



**Análisis y evaluación de la capa de transporte para el Estándar de Transmisión de
Próxima Generación ATSC 3.0.**

Mediavilla López, Luis Javier

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica y
Telecomunicaciones

Ing. Olmedo Cifuentes, Gonzalo Fernando Ph.D.

23 de agosto del 2021

Curiginal

Document Information

Analyzed document	Luis_Medivilla_tesis_final.pdf (D111181181)
Submitted	8/11/2021 7:25:00 PM
Submitted by	Olmedo Cifuentes Gonzalo Fernando
Submitter email	gfolmedo@espe.edu.ec
Similarity	6%
Analysis address	gfolmedo.espe@analysis.urkund.com



GONZALO FERNANDO
OLMEDO CIFUENTES

Ing. Olmedo Cifuentes, Gonzalo Fernando Ph.D.

Sources included in the report

W	URL: http://repositorio.espe.edu.ec/bitstream/21000/24173/1/T-ESPE-044408.pdf Fetched: 7/19/2021 4:31:50 AM		1
W	URL: https://www.atsc.org/wp-content/uploads/2016/10/A330-2016-Link-Layer-Protocol-1.pdf Fetched: 8/11/2021 7:26:00 PM		1
W	URL: https://revistatelematica.cujae.edu.cu/index.php/tele/article/download/237/221 Fetched: 5/16/2021 5:31:26 PM		2
SA	Universidad de las Fuerzas Armadas ESPE / Proyecto_Andres_Vila.pdf Document Proyecto_Andres_Vila.pdf (D98928069) Submitted by: gfolmedo@espe.edu.ec Receiver: gfolmedo.espe@analysis.urkund.com		40
W	URL: https://www.researchgate.net/publication/292345988_Delivery_of_ATSC_30_Services_With_MPEG_Media_Transport_Standard_Considering_Redistribution_in_MPEG-2_TS_Format Fetched: 8/11/2021 7:26:00 PM		1
W	URL: http://ieeexplore.ieee.org/document/7389995/ Fetched: 8/11/2021 7:26:00 PM		2
W	URL: https://github.com/justman/libatsc3.git Fetched: 8/11/2021 7:26:00 PM		2
W	URL: https://eprints.ucm.es/38081/1/T37363.pdf Fetched: 11/5/2019 1:24:15 PM		1
W	URL: https://www.atsc.org/wp-content/uploads/2017/10/A300-2017-ATSC-3-System-Standard-3.pdf Fetched: 8/11/2021 7:26:00 PM		1
W	URL: https://ieeexplore.ieee.org/document/7414448 Fetched: 8/11/2021 7:26:00 PM		1
W	URL: https://www.ngbp.org/2019/03/2019-03-10-libatsc3-core-release.html Fetched: 8/11/2021 7:26:00 PM		2

URI: <https://www.thehinducastheline.com/content/entry/6229/atsc-3.0-details-explained-part-4>



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**Análisis y evaluación de la capa de transporte para el Estándar de Transmisión de Próxima Generación ATSC 3.0.**” fue realizado por el señor **Mediavilla López, Luis Javier** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 23 de agosto del 2021



Firmado electrónicamente por:
GONZALO FERNANDO
OLMEDO CIFUENTES

.....
Ing. Olmedo Cifuentes, Gonzalo Fernando, Ph.D.

C.C.: 1711696342



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

RESPONSABILIDAD DE AUTORÍA

Yo, **Mediavilla López, Luis Javier**, con cédula de ciudadanía n° 1725864019, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Análisis y evaluación de la capa de transporte para el Estándar de Transmisión de Próxima Generación ATSC 3.0.”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 23 de agosto del 2021

Mediavilla López, Luis Javier

C.C.: 1725864019



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Mediavilla López, Luis Javier**, con cédula de ciudadanía n° 1725864019, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “**Análisis y evaluación de la capa de transporte para el Estándar de Transmisión de Próxima Generación ATSC 3.0.**” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 23 de agosto del 2021

.....
Mediavilla López, Luis Javier

C.C.: 1725864019

Dedicatorias

Dedico este trabajo de titulación a Dios, que con su guía y amor me permitió disfrutar cada momento del proceso de mi carrera, hasta el punto de concluirla con éxito.

A mis padres, los pilares fundamentales en mi vida y proceso de formación que, con su ejemplo y amor, obtuve la guía para alcanzar los objetivos que me he propuesto y me han proporcionado todas las herramientas para alcanzar la meta tan anhelada.

A mi abuelita Rosita quien siempre confió en mí y con sus palabras y acciones me daban la fuerza para continuar cada día.

A mis hermanos y amigos que con su paciencia y apoyo han aportado de gran manera al desarrollo de mis actividades dentro de mi prestigiosa universidad.

LUIS JAVIER MEDIAVILLA LÓPEZ

Agradecimientos

A mis padres, los pilares fundamentales en mi vida y proceso de formación no tengo más que decirles gracias, gracias por su ejemplo, que me sirvió de guía para desenvolverme de la mejor manera en cada una de las actividades y situaciones diaria que se presentaban en la universidad, gracias por su amor y compañía que, en los momentos más fuertes, fueron el apoyo para continuar y no desfallecer y poder concluir con éxito esta etapa tan gratificante.

A mis maestros y compañeros con los cuales convivía diariamente compartiendo experiencias y conocimientos, que me hicieron sentir como si la universidad fuera mi segundo hogar, donde en cada proyecto o tarea nos unía para realizarlos con éxito.

A mi abuelita Rosita que con cada pequeña acción y cada palabra de aliento y cariño me daba la fuerza para desarrollar de la mejor manera mis tareas, siempre siguiendo sus consejos honestidad y rectitud.

Finalmente, a la Universidad de la Fuerzas Armadas – ESPE y docentes por darme la oportunidad de estudiar y desarrollarme en el campo laboral y académico; y sobre todo a mi tutor un excelente maestro y persona.

LUIS JAVIER MEDIAVILLA LÓPEZ

Índice de Contenidos

Contenido	
Urkund.....	2
Certificación.....	3
Responsabilidad de Autoría.....	4
Autorización de Publicación.....	5
Dedicatorias	6
Agradecimientos.....	7
Índice de Contenidos.....	8
Índice Tablas.....	14
Índice Figuras.....	15
Resumen.....	19
Palabras Clave	19
Abstract	20
Key Words.....	20
Capítulo I.....	21
Planteamiento del Problema de Investigación	21
Antecedentes.....	21
Justificación e Importancia.....	23
Alcance del proyecto.....	24
Objetivos.....	25

Objetivo General	25
Objetivos específicos.....	25
Organización del Trabajo de Titulación	25
Capítulo II.....	27
Fundamento Teórico	27
La Televisión Digital.....	27
Estándares de Televisión Digital.....	28
Estándar ATSC.....	29
Estándar de Próxima Generación ATSC 3.0.....	30
Estructura General.....	31
Descripción General del Sistema ATSC 3.0.....	35
Capítulo III.....	36
Capa de Transporte.....	36
Descripción General de la Capa de Transporte para ATSC 3.0	36
Capa de Enlace (A/330 - Link Layer)	38
Servicios	39
Encapsulación de paquetes.....	39
Segmentación y reensamblaje	40
Concatenación	40
Reducción de la sobrecarga	40
Transmisión de señalización	41

	10
Arquitectura del Sistema	41
Formato del paquete ALP.....	42
Encapsulación del paquete ALP	42
Encabezado base.....	42
Encabezado adicional	45
Encabezado de extensión	49
Sub_Stream Identification	49
Header Extension.....	50
Señalización de encapsulación	51
Additional Header for Signaling Information.....	52
Extensión Tipo de Paquete	55
Additional header for Type Extension	56
Encapsulación de paquetes MPEG-2 TS.....	57
Estructura del paquete ALP al encapsular paquetes TS.....	57
Eliminar el Byte de Sincronización (Sync) de Paquetes MPEG-2 TS al encapsular en ALP	60
Eliminar paquetes nulos (Null packet) de paquetes MPEG-2 TS al encapsular en paquetes ALP.....	62
Eliminación del encabezado del paquete MPEG-2 TS al encapsular en un paquete ALP	66
Comprensión del encabezado IP	70
ROHC (Robust Header Comprension).....	71

Módulo Adaptación.....	71
Señalización, Entrega, Sincronización, y Protección de errores (A/331)	74
Características de los protocolos MMTP y ROUTE-DASH	75
Reglas para el uso de MMTP o ROUTE-DASH	76
Señalización de servicios	77
Flujos broadcast	80
Señalización de bajo nivel (LLS)	82
Formato de la tabla LSS.....	82
Identificador de servicios	85
Asignación de dirección IP	86
Tabla de lista de servicios (SLT).....	87
Descripción de la sintaxis de SLT	88
Enlace de Programador / Estudio a Transmisor (Scheduler / Studio to Transmitter Link A/324)	94
DSTP (Protocolo de transporte de fuente de datos)	95
ALTP (Protocolo de transporte del protocolo de capa de enlace ATSC).....	96
STLP (Protocolo de transporte de enlace de estudio a transmisor)	97
Capítulo IV	99
Análisis de la capa de transporte del estándar ATSC 3.0	99
Softwares para el análisis de flujos de muestra ATSC 3.0	99
Software de distribución libre para el análisis de flujos ATSC 3.0	100
Biblioteca libatsc3.....	100

Instalación de libatsc3	101
Herramienta principal de la biblioteca libatsc3 (atsc3_listener_metrics_ncurses) .	107
Datos de salida de la herramienta principal de la biblioteca libatsc3 (atsc3_listener_metrics_ncurses)	108
Global ATSC 3.0 Statistics	109
LLS Base table	110
SLT	111
Flow ATSC 3.0 Statistics	112
Flujos de datos muestra ATSC 3.0 en formato PCAP de transmisiones ATSC 3.0 MMT y ROUTE-DASH en el mercado	113
Reproducción de los flujos de datos muestra ATSC 3.0	114
Análisis de los flujos de muestra ATSC 3.0 mediante la herramienta libatsc3_listener_metrics_ncurses	116
Análisis de flujos de muestra ATSC 3.0 MMT	118
Análisis de resultados de la sección Global ATSC 3.0 Statistics	120
Parte Izquierda flujo 2018-12-17-mmt-airwavz-recalp.pcap	121
Parte derecha flujo 2018-12-17-mmt-airwavz-recalp.pcap	122
Análisis de resultados de la sección Flow ATSC 3.0 Statistics	128
Análisis de flujos de muestra ATSC 3.0 ROUTE-DASH	130
Análisis de resultados de la sección Global ATSC 3.0 Statistics	130
Parte Izquierda flujo 01.10.00.gz.pcap	131
Parte derecha flujo 01.10.00.gz.pcap	132

Análisis de resultados de la sección Flow ATSC 3.0 Statistics.....	134
Capítulo IV	136
Conclusiones y Recomendaciones.....	136
Conclusiones	136
Recomendaciones	138
Bibliografía	140

Índice Tablas

Tabla 1 <i>Conjunto de normas que conforman el Estándar ATSC 3.0</i>	33
Tabla 2 <i>Valores y significado de packet_type</i>	43
Tabla 3 <i>Payload_Configuration (PC), Valores de campo y longitud total de encabezado base</i>	45
Tabla 4 <i>Rango de valores y descripción de extension_type</i>	51
Tabla 5 <i>Valores de código para Signaling_type</i>	53
Tabla 6 <i>Valores de código para Signaling_format</i>	54
Tabla 7 <i>Valores de código para Signaling_encoding</i>	55
Tabla 8 <i>Valores de código para extended_type</i>	57
Tabla 9 <i>Valores de campo de encabezado ALP al eliminar paquetes TS nulos</i>	61
Tabla 10 <i>Valores de campo de encabezado ALP al eliminar paquetes TS nulos</i>	64
Tabla 11 <i>Valores de campo de encabezado ALP al eliminar el encabezado del paquete MPEG-2 TS</i>	68
Tabla 11 <i>Descripción de los valores para la tabla SLT según la norma A/331</i>	88
Tabla 12 <i>Codificación para el parámetro serviceCategory según la norma A/331</i>	92
Tabla 13 <i>Codificación para el atributo slsProtocol según la norma A/331</i>	93
Tabla 14 <i>Codificación para el atributo type según la norma A/331</i>	93
Tabla 15 <i>Productos en el mercado especializados para el análisis del estándar ATSC 3.0</i>	99
Tabla 16 <i>Información base LLS de la muestra 2018-12-17-mmt-airwavz-recalp.pcap</i>	122
Tabla 17 <i>Información SLT de la muestra 2018-12-17-mmt-airwavz-recalp.pcap por servicio</i>	124
Tabla 18 <i>Descripción de atributos de la sección Flow ATSC 3.0 Statistics</i>	129
Tabla 19 <i>Información base LLS de la muestra 01.10.00.gz.pcap</i>	132
Tabla 20 <i>Información SLT de la muestra 01.10.00.gz.pcap</i>	133

Índice Figuras

Figura 1 <i>Estándares de TDT adoptado por cada país en el Mundo.</i>	28
Figura 2 <i>Diagrama de bloques general del sistema ATSC.</i>	29
Figura 3 <i>Arquitectura en capas del sistema ATSC 3.0.</i>	31
Figura 4 <i>Estructura y conjunto de documentos de ATSC 3.0.</i>	32
Figura 5 <i>Esquema de la interconexión entre la capa física y la capa de transporte.</i>	37
Figura 6 <i>Diagrama lógico del transporte de datos de la capa de enlace de ATSC 3.0.</i>	38
Figura 7 <i>Diagrama de bloques y de interfaz de la arquitectura de ALP.</i>	41
Figura 8 <i>Formato del paquete ALP.</i>	42
Figura 9 <i>Estructura del encabezado base para la encapsulación de paquete ALP.</i>	43
Figura 10 <i>Estructura del encabezado adicional para paquetes individuales.</i>	46
Figura 11 <i>Estructura del encabezado adicional para segmentación paquetes individuales.</i>	47
Figura 12 <i>Estructura del encabezado adicional para concatenación de paquetes individuales.</i>	48
Figura 13 <i>Campos definidos para estructura del encabezado header extension.</i>	50
Figura 14 <i>Estructura del paquete de señalización ALP (encabezado base y encabezado adicional)</i>	52
Figura 15 <i>Sintaxis de los campos del encabezado Additional Header for Signaling Information.</i>	53
Figura 16 <i>Estructura de los paquetes ALP que contiene un encabezado Additional header for type extension.</i>	55
Figura 17 <i>Sintaxis para campo Additional header for type extension.</i>	56
Figura 18 <i>Estructura del encabezado al encapsular paquetes con formato MPEG-2 TS en paquetes ALP.</i>	58

Figura 19 <i>Sintaxis de los campos del encabezado de paquete de capa de enlace ATSC 3.0 para MPEG-2 TS.</i>	58
Figura 20 <i>Proceso de encapsular 8 paquetes MPEG-2 TS al eliminar el byte de sincronización (Sync) en un paquete ALP.</i>	61
Figura 21 <i>Proceso de encapsular en un paquete ALP paquete MPEG-2 TS eliminando paquetes nulos.</i>	63
Figura 22 <i>Proceso de encapsular 6 paquetes MPEG-2 TS eliminando 2 paquetes nulos en un paquete ALP.</i>	65
Figura 23 <i>Proceso desencapsular 6 paquetes MPEG-2 TS y agregar paquetes nulos desde un paquete ALP.</i>	66
Figura 24 <i>Proceso de encapsular paquetes MPEG-2 TS eliminando su encabezado en un paquete ALP.</i>	67
Figura 25 <i>Proceso de desencapsular paquetes MPEG-2 TS utilizando el modo eliminación de encabezado MPEG-2 TS desde un paquete ALP.</i>	69
Figura 26 <i>Estructura de la compresión y descompresión del encabezado IP.</i>	70
Figura 27 <i>Estructura funcional de alto nivel de la compresión de encabezados IP en ALP.</i>	73
Figura 28 <i>Pila de protocolos de ATSC 3.0 conceptual.</i>	74
Figura 29 <i>Pila de protocolos en el receptor ATSC 3.0.</i>	78
Figura 30 <i>Relación de las tablas SLT y SLS en los protocolos MMTP y ROUTE-DASH.</i>	79
Figura 31 <i>Asignación de LLS_group_id por cada emisora (grupo) o estación de televisión.</i>	81
Figura 32 <i>Diagrama de flujo de la sintaxis de las tablas LLS.</i>	83
Figura 33 <i>Clonación de la biblioteca libatsc3 en la dirección /Escritorio/tesis.</i>	102

Figura 34 <i>Módulos que requieren usuario y contraseña para la descarga de la biblioteca libatsc3.</i>	103
Figura 35 <i>Ingreso a la carpeta srt, para la verificación de su correcta descarga.</i>	104
Figura 36 <i>Clonación correcta de la carpeta srt, directamente desde el GitHub.</i>	105
Figura 37 <i>Instalación completa de la biblioteca libatsc3.</i>	106
Figura 38 <i>Herramienta de análisis para flujos de muestra ATSC 3.0.</i>	107
Figura 39 <i>Interfaz de la herramienta atsc3_listener_metrics_ncurses.</i>	108
Figura 40 <i>Datos de salida de la sección Global ATSC 3.0 Statistics.</i>	109
Figura 41 <i>Presentación de información propia del flujo ATSC 3.0.</i>	112
Figura 42 <i>Flujos de muestra ATSC 3.0 para su análisis.</i>	114
Figura 43 <i>Interfaces de tarjeta de red disponibles.</i>	115
Figura 44 <i>Finalización de la reproducción del flujo ATSC 3.0 MMTP.</i>	116
Figura 45 <i>Finalización de la reproducción del flujo ATSC 3.0 ROUTE-DASH.</i>	116
Figura 46 <i>Herramientas de análisis proporcionada por la biblioteca libatsc3.</i>	117
Figura 47 <i>Flujo a ser analizado.</i>	118
Figura 48 <i>Ejecución de la herramienta latsc3_listener_metrics_ncurses.</i>	119
Figura 49 <i>Herramientas tcp replay y atsc3_listener_metrics_ncurses en p0s3 lista para ser ejecutadas.</i>	119
Figura 50 <i>Información del flujo 2018-12-17-mmt-airwavz-recalp.pcap.</i>	120
Figura 51 <i>Sección Global ATSC 3.0 Statistics dividida por partes.</i>	121
Figura 52 <i>Parte izquierda de la sección Global ATSC 3.0 Statistics.</i>	122
Figura 53 <i>Servicios del flujo de muestra 2018-12-17-mmt-airwavz-recalp.pcap.</i>	123
Figura 54 <i>Información de la sección Flow ATSC 3.0 Statistics del flujo 2018-12-17-mmt-airwavz-recalp.pcap.</i>	129
Figura 55 <i>Información del flujo 01.10.00.gz.pcap</i>	130
Figura 56 <i>Sección Global ATSC 3.0 Statistics dividida por partes.</i>	131

Figura 57 <i>Parte izquierda de la sección Global ATSC 3.0 Statistics.</i>	132
Figura 58 <i>Servicios del flujo de muestra 01.10.00.gz.pcap.</i>	133
Figura 59 <i>Informacion de la sección Flow ATSC 3.0 Statistics del flujo 01.10.00.gz.pcap.</i>	135

Resumen

En el presente proyecto de titulación se define la estructura de la capa de transporte para el estándar de transmisión de próxima generación ATSC 3.0. El cuál es el primero en establecer un sistema híbrido donde contenido multimedia, servicios interactivos pueda ser transportada a través de redes de banda ancha, de radiodifusión o redes híbridas que combinen las características de cada uno de estas, con una alta calidad y eficiencia.

Se establece los conceptos definidos en la norma A/324 que establece los protocolos requeridos para la correcta interconexión entre la capa de transporte y la capa física, la norma A/330 que define el protocolo para la encapsulación de fuentes de datos pertenecientes a diferentes protocolos en un solo formato y la norma A/331 que define los protocolos de entrega de contenido y servicios utilizados por estándar ATSC 3.0, finalizando con el análisis de muestra ATSC 3.0 capturadas en el mercado utilizando herramientas de código abierto.

Palabras Clave

- **CAPA DE TRANSPORTE**
- **ATSC 3.0**
- **TRANSPORTE HÍBRIDO**
- **CAPA DE ENLACE**
- **SEÑALIZACIÓN Y ENTREGA**

Abstract

This project qualification defines the transport layer structure for the next generation transmission standard ATSC 3.0. Which is the first to establish a hybrid system where multimedia content, interactive services can be transported through broadband networks, broadcasting or hybrid networks that combine the characteristics of each of these, with high quality and efficiency.

It establishes the concepts defined in the A / 324 standard that establishes the protocols required for the correct interconnection between the transport layer and the physical layer, the A / 330 standard that defines the protocol for the encapsulation of data sources belonging to different protocols in a single format and the A / 331 standard that defines the content and service delivery protocols used by the ATSC 3.0 standard, ending with the analysis of the ATSC 3.0 sample captured in the market using open source tools.

Key Words

- **TRANSPORTATION LAYER**
- **ATSC 3.0**
- **HYBRID TRANSPORT**
- **LINK LAYER**
- **SIGNALING AND DELIVERY**

Capítulo I

Planteamiento del Problema de Investigación

Antecedentes

La televisión se ha transformado en uno de los dispositivos con mayor impacto sobre la población mundial, el desarrollo de las técnicas de procesamiento digital permitió una migración acelerada alrededor del mundo, pasando del sistema de televisión analógica al sistema de televisión digital terrestre (TDT), es así que en el año 2018 la cifras otorgadas por la Unión Internacional de Telecomunicaciones (UIT) muestra que el 37 % de los países completaron la migración hacia el sistema TDT y el 31 % se encuentran próximos a completar la misma (ITU, 2018).

Con la implementación del sistema de televisión TDT se optimiza el espectro radioeléctrico con lo cual se realiza transmisiones con resoluciones de alta definición y alta fidelidad además de otorgarnos servicios que involucren cierto grado de interactividad con el espectador. Para esto la UIT reconoce cuatro sistemas a nivel mundial para la transmisión de televisión digital terrestre (TDT) los cuales son: Advanced Television System Committee (ATSC), Digital Video Broadcasting – Terrestrial (DVB-T), Integrated Services Digital Broadcasting - Terrestrial (ISDB-T) y Digital Terrestrial Television Multimedia Broadcasting (DTMB) (UIT-R, 2016).

