se genera otro mensaje compuesto por ceros.

El objetivo fue generar mensajes “Dummies” que no sean interpretables por el receptor. Para no agregar el preámbulo y las banderas a estos mensajes, el bloque `Build AIS Frame` fue modificado. Se fijaron las banderas y el preámbulo como secuencias de ceros en el caso de que el mensaje recibido consista de 155 bits (algo que no se cumple con ninguno de los mensajes reales a enviar). Al estar el mensaje compuesto enteramente por ceros, no será interpretado por los receptores como un mensaje real. Es importante destacar que sí se tiene la desventaja de generar un cierto ruido en el canal. Algo que se confirma al realizar la prueba 6.3.9.

6.1.2. Mejoras a nivel del sistema completo

Se presentan mejoras realizadas a nivel del sistema completo. A no ser que se indique lo contrario, las optimizaciones fueron realizadas con el objetivo de cumplir lo mejor posible las restricciones de tiempo impuestas por la norma.

Al inicio del desarrollo de bloques se trabajó con bloques del tipo `Embedded Python Block`. Se trata de bloques de GNU Radio que permiten al usuario fácilmente incorporar un código personalizado en los esquemas. Para formalizar el entorno de trabajo y para evitar potenciales demoras, se generó un módulo *out-of-tree* de GNU Radio llamado *gr-itais*. En el mismo se crearon los bloques necesarios desde cero, en su mayoría en Python (a excepción del bloque `Selector_39`).

Para el trabajo se tomó como referencia al módulo *gr-ais*. El mismo, en su script *radio.py*, implementaba un esquema de GNU Radio sin interfaz gráfica. Debido a que no se contaba con una implementación de la interfaz gráfica de todos los bloques utilizados en este proyecto, se siguió la misma estrategia.

6.1. Proceso de optimización del sistema

La primera idea consistió en desarrollar un script *rf.py* en el que se encontrarán los bloques asociados a la transmisión únicamente. Así, `Potumbral` se colocaría en el proceso receptor de *radio.py*, y el resto de los bloques en el proceso transmisor de *rf.py*. La interconexión entre ambos esquemas sería con sockets TCP. En particular, GNU Radio ofrece bloques `ZMQ_PUB` y `ZMQ_SUB`. Estos permiten publicar hacia y recibir desde un socket TCP, respectivamente. Sin embargo, la transmisión a través de los sockets generó muchos problemas. No permitió el trabajo con etiquetado de muestras, los valores intercambiados no siempre eran los esperados y las demoras de tiempo eran muy significativas.

En una segunda idea se planteó la generación de un script llamado *itais_radio.py*. En el mismo se colocarían los bloques necesarios para el sistema completo, para transmisión y recepción. Esto presentó un problema que debía resolverse. GNU Radio no permite el manejo de ciclos o *loops* en sus esquemas, a no ser que se traten de comunicaciones de tipo mensaje o se armen bloques de tipo jerárquicos. En el caso particular del sistema implementado, las comunicaciones entre el bloque `Transmitter` y el bloque `Potumbral` se realizaban como arreglos de complejos y/o reales, y generaban loops en el esquema de GNU Radio.

La generación de un bloque jerárquico sería compleja, considerando la cantidad de subbloques que se agregarían. Además se dificultaría la detección y corrección de errores en el código. En el esquema realizado, el bloque `Transmitter` le envía sus candidatos a `Potumbral` y recibe desde este un arreglo con un número de slot y si es posible transmitir en él o no. Se pasó entonces a la utilización de puertos de tipo mensaje para estos intercambios. Esto permitió la presencia de loops. Además, resultó que la comunicación entre bloques con mensajes era más rápida. Esto posibilitó cumplir ciertas restricciones de tiempo que no eran viables con los bloques de comunicación socket que se tenía previamente.

La idea que prevaleció para el sistema fue la segunda. Se cuenta entonces con el script *itais_radio.py*, donde se inicializan todos los bloques necesarios y sus conexiones para la implementación del sistema transmisor.

Las primeras pruebas de funcionamiento del script completo mostraron que no se estaban cumpliendo algunas restricciones de tiempo de la norma. Entre ellas, el tiempo de actualización del umbral en la inicialización del sistema o los tiempos de sensado del canal. Se realizaron modificaciones en los bloques por separado para mejorar los tiempos. Además, se probó una estrategia a nivel de sistema completo a partir de un módulo *out-of-tree* de GNU Radio llamado *gr-latency-manager* [53]. Dicho módulo permite regular el flujo de muestras entre bloques [54]. Se colocaron en distintas posiciones del esquema de GNU Radio, pero no se notaron resultados relevantes. Las principales mejoras se dieron luego de optimizar cada bloque por separado.

La incorporación de todos los bloques de transmisión y de recepción en un único esquema de GNU Radio requería un nivel de procesamiento considerable para la placa utilizada. Cuando se contaba con un esquema para transmisión y otro para recepción, se lograba recibir de forma correcta mensajes AIS en formato de secuencia NMEA. Los mismos se imprimían en la terminal. Sin embargo, al utilizar un solo esquema, no se logró la recepción de más mensajes. Se cree que

Capítulo 6. Pruebas de funcionamiento y optimización

esto se debe al alto procesamiento que se estaba realizando en ese esquema de GNU Radio, al punto de que el *scheduler* de trabajos tenía muchos bloques para atender a la vez, varios de ellos muy demandantes.

La recepción de mensajes AIS no es un objetivo de este trabajo. Se consideró relevante conseguir un sistema completo que fuera capaz de ello, más allá de que no se interpreten los mensajes recibidos, ni se actúe en función de ellos. La solución hallada fue la eliminación de los bloques de recepción de mensajes (eliminando bloques para demodulación, desarmado de tramas y pasaje a NMEA) del script *itais_radio.py*. Se retomó el script *radio.py* tal y como se presenta en el repositorio [25]. Se probaron dos estrategias para dicho script. Una de ellas fue agregar un bloque ZMQ PUB a *itais_radio.py* para enviar las muestras recibidas desde el dispositivo SDR hacia un socket. Desde *radio.py* se recibirían dichas muestras con un ZMQ SUB y se decodificarían mensajes. La otra estrategia implicó el agregado de un RTL SDR al sistema, cuyas muestras se utilizaran para la recepción de mensajes.

La estrategia decidida fue la segunda, ya que la primera no permitió la recepción de mensajes. El agregado de un nuevo dispositivo SDR aumenta el precio final del prototipo obtenido. Sin embargo, se priorizó contemplar una recepción de mensajes. A futuro, tendría sentido poder interpretar mensajes e implementar modos de funcionamiento como el asignado o el de por interrogación. Para esto, será necesario recibir mensajes.

6.2. Preparación del sistema para las pruebas

En esta sección se describirán los comandos que deben ejecutarse en la terminal para encender el sistema transmisor. Más adelante, en la sección 6.3, se describirá cada prueba realizada. Se consideró importante contar con una descripción a nivel de software. El procedimiento aquí descrito es común para todas las pruebas. En el repositorio del proyecto² se presenta en más detalle cómo poner en funcionamiento el sistema, además de explicar cómo se desarrollaron los scripts y por qué.

Previo a la configuración del software, es importante contar con algunos dispositivos de hardware. En primer lugar, serán necesarios una computadora y un puesto de trabajo adecuado con monitor, teclado y ratón. Además, será necesario conectar un módulo GPS (a través de conexión serial si se cuenta con pines I/O, a través de USB en caso contrario). Se requieren dos dispositivos SDR, uno de ellos con la posibilidad de transmitir y recibir, y otro con al menos la posibilidad de recibir.

Será necesario ejecutar tres comandos en total. Se presenta cada uno de ellos por separado, en el orden en que se deben ejecutar, y se procede a explicar su utilidad.

²La versión final del repositorio está disponible en <https://gitlab.fing.edu.uy/rominag/itais>.

6.3. Pruebas de funcionamiento y análisis de resultados

```
python get_gps_fast.py
```

El primer comando es para iniciar el script que se comunica con el módulo GPS³. Se toma la información recibida desde el sensor y se coloca la misma en el formato deseado. Se envía la información a un socket para inyectarla en GNU Radio (y poder acceder a ella desde el bloque `Sub_gps`). Este comando deberá ejecutarse desde el *background* si se desea utilizar una única terminal para el transmisor (por ejemplo, al trabajar en una instancia de virtualización de Docker). No se tendrán impresiones en terminal debido a este comando.

```
./gr-itaish/apps/ais_rx -s osmocom
```

El segundo comando permite iniciar el receptor de mensajes AIS. Se utiliza el receptor que viene por defecto en `gr-ais`, con una fuente de SDR de tipo `osmocom` (indicando que se utiliza un RTL SDR). A diferencia del comando anterior, no es necesario que este comando se ejecute. Esto se debe a que la recepción de muestras e interpretación de mensajes se hará de forma independiente. Es recomendable utilizarlo en la misma terminal. Esto es porque se imprimirán los mensajes de otras embarcaciones y se dará un aspecto más realista al transmisor. Se recibirán además algunas de las transmisiones propias.

Los mensajes recibidos se imprimirán en la terminal en formato NMEA. En caso de querer interpretarlos y ver el contenido de los mismos, existen páginas en Internet que permiten hacerlo.

```
./gr-itaish/apps/itaish_tx -s pluto
```

El último comando es el botón de inicio del sistema transmisor. Se indica que se utilizará un SDR ADALM-PLUTO. Como resultado, se obtienen primero mensajes de inicialización del transmisor y de los dos procesos de `Potumbral` que calcularán el umbral dinámico. Una vez terminan los primeros 60 s de inicialización del sistema transmisor, se indicará esto en la terminal, junto con el siguiente slot en el que se buscará transmitir. En el funcionamiento en régimen, se imprimirá cada vez que se envían los slots candidatos a `Potumbral` previo a una transmisión, se indicará el resultado del sensado del canal en esos slots y se imprimirá un mensaje cuando se logró transmitir.

6.3. Pruebas de funcionamiento y análisis de resultados

Se realizaron un total de nueve pruebas de funcionamiento. Se hicieron pruebas de verificación, cuyo fin fue comprobar que la implementación obtenida es correcta. Se hicieron también pruebas de medición, en las que se buscó medir ciertos

³El nombre del script puede cambiar según la placa en la que se ejecute. Esto se debe a que la conexión placa-módulo GPS es a través de USB en el NUC y a través de pines I/O en la Raspberry Pi.

Capítulo 6. Pruebas de funcionamiento y optimización

parámetros del transmisor resultante.

Previo a analizar en detalle cada una de las pruebas realizadas, se describirá a continuación un punteo general del funcionamiento del sistema. Este comportamiento más general fue observado a medida que se agregaron algunas funcionalidades. Los objetivos más específicos del proyecto como acceso al medio, generación y codificación de mensajes son evaluados con pruebas especialmente diseñadas.

- La inteligencia general del sistema funciona de forma correcta.
 - Los temporizadores son fijados con el valor correspondiente y se decrementan de forma esperada.
 - El intercambio de mensajes entre el bloque `Transmitter` y `Potumbral` se hace correctamente. `Potumbral` sensa solo los slots candidatos y `Transmitter` envía los mensajes solo en aquellos donde puede transmitir.
- El intervalo de transmisión del mensaje 18 se fija correctamente de acuerdo con la velocidad indicada por el GPS.
- Si el sistema no cuenta con GPS fix, no transmite mensajes.
- Las transmisiones consecutivas alternan el canal, excepto por el mensaje 24 A y el 24 B que se envían en el mismo.
- En caso de que no se pueda transmitir en ninguno de los 10 slots candidatos, la transmisión falla y se reinicia el temporizador. En esas circunstancias el siguiente mensaje se envía en el mismo canal donde debía enviarse el fallido.

A continuación se detalla cada una de las pruebas, explicando su cometido, el procedimiento seguido y los resultados obtenidos. Se realizan breves análisis de dichos resultados y se mencionan posibles causas y consecuencias. Las pruebas no están ordenadas de ninguna manera específica. Algunas de ellas se realizaron solamente en la placa Intel NUC, otras se realizaron también en la placa Raspberry Pi. El hardware en el que se ejecutan las pruebas provocó distintos resultados, por lo que se consideró de relevancia hacer distinciones para algunas pruebas. Aquellas pruebas que fueran sensibles a la capacidad de cómputo de la placa en que se ejecutaban fueron probadas en ambas placas.

6.3.1. Prueba 1: Potencia de transmisión

Se buscó medir la potencia de transmisión obtenida con el sistema armado. Esta prueba se ejecutó en las dos placas consideradas en el proyecto. Se utilizó un medio cableado.

Se conectó el dispositivo ADALM-PLUTO a un atenuador y luego a través de un cable a un analizador de espectros. Se utilizó la opción de “Max hold” para guardar el valor de la potencia recibida por el analizador. Se tomaron 10 medidas de transmisiones para cada placa, considerando cinco veces el canal A y

6.3. Pruebas de funcionamiento y análisis de resultados

cinco veces el canal B. Luego se promediaron estas medidas. Para los resultados numéricos fue necesario tener en cuenta una atenuación de 5 dB configurada en el ADALM-PLUTO y el atenuador de 30 dB agregado al circuito por precaución.

