



TESINA DE LICENCIATURA

Título: Integración de servicios audiovisuales de TV terrestre e IPTV

Autor: Santiago Seifert

Director: Dr. Federico Balaguer

Carrera: Licenciatura en Informática

Resumen

Este trabajo busca el aumento de la lista de servicios provista por la televisión digital terrestre. El objetivo es extender la norma ISDB-Tb (utilizada en Argentina) para la inclusión de servicios audiovisuales transmitidos por red IP. Con esto, se combina la robustez de la emisión broadcast por radiofrecuencia y la flexibilidad de IPTV. El resultado es un sistema híbrido con una lista de servicios con tamaño comparable a la televisión analógica por cable, con la confiabilidad, la calidad y el alcance de la televisión digital terrestre. El impacto de este trabajo se evalúa sobre una infraestructura real con flujos de transporte de referencia.

Palabras Claves

- *Televisión digital terrestre*
- *Extensión de lista de servicios*
- *ISDB-Tb*
- *IPTV*
- *Sistema híbrido de transmisión*
- *Extensión de formato MPEG 2 Transport stream*
- *Construcción de flujos de transporte*
- *Reproductor de Televisión Digital Terrestre*
- *Extensión de OpenCaster*
- *Modificación de Wari, proyecto Kuntur*

Trabajos Realizados

- *Esquema de extensión del formato MPEG-2 Transport Stream*
- *Prototipo de transmisión de la norma ISDT-Tb extendida y construcción de flujos de transporte de referencia.*
- *Prototipo de recepción del formato híbrido de transmisión ISDB-Tb e IPTV.*
- *Análisis de impacto y beneficio del nuevo formato de transmisión.*

Conclusiones

Los sistemas híbridos de transmisión de servicios audiovisuales responden a un esfuerzo sostenido a lo largo de los últimos años para incrementar la matriz de contenidos y audiencia. El enfoque ordenado y simple de este trabajo cumple con los objetivos planteados, abriendo además un abanico atractivo de posibles búsquedas tecnológicas que responden a las agendas de muchas de las empresas más grandes del mundo.

Trabajos Futuros

- *Reubicación de objetos en Object-Based Broadcasting.*
- *Reubicación por eventos.*
- *Soporte de IPV6.*
- *Omisión de PMT pequeñas.*

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Hipótesis de trabajo	3
1.3. Objetivos de la solución	3
1.4. Contribución y resultados obtenidos	4
1.5. Estructura del documento	4
2. Introducción técnica	7
2.1. Conceptos Preliminares	8
2.2. ISDB-Tb	9
2.2.1. Modulación y bitrate disponible	9
2.2.2. MPEG Transport stream	11
2.2.3. Estructura de los paquetes MPEG-TS	11
2.2.4. Recepción de señal ISDB-Tb	12
2.2.5. Tablas	14
2.2.6. Decodificación de tablas en flujos de transporte	18
2.3. Generación de contenido de señal ISDB-Tb	19
2.3.1. OpenCaster	19
2.4. Recepción y decodificación de señal ISDB-Tb	21
2.4.1. Reproductor de Televisión Digital Wari	21
2.4.2. Construcción de lista de servicios en Wari	24
2.4.3. Demultiplexación y reproducción de servicios en Wari	25
2.5. IPTV, <i>Internet protocol television</i>	28
2.5.1. Multicasting IP	30
3. Diseño y desarrollo	31
3.1. Señalización de los servicios	31
3.1.1. Extensión de ISDB-Tb	32
3.1.2. <i>Elementary Streams Relocation Descriptor</i>	34
3.2. Flujo de transporte de referencia	36
3.2.1. Extensión de OpenCaster	36
3.2.2. Extensión del flujo de transporte de Canal 23	36
3.3. Emisión multicast de los flujos elementales por IP	43

3.4. Recepción de contenidos	46
3.4.1. Construcción de Mara	49
4. Evaluación	55
4.1. Infraestructura	55
4.1.1. Infraestructura de prueba	56
4.1.2. Análisis del flujo de transporte como formato contenedor en redes IP	57
4.2. Análisis del trabajo	59
4.3. Número de servicios	59
4.4. Flujos elementales múltiples	61
4.5. Guía electrónica de programación	62
4.6. <i>Closed Captioning</i>	63
4.7. Gestión de audiencias	64
4.8. Aplicaciones Interactivas	65
4.9. Tiempo de sintonización	65
5. Conclusiones y Trabajo Futuro	69
5.1. Crecimiento de la lista de servicios	69
5.1.1. Características importantes de la combinación ISDB-Tb con IP	70
5.2. Mara	70
5.3. Trabajo Futuro	71
5.3.1. Reubicación de objetos en <i>Object Based Broadcasting</i>	71
5.3.2. Reubicación por eventos	71
5.3.3. Soporte de IPV6	72
5.3.4. Omisión de PMT pequeñas	72
5.4. Conclusión	73
A. Introducción al <i>MPEG-2 Transport Stream</i>	75
A.1. Codificación y empaquetado de los datos	75
A.1.1. Formato de paquetes TS	76
A.1.2. Obtención de paquetes PES a partir de paquetes TS	79
A.1.3. Terminología útil en el flujo de transporte MPEG-2	80
B. Limitación del número de servicios de la transmisión híbrida	83
B.1. Tamaño de sección	83
B.2. Disponibilidad de paquetes nulos	85
C. Estudio de Wari	87
C.1. Multithreading	90
C.2. Compilación de Wari y Mara	90
C.2.1. Troubleshooting	91

Capítulo 1

Introducción

Facebook, Apple, Google, Netflix, Amazon, son sólo algunas de las compañías más importantes de la actualidad que invierten dinero y esfuerzo en la distribución de contenidos audiovisuales. La industria televisiva mueve miles de millones de dólares diariamente a nivel mundial.

En Latinoamérica, en el 2015, cerca de 60 millones de hogares consumen alguna señal televisiva[1]. En poco más de la mitad de los casos, esta señal es paga. En Argentina, particularmente, son 5 millones los hogares con televisión. Cerca de un 83 % (más de 4 millones) de los hogares consumen una señal televisiva paga[2].

Estas estadísticas responden a un crecimiento sostenido a lo largo de los últimos años y evidencian la relevancia que posee la industria, no sólo en la región, sino a nivel mundial. La explotación económica viene acompañada de una consiguiente búsqueda tecnológica por satisfacer cada vez mejor la demanda.

Hace varios años ya, Brasil encabeza un movimiento regional para aunar a los gobiernos de Latinoamérica bajo un mismo estándar de televisión digital terrestre. Esta unificación significa un incentivo al intercambio técnico, científico y, sobre todo, de contenido televisivo.

1.1. Motivación

A pesar de la mejora radical que conlleva la señal digital por sobre la analógica, la televisión terrestre presenta un problema ineludible: la cantidad de servicios que ofrece es insuficiente. Al momento de escribir este informe, la Ciudad Autónoma de Buenos Aires presenta la mayor cantidad de servicios entregados en el país por la televisión digital abierta, un total de 28 servicios. Por otro lado, en una red privada de transmisión de televisión analógica por cable normal, el número de servicios asciende a 70. A pesar de que la digitalización de la señal de radiofrecuencia optimiza la utilización del medio de transmisión, la televisión digital terrestre no puede competir con la televisión por cable en lo que a tamaño de lista de servicios respecta.

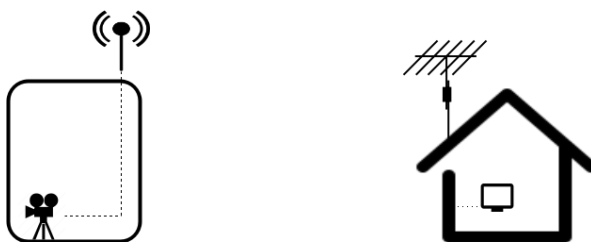


Figura 1.1: Esquema de la emisión de televisión digital terrestre.

El número de *canales*¹ de la televisión digital terrestre depende del número de bandas de radiofrecuencia asignadas a la transmisión de tal servicio. Es improbable que este aspecto cambie radicalmente en el futuro, dado que las tendencias actuales a nivel mundial consisten en priorizar la asignación de bandas de radiofrecuencia a la telefonía móvil (en Argentina está vigente el decreto 2426/2012). Esto significa que si se busca engrosar la programación de la televisión digital terrestre, es necesario analizar otras posibilidades.

En la región latinoamericana, los países llevan adelante medidas para la creación de redes de acceso a internet de alta velocidad. UNASUR[3] aprobó el tendido de un gran anillo de fibra óptica para la interconexión de las redes existentes. En Argentina, se encuentra en vigencia una política nacional de telecomunicaciones denominada Argentina Conectada[4].

En el marco de desarrollo de semejante infraestructura, la provisión de servicios televisivos por red IP a gran escala es una realidad alcanzable. Hay características de las redes IP que las alejan de ser ideales para la tarea. Un ejemplo concreto de esto es la susceptibilidad a sobrecargas de consumo. Estas redes se basan en la conmutación de paquetes, cuando la exigencia supera la capacidad de la infraestructura, la eficiencia del medio disminuye, causando pérdida de paquetes y tiempos de latencia excesivos para sistemas de tiempo real. Esta es una complicación tangible de cualquier sistema de televisión por IP.

Por otro lado, las redes IP presentan cualidades positivas que no se encuentran en las emisiones *broadcast* de la televisión terrestre. Concretamente, son entornos de comunicación más flexibles y proveen un canal de retorno, posibilitando el envío de información desde el cliente hacia el servidor. Si la sobrecarga de consumo no es un problema, el tamaño de la lista de servicios disponible es arbitrario. En otras palabras, cada dispositivo direccionable de una red IP podría potencialmente ofrecer un servicio audiovisual. A su vez, si un servicio no es requerido por ningún usuario, el mismo no impone exigencia sobre la infraestructura, en contrapartida con la emisión *broadcast* donde todo debe ser enviado constantemente, independientemente de los consumidores.

Durante el año 2014, en Brasil, se llevó adelante el mundial de fútbol organizado por

¹El término *canal* se suele utilizar en la televisión analógica en reemplazo del término **servicio** por la correspondencia unívoca entre ambos. En el caso de televisión digital, cada canal puede y suele contener varios **servicios**. Éste último es el término correcto.

la FIFA. Con la popularidad del deporte en Argentina, al momento de la definición de semifinal por penales contra Holanda, casi 10 millones² de espectadores estaban sintonizando el evento deportivo. En ocasiones como esta, el fallo del servicio de televisión no es una opción.

Como se mencionó, a diferencia de la televisión digital terrestre, IPTV permite listas de servicios arbitrariamente grandes, aunque no sea aconsejable que los servicios críticos se encuentren en ellas. Por el contrario, la televisión digital terrestre es ideal para estas situaciones siendo que no se ve amenazada en escenarios con semejante exigencia. Es observable que las cualidades de ambos medios se complementan.

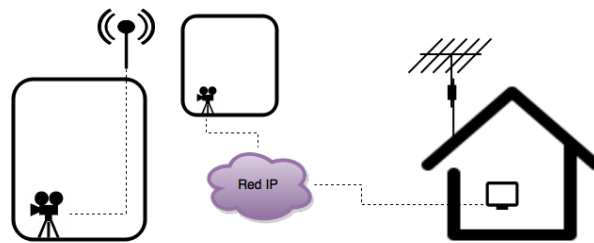


Figura 1.2: Esquema de la emisión de televisión digital terrestre en combinación con IPTV.

1.2. Hipótesis de trabajo

Se espera que el trabajo realizado en esta tesina ataque la limitación de tamaño de la lista de servicios de las señales ISDB-Tb a través de la combinación con la emisión por red IP. El resultado será un sistema híbrido que complementa la robustez de las emisiones *broadcast* por radiofrecuencia terrestre con la flexibilidad de la conmutación de paquetes de las redes IP. En consecuencia, la lista de servicios percibida del primer medio aumentaría sin la necesidad de utilizar una porción mayor del espectro radioeléctrico empleando, en su lugar, un canal IP.

1.3. Objetivos de la solución

Retrocompatibilidad

El estándar ISDB-Tb ya es utilizado en un gran número de sistemas de recepción. Una cualidad esencial de una extensión con las características de este trabajo es la compatibilidad con versiones anteriores. Es decir, aquellos servicios que se sigan transmitiendo por radiofrecuencia, deben seguir siendo accesibles.

²<http://www.tutele.net/2014/07/10/audiencias-tv-casi-10-millones-de-espectadores-siguen-los-penaltis-de-paises-bajos-argentina/>

Minimización de cambios

Una cualidad buscada en el diseño debe ser la minimización del número de cambios introducidos al estándar. Los motivos para esto son tanto mantener simplicidad en la extensión como requerir la menor cantidad de trabajo posible en los cambios de receptores ya funcionales.

Aprovechamiento eficiente de la infraestructura

En el caso de las redes IP, el uso eficiente de los recursos de red, como *buffers* o ancho de banda es importante para proteger la infraestructura de transmisión, a diferencia de la emisión *broadcast* de ISDB-Tb cuya integridad no sufre el aumento de consumidores. Específicamente es necesario evitar cualquier tipo de emisión redundante en la red IP.

Transparencia para el usuario

Uno de los objetivos con los que se desarrolla este trabajo es el de proveer los servicios al usuario sin que perciba diferencia alguna entre los servicios de los distintos medios.

1.4. Contribución y resultados obtenidos

- Esquema de extensión del formato MPEG-2 Transport Stream.
- Prototipo de transmisión de la norma ISDB-Tb extendida y construcción de flujos de transporte.
- Prototipo de recepción del formato híbrido de transmisión ISDB-Tb e IP.
- Análisis de impacto y beneficio del nuevo formato de transmisión.

1.5. Estructura del documento

Este documento se compone de cuatro capítulos, además del actual:

- **Introducción Técnica:** Explora los elementos técnicos necesarios para la comprensión de la solución desarrollada.
- **Diseño y desarrollo:** Expone el diseño e implementación del esquema de transmisión que combina ISDB-Tb e IPTV. Esto comprende desde la extensión de la norma de televisión digital terrestre hasta la modificación de un reproductor de TV para la creación de uno que soporta la nueva funcionalidad.
- **Evaluación:** Analiza los resultados expuestos en el capítulo anterior comparándolos con la norma tradicional.
- **Conclusión:** Resume lo comprendido por el informe y plantea puntos de trabajo futuro para la complementación de esta tesina.

Al final de este informe, también, se incluyen tres apéndices que proveen lectura complementaria útil:

- **Apéndice A:** Presenta una introducción útil para el lector que desea familiarizarse con el formato *MPEG Transport Stream*, del que depende radicalmente el trabajo realizado.
- **Apéndice B:** Expone en detalle una de las mediciones realizadas sobre el desarrollo. La naturaleza de esta medición se explicará en el capítulo de Evaluación.
- **Apéndice C:** Realiza un recorrido introductorio por el software utilizado como piedra fundamental en el capítulo de Diseño y Desarrollo.

Capítulo 2

Introducción técnica

La Televisión Digital(TVD) es un servicio innovador que significa la mayor revolución tecnológica en el rubro televisivo desde la televisión a color en la década del '50. El término **Televisión Digital** define la transmisión de audio y video a través de una señal procesada y multiplexada digitalmente, en contrapartida con la señal analógica utilizada por la *televisión analógica*, antecesora de la primera. Muchos países continúan hoy en día reemplazando la televisión analógica por televisión digital, haciendo un mejor y más variado uso del espectro de radiofrecuencia.

Las ventajas de la televisión digital por sobre la analógica son numerosas:

1. Exhibe una mejor calidad de sonido e imagen.
2. Permite contenidos en alta definición.
3. Posibilita la multiprogramación, dado que permite la transmisión de varias señales en el mismo ancho de banda asignado a la emisora. Esta es una diferencia sustancial respecto de la correspondencia unívoca entre servicio y canal de radiofrecuencia de la televisión analógica.
4. Permite que un mismo servicio provea varios flujos distintos de un mismo tipo. Por ejemplo, para transmitir una película con audio en varios idiomas.¹
5. Permite brindar servicios complementarios, como aplicaciones interactivas y guía electrónica de programación.
6. Ahorra ancho de banda. Es posible transmitir más servicios en el mismo ancho de banda.
7. Permite brindar servicios a una mayor cantidad de dispositivos, como teléfonos celulares, computadoras portátiles, o GPS's con sintonizador. Esto se conoce como *producción para multiplataforma*. Un mismo servicio se puede reproducir en distintos dispositivos, con diferentes soportes de resolución.

¹La televisión analógica utiliza técnicas (como *Second Audio Program*) para la emisión de dos alternativas de audio o *closed caption*. No obstante, no se puede hablar de multiplicidad de flujos, dado que se limita a tipos específicos de flujos y, en general, a dos alternativas.

2.1. Conceptos Preliminares

La televisión digital utiliza *bits* para codificar tanto contenidos audiovisuales como metainformación. Las señales digitales pueden ser canalizadas por distintos medios y viajar con distintos formatos. Los principales tipos de televisión digital son (ver [5]):

- Terrestre: Emite la señal a través del espectro de radiofrecuencia de la atmósfera terrestre sobre las bandas UHF².
- Satelital: Consiste en retransmitir desde un satélite una señal emitida desde un punto en la Tierra utilizando SHF³.
- Cable: Utilizan la señal propagada por redes de cable coaxial o fibra óptica.
- *Internet Protocol Television*(IPTV): El término ha adquirido múltiples significados con el crecimiento de las tecnologías. La *International Telecommunication Union*[7] define al término de la siguiente manera:

Definición 1. *IPTV define a la entrega de servicios como televisión/video/audio/-texto/imágenes/datos sobre redes basadas en IP manejadas para proveer el nivel requerido de calidad de servicio y experiencia, seguridad, interactividad y confiabilidad.*

- *Over the Top*(OTT): Consiste en la transmisión de televisión por internet. A diferencia de IPTV, al no utilizar una red manejada, no presenta garantías de calidad de servicio.

Las primeras dos categorías se denominan *de radiodifusión*, dado que utilizan el espectro radioeléctrico en emisión *broadcast*. La televisión digital terrestre y satelital se encuentran rigurosamente legisladas y reguladas en todos los países del mundo[6]. En el caso particular de la televisión digital terrestre, hay cuatro estándares ampliamente adoptados:

- *Terrestrial Integrated Services Digital Broadcasting (ISDB-T)*: Adoptado en Japón y las islas Filipinas. **ISDB-T International** (también denominado **ISDB-Tb**) es una adaptación de este estándar que ha sido aplicada en gran parte de sudamérica.
- *Digital Video Broadcasting-Terrestrial (DVB-T)*: Estándar adoptado en Europa, Australia y Nueva Zelanda.
- *Advanced Television System Committee (ATSC)*: Estándar adoptado en Estados Unidos, Canadá, México, Corea del Sur, República Dominicana y Honduras.
- *Digital Terrestrial Multimedia Broadcasting (DTMB)*: Estándar adoptado en la República Popular de China, incluyendo Hong Kong y Macao.

²*Ultra High Frequency*: designación del *International Telecommunication Union* para las frecuencias que oscilan entre los 300MHz y 3GHz

³*Super High Frequency*: designación del *International Telecommunication Union* para las frecuencias que oscilan entre los 3GHz y 30GHz

2.2. ISDB-Tb

ISDB-Tb (o ISDB-T Internacional) es un estándar de transmisión de televisión digital terrestre. Fue desarrollado originalmente por la Asociación de Empresas e Industrias de Radio (ARIB[12]) del Japón y, posteriormente, incluyó modificaciones propuestas por la Asociación Brasileña de Normas Técnicas[13]. Es la norma de televisión digital adoptada en Argentina y utilizada en la Televisión Digital Abierta (TDA[16]).

La televisión terrestre envía información a través del espectro de radiofrecuencia de la atmósfera que oscila entre los 300MHz y los 3GHz. Al ser éste un fenómeno físico intrínsecamente analógico, la información digital que se desea transmitir se codifica por medio de variaciones de amplitud y fase de la señal. Estas variaciones se denominan **modulación**. ISDB-Tb permite utilizar varios parámetros de modulación distintos.

Por encima de la capa de codificación de la información digital o modulación de bits, la norma define una forma de organización de los bytes. Ésta organización compone lo que se denomina *formato contenedor*. En el caso de ISDB-Tb, el formato contenedor de la transmisión es el **Flujo de Transporte MPEG** (*MPEG Transport Stream*), definido en MPEG-2 ISO/IEC standard 13818. Es este formato el que permite la multiplexación de servicios y flujos multimedia en una misma señal de radiofrecuencia. Por supuesto, aprovechando la efectividad de las técnicas avanzadas de modulación de información digital.

2.2.1. Modulación y bitrate disponible

En la norma ISDB-Tb, el espectro de radiofrecuencia se divide en canales físicos de 6 MHz (denominados *bandas UHF*) por los que viaja un flujo multiplexado de servicios, que denominaremos “flujo de transporte”. Cada uno de estos canales físicos dispone de un *bitrate* constante definido por los parámetros de modulación de la señal. La capacidad máxima de transmisión del sistema **ISDB-Tb** es de unos 23 Mbps[5] que debe distribuirse entre los contenidos a transmitir.

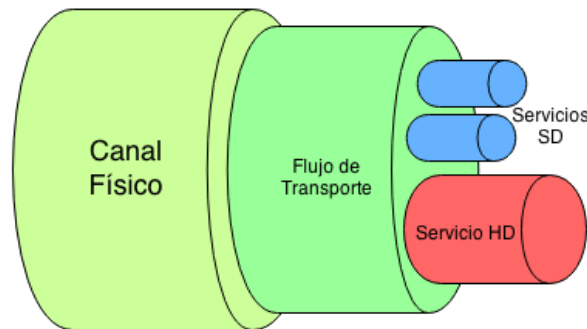


Figura 2.1: Diagrama esquemático de un flujo de transporte

La limitación de bitrate de los canales físicos establece una cota superior para el número de servicios que puede transportar, dependiendo ulteriormente del ancho de banda

requerido por los mismos. Cada servicio está compuesto a su vez por flujos más pequeños que llevan audio, video u otras componentes.

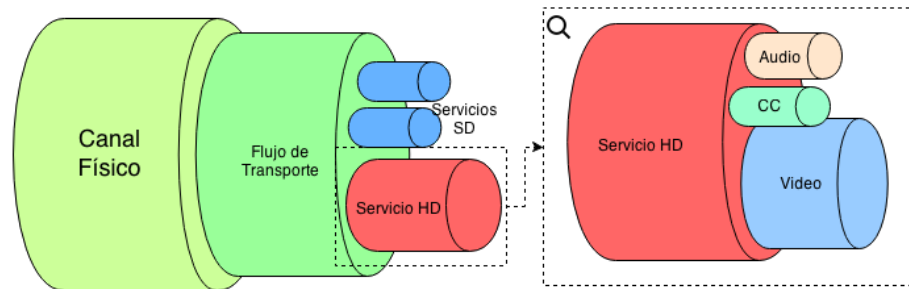


Figura 2.2: Composición de servicios en un flujo de transporte

Para ejemplificar, un fragmento de la señal emitida por el canal 23 (frecuencia 527 Mhz) de la Televisión Digital Abierta Argentina del 5 de Junio de 2013 dispone de un ancho de banda de 17,3 Mbps. Presenta 3 servicios: TV Pública HD(un video HD, dos audios), Tecnópolis(un video SD, un audio) y TV Pública(un video LD, un audio). El bitrate promedio para estos videos se presenta en Tabla 2.1.

Nombre del servicio	Tipo de video	Bitrate promedio utilizado	Porcentaje del bitrate del flujo utilizado
TV Pública HD	<i>High Definition Video</i>	11,7 Mbps	67,7%
TV Pública	<i>Low Definition Video</i>	0,3 Mbps	1,9%
Tecnópolis	<i>Standard Definition Video</i>	2,0 Mbps	11,7%
Total:	—	14 Mbps	81,3%

Tabla 2.1: Utilización de bitrate por el video en el ejemplo de Canal 23

La Tabla 2.1 indica el bitrate promedio utilizado por los videos, sumando un total de 14Mbps. Dado que el bitrate disponible total es de 17,3 Mbps, resta indicar en qué se utilizan 3.3 Mbps(18,7%) del ancho de banda siendo que, como se ha mencionado, el mismo es constante en la norma ISDB-Tb. Los audios toman un *bitrate* promedio de 0,4 Mbps entre los 4. La información requerida por el flujo de transporte para ser demultiplexable y utilizable por un decodificador comprende un 0,4% del *bitrate* total(unos 0,07Mbps). La guía electrónica de programación, por otro lado, ocupa casi el doble, con un 0,7% del ancho de banda total.

Cuando el contenido de un flujo de transporte no llega a cubrir el ancho de banda disponible del canal físico, el emisor debe completar con contenido *dummy* que el receptor debe ignorar. Este contenido se compone de *paquetes nulos*. De modo que en la muestra utilizada por el ejemplo anterior, un promedio de 2,4 Mbps son llenados con paquetes nulos. Contenidos con la finalidad de completar el bitrate disponible.

Considerando los promedios del ejemplo, podría pensarse en la posibilidad de agregar un servicio LD sin riesgo de que falte bitrate. No obstante, es importante tener en mente

que las magnitudes presentes son calculadas por promedio. En el caso del servicio HD, el video alcanza los 24 Mbps, por momentos. Por la variabilidad del bitrate de los contenidos multimedia, es necesario que la suma de los bitrate promedio sea inferior al bitrate total si se busca una calidad estable en los servicios.

Ya es conocida la importancia de la modulación de información digital en este formato de televisión. Aún así, este proceso no es suficiente para la entrega de contenidos. Aún es necesario organizar los bytes para que tengan significado en el receptor. De otra manera, el receptor observaría un flujo ininterrumpido de bytes dispuestos fuera de contexto. A fin de interpretar los mismos es necesario agruparlos de alguna manera y asignarles un significado. Éste es el objetivo del formato contenedor *MPEG-2 Transport Stream*.

2.2.2. MPEG Transport stream

El formato contenedor de los datos que viajan en las emisiones ISDB-Tb, está definido por el estándar **MPEG-2 Transport Stream** o simplemente *flujo de transporte*. El flujo de transporte MPEG posee características que lo hacen ideal para la televisión terrestre.

- Está diseñado para ambientes de transmisión o persistencia no confiables, es decir, con presencia de errores y pérdida. En el caso de ISDB-Tb, este medio son los canales físicos de radiofrecuencia.
- El flujo de transporte MPEG permite la multiplexación de servicios con bitrates variables e independientes. Esto permite aprovechar al máximo la capacidad de transmisión de los canales de radiofrecuencia. Incluye mecanismos para la identificación y presentación de estos servicios.
- Especifica mecanismos para la sincronización de flujos de audio y video.
- Es un formato extensible para adecuarse a los medios sobre los que se utiliza.

La modulación de la señal permite pensar en cada canal físico como un flujo constante de bits. Para organizarlo, el flujo de transporte MPEG agrupa los bits en paquetes de 188 bytes cada uno. Los paquetes de longitud fija poseen la característica de simplificar la inclusión de campos para la detección y corrección de errores. A su vez, permite transportar varios servicios en una misma línea de tiempo, sin la necesidad de contar con una referencia de reloj común a todos ellos, puesto que se intercalan paquetes de cada uno.