La UIT ha puesto especial interés en la convergencia de las nuevas tecnologías digitales con los servicios IP, esto debido a que la transmisión de servicios de televisión en múltiples dispositivos portátiles representa el 80 % de todo el tráfico de internet (ITU, 2019). Por lo que el comité Advanced Television Systems Committee (ATSC), el cual “es una organización internacional sin fines de lucro que desarrolla estándares voluntarios para la televisión digital” Plantea en el año 2015 la generación del estándar ATSC 2.0

(ATSC: Standard: A/107 - ATSC 2.0 Standard. A/107 – ATSC 2.0 Standard, 2016), con el fin de ser la primera organización en proponer un sistema de televisión híbrida e interactiva el cual tenga la capacidad de acceder a internet y ser compatible con su versión anterior ATSC (ATSC 1.0). Los servicios que se incluyen en la versión ATSC 2.0 están dirigidos a generar la posibilidad de transmitir contenido por internet o aire, por lo que no se le considera un estándar de segunda generación de TDT ya que sus servicios están orientados a la televisión interactiva e híbrida.

En el otoño del 2011, la organización ATSC forma el Grupo de Tecnología 3 (TG-3), para realizar el diseño de un sistema de transmisión de próxima generación, para lo cual se genera una convocatoria a nivel mundial para obtener información de los requisitos que debería implementarse en este sistema de acuerdo a los intereses de los países y organizaciones. Con esta base se desarrolla las capacidades de un nuevo sistema, convirtiéndose en una guía en la preparación del conjunto de estándares ATSC 3.0 con diferencias relevantes con sus versiones anteriores (ATSC 1.0 y ATSC 2.0) lo que lo hace incompatible con las mismas (Advanced Television Systems Committee, 2020).

ATSC 3.0 está diseñado con el propósito de integrar la transmisión por internet o aire además permite a los distribuidores ofrecer nuevos servicios de interactividad con los usuarios. Por lo que la UIT por medio de la circular administrativa CACE/940 emitida el 19 de diciembre 2019, publica que ha adoptado ATSC 3.0 como un estándar de transmisión digital recomendado, abriendo el camino para que los países interesados alrededor del mundo evalúen e implementen el estándar de transmisión digital basado en IP, el primero de su tipo en el mundo (ITU, 2019).

Justificación e Importancia

El Advanced Television Systems Committee (ATSC) traducido al español como “Comité de Sistemas de Televisión Avanzada” anunció que la “Unión Internacional de Telecomunicaciones” (UIT) adoptó ATSC 3.0 como un estándar de transmisión digital recomendado. Lo que allana el camino en todo el mundo para que los países evalúen y usen ATSC 3.0, que también se lo puede conocer mediante el nombre de “NEXTGEN TV” (Soseman, 2016). Esta nueva generación de televisión digital cambiará la forma tradicional en la que se presenta entretenimiento por video en el hogar, que actualmente es relativamente simple ya que consiste en que el espectador se coloque frente al televisor para ver las transmisiones que se encuentran programadas. Hoy en día los usuarios demandan tener la posibilidad de ver cualquier contenido en cualquier dispositivo y en cualquier lugar, este contenido es transmitido por diversos medios como cable, satélite, aire, internet o almacenado localmente. Por lo que la industria de la transmisión debe evolucionar para adaptarse a estos requerimientos. Los radiodifusores necesitan un nuevo sistema para admitir nuevos comportamientos de visualización, el cual debe incluir la capacidad de evolucionar con las demandas de los consumidores y consecuentemente proporcionar una extensibilidad que permita la adaptación futura. Para lo cual el estándar de transmisión de próxima generación ATSC 3.0 se encuentra destinado abordar estos requerimientos utilizando técnicas avanzadas de transmisión y codificación de video y audio y así poder brindar servicios nuevos y creativos a los espectadores (Chernock R. , 2015).

A pesar que el Ecuador adoptó en el año 2010 el estándar de “Televisión Digital ISDB-T”, también denominado ISDB-Tb, por sus adaptaciones brasileñas (MINTEL, s.f.), se ve indispensable continuar con el estudio de futuros sistemas de TV, que permitan transmisiones en formatos 4K y 8K en broadcaster y al mismo tiempo integren streaming

de TV, como una actividad por defecto a través de Internet, lo que podría ser utilizado en aplicaciones futuras en TV abierta o en TV por pago, permitiendo el planteamiento de nuevos modelos de negocios que se proyecten conforme a las necesidades de los usuarios, se requiere que los servicios broadcast que su funcionamiento solo se permitía en frecuencias temporales, ahora también pueden ser utilizados en áreas pequeñas como estadios o espectáculos que aglomeren gran cantidad de público y posean pantallas que soporten calidad Ultra HD, por lo que se ve al estándar ATSC 3.0 como la llave a que se consolide dichos modelos de negocios.

EL estándar ATSC 3.0 ha sido desarrollado durante los últimos años y todavía queda mucho trabajo de investigación por realizar y aportar, lo que motiva a que Rich Chernock, presidente de ATSC TG3 invite a todas las organizaciones interesadas en trabajar en el estándar, a unirse a participar. (Chernock R. , 2015), lo que genera una gran oportunidad para continuar trabajando en investigación de punta de TV Digital.

Alcance del proyecto

Con el desarrollo del presente proyecto de investigación se realizará el análisis y evaluación de la estructura de la capa de transporte definida para el estándar de televisión de siguiente generación ATSC 3.0, que permita conocer la interacción entre las transmisiones por aire y la recepción de datos multimedia por Internet y la importancia que genera esta tecnología en el desarrollo e implementación de servicios interactivos.

Para lo cual se recopilará información actual, concisa y clara que permita conocer y sintetizar las características y ventajas que posee la capa de transporte con estandarización ATSC 3.0, características de los protocolos de comunicación que se emplean para la entrega de medio y los formatos de archivos que utilizan cada uno de

ellos, con lo cual se podrá evaluar su estructura y así determinar el comportamiento, y la forma de comunicación del estándar.

Con esto se podrá estructurar un flujo de transporte estandarizado para ATSC 3.0. Con lo cual se pretende que sea establezca como la base de partida para la generación de nuevos proyectos de investigación que aporten al aprendizaje sobre las nuevas tecnologías de TV Digital.

Objetivos

Objetivo General

Analizar y Evaluar la estructura de la capa de transporte definida para el estándar de televisión de siguiente generación ATSC 3.0.

Objetivos específicos

- Recopilar información acerca de las características, ventajas y protocolos que posee la estructura de la capa de transporte del estándar ATSC 3.0.
- Analizar la estructura de la capa de transporte para el estándar ATSC 3.0.
- Evaluar los protocolos de comunicación para la entrega de medios multimedia y la interacción del canal de retorno que existe en la capa de transporte de ATSC 3.0.
- Estructurar un flujo de transporte estandarizado para ATSC 3.0.

Organización del Trabajo de Titulación

El presente trabajo se distribuyó de la siguiente manera:

El documento está constituido por 5 secciones, el capítulo I trata de los antecedentes, justificación, el alcance, los objetivos: general y específicos, en el capítulo II se desarrolla la parte teórica, donde se revisa diferentes conceptos referidos a la evolución de la televisión digital y la estructura del estándar de transmisión de próxima

generación ATSC 3.0, el capítulo III presenta los conceptos de estructuración de la capa de transporte, formato de paquetes, tablas de señalización y los protocolos implementados para el transporte de información utilizando el estándar ATSC 3.0, en el capítulo IV se realiza un análisis de la capa de transporte mediante la utilización de flujos de muestra ATSC 3.0, finalizando con el capítulo V donde se presenta las conclusiones y recomendaciones obtenidas al desarrollar del proyecto.

Capítulo II

Fundamento Teórico

La Televisión Digital

Un sistema de televisión transmite señales de audio y video a través de ondas electromagnéticas, lo cual permite que millones de hogares disfruten del entretenimiento y accedan a información. La primera transmisión de imágenes fue generada en el año de 1920, las cuales daban la posibilidad de observar un rostro humano, la televisión en blanco y negro tiene como inicio la finalización de la segunda guerra mundial, la televisión analógica a color comienza su desarrollo en el año de 1967 con la cual se obtenía una experiencia visual mejorada. La demanda de una mejor calidad en los servicios de audio y video por parte de los usuarios y los importantes avances tecnológicos en el campo del procesamiento de señales, conlleva al desarrollo de la televisión digital terrestre (TDT) (Song et al. 2015).

Esta nueva tecnología permite codificar y transmitir las señales de audio y video en un formato digital, sustituyendo a las señales analógicas, aprovechando el espectro radioeléctrico y aumentando la calidad del contenido transmitido. La señal TDT se transmite por ondas electromagnéticas terrestres, y se las recepta con el uso de antenas UHF convencionales. Los televisores actuales cuentan con un decodificador integrado, por lo que no requiere de dispositivos adicionales para receptar la señal (Chang & Narváez, 2015) .

Con la TDT se puede cubrir una área muy grande al adoptar la tecnología de red de frecuencia única, la cual puede utilizar el mismo canal TDT para transportar los mismos programas de TV con diferentes transmisores, se mejora la capacidad de anti-interferencia ya que, al utilizar una señal digitalizada permite trabajar con una secuencia

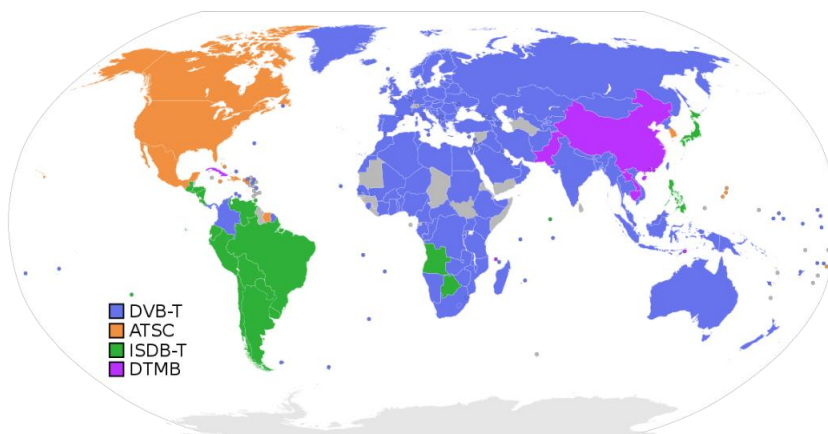
binaria (dos niveles), con lo cual al tener una amplitud de ruido que no sobrepase cierto nivel se lo puede eliminar de manera eficiente, se puede obtener al menos un programa HDTV, 10 programas SDT y 20 programas DTV con la calidad VHS en un solo canal de TV analógico dependiendo del esquema de comprensión de codificación de video utilizado en el sistema TDT, lo que reduce el consumo de ancho de (Song et al. 2015).

Estándares de Televisión Digital

La Union Internacional de las Telecomunicaciones definió de forma oficial a cuatro estándares de TDT que han sido desarrollados en todo el mundo: ATSC (Advanced Television System Committee) diseñado para sustituir el sistema de televisión analógica de color NTSC en los EEUU, DVB-T (Digital Video Broadcasting-Terrestrial) creado por la Organización de Radiodifusión de Video Digital de la Unión Europea, DTMB (Digital Terrestrial Television Multimedia Broadcasting) desarrolla en China e ISDB-T (Integrated Service Digital Broadcasting-Terrestrial) recomendado por Japón. Los estándares de TDT adoptados internacionalmente se observa en la figura 1.

Figura 1

Estándares de TDT adoptado por cada país en el Mundo.



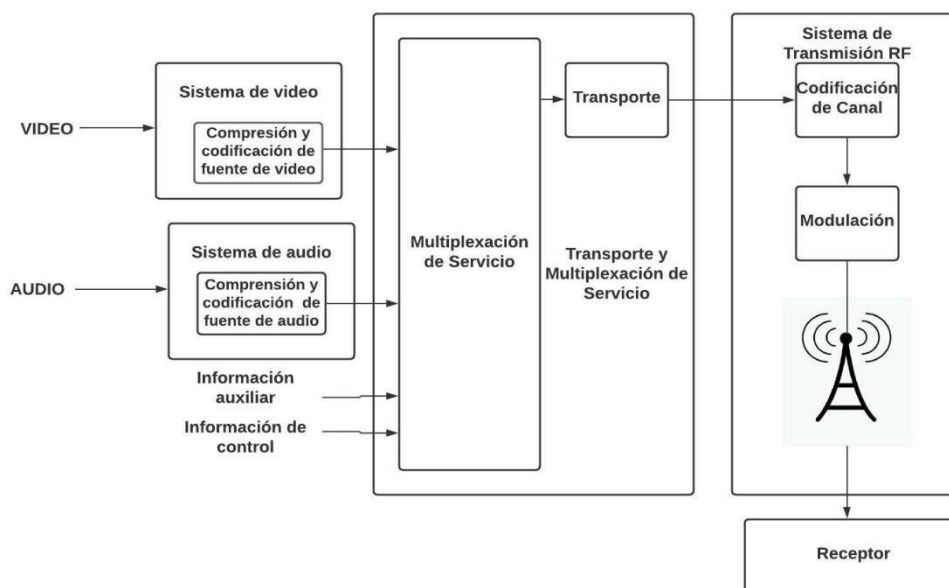
Nota. Se muestra la distribución de los estándares de TDT representados por colores adoptado por cada país.

Estándar ATSC

En 1982 es creado Advanced Television Systems Committee (ATSC) por un consorcio de empresas que llevan el nombre de Gran Alliance, para definir el estándar de TDT que adoptaría EEUU para su sistema de transmisión terrestre y distribución por cable para recepción de programas HDTV en terminales fijos. El estándar ATSC consta de un sistema de transmisión de audio, video y datos auxiliares de alta calidad con una velocidad de transmisión de datos de 19 Mbps empleando como medio de transmisión un canal cuyo ancho de banda es de 6 MHz. El sistema ATSC está estructurado con los subsistemas de codificación y compresión de fuente (audio, video y datos), multiplexación y transporte de servicios, transmisión de RF. La figura 2 presenta el diagrama de bloques de manera general del sistema ATSC.

Figura 2

Diagrama de bloques general del sistema ATSC.



Nota. Esta figura muestra la distribución por bloques que conforman el sistema ATSC. Tomada de “Digital Terrestrial Television Broadcasting”: Technology and System (p. 165) (Song et al. 2015).

Para el subsistema de Codificación y Comprensión de datos y de transporte de servicios, se utiliza el formato del estándar Moving Picture Experts Group 2 (MPEG-2), el cual es compatible con diferentes formatos de transporte de otros medios digitales (Song et al. 2015).

En el año 2013 se anuncia ATSC 2.0 y se oficializa en el año 2015 por medio del documento A/107, en el cual se especifica los servicios de TDT compatibles con su versión anterior y los cuales abogan por la televisión interactiva e híbrida, al tener la capacidad de conectarse a una red internet con lo cual permite transmitir contenido multimedia por broadcast o utilizando el protocolo de internet, adquirir videos por solicitud y cuenta con nuevos códigos avanzados de compresión, para video H.264 y para audio (AAC).

ATSC 2.0 no pertenece a la segunda generación, ya que los desarrollos de sus servicios se encuentran dirigidos a la televisión interactiva e híbrida, y no a la mejora en la capa física, ni al aumento del flujo de información, es así que en 2016 Michael Dolan director del grupo de tecnología TG1 en ATSC expresa “la conclusión y publicación exitosa de todas las diversas partes de ATSC 2.0, muchas de las cuales proporcionan una base sólida para el estándar ATSC 3.0 de próxima generación” (Dolan, 2016) .

Estándar de Próxima Generación ATSC 3.0

ATSC forma el Grupo de tecnología 3 (TG-3) en el año 2011, con el propósito de desarrollar un sistema de transmisión de próxima generación. Emitiendo una convocatoria a nivel internacional a diferentes organizaciones para recopilar información de los

requisitos que debe incluir el sistema. Con esto se establece las capacidades del sistema en un marco general, que sirve de guía en la elaboración del conjunto de documentos técnicos que conforman el estándar ATSC 3.0. El cual se basa en una arquitectura definida en 3 capas: Física, Administración y Protocolos, más la de Aplicación y Presentación, que facilita la flexibilidad y extensibilidad en la implementación del sistema (ATSC Standard: A/300:2020, “ATSC 3.0 System”, 2020). La figura 3 muestra un esquema general de la arquitectura en capas del sistema ATSC 3.0.

Figura 3

Arquitectura en capas del sistema ATSC 3.0.



Nota. La figura representa de manera general la arquitectura basada en capas del sistema ATSC 3.0. Tomado de Doc. A/300:2020 por Advanced Transmission Systems Communications (ATSC) (p. 11), 2020, Washington, D. C. 20006 202-872-9160.

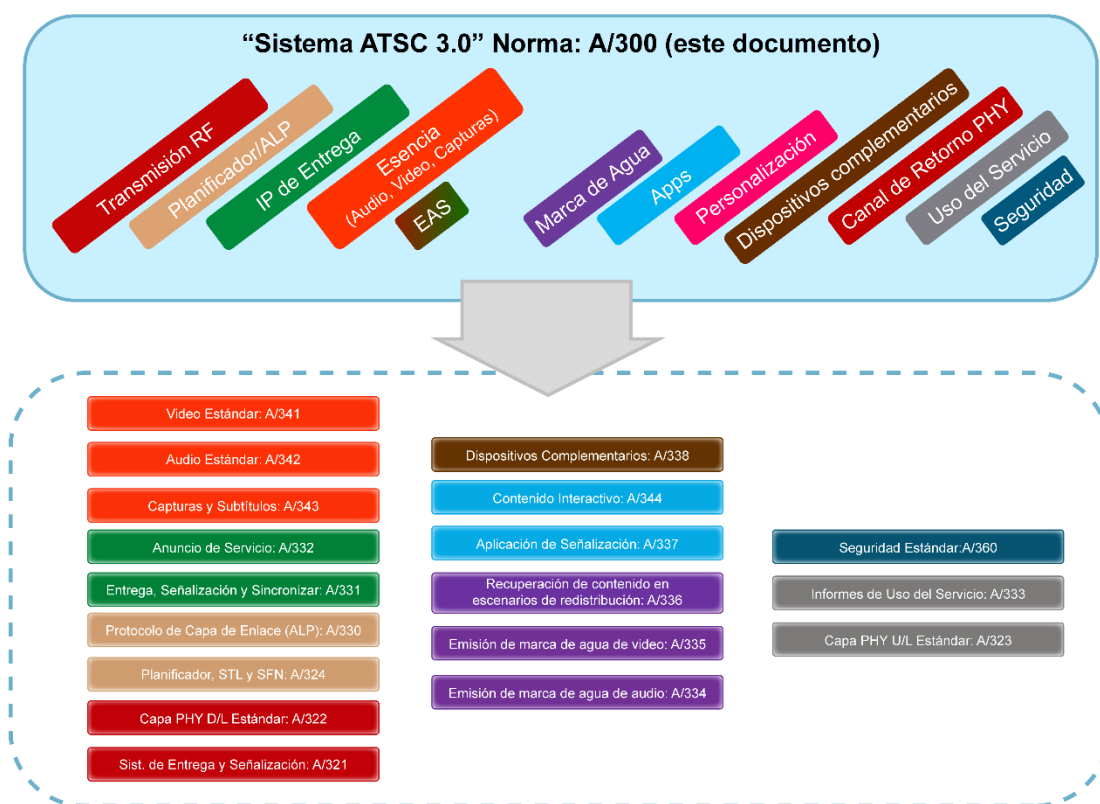
Estructura General

ATSC 3.0 provee servicios de TV con calidad 4K de ultra alta definición (UHD), TV móvil de alta definición (HD) y varios servicios de datos simultáneos, los cuales se encuentran establecidos en las capas superiores del protocolo IP que genera de manera eficiente las redes de radiodifusión y banda ancha (Lee et al. 2018). Esto es posible gracias a la implementación de tecnologías mejoradas como la multiplexación por división de frecuencia ortogonal (OFDM), el código de verificación de paridad de baja densidad

(LPDC), la constelación no uniforme (NUC), la tecnología Layered Division Multiplexing (LDM) y la utilización de los protocolos de transporte MMTP (MPEG Media TransportC Protocol) y ROUTE (Real-Time Object Delivery over Unidireccional Transport) (Jung et al. 2019). Para lo cual el estándar ATSC 3.0 describe cada una de estas tecnologías en un conjunto de documentos de forma detallada, La estructura de estos documentos se la puedes observar en la figura 4.

Figura 4

Estructura y conjunto de documentos de ATSC 3.0.



Nota. La figura muestra la forma en que se encuentra estructurada los documentos que conforman el estándar ATSC 3.0. Tomado de Doc. A/300:2020 *por Advanced Transmission Systems Communications (ATSC)* (p. 13), 2020, Washington, D. C. 20006 202-872-9160.

El objetivo de realizar dicha división es facilitar la evolución independiente de cada aspecto del estándar sin afectar la estructura general (ATSC Standard: A/300:2020, “ATSC 3.0 System”, 2020). En la tabla 1 se puede encontrar la descripción breve del contenido de cada uno de los documentos pertenecientes al estándar ATSC 3.0.

Tabla 1

Conjunto de normas que conforman el Estándar ATSC 3.0.

Título	Estándar	Descripción
Seguridad	A/360	Se especifica las funciones de protección y seguridad definidas para el sistema ATSC 3.0.
Contenido Interactivo	A/344	Se especifica el entorno para contenido interactivo para el sistema ATSC 3.0
Leyendas & Subtítulos	A/343	Se especifica el transporte de leyendas para audios débiles y subtítulos en los transportes ROUTE-DASH y MMT de ATSC 3.0.
Audio	A/342	Se define un marco común para los sistemas de audio en las transmisiones ATSC 3.0 y restricciones asociadas a las tecnologías de codificación de audio.
Video	A/341	Se establece que el sistema ATSC 3.0 soporta múltiples tecnologías de codificación de video. Los servicios de televisión que incorporen ATSC 3.0 deben utilizar un sistema de Alto Rango Dinámico (HDR).
Dispositivos Complementarios	A/338	Se define el protocolo de comunicación entre un receptor primario ATSC 3.0 y un dispositivo complementario. Para presentar contenido complementario relacionado al principal.
Aplicación Evento de Entrega	A/337	Se especifica la entrega de la aplicación de los eventos ejecutados en ATSC 3.0.
Recuperación de Contenido en Escenarios de Redistribución	A/336	Se especifica los formatos de carga útil para las marcas de agua de audio y video, el método de reconocimiento automático de contenido de huellas dactilares y métodos para solicitar y recuperar la señalización de difusión de contenido ATSC 3.0 por medio de banda ancha.
Emisión de Marca de Agua de Video	A/335	Se especifica la tecnología de marca de agua de video utilizada para la recuperación de contenido transmitido por ATSC 3.0.
Emisión de Marca de Agua de Audio	A/334	Se define la utilización de la tecnología VP1 de marca de agua de audio para la recuperación de contenido transmitido por ATSC 3.0.