Los resultados obtenidos son muy similares para ambas placas. Esto se debe a que la potencia de transmisión no depende de la placa, sino del dispositivo SDR utilizado. En la placa NUC se obtuvo una potencia de transmisión máxima de -1,30 dBm en el canal A y de -1,40 dBm en el canal B. En la placa Raspberry Pi, en ambos canales se obtuvo una potencia de transmisión máxima de -1,55 dBm.

La norma ITU-R M.1371-5 establece que la potencia de transmisión debe ser de 1 Watt para estaciones de clase B CS⁴. Este número no fue cumplido, debido a que el dispositivo SDR de transmisión no alcanza valores de potencia tan altos [55]. En la banda de frecuencias de 162 MHz, el ADALM-PLUTO tiene una máxima potencia de transmisión de 6 dBm, lo que equivale a 3,98 mW. La principal consecuencia que esto implica es no lograr el alcance deseado con las transmisiones. Sin embargo, podría mejorarse el alcance agregando amplificadores o utilizando antenas diseñadas para la banda de frecuencias correspondiente⁵.

6.3.2. Prueba 2: Tiempo de cambio de canal

Se midió el tiempo que requiere el transmisor implementado en cambiar el canal de transmisión en el que está sintonizado. Esta prueba se ejecutó en las dos placas consideradas en el proyecto.

Es importante aclarar que la norma no indica cómo se debe medir este tiempo. Dada la implementación, el tiempo que lleva cambiar el canal de sintonización es el tiempo entre que el bloque `Transmitter` envía el canal de transmisión y que el bloque `Selector` recibe dicha información. Se agregaron impresiones en terminal apenas posteriores a la transmisión y recepción del canal, con precisión de microsegundos. Se tomaron 10 medidas para cada placa y se promedió el tiempo final.

Se presentan en la tabla 6.1 los resultados obtenidos. Es posible observar una demora significativamente mayor para la Raspberry Pi. Se cree que esto se debe a que el procesador de dicha placa es más lento que el procesador de la placa NUC. En general, durante las pruebas, es sencillo notar que la Raspberry Pi tiene un desempeño peor que el de la placa NUC.

La norma ITU-R M.1371-5 indica que el valor del tiempo de cambio de canal debe ser como máximo de 25 ms. La implementación realizada cumple con dicha especificación de forma muy suelta. Se cree que esta diferencia se debe a la implementación de un transmisor siempre sintonizado en 162 MHz, cuyas transmisiones alternan entre el canal A y el canal B a partir de un bloque selector de flujo. Si bien se cumple con el tiempo requerido, se tiene la desventaja de que en algunas transmisiones el cambio de canal no es efectivo. Además, en algunos envíos se tiene una pequeña señal molesta en un canal cuando se transmite en el otro.

⁴El valor de 1 Watt equivale a 30 dBm.

⁵En el caso de utilizar un amplificador, bastaría uno de 31 dB para que ambas placas logren una potencia similar a 1 W (30 dBm).

Capítulo 6. Pruebas de funcionamiento y optimización

Tabla 6.1: Tiempo promedio que lleva el cambio de canal en cada placa.

	NUC	Raspberry Pi
Tiempo de cambio de canal	82,3 μ s	216,4 μ s

Se realizó una prueba en la que se transmitieron 20 mensajes y se observaron los canales de frecuencia en los que los mensajes fueron transmitidos. Se obtuvo que 8 mensajes se transmitieron únicamente en el canal correspondiente y un mensaje se transmitió en el canal incorrecto. Los otros 11 mensajes se transmitieron en ambos canales a la vez. Esto demuestra que el mecanismo de cambio de canal no funciona de forma adecuada todo el tiempo. Se cree que esto se debe a que el bloque `Selector_39` no funciona correctamente. La principal consecuencia de este ocasional mal funcionamiento es que se ocupan ambos canales de transmisión, potencialmente interfiriendo con transmisiones ajenas.

6.3.3. Prueba 3: Tiempo de actualización del umbral

Se verificó que un minuto fuera tiempo suficiente para la actualización del umbral dinámico. Esta prueba se ejecutó en las dos placas consideradas en el proyecto.

La prueba consistió en agregar una instrucción de impresión en el momento en que el umbral cambia desde el valor mínimo con el que se inicializa. Se obtuvo como resultado que un minuto es suficiente para la actualización del umbral en la placa NUC, cumpliendo el tiempo establecido por la norma ITU-R M.1371-5. Sin embargo, esta cota no se cumple en la placa Raspberry Pi. Si además de tener la instancia del transmisor, se ejecuta una instancia del receptor de *gr-ais*, se requiere de 1 minuto 34 segundos en promedio para el cambio del umbral. Por otro lado, si no se tiene una instancia de recepción de mensajes, se requiere de 1 minuto 4 segundos en promedio para el cambio del umbral. En la placa NUC, un minuto es suficiente para ambas situaciones.

Esta diferencia se debe una vez más a que la Raspberry Pi es más lenta en procesamiento que la placa NUC. En consecuencia le llevará más tiempo detectar cambios en el canal. Una posible solución sería tomar las muestras de a 20 para determinar la potencia actual en vez de cada de cada 10. Si bien esto aceleraría el tiempo de procesamiento, sería más impreciso pues se estaría asumiendo que una muestra es representativa de las siguientes 19.

6.3.4. Prueba 4: Intervalos de reporte

Se midieron los intervalos de reporte entre mensajes transmitidos, buscando contrastarlo con los valores esperados (presentados en la tabla 2.2, para una embarcación en movimiento⁶). Además, se buscó tener valores de cuántas transmi-

⁶Se utilizó información de GPS falsa con velocidades altas. Esto permitió transmisiones de información dinámica más cercanas en el tiempo.

6.3. Pruebas de funcionamiento y análisis de resultados

Tabla 6.2: Resultados de prueba de funcionamiento para verificar intervalos de reporte.

	NUC	Raspberry Pi
Cantidad de veces que se saltea una transmisión o la resta de un minuto	Se saltó una resta de un minuto del temporizador 24. Se omitió un mensaje 24 A porque llegó un mensaje 18 y fue priorizado.	Se saltó cuatro transmisiones del mensaje 18. Se saltó cuatro restas de un minuto del temporizador 24 A y una resta del temporizador 24 B.
Promedio de tiempo entre transmisiones del mensaje 18	26 segundos	26,34 segundos (sin considerar intervalos de pérdidas), 35,58 segundos (considerando intervalos de pérdidas)
Promedio de tiempo entre transmisiones del mensaje 24 A	6 minutos, 55 segundos	9 minutos, 40 segundos
Promedio de tiempo entre transmisiones del mensaje 24 B	55 segundos	1 minuto, 55 segundos

siones se omitían por procesamiento lento de la placa. Esta prueba se ejecutó en las dos placas consideradas en el proyecto.

La prueba consistió en la transmisión de mensajes durante 15 minutos. Este tiempo debería ser suficiente para observar treinta mensajes de tipo 18 y dos mensajes 24 (con partes A y B). Para la obtención de los tiempos entre transmisiones, se observó un temporizador en el que se marcó cada vez que se transmitió un mensaje. Es importante recordar el mecanismo de elección de los slots candidatos en transmisión. La norma indica que, una vez se considera el slot central de la transmisión, se deben elegir de forma aleatoria 10 slots candidatos que se encuentren a 187 slots de distancia o menos del slot central. Dado que se tienen 2.250 slots por minuto, 187 slots equivalen a 4,9 s. Esta prueba fue realizada sin otros transmisores cerca, por lo que el transmisor pudo transmitir siempre en el primer slot candidato. Los resultados muestran que, en general, se transmite unos 5 s antes del tiempo esperado, debido al mecanismo de elección de slots.

En la tabla 6.2 se presentan los resultados obtenidos. Se analiza en los siguientes párrafos la situación en cada placa.

En el caso de la placa NUC, se observaron los treinta mensajes 18 esperados y dos mensajes 24 completos. Los promedios de tiempos entre transmisiones de estos mensajes son los presentes en la tabla 6.2. Para el caso del mensaje 18, los tiempos son correctos. Para el caso del mensaje 24 A, ambas veces ocurrió que se omitió una resta de un minuto al temporizador asociado. Esto resultó en que la

Capítulo 6. Pruebas de funcionamiento y optimización

transmisión se demore un minuto extra. El diseño del código fue realizado teniendo en mente que se podrían procesar todas las muestras de todos los slots. Si, por algún motivo, algún slot no es considerado, puede ocurrir que se pierda una resta de un temporizador o que se omita una transmisión. Esto no suele ocurrir en la placa NUC. Ocurrió una vez durante la prueba, para el mensaje 24. La otra pérdida se debió a que se debía transmitir un mensaje 18 y un mensaje 24 casi al mismo tiempo, y el transmisor implementado da prioridad al mensaje 18⁷. Sin embargo, notar que al restarle un minuto a los intervalos se recuperan valores correctos.

En el caso de la placa Raspberry Pi, se observaron veintiséis mensajes 18 y un mensaje 24 completo. La pérdida de mensajes se debe a que el procesador es más lento y es común perder más muestras y slots que con la placa NUC. En la tabla 6.2 se puede ver que los intervalos entre los mensajes 18 transmitidos llevan a un promedio de 35,58 s al considerar los cuatro intervalos de pérdidas (que pasan a ser de unos 25 s a 1 minuto y 25 s). Si se descartan dichos valores, el promedio de intervalos de reporte del mensaje 18 es 26,34 s. Para el mensaje 24, los valores obtenidos son muy lejanos a los correctos, debido a varias pérdidas de restas del temporizador.

En general se obtuvieron los intervalos de reporte esperados para las transmisiones en la placa NUC, pero no en la placa Raspberry Pi. La principal consecuencia de esto es la posibilidad de que embarcaciones cercanas tengan información desactualizada. Esto ocurre al perderse mensajes con información dinámica sobre la ubicación del transmisor implementado o sobre sus planes de movimiento. La pérdida de mensajes 24, con información estática, no debería resultar en problemas significativos de seguridad marítima.

6.3.5. Prueba 5: Transmisión de mensajes interpretables

Se evaluó la eficacia del transmisor a la hora de enviar sus mensajes. Esto es, de todas las veces que se imprime en terminal que un mensaje fue enviado, cuantas veces dicho mensaje es visible en el espectro. Además, se verificó cuántas veces el mensaje puede recibirse por un receptor AIS. Se utilizó el aire como medio de transmisión.

Para esta prueba se utilizó la herramienta *SDR++* que permite visualizar el espectrograma en la frecuencia sintonizada. Como receptor AIS se utilizó *gr-ais* sin ninguna modificación.

Tanto para la placa NUC como para la Raspberry Pi se corrió el transmisor implementado hasta que transmitiera 20 mensajes. Cada vez que se imprimía en terminal que un mensaje era enviado, se miraba en *SDR++* si la transmisión era efectiva. A su vez, se miraba en otro equipo con *gr-ais* si el mensaje era recibido.

En la placa NUC, de los 20 mensajes que se enviaron según el transmisor implementado, 19 fueron visualizados en el espectro. De esos 19 que fueron efectivamente enviados, 7 fueron recibidos por el receptor *gr-ais*. En la Raspberry Pi,

⁷Esta convención fue fijada por el grupo de trabajo. Se consideró de mayor importancia transmitir información del movimiento de la embarcación, por sobre información estática de identificación.

6.3. Pruebas de funcionamiento y análisis de resultados

de los 20 mensajes, 18 fueron visualizados en el espectro de los cuales 8 fueron recibidos por *gr-ais*. Estos resultados podrían implicar una buena eficacia por parte del transmisor, que transmite el 90 % (o más) de las veces que debe hacerlo. Sin embargo, el hecho de que se observe algo en el espectro no implica que se esté transmitiendo lo indicado, o de forma correcta. En particular, se detectó que la recepción no es buena. Los mensajes fueron interpretados por el receptor menos del 40 % de las veces. Se consideró relevante investigar posibles fuentes para este funcionamiento.

En primer lugar se evaluó el desempeño de *gr-ais*, utilizando el transmisor *gr-aistx*. Esto aísla del problema al código desarrollado para el proyecto. El transmisor *gr-aistx* envía mensajes fijos como strings en un bloque `Message Strobe`. No se implementa el acceso al medio, ni la generación de mensajes con información real. Se enviaron 20 mensajes con el archivo de ejemplo de *gr-aistx* y se evaluó cuántos mensajes son recibidos por *gr-ais*. Se observaron en el espectro 17 de los mensajes transmitidos, y solamente 7 fueron recibidos. Esto indica que el problema puede tener distintas fuentes: el receptor utilizado (*gr-ais*), la generación de la trama o su transmisión. Así, se confirma que la fuente de error no está en los bloques desarrollados en el proyecto.