2.2.3. Estructura de los paquetes MPEG-TS

Desde el punto de vista de un receptor de señal, el flujo de transporte es una serie de dígitos binarios, de modo que es imposible definir dónde termina un paquete y comienza el siguiente. El receptor también necesita saber qué información viaja en el interior de un paquete, para interpretar el contenido binario. Con estos objetivos, entre otros, cada paquete de un flujo de transporte contiene un prefijo de 4 *bytes*.

El primer byte de ellos es el **byte de sincronización**, que sirve al receptor para identificar el comienzo de los paquetes. Transporta siempre el valor 47_H . Entonces, cuando el receptor sintoniza un canal de radiofrecuencia dado, debe utilizar este campo para encontrar el comienzo de los paquetes. Este proceso se conoce como *sincronización*. El estándar **MPEG-TS**[8] define que, al encontrar 5 bytes de sincronización intercalados por 187 bytes, el receptor puede asumir que se ha sincronizado al flujo.

Otro campo del prefijo de los paquetes de igual importancia que el byte de sincronización transporta el *packet identifier* (**PID**) del paquete. Consiste en un entero sin signo de 13 *bits* que identifica los contenidos. Gracias a este valor, el receptor puede obtener los paquetes de audio, video y datos. En general, es necesario conocer el PID con el que viaja una entidad del formato a fin de buscar la información que contiene.

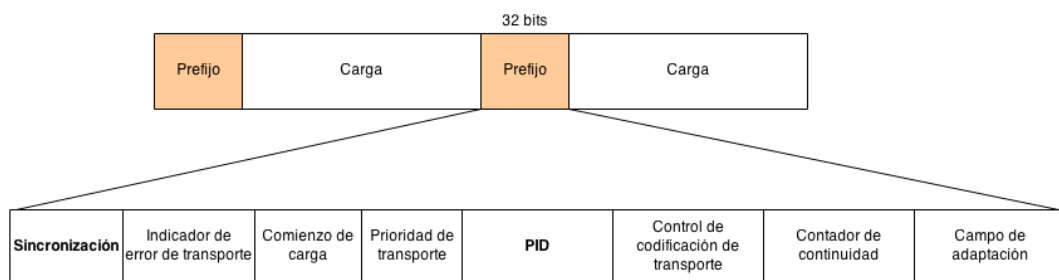


Figura 2.3: Estructura de un paquete de flujo de transporte

Para información más detallada relacionada con la estructura de los paquetes del flujo de transporte ver apéndice A.

2.2.4. Recepción de señal ISDB-Tb

Los receptores de señal ISDB-Tb tienen dos tareas básicas. La primera tarea es entregar al usuario la lista de servicios disponibles. La segunda es reproducir aquel servicio que el usuario elija. Ambas necesitan la información provista a través del formato contenedor, el flujo de transporte MPEG.

Construcción de la lista de servicios

A diferencia de la televisión analógica, en televisión digital cada banda UHF contiene más de un servicio. Si se desea conocer los servicios que viajan por un canal, es necesario sintonizar ese canal e identificar los servicios a través de la información que contiene el flujo de transporte.

La Tabla 2.2 presenta una lista de servicios de la **Televisión Digital Abierta**. La primera columna indica el canal físico por el que viaja cada flujo de transporte y la segunda, todos los servicios asociados al mismo.





















Canal	Servicio	Logotipo	Formato de video ⁴
22 (521MHz)	Encuentro		SD (576i, 50Hz)
	Paka Paka		SD (576i, 50Hz)
	TaTeTi		SD (576i, 50Hz)
	INCAA TV		SD (576i, 50Hz)
	Encuentro LD		LD (240p, 30Hz)
23 (527MHz)	TV Pública HD		HD (1080i, 50Hz)
	Tecnópolis		SD (576i, 50Hz)
	TV Pública		LD (240p, 30Hz)
24 (533MHz)	DeporTV		SD (576i, 50Hz)
	Vivra		SD (576i, 50Hz)
	Suri TV		SD (576i, 50Hz)
	Arpeggio		SD (576i, 50Hz)
	Viajar		SD (576i, 50Hz)
	Depor TV LD		LD (240p, 30Hz)
25 (539MHz)	CN23		SD (576i, 50Hz)
	C5N		SD (576i, 50Hz)
	Telesur		SD (576i, 50Hz)
	360 TV		SD (576i, 50Hz)
	Construir TV		SD (576i, 50Hz)
	CN 23 LD		LD (240p, 30Hz)

Tabla 2.2: Lista de servicios de la plataforma nacional de la Televisión Digital Abierta

Cada canal físico contiene un flujo de transporte independiente, que multiplexa a su vez varios servicios. Un sintonizador puede decodificar un único canal físico en un momento dado. De modo que para indicar al usuario la lista completa de servicios debe

⁴La letra minúscula que acompaña al formato de video define el tipo de scan utilizado. *i* define entrelazado y *p* define progresivo.

memorizarlos previamente. A este proceso se lo conoce como “*scan*”.

El *scan* consiste en recorrer todas las frecuencias posibles, sincronizándose y memorizando información del sistema. A partir de esta información, el receptor puede conocer los nombres de los servicios y su tipo. A su vez, recordará en qué canal se encuentra cada uno, para su posterior sintonización.

Asumiendo que se desea construir la lista de servicios correspondiente a la Tabla 2.2, el *scan* consiste en recorrer los canales 22, 23, 24 y 25, memorizando toda la información que provee cada flujo de transporte. Ejemplo de esta información, en términos del Canal 23, son el nombre de la red transmisora (“RTA C23”), los nombres de los servicios (“TV Pública”, “TV Pública HD” y “Tecnópolis”) y el nombre del flujo de transporte (“TV Pública”). Estos tres servicios se sumarían a todos los de otros flujos de transporte de otros canales físicos, como “Encuentro” de canal 22 (Frecuencia 521MHz).

El proceso de *scan* puede demorar algunos minutos, dependiendo de la cantidad de frecuencias recorridas. Por eso, la información obtenida y construida hasta aquí suele ser persistente. Así, aún luego de apagar el sistema de recepción, el usuario no requerirá volver a repetir este procedimiento.

Sintonización de servicios

Esta tarea se puede realizar únicamente una vez la lista de servicios está disponible y se debe llevar a cabo cada vez que el usuario solicita un servicio.

1. Si el usuario está mirando “TV Pública HD” y decide sintonizar “Encuentro”, es necesario realizar un salto de canal en el receptor. El primero viene por Canal 23 (Frecuencia 527MHz) y el segundo por Canal 22 (Frecuencia 521MHz).

Si por el contrario el usuario quisiera sintonizar “Tecnópolis” en lugar de “Encuentro”, este paso podría ignorarse, dado que ambos se encuentran en el mismo flujo de transporte, ya sintonizado.

El proceso de cambiar de canal físico, implica demodular la nueva señal y sincronizarse con la emisión, para distinguir el comienzo de cada paquete e interpretar correctamente la información.

2. Una vez el receptor esté recibiendo el flujo de transporte buscado, debe lograr distinguir y filtrar los paquetes que pertenecen al servicio requerido por el usuario. Es decir, si se sintoniza “Encuentro”, el receptor debe decodificar el video de este servicio de televisión, y no de “TV Pública HD”. Lo mismo ocurre con el audio y otros flujos multimedia.
3. Para finalizar, todo el contenido multimedia debe ser redirigido a los dispositivos de entrega: Televisor, monitor y altavoces, dispositivo móvil, etc.

2.2.5. Tablas

En la televisión analógica, un sistema de recepción se limita a sintonizar un canal físico, interpretar la señal y reproducirla para el usuario. La televisión digital requiere

de procesos más complejos, como la construcción de la lista de servicios. Para estos, el receptor requiere información por parte del emisor fuera del contenido multimedia propiamente dicho. Para entregar dicha información, ISDB-Tb utiliza **tablas**.

Las tablas son estructuras sintácticas concebidas en el formato **MPEG-TS** que cumplen una función específica como transportadores de información. Existe un conjunto de cuatro tablas denominado *Program Specific Information* (**PSI**, creadas por el formato MPEG-TS) que interviene directamente en la demultiplexación de servicios. Estas son **PAT**, **CAT**, **NIT** y **PMT**, que serán exploradas a continuación. Como el formato MPEG-TS trasciende los dominios específicos como la televisión digital, es necesario que algunas tablas sean agregadas por normas como ISDB-Tb. De las tablas ajenas al PSI, sólo la **SDT** es de interés para este trabajo, que es agregada para la distribución de TV digital. La lista completa de tablas en la norma ISDB-Tb se define en ARIB-STD B10[9].

La figura 2.4 muestra la relación entre las distintas tablas. Las referencias entre tablas se realizan a través del PID que tienen asignado. A continuación se desarrolla la utilidad de cada una.

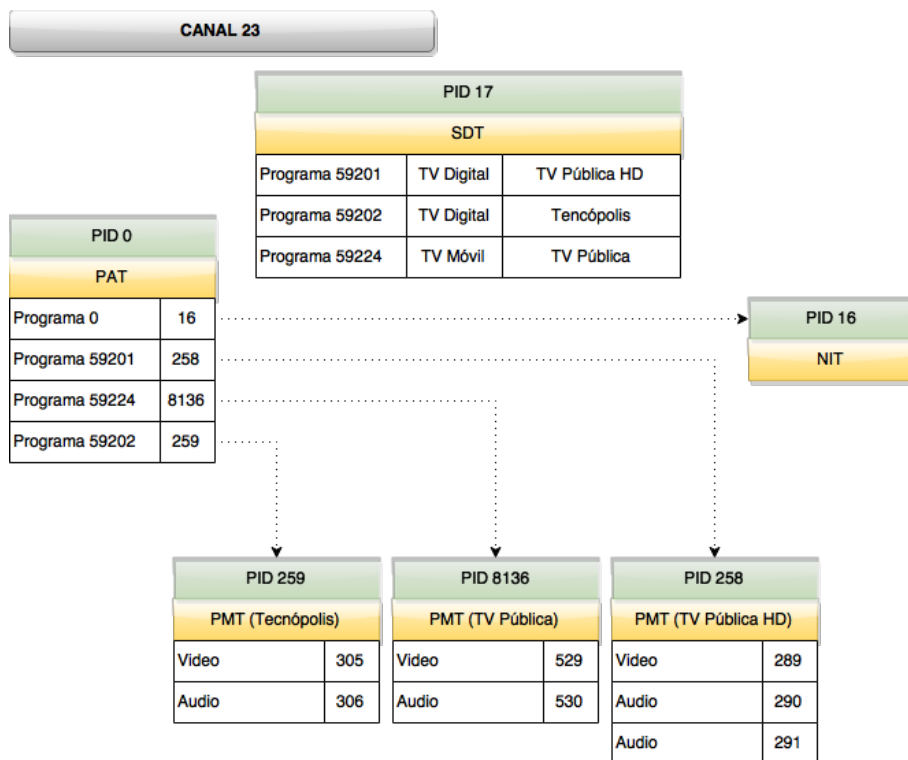


Figura 2.4: Esquema general de tablas en el ejemplo de Canal 23

Es importante notar que los paquetes del flujo de transporte tienen un tamaño de 188 bytes y disponen de aún menos para cargar contenidos. Las tablas pueden exceder este tamaño. Para zanjar este problema, las mismas viajan en *secciones*, que son estructuras

de longitud variable y se dividen entre tantos paquetes como sean necesarios. El receptor debe reconstruir las secciones para obtener las tablas que contiene.

Program allocation table (PAT)

La **PAT** es la primera tabla que debe buscar el decodificador para sintonizar un servicio. Viaja con **PID** 0. Existe una **PAT** por flujo de transporte y asocia a cada servicio con un **PID**. La identificación del servicio se hace a través del número de *service_id*. El término programa se puede utilizar como sinónimo de servicio.

<i>service_id</i>	Program PID (o PID de PMT)
59201	258
59202	8136
59224	259

Tabla 2.3: Contenidos de la PAT emitida por Canal 23

Program Map Table (PMT)

La **PMT** define los contenidos multimedia que corresponden a un programa. El **PID** que tiene asignado un servicio por **PAT**, es el **PID** con el que viaja la **PMT** de tal servicio. Esta tabla le permite al decodificador acceder a los paquetes que llevan los flujos con contenidos multimedia a través de sus respectivos **PID**.

Todos los paquetes de un mismo **PID** que transportan contenido multimedia componen lo que se denomina *flujo elemental*. De modo que la **PMT** permite al receptor encontrar los flujos elementales.

Tipo de flujo	PID del flujo
27(Video)	289
17(Audio)	290
17(Audio)	291

Tabla 2.4: PMT que viaja con PID 258 en el ejemplo de Canal 23

Service Description Table (SDT)

La **SDT** describe con más detalle los programas transportados por un flujo de transporte. Viaja con **PID** 17. Es una tabla descriptiva, como el nombre lo indica, en términos de que incluye información textual sobre el servicio, como su nombre. También establece de qué tipo de servicio se trata: radio, TV, *etc.*

<i>service_id</i>	Nombre del servicio	Tipo del servicio
59201	TV Pública HD	Televisión
59202	Tecnópolis	Televisión
59224	TV Pública	(Data Service) Televisión Móvil

Tabla 2.5: Contenidos de la SDT emitida por Canal 23

Network Information Table (NIT)

La transmisión de la **NIT** es obligatoria en los sistemas de televisión terrestre. Se envía en paquetes con **PID** 16 y contiene información obligatoria sobre la organización física de la red de transmisión que la emite, pero también de forma opcional puede enviar información de otras redes.

Descriptores

Dentro de las tablas, parte de la información viaja en un formato de longitud variable denominado “descriptores”. Cada descriptor es identificado numéricamente de forma unívoca por un *descriptor_tag* que ocupa un *byte*. El segundo elemento que viaja en un descriptor es un campo denominado *descriptor_length* que denota la longitud l de la estructura en *bytes*. Los siguientes l *bytes* transportan el contenido del descriptor propiamente dicho.

Los descriptores viajan en *loops* de descriptores, y no existe una limitación para la cantidad que pueden viajar en un mismo *loop*. Tampoco existen restricciones para el orden en que deben viajar los descriptores y el mismo no tiene influencia sobre el significado de los mismos. Una lista completa de los tipos de descriptores de ISDB-Tb se provee en ARIB-STD B10[9].

Un ejemplo de descriptor es el *service_descriptor* de la SDT. Es identificado por el *descriptor_tag* 72 y transporta una descripción de un servicio, incluyendo el nombre. En el caso del ejemplo del canal 23, la SDT transporta un *service_descriptor* por cada uno de los servicios. Del mismo desde donde se obtienen los nombres “TV Pública”, “TV Pública HD” y “Tecnópolis”.

Flujos elementales

El formato con que viajan los audios y videos se denomina *Packetized elementary stream*(**PES**). Cada paquete **PES** de este formato es de longitud variable y, de forma general, se puede asumir más grande que un paquete de flujo de transporte. Se puede establecer una analogía entre los paquetes **PES** y las secciones. Igual que las secciones, los paquetes **PES** se dividen entre los paquetes de flujo de transporte.

A fin de posibilitar la reconstrucción, todos los paquetes que pertenecen al mismo flujo elemental viajan con el mismo **PID**. Todos estos paquetes con igual **PID** componen un único flujo elemental. Típicamente, un servicio de televisión dispone de al menos un

flujo elemental de audio y un flujo elemental de video. En el caso de “Televisión Pública HD” del ejemplo anterior, el servicio dispone de un video HD y dos audios.

En el ejemplo, el servicio “Televisión Pública HD”, el video viaja con el PID 289 y los audios con PID 290 y 291. Cabe notar que los flujos elementales también pueden corresponder a *closed caption* y contenido interactivo.

2.2.6. Decodificación de tablas en flujos de transporte

Es sumamente importante para este informe analizar cómo influyen las tablas en los procesos de *scan* y selección de servicios. Es a través de estos procedimientos que la combinación de medios es posibles. En primer instancia, el *scan* permite que los servicios ajenos al flujo de transporte sean presentados al usuario. Y el segundo de ellos efectiviza la búsqueda de información multimedia para su posterior reproducción. Claro está, si se requiere modificar el protocolo estándar es menester comprenderlo.

Las tareas de construcción de lista de servicios y sintonización de servicios que se trataron en la Sección 2.2.4 se basan en la información provista por las tablas de la norma ISDB-Tb. A continuación se detalla cómo influye cada una:

Tablas en la construcción de lista de servicios

El proceso de construcción de la lista de servicios se resume a recorrer todas las frecuencias por las que viajan flujos de transporte, capturando la información contenida por la **SDT** y **NIT**. El receptor, con la información incluida en ambas tablas (de cada flujo) y en sus respectivos descriptores, ya puede ofrecer la lista completa. En este paso se genera la asociación entre servicio y flujo de transporte, similar a la Tabla 2.2. Es decir, el proceso de *scan* le indica al receptor que “TV Pública HD” es transmitido por Canal 23, al igual que “Tecnópolis” y “TV Pública”. “DeportTV” por Canal 24, y así sucesivamente.

Tablas en la sintonización de servicios

1. El primer paso es sintonizar el canal físico por el cual viaja el flujo de transporte que lleva el servicio deseado. El mismo se obtiene de la asociación generada en el *scan*.
2. Una vez el canal físico está en sintonía, y el receptor sincronizado con el flujo de transporte, el receptor debe reconstruir la sección que viaja con **PID 0**, que corresponde a la **PAT**.

En el ejemplo de Canal 23, en la tabla encontrará la lista con los 3 servicios. Sabiendo por la **SDT** que el nombre “TV Pública HD” corresponde al *service_id* 59201, de la **PAT** se obtiene el PID con que viaja la **PMT**, que en el ejemplo es 258.

3. Con el **PID** de la **PMT**, el receptor reconstruye la sección que lleva la tabla. De ella extrae los **PID** de los flujos elementales. “TV Pública HD” envía dos flujos de

audio (**PIDs** 290 y 291) y un flujo de video HD con **PID** 289. Con éstos **PID**, ya es posible reconstruir los contenidos multimedia.

4. Los flujos elementales deben decodificarse y enviarse a los periféricos de entrega.

2.3. Generación de contenido de señal ISDB-Tb

El proceso de transformación de la información desde que se captura en un estudio de televisión hasta que es emitida por una antena emisora está compuesto por tres etapas:

- a **Generación de contenidos:** en una estación de televisión hay un conjunto de tecnologías para este fin. Cámaras, *switchers*, generadores de efectos, consolas de audio, *etc.* que entregan flujos elementales (audio y video) de alta velocidad para su posterior codificación. En general, cada flujo se entrega a **Codificación** por separado.
- b **Codificación y empaquetado:** en esta etapa se realiza la compresión del audio y video, para adaptarla a la capacidad de transmisión de los canales físicos. La información contenida en cada flujo elemental es segmentada en paquetes de bits llamados PES (*Packetized elementary streams*), que luego son divididos nuevamente a paquetes **MPEG-TS** de 188 bytes. En general, cada programa genera tres flujos de paquetes **MPEG-TS**: video, audio y datos.
- c **Señalización y multiplexación de programas:** en la etapa final se compone el flujo binario **MPEG-TS** de entrega. Integra el audio, video y datos de los servicios que lo compongan, completamente optimizado para la transmisión. En este punto se incorporan las tablas que señalizan los servicios.

Las herramientas involucradas en la señalización y multiplexación de programas deben seguir escritos lineamientos de la norma ISDB-Tb (en concordancia con el formato **MPEG-TS**). Por ejemplo, cada **PMT** debe encontrarse en el flujo de transporte con una periodicidad de 200 milisegundos. Existen varios proyectos de utilidades de televisión digital que resuelven estas exigencias.

2.3.1. OpenCaster

OpenCaster es un proyecto de código abierto (bajo licencia GPL[19]) que incluye muchas de las herramientas necesarias para la construcción de flujos de transporte de acuerdo al formato **MPEG-TS**. Estas van desde la creación de paquetes, secciones y tablas hasta la multiplexación del *ts*.

OpenCaster provee una API para *Python* desde la cual se pueden ensamblar tablas aprovechando *hotspots* dispuestos en las clases que modelan entidades de las normas de TVD más populares. Una vez construida una instancia de una tabla, la misma se serializa a un arreglo de *bytes* con el formato **MPEG-TS**. Esta API provee una librería de elementos que incluye tablas y descriptores, junto con los campos asociados a cada uno.

Generación de Tablas

La utilización de la API requiere crear un *script* que genere las estructuras buscadas. Desde el mismo se importan las clases provistas, se instancian y luego, el comportamiento heredado permite serializarlas y empaquetarlas. El producto de la ejecución del *script* son secciones ya divididas en paquetes listos para ser añadidos al flujo de transporte.

Fragmento 2.1: Ejemplo de creación de tablas en OpenCaster

```

1 pmt = program_map_section(
2   program_number = 59202,
3   PCR_PID = 2064,
4   program_info_descriptor_loop = [],
5   stream_loop = [
6     stream_loop_item(
7       stream_type = 2,
8       elementary_PID = 2064,
9       element_info_descriptor_loop = []
10    ),
11    stream_loop_item(
12      stream_type = 3,
13      elementary_PID = 2068,
14      element_info_descriptor_loop = []
15    ),
16  ],
17  version_number = 1,
18  section_number = 0,
19  last_section_number = 0,
20 )

```

Multiplexación del flujo de transporte

Una vez que se poseen todas las componentes del flujo, las mismas se deben multiplexar. Este proceso consiste en inyectar los paquetes TS al flujo de transporte, de modo que se respete la periodicidad requerida por las distintas entidades. Por ejemplo, en el flujo de transporte resultante, una instancia de la PMT debe aparecer cada 200ms a lo sumo. En el caso de los flujos elementales multimedia, es necesario tener en cuenta el *bitrate* para evitar que la distribución de la carga final sea errónea causando así *overflows* o *underflows*. La multiplexación puede ser realizada a través de una herramienta provista en el proyecto OpenCaster llamada `tscbrmuxer`. Con esta herramienta se puede determinar el *bitrate* final del *ts*.

En las primeras dos secciones de este capítulo se exploraron conceptos importantes de la norma ISDB-Tb, como la influencia de paquetes, tablas y descriptores en el mecanismo de multiplexación de servicios. Esto dio lugar a la exploración del procedimiento de construcción de un flujo de transporte. A continuación, se expone el extremo opuesto del sistema, la recepción, utilizando como caso de estudio un reproductor de televisión digital existente.

2.4. Recepción y decodificación de señal ISDB-Tb

El estándar[14] establece especificaciones deseadas de recepción, decodificación y reproducción de los servicios de un flujo de transporte. Existen problemas complejos asociados al proceso:

- La mayoría de los eventos de recepción y reproducción son asincrónicos. La lectura de tablas, en concreto, implica esperar la llegada al receptor de todos los paquetes necesarios para construir la sección por la que viaja. Salvo por las limitaciones de tiempo que impone la norma sobre la emisión de las diferentes tablas, no hay forma de predecir el momento en que esto ocurre. En concreto, el estándar establece que la PMT debe ser transmitida en un período no mayor a 200 milisegundos. A partir de ahí, el decodificador no posee más garantías para la espera.
- Es una aplicación que debe responder correctamente a las exigencias de un sistema *soft real time*[22].
 - Por un lado, el flujo de transporte es una fuente constante de paquetes que se acumulan en un *buffer*. Si el sistema no llega a consumir suficientes paquetes en un intervalo de tiempo determinado, esto puede incurrir en un *overflow*, lo cual significa que existe pérdida, y esto a su vez se traduce en una degradación del servicio.
 - Por el otro lado, la presentación de los cuadros del video debe responder a las exigencias de un sistema multimedia. La omisión de cuadros, si ocurre con suficiente periodicidad, también implica una degradación de la calidad.

El manejo de los eventos asincrónicos exige el uso de concurrencia. Esto se puede ver reflejado en la figura 2.5 y la figura 2.6 que incluyen diagramas de flujo que representan el proceso de *scan* y reproducción de un servicio, respectivamente. Es útil analizar la adaptación de estos procesos en el marco práctico de un reproductor de televisión digital. Para esto, se introduce *Wari*.

2.4.1. Reproductor de Televisión Digital Wari

Wari es un reproductor de televisión digital en la norma ISDB-Tb ideado para ser utilizado en computadoras personales. Es actualmente desarrollado y mantenido por Lífia[20], bajo la licencia LGPL[18]. Está desarrollado mayoritariamente en C++ y Lua. Información útil respecto al reproductor se puede encontrar en el apéndice C. En esta sección se exploran los conceptos estrictamente necesarios para el desarrollo de esta tesina.