Título	Estándar	Descripción
Informe de Uso del Servicio	A/333	Se especifica los métodos para informar el uso de los servicios ATSC 3.0.
Anuncio de servicio	A/332	Se especifica el método para el anuncio de servicios en transmisiones ATSC 3.0.
Entrega, Señalización, Sincronización y Protección de Errores	A/331	Se especifica los diferentes mecanismos técnicos y procesos de señalización de los servicios, la entrega de servicios y contenido basada en IP de ATSC 3.0 a través de redes de banda ancha e híbrida (transmisión/banda ancha), idiomas, audio, textos para sordos, subtítulos y servicios de emergencia.
Protocolo Capa de Enlace	A/330	Se define el protocolo "Link-Layer Protocol" (ALP) para ATSC 3.0. El cual corresponde a la capa de enlace de datos del modelo OSI de 7 capas. Con lo cual se logra la encapsulación eficiente de IP, la señalización de la capa de enlace y de los paquetes de flujo de transporte (TS) MPEG-2.
Programador y enlace Estudio al Transmisor (STL)	A/324	Se define la interfaz entre la capa de transporte y la capa física para ATSC 3.0, el cual consiste de protocolos estándar para el transporte de paquetes del protocolo de enlace ATSC 3.0 y paquetes de enlace de estudio a transmisor (STL)
Canal Dedicado de Retorno	A/323	Se define el sistema de canal de retorno dedicado (DRC) para ATSC 3.0, que incluye la capa física y especificaciones de la capa de control de acceso al medio (MAC)
Protocolo de Capa Física	A/322	Se define el protocolo que describe el sistema de transmisión de RF de Descarga (es decir, del transmisor de difusión al receptor del consumidor) de la forma de onda, la modulación y la codificación de la capa física ATSC 3.0
Captura y Señalización del Sistema	A/321	Se especifica el proceso de captura del sistema y la arquitectura de señalización para la capa física ATSC 3.0. "Bootstrap" de ATSC 3.0 es el medio utilizado para transportar dicha información además que otorga un punto de entrada universal a la forma de onda de difusión de ATSC 3.0.

Nota. Esta tabla presenta una breve descripción de cada uno de las normas que conforman el estándar ATSC 3.0.

Descripción General del Sistema ATSC 3.0

“El trabajo en ATSC 3.0 se ha dividido en tres capas funcionales: capa física, capa de transporte y la capa de aplicación y presentación” como lo menciona Rich Chernock presidente del Grupo de Tecnología ATSC sobre ATSC 3.0 (TG3), en su publicación ATSC 3.0 Next Generation Digital TV Standard—An Overview and Preview of the Issue (Chernock et al. 2016). En el siguiente capítulo se describe la capa de transporte del estándar ATSC 3.0.

Capítulo III

Capa de Transporte

Descripción General de la Capa de Transporte para ATSC 3.0

Según la norma A/324 la capa de transporte se define como:

“La capa de transporte es un protocolo funcional el cual define el formato de los datos que se entregarán a los receptores después de su recuperación del formato requerido para su entrega física por la Capa Física. Proporciona comunicación lógica entre procesos de aplicación que se ejecutan en diferentes hosts dentro de una arquitectura en capas de protocolos y otros componentes de red.” (ATSC Standard: Scheduler / Studio to Transmitter Link, 2021, pág. 13)

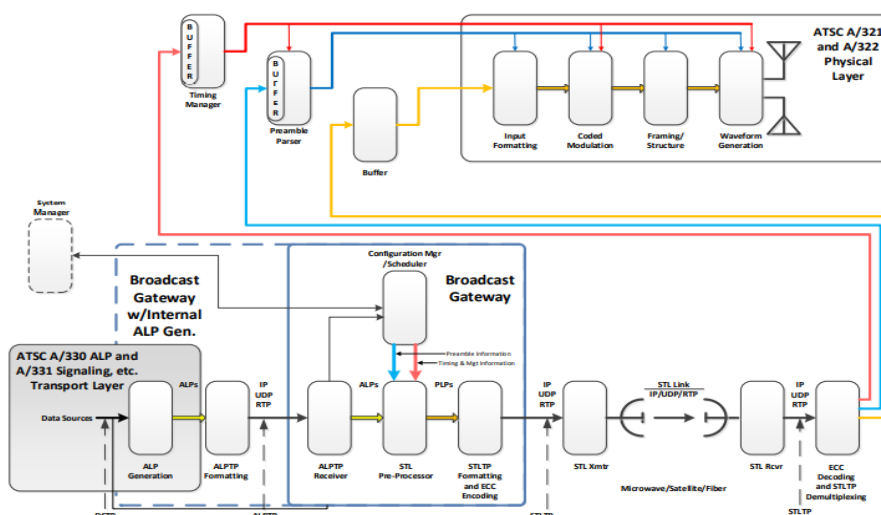
En el estándar ATSC 3.0 la capa de transporte emplea la encapsulación del protocolo IP para la transmisión y la entrega de archivos, sustituyendo la encapsulación del sistema MPEG-2 TS adoptado por el estándar ATSC 1.0. Con esto el estándar ATSC 3.0 permite que la radiodifusión se convierta en parte del Internet. Creando así nuevos servicios y modelos de negocio para los organismos de radiodifusión. Con el transporte IP se abre la puerta a incorporar servicios híbridos, donde los componentes de estos pueden entregarse por radiodifusión y ancho de banda logrando que se puedan sincronizar y combinar según sea necesario para crear nuevos servicios. Al utilizar el transporte IP en la entrega de contenido de transmisión como fragmentos de archivos de medios (parecido a la entrega de transmisión a través de Internet), en lugar de flujos continuos de bits, habilita la capacidad de hacer la inserción de publicidad (contenido) en el receptor de manera personalizada, simplemente entregando segmentos de archivo junto con una lista de reproducción asociada (Chernock et al. 2016).

Para el sistema ATSC 3.0 la capa de transporte se define en los estándares: Protocolo de Capa de Enlace “Link Layer Protocol (ALP)” (A/330), en el estándar de Entrega, Señalización, Sincronización y Protección de Errores “Signaling, Delivery, Synchronization, and Error Protection” (A/331) y en parte del estándar de Enlace de programador / estudio a transmisor “Scheduler / Studio to Transmitter Link”(A/324), en dichos documentos se puede encontrar el procedimiento para encapsulación de diferentes formatos de datos en paquetes ALP (paquetes de capa de enlace), dos métodos de entrega y presentación de servicios de difusión (MMT y ROUTE-DASH) utilizados para el sistema ATSC 3.0 y uso de los protocolos de transporte ALPTP (Atsc 3.0 Link Layer Protocol Transport Protocol), DSTP (Data Source Transport Protocol) y STLP (Studio to Transmitter Link Transport Protocol).

La figura 5 presenta el esquema de forma general de la interconexión entre la capa física y la capa de transporte del sistema ATSC 3.0 y los nombres de los estándares que definen la capa de transporte, como se muestra en la parte inferior izquierda de la figura 5.

Figura 5

Esquema de la interconexión entre la capa física y la capa de transporte.



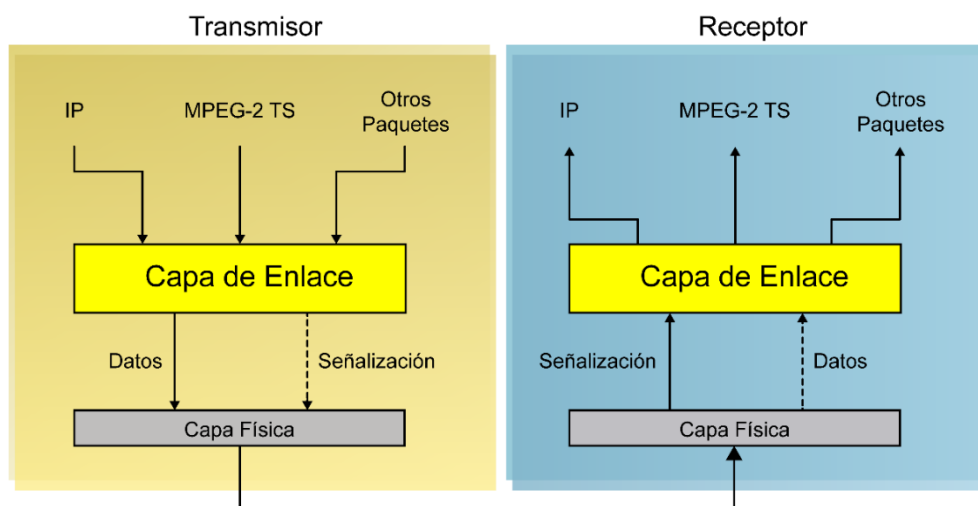
Nota. Tomado de Doc. A/324:2021 por *Advanced Transmission Systems Communications (ATSC)* (p. 18), 2021, Washington, D. C. 20006 202-872-9160.

Capa de Enlace (A/330 - Link Layer)

La capa de enlace “Link Layer” nos da la posibilidad de realizar el envío y la recepción de información entre la capa física y la de red. Durante el proceso de transmisión la capa de enlace transporta los datos desde la capa de red hacia la capa física, por otro lado, en el proceso de recepción la capa de enlace transporta los datos desde la capa física hacia la capa de red. El objetivo de la capa de enlace es abstraer los diferentes tipos de paquetes de datos (IP, MPEG-2 TS, Otro tipo de paquetes) que ingresen al sistema y transformarlos a un solo formato, el cual pueda ser procesado por la capa física. La figura 6 se puede observar el diagrama lógico de la capa de enlace de ATSC 3.0 (ATSC Standard:Link-Layer Protocol (A/330), 2019).

Figura 6

Diagrama lógico del transporte de datos de la capa de enlace de ATSC 3.0.



Nota. El diagrama muestra dos flujos lógicos (**Datos**, **Señalización**), para la implementación física se utiliza una sola conexión, **Otros paquetes** hace referencia a

paquetes de datos que pueden ser implementados en el futuro, con esto se asegura la extensibilidad del sistema. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 5), 2019, Washington, D. C. 20006 202-872-9160.

El proceso que se ejecuta en el interior de la capa de enlace se denomina Protocolo de capa de enlace (ALP) y los paquetes generados por este protocolo se denominan paquetes ALP, con este protocolo se asegura que los datos que ingresan se puedan transmitir de manera eficiente entre la capa de red y la capa física. Las operaciones que se incluyen en este protocolo son encapsulación, comprensión y señalización de la capa de enlace (ATSC Standard:Link-Layer Protocol (A/330), 2019).

Servicios

Los servicios que provee el protocolo ALP se describen brevemente a continuación:

Encapsulación de paquetes

ALP tiene la capacidad de encapsular cualquier tipo de paquete de datos, entre estos incluye los paquetes IP ("Paquete IP" se usa como referencia a un paquete IPv4) y MPEG-2 TS (los paquetes MPEG-2 TS son considerados un tipo especial de capa de red). Cada paquete de capa de red o cualquier otro tipo de paquete de datos que ingrese al sistema, es transformado a un paquete genérico ALP, cuyo formato es el único aceptado por la capa física. Al trabajar con paquetes muy grandes o pequeños, se utiliza la segmentación y concatenación con el propósito de utilizar de manera eficiente los recursos de capa física.

Segmentación y reensamblaje

La segmentación consiste en dividir en dos o más partes el paquete de capa de red, cada segmento se encapsula en un paquete ALP y se transmite respetando el orden de la posición original, los encabezados de los paquetes de la capa de red incluyen secciones de protocolo para realizar la segmentación en el transmisor y reensamblaje en el receptor.

Concatenación

La concatenación permite combinar una serie de paquetes de capa de red pequeños en el payload de un paquete ALP, el cual incluye secciones de protocolos para la concatenación.

Reducción de la sobrecarga

ALP reduce de manera significativa la sobrecarga para el transporte de datos en la capa física, ya sea en paquetes IP o paquetes MPEG-2 TS, como se menciona a continuación.

- **Reducción de la sobrecarga IP**

Los paquetes IP poseen encabezados de tamaño fijo, sin embargo, la información que participa en una comunicación puede ser redundante, con ALP se reduce la sobrecarga de transmisión al comprimir los encabezados de los paquetes IP, incluyendo encabezados de paquetes UDP.

- **Reducción de la sobrecarga MPEG-2 TS**

ALP reduce la sobrecarga para transportar de manera eficiente paquetes MPEG-2 TS. Esto se logra al eliminar el byte de sincronización lo cual reduce la sobrecarga en un byte por paquete MPEG-2 TS, al eliminar paquetes MPEG-2 TS nulos de 188 bytes

de manera que puedan reinsertarse en el receptor y finalmente la eliminación de encabezados común.

Transmisión de señalización

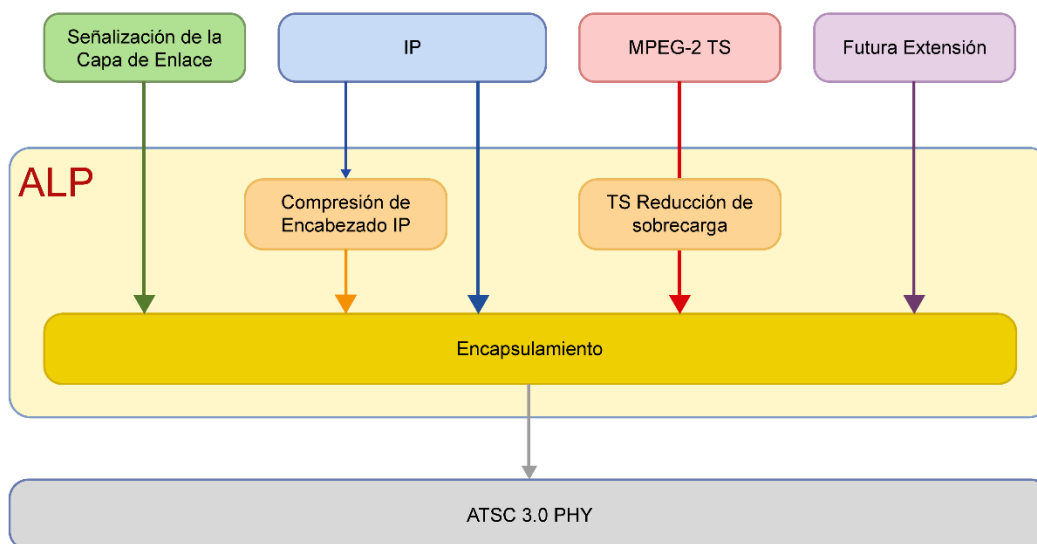
ALP proporciona un formato específico para paquetes de señalización con lo que se permite el transporte de señalización de la capa de enlace.

Arquitectura del Sistema

ALP acepta paquetes de capa de red IPV4 o MPEG-2 TS como paquetes de entrada, en el futuro se puede implementar otro tipo de extensiones de paquetes de datos y protocolos de entrada. ALP especifica el formato y la señalización de la capa de enlace, incluyendo información sobre el mapeo de flujos de paquetes IP específicos a conductos de datos en la capa física. Además, incorpora mecanismos para mejorar la eficiencia de la transmisión, implementado algoritmos de eliminación y compresiones de encabezado como se muestra en la figura 7 (ATSC Standard:Link-Layer Protocol (A/330), 2019).

Figura 7

Diagrama de bloques y de interfaz de la arquitectura de ALP.



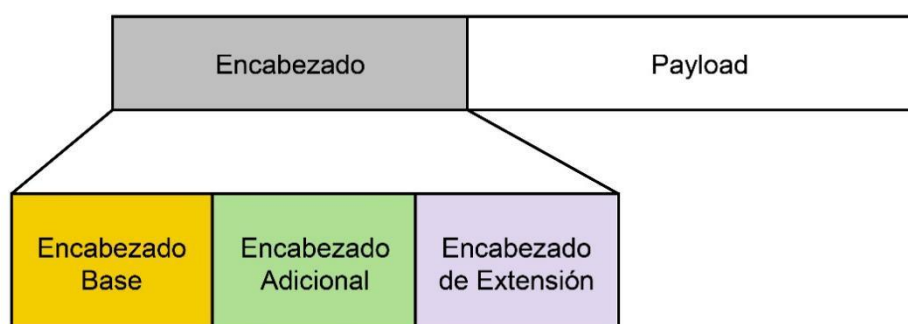
Nota. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 7), 2019, Washington, D. C. 20006 202-872-9160.

Formato del paquete ALP

Un paquete ALP se encuentra formado por el encabezado, el cual está constituido por un encabezado base, un encabezado adicional dependiendo de los campos de control del encabezado base y un encabezado de extensión opcional. Seguido de estos campos se encuentra la carga Payload de datos, como se puede observar en la figura 8.

Figura 8

Formato del paquete ALP.



Nota. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 7), 2019, Washington, D. C. 20006 202-872-9160.

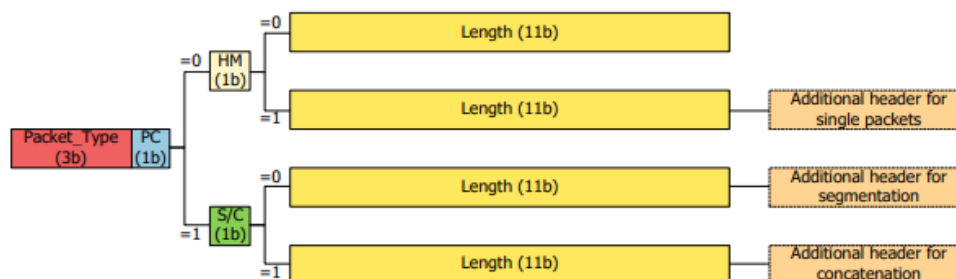
Encapsulación del paquete ALP

Encabezado base

El encabezado base para la encapsulación de paquetes ALP tiene una estructura jerárquica como se muestra en la figura 9.

Figura 9

Estructura del encabezado base para la encapsulación de paquete ALP.



Nota. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 8), 2019, Washington, D. C. 20006 202-872-9160.

Los campos que pertenecen a esta estructura presentada en la figura 9 son los siguientes:

Packet_type: Posee una longitud de 3 bits, los cuales indican el tipo de paquete de datos de entrada antes de la encapsulación en el paquete ALP, como se muestra en la tabla 2.

Tabla 2

Valores y significado de packet_type.

Valor de packet_type	Descripción
000	Paquete IPV4
001	Reservado
010	Paquete IP comprimido
011	Reservado
100	Paquete de señalización de capa de enlace
101	Reservado
110	Extensión de tipo de paquete
111	MPEG-2 TS

Nota. La Tabla muestra la descripción para cada valor de Packet_Type en formato binario. Tomado de Doc. A/330:2019 *por Advanced Transmission Systems Communications (ATSC)* (p. 9), 2019, Washington, D. C. 20006 202-872-9160.

Payload_configuration (PC): Posee una longitud de 1 bit, que indica la configuración de payload. Donde sí PC = "0" indica que en el paquete ALP se encuentra un paquete de entrada completo y el próximo campo es header_mode (HM). Si PC = "1" indica que en el paquete ALP se encuentra más de un paquete de entrada (concatenación) o una parte de un paquete de entrada grande (segmentación) y el próximo campo es segmentation concatenation (S/C).

Header_mode (HM): Posee una longitud de 1 bit, el cual indica si existe o no un encabezado adicional, donde si HM = "0" indica que no existe encabezado adicional para el paquete de entrada y la longitud de payload del paquete ALP es inferior a 2048 bytes. Si HM = "1" indica que existe un encabezado adicional para el paquete de entrada y la longitud de payload es de 2047 bytes,

Segmentation_concatenation (S/C): Posee una longitud de 1 bit, si S/C= "0" indica que el payload transporta un segmento de un paquete de entrada y un encabezado adicional para la segmentación, Si S/C = "1" indica que el payload transporta más de un paquete de entrada y un encabezado adicional para la concatenación. Para ingresar a esta configuración el valor de PC debe ser igual "1".

Length: Posee una longitud de 11 bits, los cuales pertenecen a los 11 bits menos significativos (least significant bits - LSB) de la longitud total en bytes del payload transportado por el paquete ALP. Si existe un campo length_MSB en el encabezado adicional el campo length se concatena con el campo length_MSB y representa la parte

del valor LSB de la longitud payload real. Se puede configurar 4 tipos de paquetes: un paquete sin encabezado adicional, un paquete con encabezado adicional, un paquete segmentado y un paquete que sea resultado de la concatenación de varios paquetes. La sintaxis de dicha configuración de paquetes se la puede observar en la tabla 3.

Tabla 3

Payload_Configuration (PC), Valores de campo y longitud total de encabezado base

Valor PC	Descripción	Siguiente Campo	Valor	Tamaño de encabezado adicional	Campo encabezado adicional	Longitud total de encabezado (sin encabezado total opcional)
0	Paquete individual	H	0	-	-	2 bytes
		M				
		H	1	1 byte	Length_MSB	3 bytes
1	Segmentación	M				
		S/C	0	1 byte	Seg_SN, LSI	3 bytes
	Concatenación	S/C	1	$1 + [count + 1] * 1.5$ bytes	Length_MSB, count, Component_length	$3 + [count + 1] * 1.5$ bytes

Nota. La tabla muestra la sintaxis de configuración para el encabezado base, el tamaño original de un paquete individual es de 2 bytes. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 10), 2019, Washington, D. C. 20006 202-872-9160.

Encabezado adicional

Existen 3 tipos de encabezados adicionales que se describen a continuación:

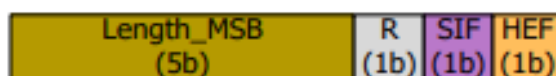
- **Encabezado adicional para paquetes individuales**

Si header_mode (HM) = "1", se incorpora un encabezado adicional para paquetes individuales. HM será igual a "1" cuando la longitud del payload del paquete ALP sea mayor a 2047 bytes. Los campos que estructuran el encabezado adicional para paquetes

individuales se presentan en la figura 10 y cada campo que se presenta en dicha figura se detalla a continuación.

Figura 10

Estructura del encabezado adicional para paquetes individuales.



Nota. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 10), 2019, Washington, D. C. 20006 202-872-9160.

Length_MSB: Son los 5 bits más significativos (Most Significant Bits “MSBs”) de la longitud total en bytes del payload del paquete ALP actual, estos se concatenan con el campo Length, el cual contiene los 11 bits menos significativos (LSB) para obtener la longitud total de payload, cuya máxima longitud que se puede señalar es de 65,535 bytes. Si el payload del paquete ALP es menor que 2047 bytes y se requiere añadir el encabezado opcional al paquete ALP, este se establecerá en ceros.

SIF (Sub-Stream Identifier Flag): Este bit permite indicar si el encabezado opcional que identifica el subflujo está presente después del campo HEF. Si SIF = “0”, significa que no existe sub_stream_identification () en el paquete ALP. Por otro lado, si SIF= “1” se hará presente sub_stream_identification () después del encabezado adicional en el paquete ALP.