La siguiente verificación consistió en utilizar otro receptor de código libre, AIS catcher [27]. Este proyecto está escrito en código C++, tiene una comunidad más establecida y está más documentado que *gr-ais*. No fue considerado para el trabajo porque no estaba basado en GNU Radio. Se enviaron nuevamente 20 mensajes y el receptor detectó 8 de ellos. Se probó también transmitir con *gr-aistx* y recibir con AIS catcher, obteniendo 7 mensajes de 20 en recepción. Estas pruebas indican que el problema no está en el receptor utilizado.

La posible fuente de error se encuentra entre el momento en que la trama es generada y enviada. No se cree que los bloques de modulación GMSK o pasaje al canal de frecuencia indicado sean problemáticos, pues están implementados con GNU Radio. Se considera que el error puede generarse en el bloque `Build AIS Frame`, en el bloque `Selector_39` o en el bloque `Pluto Sink`. También se considera que el error puede generarse en buffers de todo el sistema.

Se realizó una prueba para analizar si el bloque `Selector_39` generaba el problema. Quizá al cambiar el canal sintonizado quedaban muestras restantes del flujo de bits del otro canal. En la prueba se quitó el bloque, transmitiendo siempre en el canal A. Nuevamente, de 20 mensajes transmitidos se recibieron 7. Se descarta entonces este bloque como fuente de error.

Todas estas pruebas extras se realizaron en la placa NUC. Se tiene entonces que el error puede surgir del bloque `Build AIS Frame` o en el bloque `Pluto Sink`. No se profundizó más en este problema. Se tienen algunas ideas para continuar la exploración en el futuro. Por ejemplo, podría tener sentido inspeccionar la forma de onda de los mensajes. Para esto, existen herramientas como Inspectrum [56]. Quizá se podría verificar si se observa algo fuera de lo normal en los mensajes que no son recibidos.

La principal consecuencia de este mal funcionamiento es que el sistema no es confiable desde el punto de vista de la seguridad marítima. Dado que más de la

Capítulo 6. Pruebas de funcionamiento y optimización

mitad de las transmisiones no podrán ser interpretadas de forma correcta, podría implicar que otras embarcaciones no conozcan su presencia.

6.3.6. Prueba 6: Contenido de los mensajes

En esta prueba se verificó la coherencia de los mensajes enviados. Esto significa verificar que su contenido tiene información correcta en cuanto a ubicación, velocidad, dirección y curso.

Para lograr el objetivo se tuvo el mismo procedimiento que en la prueba anterior, donde se corrió el transmisor implementado hasta que enviara 20 mensajes. Esto se hizo para la placa NUC y para la Raspberry Pi. Por otro lado se tuvo un equipo con *gr-ais* para poder recibir los mensajes.

Independientemente de la cantidad de mensajes interpretados por *gr-ais*, lo que se quiso ver es cuántos de ellos contaban con un contenido correcto. Para esto se tomaron los mensajes recibidos por el receptor, que son impresos en formato NMEA en la terminal. Se hizo la traducción de todos ellos en un traductor en línea⁸ y se observó que cada campo tuviera el contenido adecuado.

Se observó que tanto para la Raspberry Pi como para la placa NUC, todos los mensajes interpretados contaban con campos coherentes. Lo único que llamó la atención en esta prueba es que la velocidad no era nula a pesar de estar en reposo. Lo que se pudo ver es que la antena GPS no arrojaba valores nulos incluso cuando no tenía ningún tipo de movimiento. Los mensajes son entonces codificados de forma correcta de acuerdo a la información que reciben por parte del GPS.

6.3.7. Prueba 7: Acceso al medio compartido

Se buscó comprobar que el mecanismo de acceso al medio implementado funciona de manera correcta. Dado que se utiliza un sensado del canal, se verificó que el transmisor puede detectar cuando otra estación está transmitiendo, y que en dicho caso evitará transmitir. Esta prueba se ejecutó en las dos placas consideradas en el proyecto. Se utilizó el aire como medio de transmisión.

La idea inicial fue verificar que no se transmitiría específicamente en un slot donde alguien está transmitiendo. Sin embargo, esto presentó dos problemas. En primer lugar, los slots candidatos se definen de forma aleatoria, por lo que no se conoce de antemano cuáles slots serán seleccionados. En segundo lugar, se habría necesitado mucha precisión en el encendido y apagado de un transmisor externo. La prueba se realizó entonces a nivel de transmisión.

Se encendió un transmisor externo por 40 s y luego se apagó por 20 s, y esto fue repetido varias veces. De esta manera, se buscó que las transmisiones fallaran por completo, intentando sin éxito en los 10 slots candidatos. Fue necesario transmitir un tono en el canal considerado durante una fracción de un minuto. Esto se debe a que el umbral es dinámico y se hubiera actualizado de haber transmitido durante un minuto completo.

⁸Las traducciones de NMEA para ver el contenido del mensaje se hicieron en el sitio: <https://www.aggsoft.com/ais-decoder.htm>.

6.3. Pruebas de funcionamiento y análisis de resultados

Se utilizó un dispositivo HackRF One de Great Scott Gadgets como transmisor externo [57]. Se transmitió un tono cinco veces en cada canal para simular una transmisión de otra estación por un tiempo prolongado.

Se consideraron diez transmisiones para cada placa, cinco en cada canal. Se observó que con la placa NUC el funcionamiento fue el esperado; todas las transmisiones que debían fallar, fallaron. Se obtuvo un 100 % de éxito en el mecanismo de acceso al medio. Por otro lado, en la placa Raspberry Pi, se tuvo una transmisión por canal que se realizó exitosamente, incluso aunque hubiera otra estación transmitiendo. Esto implica que el 80 % de las veces se realiza un buen acceso al medio compartido. La principal causa de este mal funcionamiento es la demora en el procesamiento de las muestras, que genera que la potencia actual considerada en cierto momento sea en realidad un valor viejo. Así, al compararla con el umbral, se obtiene que es posible transmitir (ya que se trata de una potencia baja, como si nadie estuviera utilizando el canal).

En ambas placas se tiene un funcionamiento correcto la mayor parte del tiempo. Es importante mencionar que todas las transmisiones que no debían fallar, no fallaron. En general, el acceso al medio cuando no hay nadie transmitiendo funciona de manera correcta. Esto es importante porque implica que el prototipo diseñado cumple una de sus principales funciones y porque se cumple entonces uno de los objetivos del presente proyecto.

6.3.8. Prueba 8: Primer acercamiento al retardo de transmisión

Se midió el retardo entre que el bloque `Transmitter` indica que se debe hacer una transmisión, hasta que el mensaje es efectivamente conformado y luego cuando el mensaje es recibido.

Para lograr el objetivo de esta prueba, se agregaron impresiones en terminal enseguida después de que el bloque `Transmitter` hace el pedido de la transmisión y luego de que la trama del mensaje es conformada en el bloque `Build AIS Frame`. Ambas impresiones tienen una referencia temporal teniendo en cuenta el tiempo UTC del equipo. Al receptor se le agregó también una referencia temporal junto al mensaje recibido en su impresión. El sistema implementado se ejecutó hasta que se transmitieron 10 mensajes, registrando todos los valores obtenidos y calculando el retardo total. También se ejecutó el receptor `gr-ais` para poder llevar el registro de cuándo se recibía el mensaje. Los valores registrados se pueden ver en la tabla 6.3.

El promedio del retardo total es de 1,57 s. Si bien este no es un valor ideal, ya que el receptor recibe el mensaje algunos slots después de lo que debería, se consideró un valor bueno para un primer prototipo. Fue de interés refinar este valor. Esto se debe a que las mediciones realizadas no indicaban exactamente cuál era el tiempo de la transmisión sino el tiempo de la recepción. Es claro que el principal problema de este resultado es que no se está transmitiendo en el slot que efectivamente fue sentido. En la tabla 6.3, de cualquier forma, es de interés la segunda columna. La misma muestra que se requiere un tiempo equivalente a casi cuatro slots de tiempo únicamente para generar la trama de salida.

Capítulo 6. Pruebas de funcionamiento y optimización

Tabla 6.3: Tabla con los tiempos en segundos entre cada impresión de referencia temporal.

Número de mensaje	Retardo de generación del mensaje (s)	Retardo de transmisión del mensaje (s)	Retardo total (s)
1	0,105	1,489	1,594
2	0,192	1,587	1,779
3	0,144	1,454	1,598
4	0,108	1,398	1,506
5	0,134	1,460	1,594
6	0,103	1,479	1,582
7	0,167	19,365	19,532
8	0,105	1,202	1,307
9	0,101	1,602	1,703
10	0,103	1,239	1,342
11	0,163	1,529	1,692

Se debe tener en cuenta también que las medidas fueron hechas con un receptor que no es de nivel profesional. Además, por los valores observados en la tabla 6.3, el retardo más importante se da entre que el mensaje es efectivamente enviado y se imprime en terminal el mensaje en sí. Esto da un indicio de que la conformación de la trama no es el lugar más problemático para el retardo. Como los últimos bloques para la transmisión (luego de generada la trama) son bloques propios de GNU Radio (*GMSK Mod* y *Pluto Source*), se cree que el retardo más importante se concentra en el procesamiento de *gr-ais*.

Continuando con los resultados de la prueba, se observa en la fila 7 de la tabla 6.3 que hay un valor muy elevado de retardo, de unos 19 segundos. Dicho valor no se consideró para el promedio ya que se tomó como un outlier. Esto es porque el valor es más de 10 veces mayor que el resto de los valores e incluso fue un mensaje que no se pudo interpretar. Se cree que alguna transmisión falló (razón por la que se imprimieron las primeras dos referencias temporales) y unos segundos después se recibió un mensaje corrupto de alguna embarcación cercana. Por esta razón, cuando el mensaje quiso interpretarse desde un decodificador de NMEA, no se pudo ver ningún mensaje válido.

6.3.9. Prueba 9: Retardo de transmisión

Se realizó una prueba más elaborada para identificar el momento en el que se transmite. Esta prueba se realizó en la placa NUC.

Se utilizó un osciloscopio para observar la señal PPS del módulo GPS y la señal de la antena del ADALM-PLUTO. Un canal del osciloscopio se conectó al pin PPS del módulo GPS. El otro canal se conectó directamente a la antena de transmisión. Las transmisiones debieron realizarse en una frecuencia central menor a la usual, debido a que el ancho de banda del osciloscopio utilizado es de 100 MHz.

6.3. Pruebas de funcionamiento y análisis de resultados



Figura 6.1: Captura de la pantalla del osciloscopio donde se observan dos transmisiones. Se señala una de ellas en verde.

Considerando que el ADALM-PLUTO puede transmitir como mínimo señales de 70 MHz, se centraron las transmisiones en 82 MHz.

Fue posible observar en la pantalla ambas señales e identificar transmisiones. En la figura 6.1 se señala una transmisión, caracterizada por un instante de mayor amplitud en la señal de la antena. Es importante mencionar que se verificó en la terminal que se hubiera impreso un mensaje indicando la transmisión, para comprobar que se envió algo. Habiendo identificado las transmisiones y con el entorno de trabajo preparado, se procedió a la prueba.

El objetivo fue medir el tiempo entre que el mensaje debe ser enviado (se genera la orden de enviar en el bloque `Transmitter`) y que se observa la señal en el osciloscopio. Para saber a cuál segundo UTC se refería cada pulso PPS se controló el tiempo UTC en un celular y se marcó en el osciloscopio el pulso asociado al segundo cero de un minuto. Luego se contó la cantidad de pulsos hasta el pico en la señal de la antena y se aplicó un zoom para mayor precisión. Así, fue posible identificar en qué momento se observa la transmisión en relación a UTC. Con respecto al momento de la orden de transmisión, el tiempo UTC se imprime en la terminal junto con la orden.

En la tabla 6.4 se presentan los resultados de la prueba. Se observa que efectivamente el promedio del retardo de transmisión entre la orden de transmitir y la transmisión es de unos 127 ms. Este tiempo es mejor que el obtenido en la prueba descrita 6.3.8, lo que indica que el mayor retardo está en el procesamiento en recepción. El tiempo obtenido implica casi 5 slots de tiempo. A pesar de no ser inmediato al slot correcto, no es un retardo que se considere malo para un primer prototipo.

En la tabla 6.4 también es interesante ver el tiempo entre que la trama fue generada y que la misma es transmitida. Este tiempo es en promedio 24,6 ms. El hecho de que sea significativamente menor podría permitir estrategias como tener

Capítulo 6. Pruebas de funcionamiento y optimización

Tabla 6.4: Tabla con los tiempos en segundos entre cada impresión de referencia temporal.