Los diagramas de secuencia presentados en este informe presentan un único hilo de ejecución. En realidad, *Wari* utiliza un *dispatcher* de tareas a través del cual los hilos se asignan invocaciones de funciones que deben ser realizadas cuando el control está disponible. Para la notificación de finalización de estas tareas (como podría ser la lectura satisfactoria de una tabla), se puede utilizar este mismo mecanismo o invocación

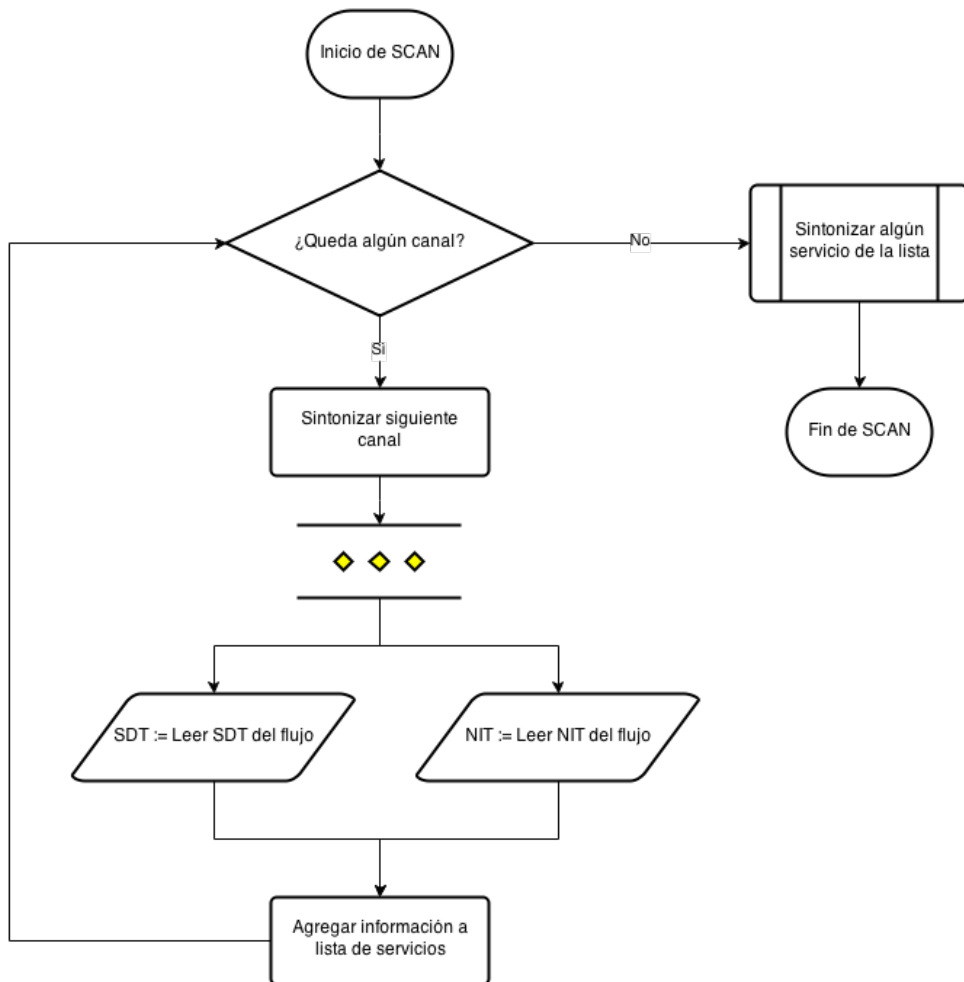


Figura 2.5: Diagrama de Flujo del procedimiento tradicional de Scanning

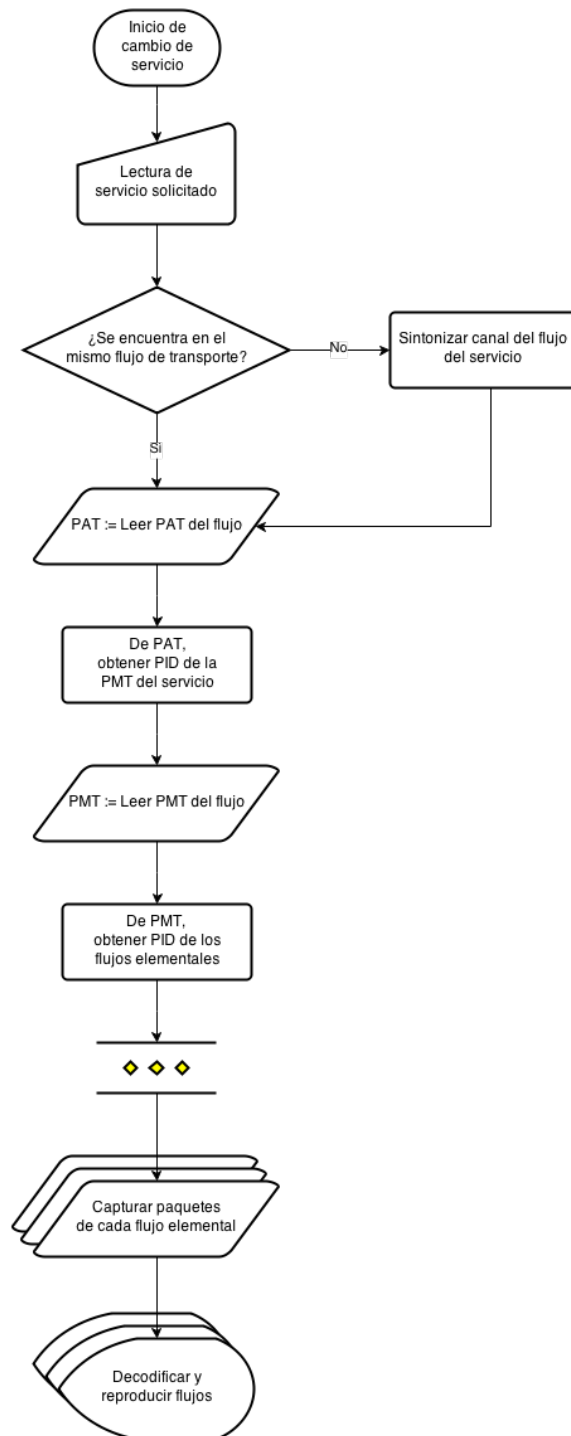


Figura 2.6: Diagrama de Flujo del procedimiento tradicional de Reproducción de Servicios

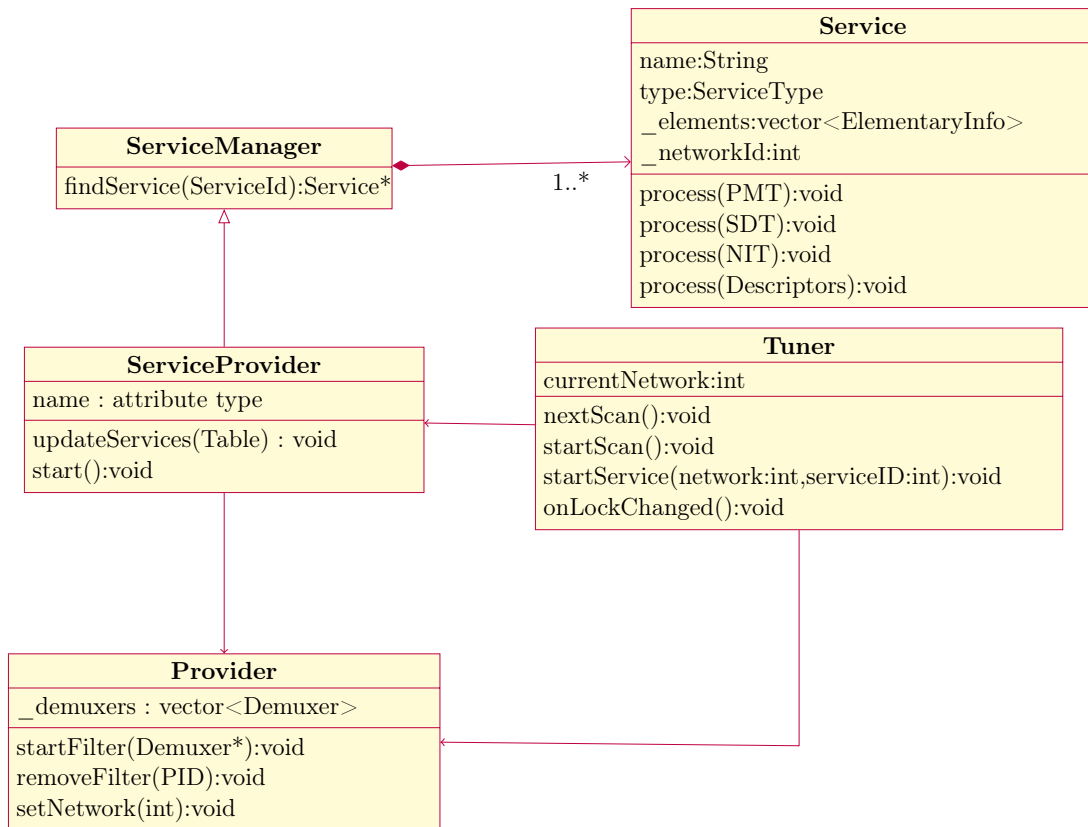


Figura 2.7: Diagrama de Clases de parseo de descriptores en Wari

de funciones *callback*. Si bien la ejecución de los métodos es en realidad diferida, la simplificación ayuda drásticamente la lectura y comprensión de los procesos.

2.4.2. Construcción de lista de servicios en Wari

Contruir la lista de servicios consiste en identificar todos los servicios que aparecen en todos los flujos de transporte disponibles para el sistema receptor. En el caso común, recorrer las **SDT** de todos los flujos de transporte sería suficiente, dado que esta indica los nombres y tipos de los servicios. Wari da un paso extra e involucra también las tablas **PAT** y **PMT**. Si bien este no es exactamente igual al procedimiento descrito por el estandar, garantiza que los servicios que aparecen en la lista de servicios sean seleccionables para el usuario⁵. El procedimiento completo se describe en la figura 2.8, la figura 2.7 muestra un enfoque estático de la lógica utilizada en Wari a través de un diagrama de clases.

⁵No es algo completamente extraño encontrar flujos de transporte mal formados donde no se emiten tablas necesarias o hay errores de referencia por PID.

La solicitud del usuario de un *scan* lleva a la invocación del método `startScan` de la clase `Tuner`. Ésta solicitará sistemáticamente al `Provider` que sintonice las frecuencias disponibles. Sobre cada una de las frecuencias, el `ServiceProvider` esperará por la **SDT**, la **PAT**, y la **PMT**. Cuando las tablas llegan al `Provider`, se notifica a las instancias de la clase `Service` correspondiente, y esta se actualiza acordemente. Parte del procesamiento de las tablas, como se indica en la figura 2.8, implica procesar también los descriptores que incluye la tabla.

Para obtener las tablas **SDT**, **PAT**, y **PMT**, es necesario antes construir las secciones por las que viajan. A su vez, las secciones requieren de los paquetes sobre los que viajan. Los paquetes se asocian a las tablas a través del **PID**. Cada tabla tiene un **PID** asignado.

Para obtener los paquetes de un **PID** dado, el sistema utiliza los *demuxers*. Los *demuxers* son *listeners* que se ubican sobre el flujo de transporte que esperan un paquete con un **PID** dado. De modo que cuando el `ServiceProvider` desea construir la lista de servicios, le pide al `Provider` que cree un *demuxer* para cada tabla que necesitan, a través del método `startFilter(aDemuxer)`⁶.

Un diagrama de la relación estática entre las clases del sistema se presenta en la figura 2.7. Descripción de la responsabilidad de las clases:

- `Tuner` coordina el proceso de *scan* desde el tope de la jerarquía de abstracción. La clase `Tuner` responde a todos los requerimientos del *Zapper*⁷ que utiliza el usuario.
- `ServiceProvider` junto con `ServiceManager` gestionan los servicios (que se modelan a través de la clase `Service`), durante todo su ciclo de vida. Es esta clase la que posee la lista de servicios propiamente dicha.
- `Provider` encapsula y abstrae las operaciones de *hardware* del dispositivo de sintonización subyacente. También gestiona los *demuxers*.
- Cada instancia de `Service`, finalmente, modela un servicio que viene por alguno de los flujos de transporte disponible. Guarda información como *service_id* nombre y tipo.

2.4.3. Demultiplexación y reproducción de servicios en Wari

El proceso de demultiplexación de servicios consiste en la reconstrucción de los flujos elementales originales, que habían sido fragmentados en paquetes TS para su multiplexación en el flujo de transporte. Antes de comenzar la recolección de estos paquetes TS, el receptor debe conocer los **PID** de todos los flujos elementales requeridos para reproducir enteramente el servicio. Por ejemplo, en el caso de “TV Pública HD”, se deben obtener del flujo de transporte todos los paquetes de **PID** 289 y 290, que corresponden al video HD y al primero de los audios, respectivamente, como se indica en la figura 2.4. La carga útil contenida en los paquetes TS se ensambla para obtener paquetes PES, que pueden

⁶El método se llama `startFilter` en lugar de `startDemuxer` porque la componente que conecta el *demuxer* con el flujo de transporte se llama *filter*.

⁷Se denomina *Zapper* a un dispositivo utilizado para sintonizar servicios en un televisor.

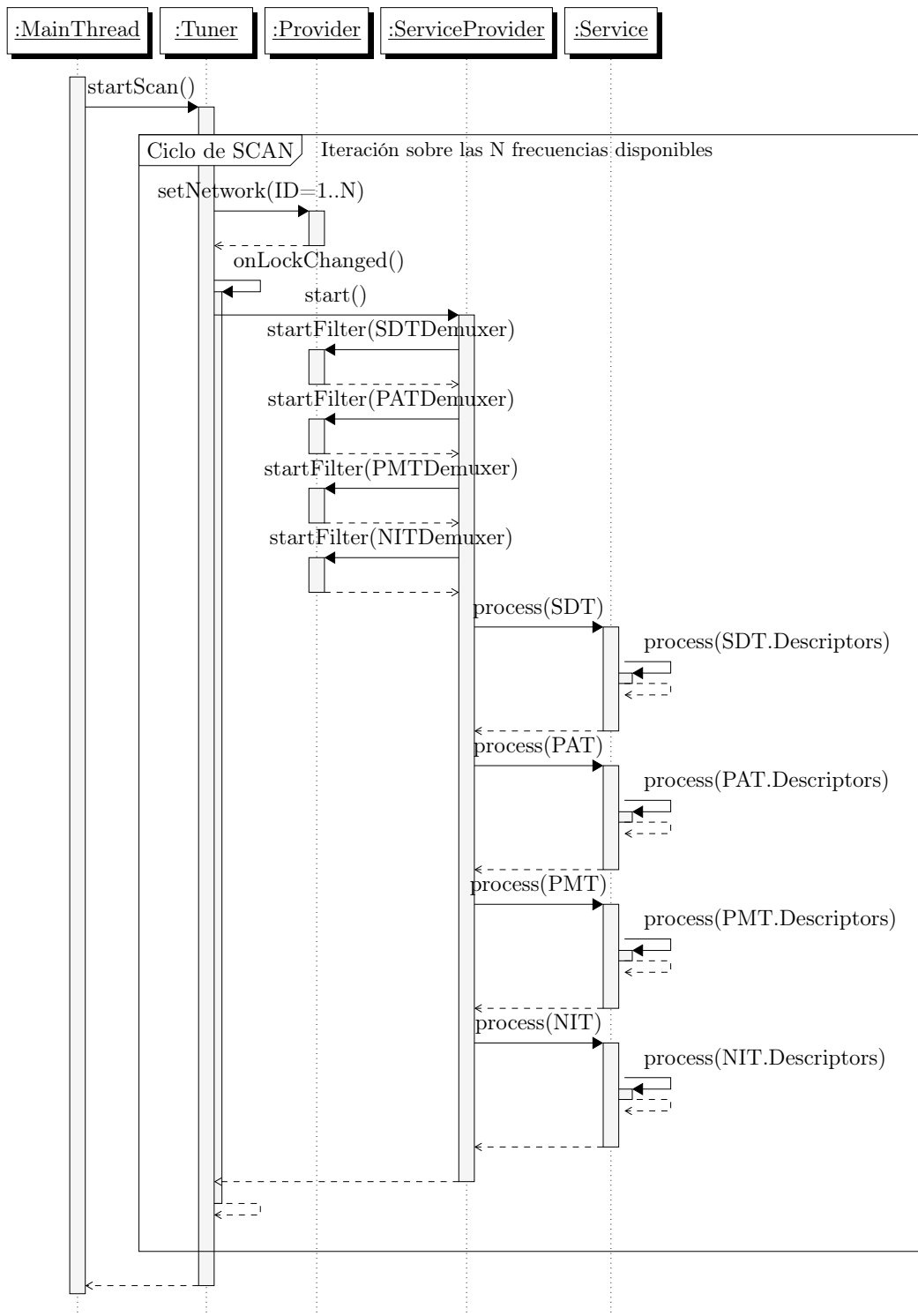


Figura 2.8: Diagrama de Secuencia de construcción de lista de servicios en Wari

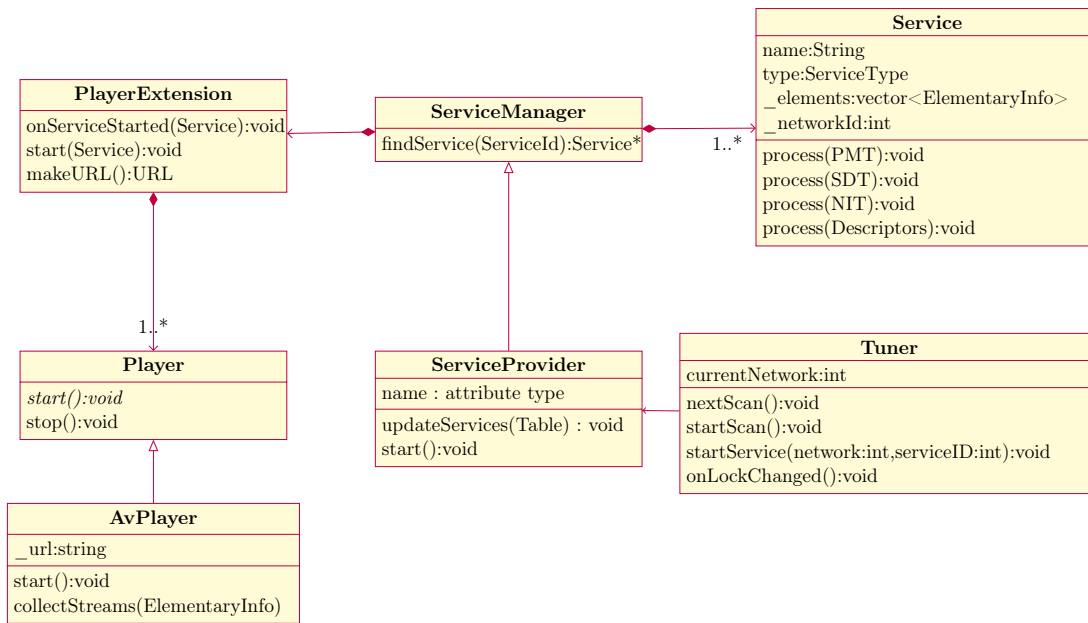


Figura 2.9: Diagrama de clases de infraestructura de reproducción en Wari

ser interpretados por un reproductor multimedia. Información más detallada se puede encontrar en el apéndice A.

El proceso de reproducción de flujos elementales comienza una vez se dispone de los flujos elementales reconstruidos, luego que se ha realizado *buffering* de suficientes paquetes. Consiste en decodificarlos y obtener las imágenes y el audio que componen al servicio. En Wari, este proceso es realizado por alguna librería de reproducción existente, como VLC[21]. Wari sólo se limita a facilitar el acceso a los flujos elementales a la librería de reproducción elegida⁸. La figura 2.9 muestra un diagrama de clases de la infraestructura relacionada con la recolección de flujos elementales.

Las clases representadas en la figura 2.9 son:

- **PlayerExtension**: Representa la máxima abstracción en términos de reproducción de servicios que provienen de un flujo de transporte, gestiona los mecanismos de comunicación internos.
- **Player**: Es la superclase de la jerarquía de destinos de reproducción. Cualquier clase que vaya a representar un tipo de reproducción, debe ser subclase de **Player**.
- **AvPlayer**: Representa el reproductor de Audio/Video de servicios que provienen de un flujo de transporte que se captura a través de un dispositivo sintonizador.
- **ServiceManager** y **ServiceProvider**: La jerarquía gestiona el ciclo de vida de los

⁸Este es uno de los tantos aspectos parametrizables en la configuración de compilación de Wari.

servicios y encapsula todo el procesamiento relacionados con ellos, como el que se realiza en la captura de tablas relacionadas con servicios.

- **Tuner:** Clase que provee la interfaz genérica de un *Zapper*. Permite escanear y sintonizar servicios.
- **Service:** Clase que modela a los servicios de los flujos de transporte disponibles para el receptor. Almacenan toda la información vinculada a los mismos, como nombre, id, o información de flujos elementales.

La figura 2.10 presenta un diagrama de secuencia del proceso de recolección y reproducción de flujos elementales a partir de la solicitud de sintonización de un servicio. Este protocolo deriva del formato MPEG-TS.

En Wari, cuando el usuario solicita el cambio de servicio, el control llega a la clase **Tuner**. A partir de entonces, el primer paso es buscar la instancia de la clase **Service** que corresponde al servicio solicitado por el usuario. Una vez que se dispone de esta, se le debe indicar al **PlayerExtension** que se busca reproducir el servicio. Éste obtiene de la instancia de **Service** la información asociada a los flujos elementales que se deben reproducir, como los **PID** de los mismos. Toda esta información es enviada al **AvPlayer**.

Los **PID** permiten ubicar filtros sobre los flujos de transporte. El filtro, a su vez, facilita la recepción de todos los paquetes con un **PID** determinado. Es el **AvPlayer** quien crea los filtros para los flujos elementales del servicio a reproducir. Sin embargo, aún es necesario indicarle al reproductor la forma de reproducirlos. Esto se entrega en un *Uniform Resource Locator* o URL.

El URL de reproducción se construye en el método `makeURL` de **PlayerExtension**. Este URL se almacena en el **AvPlayer** para que posteriormente, cuando el proceso de cambio de canal haya terminado, el *main thread* se lo solicite y se lo envíe al reproductor de audio y video. En el caso de selección de servicios tradicional, este URL refiere al dispositivo de sintonización y parámetros que sirven, en conjunto, para que la librería de reproducción multimedia pueda obtener los paquetes PES directamente. La observación pertinente a realizar es que los URL disponen de un dominio casi universal. Esta flexibilidad permite que se localicen no sólo dispositivos de hardware, sino también recursos en una red IP, o potencialmente recursos web. Este parámetro es un excelente punto de introducción para nuevos medios de transmisión, permitiendo la fácil reproducción de IPTV en el esquema global de transmisión.

2.5. IPTV, *Internet protocol television*

La definición 1 de la *International Telecommunication Union*, presentada en la Sección 2.1, explica el término forma abstracta, en consideración de las posibles variantes del uso del término. Pone énfasis en la garantía de calidad y confiabilidad. Siendo que estos aspectos de la transmisión de televisión a través de internet no son centrales para este trabajo, se buscará acotar la definición para que se adapte al uso que se le dará en los capítulos siguientes. Previo a introducir la misma, es necesario explorar aspectos específicos del medio para diferenciarlo del medio terrestre.

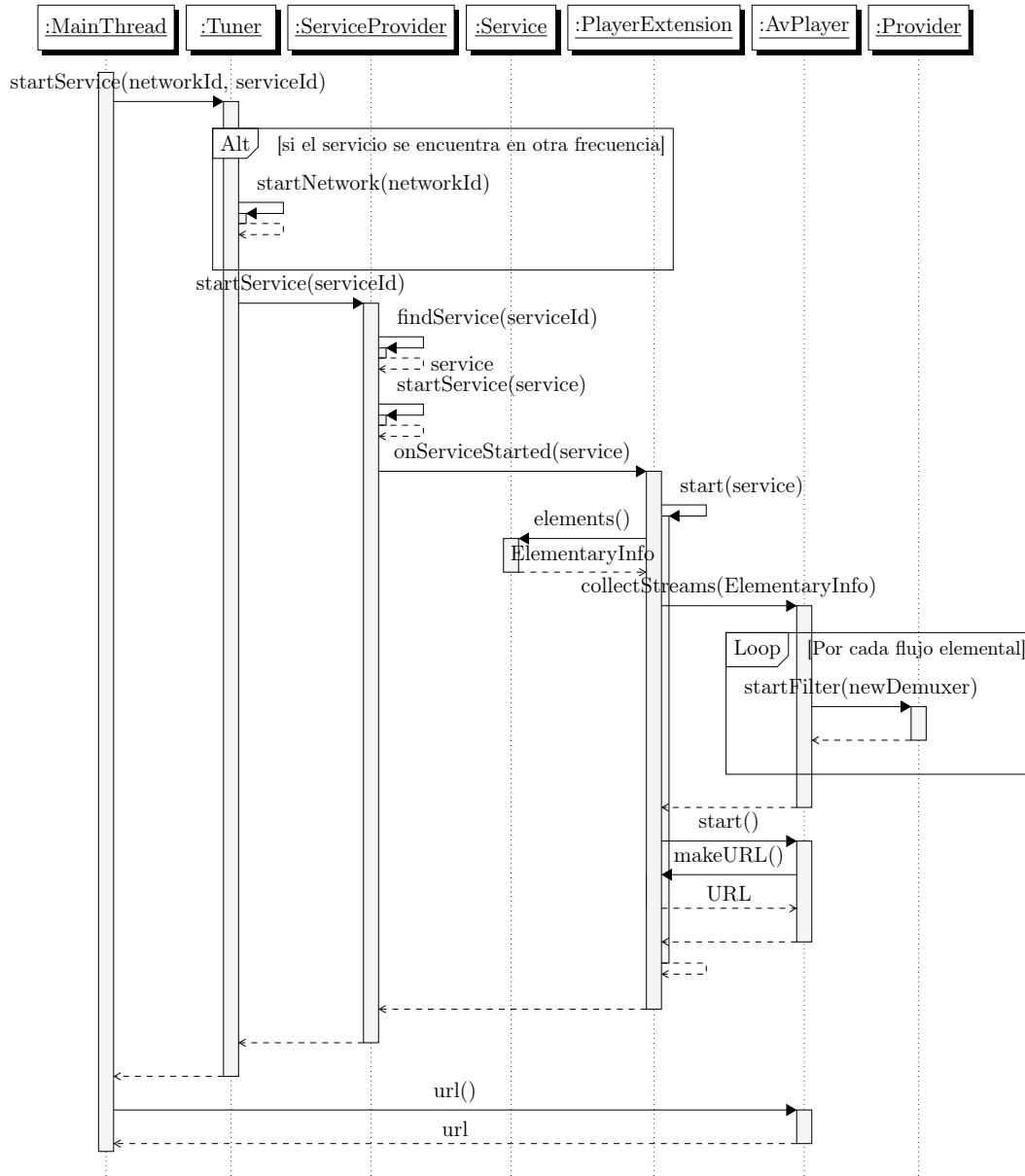


Figura 2.10: Diagrama de secuencia de infraestructura de reproducción en Wari

- **Bitrate variable:** A diferencia de la señal radioeléctrica, el uso de ancho de banda se adapta a los contenidos transmitidos.
- **Conmutación de paquetes:** La delimitación de paquetes en las redes IP es una garantía de la infraestructura, no es necesario que el formato contenedor respete tamaños de información específicos, ni que el receptor se sincronice⁹.
- **Multicasting:** Las redes IP permiten la emisión multicast, ideal para los contenidos televisivos. La señal ISDB-Tb llega a todos los receptores (con alcance físico) por igual y estos optan por consumirla o ignorarla. En las redes IP, los *hosts* deben solicitar la emisión para que esta les sea provista. Además, contar con la dirección IP y el puerto del grupo son suficientes para identificarlo y obtener los contenidos que emite.

El objetivo de este trabajo es incrementar el tamaño de la lista de servicios sin incurrir en una mayor exigencia de ancho de banda en el flujo de transporte. A partir del objetivo se puede elaborar una nueva definición de IPTV que se adecúa más al ámbito de desarrollo de estos objetivos.

Definición 2. *IPTV define la entrega de servicios audiovisuales en emisión multicast a través de una red IP, donde cada servicio se limita a la multiplexación de un flujo de audio y uno de video junto con la información indispensable para la reproducción simultánea y sincronizada.*

2.5.1. Multicasting IP

El multicasting IP[10] es un método de envío de datagramas a través de una única transmisión, a un grupo de receptores interesados. Es una técnica *uno a muchos* utilizada en la entrega de contenidos multimedia en tiempo real, como es el caso de IPTV.

El protocolo más común de capa de transporte utilizado para multicast IP es UDP[11]. UDP no posee protección ante pérdida de paquetes o errores de transmisión, de modo que es necesario contemplar estos problemas en alguna de las capas superiores.

Para soportar multicasting, los *routers* de una red IP debe dar soporte y permitir su uso a los *host* de la misma. El multicasting IP consiste en un conjunto de *grupos multicast* identificados unívocamente por una dirección IP. En el caso de IP versión 4, las direcciones que identifican grupos multicast van desde *224.0.0.0* hasta *239.255.255.255*. Cuando un *host* desea comenzar a recibir los paquetes correspondientes a una emisión multicast, debe unirse al grupo multicast por el cual se realiza la emisión. Cabe notar que los paquetes UDP poseen un puerto destino, de modo que también es necesario conocer el puerto al que se envían estos paquetes.

Con *multicasting* IP acaba la lista de conceptos necesarios para la comprensión del desarrollo de esta tesina, demostrado en el capítulo a continuación. Lo que resta estudiar es el acople de las piezas existentes, evitando construir agentes de software complejos que rompan compatibilidades y los objetivos de este trabajo.

⁹Siempre y cuando se alineen los comienzos de paquetes TS y paquetes IP.