HEF (Header Extension Flag): Cuando HEF = “1”, indica que el encabezado opcional header_extension () se encuentra presente después del encabezado adicional para futuras extensiones del encabezado ALP. Si HEF = “0” header_extensión () no está presente.

- **Encabezado adicional para paquetes segmentados**

Si S/C = "0", se implementará el encabezado adicional para la segmentación para paquetes individuales, los campos que estructuran este encabezado se presentan en la figura 11 y cada campo que se presenta en dicha figura se detalla a continuación.

Figura 11

Estructura del encabezado adicional para segmentación paquetes individuales.



Nota. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 11), 2019, Washington, D. C. 20006 202-872-9160.

Segment_sequence_number (Seg_SN): El valor de estos 5 bits indicarán el orden de cada segmento que sea transportado por el paquete ALP, es así que primer segmento de un paquete de entrada transportado por un paquete ALP tendrá un valor de "0x0". Este valor aumentará en una unidad por cada segmento que pertenezca al paquete de entrada segmentado. Cuando SEG_SN = "0x0", last_segment_indicator no puede ser igual a "1".

Last_segment_indicator (LSI): Cuando LSI = "1", indica que el segmento en el payload es el último del paquete de entrada. Si LSI = "0" indica que aún no se ha llegado al último segmento del paquete de entrada.

SIF (Sub_stream Identifier Flag): Este bit permite indicar si el encabezado opcional que identifica el subflujo está presente después del campo HEF. Si SIF = "0", significa que no existe sub_stream_identification () en el paquete ALP. Por otro lado, si

SIF= “1” se hará presente sub_stream_identification () después del encabezado adicional en el paquete ALP.

HEF (Header Extension Flag): Cuando HEF = “1”, indica que el encabezado opcional header_extension () se encuentra presente después del encabezado adicional para futuras extensiones del encabezado ALP. Si HEF = “0” header_extensión () no está presente.

- **Encabezado adicional para paquetes concatenados**

Si S/C = “1”, se implementará el encabezado adicional para concatenación para paquetes individuales, los campos que estructuran este encabezado se presentan en la figura 12 y cada campo que se presenta en dicha figura se detalla a continuación.

Figura 12

Estructura del encabezado adicional para concatenación de paquetes individuales.



Nota. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 12), 2019, Washington, D. C. 20006 202-872-9160.

Length_MSB: Son los 4 bits más significativos (Most Significant Bits “MSBs”) de la longitud total en bytes del payload del paquete ALP. La longitud máxima del payload para concatenaciones de 32.767 bytes.

Count: Este campo indica el número de paquetes de entrada incluido en paquete ALP. El mínimo número de paquetes concatenados es 2 y el máximo es de 9.

SIF (Sub_stream Identifier Flag): Este bit permite indicar si el encabezado opcional que identifica el subflujo está presente después del campo HEF. Si SIF = "0", significa que no existe sub_stream_identification () en el paquete ALP. Por otro lado, si SIF= "1" se hará presente sub_stream_identification () después del encabezado adicional en el paquete ALP

Component_length: Estos 12 bits indican la longitud de cada paquete en bytes. Los campos Component_length se incluyen en el mismo orden que los paquetes presentes en el payload y el número de campos se indicará mediante count+1. Cuando un encabezado ALP contiene un número impar de campos Component_length, le seguirán 4 bits después del último campo Component_length, los cuales servirán de relleno y se establecerán en 0.

Encabezado de extensión

Se tiene la posibilidad de extender el paquete ALP, al añadir un encabezado de extensión opcional de acuerdo a la señalización de cada una de las banderas de estructura de encabezado adicional (SIF o HEF). Si se establece bandera SIF, el encabezado de extensión tendrá una estructura sub_stream_identification (). Por otro lado, si la bandera HEF se encuentra activada, el encabezado de extensión se implementará y debe contener un encabezado opcional header_extension (). En el caso de tener activadas tanto SIF como HEF, la estructura tendrá el orden sub_stream_identification () y a continuación la estructura header_extension ().

Sub_Stream Identification

El encabezado opcional para sub-stream identification sirve como filtro de flujo de paquetes específicos a nivel de capa de enlace. Un ejemplo sub-stream puede ser el rol del identificador de servicio en un flujo ALP, en el cual se transporta múltiples servicios.

La información de mapeo entre un flujo de capa superior y el valor SID (“Sub-stream identifier”, 8 bits permiten indicar el sub-stream identifier para el paquete ALP) correspondiente al flujo de capa superior se proporciona como una tabla de mapeo de enlace.

Header Extension

Este campo hace referencia a los campos extendidos que pueden ser utilizados en el futuro. El receptor al no entender cualquier campo header extensions, este lo ignorará. Los campos definidos para el header extension se lo presenta en la figura 13 y se los detalla a continuación.

Figura 13

Campos definidos para estructura del encabezado header extension.

Syntax	No. of bits	Format
header_extension() {		
extension_type	8	uimsbf
extension_length_minus1	8	uimsbf
for (i=0; i<=extension_length_minus1; i++) {		
extension_byte	8	uimsbf
}		
}		

Nota. La figura muestra los campos que conforman la estructura del header extension encabezado opcional para paquetes individuales. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 14), 2019, Washington, D. C. 20006 202-872-9160.

Extension_type: Estos 8 bits describen el tipo de header_extension () que se tiene, como se muestra en la tabla 4. Si el valor de extension_type se encuentra el rango

de User Private, los datos de `extension_byte` se pueden utilizar para otras implementaciones.

Tabla 4

Rango de valores y descripción de `extension_type`.

extension_type	Descripción
0x00-0XEF	Reservado
0XF0-0XFF	User Private

Nota. La tabla muestra el rango de valores que puede adoptar `extension_type` y su descripción. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 14), 2019, Washington, D. C. 20006 202-872-9160

Extension_length_minus1: Estos 8 bits indican uno menos que el número de `extension_byte` que sigue.

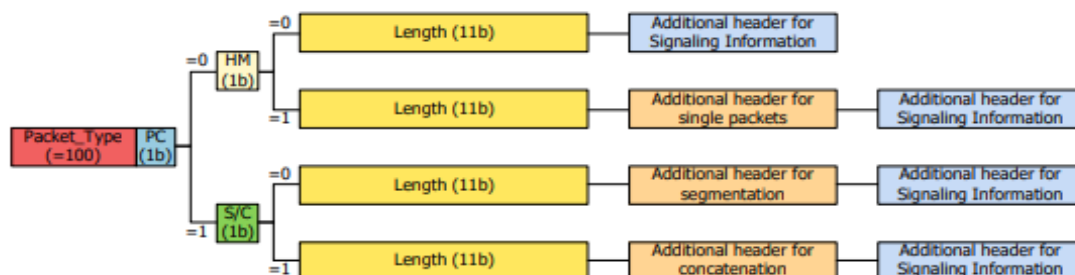
Extension_byte: Este byte representa el valor de `header_extension ()`.

Señalización de encapsulación

La señalización de la capa de enlace se encuentra incorporada en los paquetes ALP. Los paquetes de señalización se identifican por el campo `packet_type` del encabezado base con un valor igual a "100". La estructura de los paquetes ALP conteniendo un encabezado adicional para la señalización de información de muestra en la figura 14.

Figura 14

Estructura del paquete de señalización ALP (encabezado base y encabezado adicional).



Nota. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 15), 2019, Washington, D. C. 20006 202-872-9160.

Para el encapsulado de señalización, el paquete ALP incluye el encabezado adicional llamado *Additional Header for Signaling Information* tiene su propia tabla de señalización (*Link Mapping Table*). La tabla mencionada anteriormente es considerada como un *payload* del paquete ALP que tiene una longitud establecida en los campos *length*, *length_MSB* y *component_length* del encabezamiento del paquete ALP.

Additional Header for Signaling Information

La figura 15 presenta la sintaxis de los campos que estructuran el encabezado adicional llamado *signaling_information_hdr ()*. Cada uno de los campos presentados en dicha imagen se detallan a continuación.

Figura 15

Sintaxis de los campos del encabezado Additional Header for Signaling Information.

Syntax	No. of bits	Format
signaling_information_hdr() {		
signaling_type	8	uimsbf
signaling_type_extension	16	bslbf
signaling_version	8	uimsbf
signaling_format	2	uimsbf
signaling_encoding	2	uimsbf
reserved	4	'1111'
}		

Nota. La figura muestra los campos del encabezado adicional Additional Header for Signaling Information su tamaño en bits y formato. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 15), 2019, Washington, D. C. 20006 202-872-9160.

Signaling type: Estos 8 bits indican el tipo de señalización como se muestra en la tabla 5.

Tabla 5

Valores de código para Signaling_type.

Signaling_type	Descripción
0x00	Reservado
0x01	Tabla de mapeo de enlace (Link Mapping Table)
0x02	Tabla de descripción ROHC-U
0x03-0XEF	Reservado
0XF0-x0FF	User Private

Nota. La tabla muestra el código de valores del campo signaling_type y la descripción correspondiente. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 15), 2019, Washington, D. C. 20006 202-872-9160.

Signaling_type_extension: Consta de 16 bits los cuales permiten indicar el atributo de la señalización. El valor de este campo de definen dentro de cada tabla de señalización

Signaling_version: Consta de 8 bits, que indica la versión de señalización. Este valor incrementa en 1 al presentarse cambios en cualquier dato de la señalización, identificado por signaling_type. Este valor se ajustará a 0 después de alcanzar su valor máximo.

Signaling_format: Consta de 2 bits, que indican el formato que tiene los datos de señalización, como se detalla en la tabla 6.

Tabla 6

Valores de código para Signaling_format.

Signaling_format	Descripción
00	Binario
01	XML
10	JSON
11	Reservado

Nota. La tabla el código de valores del campo signaling_format y la descripción correspondiente. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 7), 2019, Washington, D. C. 20006 202-872-9160.

Signaling_encoding: Estos 2 bits especifican el formato de codificación/compresión. Los valores de código del campo signaling_encoding se describen en la Tabla 7. Cuando el campo signaling_format indica el valor Binario ("00"), el campo signaling_encoding se establecerá en "00".

Tabla 7

Valores de código para *Signaling_encoding*.

<i>Signaling_encoding</i>	Descripción
00	Sin compresión
01	DEFLATE (RFC 1951)
10	Reservado
11	Reservado

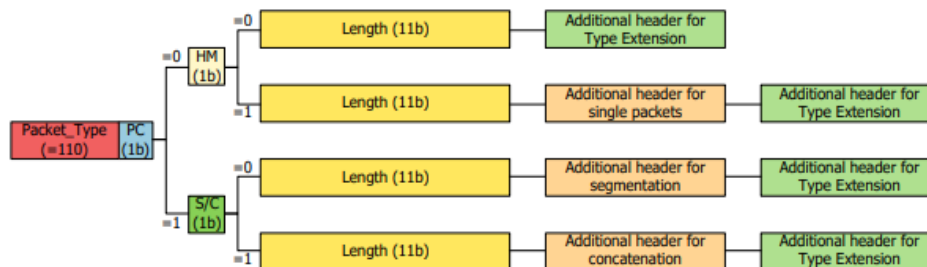
Nota. La tabla muestra el código de valores del campo *signaling_encoding* y la descripción correspondiente. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 7), 2019, Washington, D. C. 20006 202-872-9160.

Extensión Tipo de Paquete

Con el objetivo de proveer un mecanismo que permita incorporar al sistema ATSC 3.0 un número ilimitado de paquetes de entrada, pertenecientes a diferentes protocolos en el futuro, se establece el campo, Encabezado Adicional “*Additional Header*”. La extensión de tipo de paquete debe ser usada solamente cuando el valor del *packet_type* se establece en el valor de ‘110’ como se indica en la tabla 2. En la figura 16 se muestra el caso en el que la estructura del paquete ALP contiene un Encabezado Adicional para el Tipo de Extensión “*Additional Header for type extension*”.

Figura 16

Estructura de los paquetes ALP que contiene un encabezado *Additional header for type extension*.



Nota. La figura muestra la estructura lógica del paquete ALP con un encabezado adicional para la extensión de paquetes de datos pertenecientes a otros protocolos y así incluirlos al sistema ATSC 3.0 en el futuro. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 16), 2019, Washington, D. C. 20006 202-872-9160.

Additional header for Type Extension

El Encabezado Adicional para Extensión de Tipo constara de `type_extension_hdr()`, el cual se encuentra constituido con los campos establecidos en la figura 17.

Figura 17

Sintaxis para campo Additional header for type extension.

Syntax	No. of bits	Format
<code>type_extension_hdr() { extended_type }</code>	16	uimsbf

Nota. La figura muestra los campos del encabezado adicional Additional header for Type Extension su tamaño en bits y formato. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 17), 2019, Washington, D. C. 20006 202-872-9160.

Extended_type: Campo de 16 bits el cual permite identificar el tipo de protocolo al que pertenece el paquete entrada encapsulado en el payload de un paquete ALP. Este campo no se puede utilizar para identificar los protocolos definidos en la tabla 2. Los valores de `extended_type` se encuentran reservados, como se muestra en la Tabla 8.

Tabla 8

Valores de código para *extended_type*.

<i>extended_type</i>	Descripción
0x00-0xFF	Reservado

Nota. La tabla muestra el código de valores del campo *extended_type* y la descripción correspondiente. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 17), 2019, Washington, D. C. 20006 202-872-9160.

Encapsulación de paquetes MPEG-2 TS

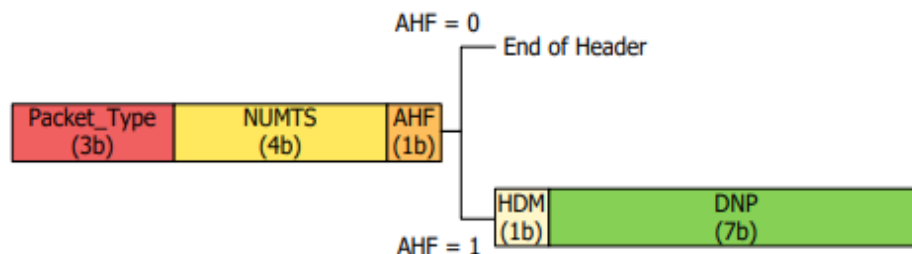
El formato del paquete ALP pueden incluir paquetes MPEG-2 TS al establecer el valor del campo *packet_type* del encabezado base con el valor “111”, como indica en la tabla 2. Varios paquetes TS pueden ser encapsulados dentro de un paquete ALP. Además, ALP permite mejorar la eficiencia de transmisión de MPEG-2 TS, al eliminar el byte de sincronización (0x47) del paquete TS, los paquetes nulos y encabezados TS con formatos similares.

Estructura del paquete ALP al encapsular paquetes TS

La estructura general del paquete ALP al encapsular paquetes con el formato MPEG-2 TS se muestra en la figura 18.

Figura 18

Estructura del encabezado al encapsular paquetes con formato MPEG-2 TS en paquetes ALP.



Nota. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 18), 2019, Washington, D. C. 20006 202-872-9160.

La sintaxis de los campos de la estructura para encapsulación de paquetes MPEG-2 TS se muestra en la figura 19, a continuación, se detalla cada uno de estos campos.

Figura 19

Sintaxis de los campos del encabezado de paquete de capa de enlace ATSC 3.0 para MPEG-2 TS.

Syntax	No. of bits	Format
ATSC3.0_link_layer_packet() {		
packet_type	3	'111'
NUMTS	4	uimsbf
AHF	1	bslbf
if (AHF == 1) {		
HDM	1	bslbf
DNP	7	uimsbf
}		
}		

Nota. La figura muestra la sintaxis de los campos del encabezado de paquete ALP al encapsular paquetes MPEG-2 TS, el tamaño en bits y formato. Tomado de Doc.

A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 18), 2019, Washington, D. C. 20006 202-872-9160.

Packet_type: Campo de 3 bits que define el tipo de paquete de entrada como se muestra en la tabla 2. Packet_type debe tener un valor de “111” para encapsular paquetes MPEG-2 TS.

NUMTS (Number of TS packets): Campo de 4 bits que indica el número de paquetes TS encapsulados en el payload de un paquete ALP. El número máximo de paquetes TS es 16. Si el valor de NUMTS es igual a “0” indicarán que el paquete ALP transporta 16 paquetes TS. Para los otros valores de NUMTS corresponde al valor indicado de número de paquetes (NUMTS = “0001”, significa que se está transportando 1 paquete TS).

AHF (Additional Header Flag): Este bit funciona como bandera que permitirá identificar si se encuentra presente un encabezado adicional. Si AHF = “0” indica que no se encuentra presente ningún encabezado adicional. Si AHF = “1” indica que se encuentra presente un encabezado adicional con un tamaño de 1 byte, que se eliminaron paquetes TS nulos o se aplicó la compresión de encabezado TS.

HDM (Header Deletion Mode): Campo de 1 bit que indica si el proceso de eliminar el encabezamiento TS en el paquete ALP se puede ejecutar. Si HDM = “1” indica que se procede a eliminar del encabezado TS, por otro lado, si HDM = “0” no se ejecutará el proceso de eliminar el encabezamiento TS.

DNP (Deleted Null Packets): Campo de 7 bits que permiten identificar el número de paquetes TS nulos eliminados. La cantidad máxima de paquetes TS nulos que se pueden eliminar es de 128. En el caso que HDM = “0”, el valor de DNP = “0”, lo que indica que se eliminaron 128 paquetes nulos. Si el valor de HDM = “1”, el valor de DNP = “0”,

indicaran que no se eliminaron paquetes nulos, si DNP adquiere otro valor diferente de cero por ejemplo DNP = "5" significa que se eliminaron 5 paquetes nulos.

Eliminar el Byte de Sincronización (Sync) de Paquetes MPEG-2 TS al encapsular en ALP

Para encapsular paquetes TS en el payload de un paquete ALP, se procede a eliminar el byte SYNC (0x47) el cual se encuentra al comienzo de los paquetes TS. Con esto la longitud de un paquete MPEG-2 TS encapsulados en el payload de un paquete ALP será de 187 bytes a diferencia de los 188 bytes que se encuentra en los paquetes MPEG-2 TS originales.

ALP tiene la capacidad de transportar paquetes MPEG-2 TS sin el byte de sincronización (0x47). Se presenta el caso de encapsular 8 paquetes MPEG-2 TS en un paquete ALP, esto permitirá identificar los pasos que sigue el procedimiento de encapsular de paquetes TS al eliminar el byte de sincronización (Sync), los pasos son los siguientes:

1. Eliminar el byte de sincronización para MPEG-2 TS (0x47), con esto la longitud del paquete MPEG-2 TS se reduce a 187 bytes de los 188 bytes que son originalmente.
2. Agrupar en 8 paquetes MPEG-2 TS en la carga útil de un paquete ALP (la longitud total de la carga útil es $187 \times 8 = 1496$ bytes)
3. Se genera el encabezado ALP de dos bytes de longitud con los valores de la tabla 9.

Tabla 9

Valores de campo de encabezado ALP al eliminar paquetes TS nulos.

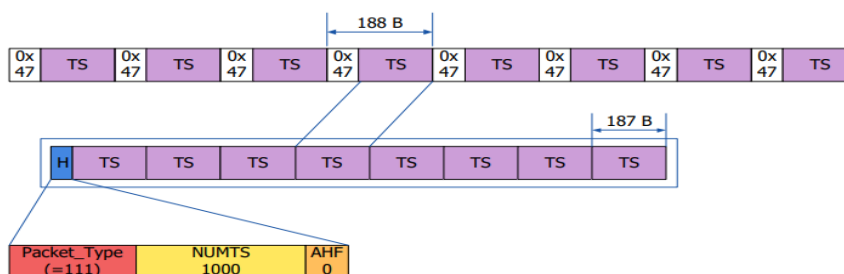
Campo	Valor	Descripción
packet_type	111	Paquete MPEG-2 TS
NUMTS	1000	8 paquetes MPEG-2 TS
AHF	0	No existen paquetes nulos eliminados antes del primer paquete MPEG-2 TS encapsulado.

Nota. La tabla muestra los valores de campo del encabezado ALP al encapsular paquetes TS en un paquete ALP elimina.

Con esto se logra ahorrar 7 bytes en el proceso de encapsulación de paquetes MPEG-2 TS en un paquete ALP, La figura 20 muestra el proceso de encapsulación descrito anteriormente.

Figura 20

Proceso de encapsular 8 paquetes MPEG-2 TS al eliminar el byte de sincronización (Sync) en un paquete ALP.



Nota. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 39), 2019, Washington, D. C. 20006 202-872-9160.

Eliminar paquetes nulos (Null packet) de paquetes MPEG-2 TS al encapsular en paquetes ALP

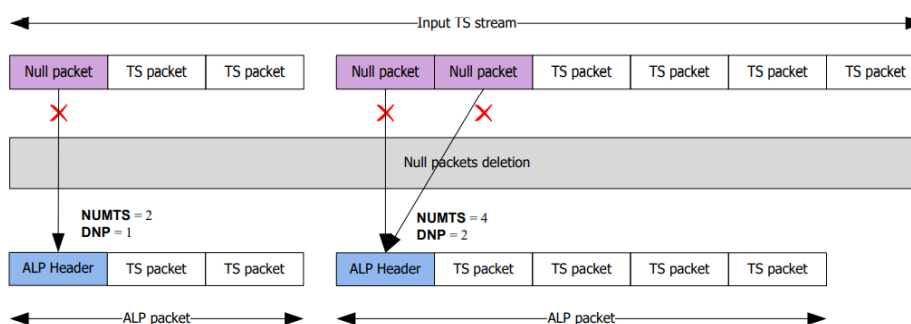
El flujo de transporte requiere que la velocidad de bit permanezca constante a través del tiempo, tanto a la salida del multiplexor en el emisor como a la entrada del demultiplexor en el receptor, además el delay que se presenta de extremo a extremo debe permanecer constante. Para garantizar esto se utiliza paquetes nulos en el Transport Stream (TS), lo que permiten acondicionar servicios de tasa de bits variables en un flujo de tasa de bits constante, esto puede causar una sobrecarga en la transmisión de información, para evitar esto se puede eliminar paquetes TS nulos (paquetes con un PID = 0X1FFF) y volverlos a insertar en el receptor en el lugar exacto que se encontraban originalmente, garantizando así una tasa de bits constante.

Antes de generar un paquete ALP, un contador que permitirá identificar los paquetes nulos eliminados, debe ser reseteado en cero, su valor incrementará por cada paquete nulo eliminado que precede al primer paquete útil que se encapsulará en el payload del paquete ALP actual. El valor de este contador se utilizará para actualizar el valor del campo DNP (Deleted Null Packets). Posterior a esto un grupo de paquetes TS útiles consecutivos se encapsulará en el payload del paquete ALP actual y con esto se puede determinar los valores de cada campo del encabezado ALP.