Número de mensaje	Retardo de generación del mensaje (s)	Retardo de transmisión del mensaje (ms)	Retardo total (s)
1	0,101	24	0,125
2	0,104	30	0,134
3	0,101	25	0,126
4	0,101	19	0,120
5	0,107	25	0,132

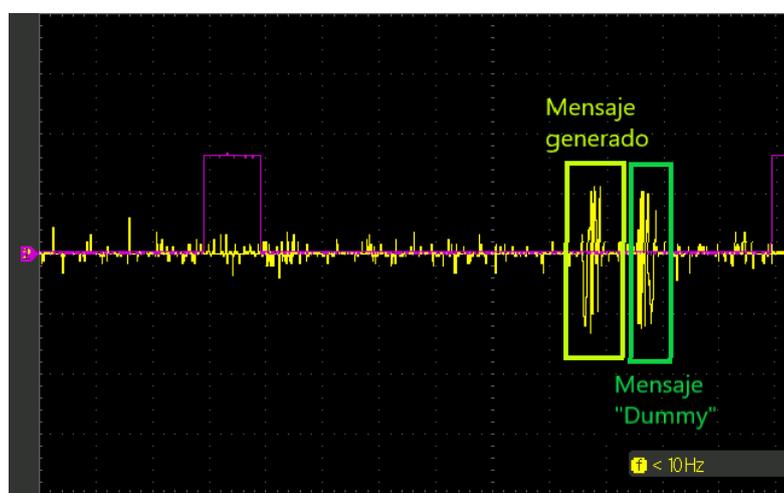


Figura 6.2: Acercamiento a las señales en el osciloscopio en el momento de una transmisión. Se observan dos mensajes transmitidos: en primer lugar el mensaje generado, y luego un mensaje “Dummy”. La escala utilizada no se aprecia en la imagen, pero cada recuadro en el eje temporal es de 100 ms.

los mensajes casi generados por completo y agregarles solo información dinámica al momento de transmitir. Esto disminuiría el retardo entre el momento en que se debe transmitir y el momento en el que se transmite.

Más allá de los tiempos obtenidos, existen otros dos aspectos curiosos observados en esta prueba. En primer lugar, como se ilustra en la figura 6.2, es posible observar dos mensajes en cada transmisión. El segundo mensaje se debe a la presencia del “Dummy” generado. Efectivamente se observa su presencia en el canal, confirmando que es necesaria una mejor solución para el problema a partir del que surgió este mensaje. Además, la escala del osciloscopio permite medir aproximadamente la duración de las transmisiones. Se obtuvo que en promedio cada transmisión dura 40 ms, tiempo que debe ser contrastado con los 27 ms de duración de un slot. Al tener acceso al valor, se considera que es bastante cercano a lo que debería ser, más allá de que sea necesario seguir trabajando sobre él.

En este capítulo se analizó el prototipo implementado. Se describieron las modificaciones realizadas para obtener un mejor funcionamiento. Se presentaron tam-

6.3. Pruebas de funcionamiento y análisis de resultados

bién las pruebas realizadas, así como los resultados obtenidos. Se realizó un primer análisis de los resultados y se evaluaron posibles causas y consecuencias.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 7

Conclusiones

Una parte importante de todo proyecto es su final. Los recursos humanos suelen tener límites de tiempo para el trabajo sobre un cierto tema. En el caso particular del proyecto de fin de carrera, se trabajó durante un año y dos meses para resolver el problema descrito en la presente documentación. Se lograron avances interesantes y destacables. Estando cerca del límite temporal para la conclusión del trabajo, se debe dar cierre al desarrollo, tomar un paso hacia atrás y reflexionar sobre el camino recorrido.

Este capítulo se divide en tres partes. En primer lugar se realiza un análisis del trabajo. Se comparan los objetivos iniciales con los logros obtenidos. En una segunda sección se describe el trabajo a futuro. Este proyecto funciona como un primer prototipo de un sistema transmisor de la tecnología AIS. Pensando en la posibilidad de continuar el trabajo en los próximos años, ya sea por parte del mismo equipo o por nuevos participantes, es importante dejar claro qué aspectos son candidatos a mejoras. Finalmente, se cierra la documentación del proyecto indicando las contribuciones realizadas. Se resaltan las novedades del trabajo y cómo estas han impactado o continuarán impactando a la comunidad.

7.1. Reflexiones

Tal como se expresó en el capítulo 1, se definieron objetivos para el presente proyecto. Una vez concluido el trabajo, es importante determinar cuáles objetivos se cumplieron completamente. También es importante determinar aquellos que se cumplieron de forma parcial y si hay alguno que no se pudo cumplir. Este análisis permite evaluar el funcionamiento del sistema implementado y qué requerimientos cumple. Los objetivos del proyecto serán evaluados a través de los criterios de éxito definidos en el plan de proyecto. Esto permite evaluar el trabajo realizado y poder determinar si, en rasgos generales, los resultados obtenidos son buenos. Se tienen cinco criterios de éxito. Es sencillo relacionarlos con la documentación escrita, como principal prueba de cumplimiento total o parcial de ellos.

En primer lugar se buscó comprender los puntos fundamentales de la norma ITU-R M.1371-5, que describe la tecnología AIS considerada. Esto fue realizado

Capítulo 7. Conclusiones

durante los primeros meses del proyecto, quedando reflejado en el capítulo 2. Se comprendió el funcionamiento de las estaciones de clase B, así como las principales diferencias entre estaciones de clase SO y CS. Fue posible decidir cuál de las dos era más adecuada para el proyecto a partir de la generación de simulaciones. Esto demuestra que los mecanismos fueron comprendidos con profundidad y fueron manejados de forma correcta a lo largo del proyecto.

El segundo criterio está asociado a la capacidad de acceder al medio cuando este sea compartido, desarrollado en el capítulo 4. Se describió la implementación de un mecanismo de sensado del canal y se comprobó el funcionamiento del mismo a partir de pruebas, en el capítulo 6. En particular, se logró acceder al medio compartido de forma correcta el 100 % del tiempo con la placa Intel NUC y el 80 % del tiempo con la placa Raspberry Pi. Ambas detectaron la presencia de otras estaciones transmitiendo y fijaron slots como ocupados, evitando su utilización. Dentro de este criterio se encuentra también la posibilidad de transmitir, recibir e interpretar mensajes externos. La transmisión es posible, pero no se cumplen ciertas restricciones de tiempo impuestas por la norma. La recepción se agregó como un módulo aparte utilizando *gr-ais*. La interpretación de mensajes recibidos fue caracterizada como una posible línea de trabajo a futuro, como se describe más adelante en este capítulo.

Otro criterio establecido se relaciona a la capacidad de codificar y generar mensajes que puedan ser correctamente interpretados. Esto fue descrito en el capítulo 5. Se generaron y codificaron los mensajes relevantes para estaciones de clase B CS. Como se comprobó en las pruebas del capítulo 6, los mensajes generados fueron interpretados de manera correcta por otros receptores de la tecnología. Además, los campos de los mensajes fueron completados tomando información de un sensor GPS local, transmitiendo datos confiables y realistas.

El siguiente criterio de éxito busca que el envío de mensajes se haga de acuerdo a lo establecido en la norma ITU-R M.1371-5. En particular, que el contenido y su intervalo de transmisión sea correcto. La comunicación con el GPS local permite la obtención de parámetros necesarios para conformar los mensajes y determinar sus intervalos de reporte. La información se codifica y se observa en el espectro en los intervalos establecidos por la norma. En este punto se debe aclarar que el objetivo se cumple de forma parcial. Si bien se observa en el espectro el envío de los mensajes, estos son recibidos por un receptor AIS un 40 % de las veces. Sobre este aspecto se debe trabajar para lograr un sistema que pueda ser utilizado en un caso real pero se identificaron posibles fuentes del problema y donde continuar buscando el error.

El último criterio de éxito hace referencia a la implementación de un sistema completo, que cumpla con todos los requisitos previos y englobe los módulos ya revisados. Esto se cumplió satisfactoriamente de forma parcial. Si bien el sistema logra acceder al medio y codificar los mensajes, existen aspectos sobre restricciones de tiempos y envíos de mensajes que deben pulirse. Para lograr un sistema transmisor AIS de clase B completo, es necesario dedicarle tiempo extra a algunos aspectos. Estos puntos fueron identificados en la documentación, donde se remarca cuáles son las puntas a atacar en cada uno.

7.2. Trabajo a futuro

Los objetivos planteados fueron cumplidos, algunos de forma parcial. Un aspecto no menor es que fueron cumplidos según el cronograma y la planificación definidos al inicio del proyecto. Se logró mantener el nivel de trabajo durante el año transcurrido, llegando a los hitos del proyecto con los avances indicados en un inicio. Esto permite ver que el desarrollo del proyecto fue acorde a lo esperado.

Es importante realizar un último análisis para evaluar la completitud del proyecto realizado. Uno de los principales objetivos consistió en desarrollar un sistema que pueda ser categorizado de bajo costo. Es de relevancia entonces verificar que los componentes del sistema transmisor implementado resultan en precios inferiores a los disponibles en plaza (presentados en el apéndice A).

El sistema transmisor implementado consiste en los siguientes elementos de hardware: una placa para utilizar como computadora, un ADALM-PLUTO, un módulo GPS y una antena activa de GPS. Además, para lograr la recepción de mensajes, será necesario un RTL SDR extra. Se presentan en la tabla 7.1 los costos de cada uno de los dispositivos considerados. Es importante mencionar que solamente el módulo GPS y la correspondiente antena fueron adquiridos por el grupo de trabajo. El resto de los equipos fueron brindados por compañeros del Instituto de Ingeniería Eléctrica.

Se tiene un precio total de 355 dólares para la versión con la placa Raspberry Pi y un precio total de 610 dólares para la versión con la placa NUC. No se consideraron precios de envío a Uruguay de las partes. Los costos no son comparables de forma directa con aquellos de los equipos en plaza pues no se cumple con todas las restricciones de los equipos comerciales. En particular, este prototipo no cuenta con un sistema de impermeabilización y algunos equipos fueron adquiridos fuera de Uruguay, reduciendo algunos costos. Sin embargo, los equipos comerciales suelen ser producidos en masa, lo cual reduce costos. Independientemente de esto, se evaluó dónde se posicionaría este prototipo en el mercado. Con ambas placas se obtiene un total más económico que el de la gran mayoría de los dispositivos AIS en Uruguay. Esto permite confirmar que la implementación realizada cumple con el objetivo de ser de bajo costo, aunque reste considerar algunos aspectos.

7.2. Trabajo a futuro

Como el tiempo de dedicación a los proyectos suele ser acotado, deben plantearse objetivos acordes a él. Muchas veces esto implica establecer prioridades sobre qué características implementar. Así, se tienen aspectos del proyecto que se podrían mejorar en caso de contar con tiempo para ello.

A lo largo del proyecto, se priorizó la funcionalidad general del sistema. Es decir, tener un sensado del canal correcto y en base a él poder codificar y enviar mensajes de forma acorde. Por este motivo, algunos detalles de implementación que no fueran tan relevantes se dejaron de lado. En esta sección se presentan algunas líneas de posible trabajo a futuro, que buscan completar la implementación realizada o mejorar algunos aspectos.

En primer lugar, se parte de lo que la norma establece que debe cumplir toda estación AIS. Esto es:

Capítulo 7. Conclusiones

Tabla 7.1: Costos finales del prototipo implementado. Se indican como referencias las hojas de datos de cada dispositivo.

	Precio
ADALM-PLUTO [45]	230 dólares
Módulo GPS [43] y antena [44]	25 dólares
Placa Raspberry Pi [42]	70 dólares
Placa Intel NUC [41]	325 dólares
RTL SDR [58]	30 dólares
Precio final con placa Raspberry Pi	355 dólares
Precio final con placa Intel NUC	610 dólares

1. El sistema debe enviar a las demás embarcaciones la información dinámica y otra información de la embarcación de forma automática.
2. El sistema debe ser capaz de recibir y procesar los mensajes de interrogación.
3. El sistema debe ser capaz de enviar información adicional asociada a la seguridad cuando se lo pidan.
4. El sistema debe ser capaz de operar de forma continua con la embarcación en movimiento o anclada.
5. El sistema debe usar técnicas de TDMA de forma sincronizada.
6. El sistema debe tener la capacidad de operar en tres modos: autónomo, asignado e interrogación.

Por lo establecido en el documento, se puede concluir que los puntos 1, 4 y 5 se cumplen de forma total o parcial¹. El punto 1 y el 5 se cumplen de forma parcial. Como se estableció en este documento, no todos los mensajes que deben ser enviados se pueden recibir de forma correcta, por lo que el punto 1 se cumple pero no de forma total. El punto 5 también se cumple de forma parcial. Si bien el prototipo logra una buena sincronización con UTC y un correcto sensado del canal, a la hora de enviar los mensajes se tiene un retardo muy elevado, tomando slots extra. Sobre estos aspectos se debe trabajar en detalle para lograr un sistema que se pueda utilizar en un caso real.

Los requerimientos 2, 3 y 6 no se cumplen en este primer prototipo. Esos tres requerimientos están relacionados con lo ya establecido de que el foco no estuvo en poder procesar los mensajes recibidos, sino en el envío de mensajes y en el acceso al medio. Al no procesar los mensajes recibidos para interpretar su información,

¹El punto 5 se cumple con la precisión de nanosegundos requerida únicamente en la Raspberry Pi. Esto se debe a la sincronización UTC a través de la señal PPS.

no es posible contestar interrogaciones (punto 2), ni enviar información adicional cuando se demande (punto 3). De esta forma, el sistema solo puede funcionar en el modo autónomo (punto 6).