Capítulo 3

Diseño y desarrollo

El objetivo de este capítulo es explicar el diseño de un sistema de entrega de televisión digital que integra emisión ISDB-Tb e IPTV. El desarrollo se divide en cuatro objetivos a los que les corresponden las cuatro secciones de este capítulo:

- **Señalización de los servicios:** Se denominará *señalización* a la entrega de información sobre los servicios para la construcción de la lista y su reproducción. Esta sección abarcará la extensión de ISDB-Tb necesaria para combinar la norma con IPTV.
- **Flujo de transporte de referencia:** Esta sección describe los pasos de construcción de un flujo de transporte que exponga las modificaciones diseñadas en la sección anterior para la norma ISDB-Tb.
- **Emisión multicast de los flujos elementales por IP:** Diseño del método de emisión de IPTV por multicast. Definición del formato contenedor y los parámetros de emisión.
- **Recepción de contenidos:** Diseño y desarrollo de un prototipo de recepción del sistema combinado.

3.1. Señalización de los servicios

Para combinar los sistemas de transmisión es necesario definir qué servicios viajan la señal ISDB-Tb tradicional y cuáles a través de la red IP. A su vez, los dispositivos de recepción deben ser capaces de clasificar y obtener los servicios para entregarlos. Usando la infraestructura que actualmente provee la norma ISDB-Tb, sólo es posible señalar servicios que se encuentran en el mismo flujo de transporte.

Como hipótesis de trabajo, la señalización poseerá las siguientes características:

- Será centralizada. La información para construir la lista de servicios y sintonizarlos se proveerá por un único medio, evitando redundancia y problemas de inconsistencia de información.

- Será transparente. El usuario podrá abstraerse del origen de los servicios al consumir alguno de ellos.
- Estará disponible para todos los dispositivos de recepción, independientemente de la posibilidad de acceso a internet o una red.
- Hará uso del sistema de señalización que ya provee la norma ISDB-Tb a través del formato MPEG-TS.

3.1.1. Extensión de ISDB-Tb

De acuerdo a los cuatro puntos anteriores, se busca que la construcción de la lista de servicios incluya todos los servicios sin distinción. Sin embargo, esta lista contendrá dos tipos de servicios. Los servicios **tradicionales** y los servicios **reubicados**. Los primeros son aquellos que soporta por defecto la norma, la segunda categoría denomina a los servicios que están señalizados a través del flujo de transporte ISDB-Tb, pero que el receptor debe ir a buscar a la red IP.

En su estado actual, la norma ISDB-Tb no permite señalar servicios que se encuentran fuera del flujo de transporte, no existe el concepto de servicio reubicado. Por esto, el nuevo soporte requiere una extensión de la norma. Sin embargo, a fin de no rediseñar por completo el formato contenedor, los datos que sea necesario introducir se deben adecuar a algún mecanismo de entrega provisto por MPEG-TS (campos, descriptores o tablas). La figura 3.1 representa esquemáticamente el comportamiento buscado.

El primero objetivo, que es incluir todos los servicios a la lista provista al usuario no requiere el uso de mecanismos ajenos a la norma. Para incluir un servicio reubicado en la lista, basta con introducir una entrada a la SDT, una entrada a la PAT y la aparición de una instancia de la PMT correspondiente al servicio que se desea agregar. Estos tres elementos son suficientes para que Wari lo reconozca. A partir de esto se puede afirmar que el proceso de *scanning* no requiere modificaciones.

El segundo objetivo consiste en entregar al usuario el audio y el video correspondientes al servicio reubicado eventualmente seleccionado. Se dijo que para cumplir el primer objetivo, se requería la existencia de la entrada en la PAT y la instancia de la PMT correspondiente. Entonces, el reproductor encontraría esta entrada en la PAT, con ella ubicaría la PMT, pero fallaría al intentar ubicar los flujos elementales del servicio. Ya sea, porque el *loop* de flujos estaría vacío o bien contendría entradas inválidas. En este punto, es imposible indicar que el audio y el video viajan por un medio distinto que el flujo de transporte. Este es el punto de introducción de la extensión.

Las alternativas para extender el formato son básicamente 3, 1) la adición de un campo a la PMT, 2) la creación de una nueva tabla o 3) la creación de un nuevo descriptor para la PMT. Veremos que la única alternativa real es la tercera.

El primer punto rompe con el objetivo de retrocompatibilidad. Si un reproductor regular intentara analizar los contenidos de la PMT modificada, el proceso de *parsing* probablemente fallaría o, peor, interpretaría erróneamente los contenidos. Con respecto al segundo punto, la norma ISDB-Tb establece que cuando un receptor se encuentra

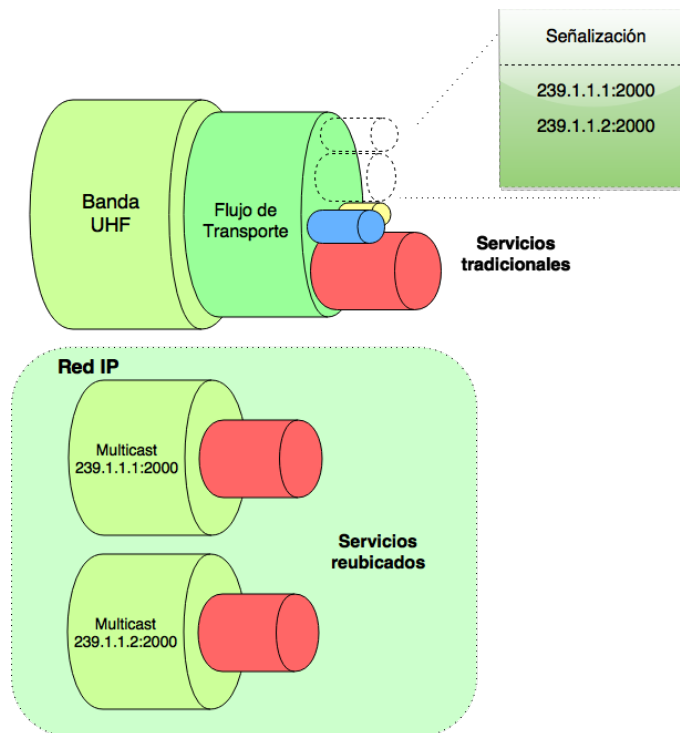


Figura 3.1: Diagrama de un flujo de transporte con señalización a dos servicios reubicados

con alguna entidad que no logra reconocer, debe ignorarla. Esto exige que el flujo de transporte indique de alguna manera la presencia de la nueva tabla(o de las instancias de la nueva tabla, si se utilizara una por servicio), en cuyo caso se vuelve al problema inicial de señalar una nueva entidad en el flujo de transporte. Alternativamente, se podría preprogramar al receptor para asignar un PID a una nueva tabla, haciendo que la PMT no intervenga en absoluto en la sintonización pero esto, nuevamente, rompe diametralmente el objetivo de introducir cambios mínimos.

En resumen, la solución consiste en la creación de un descriptor para la PMT. La información mínima que se necesita que contenga consiste en dos elementos. Una dirección IP que identifica al grupo *multicast* por el que se emite el servicio reubicado, y el puerto al que se dirigen los datagramas. Se lo nombrará *Elementary Streams Relocation Descriptor*.

3.1.2. *Elementary Streams Relocation Descriptor*

Siguiendo el segundo de los principios de diseño **SOLID**(*Open/Closed Principle*)[17], el desarrollo busca evitar modificar el formato MPEG-TS tanto como sea posible. En cambio, intenta extenderlo. Se busca agregar los comportamientos por medio de entidades nuevas en lugar de modificar las existentes. Para señalar nuevos servicios cuyos flujos elementales se deben obtener desde una red IP, se define un nuevo descriptor en concordancia con el formato **MPEG-TS**, de nombre *Elementary Streams Relocation Descriptor*.

El *Elementary Streams Relocation Descriptor* sólo podrá hallarse, de forma opcional, en la **PMT** e implica que el servicio correspondiente a la tabla es un servicio reubicado. El descriptor tendrá asignado el *descriptor_tag* 170 (0xAA en hexadecimal). No existe, actualmente, un descriptor en la norma ISDB-Tb con el mismo nombre o igual *descriptor_tag* o semántica similar. Para que el receptor sea capaz de obtener los flujos elementales, es necesario identificar el grupo multicast y el puerto al que se envían los flujos. Para esto, es necesario proveer a través del descriptor esta información. Y para ello, es menester definir su estructura sintáctica.

La Tabla 3.1 define la sintaxis del descriptor:

Elemento	Rango de valores	Cantidad de bits	Bit de comienzo	mnemotécnico ¹
<i>descriptor_tag</i>	170	8	0	uimsbf ²
<i>descriptor_length</i>	6	8	8	uimsbf
Grupo Multicast	3758096384 – 4026531839	32	16	uimsbf
Puerto	1 – 65535	16	48	uimsbf

Tabla 3.1: *Sintaxis del Elementary Streams Relocation Descriptor*

¹Sigla que identifica al tipo del campo; comúnmente utilizada en documentos de especificación de sintaxis.

²**uimsbf** significa *unsigned integer, most significant bit first*. Lo cual significa *entero sin signo, bit más significativo primero*.

Descripción de los elementos:

- *descriptor_tag*: Elemento requerido por la definición sintáctica de los descriptores MPEG-TS. Lleva el valor 170 para identificar al *Elementary Streams Relocation Descriptor* en el *descriptor loop*.
- *descriptor_length*: Elemento requerido por la definición sintáctica de los descriptores. El *Elementary Streams Relocation Descriptor* tiene una longitud fija de 8 bytes. De modo que este campo siempre lleva el valor 6 (no incluye los primeros dos campos del descriptor).
- **Grupo Multicast**: Grupo Multicast al que se debe unir el receptor para capturar los flujos elementales del servicio al que está asociada la PMT en la que viaja este descriptor. El rango de valores se debe a que las direcciones IP asignadas a los grupos multicast se hallan en el rango 224.0.0.0 – 239.255.255.255. Estas direcciones IP codificadas en *big-endian* (el byte más significativo en la menor dirección de memoria) resultan en esos valores.
- **Puerto**: El puerto destino con el que viajan los datagramas sobre la red. Son dos bytes codificados en *big-endian*. Cabe notar que el valor 0 representa un número de puerto inválido.

A modo de ejemplo, si se desea indicar que los flujos elementales de un servicio viajan por el grupo multicast 239.1.1.1 y puerto 10000, se debe incluir el siguiente *Elementary Streams Relocation Descriptor* en la **PMT** del servicio. Nótese que el descriptor tiene una longitud fija de 8 bytes, contando campos obligatorios de *descriptor_tag* y *descriptor_length*.

Elemento	Significado del contenido	Contenido real(Decimal)	Contenido real (Hexadecimal)
<i>descriptor_tag</i>	170	170	0xAA
<i>descriptor_length</i>	6	6	0x06
Grupo Multicast	239.1.1.1	4009820417	0xEF010101
Puerto	10000	10000	0x2710

Tabla 3.2: Ejemplo de uso del Elementary Streams Relocation Descriptor

Byte	0	1	2	3	4	5	6	7
Valor del byte	0xAA	0x06	0xEF	0x01	0x01	0x01	0x27	0x10

Tabla 3.3: Volcado de contenido (hexadecimal) del ejemplo de uso del Elementary Streams Relocation Descriptor

En conclusión, la extensión diseñada consiste en un descriptor que agrega semántica a la norma ISDB-Tb. Para indicar que un servicio posee sus flujos elementales emitidos por una red IP, alcanza con insertar el *Elementary Streams Relocation Descriptor* al *loop* de descriptores de su **PMT**.

3.2. Flujo de transporte de referencia

El flujo de transporte de una señal ISDB-Tb es una secuencia de bits organizada por el formato MPEG-TS. En el caso de la emisión de televisión digital, parte del proceso se realiza en tiempo real, al momento de multiplexar varios servicios que provienen de distintos orígenes, para luego modular la señal a emitir por el espectro radioeléctrico.

En el caso del flujo de transporte de referencia para este desarrollo, éste se volcará en un archivo para su persistencia. Los flujos de transporte se persisten representando los paquetes en secuencia, donde cada uno es un fragmento de 188 bytes, que se podría persistir aisladamente, de ser necesario. Así, una tabla que ocupa un único paquete se persiste como un archivo de 188 bytes.

Esta sección se divide en dos partes. En la primera se expone la extensión realizada a OpenCaster, requerida para representar el *Elementary Streams Relocation Descriptor*. En la segunda se desarrolla paso a paso la creación del flujo de transporte de referencia, junto con lo requerido para la construcción de otros flujos de transporte.

3.2.1. Extensión de OpenCaster

Para facilitar la construcción del nuevo descriptor, se introduce una extensión a OpenCaster. La versión actual del proyecto ya cuenta con funcionalidad para representar la mayoría de las entidades de la norma ISDB-Tb. La extensión le permite a las entidades de MPEG-TS modeladas reconocer el descriptor *Elementary Streams Relocation Descriptor*, como si fuera parte del estándar, para así incluirla en la **PMT**, de igual manera que con cualquier otro descriptor. La figura 3.2 incluye un diagrama de clases que explica la jerarquía de descriptores de OpenCaster, incluyendo el nuevo descriptor.

OpenCaster utiliza una jerarquía de clases para modelar las entidades **MPEG-TS** que se incluyen en un flujo de transporte. La extensión consiste en la creación de la clase `elementary_streams_relocation_descriptor`, que hereda de `Descriptor`, como se muestra en figura 3.2. El *Elementary Streams Relocation Descriptor* se modela con una clase con dos variables de instancia. Una asociada al grupo multicast y la segunda asociada al puerto. Sobreescribiendo el método `bytes`, que hereda de la clase `Descriptor`, las instancias se empaquetan en la estructura sintáctica definida en la primer sección. En el fragmento 3.3 se proveen ejemplos de utilización de esta clase.

3.2.2. Extensión del flujo de transporte de Canal 23

A fin de evitar construir un flujo de transporte completo, se continuará sobre el ejemplo de canal 23, al cual se le incorporarán dos servicios. Similar a la figura 2.4, la

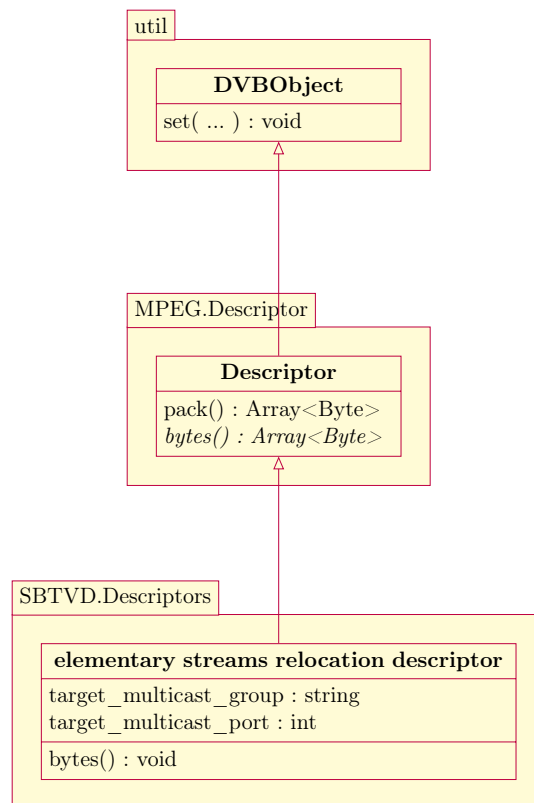


Figura 3.2: Diagrama de Clases de OpenCaster con el nuevo descriptor

figura 3.3 expande el cuadro y presenta un esquema del flujo de transporte resultante esperado.

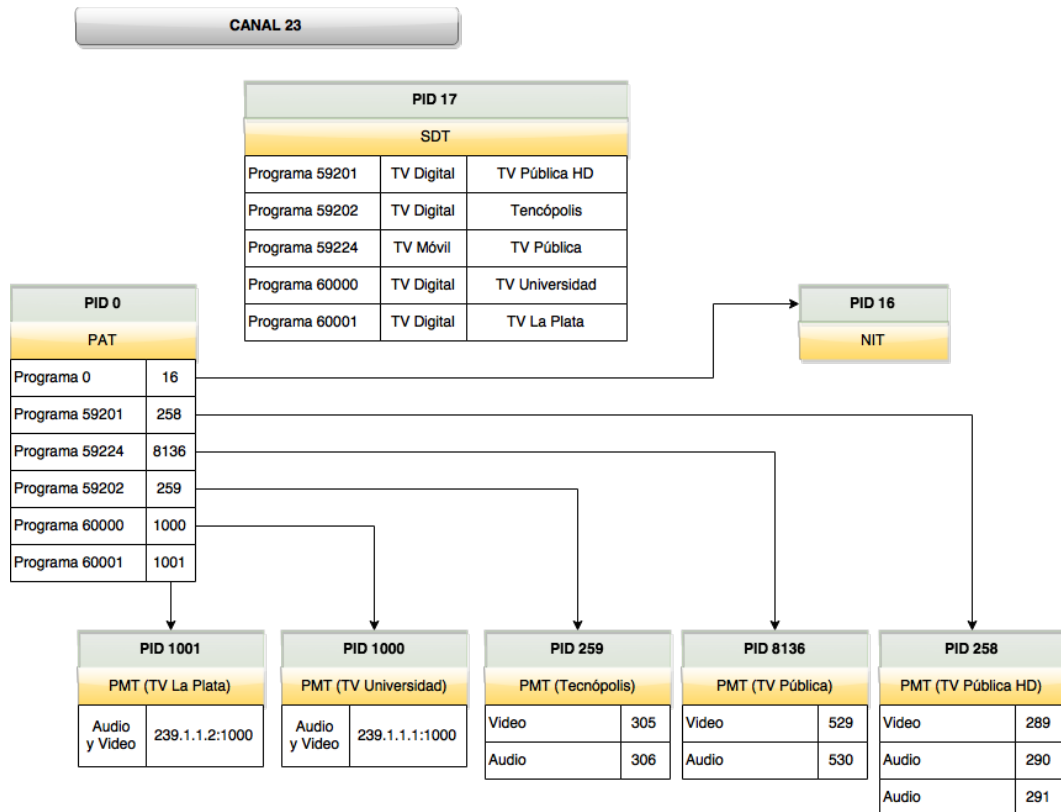


Figura 3.3: Esquema del flujo de transporte de Canal 23 extendido

El flujo de transporte base, ya conocido, posee tres servicios: “TV Pública HD”, “Tecnópolis” y “TV Pública”. A éstos se le añadirán dos, cuyos flujos elementales serán emitidos a través de dos grupos multicast distintos. Para empezar es importante definir cuáles son los nuevos servicios que se van a agregar. A fines ilustrativos, llamaremos a estos servicios “TV Universidad” y “TV La Plata”. Los pasos de adición de los servicios se explican a continuación.

Adición de servicios a SDT

Por cada servicio que se busca agregar a un flujo de transporte, es necesario agregar un *service_descriptor* a la **SDT**. Éste descriptor posee la siguiente información:

1. *service_id* del servicio.
2. Nombre del servicio.


```

29     ],
30   ),
31
32   service_loop_item(
33     service_ID = SERVICE_LA_PLATA_ID,
34     EIT_schedule_flag = 0,
35     EIT_present_following_flag = 0,
36     running_status = 0,
37     free_CA_mode = 0,
38     service_descriptor_loop = [
39       service_descriptor(
40         service_type = 1,
41         service_provider_name = "",
42         service_name = "TV La Plata",
43       ),
44     ],
45   ),
46 ],
47 version_number = 2,
48 section_number = 0,
49 last_section_number = 0,
50 )

```

Adición de servicios a la PAT

La **PAT** les asigna **PID**'s a las **PMT** de los servicios del flujo de transporte. Habiendo ya definido los *service_id*, sólo resta definir el valor numérico de dichos **PID**. En este caso, es importante cuidar que no se utilice un número de **PID** reservado o utilizado por alguna otra componente del flujo. Se incluye una lista de **PIDs** reservados en ARIB-STD B10[9]. Los **PID** asignados a las **PMT** serán 1000 para “TV Universidad” y 1001 para “TV La Plata”.

Fragmento 3.2: Creación de PAT en OpenCaster

```

1 pat = program_association_section(
2   transport_stream_id = TS_ID_ORIG,
3   program_loop = [
4     program_loop_item(
5       # Programa especial para la tabla NIT
6       program_number = 0,
7       PID = 16,
8     ),
9     program_loop_item(
10      program_number = SERVICE_1_ID,
11      PID = SERVICE_1_PMT_PID,
12    ),
13    program_loop_item(
14      program_number = SERVICE_2_ID,

```

```

15     PID = SERVICE_2_PMT_PID ,
16   ),
17   program_loop_item(
18     program_number = SERVICE_3_ID ,
19     PID = SERVICE_3_PMT_PID ,
20   ),
21   program_loop_item(
22     program_number = SERVICE_UNIVERSIDAD_ID ,
23     PID = SERVICE_UNIVERSIDAD_PMT_PID ,
24   ),
25   program_loop_item(
26     program_number = SERVICE_LA_PLATA_ID ,
27     PID = SERVICE_LA_PLATA_PMT_PID ,
28   ),
29 ],
30 version_number = 3 ,
31 section_number = 0 ,
32 last_section_number = 0 ,
33 )

```

Inserción de PMT's nuevas

Como son dos los servicios a agregar al flujo de transporte, es necesario crear dos **PMT**. Una por cada uno. Las **PMT** de los servicios que ya existen se conservarán intactos.

Ya se definieron los **PID** y *service_id* de los servicios nuevos. El único elemento restante es el *Elementary Streams Relocation Descriptor* (siendo que el *loop* de flujos elementales se mantiene vacío). Éste, a su vez, debe incluir grupo multicast y puerto por los que se emiten los flujos elementales. El uso del nuevo descriptor se puede observar en las líneas 5 y 17 del fragmento 3.3. Se asigna el grupo 239.1.1.1 a “TV Universidad” y 239.1.1.2 a “TV La Plata”. Ambos por puerto 1000.

Fragmento 3.3: Creación de PMT's en OpenCaster

```

1 pmt_universidad = program_map_section(
2   program_number = SERVICE_UNIVERSIDAD_ID ,
3   PCR_PID = 8191 ,
4   program_info_descriptor_loop = [
5     elementary_streams_relocation_descriptor(
6       target_multicast_group="239.1.1.1", target_multicast_port
7       =1000)
8   ],
9   stream_loop = [],
10  version_number = 0 ,
11  section_number = 0 ,
12  last_section_number = 0 ,
13 )

```

```

13 pmt_la_plata = program_map_section(
14     program_number = SERVICE_LA_PLATA_ID,
15     PCR_PID = 8191,
16     program_info_descriptor_loop = [
17         elementary_streams_relocation_descriptor(
18             target_multicast_group="239.1.1.2", target_multicast_port
19             =1000)
20     ],
21     stream_loop = [],
22     version_number = 0,
23     section_number = 0,
24     last_section_number = 0,
25 )

```

Multiplexación

Finalmente, sólo resta multiplexar el nuevo flujo de transporte. Una posibilidad es extraer todos los servicios del flujo de transporte y remultiplexarlo, intercalando las nuevas tablas en el intervalo adecuado. Este complejo proceso es necesario cuando los servicios poseen un *bitrate* necesario no despreciable. Sin embargo existe una solución más simple.

Las tablas generadas con los script incluidos poseen la característica de ocupar un único paquete. Por eso, es posible reemplazar las antiguas **SDT** y **PAT** por las nuevas, generadas con OpenCaster. La **PMT** por otro lado, introduce un inconveniente. Las dos nuevas **PMT** (de los servicios “TV Universidad” y “TV La Plata”) no se encuentran en el flujo de transporte original, e insertarlas en el medio, implica desplazar todos los paquetes siguientes una posición cada vez que se inserta. Esto incurre en una modificación del *bitrate* y la consecuente degradación de los servicios.

Con el objetivo de mantener exactamente el mismo *bitrate*, se pueden ubicar en el lugar de paquetes nulos, reemplazándolos. Para hacer la inserción, es necesario conocer el *bitrate* de emisión del flujo de transporte y, a su vez, calcular la posición de las inserciones a realizar, dado que la **PMT** debe aparecer en un período de 200 milisegundos.

Para realizar estas tareas se utilizará la librería *ts_util*[15] desarrollada específicamente para este trabajo. En la multiplexación intervienen `packet_replacer` (reemplaza los paquetes de un flujo de transporte de un PID definido por otro) y `packet_inserter` (inserta un paquete en un flujo de transporte en el lugar de un paquete nulo, a fin de que aparezca en un período dado). El fragmento 3.4 incluye las líneas utilizadas en la multiplexación final del flujo de transporte de referencia.

Fragmento 3.4: Multiplexación a través de ts_util

```

1 packet_replacer.out -s frec_527.ts -o frec_527_with_pat.ts -p pat.
   ts
2 packet_replacer.out -s frec_527_with_pat.ts -o
   frec_527_with_pat_and_sdt.ts -p sdt.ts

```

```
3 packet_inserter.out -s frec_527_with_pat_and_sdt.ts -o
  frec_527_with_pat_sdt_and_uni_pmt.ts -p pmt_universidad.ts -i
  190
4 packet_inserter.out -s frec_527_with_pat_sdt_and_uni_pmt.ts -o
  frec_527_final.ts -p pmt_la_plata.ts -i 190
```

El objetivo de este trabajo es vincular la señal de ISDB-Tb con las redes IP. El nuevo descriptor zanja la brecha de identificar un grupo multicast dentro de esta red. Sin embargo, para que esta señalización tenga sentido, es necesario que efectivamente exista una emisión audiovisual a través de la red, por el grupo y hacia el puerto determinados. La sección siguiente resuelve el problema de la transmisión de servicios a través de redes IP.

3.3. Emisión multicast de los flujos elementales por IP

La emisión por multicast IP consiste en el envío de datagramas a múltiples receptores sin la necesidad para el emisor de enviar más de una copia. Hay varios problemas a resolver en este aspecto del trabajo. El más importante de ellos es decidir el formato contenedor con que se envían los contenidos audiovisuales. El formato elegido será el flujo de transporte MPEG por los siguientes motivos:

- La infraestructura disponible en los reproductores de televisión existentes garantiza soporte exclusivamente al formato **MPEG Transport Stream**. La necesidad de cambiar este aspecto iría en contra del objetivo de minimizar los cambios necesarios.
- Siendo que los envíos se realizan sobre el protocolo UDP, no existe *a priori* protección ante pérdida de paquetes, responsabilidad que recae sobre capas superiores. El flujo de transporte fue diseñado específicamente para este tipo de entornos.
- El formato MPEG-TS es extremadamente flexible, de modo que permite escalar sobre el mismo diseño hacia características nuevas en un trabajo futuro.
- La división en paquetes de 188 bytes resulta adecuada para la división en datagramas de red IP.

A pesar de las motivaciones de la elección, existen algunas diferencias con el uso que se le da en televisión digital terrestre:

- El bitrate de emisión de una red IP no es estrictamente constante, de modo que los paquetes nulos incurren en un desperdicio de ancho de banda.
- Por el mismo motivo, los paquetes no se envían en un flujo constante, si no que se envían en ráfagas. La emisión debe evitar el *overflow*, es decir que el ritmo de envío se debe adecuar al *bitrate* del servicio y no al ancho de banda disponible.

- En el caso de la televisión digital terrestre, cada canal tiene un bitrate independiente y se busca aprovechar cada uno al máximo. En el caso de la transmisión multicast, cada servicio se puede enviar en un grupo distinto, para ahorrar el ancho de banda utilizado por cada usuario. Por esto, la multiprogramación no tiene sentido en un medio flexible como la red IP.