Al ingresar el paquete ALP generado en la capa física, el contador DNP restablecerá su valor a cero. Si el contador DNP alcanza su valor máximo y el siguiente paquete también es un paquete nulo, este paquete nulo se mantiene como un paquete útil y se encapsula en payload del paquete ALP actual. Cada paquete ALP contendrá al menos un paquete TS útil en su payload, La descripción de este último proceso se lo detalla en la figura 21.

Figura 21

Proceso de encapsular en un paquete ALP paquete MPEG-2 TS eliminando paquetes nulos.



Nota. La figura muestra el proceso de eliminar paquetes MPEG-2 TS nulos al tener presente que el siguiente paquete también es un paquete MPEG-2 TS nulo y su encapsulación en un paquete ALP. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 19), 2019, Washington, D. C. 20006 202-872-9160.

La figura 21 muestra que en el primer paquete ALP se ha eliminado un paquete nulo, antes de encapsular dos paquetes TS útiles, el siguiente paquete es un paquete nulo por lo que el paquete ALP se completa y el contador DNP se restablece a cero. Por lo que los valores del encabezado de este paquete ALP a transmitir son NUMTS = '2' y DNP = '1'.

A continuación, en el siguiente paquete ALP se eliminan 2 paquetes nulos antes de encapsular cuatro paquetes TS útiles, por lo que los valores del encabezado de este paquete ALP a transmitir son NUMTS = '4' y DNP = '2'.

ALP tiene la capacidad de eliminar paquetes MPEG-2 TS nulos antes del primer paquete MPEG-2 TS encapsulado en un paquete ALP e informar al receptor el número de paquetes eliminados a través del encabezado del paquete ALP. Se presenta el caso

de encapsular 6 paquetes MPEG-2 TS y eliminando 2 paquetes nulos (Null Packet) en un paquete ALP, los pasos son los siguientes:

1. Contar y eliminar paquetes nulos (paquetes con un PID = 0X1FFF).
2. Eliminar los bytes de sincronización (0X47) para encapsular los paquetes MPEG-2 TS (con esto se reduce la longitud de cada paquete MPEG-2 TS de 187 bytes a 188 bytes)
3. Agrupe seis paquetes MPEG-2 TS en la carga útil de un paquete ALP (la longitud total de la carga útil es $187 \times 6 = 1122$ bytes)
4. Se genera el encabezado ALP de dos bytes de longitud con los valores de la Tabla 10.

Tabla 10

Valores de campo de encabezado ALP al eliminar paquetes TS nulos.

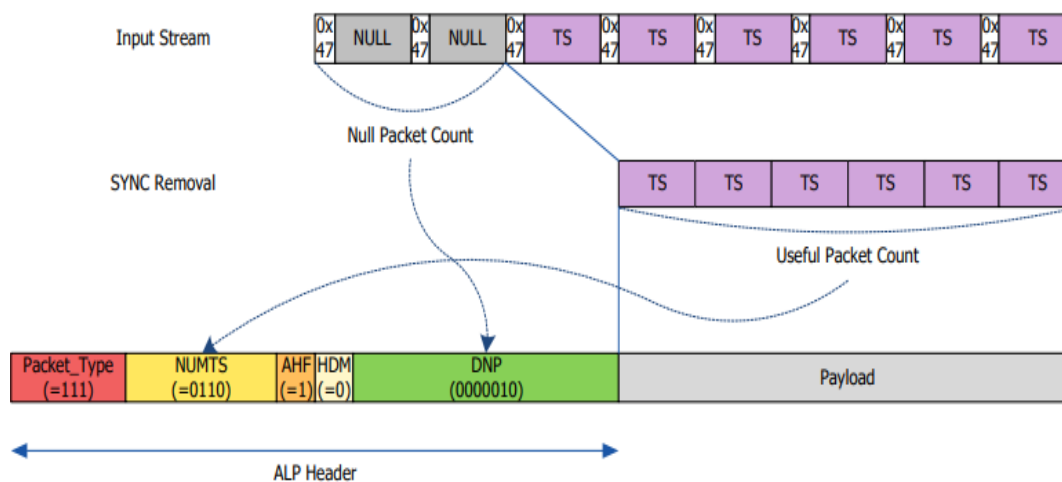
Campo	Valor	Descripción	
packet_type	111	Paquete MPEG-2 TS	<i>Nota.</i>
NUMTS	0110	6 paquetes MPEG-2 TS	<i>La</i>
AHF	1	Existen paquetes nulos eliminados antes del primer paquete MPEG-2 TS encapsulado.	<i>tabla</i>
HDM	0	No se elimina el encabezado TS	
DNP	000001 0	Se elimina 2 paquetes TS nulos	

muestra los valores de campo del encabezado ALP al eliminar paquetes TS nulos.

El paquete ALP resultante tiene una longitud de 1124 bytes y ahorra 380 bytes en comparación con el caso en el que los ocho paquetes MPEG-2 TS se alimentan directamente a la capa PHY. La figura 22 muestra el proceso de encapsulación descrito anteriormente.

Figura 22

Proceso de encapsular 6 paquetes MPEG-2 TS eliminando 2 paquetes nulos en un paquete ALP.



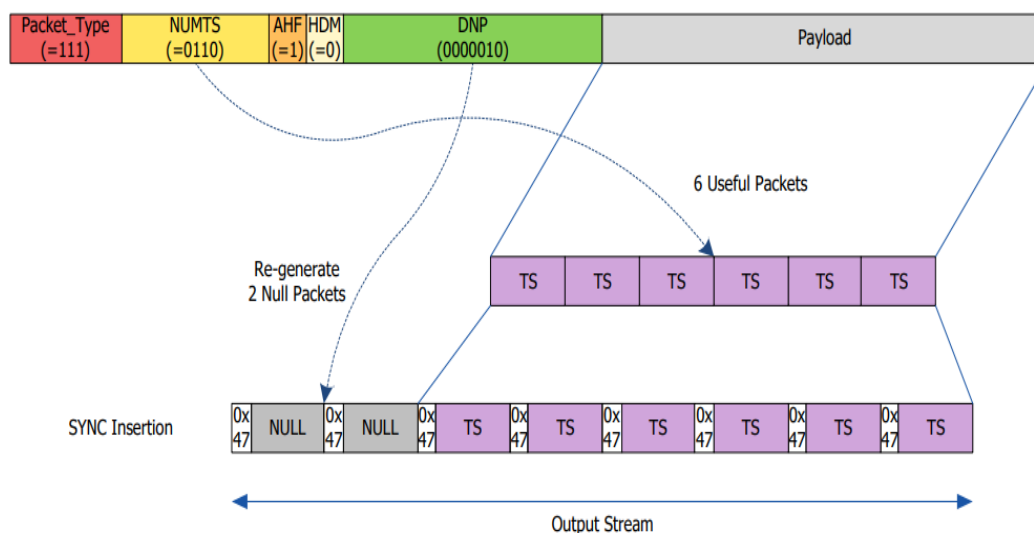
Nota. Tomado de Doc. A/330:2019 por *Advanced Transmission Systems Communications (ATSC)* (p. 39), 2019, Washington, D. C. 20006 202-872-9160.

El proceso de desencapsulación de los paquetes MPEG-2 TS y la inserción de los paquetes MPEG-2 TS nulos se pueden describir en los siguientes pasos:

1. Consultar el valor del campo DNP, para conocer el número de paquetes MPEG-2 TS nulos eliminados.
2. Consultar el valor del campo NUMTS para obtener el número de paquetes MPEG-2 TS encapsulados en paquete ALP.
3. Insertar el byte de sincronización (sync) (0x47).
4. Generar los paquetes TS nulos eliminados de acuerdo al valor que se indique en el campo DNP, los paquetes TS nulos se insertarán antes de los paquetes MPEG-2 TS útiles. La figura 23 muestra el proceso de desencapsulación anteriormente descrito.

Figura 23

Proceso desencapsular 6 paquetes MPEG-2 TS y agregar paquetes nulos desde un paquete ALP.



Nota. La figura muestra el proceso de desencapsular 6 paquetes MPEG-2 TS y la posterior agregación de los 2 paquetes nulos previamente eliminados. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 40), 2019, Washington, D. C. 20006 202-872-9160.

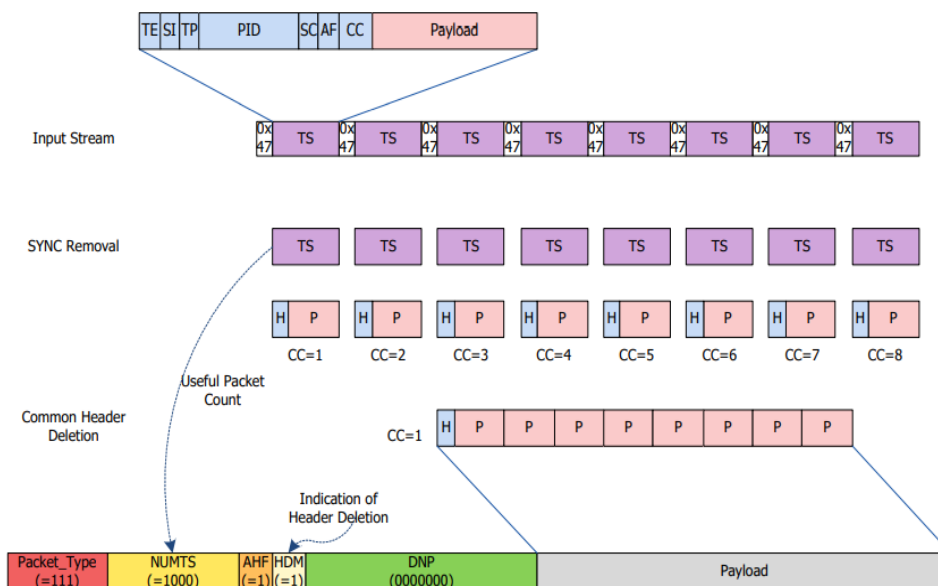
Eliminación del encabezado del paquete MPEG-2 TS al encapsular en un paquete ALP

Al encapsular dos o más paquetes MPEG-2 TS sucesivos, los campos del encabezado se mantienen sin ningún cambio, solo el campo CC (contador de continuidad), aumenta su valor secuencialmente, gracias a esto se puede enviar una sola vez el encabezado del paquete MPEG-2 TS dentro del primer paquete MPEG-2 TS y los demás encabezados se pueden eliminar. El campo *HDM* indica si se elimina o no el encabezado, si el valor de *HDM* = '1' se eliminará los encabezados de los paquetes MPEG-2 TS.

Como ejemplos se puede presentar un paquete ALP en el que se encuentran 8 paquetes con formato MPEG-2 TS, los cuales poseen el mismo encabezado excluyendo el campo CC, como lo muestra la figura 24.

Figura 24

Proceso de encapsular paquetes MPEG-2 TS eliminando su encabezado en un paquete ALP.



Nota. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 41), 2019, Washington, D. C. 20006 202-872-9160.

El proceso presentado en la figura 24 se puede describir con los siguientes pasos:

1. Agrupar 8 paquetes TS con el mismo encabezado, excluyendo el campo CC.
2. El primer encabezado del paquete MPEG-2 TS (excluyendo el byte de sincronización) se mantiene y se eliminan en los otros siete paquetes MPEG-2 TS (la longitud total de la carga útil es $3 + 184 \times 8 = 1475$ bytes).
3. Se genera el encabezado ALP de dos bytes de longitud con los valores de la tabla

Tabla 11

Valores de campo de encabezado ALP al eliminar el encabezado del paquete MPEG-2 TS.

Campo	Valor	Descripción
packet_type	111	Paquete MPEG-2 TS
NUMTS	1000	8 paquetes MPEG-2 TS
AHF	1	Existen paquetes nulos eliminados antes del primer paquete MPEG-2 TS encapsulado.
HDM	1	Se elimina el encabezado TS
DNP	000000 0	No existe paquetes TS nulos

Nota. La figura muestra los valores de campo del encabezado ALP al eliminar paquetes TS nulo.

El paquete ALP resultante tiene una longitud de 1477 bytes y ahorra 27 bytes en comparación con el caso en el que los ocho paquetes MPEG-TS se alimentan directamente a la capa PHY.

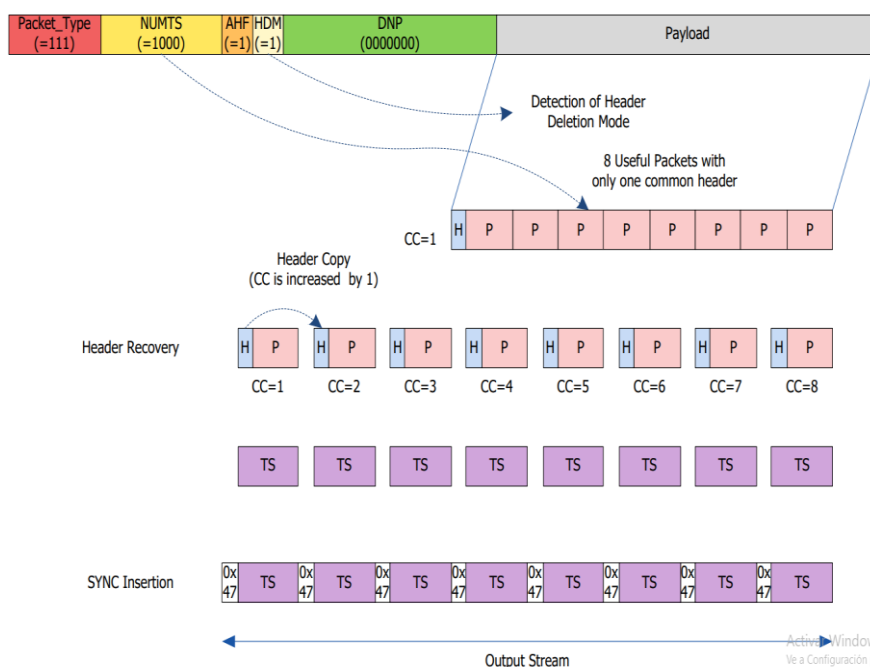
El proceso de desencapsulación de los paquetes MPEG-2 TS y la recuperación del encabezado MPEG-2 TS en el modo de eliminación de encabezado de paquetes MPEG-2 TS nulos eliminados se pueden describir en los siguientes pasos:

1. Se revisa el campo HDM para detectar el modo de eliminación
2. Con el campo NUMTS se obtiene el número de paquetes MPEG-2 TS encapsulados en el paquete ALP.
3. El primer paquete MPEG-2 TS está constituido por un encabezado de 3 bytes y un *payload* de 184 bytes, y los paquetes TS restantes solo incluyen el valor de *payload* de 184 bytes.

4. Se genera los encabezados de paquetes MPEG-2 TS utilizando como referencia el encabezado del primer paquete MPEG-2 TS, con lo cual cada campo CC sucesivo aumenta en uno.
5. Se inserta el byte de sincronización. La figura 25 muestra el proceso de desencapsulación anteriormente descrito.

Figura 25

Proceso de desencapsular paquetes MPEG-2 TS utilizando el modo eliminación de encabezado MPEG-2 TS desde un paquete ALP.



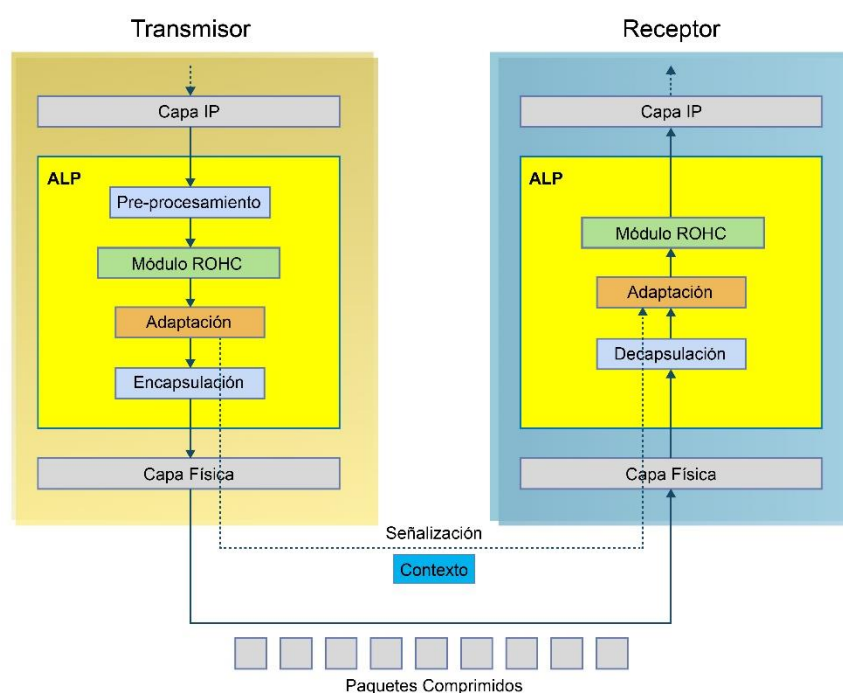
Nota. La figura muestra el proceso de desencapsulación de paquetes MPEG-2 TS utilizando el modo eliminación de encabezado MPEG-2 TS Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 42), 2019, Washington, D. C. 20006 202-872-9160.

Compresión del encabezado IP

Para el estándar ATSC 3.0, la capa de enlace cuenta con un esquema de compresión de encabezados IP cuya estructura se encuentra formada por dos módulos: encapsulación y desencapsulación de encabezado (*ROHC*) y el módulo de adaptación como se lo puede observar en la figura 26.

Figura 26

Estructura de la compresión y descompresión del encabezado IP.



Nota. Tomado de Doc. A/330:2019 por Advanced Transmission Systems Communications (ATSC) (p. 21), 2019, Washington, D. C. 20006 202-872-9160.

El esquema de compresión de encabezado IP se basa en la compresión de encabezado robusta (ROHC), el cual en el transmisor permite reducir el tamaño de encabezado de cada uno de los paquetes, posterior a esto, el módulo Adaptación extrae información de contexto y forma la información de señalización para cada flujo de

paquetes de datos. Al llegar al receptor, como primer paso el módulo Adaptación analiza la información de señalización asociada al flujo de paquete de datos, con esto el descompresor *ROHC* reconstruye el paquete IP original recuperando el encabezado del paquete IP.

El módulo pre-procesamiento se incluye en el transmisor para manejar de una manera óptima la compresión de encabezado de paquetes IP. Con este módulo se simplifica la cantidad de flujo IP para el módulo de compresión *ROHC*, ya que al trabajar con paquetes IP fragmentados puede causar una operación ineficiente por parte de *ROHC*, este módulo por lo general no se incluye en el receptor.

ROHC (Robust Header Compression)

Las características del enlace de Broadcasting de ATSC 3.0, sugieren que el framework de *ROHC* debe trabajar en el modo unidireccional (Robust Header Compression-Unidireccional Mode), en este modo se definen múltiples perfiles de compresión de encabezado, los cuales indican la combinación de protocolos específicos. Para el sistema ATSC 3.0 el perfil adoptado es el 0x0002 (*ROHC UDP*) que incorpora la combinación de los protocolos UDP/IP.

Módulo Adaptación

Si la transmisión de información se efectúa a través de un enlace unidireccional, y el receptor no obtiene la información de contexto, el módulo descompresor *ROHC* no posee la capacidad de recuperar el encabezado del paquete recibido hasta que reciba la información de contexto completo. Lo cual causa un retraso en el cambio de canal y un retraso en el encendido. Para evitar esto, la información de contexto y los parámetros de configuración entre el compresor *ROHC* y el descompresor *ROHC* deben incorporarse en el envío del flujo de paquetes.

Con el módulo Adaptación se permite transmitir fuera de la banda de la información de contexto y de los parámetros de configuración. Esto gracias a utilizar la señalización de la capa de enlace. Por lo tanto, la función de Adaptación se utiliza para reducir el retraso de cambio de canal y el error de descompresión debido a la pérdida de información de contexto.

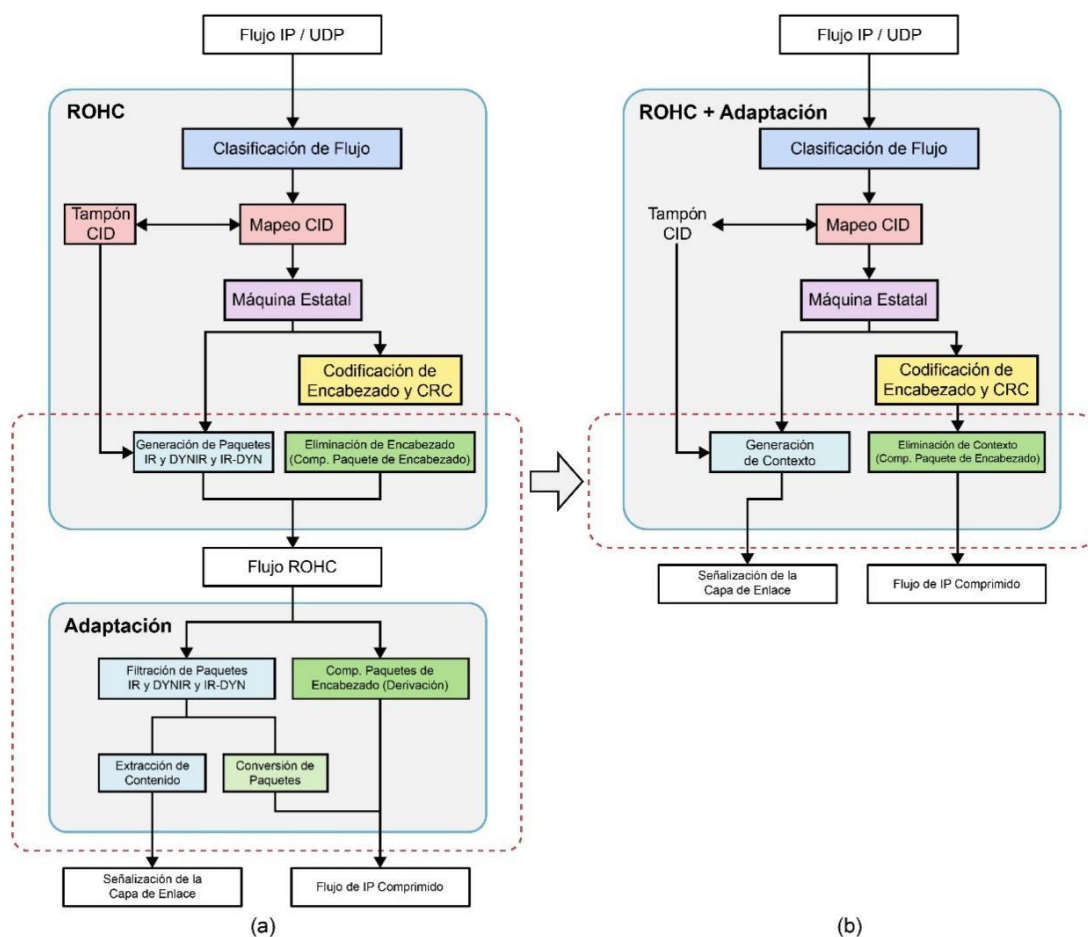
Al implementar el módulo de Adaptación se lo puede integrar con el módulo *ROHC*. La figura 27 muestra la estructura funcional del mecanismo de compresión de encabezados IP en ALP.