De forma similar, en la norma se establece que las estaciones de clase B CS deben poder escuchar las reservas de las estaciones base para no considerar los slots reservados por ellas como candidatos. Este punto tampoco se cumple por la incapacidad del sistema de tomar acción de acuerdo con lo que recibe. Sin embargo, si una estación base comienza su transmisión al principio del slot (tal como debería), el sistema implementado detectará una potencia del canal más alta que el umbral y no transmitirá.

En consecuencia, para poder operar en los tres modos y cumplir con los seis puntos, una de las tareas principales a abordar es la interpretación de los mensajes recibidos. Con un poco de trabajo adicional se podrá también interpretar reservas de estaciones base. Este último aspecto puede ser también un puntapié para implementar un sistema AIS de clase B SO.

A nivel de capa física, la norma describe requerimientos generales de funcionamiento. Los requerimientos principales, como los canales de frecuencia en que se debe operar, la tasa de bits y parámetros de modulación, se describen en la tabla 3 de la norma ITU-R M.1371-5. Se cumple con todos estos requerimientos, a excepción de la potencia de transmisión, que debe alcanzar 1 Watt. Como se explicó en el capítulo 6, no es posible transmitir con esa potencia usando un ADALM-PLUTO, pero podría analizarse la utilización de amplificadores. Más allá de la potencia de transmisión obtenida, se considera que tiene sentido evaluar otros dispositivos SDR para transmisión. Las tablas 5 y 6 de la norma describen otros parámetros de funcionamiento de capa física, como errores en la potencia o en la frecuencia de la portadora, o transmisiones espúreas. Estos aspectos no fueron tenidos en cuenta, ya que sería complejo verificar su cumplimiento. En caso de querer utilizar este sistema en escenarios reales, es necesario realizar las medidas necesarias para evaluar el cumplimiento de dichos requerimientos.

Una funcionalidad no implementada a nivel de capa física consiste en el apagado del transmisor en ciertos casos. La norma establece que si el slot de transmisión termina y se sigue transmitiendo, será necesario dejar de modular la señal enviada. Además, si esta transmisión continuara por más de un segundo luego del final del slot correspondiente, el transmisor debería apagarse de forma automática. Ninguno de estos casos fue implementado. Los motivos para la no implementación son principalmente dos. En primer lugar, se deberían haber manejado interrupciones dentro de GNU Radio, tema que habría tomado tiempo e investigación extra para lograr implementar. Por otro lado, las transmisiones demoran más de un slot transmitirse. Si se implementara algún mecanismo de interrupción de la transmisión, debido a las demoras inevitables del transmisor desarrollado, se corre el riesgo de que nunca se transmitan los mensajes generados. Es claro que la disminución del tiempo de transmisión es un aspecto en el que se debe seguir trabajando.

El proceso de pruebas de la estación es también un punto de mejora. Se buscó realizar una gran variedad de pruebas en poco tiempo, para tener una idea general de la funcionalidad de la estación. Sin embargo, se considera que algunas pruebas

Capítulo 7. Conclusiones

deberían contar con mayor cantidad de iteraciones para obtener conclusiones más fiables. Esto sería posible, entre otras cosas, automatizando las pruebas para poder tener más realizaciones.

Finalmente, en lo que respecta a GNU Radio, no se implementó una interfaz gráfica para los bloques creados. Esto se debió a que el desarrollo del proyecto se basó en esquemas escritos en código. A pesar de que los bloques no requerían una interfaz gráfica para este proyecto, se cree que podría ser de utilidad en el futuro.

La estación implementada no está lista para ser utilizada en ambientes reales. Esto se debe a los aspectos no implementados y al cumplimiento de solamente algunos requerimientos de la norma. Si comenzaran a utilizarse este tipo de estaciones, correspondería analizar el uso del medio compartido. Tendría sentido agregar un caso a las simulaciones realizadas para observar el impacto de una estación CS en el canal. Si se fija la cantidad de estaciones SO en el valor máximo supuesto para el caso de estudio (100 estaciones), es posible aumentar la cantidad de estaciones CS hasta ver cierto porcentaje de pérdidas en el canal. Esto permite tener una idea de la máxima cantidad de estaciones CS que se podrían utilizar para un buen funcionamiento.

7.3. Contribuciones

El presente trabajo describe la implementación de un sistema transmisor para la tecnología AIS. Por un lado se obtuvo una solución de hardware, indicando dispositivos necesarios y justificando precios económicos. Por otro lado, quizá la principal contribución radica en la solución de software obtenida. Se desarrolló un módulo *out-of-tree* de GNU Radio, que quedará disponible en un repositorio de Gitlab². Este contiene la información necesaria para su descarga, instalación y ejecución por parte de cualquier usuario que quiera hacer uso del transmisor implementado. El código desarrollado es de acceso y uso libre, con el fin de aportar a la comunidad en línea. Es importante resaltar que este proyecto no podría haberse realizado si no existieran proyectos previos y de uso libre como *gr-ais* o *gr-aistx*.

A partir de resultados iniciales del presente proyecto, se redactó un breve artículo con el que se participó de la XXX Jornada de Jóvenes Investigadores en Asunción, Paraguay. Se realizó una presentación oral y una presentación de póster. En el evento se presentaron las simulaciones del acceso al medio, donde se analizó el efecto de los distintos tipos de acceso al canal. El trabajo realizado nos hizo acreedores de una distinción por la presentación de póster.

Se escribió también un artículo que fue enviado a consideración para la XIII LANC (Latin American Networking Conference) a realizarse en Bahía Blanca en agosto de 2024. En este artículo se presenta el proyecto de forma más general, donde se enmarcan los resultados finales.

Además de aportar a la comunidad en línea y a la comunidad científica, el proyecto permitió apoyar a la comunidad local. Los conocimientos obtenidos en

²La versión final del repositorio está disponible en <https://gitlab.fing.edu.uy/rominag/itais>.

7.3. Contribuciones

este trabajo se utilizaron para la creación de un proyecto del curso Taller de Introducción a la Ingeniería Eléctrica - Comunicaciones Inalámbricas. Estudiantes de primer año de la Facultad de Ingeniería de la Universidad de la República utilizarán parte del código desarrollado para aprender sobre Python y tecnologías de comunicaciones inalámbricas.

Habiendo escrito el capítulo de conclusiones, se da por terminada la documentación del proyecto “Implementación de un transmisor AIS de bajo costo basado en SDR”. Este trabajo fue realizado en Montevideo, Uruguay, entre marzo del 2023 y abril del 2024.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice A

Estudio del mercado

Se presenta en este apéndice un estudio de las ofertas de estaciones AIS disponibles en el mercado uruguayo. Además, se presenta un valor estimado del costo total de una de estas estaciones, sumando los precios de las antenas u otros agregados que sean necesarios.

Se realizaron búsquedas enfocadas en dispositivos de clase B similares al transmisor a implementar. En varios casos, cuando el anuncio del dispositivo no detallaba el tipo de la estación, se verificó la potencia de transmisión del mismo¹. Se tiene que las estaciones AIS de clase A deben transmitir con potencia de transmisión máxima de 12.5 W. Por otro lado, las estaciones de clase B SO tienen una potencia máxima de transmisión de 5 W, y las estaciones de clase B CS transmiten con una potencia máxima de 2 W. Estas diferencias permitieron identificar qué estaciones estaban siendo consideradas.

Las primeras ofertas evaluadas son las de la empresa Aeromarine. Se trata de una compañía internacional con tiendas en Uruguay que comercializa, entre otras cosas, dispositivos para navegación, describiendo su área de trabajo como “electrónica marina”. Se presentan en la tabla A.1 los dispositivos analizados, junto con su precio y algunas características a detallar. Se observan precios que varían en un rango considerable (entre 460 y 1.500 dólares). Aeromarine vende dispositivos de varias compañías, entre ellas Raymarine y ONWA. En algunos foros de Internet, los usuarios no recomiendan utilizar dispositivos de la marca ONWA, sino que recomiendan usar marcas más establecidas como Raymarine o Garmin [59]. Esto es un factor interesante, dado que los equipos de la marca ONWA son los más económicos que vende la empresa Aeromarine. Es importante resaltar que todos los dispositivos requieren de la compra de una antena VHF para su funcionamiento. Aeromarine comercializa estas antenas, con precios que van entre los 85 y 290 dólares. Para los equipos que no incluyen una referencia de GPS, se debe agregar una antena GPS (que Aeromarine ofrece por 70 dólares). Considerando las recomendaciones de los usuarios de Internet, el equipo más económico y recomendable

¹Notar que a veces no se detallaba la potencia de transmisión del dispositivo en el anuncio de venta en Uruguay. En esos casos, se buscó en Internet la documentación oficial del dispositivo ofrecido. A pesar de que en general tampoco se aclaraba el tipo de estación, sí se detallaba la potencia de transmisión máxima.

Apéndice A. Estudio del mercado

Tabla A.1: Dispositivos AIS de clase B en venta por Aeromarine.

Dispositivos en venta por Aeromarine			
Nombre	Precio (USD)	Tipo	Detalles adicionales
Transceptor Raymarine E32158 AIS650	1.329,80	CS	
Raymarine AIS700	1.500,00	SO	
Transceptor Em-Trak AIS B300	714,00	CS	Ya incluye una referencia de GPS, por lo que no sería necesaria su compra.
Transmisor/Receptor AIS Em-Trak B330	927,00	CS	
GPS/AIS Plotter ONWA KP-38A	710,00	SO	Se incluye un display y una referencia interna de GPS.
AIS ONWA KS-200A	460,00	SO	

Tabla A.2: Estaciones AIS en venta por la empresa Garmin.

Dispositivos en venta por Garmin			
Nombre	Precio (USD)	Tipo	Detalles extra
AIS 800	1.349,00	SO	GPS integrado.

sería el transceptor Em-Trak AIS B300, al que habría que sumarle una antena VHF con costo de al menos 85 dólares. Se tiene entonces que en Aeromarine, la compra de una estación AIS de clase B requeriría un gasto de unos 800 dólares.

Se evaluaron a continuación ofertas de Garmin, también una empresa internacional que distribuye algunos de sus productos en Uruguay. En particular, se ofrece un único modelo de estación AIS, descrita en la tabla A.2. A este precio será necesario sumarle la incorporación de una antena VHF. La misma empresa comercializa un modelo de antena VHF a 123 dólares. El gasto total que se requeriría para una estación con tecnología de Garmin sería de 1.472 dólares.

Habiendo estudiado las dos principales empresas que comercializan estos dispositivos en nuestro país, se presentan en la tabla A.3 otras ofertas de estaciones AIS. Las mismas están publicadas en el sitio web de ventas Mercado Libre. Los anuncios se caracterizan por tener poca información sobre las estaciones, y por garantías a corto plazo. Ambas cosas se consideran poco confiables, teniendo en cuenta que algunos dispositivos presentan precios económicos.

Dada la tabla A.3, se toma como referencia al primer dispositivo, un transceptor AIS de marca Matsutec. Se trata del modelo HP528A, que se menciona en [60] como una opción que ha demostrado funcionar bien. Para un funcionamiento básico, además de comprar el transceptor, será necesaria la compra de una antena VHF. En Mercado Libre es posible encontrar estos productos con un precio que ronda los 100 dólares. Esto implicaría que la compra de una estación AIS de clase B, sin

Tabla A.3: Otros dispositivos AIS de clase B en venta en el mercado uruguayo.

Otros dispositivos en venta			
Nombre	Precio (USD)	Tipo	Detalles extra
Transceptor AIS con GPS Matsutec HP528A	650,00	CS	GPS integrado e incluye una antena GPS. Se ofrece una garantía a corto plazo (seis meses), pero algunos usuarios han reportado buenos resultados de uso [60]. Requiere una antena de VHF con características especiales.
Transceptor NAIS 500	950,00	CS	Incluye antena GPS.
Transceptor AIS Alltek Camino 108	650,00	CS	
Equipo GPS - AIS + Plotter 708A - ONWA	929,00	SO	El equipo AIS está integrado a un display.

considerar a las principales empresas que comercializan esta tecnología en Uruguay, tendría un costo aproximado de 750 dólares.

Es de interés observar que, para los tres casos estudiados, las estaciones clase B SO suelen tener precios más elevados que las estaciones de clase B CS. Esto se debe a varios motivos.

El principal motivo de la diferencia de precios es la cantidad de especificaciones que cada tipo de estación debe cumplir. Los sistemas de clase B SO tienen parámetros y tiempos de funcionamiento más estrictos. Estos aumentan la dificultad del diseño y las garantías que el mismo debe aportar al usuario.