La figura 3.4 presenta un esquema similar a la figura 3.1, con el agregado de emitir los flujos elementales en formato MPEG-TS y utilizando el *Elementary Streams Relocation Descriptor* para la señalización de los servicios reubicados.

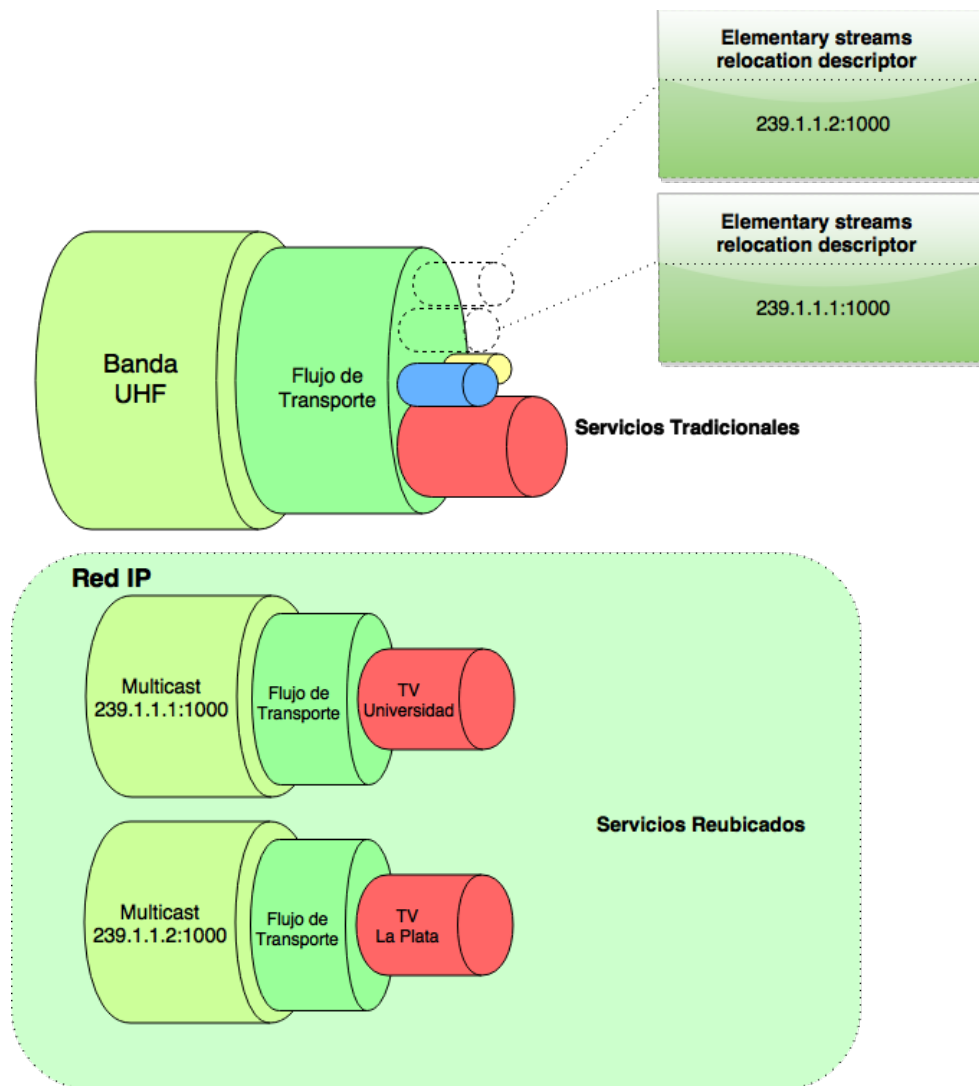


Figura 3.4: Diagrama de un flujo de transporte con señalización a dos servicios, emitidos con formato MPEG-TS.

Existe un gran número de herramientas para la realización de la emisión de los flujos elementales. En este trabajo se ha utilizado la herramienta `packet_streamer`, también incluida en el proyecto `ts_util`[15], creado con el propósito de asistir en el desarrollo del prototipo de sistema descrito en este informe.

Los flujos de transporte emitidos a través de redes IP cargarán un único servicio con un único flujo elemental de audio y un único flujo elemental de video. Además, no contendrán paquetes nulos, por razones ya expuestas. Sin embargo, es necesario emular un *bitrate* constante en la implementación del software de emisión.

Si los paquetes del flujo de transporte se enviaran al ritmo máximo que permitiera la infraestructura, el receptor eventualmente sufriría un *overflow*. Es deseable aprovechar la máxima unidad de transmisión permitida por la red sin fragmentación de paquetes (tamaño máximo permitido de un datagrama, ejemplo: 1500 *bytes* en *Ethernet*), para minimizar el *overhead* generado por el proceso de *routing* y los *headers* de los datagramas.

La función f incluida a continuación define la cantidad de paquetes que deben haber sido enviados hasta el instante t .

$$f(t) = \frac{t[\text{seg}] * b[\text{bits/seg}]}{1504[\text{bits/pkt}]} + v[\text{bytes}] \text{ div } 188[\text{bytes/pkt}]$$

Donde t es un instante en segundos. b es el *bitrate* (*bits/segundo*) original del flujo de transporte a emitir. 1504 es la cantidad de bits en un paquete. v es la máxima unidad de transmisión de la red en bytes. Y 188 es el tamaño de un paquete de flujo de transporte en bytes.

La evaluación de f retorna una cantidad de paquetes que deben haber ser enviados para un instante t , contando paquetes nulos. La emisión debe ignorarlos, pero llenar la ventana tanto como sea posible. El fragmento 3.5 incluye un pseudocódigo que hace uso de f para explicar el procedimiento de emisión de paquetes, respetando el *bitrate* e ignorando paquetes nulos. Además realiza una lectura circular, al terminar el archivo, la emisión comienza nuevamente.

Fragmento 3.5: Pseudo código de emisión IP

```

1 iterador := 0
2 MIENTRAS( VERDADERO )
3   SI ( iterador = paquetes_en_flujo )
4     iterador := 0
5   SI ( iterador < f( instanteActual() ) )
6     d := obtenerNuevoDatagrama()
7     MIENTRAS( d.entranMasPaquetes() y iterador < paquetes_en_flujo )
8       SI ( flujo[iterador].noEsPaqueteNulo() )
9         d.agregarPaquete[flujo[iterador]]
10        iterador := iterador + 1
11        d.enviarPorRed()

```

3.4. Recepción de contenidos

A continuación se revisan las componentes necesarias para desarrollar un sistema de transmisión de televisión digital:

- Una señal ISDB-Tb extendida modulada sobre el espectro radioeléctrico. La infraestructura de hardware y software requerida para esta tarea no cambia respecto a una emisión ISDB-Tb común. Sólo cambia el flujo de transporte a emitir.
- Emisión de flujos elementales de los servicios a través de un grupo multicast en una red IP manejada.

- Un dispositivo de software y hardware capaz de recibir la señal ISDB-Tb, y capturar los flujos elementales desde la red IP disponible cuando el usuario sintoniza un servicio que lo requiere.

La última componente listada concierne a la recepción, y es la única que resta desarrollar en este trabajo. A continuación se introduce *Mara*, una modificación de *Wari* que es capaz de consumir el *Elementary Streams Relocation Descriptor* y entregar la lista extendida completa de servicios de un flujo de transporte con el descriptor.

El desarrollo de *Mara* se ubica sobre *Wari* aprovechando la absoluta mayoría de funcionalidades que no sufren cambios respecto a ISDB-Tb tradicional. Específicamente, *Mara* se comporta de manera idéntica a *Wari* cuando consume un flujo de transporte regular.

Si *Wari* recibiera un flujo de transporte que incluye el *Elementary Streams Relocation Descriptor*, sería capaz de construir la lista de servicios, pero no sería capaz de reproducir los flujos elementales. Entonces, la etapa final de este desarrollo es construir un prototipo de software de recepción que sea capaz de capturar el flujo de transporte extendido y reproducirlo. En resumen, los pasos a realizar por el reproductor son:

- Construir la lista de servicios disponibles.
- Identificar aquellos que posean el *Elementary Streams Relocation Descriptor* en su PMT.
- Asociar el servicio sintonizado por el usuario a los flujos elementales apropiados:
 - Si la **PMT** del servicio no posee el descriptor agregado, el receptor se debe comportar de manera corriente.
 - Si la **PMT** sí posee el descriptor asociado, entonces debe unirse al grupo multicast indicado en el mismo y capturar los flujos elementales desde ese medio.

La figura 3.5 presenta un diagrama de flujo para la reproducción de servicios con la inclusión de la nueva funcionalidad.

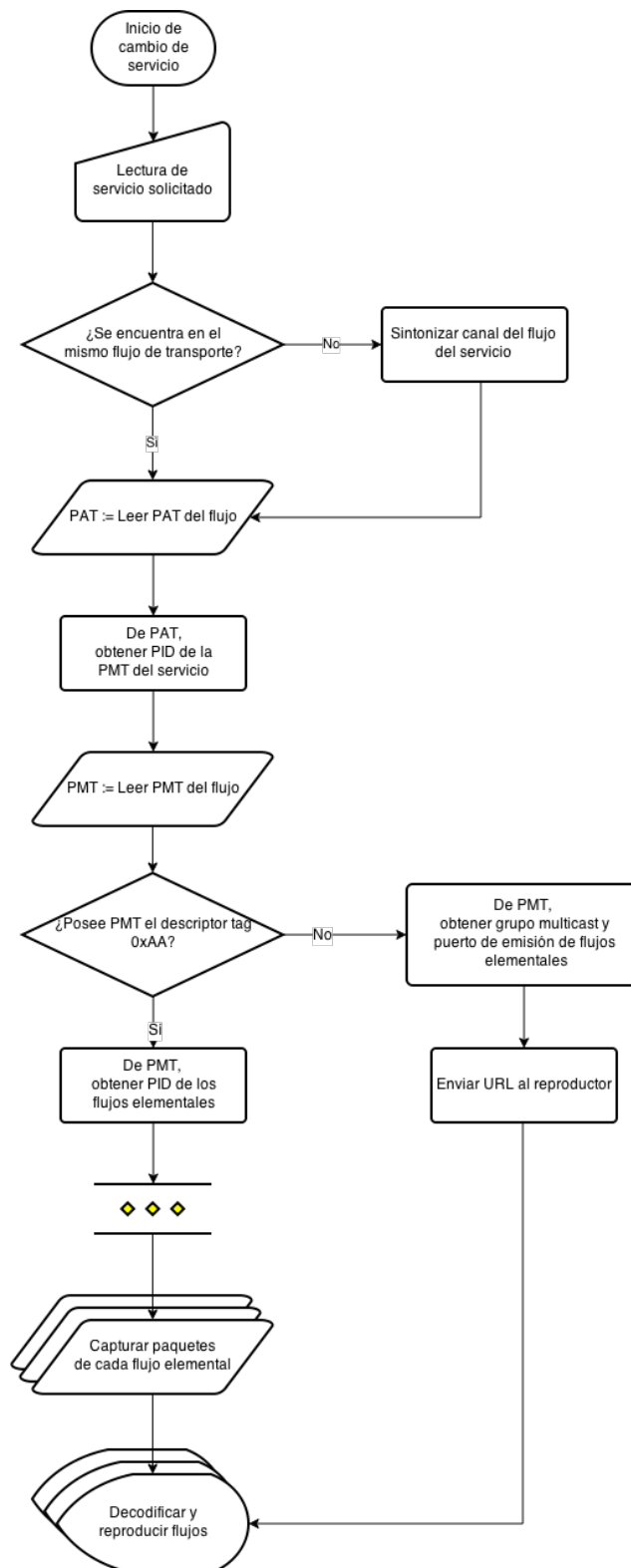


Figura 3.5: Diagrama de flujo modificado para la recepción del Elementary Streams Relocation Descriptor.

3.4.1. Construcción de Mara

Las modificaciones a realizar sobre el reproductor *Wari* se agrupan en tres responsabilidades distintas:

1. **Modificación del modelo de servicios:** Para empezar, Mara debe modelar los servicios reubicados, lo cual es imposible en Wari. Para esto, se debe extender la lista de atributos de la clase `Service`.
2. **Parsing del descriptor:** El sistema debe ser capaz de capturar el descriptor, identificarlo y obtener sus contenidos. Estos contenidos, luego, deben verse reflejados en el estado del sistema, sobre las modificaciones del ítem anterior.
3. **Reproducción:** Finalmente, cuando el usuario sintoniza un servicio, el contenido audiovisual de la reproducción debe ser adquirido del medio correcto, usando la información guardada en el modelo de la clase `Service`.

Modificación del modelo de servicios

El diagrama de clases presentado en la figura 2.9 presenta el modelo de los servicios de Wari, a través de la clase `Service`. Es necesario agregar 3 atributos para representar de manera simple la característica de los servicios de poseer flujos elementales fuera del flujo de transporte. Esto se refleja en la figura 3.6.

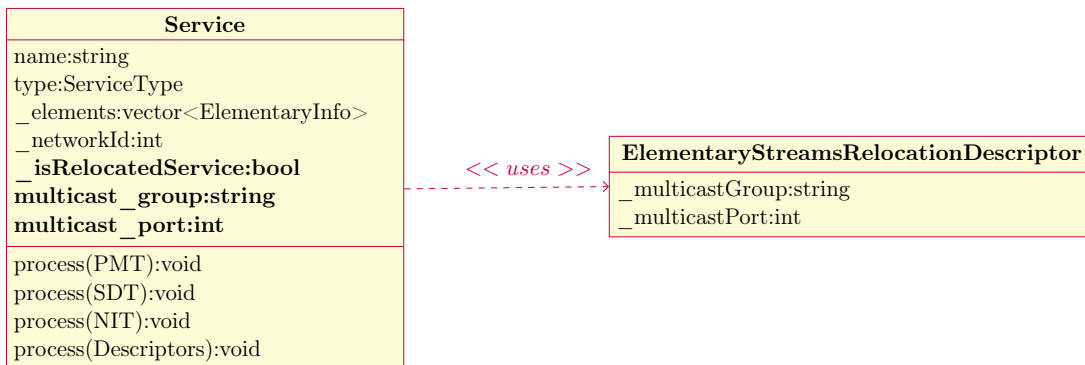


Figura 3.6: Modelo del servicio en Mara

En la figura se resaltan las tres nuevas variables de instancia de la clase.

- `_isRelocatedService`: Valor *booleano*³ que establece si los flujos elementales del servicio deben obtenerse desde una red IP o no.

³Tipo que posee únicamente dos valores posibles que intentan denotar los valores lógicos de verdad y falsedad, generalmente denominados *verdadero* (o *true*) y *falso* (o *false*).

- `_multicastRelocatedGroup`: Dirección IP que identifica el grupo multicast de la emisión de flujos elementales pertenecientes al servicio.
- `_multicastRelocatedPort`: Puerto destino de los datagramas UDP por los que viajan los paquetes TS⁴ que transportan las partes de los flujos elementales.

Parsing del descriptor

En detalle, el *parsing* de descriptors en Mara implica los siguientes cambios en Wari:

- Cuando en el *loop* de descriptors se halla el *descriptor_tag* 170, se debe invocar a la función que interpreta el contenido.
- Este contenido se debe invocar a un modelo en memoria del descriptor, un registro.
- El servicio debe ser modificado acordemente, modificando su estado a partir del registro que representa al descriptor.

El fragmento 3.6 es utilizado en Mara para obtener una instancia del *Elementary Streams Relocation Descriptor* a partir del flujo de transporte sintonizado. El macro `DESC_PARSE(descriptors, elementary_streams_relocation, d)` genera la invocación del *parser* del descriptor, incluido en el fragmento 3.7.

Fragmento 3.6: Captura de Elementary Streams Relocation Descriptor en Mara

```

1
2 bool Service::process(const desc::Descriptors &descriptors) {
3     bool changed = false;
4
5     // ...
6     // Se excluye la captura de otros descriptors
7
8     {
9         desc::ElementaryStreamsRelocationDescriptor d;
10        if(DESC_PARSE(descriptors, elementary_streams_relocation, d)){
11            _isRelocated = true;
12            _multicastPort = d.port;
13            _multicastGroup = d.ipMulticast;
14        }
15    }
16    // Save descriptors
17    _descriptors += descriptors;
18
19    return changed;
20 }

```

El fragmento 3.7 incluye el código que interpreta los contenidos binarios del descriptor y realiza las conversiones léxicas necesarias. Construye la instancia del descriptor

⁴Denominación común a los paquetes de un flujo de transporte.

que posteriormente se utiliza para modificar el modelo del servicio, representado en la figura 3.6.

Fragmento 3.7: Parsing de Elementary Streams Relocation Descriptor en Mara

```

1 bool fnc_elementary_streams_relocation(
2     ElementaryStreamsRelocationDescriptor &desc,
3     BYTE *descPayload){
4     DTV_ASSERT(DESC_TAG(descPayload)
5         == ident::elementary_streams_relocation );
6     std::string ip;
7     using boost::lexical_cast;
8     using std::string;
9     ip += lexical_cast<string>(static_cast<int>(descPayload[2]));
10    ip += ".";
11    ip += lexical_cast<string>(static_cast<int>(descPayload[3]));
12    ip += ".";
13    ip += lexical_cast<string>(static_cast<int>(descPayload[4]));
14    ip += ".";
15    ip += lexical_cast<string>(static_cast<int>(descPayload[5]));
16    desc.ipMulticast = ip;
17    desc.port = (descPayload[6] << 8) + descPayload[7];
18    LTRACE("desc",
19        "Elementary Streams Relocation Descriptor found. IP = %s
20        PORT = %d",
21        ip.c_str(), desc.port);
22    return true;
}

```

Reproducción

Recolectar los flujos elementales implica solicitar al dispositivo de sintonización que filtre los paquetes de acuerdo a su **PID**. No obstante, esto corresponde sólomente cuando es necesario demultiplexar servicios de un flujo e interviene un dispositivo sintonizador. En el caso de recepción de un flujo por IP, este paso puede omitirse.

A partir de los **PID** filtrados, las clases de demultiplexación construyen un URL de reproducción. Cuando se halla el *Elementary Streams Relocation Descriptor*, esta URL debe ser reemplazada por la obtenida desde el descriptor, agregando el **esquema**⁵. La figura 3.8 explica cómo cambia el diagrama de secuencia en Mara, respecto de wari, mientras que el fragmento 3.8 incluye el código de construcción de la URL de reproducción, a partir de la información del modelo de la clase **Service**.

⁵Indicador de protocolo al comienzo del URL. Ejemplo: `udp://192.1.1.1:1000`. El esquema de la url es `udp`.

Fragmento 3.8: Construcción de URL de reproducción en Mara

```

1 if (extension()->currentlyPlaying->isRelocated){
2   _url = std::string("udp://@") + extension()->currentlyPlaying->
3     multicastGroup + std::string(":") +
4     boost::lexical_cast<std::string>(extension()->
5       currentlyPlaying->multicastPort);
6 }else{
7   util::Url url;
8   extension()->makeURL( "srvidtv", &url );
9   _url = url.id();
10 }

```

Finalmente, la figura 3.9 es una captura de pantalla de Mara en funcionamiento. En la imagen se pueden ver los servicios “TV Universidad” y “TV La Plata” conformando la lista de servicios disponible. El servicio sintonizado es “TV Universidad”, que está siendo consumido desde multicast.



Figura 3.9: Captura de pantalla de Mara consumiendo el flujo de transporte de referencia

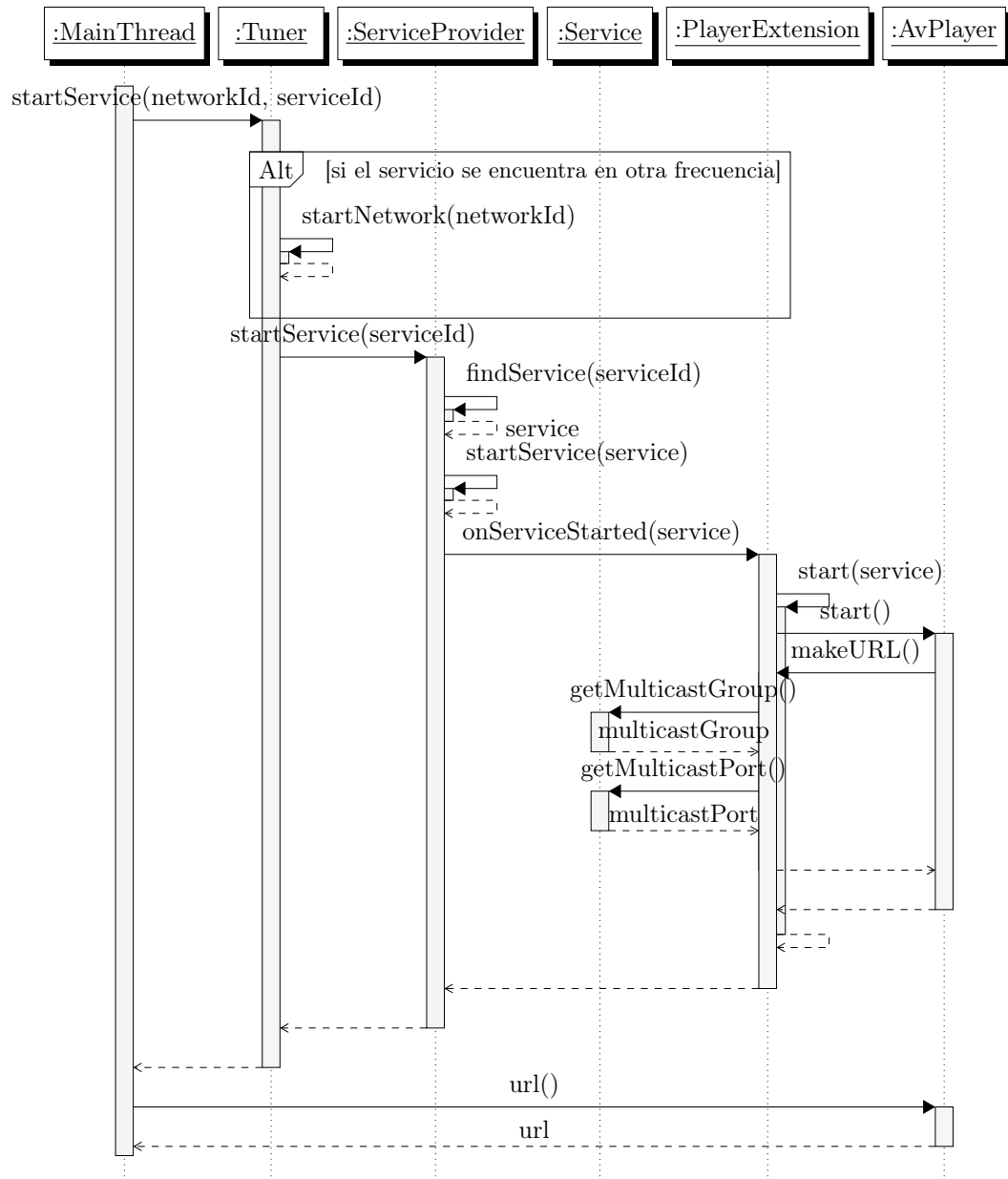


Figura 3.8: Diagrama de secuencia de infraestructura de reproducción en Mara, para un servicio con Elementary Streams Relocation Descriptor

Capítulo 4

Evaluación

Este capítulo expone las pruebas realizadas sobre el sistema desarrollado y evalúa su comportamiento frente a los casos de uso planteados. El objetivo es estudiar cómo mejora el formato extendido sobre el estándar ISDB-Tb tradicional. Asimismo, se exponen las limitaciones del nuevo diseño.

4.1. Infraestructura

La figura 4.1 representa la infraestructura involucrada en la emisión de servicios a través de radiofrecuencia y una red IP.

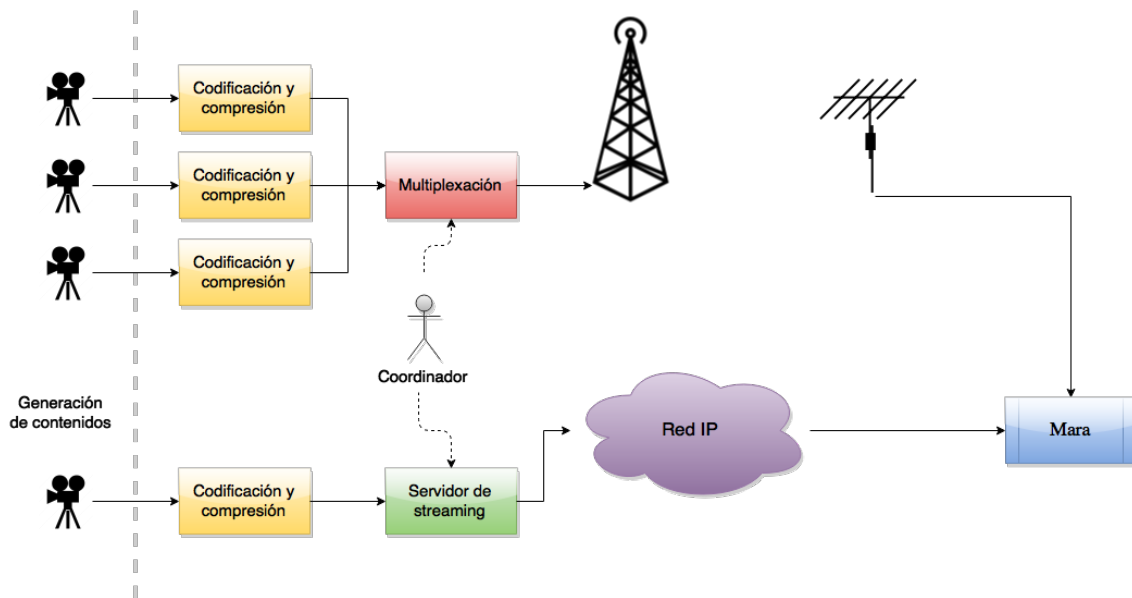


Figura 4.1: Infraestructura utilizada en la emisión de servicios con el esquema extendido.

En adición a los elementos comunes de televisión digital en la norma ISDB-Tb tradicional, el nuevo esquema necesita de dos entidades al final del *pipeline* de producción de contenidos para soportar el *streaming* a través de una red IP: el **servidor de streaming** y un **coordinador**.

Servidor de streaming

El servidor de *streaming* consiste en un *host* con conexión a la red IP. Su responsabilidad es emitir el contenido audiovisual del servicio reubicado que le corresponde. Los paquetes deben ser dirigidos al grupo y puerto multicast indicados en la información provista a través del *elementary streams relocation descriptor* de la norma ISDB-Tb extendida.

Coordinador

Para garantizar que la información provista por el nuevo descriptor y la información de *multicasting* sean congruentes, se requiere esta nueva entidad. El *coordinador* decide qué servicios se emiten en la señal ISDB-Tb de forma tradicional y cuáles por la red IP. Luego, define qué grupo multicast y puerto se usan para la emisión. Con esta información se completa el *elementary streams relocation descriptor* y se configura el servidor de *streaming* multicast.