La figura 27 (a) representa una estructura separada entre el módulo *ROHC* y el módulo Adaptación. Cuando el flujo IP / UDP se envía al módulo *ROHC*, el compresor *ROHC* clasificará cada flujo de paquetes en función de la información de contexto estático. Para la misma información de contexto estático, se asigna el mismo *CID* (Identificador de contexto). Y estos *CID* se gestionarán en el búfer *CID*. Según la máquina de estado, cada encabezado de paquete IP / UDP se cambia a paquete *IR* (inicialización y actualización) o *IR-DYN* (IR-dinámico) o se codifica en encabezado comprimido. El flujo *ROHC* consta de estos paquetes *IR*, *IR-DYN* y comprimidos. El módulo de adaptación detecta el paquete *IR* e *IR-DYN* del flujo de paquetes *ROHC* y extrae la información de contexto. La cadena estática y la cadena dinámica se pueden extraer del paquete *IR* y la cadena dinámica se puede extraer del paquete *IR-DYN*. Después de extraer la información de contexto, cada paquete *IR* e *IR-DYN* se convierte en un paquete comprimido. El paquete comprimido convertido se incluye y se transmite dentro del flujo de paquetes *ROHC* en el mismo orden que el paquete *IR* o *IR-DYN*, reemplazando el paquete original. La información de contexto se configura y encapsula en la señalización de la capa de enlace. En el punto de vista lógico, esta estructura se puede considerar. Sin embargo, si se utilizará una solución de terceros de *ROHC* para la implementación, se puede utilizar esta estructura lógica.

La figura 27 (b) muestra la estructura combinada que genera la señalización de la capa de enlace que incluye información de contexto, en lugar del paquete IR o IR-DYN en el compresor ROHC. A excepción de esta operación, se pueden reutilizar otras estructuras de las estructuras ROHC existentes. Desde el punto de vista de la implementación, se puede considerar la estructura presentada en la Fig. 21 (b). La complejidad no aumentará con esta estructura en comparación con la estructura ROHC original (Kwon et al. 2016).

Figura 27

Estructura funcional de alto nivel de la compresión de encabezados IP en ALP.



Nota. La parte (a) presenta la estructura separada entre el módulo ROHC y el módulo de Adaptación. En la parte (b) se presenta la estructura combinada de la función ROHC y la función de adaptación. Adaptado de The ATSC Link-layer Protocol (ALP): Design and Efficiency Evaluation, 2016.

Señalización, Entrega, Sincronización, y Protección de errores (A/331)

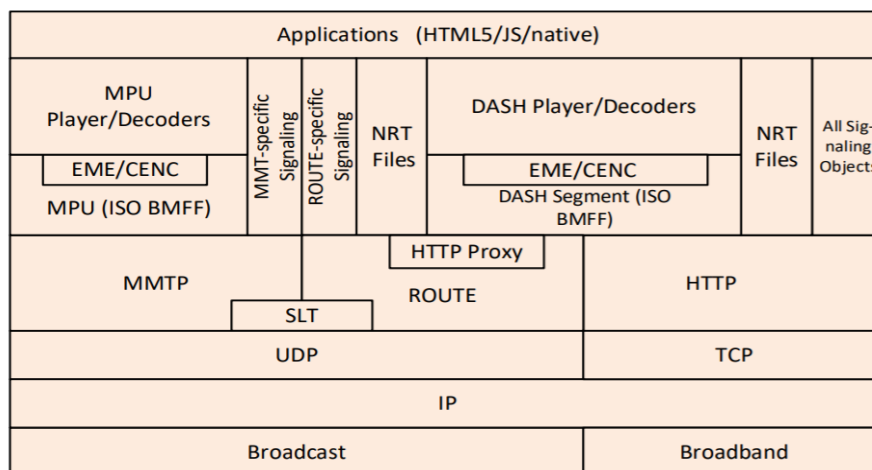
La norma Señalización, Entrega, Sincronización, y Protección de errores “*Signaling, Delivery, Synchronization, and Error Protection*” (A/331) especifica diferentes mecanismos y procesos para la entrega, señalización y transporte de contenido y servicios ATSC 3.0 sobre el protocolo IP hacia receptores ATSC 3.0 a través de redes híbridas (broadcast/broadband), los cuales son los protocolos:

- MMTP (Protocolo de Transporte de Medios MPEG)
- ROUTE (Entrega de Objetos en Tiempo Real sobre Transporte Unidireccional)

Estos protocolos pertenecen a la pila de protocolos de ATSC 3.0, como lo indica la figura 28.

Figura 28

Pila de protocolos de ATSC 3.0 conceptual.



Nota. Tomado de Doc. A/331:2020 por Advanced Transmission Systems Communications (ATSC) (p. 15), 2020, Washington, D. C. 20006 202-872-9160.

El método MMTP que se muestra en el lado izquierdo de la figura 28, se basa en *MPEG Media Transport (MMT)* utiliza el protocolo de transporte (MMTP) para la entrega de unidades de procesamiento de medios (MPU) y el método *ROUTE* que se muestra en el centro de la figura 28, se basa en *MPEG DASH (ROUTE-DASH)* (ATSC Standard: Signaling, Delivery, Synchronization, and Error Protection, 2020).

Características de los protocolos MMTP y ROUTE-DASH

Los protocolos *MMTP* y *ROUTE-DASH* permiten dar gestión a las siguientes características del sistema ATSC 3.0:

- Realizar transmisiones en tiempo real.
- La entrega eficiente y robusta de objetos basados en archivos.
- Soporte para la adquisición rápida de servicios por parte de los receptores (cambio rápido de canal).
- Soporte para servicios híbridos (broadcast/broadband).
- Corrección de errores hacia adelante (FEC) de alta eficiencia.
- Compatibilidad dentro de la infraestructura de transmisión. Con formatos y métodos de entrega desarrollados para el uso común dentro de Internet.
- Soporte para Gestión de derechos digitales (DRM), cifrado de contenido y seguridad.
- Soporte para definiciones de servicio en las que todos los componentes del servicio se entregan a través de la ruta de banda ancha (la adquisición de tales servicios todavía requiere el acceso a la señalización entregada en la emisión).
- Señalización para admitir códecs de audio y video de última generación.
- Entrega de contenido multimedia por solicitud.

- Entrega no multiplexada de componentes del servicio (por ejemplo, video y audio en transmisiones separadas).
- Soporte para transmisión adaptativa en contenido de transmisión de banda ancha.
- Vinculación adecuada a las funciones de la capa de aplicación, como ESG (Guía de Servicios Electrónicos) y contenido interactivo.

Reglas para el uso de MMTP o ROUTE-DASH

Las reglas para el uso de los protocolos MMTP o ROUTE-DASH en el transporte de contenido o servicios ATSC 3.0 serán las siguientes:

Para la entrega de transmisión de un servicio que no tenga una función basada en alguna aplicación ATSC 3.0, el contenido puede ser transportado por cualquiera de los dos protocolos (pero no por ambos) utilizando:

- Una o más sesiones de ROUTE-DASH.
- Una o más sesiones MMTP.

Para la entrega de transmisión de un servicio vinculado a una función basada en alguna aplicación ATSC 3.0, los componentes de contenido del servicio pueden ser transportado utilizando:

- Una o más sesiones de ROUTE-DASH.

No se permitirá el uso de MMTP y ROUTE-DASH para la transmisión de componentes multimedia en el mismo servicio.

Para la entrega de un servicio basado en una aplicación, un servicio ESG, un servicio EA o un servicio de datos DRM, los componentes de contenido del servicio son transportados utilizando:

- Una o más sesiones de ROUTE-DASH.

Cada sesión de ROUTE-DASH comprende uno o más canales de transporte de codificación en capas (LCT), el cual transporta en su totalidad, o en parte, los

componentes de contenido que componen el servicio ATSC 3.0. En la entrega de servicios de transmisión continua, un canal LCT puede transportar un componente individual de un servicio de usuario, como un flujo de audio, video o subtítulos. Los medios de transmisión están formateados como segmentos DASH.

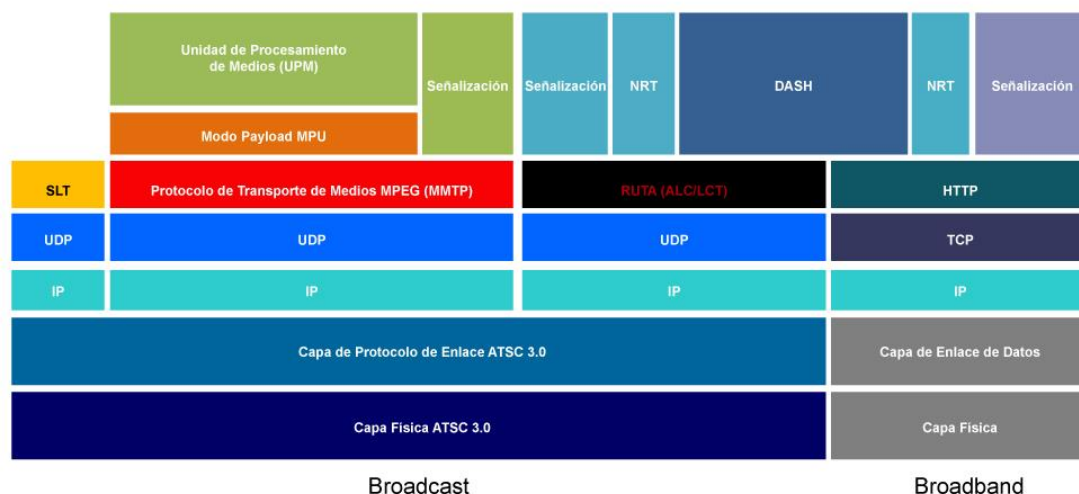
Cada sesión de MMTP puede comprender uno o más flujos de paquetes MMTP que transportan mensajes de señalización de MMT en su totalidad, o en parte, los componentes del contenido. En el flujo de paquetes MMTP se puede transportar mensajes de señalización MMT o componentes formateados por MMT mediante las unidades de procesamiento de medios (MPU).

Señalización de servicios

Los servicios y contenido ATSC 3.0 utiliza la funcionalidad de transporte de flujos objetos y archivos utilizando los protocolos de transporte de medios MPEG (MMTP) o el protocolo de entrega de objetos en tiempo real sobre transporte unidireccional (ROUTE) ya antes mencionados, Estos protocolos operan en multidifusión UDP/IP sobre la transmisión de capa física, y están habilitados por el protocolo HTTP en una unidifusión TCP/IP sobre la capa física de banda ancha (broadband). La capa de gestión de servicios admite principalmente los medios para el descubrimiento y la adquisición de servicios para permitir que diferentes tipos de servicios, como TV lineal y / o servicio de aplicación HTML5, sean transportados por las capas de entrega y física subyacentes. La figura 29 muestra la de forma concisa la pila de protocolos que se encuentran en un receptor ATSC 3.0.

Figura 29

Pila de protocolos en el receptor ATSC 3.0.



Nota. La figura presenta la pila de protocolos que se encuentran en un receptor ATSC 3.0. Tomado de Doc. A/331:2020 por Advanced Transmission Systems Communications (ATSC) (p. 16), 2020, Washington, D. C. 20006 202-872-9160.

La señalización de servicio proporciona información de descripción y descubrimiento de servicios, y consta de dos componentes funcionales: señalización de arranque a través de la tabla de lista de servicios (SLT) similar a la tabla de asociación de programas (PAT) que se encuentra en MPEG-2 y señalización de capa de servicio (SLS) que proporciona el descubrimiento y adquisición de servicios y contenido ATSC 3.0.

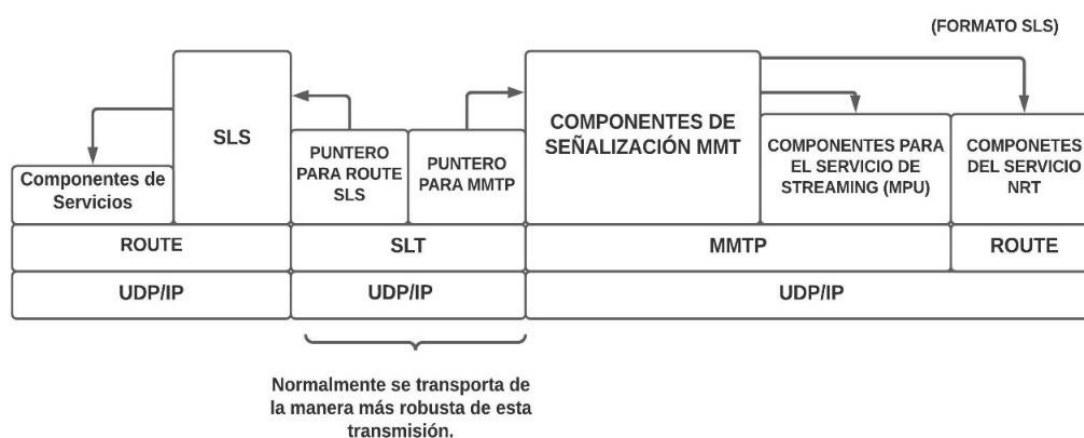
La información que se encuentra en dichas tablas es la requerida para la adquisición y descubrimiento de servicios o contenido ATSC 3.0.

La tabla SLT permite al receptor crear una lista de servicios básica y arrancar el descubrimiento de la tabla SLS para cada servicio o contenido ATSC 3.0. La tabla SLT puede permitir una adquisición muy rápida de información básica del servicio. La tabla

SLS, permite al receptor descubrir y acceder a los Servicios ATSC 3.0 y a sus diferentes componentes de contenido. La relación entre SLT y SLS para la señalización de ROUTE (para los servicios ROUTE-DASH) y la relación entre SLT y SLS para la señalización de MMT (para los servicios que utilizan el flujo de datos de MMTP / MPU) se muestran en la figura 30.

Figura 30

Relación de las tablas SLT y SLS en los protocolos MMTP y ROUTE-DASH.



Nota. La figura presenta la relación entre la tabla SLT y SLS tanto para el protocolo ROUTE (ROUTE-DASH) como para el protocolo MMT (MMTP/MPU). Tomado de Doc. A/331:2020 por *Advanced Transmission Systems Communications (ATSC)* (p. 17), 2020, Washington, D. C. 20006 202-872-9160.

Para los servicios ROUTE-DASH entregados a través de transmisión, el SLS se transporta por ROUTE/UDP/IP en uno de los canales de transporte LCT, incluida la sesión ROUTE, a una velocidad adecuada para admitir la unión y la conmutación rápidas de canales. Para la transmisión de MMTP/MPU entregada a través de la transmisión, el SLS se transmite mediante mensajes de señalización MMTP, a una velocidad adecuada para

admitir la rápida unión y conmutación de canales. En la entrega de banda ancha, el SLS se transporta a través de HTTP (S) /TCP/IP.

Flujos broadcast

En el sistema ATSC 3.0 un flujo de broadcast es la abstracción de un canal de RF (Radio Frecuencia), que se define en términos de una frecuencia portadora centrada dentro de un ancho de banda específico. Se identifica por el área geográfica y la frecuencia (ATSC Standard: Signaling, Delivery, Synchronization, and Error Protection, 2020).

Parte de un canal RF consta de una tubería de capa física (PLP). Cada PLP tiene ciertos parámetros de modulación y codificación e incluirá las siguientes tablas.

- Señalización de bajo nivel (LLS)
- Señalización de capa de servicios (SLS)

En al menos uno de los PLP que se encuentren transportando el contenido o servicio ATSC 3.0.

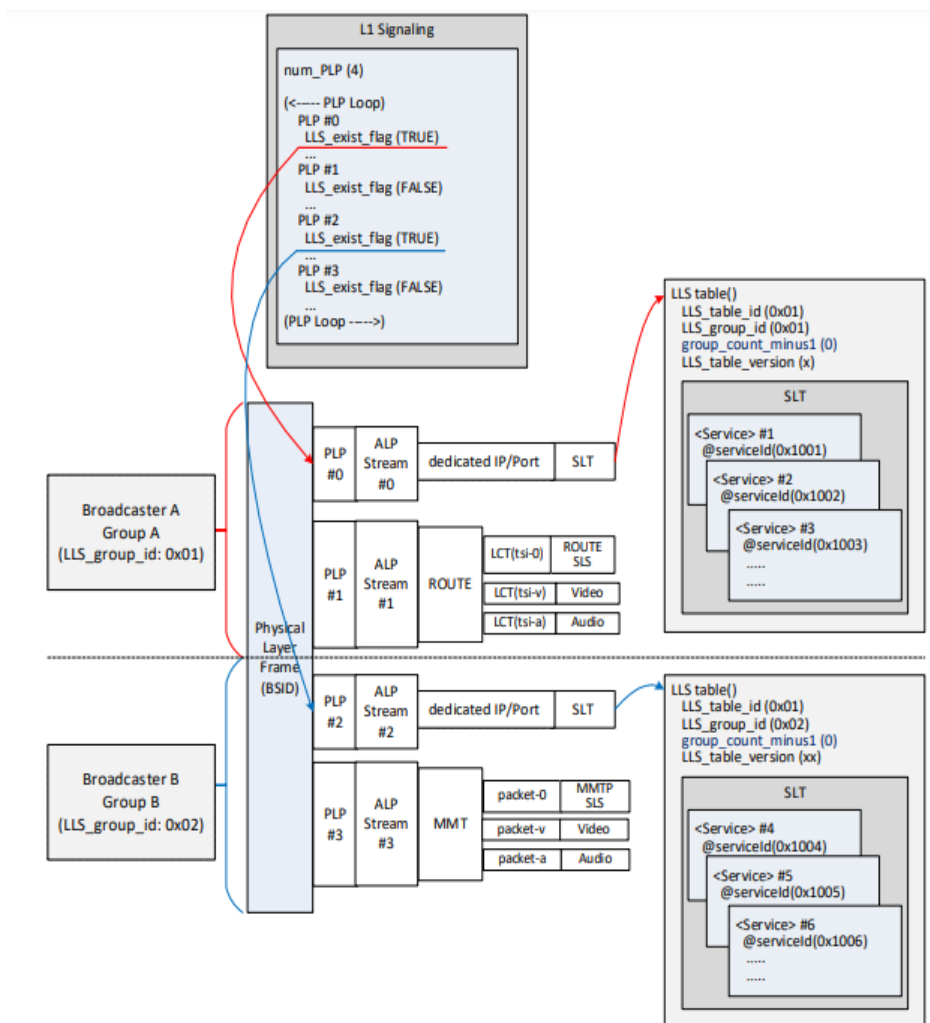
Hay un cierto número de "grupos" presentes en una transmisión determinada. Un grupo es una abstracción de una emisora de televisión o estación de televisión. La identidad de un grupo se expresa en señalización mediante un ***LLS_group_id*** definido en el formato de tabla LLS (Advanced Television Systems Committee, 2020).

Cada servicio en una transmisión está asociado con un solo valor ***LLS_group_id***. Esta característica de grupo permite que varias emisoras de televisión operen en un solo canal de RF con un grado sustancial de independencia, por ejemplo, una emisora puede enviar sus servicios bajo un grupo único señalado en ***LLS_group_id***.

La figura 31 presenta un ejemplo de la asignación del ***LLS_group_id*** para cada emisora de televisión (grupo).

Figura 31

Asignación de LLS_group_id por cada emisora (grupo) o estación de televisión.



Nota. La imagen muestra la asignación de valores al LLS_group_id al tener dos Grupos (emisoras de televisión). Tomado de Doc. A/331:2020 por Advanced Transmission Systems Communications (ATSC) (p. 19), 2020, Washington, D. C. 20006 202-872-9160.

Señalización de bajo nivel (LLS)

La señalización de bajo nivel (LLS) es información transportada directamente en el payload del paquete IP, en direcciones o puertos específicos, dedicados a esta función, las tablas elementales que se pueden encontrar en la señalización LLS son de los siguientes tipos:

- Tabla de lista de servicios (SLT)
- Tabla de clasificación de regiones (RRT)
- Fragmento de tiempo del sistema
- Tabla de alerta de emergencia avanzada (AEAT)
- Fragmento de notificación de mensaje en pantalla.

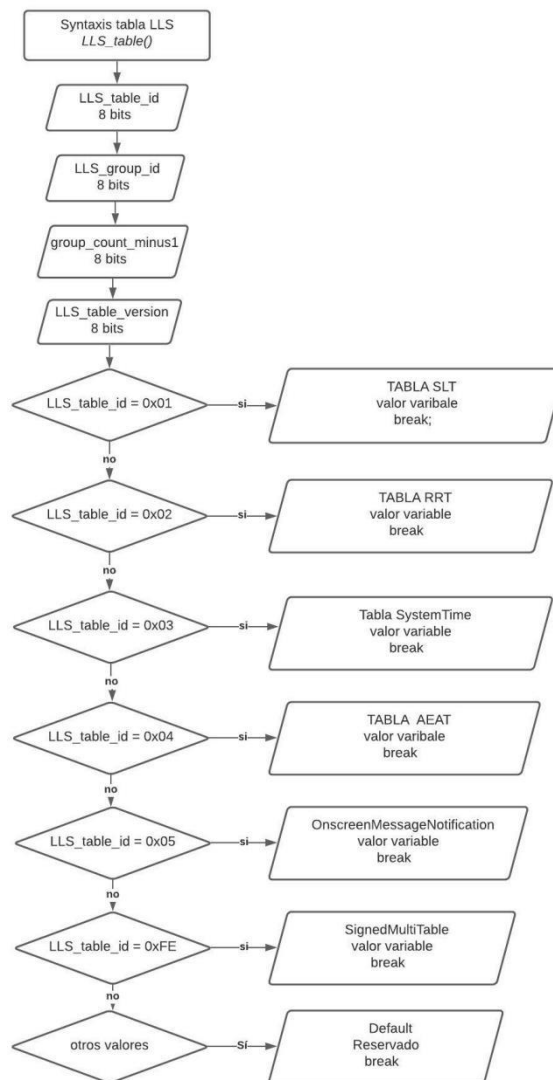
Existen tablas LLS adicionales como se indica en el Registro de puntos de código ATSC, donde se incluye la tabla de datos de certificación como se define en la norma A/360 y la tabla de canales de retorno dedicados como se define en la norma A/323. (ATSC Standard: Signaling, Delivery, Synchronization, and Error Protection, 2020).