Las estaciones de clase B SO utilizan un mecanismo de acceso al medio compartido basado en la multiplexación por división de tiempo (TDMA, por sus siglas en inglés). En particular, se definió TDMA auto-organizado para su uso en la tecnología AIS. Este mecanismo es considerablemente más complejo que el utilizado en las estaciones CS, ya que requiere una cierta interacción y organización entre las embarcaciones. El grado de dificultad agregado por este mecanismo también es considerado en el precio del producto.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice B

Radio Definida por Software

La tecnología de Radio Definida por Software (SDR, por sus siglas en inglés) es un sistema de comunicaciones introducido en las últimas décadas. La base de los dispositivos SDR consiste en la sustitución de elementos de hardware analógico (filtros, amplificadores, moduladores) por procesos de software [61]. Dichos procesos se implementan en computadoras o como parte de sistemas embebidos.

El término surgió hace alrededor de treinta años [62]. El Foro SDR trabajó en conjunto con el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés) para definir SDR [63]. En primer lugar, se considera como *radio* a cualquier tecnología o dispositivo para la transmisión de información a través de ondas electromagnéticas [64]. Una vez definido el término *radio*, se considera una radio definida por software a cualquier radio en la que alguna o todas las funciones de la capa física están definidas a través de software. Estas funciones incluyen definición de parámetros de radiofrecuencia relevantes (frecuencia central a sintonizar, frecuencia para el muestreo) o de procesos de (de)modulación o (de)codificación de señales.

Previamente, los circuitos para aplicaciones de radiofrecuencia debían ser diseñados de forma específica para ciertas aplicaciones. En el caso de requerir modificaciones, las mismas debían realizarse a través de intervención humana [63]. Esto resultaría en costos más elevados, y riesgos de arruinar los dispositivos al intentar modificarlos. La introducción de dispositivos SDR simplifica el desarrollo de equipos y permite el prototipado de tecnologías nuevas. Además, la detección y corrección de errores en el procesamiento de señales se puede realizar de manera más sencilla a través de software [63].

La tecnología es actualmente muy utilizada en varias áreas. Su importancia está relacionada a la creciente necesidad de la sociedad de comunicarse. En particular, la comunicación se realiza de distintas maneras e involucra cada vez más tipos de dispositivos. La posibilidad de modificar estos equipos de radio de forma sencilla y sin implicar costos adicionales se ha vuelto muy relevante [63].

Distintos avances en semiconductores y dispositivos para el procesamiento digital de señales han permitido que la tecnología encuentre su posición en la industria [62].

Esta página ha sido intencionalmente dejada en blanco.

Apéndice C

Configuración de un servidor NTP

Los sistemas AIS utilizan mecanismos de acceso al medio compartido basados en divisiones de tiempo. Se divide el tiempo en minutos, denominados cuadros o frames. Cada uno de estos cuadros se divide en 2.250 slots de tiempo, cada uno de aproximadamente 27 ms de duración. Las transmisiones de los equipos AIS se organizan en estos slots de tiempo. Es de suma importancia que todas las estaciones AIS tengan la misma referencia temporal, y por tanto sepan cuándo empiezan o terminan los slots de tiempo. Se utiliza la referencia universal UTC para la coordinación entre estaciones.

El sistema transmisor implementado utiliza CSTDMA como mecanismo de acceso al medio. Tanto los procesos receptores, como el proceso transmisor, deben tener acceso a la referencia temporal UTC. En particular, se utiliza la librería *time* de Python y la función *utcnnow()* para acceder al reloj de la computadora local. Se vuelve relevante sincronizar la hora local a UTC. Para esto, es común utilizar módulos GPS que generen una salida de un pulso por segundo (PPS, por sus siglas en inglés) sincronizada con UTC al nanosegundo. La señal de PPS se utiliza para configurar un servidor del protocolo de tiempo de red (NTP).

En este anexo se indica el procedimiento necesario para la configuración de un servidor NTP en una placa Raspberry Pi 4 modelo B. Se tomó como referencia un tutorial en Internet [65]. Se adjuntan los comandos requeridos, con eventuales comentarios que se consideran relevantes para replicar el trabajo en el futuro.

C.1. Requerimientos previos

Se presenta a continuación una lista de los materiales necesarios para la configuración. Es importante contar con conexión estable a Internet.

- Placa Raspberry Pi 4 modelo B. La misma cuenta con una tarjeta SD de 32 GB de memoria.
- Módulo GPS. Es importante que tenga una salida PPS.
- Cable hembra-hembra.

Apéndice C. Configuración de un servidor NTP

- Puesto equipado con monitor, teclado y ratón.

Será necesario conectar la salida PPS del módulo GPS al pin GPIO 18 de la Raspberry Pi. La numeración GPIO no es la que está indicada en la placa, se debe acceder a la documentación oficial para obtener el mapeo de pines [66].

C.2. Configuración del servidor

Se trabajó con una placa Raspberry Pi con un sistema operativo limpio. En particular, se utilizó un sistema operativo de Raspberry Pi basado en Debian GNU/Linux 11 (Bullseye) para procesador ARM 64. La versión del sistema operativo instalada es del 22 de septiembre de 2022. Es importante aclarar que, tal como se menciona en el tutorial seguido, versiones más recientes de Raspberry Pi OS son propensas a errores en la configuración [65].

C.2.1. Habilitación de pines de entrada/salida

Es necesario configurar la placa para poder utilizar pines de entrada/salida. En la terminal, ejecutar `sudo raspi-config`. Seleccionar la opción 3, con opciones de interfaz y luego elegir un puerto serial. Al presentarse la pregunta *Would you like a login shell to be accessible over serial?* responder No. Al presentarse la pregunta *Would you like the serial port hardware to be enabled?* responder Yes. Se debe salir de la configuración y reiniciar la placa para que estos cambios queden efectuados.

Una prueba breve para verificar que es posible acceder al dispositivo es hacer `cat` en el mismo. Para ello, será necesario además tener conectados los pines de transmisión y recepción uART del módulo GPS a los pines 14 y 15 de la placa. Sabiendo el nombre del dispositivo (que suele ser `ttyS0`, o `ttyACM0`), la prueba consiste en ejecutar `cat /dev/ttyACM0` varias veces seguidas. Si en alguna de ellas se ven como resultado mensajes NMEA, es porque el acceso al dispositivo fue correcto, y los pines están configurados.

C.2.2. Descarga de *gpsd*

Para comenzar, actualizar y descargar *gpsd*, una herramienta de software que recibe datos de módulos GPS.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt install gpsd
```

En el archivo `/etc/default/gpsd`, agregar las siguientes líneas.

```
GPSD_OPTIONS=''-n -G''
DEVICES=''/dev/ttyS0''
```

C.2. Configuración del servidor

```
USBAUTO='false'
```

Habilitar *gpsd* y verificar que está funcionando.

```
sudo systemctl enable gpsd
sudo systemctl start gpsd
gpsmon
```

C.2.3. Configuración de PPS

Para poder tener una señal con un pulso por segundo con alta precisión, es necesario habilitar un controlador.

En el archivo */boot/cmdline.txt* se debe agregar al final la línea siguiente: `bcm2708.pps-gpio-pin=18`.

De forma similar, en el archivo */boot/config.txt* se debe agregar al final la línea: `dtoverlay=pps-gpio,gpiopin=18`.

También se debe modificar el archivo */etc/modules-load.d/modules.conf* agregando las líneas `pps-gpio` y `pps-ldisc`.

Luego de modificar estos tres archivos se debe reiniciar el equipo y después instalar el paquete *pps-tools*: `sudo apt-get install pps-tools`.

Se puede verificar si las modificaciones hicieron efecto ejecutando los comandos siguientes:

```
dmesg | grep pps
sudo ppstest /dev/pps0
```

El primer comando debe mostrar una salida que indica que hay una nueva fuente de PPS. El segundo debe arrojar una fuente 0 seguida de valores de *assert* y de secuencia, donde dichos números deben incrementar en uno en cada línea.

C.2.4. Configurando chrony como servidor de tiempo

Se utilizará *chrony* como servidor de tiempo preciso para obtener un NTP. Para esto, se ejecutó el comando `sudo apt-get install chrony`.

Es posible tener errores con permisos de usuarios para *chrony*. Por este motivo, se deben configurar reglas *udev*. En primer lugar, ejecutar el comando `ps aux | grep chronyd` para verificar que el usuario y el grupo del proceso *chronyd* es `chrony`. Se utiliza `/dev/ttyS0` como la conexión serial, aunque el nombre podría ser otro. Se deberá crear un archivo `/etc/udev/rules.d/pps-sources.rules` y agregarle las siguientes líneas.

Apéndice C. Configuración de un servidor NTP

```
KERNEL==``pps0``, OWNER==``root``, GROUP==``_chrony``,  
MODE==``0660``  
KERNEL==``ttyS0``, RUN+=``/bin/setserial -v /dev/%k  
low_latency irq 4``
```

Para que las reglas se apliquen, se puede reiniciar el dispositivo o ejecutar el siguiente comando:

```
sudo udevadm control -- reload-rules && sudo udevadm trigger.
```

La configuración de `chrony` puede estar ubicada en distintos lugares del sistema de archivos, según la versión utilizada. Las opciones más comunes son las siguientes: `/etc/chrony/chrony.conf` o `/etc/chrony.conf`. Se debe agregar al final las siguientes líneas.

```
refclock PPS /dev/pps0 lock GPS  
refclock SHM 0 refid GPS precision 1e-1 offset 0.01 delay  
0.2 noselect
```

Habilitar e iniciar `chrony`.