4.1.1. Infraestructura de prueba

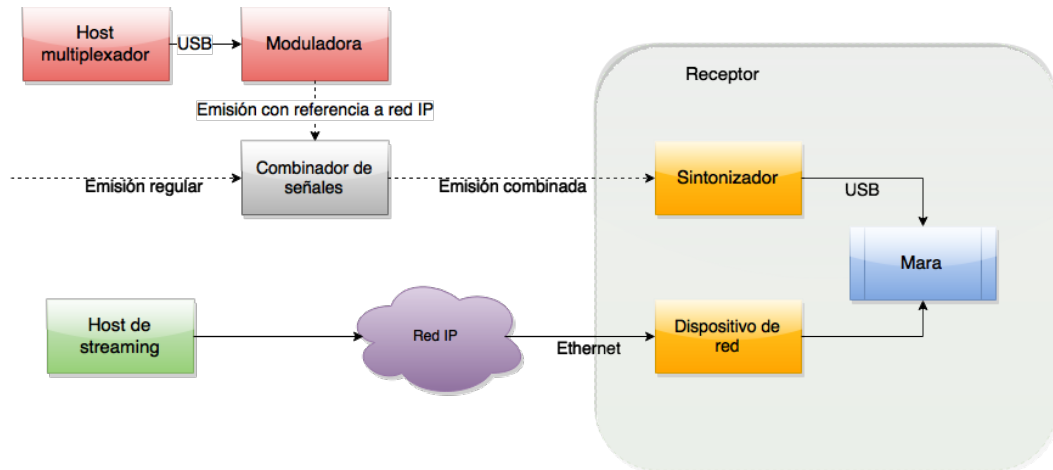


Figura 4.2: Esquema de las pruebas

Para las pruebas realizadas cuyos resultados se exponen en este capítulo se utilizó un banco de trabajo representado en la figura 4.2. Las entidades que pueden observarse son:

- **Host Multiplexador:** Controla la moduladora y provee el *transport stream* extendido que se emite a través de la señal modulada. En este punto se introducen los elementos necesarios para señalar los servicios que se obtienen a través de la red IP, identificada por la dirección IP asignada por el coordinador. Por supuesto, en un escenario ficticio como este, la coordinación se trata de configurar correctamente descriptores y *streaming* únicamente.
- **Moduladora:** Convierte el *transport stream* en una señal radioeléctrica modulada.
- **Combinador de señales:** Es un dispositivo simple que mezcla la señal modulada que transporta el *transport stream* generado y una señal regular de Televisión Digital Abierta. El objetivo de esta combinación es estudiar la convivencia del flujo de transporte generado con otros tradicionales.
- **Sintonizador:** El sintonizador es la contrapartida de la moduladora, convierte una señal modulada en un *bitstream* interpretable por *Mara*.
- **Mara:** Computadora que corre el reproductor de televisión y entrega los servicios al usuario, consumiendo de una u otra fuente de manera transparente.
- **Red IP:** Red con soporte de la *suite* de protocolos IP, incluyendo emisión multicast.
- **Host de *streaming*:** Emite el flujo de transporte que incluye el servicio reubicado a través del grupo indicado por el coordinador.

4.1.2. Análisis del flujo de transporte como formato contenedor en redes IP

Una emisora ISDB-Tb modula una señal que puede ser directamente interpretada por un decodificador. Desde el punto de vista del reproductor de TV, la información que arriba por el dispositivo de sintonización es una secuencia de bytes, no existen para él los aspectos específicos de modulación involucrados. Esta secuencia de bytes recibida se interpreta directamente como un flujo de transporte. Notablemente, este formato provee todas las capas de abstracción necesarias para la comunicación, solucionando problemas como sincronización, multiplexación y corrección de errores.

El caso de las redes IP es radicalmente distinto. Las mismas se basan en el intercambio de paquetes a través de medios indeterminados. Para abstraer los detalles propios de implementación de la comunicación, las redes IP respetan un modelo de capas. El término IP refiere solo a una de estas capas. Pero la información útil que arriba a un *host* no abarca el total recibido, sino que se obtiene luego de un desempaqueamiento que elimina las cabeceras necesarias por las capas inferiores. Entonces, si se asume al flujo de transporte como el punto de vista de carga útil, ISDB-Tb no sufre *overhead*. Las redes IP, en cambio, sí lo hacen. El objetivo de esta sección es analizar cómo influye el *overhead* de los servicios reubicados.

Existe un aspecto negativo de utilizar el flujo de transporte a través de una red IP, como ocurre en los servicios reubicados. Por la duplicidad de capas de abstracción de

comunicación mencionada, el *overhead* es en algunos casos redundante. Por ejemplo, los mecanismos de sincronización que provee el flujo de transporte no son necesarios en las redes IP si se alinean los paquetes. También, tanto el *transport stream* MPEG como el protocolo UDP proveen *checksum*¹.

En términos de la figura 4.1, el enlace entre el servidor de *streaming* y la red IP resulta un eslabón clave en el esquema. Especialmente si se trata de un servidor que debe transmitir varios servicios por la red. La sobrecarga de esta vía implicaría que se vean afectados todos los servicios que emite en caso de alcanzar su límite de ancho de banda. Por el contrario, es más extraño encontrar sobrecargas sobre los enlaces de *última milla*². Se puede asumir que estos enlaces van a transportar un pequeño número de servicios simultáneamente, probablemente uno.

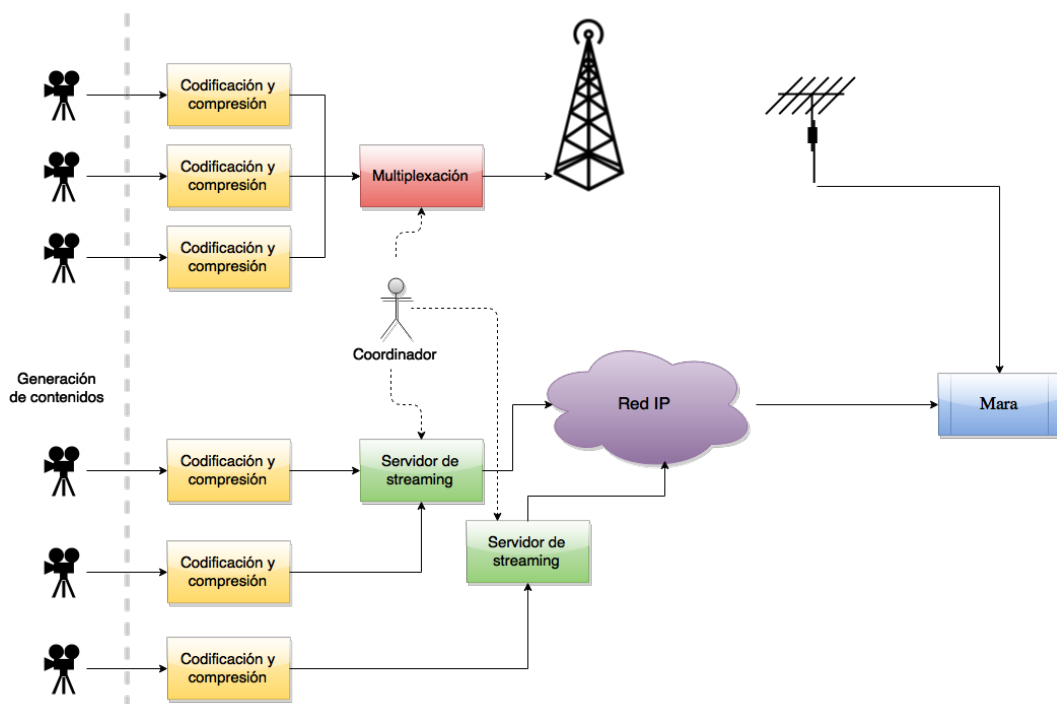


Figura 4.3: Infraestructura con múltiples servidores de streaming

La figura 4.3 ilustra el caso en que los servicios se distribuyen a través de múltiples servidores de *streaming*. En este caso, el cuello de botella podría hallarse en alguno de los

¹Si bien podría argumentarse que esto mejora la calidad de control de errores, la realidad es que viola la razón fundamental detrás del modelo de capas.

²Término utilizado en telecomunicaciones para referir al último extremo de la red que entrega conexión a los clientes.

enlaces *backbone*³ si se tratara de una arista de corte⁴. Sin embargo, este tipo de enlaces dispone del mayor ancho de banda en toda la red.

Encontrar un cuello de botella en la red requiere de una topología, pero es común que los *hosts* posean un único *access point* (y, por ende, un único enlace) a la red. El ancho de banda del mismo limita el número y la calidad de los servicios que puede emitir un único servidor de *streaming*. Para ejemplificar, se presenta la Tabla 4.1.

Ancho de banda disponible	Número de servicios SD	Número de servicios HD
30 Mbps	14	2
100 Mbps	47	8
1 Gbps	476	82

Tabla 4.1: Cantidad de servicios que pueden ser emitidos por un enlace, considerando ancho de banda requerido.

Los valores presentados se basan en mediciones realizadas sobre conexiones ethernet sobre par trenzado, donde el *overhead* de comunicación es de un 4,1 %, en promedio. Este valor proviene del envío de 7 paquetes⁵ de flujo de transporte (1316 bytes) sobre la cabecera del paquete UDP(8 bytes), la cabecera del paquete IP(20 bytes) y la cabecera de la trama *ethernet*(26 bytes).

4.2. Análisis del trabajo

A fin de evaluar la extensión de la norma y el reproductor Mara en un marco práctico, se analizan diversos aspectos relevantes utilizando la infraestructura presentada en figura 4.1 para las ejecución de pruebas. La Tabla 4.2 presenta y clasifica los resultados de las evaluaciones. A su vez, incluye referencias a las secciones posteriores, que desarrollan más detalladamente cada prueba o aspecto del análisis. Es importante remarcar que la columna titulada **ISDB-Tb + IP** analiza los aspectos inherentes a los servicios enviados por IP. Es decir, el diagrama no establece que es imposible enviar *Closed Caption* con la extensión de la norma. Sí afirma que es imposible enviarlo para aquellos servicios que se transmiten por IP.

4.3. Número de servicios

El objetivo final del nuevo diseño es extender la lista de servicios provistos por una señal ISDB-Tb convencional, a través del aprovechamiento de redes IP disponibles para

³Principales enlaces de datos en una red. En este caso, en particular, refiere a una red IP.

⁴En teoría de grafos, refiere a una arista que, si se elimina, aumenta el número de componentes conexas de la topología.

⁵Es imposible enviar un número mayor de paquetes, dado que es ideal respetar el tamaño de ventana más común para redes IP de 1500 bytes.

Tabla 4.2: Resumen de características de la solución elaborada

Aspecto	ISDB-Tb Tradicional	ISDB-Tb + IP	Ver	Resumen
Número de servicios	Hasta 5 SD y 1 LD	Hasta 50 HD	Sección 4.3	Con el uso del nuevo descriptor, el máximo número de servicios asciende a 50. La cota es consecuencia del tamaño máximo de la sección de la SDT.
Múltiples Flujos elementales	✓	✗	Sección 4.4	Mara no posee la funcionalidad necesaria para entregar múltiples flujos elementales intercambiables. Por otro lado, su inclusión incurriría en un desperdicio innecesario de ancho de banda.
Guía electrónica de programación	✓	✗	Sección 4.5	El soporte que ofrece MPEG para la descripción de guía electrónica de programación no basta para el número de servicios que permite la extensión.
Closed Caption	✓	✗	Sección 4.6	Mara actualmente no soporta closed captioning para los servicios que se envían por IP. Dependiendo del tipo de los mismos, sería fácil adaptar el reproductor para soportarlo.
Gestión de audiencias	✗	✓	Sección 4.7	En el caso de ISDB-Tb convencional, es imposible determinar qué servicio consume cada usuario. Por el contrario, esto es posible con los servicios enviados por IP.
Aplicaciones interactivas	✓	✗	Sección 4.8	Mara no provee soporte para esta funcionalidad. Si se deseara posibilitar esto, sería necesario un uso diferente de este mecanismo respecto al que utiliza ISDB-Tb.
Velocidad de sintonización	1,8s	Sintonización + buffering (1,7s)	Sección 4.9	Aún para los servicios que se transmiten por IP, es necesario sintonizar la banda de radiofrecuencia que transporta el flujo que señala al servicio.

el decodificador. En consecuencia se debe analizar la limitación del tamaño de la lista de servicios, considerando la nueva posibilidad de usar el *Elementary Streams Relocation Descriptor*.

Son varios los aspectos que se deben considerar para determinar la verdadera limitación del número de servicios en un flujo de transporte, de acuerdo al formato MPEG. En esta sección se mencionan estos aspectos y se incluye el resultado del análisis, pero para una discusión completa de este punto, ver el apéndice B.

- **Tamaño de sección:** El tamaño de sección para la mayoría de las tablas de ISDB-Tb es de 1024 *Kilobytes*. Esto limita el número de entradas y descriptores que pueden incluirse en ellas. Las tablas afectadas son la PAT y la SDT.
 - **PAT:** Transporta contenidos binarios, por lo que la señalización no ocupa demasiado espacio. En la absoluta mayoría de los casos, toda la información necesaria cabe en un único paquete TS. En una sección de PAT pueden señalizarse hasta 252 servicios.
 - **SDT:** Transporta información textual sobre el flujo de transporte. En primer lugar esto significa que los contenidos son los más pesados de todas las tablas obligatorias. En segundo lugar, la cantidad de bytes utilizados para la misma depende de los servicios que describe. Para un análisis que supone nombres de servicios de 9 caracteres en promedio, la SDT ubica la cota superior en el número de servicios en 50.
- **Disponibilidad de paquetes nulos:** La inclusión de servicios no implica que las PMT crezcan, en su lugar aparecen nuevas instancias de la tabla. Es la única tabla con estas características.
 - **PMT:** La adición de un servicio, de acuerdo al capítulo anterior, requiere del reemplazo de un paquete nulo por uno que contenga una instancia de esta tabla. El formato MPEG establece que debe haber una separación temporal máxima de 200 milisegundos entre instancias de la PMT para un mismo servicio. Con esta restricción, es necesario considerar el *bitrate* del flujo de transporte en el análisis. Para un flujo de transporte similar al de Canal 23 estudiado previamente, la limitación de este aspecto del formato fija el número máximo de servicios en 200.

En conclusión, el número máximo de servicios, si bien depende de los contenidos del flujo de transporte, responde directamente al formato de específico de la SDT. Y para el caso promedio de 9 caracteres por nombre de servicio, este número es de 50 servicios por flujo de transporte.

4.4. Flujos elementales múltiples

Es normal que en los servicios de televisión digital se incluyan varios flujos elementales de un mismo tipo para un mismo servicio. Por ejemplo, la emisión de un partido de

fútbol podría incluir múltiples audios para permitir la elección del relato a reproducir, posiblemente en distintos idiomas. Esto no suele ocurrir para el video.

En el caso de ISDB-Tb extendido, la red IP emite un flujo de transporte simple. En otras palabras, envía un flujo de transporte que contiene una única señal de video y de audio. Esto puede considerarse una limitación del diseño pero resulta importante, para los servicios que viajan por IP, la eliminación de toda información redundante. Si se emitieran múltiples flujos de audio a través del mismo grupo, entonces algunos de ellos serían ignorados, incurriendo en un desperdicio innecesario de ancho de banda. En otras palabras, todos los audios, salvando aquel que se está reproduciendo, serían inútiles.

Para evitar tal desperdicio, sería posible enviar cada flujo por un grupo multicast distinto. Existen técnicas de transmisión de contenidos donde el ensamble del servicio final percibido se realiza en el cliente de reproducción. Este aspecto de la solución se visita en el capítulo final de Conclusión, en la Sección 5.3.1, como parte de las propuestas de trabajo futuro.

4.5. Guía electrónica de programación

La guía electrónica de programación (o EPG por sus siglas en inglés) contiene información acerca de los eventos en los servicios del flujo de transporte. La información que compone la guía de programación se construye por las emisoras de contenido y se envía a los decodificadores a través de la EIT (*Event Information Table*). Las secciones que componen una EIT pueden alcanzar los 4096 bytes. Cabe remarcar que cada sección guarda un bloque de tres horas de eventos.

Se suele denotar EIT-k a cada sección, donde un k mayor denota un bloque más lejano en el tiempo. Así, EIT-0 describe los eventos de las próximas tres horas y EIT-1 los eventos de las tres horas siguientes. En consecuencia hay 4096 bytes para describir los eventos que ocurren en todos los servicios en el intervalo de tiempo que le corresponde a la sección.

El tamaño de cada descripción de evento depende del título del mismo y del texto asociado. Por esto, es imposible determinar cuántos servicios se pueden describir, pero se puede realizar una estimación sobre el espacio disponible para cada evento, dada una lista de servicios.

Asumiendo que la lista de servicios del flujo de transporte posee 50 entradas (limitación impuesta por la sección de la SDT, ver Sección 4.3), y que en tres horas cada servicio auspicia dos eventos, se deben describir 100 eventos.

La figura 4.4 ilustra el formato de una sección de la EIT. El encabezado de la EIT ocupa los primeros 14 bytes, y cada evento, a su vez, tiene un tamaño base de 12 bytes. Esto significa que de los 4096 bytes disponibles de la sección, 2882 pueden utilizarse para títulos y descripción de eventos. Lo cual da un promedio de 29 bytes por evento para título y descripción.

La figura 4.5 establece una relación entre el número de eventos a describir y la cantidad de caracteres disponibles. A partir de los 20 eventos, la disponibilidad cae debajo de los 200. En resumen, no es posible describir acordemente todos los servicios incluidos en el

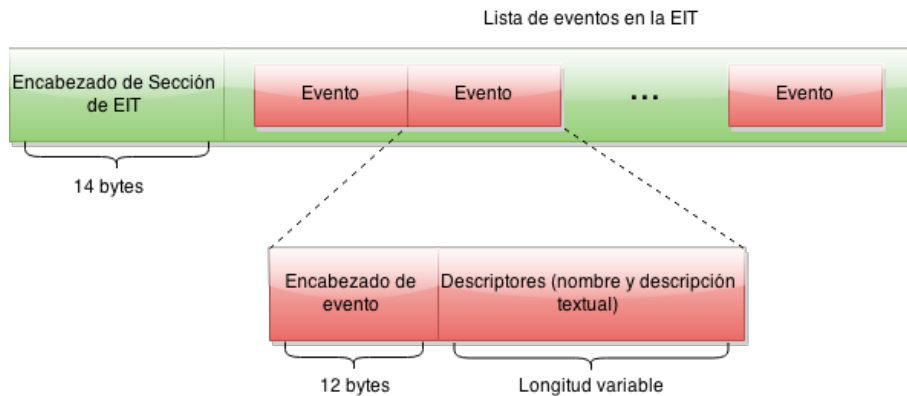


Figura 4.4: Representación ilustrativa de la sintaxis de una sección de EIT

flujo de transporte extendido a través de la EIT, salvando la inclusión del nombre del evento y la información textual asociada a alguno de ellos.

4.6. *Closed Captioning*

Existen varios estándares ampliamente difundidos para la transmisión de *Closed Captioning*. El formato estándar para ISDB-Tb se define en ARIB-B37. Otro tipo también muy difundido, pero más popular en otras normas como ATSC, es CEA 608 y su sucesor CEA 708[23], denominados comúnmente simplemente por el número. Existe una diferencia principal entre ambos formatos que propone una clasificación de los estándares de subtítulo.

- **Embebidos:** En esta categoría se encuentran los formatos 608 y 708. Los subtítulos de esta categoría se incluyen en un espacio preasignado de un flujo externo. Por ejemplo, el formato de video H.264 provee un espacio de usuario para incluir, entre otras cosas, subtítulos. La característica que define a esta categoría es que no necesitan señalización en la PMT, dado que los *Closed Captioning* forman parte del flujo elemental de video.
- **Señalizados:** Categorizamos en este grupo a los canales de *Closed Captioning* que deben ser referidos por la **PMT** y que viajan en un flujo elemental con su propio PID. En esta categoría entra el formato ARIB-37.

Mara no posee la funcionalidad de reproducir ninguno de los dos tipos. La imposibilidad de reproducir subtítulos señalizados viene dada por el diseño de la señalización. La imposibilidad de reproducir los subtítulos embebidos se debe a la forma que Wari reproduce los flujos elementales. La entrega de *Closed Captioning* de la primer categoría requeriría de Mara leer la PMT del servicio reubicado para demultiplexar los paquetes que

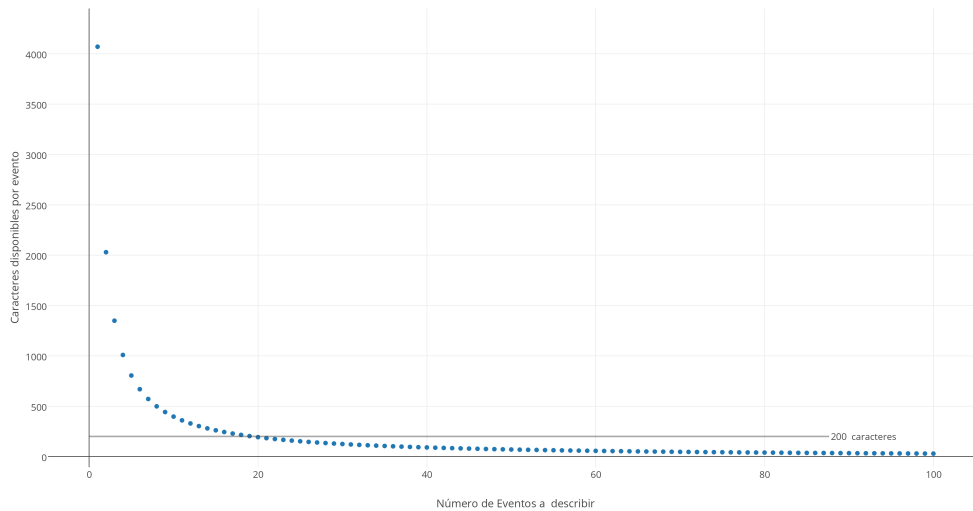


Figura 4.5: Gráfico de puntos que describe la disponibilidad de caracteres para la descripción de eventos en la EIT

transportan los datos de subtítulo. Para esto, *mpegparser* debería filtrar los paquetes involucrados y realizar el *parsing*.

De forma mucho más simple, los subtítulos de la segunda categoría pueden ser manejados directamente por el reproductor, dado que forman parte del video. El cambio necesario implica enviar un mensaje al reproductor para active la pista de subtítulos correspondiente. La complejidad de la modificación depende el estándar de *Closed Caption*, pero no implica la lectura del flujo de transporte en *mpegparser*, facilitando drásticamente la implementación.

4.7. Gestión de audiencias

La emisión en las señales ISDB-Tb es *broadcast*. En este esquema los consumidores son anónimos, lo cual implica que es imposible determinar si un receptor está consumiendo un servicio dado. En el caso de los servicios reubicados, las redes IP identifican a los *hosts* unívocamente a través de una dirección IP. Para saber si un receptor consume un servicio transportado por IP, éste puede ser asociado a una dirección y determinar a qué grupo multicast se ha unido.

Utilizando este tipo de técnicas, sería posible para una entidad con autoridad sobre la red IP que emite los servicios determinar el número de usuarios que consumen determinado servicios, así como ubicación geográfica y otros datos más específicos. Este tipo de mediciones no se puede realizar sobre ISDB-Tb tradicional y, en general, tampoco sobre cualquier esquema *broadcast*.

4.8. Aplicaciones Interactivas

Digital storage media command and control (DSM-CC) es el mecanismo que utiliza la televisión digital para la distribución de aplicaciones interactivas, definido en MPEG-2, parte 6. DSM-CC provee un conjunto de herramientas para el desarrollo de canales de control en un modelo cliente/servidor. Empero, es imposible implementar directamente este modelo en la norma ISDB-Tb por la naturaleza subyacente, donde no existe un canal de retorno para el cliente.

Las emisiones *broadcast* de televisión digital, entonces, hacen uso de un carrusel de datos para solucionar este problema. El mismo consiste en la emisión periódica de los datos disponibles. Para que el cliente pueda acceder al cuerpo de datos que intenta obtener, simplemente debe esperar su emisión. Existen dos motivos por los que el nuevo diseño no puede reubicar contenido DSM-CC a una red IP.

- **Implementación:** Actualmente, los contenidos recibidos a través de la red IP son alimentados directamente al reproductor audiovisual. Esto elimina la posibilidad de proveer el *middleware* necesario para la ejecución de las aplicaciones. *Mara* no posee la funcionalidad necesaria para redireccionar el contenido DSM-CC fuera del reproductor de video y audio hacia el intérprete de comandos.
- **Diseño:** En una red IP existe un canal de retorno y el modelo cliente/servidor se aplica de manera natural. Además, el ancho de banda no es un flujo constante que debe ser aprovechado, sino que su mala utilización incurre en una sobrecarga innecesaria. Por estos motivos, la adición de soporte DSM-CC sugiere una implementación del protocolo diferente de la utilizada por receptores ISDB-Tb.

Si se solucionara la ausencia de implementación de *Mara*, la utilización de contenido DSM-CC en el formato establecido en la norma ISDB-Tb incurriría en un desaprovechamiento considerable del ancho de banda. Especialmente si se tiene en cuenta que la versión original del conjunto de herramientas DSM-CC se adecúa perfectamente al modelo de comunicación unicast con retorno que ofrecen las redes IP.

En conclusión, mientras los servicios audiovisuales de una señal ISDB-Tb se reubican de forma natural como flujos a las redes IP, las aplicaciones interactivas deben ser tratadas de forma diferente. Preferiblemente utilizando el soporte completo que ofrece DSM-CC, para eliminar la necesidad del uso de un carrusel de datos.

4.9. Tiempo de sintonización

Utilizando la infraestructura de prueba presentada en la figura 4.2 se realizaron mediciones sobre el tiempo de selección de un servicio. La figura 4.6 presenta los resultados obtenidos clasificándolos por el servicio *destino*, dado que este valor es independiente del servicio *origen*. Hay cuatro escenarios posibles:

- El servicio destino se transmite por el mismo canal de radiofrecuencia, de forma tradicional. En la figura 4.6 corresponde al destino *regulares*, misma frecuencia.

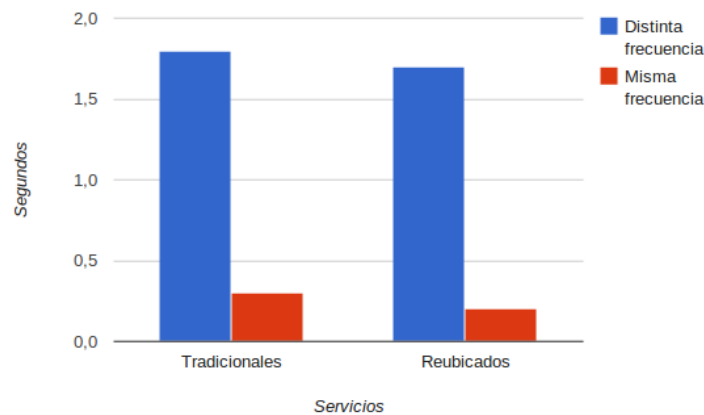


Figura 4.6: Tiempos de sintonización

- El servicio destino se transmite de forma tradicional, pero por una frecuencia distinta a la sintonizada actualmente. En el gráfico corresponde a los servicios regulares, distinta frecuencia.
- El servicio destino se transmite por una red IP, y es señalado por el flujo de transporte sintonizado actualmente. En el gráfico, servicios reubicados, misma frecuencia.
- El servicio destino se transmite por la red IP, y es señalado por un flujo de transporte ajeno al actual. En el gráfico, servicios reubicados, distinta frecuencia.