Formato de la tabla LSS

Para los datos LSS que son entregados por paquetes UDP/IP su formato debe establecerse de acuerdo al tren de bits que se especifica en la figura 32. El primer byte de cada paquete UDP/IP que se encuentre transportando datos LLS, será el principio de una **LLS_table ()**. Una tabla LLS, la cual tendrá un valor máximo de longitud limitado por el paquete IP más grande que se puede entregar desde la capa física, el cual es de 65,507 bytes, este valor es el resultado del payload máximo de datos con formato UDP (65,535 bytes), menos 20 bytes que son utilizados para el encabezado IP y menos 8 bytes para el encabezado UDP, obtenemos el valor de 65,507 bytes. (ATSC Standard: Signaling, Delivery, Synchronization, and Error Protection, 2020).

Figura 32

Diagrama de flujo de la sintaxis de las tablas LLS.



Nota. La figura presenta el tipo de tabla que se encuentra en la señalización LLS, dependiendo de la sintaxis de bits que se encuentra en valor hexadecimal. Adoptado de Doc. A/331:2020 por Advanced Transmission Systems Communications (ATSC) (p. 26), 2020, Washington, D. C. 20006 202-872-9160.

A continuación, se presenta la descripción de cada campo que se encuentra en la figura 32.

LLS_table_id: Valor de 8 bits que permite identificar el tipo de tabla a ser entregada por el cuerpo en la información del paquete UDP. El valor 0x00 se encuentra reservado para servicios ATSC, y los valores indefinidos serán reservados para el uso de la industria.

LLS_group_id: Valor de 8 bits que permite asociar la instancia *del LLS_table ()* con un grupo de tablas con el mismo *LLS_group_id*. El alcance de este campo es el flujo de difusión. El valor de *LLS_group_id* será único dentro de este flujo de difusión.

group_count_minus1: Valor de 8 bits que indicará el número total de los diferentes grupos de tablas LLS, que se encuentran el tren de paquetes ALP transportado en una tubería de capa física (PLP). Un valor de 0 indica que *LLS_table ()* lleva un solo valor de *LLS_group_id*, un valor de 1 indica que hay dos presentes y así progresivamente.

LLS_table_version: Valor de 8 bits que se incrementará en 1 siempre que los datos de la tabla se identifican mediante una combinación de cambios *LLS_table_id* y *LLS_group_id*. Cuando el valor llega a 0xFF, el valor se establecerá en 0x00.

SLT: Tabla de lista de servicios en formato XML.

RRT - Una instancia de una Tabla de Región de Calificación conforme a la estructura de Tablas de Región de Calificación.

SystemTime: Fragmento de tiempo del sistema en formato XML.

AEAT: Fragmento de la tabla de alerta de emergencia avanzada en formato XML que se ajusta a la estructura del formato de mensaje de alerta de emergencia avanzada (AEA-MF).

OnscreenMessageNotification: Formato XML de notificación de mensajes en pantalla.

SignedMultiTable: Conjunto de una o más instancias de tabla LLS.

SignedMultiTable, Sin embargo, el contenido de cada tabla que contiene se comprime individualmente, o no, según lo definido por cada tabla.

UserDefined: Conjunto de datos XML definidos por el usuario.

Identificador de servicios

De acuerdo a la norma A/331 cada servicio se puede identificar de dos maneras: una compacta donde se utiliza en la tabla SLT, que es única dentro de un único conjunto de PLP vinculados, y de una manera global la cual se utiliza en el ESG (Guía de Servicios Electrónicos).

Cada sesión del protocolo ROUTE se identifica mediante una dirección IP de origen, una dirección IP de destino y un número de puerto de destino. Un canal LCT (asociado con los componentes de servicio que transporta) se identifica mediante un identificador de sesión de transporte (TSI) que es único dentro del alcance de la sesión de ROUTE principal y, además, único dentro del alcance de todas las sesiones de ROUTE definidas.

Para un Servicio dado en el S-TSID, las propiedades comunes a los canales LCT y ciertas propiedades exclusivas de los canales LCT individuales se dan en una estructura de señalización de ROUTE denominada Descripción de instancia de sesión de transporte basada en servicios (S-TSID), que forma parte de la señalización de la capa de servicio. Cada canal LCT se transporta a través de una única tubería de capa física (PLP). Cada PLP puede contener uno o más canales LCT.

Es posible que los diferentes canales LCT de una sesión ROUTE no estén contenidos en diferentes conductos de capa física. Las propiedades descritas en el S-TSID incluyen el valor TSI para cada canal LCT, descriptores para los objetos o archivos de entrega y parámetros FEC de la capa de aplicación.

Una sesión MMTP se identifica mediante la dirección IP de destino y el número de puerto de destino. Un flujo de paquetes MMTP (asociado con los componentes del servicio que transporta) se identifica mediante un packet que es único dentro del alcance de la sesión MMTP principal. Las propiedades comunes a todos los flujos de paquetes MMTP y ciertas propiedades de los flujos de paquetes MMTP individuales se dan en la información SLS. Las propiedades de cada sesión MMTP vienen dadas por mensajes de señalización MMT, que pueden llevarse a cabo dentro de la sesión MMTP relacionada.

Asignación de dirección IP

De acuerdo a la norma A/331 la información de señalización LLS debe ser transportado por paquetes IP con dirección 224.0.23.60 y el puerto de destino 4937 / UDP.

Todos los paquetes IP que no sean paquetes IP LLS deberán llevar una dirección IP de destino:

- a) Asignada y reservada por un mecanismo que garantice que las direcciones de destino en uso sean únicas en una región geográfica.
- b) Deben encontrarse en el rango de 239.255.0.0 a 239.255.255.255, donde el bit del tercer octeto corresponderá a un valor de *SLT.Service@majorChannelNo*, registrado en el organismo de radiodifusión para su uso en el área de servicio de la transmisión de radiodifusión, con las siguientes salvedades:
 - Si una entidad de radiodifusión opera transportando diferentes servicios en múltiples frecuencias de RF con todo o parte de su área de servicio en común, cada combinación de dirección IP y puerto será única en todas esas emisiones de radiodifusión.

- En el caso de que haya múltiples flujos LLS (por lo tanto, múltiples SLT) en una emisión de difusión dada, cada combinación de dirección IP y puerto en uso para flujos no LLS será única en todos los servicios en la emisión de difusión agregada.
- Para minimizar los impactos sobre la complejidad de la redistribución de paquetes de multidifusión IP en redes locales, se debe minimizar el número total de direcciones y puertos de multidifusión IP en uso por un servicio dado.
- La dirección y los números de puerto de una secuencia IP / UDP que transporta cualquier componente de un servicio en tiempo real solo cambiarán en un RAP de Servicio en tiempo real de ese Servicio.

Tabla de lista de servicios (SLT)

La tabla de lista de servicios (SLT) es uno de los tipos de instancia de información de LLS. La función del SLT es similar a la de la tabla de asociación de programas (PAT) en los sistemas MPEG-2. Para un receptor que se encuentra por primera vez con la emisión de transmisión, este es el lugar para comenzar. El SLT admite un escaneo rápido de canales que permite a un receptor crear una lista de todos los servicios que puede recibir, con su nombre de canal, número de canal, etc., además la tabla SLT proporciona información de arranque que permite al receptor descubrir la información SLS para cada uno de los servicios. Para los servicios entregados por el protocolo ROUTE-DASH, la información de arranque incluye la dirección IP de origen, la dirección IP de destino y el puerto de destino del canal LCT que transporta el SLS específico de la sesión ROUTE. Para los servicios prestados por MMTP-MPU, la información de arranque incluye la dirección IP de destino y el puerto de destino de la sesión MMTP que lleva el SLS específico del protocolo MMTP (ATSC Standard: Signaling, Delivery, Synchronization, and Error Protection, 2020).

La tabla SLT admite exploraciones rápidas de canales y adquisición de servicios al incluir la siguiente información sobre cada servicio en el flujo de transmisión:

- Información necesaria para permitir la presentación de una lista de servicios que sea significativa para los espectadores y que pueda respaldar la selección inicial del servicio mediante el número de canal o la selección hacia arriba / hacia abajo.
- Información necesaria para ubicar la señalización de la capa de servicio para cada servicio enumerado.

Cada SLT se repetirá en la información de señalización LLS en el que se esté transportando al menos una vez cada 5 segundos. Para reducir el tiempo de exploración de canales en el receptor, cada SLT debe repetirse con más frecuencia, preferiblemente una vez por segundo.

La información que se encuentra en la tabla SLT se presentará como un documento en formato XML, el cual contiene un elemento raíz SLT que se ajusta a las definiciones en el esquema XML que tiene espacio de nombres.

Descripción de la sintaxis de SLT

La tabla 11 describe de manera concisa la estructura del elemento SLT de una manera ilustrativa. Las especificaciones que siguen a la tabla dan la semántica a los diferentes atributos.

Tabla 12

Descripción de los valores para la tabla SLT según la norma A/331.

Nombre del atributo	USO	TIPO DE DATO	DESCRIPCIÓN CORTA
SLT			Elemento raíz del SLT
@bsid	1	slt:listOfUnsignedShort	Identifica una o más transmisiones que comprende diferentes servicios.

Nombre del atributo	USO	TIPO DE DATO	DESCRIPCIÓN CORTA
SLTCapabilities	0..1	sa:CapabilitiesType	Capacidades necesarias para decodificar y presentar de manera significativa el contenido de todos los servicios en esta instancia de SLT.
SLTInetUrl	0..N	anyURI	Dirección URL base que permite adquirir archivos de señalización de capa de servicio o ESG disponibles a través de banda ancha para los servicios de este SLT.
@urlType	1	unsignedByte	Tipo de archivos disponibles con esta URL.
Service	1..N		Información de servicio.
serviceld	1	unsignedShort	Número establecido en el servicio dentro del alcance del área de difusión.
@globalServiceID	0..1	anyURI	Un URI (Identificador uniforme de recursos) único a nivel mundial que ha establecido en el servicio ATSC 3.0. Este atributo no está presente para los servicios de datos ESG, EAS y DRM.
@sltSvcSeqNum	1	unsignedByte	Versión de la información del servicio SLT.
@protected	0..1	boolean	Indica si uno o más componentes necesarios para la presentación significativa del servicio se encuentran protegidos (por ejemplo, encriptados).
@majorChannelNo	0..1	unsignedShort 1..999	Número de canal principal del Servicio.

Nombre del atributo	USO	TIPO DE DATO	DESCRIPCIÓN CORTA
@minorChannelNo	0..1	unsignedShort 1..999	Número de canal menor del Servicio.
@serviceCategory	0..1	unsignedByte	Categoría de servicio, codificada según la Tabla 12
@shortServiceName	0..1	string	Nombre corto del servicio.
@hidden	0..1	boolean	Indica si el servicio está destinado a pruebas o uso exclusivo, y no debe ser seleccionado por receptores de TV normales.
@broadbandAccessRequired	0..1	boolean	Indica si se requiere acceso de banda ancha para que un receptor haga una presentación significativa del servicio.
@essential	0..1	boolean	Indica si la parte esencial del servicio se entrega a través de la transmisión actual.
@drmSystemID	0..1	listOfanyURI	Especifica los ID (Identificadores) del sistema DRM relacionados con el servicio.
SimulcastTSID	0..1	unsignedShort	Identificador de un flujo de transmisión ATSC 1.0 que lleva el mismo contenido de programación.
@simulcastMajorChannelNo	0..1	unsignedShort 1..999	Número de canal principal del servicio ATSC 1.0 con el mismo contenido de programación.
@simulcastMinorChannelNo	0..1	unsignedShort 1..999	Número de canal menor del servicio ATSC 1.0 que tiene el mismo contenido de programación.

Nombre del atributo	USO	TIPO DE DATO	DESCRIPCIÓN CORTA
SvcCapabilities	0..1	sa:CapabilitiesType	Capacidades requeridas para decodificar y presentar de manera significativa el contenido del servicio.
BroadcastSvcSignaling	0..1		Ubicación, protocolo, dirección, información de identificación para la señalización de transmisión.
@slsProtocol	1	unsignedByte	Protocolo utilizado para entregar la señalización de la capa de servicio para el Servicio, codificado en la tabla 13.
@slsMajorProtocolVersion	0..1	unsignedByte	Número de versión principal del protocolo utilizado para entregar la señalización de la capa de servicio para el servicio.
@slsMinorProtocolVersion	0..1	unsignedByte	Número de versión menor del protocolo utilizado para entregar la señalización de la capa de servicio para el servicio.
@slsDestinationIpAddress	1	IPv4address	Una cadena que contiene la dirección de destino IPv4 con puntos de los paquetes que transportan datos SLS de difusión para el servicio.
@slsDestinationUdpPort	1	unsignedShort	Número de puerto de los paquetes que transportan datos SLS de difusión para el servicio.
@slsSourceIpAddress	0..1	IPv4address	Una cadena que contiene la dirección de origen IPv4 con puntos de los paquetes que

Nombre del atributo	USO	TIPO DE DATO	DESCRIPCIÓN CORTA
			transportan datos SLS de difusión para este servicio.
SvcInetUrl	0..N	anyURI	URL para acceder a la señalización de Internet para este Servicio.
@urlType	1	unsignedByte	Tipo de archivos disponibles con esta URL.
OtherBsid	0..N	sl:ListOfUnsignedShort	Identificador de las diferentes transmisiones que se entreguen, donde se puede encontrar duplicados o partes del servicio.
@type	1	unsignedByte	Indica si la Transmisión se identifica por el OtherBsid, codificado en la tabla 14.

Nota. Cada uno de los nombres de atributos presentados, se encuentran de la misma manera en el documento SLT con formato XML. Tomado de Doc. A/331:2020 por *Advanced Transmission Systems Communications (ATSC)* (p. 28-29), 2020, Washington, D. C. 20006 202-872-9160.

Tabla 13

Codificación para el parámetro serviceCategory según la norma A/331.

serviceCategory	Significado
0	Reservado ATSC
1	Servicio A/V lineal
2	Servicio solamente audio lineal
3	Servicio basado en APP
4	ESG Service (program guide)

Nota. ServiceCategory posee un valor de 8 bits que indica la categoría de este servicio. Tomado de Doc. A/331:2020 por Advanced Transmission Systems Communications (ATSC) (p. 31), 2020, Washington, D. C. 20006 202-872-9160.

Tabla 14

Codificación para el atributo sIsProtocol según la norma A/331

sIsProtocol	Significado
0	Reservado ATSC
1	ROUTE
2	MMTP
Otros valores	Reservado ATSC

Nota. sIsProtocol permite identificar el tipo de protocolo de entrega de la señalización de la capa de servicio utilizado por el servicio. Tomado de Doc. A/331:2020 por Advanced Transmission Systems Communications (ATSC) (p. 32), 2020, Washington, D. C. 20006 202-872-9160.

Tabla 15

Codificación para el atributo type según la norma A/331.

type	Significado
0	Reservado ATSC
1	Duplicado
2	porción
Otros valores	Reservado ATSC

Nota. Type tiene la longitud de byte que permite identificar si el tren de difusión identificado por el OtherBsid entrega un duplicado o una parte del servicio. Tomado de Doc. A/331:2020 por Advanced Transmission Systems Communications (ATSC) (p. 33), 2020, Washington, D. C. 20006 202-872-9160.

Enlace de Programador / Estudio a Transmisor (Scheduler / Studio to Transmitter Link A/324)

La implementación del sistema ATSC 3.0 está estructurado por la interconexión de un sin número de capas. La interconexión entre las capas de Transporte y Física constituye una de las fundamentales del sistema ATSC 3.0. Estas permiten la interoperación entre los diversos segmentos del sistema, para lograr este objetivo se crean protocolos que permitan que una gran variedad de equipos necesarios para establecer el sistema ATSC 3.0, puedan ensamblarse. En la norma A/324 se pueden encontrar 4 protocolos definidos:

- ALPTP (Protocolo de transporte del protocolo de capa de enlace ATSC).
- STLP (Protocolo de transporte de enlace de estudio a transmisor).
- DSTP (Protocolo de transporte de fuente de datos)
- DSCP (Protocolo de control de fuente de información)

En base a la Figura 5 se procede a describir de manera concisa el funcionamiento de los protocolos ALTP, STLP y DSTP que intervienen directamente entre la interconexión de la capa de Transporte y la Capa Física.

Los tres protocolos a ser analizados permiten transportar diversos contenidos, información de control y metadatos que componen una transmisión ATSC 3.0, cada uno de estos protocolos utilizan el protocolo de túnel común (CTP), el cual permite ocultar varios aspectos del protocolo individual con el objetivo de cada uno de estos puedan ser tratados de manera similar cuando se enrutan a través de un estudio de transmisión. El

CTP encapsulará estos diversos trenes tunelizados en un tren de túnel RTP / UDP / IP basado en SMPTE ST 2022-1 (Corrección de errores de reenvío para el transporte de video / audio en tiempo real a través de redes IP) (ATSC Standard: Scheduler / Studio to Transmitter Link, 2021).

DSTP (Protocolo de transporte de fuente de datos)

El Protocolo de transporte de fuente de datos (DSTP) proporciona una solución para transferir datos de contenido para el sistema ATSC 3.0, que se encuentren asociados a metadatos y señalización a través de una red IP típica, teniendo como emisor fuentes de datos (MMTP o ROUTE-DASH) que tendrán como receptor puertas de enlace broadcast. Definiendo un encabezado de información de paquetes tunelizados que permite evitar modificaciones en los paquetes tunelizados de fuentes de datos. Este encabezado proporciona información para ayudar al generador de ALP a enrutar los diversos paquetes al PLP apropiado. Si bien se espera que las salidas de la fuente de datos sean propicias para el transporte en redes IP típicas, no proporcionan los recursos para el transporte de metadatos y señalización relacionados con el contenido que debe comunicarse a las puertas de enlace broadcast. Además, los paquetes de la fuente de datos deben mantener su orden original y no se puede tolerar la pérdida de paquetes, utiliza el protocolo el Protocolo de túnel común (CTP). Así cada paquete de fuente de datos se le antepone un encabezado de información de paquetes en diferentes túneles para transportar los metadatos y la información de señalización, que son requeridos para comunicarse desde las fuentes de datos a las puertas de enlace broadcast. Los encabezados de túnel DSTP y los encabezados de información de paquetes de túnel son descartados por la puerta de enlace broadcast una vez que se procesan los metadatos contenidos en ellos.

Los Paquetes Tunelizados dentro del DSTP pueden firmarse criptográficamente utilizando el Sistema de Seguridad de Contenido definido para el CTP. La etiqueta GMAC (Código de Autenticación de Mensajes Basado en GCM) que resulta del proceso de firma que se incluye en el encabezado de información de paquetes en túnel para cada paquete firmado. Cuando la firma está activa, el flujo de datos de seguridad definido por el sistema de seguridad de contenido se canaliza a través del DSTP utilizando encabezados de información de paquetes de túnel truncados (ATSC Standard: Scheduler / Studio to Transmitter Link, 2021).

ALTP (Protocolo de transporte del protocolo de capa de enlace ATSC)

El ATSC Protocolo de Transporte del Protocolo de Capa de Enlace ATSC (ALPTP) proporciona una solución para transferir paquetes del Protocolo de Capa de Enlace ATSC (ALP). Mediante una red IP de características típicas entre dos dispositivos que se encuentran separados. Los paquetes ALP representan construcciones con estructuras simples que posee un solo encabezado como mínimo, más la información suficiente de longitud de paquete solo para una capa de enlace de emisión. Dichos encabezados no representan los suficientes para transferir paquetes ALP a través de una red IP típica; por lo tanto, ALPTP se define para proporcionar la información de red adicional para poder transferir los paquetes ALP sin problemas. Además, los paquetes ALP deben mantenerse en secuencia para cada PLP y no se puede tolerar la pérdida de paquetes. ALPTP utiliza el Protocolo común de tunelización (CTP). Los paquetes ALP se tunelizan a través del CTP precedidos por un encabezado de información de paquetes tunelizados que describe los metadatos asociados con el paquete ALP, incluida su prioridad e ID (Identificador) de PLP. Hay un único flujo ALP asociado con cada PLP y el ID de PLP en los encabezados de información de paquetes tunelizados ALPTP permite agrupar estos paquetes.

Se debe tener en cuenta que no existe restricciones sobre el número de flujos ALPTP que se pueden comunicar entre un generador ALP y el planificador, ya que los metadatos relacionados con la vinculación de paquetes ALP a PLP están contenidos en el encabezado de información de paquetes tunelizados de cada paquete ALP en el túnel ALPTP. Los Paquetes Tunelizados dentro del ALPTP pueden firmarse criptográficamente usando el Sistema de Seguridad de Contenido definido para el CTP. La etiqueta GMAC (Código de Autenticación de Mensajes Basado en GCM) que resulta del proceso de firma se incluye en el encabezado de información de paquetes en túnel para cada paquete firmado. Cuando la firma está activa, el flujo de datos de seguridad definido por el sistema de seguridad de contenido se tuneliza a través del ALPTP utilizando encabezados de información de paquetes tunelizados truncados (ATSC Standard: Scheduler / Studio to Transmitter Link, 2021).

STLP (Protocolo de transporte de enlace de estudio a transmisor)

El protocolo de transporte de enlace de estudio a transmisor (STLTP) proporciona una solución para transferir un número potencialmente grande de flujos de datos de paquetes de banda base (BBP) paralelos que transportan contenido para un número similar de flujos PLP, a los cuales se les adiciona datos de preámbulo y señalización de control del transmisor a través de una red IP típica, desde una salida de la puerta de enlace a uno o más transmisores mediante tunelización. Cada flujo de BBP tunelizado que alimenta un PLP se deriva de un flujo de ALP correspondiente, y los ID de PLP proporcionados con los flujos de ALP se utilizan para establecer las asociaciones. La codificación de corrección de errores (ECC) se especifica para mantener la entrega confiable de datos STL (Datos del protocolo de transporte de enlace de estudio a transmisor) a través de los enlaces entre sitios, pero el método ECC seleccionado requiere el uso de paquetes de tamaño uniforme en su procesamiento de datos. Para

adaptarse a este requisito, los múltiples flujos de PLP, preámbulo y control se multiplexan juntos paquete por paquete y luego se tunelizan a través de un único flujo de datos del protocolo de túnel común RTP / UDP / IP resultante. Cuando el enlace de distribución no admite multidifusión (p. Ej., Una red IP pública), también se puede utilizar una pila de protocolos RTP / UDP / IP Unicast para entregar PLP multiplexados, Preámbulo, control y flujos de seguridad (ATSC Standard: Scheduler / Studio to Transmitter Link, 2021).