```
sudo systemctl enable chrony  
sudo systemctl start chrony
```

A continuación se trabajará dentro de `chrony`. Para acceder a la terminal de `chrony`, se debe escribir `chronyc` en la terminal de comandos. Escribiendo el comando `sources` será posible visualizar las fuentes que `chrony` está utilizando para la sincronización. En particular, se indicará con un símbolo de numeral y un asterisco (`#*`) cuál es la fuente seleccionada. Si la fuente seleccionada es la señal de PPS, el servidor NTP quedó configurado por completo.

Existe la posibilidad de que el *offset* temporal sea considerable. Esto se puede observar dentro de la consola de `chrony`, escribiendo el comando `sourcestats`. Si el *offset* de GPS es mayor a 200 ms, la señal PPS no podrá sincronizarse al GPS. Es posible arreglar esto. Se debe esperar algunos minutos a que el valor del *offset* indicado en `sourcestats` se estabilice, y luego modificar la configuración de `chrony`. Se cambia el *offset* por el valor hallado luego de la estabilización en lugar de 0.01 y luego se debe reiniciar el servicio de `chrony`.

Apéndice D

Calibración con el analizador de espectro

En este apéndice se describe con más detalle el proceso de calibración del dispositivo ADALM-PLUTO llevado a cabo con el analizador de espectro. En particular, se describirán los parámetros de mayor relevancia y en qué valor fueron fijados en el equipo, teniendo en cuenta que el procedimiento seguido para lograr el objetivo ya fue explicado y esquematizado en el texto principal.

Para determinar el valor real de la potencia medida por el receptor de radio frecuencia (RF), se utilizó un analizador de espectro. Este cuenta con un adaptador de tipo N a SMA en su entrada “RF input”, donde se colocará la entrada que suministrará la señal cuya potencia se desea medir. En otras palabras, el analizador de espectro desplegará el valor real de la potencia, lo que idealmente mediría el ADALM-PLUTO. Para diversos valores se mide la potencia en ambos dispositivos y es posible estimar entonces el error de nuestro transmisor para corregirlo.

Se mencionan a continuación algunos parámetros con los que cuenta el analizador de espectro que son relevantes para realizar las mediciones.

- **Resolution bandwidth:** Determina el ancho de banda de la resolución a mostrar. Una forma más gráfica de entenderlo es que es como si fuera un filtro que se mueve en el espectro para ir mostrando los valores que ingresan en la entrada. Está directamente relacionado con la velocidad de refresco de la curva que se despliega en pantalla. Si es muy pequeño el valor, el “filtro” de refresco será pequeño y precisará más tiempo en actualizar todo el espectro que se muestra en pantalla (tiene que recorrer toda la frecuencia que se despliega). A su vez, como el filtro es pequeño la precisión será más grande. Por otro lado, si el valor es grande, la resolución será menor pero logrará un refresco más rápido. En general se inicia con un valor alto y cuando se logra algo más o menos estable se baja el valor.
- **Span:** Este valor indica cuánto del espectro se despliega en la pantalla, es decir, el ancho en Hz de lo que se verá.
- **Integration bandwidth:** Determina el ancho de banda considerado para

Apéndice D. Calibración con el analizador de espectro

integrar. Es decir, este valor determinará el ancho, centrado en la frecuencia central, que se integra de la curva para dar un valor de la potencia en el canal.

- **Center Frequency:** Frecuencia central del espectro que se despliega en pantalla.

Dada la topología ya descrita que se puede observar en la figura 4.4 y la definición de los parámetros, se fijó el valor de estos en el analizador de espectro. En particular, se buscó que el ancho de banda de integración (Integration Bandwidth) fuera igual que el ancho de banda del filtro utilizado en el receptor que formará parte del sistema transmisor. Como este es de 22 kHz, se fijó ese mismo valor para el ancho de banda de integración. Luego, la frecuencia central en el analizador de espectro se fijó en 162,025 MHz, que coincide con uno de los canales AIS y por ende nuestro receptor recibirá muestras en esa frecuencia también. Luego, para tener una mayor precisión en el espectro observado y también en la medida, se fijó el resolution bandwidth más chico posible, con un valor de 10 Hz. Para evitar tener medidas erróneas o desactualizadas, se esperaba un tiempo mayor a un minuto para considerar como válida la medida de potencia del canal arrojada por el analizador de espectro. El span se fijó con un valor de 50 kHz, permitiendo observar todo el espectro considerado para la medición de la potencia.

Por otro lado, se debió configurar el transmisor con parámetros coherentes a lo mencionado sobre el analizador de espectro y en el ADALM-PLUTO. De lo contrario, no sería posible obtener mediciones, o las obtenidas no serían representativas de la realidad. Por ende, se transmite con una frecuencia central de 161,925 MHz un tono con frecuencia de corte de 100 kHz. Esto provocará que el tono que recibirán el analizador de espectro y el ADALM-PLUTO esté en una frecuencia de 100 kHz por encima, es decir, en la frecuencia central de uno de los canales AIS (162,025 MHz). Además, el transmisor HackRF está fijado con una ganancia de 50 dB, valor que no se restó de los resultados obtenidos ya que afecta de igual manera al analizador de espectro y al ADALM-PLUTO.

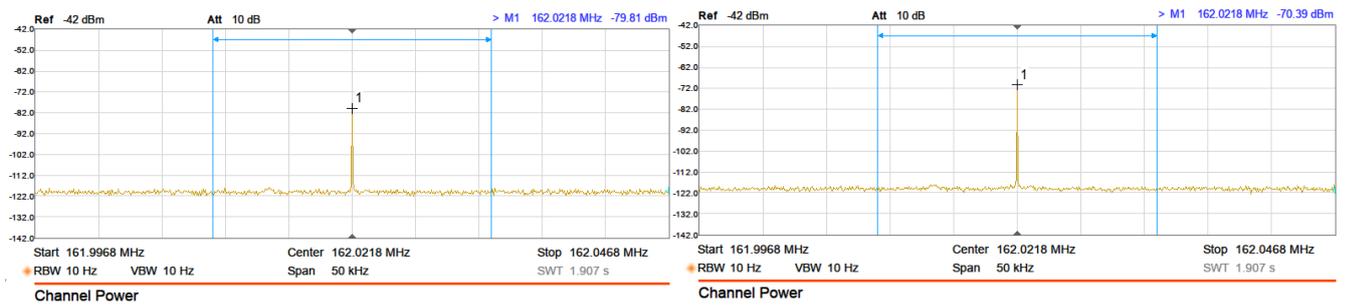
En cuanto al receptor utilizado, ya se había tenido en cuenta el hecho de que el filtro que utiliza tiene un ancho de banda de 22 kHz a la hora de fijar el integration bandwidth. Además, para el cálculo de los umbrales y de la potencia actual del canal se utiliza el bloque `Potumbral`. Al mismo se le agregan estructuras para guardar los valores de potencia relevantes. Se tomó el promedio de la potencia actual del canal durante 1 minuto y medio como medida. Este valor es el comparado con la medida arrojada por parte del analizador de espectro, dando lugar también a que este se ajuste de forma correcta ya que tiene un resolution bandwidth pequeño.

Con todos los instrumentos conectados de forma correcta y sus parámetros fijados como se especificó, se tomaron varias medidas. El tono del transmisor fue variando su amplitud desde 0,05 hasta 1, aumentando 0,05 entre cada medición. Esto dio como resultado los valores que se observan en el cuadro D.1. En particular, se ve en las sub figuras D.1a, D.1b, D.1c y D.1d los valores obtenidos en el analizador de espectro para algunos valores de amplitud.

Tabla D.1: Valores de potencia obtenidos para el analizador de espectro y ADALM-PLUTO.

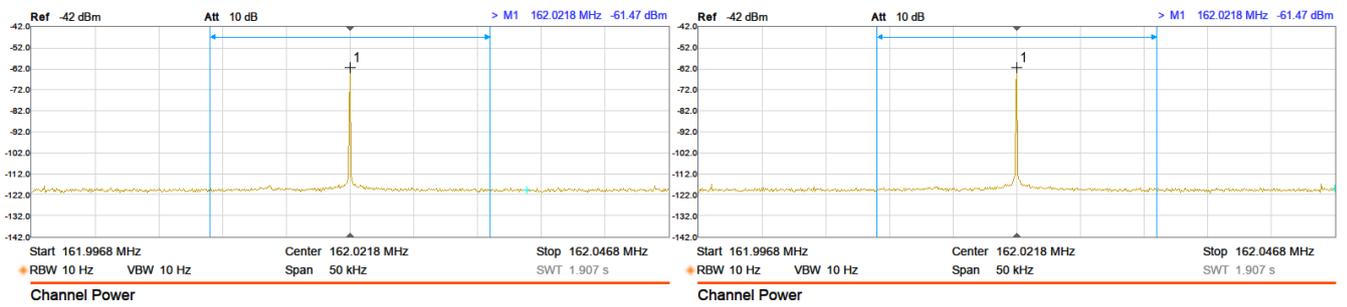
Amplitud	Analizador de espectro (dBm)	ADALM-PLUTO (dB relativo)
0,05	-77,21	-64,79
0,1	-71,47	-58,77
0,15	-68,09	-55,34
0,2	-65,66	-52,77
0,25	-63,70	-50,94
0,3	-62,15	-49,38
0,35	-60,83	-48,06
0,4	-59,66	-46,89
0,45	-58,66	-45,78
0,5	-57,75	-44,98
0,55	-55,93	-44,18
0,6	-56,19	-43,43
0,65	-55,51	-42,75
0,7	-54,87	-42,00
0,75	-54,28	-41,53
0,8	-53,73	-40,98
0,85	-53,21	-40,35
0,9	-52,73	-39,99
0,95	-52,27	-39,52
1	-51,83	-38,97

Apéndice D. Calibración con el analizador de espectro



(a) Amplitud de 0,1.

(b) Amplitud de 0,25.



(c) Amplitud de 0,85.

(d) Amplitud de 1.

Figura D.1: Potencia medida por el analizador de espectro para diferentes valores de amplitud en el transmisor.

Referencias

- [1] Munoz, R. *The Evolution of Communication through the Centuries*, 2017.
- [2] La Rocca, F. *Introducción (y algo de historia)*. Curso de Comunicaciones Digitales, Facultad de Ingeniería, Udelar., 2022.
- [3] National Geographic España. *Por qué la primera llamada de auxilio del Titanic no fue una señal de SOS*. Recuperado de <https://www.nationalgeographic.es/historia/2020/05/por-que-la-primera-llamada-de-auxilio-del-titanic-no-fue-una-senal-2020>.
- [4] H. M. Stationery Off., Harrison and Sons, Printers. Text of the Convention for the Safety of Life at Sea. <https://archive.org/details/textofconvention00inte>, 1914.
- [5] Back, G., Thompson, G. *Military communication - World War II and after.*, 2020.
- [6] Kazuhiko Hasegawa, Kojiro Hata, Kazuhisa Niwa, and Junji Fukuto. *Transmission evaluation of Ship-borne Automatic Identification System (AIS) in congested waterways*. In *2008 8th International Conference on ITS Telecommunications*, pages 18–23, 2008.
- [7] International Association of Marine Aids to Navigation and Lighthouse Authorities. *G1050 Management and monitoring of AIS information*, 2005.
- [8] International Maritime Organization. *Resolution MSC.74(69): Recommendation on Performance Standards for Universal Automatic Identification System (AIS)*, 1998.
- [9] Swedish Maritime Administration. *AIS: How a swedish innovation became a global standard*, 2019.
- [10] Travis Turgeon. *Advantages and Limitations of AIS (and How to Bridge the Gaps)*. <https://www.darkshipping.com/post/advantages-and-limitations-of-ais>, 2021.
- [11] International Telecommunications Union. *RECOMMENDATION ITU-R M.1313*: Technical Characteristics of Maritime Radionavigation Radars.* , 1997.

Referencias

- [12] Andy Norris. *Automatic Identification Systems – the Effects of Class B on the Use of Class A Systems*. *The Journal of Navigation*, 59(2):335–347, 2006.
- [13] Prefectura Nacional Naval de la República Oriental del Uruguay. *Disposición Marítima N° 94/04*. , 2004.
- [14] Recursos Acuáticos y Prefectura Naval. *Sistema de identificación automática de amplio alcance se implementa en pesca artesanal*. <https://www.gub.uy/presidencia/comunicacion/noticias/sistema-identificacion-automatica-amplio-alcance-se-implementa-pesca>, 2014.
- [15] Decreto N° 353/016. Reglamentación del art. 85 de la ley 18.996, relativo a servicios prestados a terceros para gente de mar y cursos de organización marítima internacional. <https://www.impo.com.uy/bases/decretos/353-2016>, 2016.
- [16] International Telecommunications Union. *Recommendation ITU-R M.1371-5 (02/2014): Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band* , 2005.
- [17] Riadh Essaadali, Chokri Jebali, *AIS data exchange protocol study and embedded development for maritime navigation*. In *2015 IEEE 28th Canadian Conference on Computer Engineering (CCECE)*, pages 1594–1599, 2015.
- [18] Maciej Gucma. *Low Cost AIS System for Safe Navigation*. *Journal of Konbin*, 6:235–246, 01 2008.
- [19] Marine Traffic. AISMon. <https://help.marinetraffic.com/hc/en-us/articles/205339707-AISMon>.
- [20] Kgabo Frans Mathapo. *A software-defined radio implementation of maritime AIS*. Msc. thesis, Stellenbosch : University of Stellenbosch, December 2007. Accepted: 2008-04-09T06:58:29Z.
- [21] Nicolás Molina Padrón. *Diseño e implementación de un prototipo hardware para un banco de pruebas del estándar AIS*. Thesis, 2015.
- [22] Febus Reidj G. Cruz, Ryan Christopher M. Gania, Bryx William C. Garcia, and Jared Christian R. Nob. *Implementing Automatic Identification System Transmitter on Software Defined Radio*. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNI-CEM)*, pages 1–4, 2018.
- [23] Peter Antypas. *MAIANA* . Repositorio de GitHub <https://github.com/peterantypas/maiana>, 2015.

- [24] Trendmicro. *gr-aistx* . Repositorio de GitHub <https://github.com/trendmicro/ais>, 2020.
- [25] Bistromath. *gr-ais* . Repositorio de GitHub <https://github.com/bistromath/gr-ais/tree/master>, 2020.
- [26] Astruder. *dAISy* . Repositorio de GitHub <https://github.com/astuder/dAISy>, 2013.
- [27] jvder-github. *AIS-catcher* . Repositorio de GitHub <https://github.com/jvde-github/AIS-catcher>, 2021.
- [28] ais-dotnet-github. *AIS.Net.Receiver*. Repositorio de GitHub <https://github.com/ais-dotnet/Ais.Net.Receiver>, 2019.
- [29] videgro. *Ships*. Repositorio de GitHub <https://github.com/videgro/Ships>, 2017.
- [30] International Maritime Organization. *IALA Guidelines On The Automatic Identification System (AIS): Part I - Operational Issues*, 1998.
- [31] Riadh Essaadali, Chokri Jebali, *AIS data exchange protocol study and embedded development for maritime navigation*. In *2015 IEEE 28th Canadian Conference on Computer Engineering (CCECE)*, pages 1594–1599, 2015.
- [32] Raymond, Eric S. *AIVDM/AIVDO protocol decoding* . <https://gpsd.gitlab.io/gpsd/AIVDM.html>, 2023.
- [33] Gobierno de la República Oriental del Uruguay. *Licencia de estación radioeléctrica de barco* . <https://www.gub.uy/tramites/licencia-estacion-radioelectrica-barco>.
- [34] Salehi M. Proakis, J. *Digital Communications*. 2008. Quinta edición.
- [35] Katie Terrell Hanna. HDLC (High-level Data Link Control). <https://www.techtarget.com/searchnetworking/definition/HDLC>, 2021. Recuperado: 26 de marzo de 2023.