La medición de mayor magnitud corresponde a la selección de un servicio tradicional que se transmite por un flujo de transporte distinto al actual, obligando al reproductor a realizar una sintonización de banda. La demora para tal selección es de 1,8 segundos. Por el contrario, si el servicio sí se encuentra en el mismo flujo de transporte, el tiempo de selección no supera las tres décimas de segundo.

Por las mediciones, es claro que el cuello de botella en el procedimiento radica en la sintonización de banda de frecuencia. Para los servicios cuyos flujos elementales vienen por una red IP, este proceso es innecesario, de modo que es esperable que este tiempo se reduzca drásticamente. Sin embargo, la figura 4.6 muestra lo contrario. Para explicar esto, es necesario retomar el proceso de sintonización.

El diagrama de secuencia de la figura 3.8, incluida en el capítulo Diseño y Desarrollo, expone el proceso normal de selección de un servicio reubicado para su reproducción, incluyendo la construcción de la URL de reproducción. Un bloque de condición del diagrama encapsula la verificación de correspondencia de flujo de transporte sintonizado y

flujo de transporte que contiene al servicio seleccionado. Este bloque comprende la mayoría del tiempo de demora del procedimiento completo. A simple vista, la sintonización física es innecesaria, dado que inmediatamente después, el reproductor pasa a consumir los contenidos desde la interfaz de red, en lugar del dispositivo sintonizador. En realidad, la omisión de este bloque incurriría en la imposibilidad de cambio de información de origen.

Por ejemplo, en el caso de los servicios reubicados, sería imposible cambiar el grupo multicast por el que el servidor que realiza el *streaming*. De lo contrario, los usuarios que encendieran el reproductor antes del cambio, seguirían intentando unirse al grupo multicast antiguo. El único lugar donde se reflejaría el cambio sería en la PAT que viaja en el flujo de transporte correspondiente a la señalización de servicio. Esto sería análogo a memorizar los PID de los flujos elementales provistos por una PMT y usarlos posteriormente. Un eventual cambio de PID para el flujo imposibilitaría la sintonización del servicio.

En resumen, si bien para obtener los flujos elementales de un servicio reubicado no es necesario un cambio de sintonía, el proceso es necesario para garantizar la consistencia del proceso de sintonización. De esta manera, se evita la referencia de grupos multicast obsoletos y se permite la ubicación de nuevos servicios en la red IP.

Con esta sección finaliza el detalle de las evaluaciones realizadas sobre el sistema híbrido. El capítulo final de Conclusión, a continuación, recorre las características desde el punto de vista de los objetivos planteados inicialmente. Además, incluye posibles elementos de trabajo futuro, que complementa los estudios del capítulo de Evaluación.

Capítulo 5

Conclusiones y Trabajo Futuro

En este trabajo de tesina se ha diseñado y especificado una extensión de la norma ISDB-Tb para señalar servicios que se proveen a través de una red IP. De esta manera, aquellos receptores de televisión digital que tengan acceso a la misma puedan unirse a un grupo multicast definido y entregar el servicio al usuario de manera transparente.

5.1. Crecimiento de la lista de servicios

La norma ISDB-Tb define los lineamientos para la transmisión de televisión digital en varios países del mundo, incluyendo Argentina. Para la organización de la información, ISDB-Tb usa el formato MPEG Transport Stream que permite la multiplexación de múltiples servicios en un único flujo. Para la demultiplexación, MPEG provee tablas. Las tablas entregan información necesaria al receptor y permiten el agregado de descriptores. Los hay obligatorios y opcionales; también permite el formato MPEG el uso de descriptores privados, definidos por el usuario.

Para lograr el crecimiento de la lista de servicios de una señal, se creó un nuevo descriptor que respeta las consideraciones definidas en el formato MPEG, pudiendo clasificarse como privado. El mismo incluye la información necesaria para señalar la emisión del servicio a través del protocolo IP. El *Elementary Streams Relocation Descriptor* se incluye en las PMT de los servicios que se emiten por red IP. La estructura sintáctica del descriptor permite la inclusión de un grupo multicast y un número de puerto.

En el capítulo Introducción Técnica se demostraron dos procedimientos necesarios en la reproducción de televisión digital, siendo el primero de ellos el *scan* y el segundo la sintonización de servicios. El primero, que construye la lista de servicios se mantiene intacto. Inicialmente, se recorren todas las frecuencias disponibles, recolectando las SDT de cada flujo de transporte. Las instancias de esta tabla proveen los nombres que componen la lista.

El proceso que cambia es la sintonización de servicios, pero sólo en el caso de aquellos que incluyen una instancia del *Elementary Streams Relocation Descriptor*. Si la PMT carga tal descriptor, entonces no posee flujos elementales asociados en el *transport stream*. En su lugar, es posible obtener de su *payload* el grupo multicast y puerto de emisión por

el que se emiten los flujos. Una vez el receptor dispone de los mismos, puede redirigirlos hacia su reproducción sin la necesidad de interpretar sus contenidos.

5.1.1. Características importantes de la combinación ISDB-Tb con IP

- **Retrocompatibilidad:** Cualquier software de decodificación de señal ISDB-Tb tradicional sigue funcionando normalmente cuando consume una señal extendida. No es capaz, sin embargo, de reproducir los servicios identificados con el *Elementary Streams Relocation Descriptor*. Éstos figurarán en la lista de servicios, pero el usuario no podrá consumirlos.
- **Mayor tamaño de lista de servicios:** Impuesta por el tamaño de sección de la SDT, la cota superior para el número de servicios señalizables por un flujo de transporte es de 50 (valor independiente de la calidad de los servicios), en promedio, considerando una longitud de nombre de 9 caracteres. Muy superior al máximo de 5 servicios SD y 1 LD de ISDB-Tb tradicional.
- **Escalable:** El nuevo diseño puede escalar en dos direcciones. En primera instancia, a cualquier flujo de transporte se le pueden adicionar nuevos servicios reubicados, siempre y cuando la SDT lo permita (de acuerdo al análisis del capítulo de Evaluación, Sección 4.3). En segunda instancia, se pueden agregar servicios reubicados a otras frecuencias. La señalización de cada flujo es independiente y pueden añadirse a distintas bandas sin problemas.
- **Dinámico:** A cualquier flujo de transporte se le pueden agregar los servicios en tiempo real, sin la necesidad de demultiplexar el flujo original.
- **Independiente del multiplexador:** Las estructuras utilizadas respetan por completo el estándar MPEG-2 TS. Por ende, no es necesaria ninguna maquinaria especial en la multiplexación del flujo de transporte.
- **Preserva *bitrate*:** Dado que las operaciones necesarias consisten únicamente en el reemplazo de paquetes, el *bitrate* del flujo de transporte se conserva igual.
- **Preserva dependencias:** Por la elección de tecnologías utilizadas, la transmisión híbrida no agrega dependencias al receptor.

5.2. Mara

A modo de implementación de referencia, el trabajo de tesina incluyó el desarrollo de Mara, un reproductor de televisión digital basado en Wari. A través de Mara, es posible sintonizar la lista completa de servicios de un flujo de transporte que incluye el *Elementary Streams Relocation Descriptor* uniéndose al grupo multicast correspondiente y renderizando los contenidos. Las tres características más importantes del reproductor son:

- **Transparente:** Es imposible para el usuario de Mara determinar si un servicio está siendo consumido desde la señal de radiofrecuencia o de la red IP, asumiendo que el decodificador tiene acceso a ambos medios.
- **Portable:** Puede ser utilizado en las plataformas más conocidas. Por ejemplo: Linux, Windows, OS X, Android.
- **Independiente del Hardware:** Funciona con cualquier sintonizador que soporte señales ISDB-Tb.

5.3. Trabajo Futuro

En esta sección se analizan posibles pasos en la mejora del trabajo realizado en esta tesina.

5.3.1. Reubicación de objetos en *Object Based Broadcasting*

La principal limitación del diseño yace en la rigidez de los servicios que se obtienen por IP. A diferencia de los servicios regulares, es imposible incluir varios flujos elementales de un mismo tipo. Por ejemplo, varios audios de distintos idiomas. *Object Based Broadcasting*[24] es una técnica a través de la cual múltiples recursos se emiten simultáneamente y el reproductor le permite al usuario seleccionar cómo componerlos para entregar el servicio completo.

Esta técnica presenta un escenario de mejora para la integración de ISDB-Tb e IPTV, en el que distintos flujos combinables se emiten en distintos grupos multicast. De esa manera no sólo se logra proveer varias opciones para un mismo tipo de flujo (por ejemplo, varios audios en diferentes idiomas), sino también servicios que se adecúan al usuario, dado que le permiten elegir qué se muestra en pantalla. Sin incurrir así en un desperdicio de ancho de banda para aquellos ignorados por el consumidor, como es el caso en ISDB-Tb tradicional.

Para llevar adelante esta idea, es posible aplicar el *Elementary Streams Relocation Descriptor* a flujos elementales, en lugar de un servicio completo. Así, cada uno se emite por un grupo distinto. Y el receptor se une sólo a aquellos grupos que contienen un flujo que le interesa al usuario. Luego, el software de recepción debe ensamblar estos contenidos para entregar el servicio completo.

5.3.2. Reubicación por eventos

La motivación para integrar IPTV e ISDB-Tb se basa en la forma en que se complementan ambos sistemas de entrega. Mientras el primero permite una lista de servicios potencialmente muy grande (en comparación), la integridad del segundo no padece el número de consumidores del servicio.

En el caso de televisión digital, una de las variables más importantes que determina el número de consumidores que tiene un servicio es el evento que emite. Así, los eventos

más populares suelen conseguir un mayor número de consumidores que aquellos menos populares. En el diseño presentado en este trabajo, cuando un servicio es emitido por un medio, no puede ser mudado al otro dinámicamente.

Una posibilidad para solucionar este problema como trabajo futuro, es la señalización a través de eventos. Entonces, en lugar de asociar la información de emisión a un servicio, se puede vincular a un evento. El receptor sería el encargado de realizar el *buffering* para que la transición sea transparente. En este formato, un servicio pasaría de ser emitido por un medio al otro sin los inconvenientes causados por un *scan*.

5.3.3. Soporte de IPV6

La versión más moderna del protocolo IP utiliza direcciones de 128 bits, en lugar de los 32 utilizados por IPV4. En este caso, sería necesaria una modificación de la estructura del descriptor para contener la nueva variante, o incluso permitir funcionar en cualquiera de las dos versiones. En este último caso, el descriptor requeriría utilizar longitud variable.

5.3.4. Omisión de PMT pequeñas

En el diseño actual, por cada servicio se envía una PMT que desaprovecha un 82 % de su carga útil disponible. En lugar de esto, es posible señalar los servicios reubicados directamente desde la PAT que, al transportar únicamente contenido binario, es una tabla compacta. En este caso, la modificación del receptor requiere un trabajo considerablemente mayor porque el proceso de sintonización de servicios cambia respecto de la norma, eliminando la intervención de la PMT.

Esta solución presenta algunos inconvenientes y desafíos técnicos. En primer lugar, un decodificador común podría presentar problemas al encontrar la irregularidad en la emisión. De esta manera, imposibilitando la construcción completa de la lista de servicios, o hasta ignorando todos los servicios del flujo. La ventaja primordial, en cambio, es que elimina la necesidad de cambiar paquetes nulos por PMTs, simplificando notoriamente la inserción de servicios. Si además se eliminara la restricción sobre el tamaño de secciones para las tablas PAT y SDT, el número de servicios podría ascender considerablemente.

5.4. Conclusión

Esta tesina presenta un primer acercamiento al inconveniente del número de servicios transmitidos por la televisión digital terrestre, y logra que sea comparable a la señal analógica en este aspecto. Es una solución que puede convivir perfectamente en el esquema actual de transmisión y respeta los estándares involucrados. Tiene aún muchos puntos para mejorar, pero el diseño simple y ordenado de la solución y de MPEG desplaza todo el trabajo al esfuerzo de implementación de un cliente tan poderoso como se requiera. *Mara* es una adaptación muy sencilla que sirve como referencia de lo realizado. Es deseable, en un futuro, crear una versión de *Mara* nueva, con un enfoque más modular y flexible que el que impulsó a *Wari*, cuyos objetivos fueron claros desde un inicio como reproductor de TV tradicional.

Es cierto que la digitalización puso a la señal de aire sobre la mesa, pero aún no es ni remotamente suficiente considerando la hegemonía del cable como medio. Se procuró con este trabajo acortar las distancias que se encuentran al analizar soluciones que combinen diferentes entornos. Esta iniciativa, lejos de ser novedosa, posee algunos referentes como *HbbTV* que tiene 6 años de vida al momento de escribir de este informe. Aún así dista de poseer la popularidad necesaria para ser reconocido como el mecanismo *de facto* para consumir televisión, pero da evidencia de que la iniciativa es un buen camino a seguir.

La distribución de contenidos televisivos es un tema de investigación atractivo por dos motivos. En primer lugar, disciplinas muy diversas intervienen en la construcción del servicio: Ingenieros electrónicos, expertos en telecomunicaciones, diseñadores de experiencia de usuario y hasta sociólogos y artistas, que producen contenidos. El aparato que rodea la entrega de servicios audiovisuales involucra mecanismos extremadamente complejos a niveles muy diversos de abstracción. La informática colabora apenas en una porción del desafío completo.

En segundo lugar, es un tema de desarrollo en la agenda de las empresas más importantes del mundo y el debate técnico está vigente actualmente. Los usos que se le dan actualmente apenas comienzan a experimentar interactividad y múltiples pantallas; dando así nacimiento a nuevos protocolos y formatos, como MPEG-DASH, para señales adaptativas. La vertiginosa búsqueda tecnológica sustenta el crecimiento y cambio de la matriz de contenidos y audiencia.

Años atrás, el término televisión definía a la señal de imágenes en movimiento consumida a través de un televisor. En la actualidad, definir televisión no es una tarea sencilla. Al digitalizar la señal, el televisor perdió su rol protagónico. Con la aparición de los videos *on demand*, se eliminaron las restricciones propias del *broadcasting*. Hoy, la telefonía móvil permite el intercambio de contenidos audiovisuales entre pares de forma inmediata, de modo que la producción y distribución dejó de ser centralizada o estar limitada a grandes estudios de grabación. No existe una forma de predecir el próximo cambio que sufrirá la distribución de contenidos audiovisuales. Sucede que, al igual que con internet, la televisión es simplemente el medio.

Apéndice A

Introducción al *MPEG-2 Transport Stream*

En este apéndice se revisan conceptos introductorios útiles del flujo de transporte MPEG. Inicialmente se realiza un recorrido sobre el ciclo de vida de los contenidos. Luego se estudia más en detalle la estructura de la información en el formato. Si bien este apéndice tiene carácter informativo, la referencia ulterior es el estándar oficial. En lo que a formato de datos respecta, particularmente el documento ARIB-STD B10[9].

A.1. Codificación y empaquetado de los datos

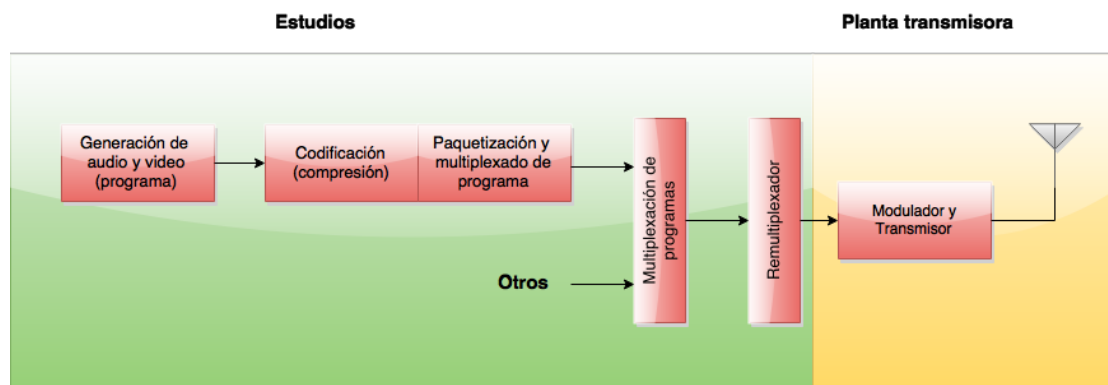


Figura A.1: Diagrama funcional de una estación de televisión digital terrestre

La figura A.1 es una imagen simplificada de una estación que produce contenidos de televisión digital. Los componentes incluidos en la imagen son los siguientes:

- **Generación de programa:** Las estaciones de TV incluyen equipamiento para la creación del servicio que se entrega finalmente al espectador. Dentro de este

equipamiento se encuentran las cámaras, los *switchers*¹, los generadores de efectos, los distribuidores, consolas de audio, etc. La salida de esta etapa es un flujo de contenido binario a tasas elevadas, no aptas para la distribución a través de radiofrecuencia.

- **Codificación:** La señal recibida es codificada y empaquetada en bloques que contienen una cantidad relativamente reducida de bits, conformando un flujo de paquetes de longitud variable.
- **Paquetización y Multiplexación:** Los paquetes producidos en la etapa anterior son nuevamente fragmentados para producir paquetes de tamaño aún menor, esta vez de longitud fija, utilizados en el flujo de transporte MPEG *transport stream*. Es necesario en esta etapa incluir datos adicionales relativos al programa, como nombre, metainformación de transmisión y de demultiplexación.
- **Multiplexación de programas:** Es poco común la transmisión de un flujo de transporte con un único servicio. En esta etapa se combinan varios para construir un MPTS, *Multiple Program Transport Stream*.
- **Remultiplexación:** Esta etapa recibe el flujo de transporte con múltiples servicios y le agrega información útil de transmisión, que incluye *timestamps* entre otras cosas. Al resultado de esta etapa se lo denomina *Broadcast Transport Stream*.
- **Modulador y transmisor:** En esta última etapa, la información es convertida en una señal de radiofrecuencia que luego es emitida por la antena.

A.1.1. Formato de paquetes TS

La información recibida por los receptores se estructura en el flujo de transporte. Este formato se adapta perfectamente a la televisión digital terrestre. Por la longitud de los paquetes, es fácil la inclusión de datos para corrección de errores. Por sus mecanismos de multiplexación, es posible incluir varios servicios simultáneamente, sin contar con una referencia de reloj común.

El flujo TS incluye paquetes de audio, video y datos. Además, otros paquetes suministran información para el receptor. En demultiplexación, la separación de paquetes es posible gracias a sus encabezados que incluyen un byte de sincronización y un PID (*Packet Identifier*), que identifica el contenido.

La figura A.2 representa un paquete de un flujo de transporte, que se compone de un total de 1504 bits. Los primeros 4 bytes forman el encabezado del paquete. El primero de estos es el byte de sincronización que siempre contiene el valor 47_H (71 en notación decimal). Siendo fija la longitud de los paquetes TS, no es necesario un campo de longitud de *payload* que es siempre 184. Cuando un decodificador halla 5 bytes de sincronización consecutivos, asume que ha logrado sincronizarse con la señal².

¹Equipo selector del video utilizado en el servicio a emitir.

²Asimismo, un error en tres bytes de sincronización consecutivos se consideran pérdida de sincronismo.

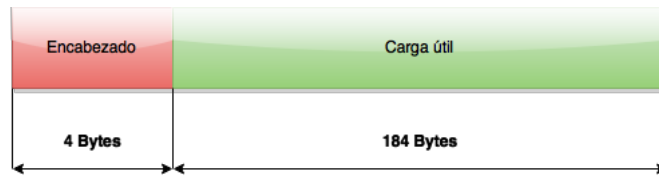


Figura A.2: Representación de un paquete TS.

La longitud de los paquetes TS es fija. Siendo que los datos no se pueden ubicar arbitrariamente hasta rellenar este tamaño, a veces es necesario completar con contenidos descartables. Con este fin y otros, existe el campo de adaptación que se ubica en el encabezado de 4 bytes. En caso de que este campo esté presente, el encabezado aumenta su tamaño.

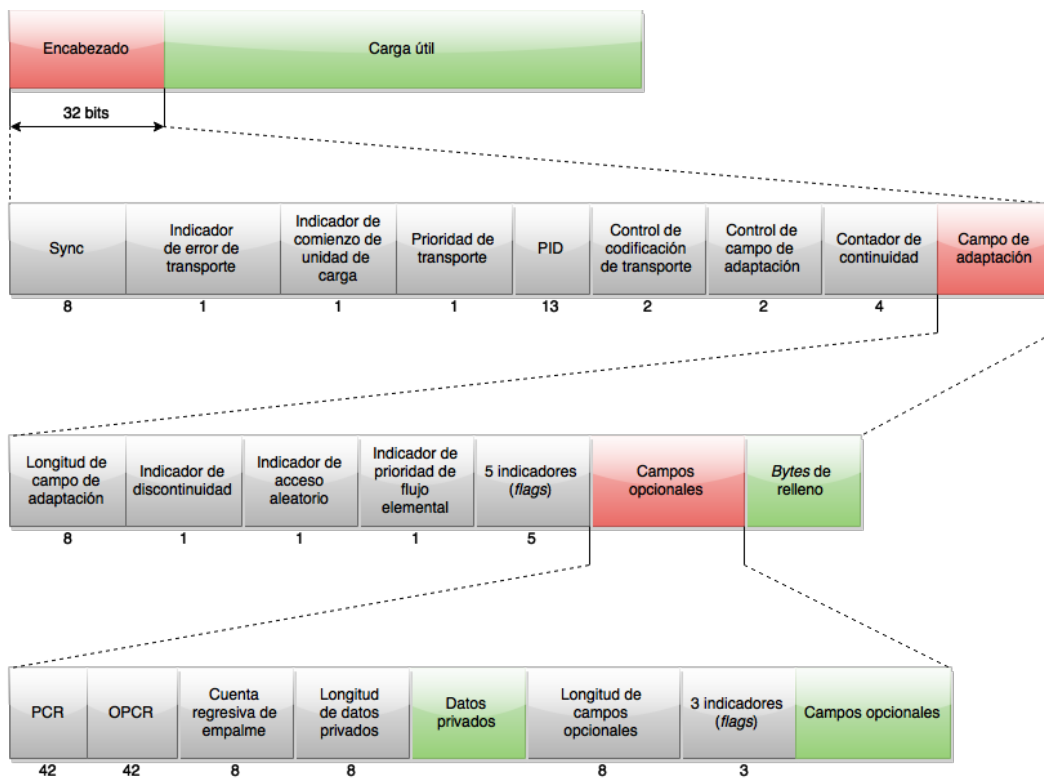


Figura A.3: Estructura del encabezado del paquete de flujo de transporte

La figura A.3 presenta en detalle la estructura del encabezado de los paquetes. La función de los campos del encabezado se detalla a continuación:

- **Sync:** Preámbulo de sincronización 47_H .
- **Indicador de error de transporte:** Señaliza la existencia de error dentro del

paquete. Cuando ese bit es puesto en 1, el decodificador procede a descartar el paquete.

- **Indicador de comienzo de unidad de carga:** Cuando este bit está en 1, el paquete transporta el comienzo de una tabla o de un paquete de flujo elemental.
- **Prioridad de transporte:** Indica que el paquete que contiene el bit en uno contiene mayor prioridad que aquellos con el mismo PID.
- **PID:** Identifica el contenido del paquete. Es el eje de la demultiplexación de contenidos.
- **Control de codificación de transporte:** El término original en inglés es *scrambling*. A diferencia de la encriptación, que opera en el campo digital, el *scrambling* suele denominar operaciones en el campo analógico y tiene el objetivo de modificar la señal para que sólo sea interpretable por aquellos que posean el aparato necesario, sea físico o una llave digital. En conclusión, este campo provee información acerca de la codificación de la señal.

Valor	Significado
00	Sin <i>scrambling</i>
01	Reservado
10	Codificado con llave par
11	Codificado con llave impar

Tabla A.1: Significado de los posibles valores de scrambling.

- **Control de campo de adaptación:** Indica la presencia o no dentro del paquete TS de los campos de adaptación y carga útil, de acuerdo a la siguiente tabla.

Valor	Significado
00	Reservado
01	Sólamente carga útil
10	Sólamente campo de adaptación
11	Campo de adaptación seguido de la carga útil

Tabla A.2: Significado de los posibles valores del control de campo de adaptación.

- **Contador de continuidad:** Contador progresivo de paquetes identificados con el mismo PID, que transportan carga útil. Su valor se incrementa en una unidad con cada paquete de la serie, realizando un *wrap around* cuando todos los bits están en uno. En ciertas situaciones, es necesario retransmitir un paquete. En estos casos, el contador no debe tener en cuenta el paquete duplicado.
- **Campo de adaptación:** Es de carácter opcional, y su extensión depende de los *flags* que posee en su interior. Una de sus importantes funciones es el relleno de

paquetes cuando su carga no alcanza a completar los 184 bytes necesarios. Este es el caso cuando se transporta la información de reloj denominada *Program Clock Reference*(PCR), que se utiliza en el receptor. A continuación se detallan los 3 campos más relevantes. Para más detalle, se puede consultar el documento ISO 13818-1.

- **Longitud de campo de adaptación:** Indica el número de bytes que tiene el campo de adaptación, contando a partir del byte siguiente a este. Si el paquete TS no contiene ninguna carga útil, entonces este campo puede contener 184 bytes de longitud. En caso de contener paquetes PES, suele ser necesario agregar el campo de adaptación para incluir relleno. Para los bytes de relleno se utilizan ceros.
- **Indicador de discontinuidad:** Se emplea para indicar si dentro del paquete se presenta alguna discontinuidad. Por ejemplo, en la base de tiempo en relación al *Program clock reference* o en el contador de discontinuidad. El valor uno indica la presencia de discontinuidad.
- ***Program Clock Reference*(PCR):** Transporta la referencia del reloj principal utilizado en los procesos de decodificación. Ocupa un campo de 42 bits, conformado por una base de 33 bits (con un *tick* de 90Khz) y una extensión de 9 bits que agrega precisión (corriendo a 27Mhz). El PCR sirve para la presentación de flujos elementales sincronizados. La norma permite un error de, a lo sumo, 500 nanosegundos.

A.1.2. Obtención de paquetes PES a partir de paquetes TS

El formato de *Packetized Elementary Stream* está definido en ISO 13818-1. Es el contenedor en el que se ubica el contenido multimedia comprimido. El problema de esta representación es que la longitud (demasiado grande) variable no resulta adecuada para su emisión en un ambiente no confiable como el espectro de radiofrecuencia.

Para obtener el flujo TS, los paquetes PES deben ser segmentados en porciones de 184 bytes, a las que se le agregan encabezados de 4 bytes para formar los paquetes TS de 188 bytes. La longitud de los paquetes PES en bytes no es necesariamente múltiplo de 184; cuando este es el caso el último paquete TS contendrá menos de la totalidad disponible de bytes como carga. La diferencia se compensa en el rellenado del campo de adaptación.

Algunas reglas generales:

- El comienzo de un paquete PES debe siempre coincidir con el inicio de carga útil de un paquete TS.
- El último byte de un PES debe coincidir con el último byte de un paquete TS.
- El contenido de los paquetes TS debe pertenecer a un mismo programa o servicio y no pueden mezclarse servicios diferentes en un mismo paquete.