- [36] All About AIS. *AIS TDMA Access schemes*. http://www.allaboutais.com/jdownloads/Access%20schemes%20technical%20downloads/ais_tdma_access_schemes.pdf, 2012. Recuperado: 21 de marzo de 2023.
- [37] Mictronics. *ais-simulator*. Repositorio de GitHub <https://github.com/Mictronics/ais-simulator>, 2020.
- [38] Mats Kågstrøm. *ais-simulator*. https://www.kagstrom.no/ais_simulator.htm, 2023.
- [39] Administración Nacional de Puertos. *Puerto de Montevideo - Arribos por categoría*. <https://www.anp.com.uy/inicio/puertos/comerciales/montevideo/estadisticas/arribos-por-categoria>. Estadísticas del año 2019.

Referencias

- [40] Hamburger Hafen. *Anzahl der Schiffsanläufe*. <https://www.hafen-hamburg.de/en/statistics/calls/>. Estadísticas del año 2019.
- [41] intel. *NUC7i3DNBE - Dual-Core i3 Motherboard*. <https://simplynuc.co.uk/wp-content/uploads/2018/02/SimplyNUCProductBrief-NUC7i3DNBE.pdf>. Hoja de datos del dispositivo.
- [42] Raspberry Pi. *Raspberry Pi 4 Model B*. <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>. Hoja de datos del dispositivo.
- [43] U-Blox. *NEO-7*. https://content.u-blox.com/sites/default/files/products/documents/NEO-7_DataSheet_%28UBX-13003830%29.pdf. Hoja de datos del dispositivo.
- [44] Bingfu. *Waterproof Active GPS Navigation Antenna*. <https://bingfushop.com/products/bingfu-waterproof-active-gps-navigation-antenna-adhesive-mount-sma-male-gps-ant>. Publicación de venta de la antena en la página del fabricante.
- [45] Analog Devices. *RF Agile Transceiver - AD9363*. <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9363.pdf>. Hoja de datos del dispositivo.
- [46] Analog Devices. *Customizing the Pluto configuration*. <https://wiki.analog.com/university/tools/pluto/users/customizing>. Tutorial de configuración del dispositivo.
- [47] Bmagistro. *gr-aistx* para gnu radio 3.8. Repositorio de GitHub <https://github.com/bmagistro/gr-aistx.git>, 2020.
- [48] *Decoding AIS Data*. <http://www.bosunsmate.org/news/article/258/Decoding-AIS-Data/>, 2007.
- [49] Wikipedia. *NMEA 0183*. https://en.wikipedia.org/wiki/NMEA_0183. Recuperado: 13 de febrero de 2024.
- [50] *AIS0. understanding AIS NMEA 0183: How it's coded*. <https://ldsrc.blogspot.com/2017/09/0-understanding-ais-nmea-0183.html>, 2017.
- [51] L. M.. Richter. *Documantación de Pyais*. <https://pyais.readthedocs.io/en/latest/index.html>.
- [52] Heramienta de software libre para el trabajo con dispositivos de tecnología SDR. *SDR++*. <https://www.sdrpp.org/>.
- [53] dkozel. *gr-latencymanager*. Repositorio de GitHub https://github.com/dkozel/gr-latency_manager, 2022.

- [54] Matt Ettus. *GRCon19 - Managing Latency in Continuous GNU Radio Flowgraphs by Matt Ettus*. Canal de YouTube de GNU Radio <https://www.youtube.com/watch?v=jq0RewceCwc>, 2019.
- [55] Analog Devices. *ADALM-PLUTO Transmit*. <https://wiki.analog.com/university/tools/pluto/users/transmit>.
- [56] miek. *inspectrum*. Repositorio de GitHub <https://github.com/miek/inspectrum>.
- [57] Great Scott Gadgets. *HackRF One*. <https://greatscottgadgets.com/hackrf/one/>.
- [58] RTL-SDR. *RTL-SDR Blog V3 Datasheet*. <https://www.rtl-sdr.com/wp-content/uploads/2018/02/RTL-SDR-Blog-V3-Datasheet.pdf>. Hoja de datos del dispositivo.
- [59] Pregunta dentro del foro de electrónica marina. *ONWA: is it any good?*. Recuperado de <http://www.clubsearay.com/index.php?threads/onwa-is-it-any-good.96680/>, 2005.
- [60] Pregunta dentro del foro náutico deportivo. *Matsutec HP 528A*. Recuperado de <https://foro.latabernadelpuerto.com/showthread.php?t=181236>, 2019.
- [61] Markus Dillinger, Kambiz Madani, Nancy Alonistioti. *Software Defined Radio: Architectures, Systems and Functions*. Wiley Sons. p. xxxiii. ISBN 0-470-85164-3., 2003.
- [62] Emerson. *Software Defined Radio: Past, Present, and Future*. <https://www.ni.com/en/perspectives/software-defined-radio-past-present-future.html>, 2023.
- [63] Wireless Innovation. *Software Defined Radio*. <https://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf>.
- [64] SDR Forum. *SDRF Cognitive Radio Definitions*. <http://www.sdrforum.org/pages/documentLibrary/documents/SDRF-06-R-0011-V100.pdf>, 2007.
- [65] Domschl. *RaspberryNtpServer*. Repositorio de GitHub <https://github.com/domschl/RaspberryNtpServer>, 2023.
- [66] Raspberry Pi. *Raspberry Pi Documentation*. <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>.

Esta página ha sido intencionalmente dejada en blanco.

Índice de tablas

2.1.	Intervalo de reporte para la información dinámica en embarcaciones de clase A. Tabla recuperada de la norma ITU-R M.1371-5 [16]. Se recuerda que un nudo es equivalente a 1,852 km/h.	15
2.2.	Intervalo de reporte para la información dinámica en embarcaciones de clase B. Tabla recuperada de la norma ITU-R M.1371-5 [16]. . .	15
2.3.	Mensajes AIS con su número identificador. Se detalla cuáles deben ser transmitidos o recibidos por estaciones AIS de clase B SO y CS. La abreviación TX hace referencia a la transmisión, mientras que la abreviación RX hace referencia a la recepción.	32
2.4.	Mensajes AIS con su número identificador. Se detalla cuáles deben ser transmitidos o recibidos por estaciones AIS de clase B SO y CS. La abreviación TX hace referencia a la transmisión, mientras que la abreviación RX hace referencia a la recepción.	33
5.1.	Campos del mensaje 18 con sus respectivos largos en bits y descripciones.	68
5.2.	Campos del mensaje 18 con sus respectivos largos en bits y descripciones.	69
5.3.	Campos del mensaje 24 parte A, con sus respectivos largos en bits y descripciones.	70
5.4.	Campos del mensaje 24 parte B, con sus respectivos largos en bits y descripciones.	71
6.1.	Tiempo promedio que lleva el cambio de canal en cada placa. . . .	86
6.2.	Resultados de prueba de funcionamiento para verificar intervalos de reporte.	87
6.3.	Tabla con los tiempos en segundos entre cada impresión de referencia temporal.	92
6.4.	Tabla con los tiempos en segundos entre cada impresión de referencia temporal.	94
7.1.	Costos finales del prototipo implementado. Se indican como referencias las hojas de datos de cada dispositivo.	100
A.1.	Dispositivos AIS de clase B en venta por Aeromarine.	106
A.2.	Estaciones AIS en venta por la empresa Garmin.	106

Índice de tablas

A.3. Otros dispositivos AIS de clase B en venta en el mercado uruguayo.	107
D.1. Valores de potencia obtenidos para el analizador de espectro y ADALM-PLUTO.	117

Índice de figuras

2.1.	Diagrama de bloques para un transmisor AIS. Considerando que se cuenta con los bits asociados a una trama AIS, se procede a aplicar la codificación NRZI. Luego se conforman los bits con un pulso gaussiano y se modula la señal en frecuencia.	17
2.2.	Ejemplo de funcionamiento de la codificación NRZI. Se observa que la señal codificada cambia de nivel cuando los datos tienen un bit en cero.	17
2.3.	Gráfica de un pulso gaussiano en el tiempo. En particular, se tiene un parámetro BT de 0,4.	18
2.4.	Gráfica de varios pulsos gaussianos en el dominio del tiempo, variando el parámetro BT . Se observa que, cuanto mayor es el parámetro BT , menor es el ancho de la gaussiana.	19
2.5.	Diagrama en el tiempo de una transmisión AIS en un slot. Se observa que se tiene un período de 1 ms al inicio del slot para encender el transmisor. Además, se tiene que terminar la transmisión antes de comenzar el siguiente slot. Esta figura fue adaptada de la figura 5 de la norma ITU-R M.1371-5 [16].	20
2.6.	Trama de un mensaje AIS. Se observan los seis campos principales, resultando en un total de 256 bits de largo. Esta figura fue adaptada de la figura 6 de la norma ITU-R M.1371-5 [16].	22
2.7.	Diagrama de flujo asociado al proceso de selección de un slot para transmitir. Se destaca el mecanismo de reutilización necesario cuando no hay suficientes slots libres. Esta figura está basada en la figura 9 de la norma ITU-R M.1371-5 [16].	25
2.8.	Esquema en el tiempo del sensado del nivel de ruido del canal realizado por una estación CS. Se destaca el monitoreo del canal solamente al inicio del slot, entre los instantes de $833 \mu s$ y $1.979 \mu s$. . .	30
3.1.	Representación de reservas de slots de tiempo por dos estaciones que utilizan el mecanismo SOTDMA para el acceso al medio compartido. Se indica también una reserva de una estación base que obstaculiza la reserva de una estación SO.	37

Índice de figuras

3.2.	Gráfica de la cantidad de intentos sobre éxitos totales en función de la cantidad de embarcaciones SO con una única embarcación CS. Se observa un decrecimiento significativo cerca de las 150 estaciones, que indica que el canal comienza a saturar. El comportamiento no es monótono decreciente debido a componentes aleatorios en la simulación.	40
3.3.	Gráfica del promedio de la cantidad de intentos sobre éxitos totales en función de la cantidad de embarcaciones SO con una única embarcación CS. Se consideran 20 realizaciones para el promedio. La simulación se realizó para valores más cercanos de cantidades de estaciones SO a medida que el total se acercaba a 200. Esto permitió un mejor estudio del momento de congestión del canal.	41
3.4.	Gráfica de la cantidad de éxitos totales en función de la cantidad de embarcaciones SO con una única embarcación CS. La simulación se realizó para valores más cercanos de cantidades de estaciones SO a medida que el total se acercaba a 200. Esto permitió un mejor estudio del momento de congestión del canal.	43
3.5.	Gráfica de la cantidad de éxitos totales en función de la cantidad de intentos para distintas proporciones de embarcaciones de clase CS. Observar que cuanto menor es la proporción de estaciones CS (aumentando la cantidad de estaciones SO) el canal es más propenso a saturarse.	44
4.1.	Esquema a alto nivel del sistema transmisor implementado, con los procesos de recepción y el proceso de transmisión. Se agrega la información intercambiada entre ambos para el correcto funcionamiento del sistema completo.	46
4.2.	Diagrama de bloques del proceso receptor. Se observan en la línea superior los bloques preexistentes en <i>gr-ais</i> , y en la línea inferior los bloques nuevos.	48
4.3.	Ilustración del cálculo del umbral dinámico en función de los 60 s anteriores. Se describe el procedimiento desde intervalos de 20 ms, conformando intervalos de 4 s. Se ilustra cómo se elige el mínimo de los últimos 15 intervalos de 4 s como el umbral actual.	50
4.4.	Diagrama con las conexiones para la normalización de la potencia. Se busca calibrar el dispositivo ADALM-PLUTO en función del analizador de espectro.	54
4.5.	Sistema utilizado para la calibración del dispositivo ADALM-PLUTO. Se presenta la versión real del esquema de la figura 4.4.	54
4.6.	Ajuste lineal del transmisor ADALM-PLUTO.	55
4.7.	Esquema de <i>examples</i> que se encuentra en el repositorio de <i>gr-aistx</i> . Se cuenta con los bloques necesarios para generar la trama a partir de los bits del campo de datos, codificar en NRZI, modular en GMSK y luego enviar la señal al canal correspondiente.	58

4.8.	Diagrama de bloques del transmisor implementado, hasta obtener los mensajes a transmitir como un string de bits. Se destacan los intercambios de información entre los bloques.	59
4.9.	Diagrama de bloques completo del transmisor. Se indican en color verde los bloques centrales de este proyecto. Se presentan con azul los bloques que realizan el procesamiento de los bits hasta que se tiene una señal para enviar al canal. Los bloques con línea gruesa continua fueron creados en el marco de este trabajo. Los bloques con línea gruesa punteada fueron modificados por el grupo. El resto de los bloques no presenta cambios.	62
5.1.	Trama de un mensaje AIS. Se observan los seis campos principales, resultando en un total de 256 bits de largo. Esta figura fue adaptada de la figura 6 de la norma ITU-R M.1371-5 [16].	66
5.2.	Diagrama de bloques para la generación de un mensaje. Se deben obtener los datos del módulo GPS previo al armado de la carga útil del mensaje. Se conforma la trama y se aplica el mecanismo de bit stuffing sobre la cadena de bits resultante. Luego se codifica en NRZI y se modula en GMSK, previo a la transmisión en el canal correspondiente.	70
6.1.	Captura de la pantalla del osciloscopio donde se observan dos transmisiones. Se señala una de ellas en verde.	93
6.2.	Acercamiento a las señales en el osciloscopio en el momento de una transmisión. Se observan dos mensajes transmitidos: en primer lugar el mensaje generado, y luego un mensaje “Dummy”. La escala utilizada no se aprecia en la imagen, pero cada recuadro en el eje temporal es de 100 ms.	94
D.1.	Potencia medida por el analizador de espectro para diferentes valores de amplitud en el transmisor.	118

Esta es la última página.
Compilado el lunes 20 mayo, 2024.
<http://iie.fing.edu.uy/>