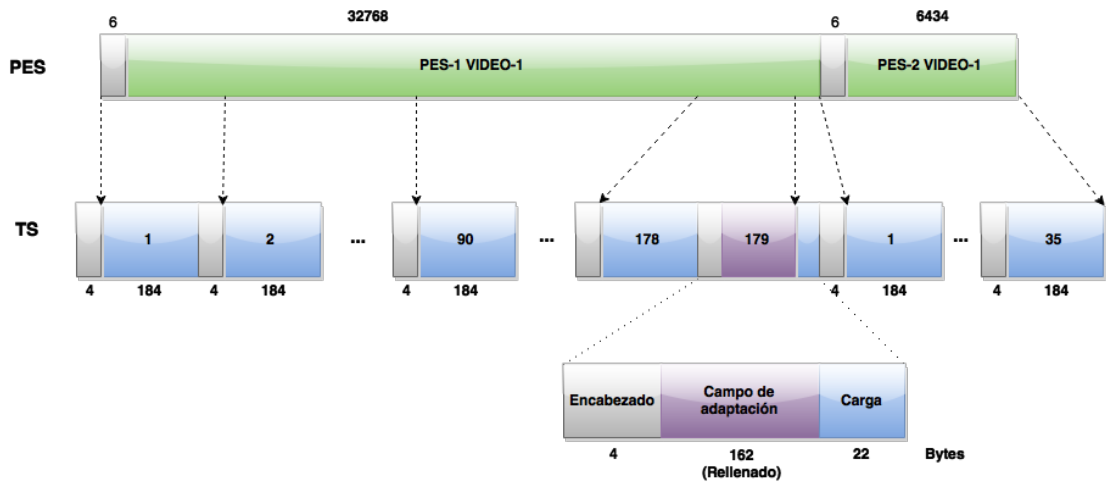


Figura A.4: Ejemplo de distribución de paquetes PES en un flujo de transporte.

La figura A.4 representa la distribución de paquetes PES a lo largo de un flujo de transporte MPEG. El primero de los paquetes no tiene una longitud múltiplo de 184, por lo que un pequeño resto debe ubicarse en el último paquete TS, que debe ser rellenado con el campo de adaptación. El segundo paquete PES del gráfico sí es múltiplo y por ende no requiere bytes de relleno.

A.1.3. Terminología útil en el flujo de transporte MPEG-2

A continuación se detallan conceptos importantes del formato MPEG *Transport Stream* representados en la figura A.5:

- **Red de transmisión o *network*:** Es una entidad que centraliza la gestión de uno o más flujos de transporte. En Argentina, Radio y Televisión Argentina S.E.³ es un ejemplo de red de transmisión.
- **Flujo de transporte:** Un flujo MPEG-2 que lleva uno o más servicios. Se transmite a través de una única banda de frecuencia de las asignadas a televisión digital y suele incluir multiprogramación. En Argentina, Canal 23 (frecuencia 527Mhz) es un ejemplo, a través del cual se transmiten TV Pública HD, Tecnópolis y TV Pública.
- **Servicio:** El término servicio responde al uso coloquial de la palabra *canal*, hecho que deriva de la señal analógica donde a cada banda de radiofrecuencia le corresponde unívocamente un servicio. TV Pública HD es un ejemplo de servicio, pero

³<http://www.rta-se.com.ar/>

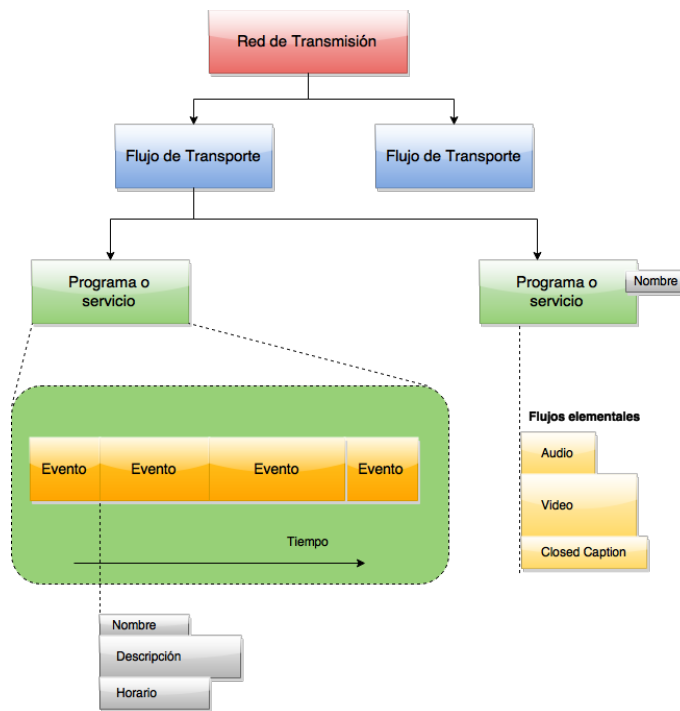


Figura A.5: Diagrama de composición de un flujo de transporte

con la señal digital, existen nuevos tipos de servicios, como los servicios de ingeniería que sirven para mejorar la calidad del servicio. Por ejemplo, las actualizaciones de *firmware* se realizan a través de servicios de ingeniería.

- **Evento:** Es un programa de televisión. Dos eventos de un mismo servicio sólo están diferenciados por el momento en que se está sintonizando un servicio específico. El término proviene del cambio de estado que puede sufrir el flujo de transporte, por ejemplo, por el cambio de los flujos elementales emitidos a través del mismo. Un ejemplo de evento es un partido de fútbol, que tiene una duración y, potencialmente, nombre, descripción, etc. si hay una guía electrónica de programación disponible para el usuario.
- **Flujo elemental o *elementary stream*:** Un evento está compuesto por uno o más flujos elementales. A su vez, puede contener más de un flujo del mismo tipo. *e.g.* un show de televisión puede estar emitiéndose en varios idiomas al mismo tiempo (un audio en inglés y otro en español). Los flujos elementales requieren de la mayor parte del ancho de banda de un flujo de transporte. Se compone de una secuencia de paquetes (no necesariamente consecutivos) con el mismo PID.

Apéndice B

Limitación del número de servicios de la transmisión híbrida

En un flujo de transporte de una señal ISDB-Tb tradicional, la única limitación considerable para el número de servicios provistos viene dada por el ancho de banda disponible para transmitir los contenidos multimedia. En el caso del formato extendido, los servicios que se envían por IP no ocupan ancho de banda(en términos de contenido multimedia) y el mismo requerido por la información de señalización es despreciable.

El número máximo de servicios con el uso del nuevo descriptor está determinado por dos restricciones fijadas por el formato MPEG. La primera es el tamaño de sección, que afecta la transmisión de tablas, limitando la cantidad de información que pueden transportar. Siendo que el tamaño requerido varía, se analiza cada tabla por separado. La segunda restricción es la frecuencia mínima de emisión de tablas.

La figura B.1 muestra cómo afectan las dos restricciones al número máximo de servicios señalizados por cada tabla. La barra roja indica cuántos servicios puede señalar una tabla si ésta ocupa una sección completa y la barra azul indica cuántos servicios se pueden señalar con un único paquete de flujo de transporte. Las secciones siguientes detallan cómo limita cada aspecto del formato MPEG al número máximo de servicios que se pueden señalar.

B.1. Tamaño de sección

Las tablas viajan en estructuras sintácticas denominadas *secciones*. El encabezado de una sección contiene su longitud en bytes en un campo de *12 bits*. En consecuencia, este campo limita la longitud a 4096 bytes. Pero el estándar ISDB-Tb, a su vez, define que sólo la sección que transporta una EIT puede alcanzar este tamaño. El resto de las secciones deben limitarse a *1024 bytes*. En consecuencia, las tablas involucradas en la señalización de servicios transportados por IP están acotadas a este tamaño.

A continuación se analiza cómo influye esta cota sobre el número total de servicios. Todas las mediciones se realizaron construyendo flujos de transporte reales, donde las

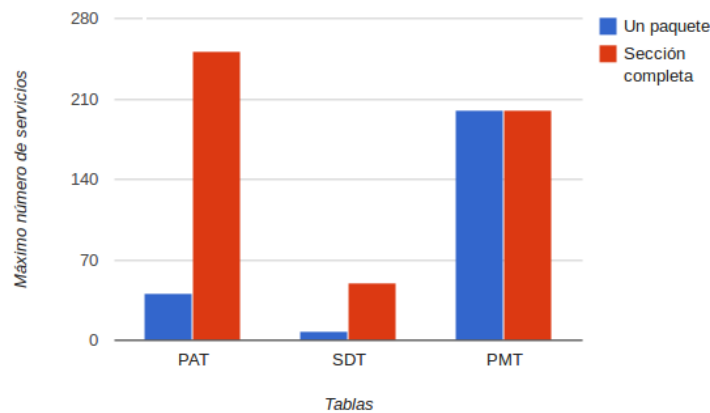


Figura B.1: Cantidad máxima de servicios permitida por cada tabla.

tablas y secciones fueron creadas y multiplexadas utilizando el conjunto de herramientas provisto por OpenCaster y *ts_util*[15].

PAT

Al momento de extender la lista de servicios de un flujo de transporte, el número de PAT's que viajan se conserva igual. Existe una PAT por flujo de transporte. Sin embargo, el tamaño de la sección que transporta la PAT impone, como muestra la figura B.1, un máximo de 252 servicios, en promedio. Este valor podría disminuir ante la inclusión de descriptores o datos adicionales, pero este no es un caso común. Si además se deseara limitar el tamaño de la PAT a un único paquete, la cota se reduce a 41 servicios.

El motivo para utilizar una PAT de un único paquete radica en la drástica simplificación de la multiplexación y decodificación del flujo de transporte. Adicionalmente, implica un pequeño ahorro en paquetes de flujo de transporte (que podría ser utilizado, por ejemplo, para agregar nuevas PMT's, aunque esto no implique una diferencia sustancial).

SDT

La SDT transporta contenido textual descriptivo de los servicios y éste le confiere dependencia respecto de los servicios provistos por el flujo de transporte. Para realizar la medición del número máximo de servicios que permite esta tabla, se utilizaron nombres de una longitud promedio de 9 caracteres. Como puede apreciarse en la figura B.1, una sección establece una cota superior de 50 servicios, mientras que una SDT de un único paquete puede describir hasta 8 servicios. Esto convierte a las SDT en una limitación real para el número de servicios señalizables por la extensión del estándar ISDB-Tb.

B.2. Disponibilidad de paquetes nulos

Finalmente, resta analizar cómo influye la PMT sobre el número máximo de servicios que se pueden señalar. A diferencia de las otras dos, la PMT no cambia en tamaño en la medida en que se agregan servicios, dado que aparecen nuevas instancias en lugar de modificar las existentes. Por lo tanto la limitación del tamaño de sección no afecta a la PMT. El problema es que cada una requiere un PID distinto y cada PID requiere de un paquete de flujo de transporte distinto. La solución planteada inserta estos paquetes donde el flujo original contenía paquetes nulos. Esta sección analiza la suficiencia de estos paquetes para la inserción de PMT's.

Cada uno de los servicios de un flujo de transporte requiere la emisión de una PMT asociada, así sea un servicio regular o uno cuyos flujos elementales se transmiten por IP. El estándar ISDB-Tb establece que el intervalo máximo de tiempo entre instancias de la tabla es de 200 milisegundos. Siendo que una sección se transporta en, al menos, un paquete, y el número de paquetes en un intervalo de tiempo es necesariamente finito, el número de servicios también lo es en consecuencia.

Para este análisis se continúa con el ejemplo del Canal 23, presentado en el capítulo Introducción Técnica. De los 17,3Mbps disponibles del flujo de transporte, unos 2,4Mbps son ocupados por paquetes nulos, los cuales pueden ser reemplazados por una PMT de un servicio reubicado. Esta cifra representa unos 1500 paquetes nulos por segundo. La norma ISDB-Tb determina que las instancias de la PMT deben encontrarse en un período regular de 200ms, un quinto de segundo. Así, la cantidad de paquetes disponibles para insertar PMT's se aproxima a los 300.

Esta estimación indica que el número máximo de servicios, si de la PMT depende, tiene una cota superior en 300. El problema de esta aproximación es que ignora la variabilidad de *bitrate* utilizado; y la cota superior real es el mínimo local absoluto de número de paquetes nulos en períodos de 200ms en todo el flujo de transporte. Siendo que este es arbitrariamente largo, es imposible determinar este número con certeza.

La figura B.2 presenta los resultados de una medición sobre un fragmento real de flujo de transporte de Canal 23. Muestra el conteo de paquetes nulos disponibles en los siguientes 200ms de reproducción para cada instante de un fragmento de flujo de transporte de dos minutos y medio de duración. Se puede apreciar que el mínimo alcanzado es de 271 paquetes para un instante que ronda el segundo 86. Esto quiere decir que el número máximo de PMT's que se puede incluir dada esta sección del flujo de transporte supera las 200 instancias.

Como la figura figura B.1 muestra, es indistinto el hecho de que la tabla ocupe un único paquete o una sección completa, dado que no es el tamaño la limitación de la tabla, sino el número de repeticiones posibles en el intervalo de tiempo establecido por el estándar ISDB-Tb. En conclusión, una estimación conservadora del número máximo de servicios que se pueden incluir en un flujo de transporte con características similares al de Canal 23, es de **200 servicios**.

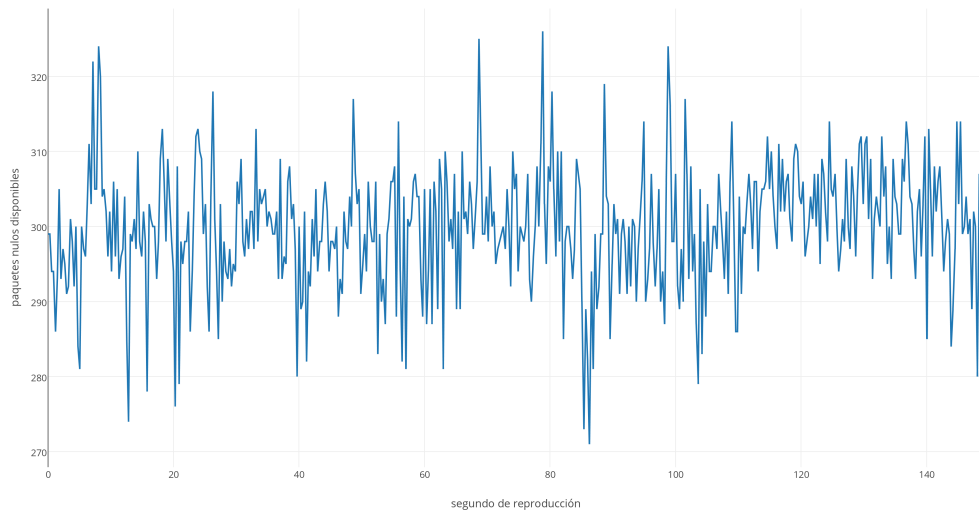


Figura B.2: Gráfico lineal de disponibilidad de paquetes nulos en el flujo de transporte

Del conjunto de entidades analizadas deviene el hecho de que la SDT es el limitante real para el número de servicios señalizables. Para un conjunto de servicios cuya longitud de nombres promedia los 9 caracteres, la cota superior es de **50 servicios por flujo de transporte**. El resto de las tablas no influyen directamente sobre esta limitación.

Apéndice C

Estudio de Wari

Wari es el reproductor de televisión digital utilizado como base para la construcción de Mara. Está destinado a computadoras personales sin restringirse a un único sistema operativo. La mejor fuente de estudio para Wari es claramente el código, que es abierto y se puede conseguir a través del sitio del Lifa[20]. No obstante, la comprensión del mismo requiere de cierta asistencia por la complejidad del dominio.

Es muy importante mencionar que Wari no existe como una entidad autónoma sino que pertenece a un proyecto más importante, denominado *Kuntur*. Wari es, de hecho, una pieza de frontend que le da apariencia a los mecanismos utilizados de fondo. *Zamba* es otra de estas piezas que utiliza exactamente los mismos mecanismos, pero en lugar de estar orientado a computadoras personales, se orienta a *Set Top Boxes*¹.

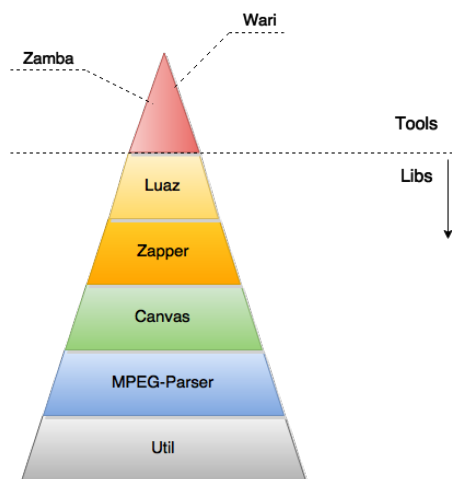


Figura C.1: Diagrama de las librerías de Kuntur.

¹Denominación para los dispositivos que toman como entrada una señal y entregan como salida contenido apto para la reproducción en un televisor. Esta entrada puede y suele ser un sintonizador. También, puede contener múltiples entradas.

Kuntur está escrito mayoritariamente en C++ y clasifica sus componentes en dos grupos. Las *Libs*² proveen los mecanismos de *backend* mencionados y las *Tools*³ son productos finales que se apoyan sobre estas. Ambos grupos se exponen en la figura C.1.

La punta de la pirámide de esta figura representa la última categoría, las tools, y es la componente que comienza el hijo de ejecución principal y luego lo delega a las libs. Las mismas se comportan como un *framework*, adueñándose luego del control. Cada bloque representa un módulo de Kuntur, que depende de los que se encuentran subyacentes en el gráfico. Cada uno de estos cumple una tarea específica del dominio que se describen en la Tabla C.1.

²Proveniente del término en inglés *libraries*.

³En inglés, herramientas. Pero el termino resulta muy confuso porque Zamba y Wari no encajan perfectamente en la definición de herramientas, por eso se usará el término tools para referirse a ellos.

Lib	Descripción
Luaz	Es la última capa de abstracción que se les provee a las <i>tools</i> . Idealmente, las mismas interactúan exclusivamente con esta librería de métodos y <i>callbacks</i> . El nombre deriva de la idea de que Luaz es un envoltorio de Zapper con <i>bindings</i> ⁴ a Lua ⁵ . Esto quiere decir que las Tools pueden limitarse a un conjunto de <i>scripts</i> Lua para implementar la interfaz que, por ser un lenguaje interpretado, colabora en el desarrollo rápido de interfaz de usuario.
Zapper	El término <i>Zapper</i> refiere a cualquier dispositivo que sirva para hacer <i>zapping</i> o cambiar canales, como un control remoto. En Kuntur, el Zapper es la única abstracción a la que acceden las tools (así sea directamente o a través de Luaz), y esta les permite realizar todas las tareas propias de un reproductor de TV.
Canvas	Responsable de la entrada/salida hacia el usuario. El principal problema que resuelve es la independización de las aplicaciones respecto del software de abstracción de hardware. De esta manera, se pueden desarrollar tools que corran sobre muy diversas plataformas, usando SDL, Cairo, etc.
MPEG-Parser	Permite que Zapper provea abstracciones basadas en los conceptos descritos en apéndice A. El contenido recibido a través de un sintonizador es completamente dinámico y requiere la demultiplexación compleja de secciones, tablas, servicios, etc. Como gran parte de los contenidos cambian con el tiempo, MPEG-Parser hace un uso fuerte de <i>callbacks</i> . Así, si un cliente como Zapper hace uso de la librería, provee métodos para ser invocados ante eventos y así es como los cambios en el flujo de transporte llegan hasta el cliente. Por ejemplo, si la programación sufre un cambio, es posible adaptar la guía electrónica de programación dinámicamente.
Util	Es una librería que incluye un conjunto de entidades necesarias en el dominio de televisión digital, como <i>buffers</i> binarios o un modelo para representación de instantes ⁶ (<i>Julian Date</i> , <i>POSIX time</i> y variantes), utilizados en el flujo de transporte. Cabe notar, también, que por ser los reproductores de televisión digital sistemas de tiempo real, las librerías requieren de <i>multithreading</i> y <i>util</i> provee los mecanismos de comunicación utilizados por el resto de las librerías.

Tabla C.1: Descripción de las diferentes libs de Kuntur.

C.1. Multithreading

Se mencionó en reiteradas ocasiones la necesidad de *multithreading* en el dominio de Wari y los reproductores de TV Digital. Para resolver esta dependencia, el proyecto Kuntur utiliza las librerías **Boost**[25], en particular, el módulo **Thread**. Ésta librería provee una abstracción portable para el manejo de múltiples *threads* y mecanismos de sincronización, de interfaz similar a los *Posix Threads*[26].

En lo que a parametrización respecta, Kuntur no requiere una configuración específica de políticas de *scheduling* o exclusión mutua, usando las que utiliza por defecto **Boost.Thread**, que a su vez son dictadas por la plataforma subyacente. Para información más específica se puede consultar la documentación que explora estos temas[27].

C.2. Compilación de Wari y Mara

Kuntur provee herramientas para la construcción de cualquier lib o tool que se desee, a través de *scripts python*. Sin embargo, una cuota de asistencia es útil en el proceso. Para empezar, es necesario contar con el código fuente. El mismo puede ser obtenido del sitio oficial del Lifa[20]. Si es Mara el software que se desea compilar, es necesario aplicar el parche provisto junto a este informe previamente. El proceso de compilación y ejecución es el mismo para Wari y Mara, y se detallan a continuación para una distribución de Linux *debian-based*:

1. Descomprimir los contenidos en *dir-kuntur*, sea `/home/user/Development/kuntur`.
2. Crear un contenido para la construcción *out of source*, sea `dir-kuntur/install.UNIX`.
3. Resolver las dependencias de software. Para un *debian-based* común, los siguientes paquetes suelen ser suficientes:
 - `cmake`
 - `lua5.1`
 - `liblua5.1-0-dev`
 - `zlib1g-dev`
 - `libxerces-c-dev`
 - `libev-dev`
 - `libboost-filesystem-dev`
 - `libboost-thread-dev`
 - `libboost-date-time-dev`

⁴Mecanismo para posibilitar la invocación de métodos o clases desde un language a otro.

⁵Lenguaje de *scripting* con semánticas extensibles, desarrollado en 1993 por la Universidad Pontificia Católica de Río de Janeiro.

⁶Representación de instantes en el tiempo a través del conteo de días desde un instante representativo prefijado.

- libglib2.0-dev
 - libfreetype6-dev
 - libgtk-3-dev
 - libboost-math-dev
 - libdb-dev
 - libvlc-dev
 - vlc
4. En el directorio *dir-kuntur*/config, crear el archivo `SetupUser.cmake` con los siguientes contenidos. Este archivo sirve para configurar el tipo de compilación deseada que, por simplicidad, será mínima para análisis.

```
1 SET(CMAKE_BUILD_TYPE "Debug" )
2 SET(SYS_PLAYER_USE_VLC 1 )
```

5. Con *dir-kuntur*/install.UNIX como ruta actual, ejecutar `../build/build.py -t wari`. Esta operación puede demorar unos minutos.
6. En *dir-kuntur*/install.UNIX/rootfs/bin/ se encontrarán todas las tools compiladas. En caso de utilizar únicamente los pasos enumerados hasta ahora, sólo se hallará Wari.
7. Para ejecutar Wari o Mara, se debe utilizar el comando(desde *dir-kuntur*/install.UNIX):

```
1 ./rootfs/bin/wari
```

C.2.1. Troubleshooting

Hay un motivo muy común por el que la ejecución de las tools incurre en un *segmentation fault* y es la presencia de un *plugin* para VLC relacionado con Lua. El problema se soluciona fácilmente eliminando tal plugin.

```
1 rm /usr/lib/vlc/plugins/lua/liblua_plugin.so
```

También es necesario considerar que para ejecutar Wari, se requiere poseer un dispositivo de sintonización de TV digital. La ausencia de tal dispositivo incurre en un error de ejecución fatal. No obstante, es posible ejecutar Wari utilizando como entrada un archivo de *Transport Stream*, en lugar de un sintonizador. Para hacerlo, se debe cambiar la línea de ejecución, agregando los siguientes parámetros:

```
1 ./rootfs/bin/wari --set tuner.provider.use=tsdata:tuner.
   provider.tsdata.use=file:tuner.provider.tsdata.file.bitrate=
   XXXXXXXX:zapper.mpeg.player=ts
```

El campo denotado con "XXXXXXX" debe ser reemplazado por el bitrate real del flujo de transporte. Este valor puede estar indicado en el nombre del archivo o, en caso de no poseerse, puede inferirse de los valores de PCR del campo de adaptación de algún flujo elemental. *Ts_util*[15] provee herramientas para realizar este cálculo.

Bibliografía

- [1] *Latin American Multichannel Advertising Council*,
<http://www.lamac.org/america-latina/metricas/total-por-tv-paga>
- [2] *Latin American Multichannel Advertising Council*,
<http://www.lamac.org/argentina/metricas/total-por-tv-paga>
- [3] *Unión de Naciones Suramericanas*,
<http://www.unasursg.org>
- [4] *Argentina conectada*,
<http://www.argentinaconectada.gob.ar>
- [5] Roberto C. Lauro, *Transmisión de televisión digital terrestre en la Norma ISDB-T b*.
Cengage Learning Argentina, 1a Ed., 2010.
- [6] Roberto C. Lauro, *Transmisión de televisión digital terrestre en la Norma ISDB-T b*.
Cengage Learning Argentina, 1a Ed., 2010.
Página 17 (capítulo 1 - Sección 2)
- [7] ITU-T Newslog, *IPTV Standardization on Track Say Industry Experts*,
Octubre 2006
- [8] *ISO/IEC 13818-1*,
Information technology - Generic coding of moving pictures and associated audio
information: Systems.
2a Ed., Diciembre 2000
- [9] *Digital Broadcasting Expert Group*(Dibeg)
<http://www.dibeg.org/techp/aribstd/harmonization.html>
- [10] *RFC 1112*, Host Extensions for IP Multicasting.
Agosto 1989
- [11] *RFC 768*, User Datagram Protocol,
Agosto 1980
- [12] *Association of radio industries and business*,
<http://arib.or.jp/english>

- [13] *Associação Brasileira de Normas Técnicas*,
<http://www.abnt.org.br>
- [14] *ARIB STD-B21*,
Receiver for digital broadcasting, ARIB standard (Desirable Specifications)
Versión 4.6, 14/3/2007
- [15] *ts_util*,
https://github.com/AquilesCanta/ts_util
- [16] *Televisión Digital Abierta*,
<http://www.tda.gob.ar/>
- [17] Robert C. Martin, *The Open-Closed Principle*,
C++ Report, Enero 1996
- [18] *GNU Lesser General Public License*,
<http://www.gnu.org/licenses/>
- [19] *GNU General Public License*,
<http://www.gnu.org/licenses/>
- [20] *Laboratorio de Investigación y Formación en Informática Avanzada*,
<http://www.lifia.info.unlp.edu.ar>
- [21] *VLC Media Player*,
<https://www.videolan.org/vlc/>
- [22] Alan C. Shaw, *Real-Time Systems*,
John Wiley & Sons, 2001
- [23] *Digital Television Closed Caption Channel*, CEA-708-E (ANSI)
- [24] *Object based broadcasting curation, responsiveness and user experience*
BBC WHP 285, Septiembre 2014
- [25] *Librerías Boost*,
<http://www.boost.org/>
- [26] *Posix threads API*,
IEEE Std 1003.1c-1995
- [27] *Boost documentation on threading, scheduling and executors*,
http://www.boost.org/doc/libs/1_58_0/doc/html/thread.html