Capítulo IV

Análisis de la capa de transporte del estándar ATSC 3.0

Softwares para el análisis de flujos de muestra ATSC 3.0

Para el análisis de la capa de transporte del estándar ATSC 3.0 existen diferentes opciones en el mercado, los cuales incorporan conjuntos de productos ATSC 3.0 destinados a desarrolladores y usuarios finales que tengan como objetivo la validación de productos, realizar pruebas de campo o laboratorio o tengan como objetivo ampliar sus conocimientos sobre el estándar ATSC 3.0, algunos de estos productos se especifican en la tabla 15.

Tabla 16

Productos en el mercado especializados para el análisis del estándar ATSC 3.0.

Producto	Empresa	Características	Precio
DTC-386-ATSC3 *Incluye: Atsc3Xpress y modulador ATSC 3.0	DekTec	Generador de prueba que permite la creación de una señal ATSC 3.0 con control total sobre los parámetros de ATSC 3.0 como subtramas, multi-PLP y una variedad de estrategias de multiplexación de PLP.	€ 1250
DTA-2131-XPRT *Incluye: Receptor terrestre / cable multiestándar para PCIe con Atsc3Xpert, C2Xpert, T2Xpert.	Dektec	Receptor y analizador ATSC 3.0 con medidas de RF avanzadas y decodificación de toda la información de señalización. Cuenta con grabación de datos PLP en archivos PCAP y salida ALP y ROUTE / MMT en tiempo real a través de IP.	€ 2505
ATSC 3.0 LabMod	testree	Este modulador es el primero en ser compatible con STL Gateway para laboratorio, permite generar señales de RF ATSC 3.0 en vivo o archivos patrones IQ que admiten todo el estándar ATSC 3.0 con soporte LDM	€ 3600

Nota. Diferentes equipos que permiten la generación, análisis y estudio de señales ATSC 3.0 que se encuentran en el mercado.

Como se indica en la tabla 15 los equipos descritos permiten tanto la generación de señales ATSC 3.0 como la captura y análisis de flujos de paquetes de datos ATSC 3.0.

Software de distribución libre para el análisis de flujos ATSC 3.0

Una opción para el análisis de la capa de transporte del estándar ATSC 3.0, es utilizar el software de distribución libre conocido como la biblioteca *libatsc3*. El cual fue desarrollado por Jason Justman, Ingeniero distinguido en ONE Media 3.0/ Sinclair Broadcast Group, este software se encuentra destinado a ampliar la comprensión del estándar ATSC 3.0, a través del procesamiento de medios, análisis de paquetes de datos PCAP de transmisión OTA (“OVER-THE AIR”) de muestra y la reproducción de medios. Su diseño se centra en la norma A/331 que se ocupa de la señalización, entrega y la sincronización de contenido y su núcleo se encuentra escrito en el lenguaje de programación C, como lo indica Jason Justman (Jason, 2019).

La biblioteca *libatsc3* puede operar sobre sistemas operativos Linux, abarca herramientas de código abierto que permiten la reproducción básica de video ATSC 3.0 MMT y ROUTE-DASH, además de la presentación de la señalización correspondiente y el procesamiento de metadatos.

Biblioteca libatsc3

Como lo indica Jason Justman, la biblioteca *libatsc3* presenta las siguientes características dentro de su funcionamiento:

- Procesamiento de tráfico de transmisión múltiple IP A/331 (datagrama UDP sin procesar) en payload de datos MMT y ROUTE-DASH.
- La salida de la biblioteca para MMT son flujos elementales tales como:

Service_id seleccionado

Cuadro init / moov (estable)

MOOF (MPU)

mpu_sequence_number

Datos de muestra de MDAT por cada packet_id

Mensaje de señalización MMT para resolución MPT packet_id

La salida de la biblioteca para ROUTE-DASH es el flujo seleccionado para un service_id.

Estas salidas de información pertenecen a la información de señalización de bajo nivel LLS y a tabla de servicios SLT, como ha parámetros propios del protocolo MMT (Jason, 2019).

Instalación de libatsc3

La biblioteca *libatsc3* es un software de distribución libre que corre bajo un sistema operativo basado en Linux. Para el desarrollo del objetivo de esta tesis se trabajó en una plataforma virtual Ubuntu 18.04.5 hospedada en VirtualBox y como pasos previos a la instalación de la biblioteca *libatsc3* se debe cumplir con los siguientes requerimientos:

- La instalación de las bibliotecas *ncurses*, *libpcap*, *cmake*, se puede instalar estas bibliotecas a través del siguiente comando:

```
sudo apt-get install libncurses5-dev libncursesw5-dev libpcap-dev cmake
```

- *Ffplay* es un software que en su gran mayoría las distribuciones Linux ya poseen instalado en sus repositorios, de no ser ese caso lo podemos instalar mediante el siguiente comando:

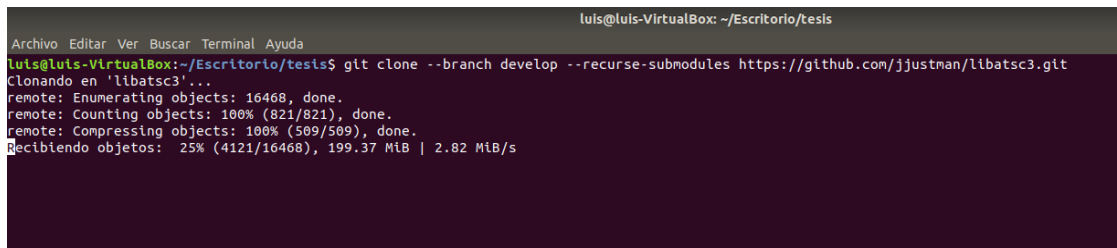
```
sudo apt-get install ffmpeg
```

A continuación, se procede a la clonación de la biblioteca *libatsc3*, la cual se encuentra disponible en la plataforma conocida como GitHub, esta es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git (GitHub, 2010). Con el siguiente comando se puede realizar la clonación de la biblioteca en la ubicación donde se requiera, para el desarrollo de esta tesis se creó previamente una carpeta con el nombre de *tesis* situada en el Escritorio como se muestra en la figura 33.

```
git clone --branch develop --recurse-submodules  
https://github.com/jjustman/libatsc3.git
```

Figura 33

Clonación de la biblioteca libatsc3 en la dirección /Escritorio/tesis.



```
luis@luis-VirtualBox: ~/Escritorio/tesis  
Archivo Editar Ver Buscar Terminal Ayuda  
luis@luis-VirtualBox:~/Escritorio/tesis$ git clone --branch develop --recurse-submodules https://github.com/jjustman/libatsc3.git  
Clonando en 'libatsc3'...  
remote: Enumerating objects: 16468, done.  
remote: Counting objects: 100% (821/821), done.  
remote: Compressing objects: 100% (509/509), done.  
Receiving objetos: 25% (4121/16468), 199.37 MiB | 2.82 MiB/s
```

Nota. La clonación de la biblioteca *libatsc3* se puede realizar en la carpeta que se requiera, en este caso se seleccionó la ubicación */Escritorio/tesis* para tener un fácil acceso.

Al momento de la clonación existen algunos módulos que solicitan un usuario y una contraseña, debido a que estos repositorios no se pueden distribuir públicamente, Jason Justman recomienda ignorar su clonación ya que la compilación del núcleo de la

biblioteca libatsc3 no requieren de dichos módulos para su correcto funcionamiento, como se muestra en la figura 34.

Figura 34

Módulos que requieren usuario y contraseña para la descarga de la biblioteca libatsc3.

```

renote: Enumerating objects: 1341, done.
renote: Counting objects: 100% (50/50), done.
renote: Compressing objects: 100% (43/43), done.
renote: Total 1341 (delta 14), reused 18 (delta 6), pack-reused 1291
Recibiendo objetos: 100% (1341/1341), 13.24 MiB | 6.02 MiB/s, listo.
Resolviendo deltas: 100% (840/840), listo.
Clonando en '/home/Luis/Escritorio/tests/libatsc3/saankhyalabs-slsdk'...
Username for 'https://dev.azure.com': Password for 'https://dev.azure.com':
fatal: Authentication failed for 'https://dev.azure.com/NGMP/Mark%20One%20Phone/_git/saankhyalabs-slsdk/'
fatal: clon de 'https://dev.azure.com/NGMP/Mark%20One%20Phone/_git/saankhyalabs-slsdk' en la ruta de submódulo '/home/Luis/Escritorio/tesis/libatsc3/saankhyalabs-slsdk' falló
Falló al clonar 'saankhyalabs-slsdk'. Reintento programado
Clonando en '/home/Luis/Escritorio/tesis/libatsc3/srt'...
renote: Enumerating objects: 8633, done.
renote: Counting objects: 100% (25/25), done.
renote: Compressing objects: 100% (18/18), done.
renote: Total 8633 (delta 10), reused 21 (delta 7), pack-reused 8608
Recibiendo objetos: 100% (8633/8633), 9.89 MiB | 5.67 MiB/s, listo.
Resolviendo deltas: 100% (6707/6707), listo.

```

Nota. Los módulos que requieren usuario y contraseña pertenecen a grupos de investigación privados, los cuales requieren de pago para su uso, para ignorar estos módulos solo basta con presionar la tecla *Enter* y continuará con la clonación.

Una vez clonada la biblioteca se debe dirigir a la ubicación de la carpeta **srt** y verificar que se haya clonado todos los archivos pertenecientes a esa carpeta como se lo muestra en la figura 35 esto se lo puede realizar mediante los siguientes comandos:

```

cd Escritorio/tesis/ libatsc3/srt
ls -l

```

Figura 35

Ingreso a la carpeta srt, para la verificación de su correcta descarga.

```
luis@luis-VirtualBox:~$ cd Escritorio/tesis/libatsc3/srt
luis@luis-VirtualBox:~/Escritorio/tesis/libatsc3/srt$ ls
luis@luis-VirtualBox:~/Escritorio/tesis/libatsc3/srt$
```

Puede que el contenido de la carpeta *srt* no se clone con el resto de repositorios de la biblioteca *libats3*, esto es debido a que La carpeta *srt* está compartida por otro proyecto padre, por lo que su clonación junto al resto de repositorios no es posible, es recomendable eliminar dicha carpeta y clonar directamente el contenido de esta carpeta desde el proyecto padre alojado igualmente en GitHub como se muestra en la figura 36, esto se lo puede realizar con los siguientes comandos:

```
sudo rm -rf Escritorio/tesis/libatsc3/srt
cd Escritorio/tesis/libatsc3
ls -l
git clone https://github.com/jjustman/srt.git
```


Figura 36

Clonación correcta de la carpeta srt, directamente desde el GitHub.

```

luis@Luis-VirtualBox:~$ sudo rm -rf Escritorio/tesis/libatsc3/srt
luis@Luis-VirtualBox:~$ cd Escritorio/tesis/libatsc3/
luis@Luis-VirtualBox:~/Escritorio/tesis/libatsc3$ ls -l
total 268
drwxrwxr-x  2 luis luis  4096 jun  2 22:52  airwavz_redzone_sdk
drwxrwxr-x 11 luis luis  4096 jun  2 21:53  android
drwxrwxr-x  2 luis luis  4096 jun  2 21:53  architecture
drwxrwxr-x  2 luis luis  4096 jun  2 21:53  Bear
drwxrwxr-x  4 luis luis  4096 jun  2 21:53  bento
-rwxrwxr-x  1 luis luis    25 jun  2 21:53  build_docs
drwxrwxr-x  2 luis luis  4096 jun  2 21:54  codornicesrq
drwxrwxr-x  2 luis luis  4096 jun  2 21:53  Collections-C
drwxrwxr-x  2 luis luis  4096 jun  2 21:53  docker
drwxrwxr-x  2 luis luis  4096 jun  2 21:54  ffmpeg
drwxrwxr-x  2 luis luis  4096 jun  2 21:53  'Flow analysis'
drwxrwxr-x  2 luis luis  4096 jun  2 22:51  gpac
drwxrwxr-x  2 luis luis  4096 jun  2 22:51  jsnm
-rw-rw-r--  1 luis luis 114508 jun  2 21:53  libatsc3.doxygen
drwxrwxr-x  2 luis luis  4096 jun  2 22:51  libdaa
drwxrwxr-x  2 luis luis  4096 jun  2 21:53  libmicrohttpd
drwxrwxr-x  2 luis luis  4096 jun  2 22:51  libtree
drwxrwxr-x  2 luis luis  4096 jun  2 22:51  libusb_android
drwxrwxr-x  2 luis luis  4096 jun  2 22:51  libwebsockets
-rw-rw-r--  1 luis luis  1065 jun  2 21:53  LICENSE
-rwxrwxr-x  1 luis luis   140 jun  2 21:53  linux-build-deps
drwxrwxr-x  2 luis luis  4096 jun  2 22:51  openHEVC
drwxrwxr-x  2 luis luis  4096 jun  2 22:52  pcre2-win-build
-rw-rw-r--  1 luis luis  11001 jun  2 21:53  README.md
drwxrwxr-x  2 luis luis  4096 jun  2 22:52  saankhyalabs-slsdk
drwxrwxr-x  6 luis luis  4096 jun  2 21:53  specs
drwxrwxr-x 17 luis luis 16384 jun  2 21:53  src
drwxrwxr-x  5 luis luis  4096 jun  2 21:53  support_scripts
drwxrwxr-x  9 luis luis  4096 jun  2 21:53  test_data
drwxrwxr-x  2 luis luis  4096 jun  2 22:52  vcpkg
drwxrwxr-x  2 luis luis  4096 jun  2 21:53  vs_build
drwxrwxr-x 13 luis luis  4096 jun  2 21:53  win-zlib123dll
drwxrwxr-x  7 luis luis  4096 jun  2 21:53  WpdPack
drwxrwxr-x  4 luis luis  4096 jun  2 21:53  xcode
drwxrwxr-x  5 luis luis  4096 jun  2 21:53  zlib-1.2.3-lib-win
luis@Luis-VirtualBox:~/Escritorio/tesis/libatsc3$ git clone https://github.com/jjustman/srt.git
Clonando en 'srt'...
remote: Enumerating objects: 8633, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 8633 (delta 10), reused 22 (delta 8), pack-reused 8608
Recibiendo objetos: 100% (8633/8633), 0.88 MiB | 2.76 MiB/s, listo.
Resolviendo deltas: 100% (6700/6700), listo.

```

A continuación, se procede a dirigirnos a la carpeta src y efectuar la instalación como se muestra en la figura 37, para lo cual se utiliza los siguientes comandos:

```

cd Escritorio/tesis/libatsc3/src
sudo ./build_linux.sh

```

Figura 37

Instalación completa de la biblioteca libatsc3.

```
tsc3_logging_extens.h:261:117: note: in definition of macro '_LIBATSC3_TIMESTAMP_INFO'
#define _LIBATSC3_TIMESTAMP_INFO(...) printf("%-24.24s:%4d:INFO :%.4f:",__FILENAME__,__LINE__,gt()); printf(_VA_ARGS__); printf("%sks",r,"n");

tsc3_sl_tlv_denod_type.c:515:2: note: in expansion of macro '_SL_TLV_DEMOD_INFO'
_Sl_TLV_DEMOD_INFO(" total_tlv_bytes_read: %llu", atsc3_sl_tlv_payload_metrics->total_tlv_bytes_read);
^
tsc3_sl_tlv_denod_type.c:529:22: warning: format '%llu' expects argument of type 'long long unsigned int', but argument 2 has type 'uint64_t [aka long unsigned int]' [-Wformat=]
_Sl_TLV_DEMOD_INFO(" total_tlv_bytes_discarded_due_to_magic_mismatch_count: %llu", atsc3_sl_tlv_payload_metrics->total_tlv_bytes_discarded_due_to_magic_mismatch_count);
^
tsc3_logging_extens.h:261:117: note: in definition of macro '_LIBATSC3_TIMESTAMP_INFO'
#define _LIBATSC3_TIMESTAMP_INFO(...) printf("%-24.24s:%4d:INFO :%.4f:",__FILENAME__,__LINE__,gt()); printf(_VA_ARGS__); printf("%sks",r,"n");

tsc3_sl_tlv_denod_type.c:529:2: note: in expansion of macro '_SL_TLV_DEMOD_INFO'
_Sl_TLV_DEMOD_INFO(" total_tlv_bytes_discarded_due_to_magic_mismatch_count: %llu", atsc3_sl_tlv_payload_metrics->total_tlv_bytes_discarded_due_to_magic_mismatch_count);
^
tsc3_sl_tlv_denod_type.c:554:22: warning: format '%llu' expects argument of type 'long long unsigned int', but argument 2 has type 'uint64_t [aka long unsigned int]' [-Wformat=]
_Sl_TLV_DEMOD_INFO(" total_alp_packet_type_ip_packets_bytes_read: %llu", atsc3_sl_tlv_payload_metrics->total_alp_packet_type_ip_packets_bytes_read);
^
tsc3_logging_extens.h:261:117: note: in definition of macro '_LIBATSC3_TIMESTAMP_INFO'
#define _LIBATSC3_TIMESTAMP_INFO(...) printf("%-24.24s:%4d:INFO :%.4f:",__FILENAME__,__LINE__,gt()); printf(_VA_ARGS__); printf("%sks",r,"n");

tsc3_sl_tlv_denod_type.c:554:2: note: in expansion of macro '_SL_TLV_DEMOD_INFO'
_Sl_TLV_DEMOD_INFO(" total_alp_packet_type_ip_packets_bytes_read: %llu", atsc3_sl_tlv_payload_metrics->total_alp_packet_type_ip_packets_bytes_read);
^
tsc3_sl_tlv_denod_type.c:557:22: warning: format '%llu' expects argument of type 'long long unsigned int', but argument 2 has type 'uint64_t [aka long unsigned int]' [-Wformat=]
_Sl_TLV_DEMOD_INFO(" total_alp_packet_type_link_layer_signalling_bytes_read: %llu", atsc3_sl_tlv_payload_metrics->total_alp_packet_type_link_layer_signalling_bytes_read);
^
tsc3_logging_extens.h:261:117: note: in definition of macro '_LIBATSC3_TIMESTAMP_INFO'
#define _LIBATSC3_TIMESTAMP_INFO(...) printf("%-24.24s:%4d:INFO :%.4f:",__FILENAME__,__LINE__,gt()); printf(_VA_ARGS__); printf("%sks",r,"n");

tsc3_sl_tlv_denod_type.c:557:2: note: in expansion of macro '_SL_TLV_DEMOD_INFO'
_Sl_TLV_DEMOD_INFO(" total_alp_packet_type_link_layer_signalling_bytes_read: %llu", atsc3_sl_tlv_payload_metrics->total_alp_packet_type_link_layer_signalling_bytes_read);
^
tsc3_sl_tlv_denod_type.c: At top level:
cc1: warning: unrecognized command line option '-Wno-writable-strings'
cc1: warning: unrecognized command line option '-Wno-c++11-extensions'
cc1: warning: unrecognized command line option '-Wno-builtin-headers'
cc1: warning: unrecognized command line option '-Wno-nullability-completeness'
cc -Wno-nullability-completeness -Wno-expansion-to-defined -Wno-builtin-headers -Wno-c++11-extensions -Wno-writable-strings -I ../android/atsc3_core/libpcr/include -I ../op
atsc3_sl_tlv_denod_type.o \
libatsc3.o \
-L/usr/local/lib -lz -ln -lthread -lpcap -lpcr2-8 -lstdc++ \
-L ../openssl/builddir/lib -lcrypto -lssl \
-o test_sl/atsc3_sl_tlv_parser_to_alp_test
cc -Wno-nullability-completeness -Wno-expansion-to-defined -Wno-builtin-headers -Wno-c++11-extensions -Wno-writable-strings -I ../android/atsc3_core/libpcr/include -I ../op
_test.c \
atsc3_sl_tlv_denod_type.o \
libatsc3.o \
-L/usr/local/lib -lz -ln -lthread -lpcap -lpcr2-8 -lstdc++ \
-L ../openssl/builddir/lib -lcrypto -lssl \
-o test_sl/atsc3_sl_tlv_parser_to_alp_reflector_test
cc -Wno-nullability-completeness -Wno-expansion-to-defined -Wno-builtin-headers -Wno-c++11-extensions -Wno-writable-strings -I ../android/atsc3_core/libpcr/include -I ../op
_metrics.c \
atsc3_sl_tlv_denod_type.o \
libatsc3.o \
-L/usr/local/lib -lz -ln -lthread -lpcap -lpcr2-8 -lstdc++ \
-L ../openssl/builddir/lib -lcrypto -lssl \
-o test_sl/atsc3_sl_tlv_parser_to_alp_test_with_metrics
uis@luis-VirtualBox:~/Escritorio/tesis/libatsc3/srcs
```

Una vez instalada la biblioteca libatsc3, se procede a abrir la carpeta *tools* que se encuentra dentro de la carpeta *src* y verificar que las herramientas de análisis se encuentren habilitadas para su uso, como se muestra en la figura 38.

Figura 38

Herramienta de análisis para flujos de muestra ATSC 3.0.

```

luis@luis-VirtualBox:~/Escritorio/tesis/libatsc3/src/tools$ ls -l
total 84764
-rwxr-xr-x 1 root root 6629848 jun  5 17:06 atsc3_alc_listener_mde_writer
-rwxr-xr-x 1 root root 6806872 jun  5 17:06 atsc3_alc_listener_mde_writer_asan
-rw-rw-r-- 1 luis luis  15826 jun  2 21:53 atsc3_alc_listener_mde_writer.cpp
drwxrwxr-x 3 luis luis   4096 jun  2 21:53 atsc3_alc_listener_mde_writer.dSYM
-rwxr-xr-x 1 root root 6652712 jun  5 17:06 atsc3_alc_listener_mde_writer_from_stltp_pcap
-rwxr-xr-x 1 root root 6830208 jun  5 17:06 atsc3_alc_listener_mde_writer_from_stltp_pcap_asan
-rw-rw-r-- 1 luis luis  15761 jun  2 21:53 atsc3_alc_listener_mde_writer_from_stltp_pcap.cpp
-rwxr-xr-x 1 root root 6891536 jun  5 17:06 atsc3_listener_metrics_ncurses
-rw-rw-r-- 1 luis luis  21841 jun  2 21:53 atsc3_listener_metrics_ncurses.cpp
drwxrwxr-x 3 luis luis   4096 jun  2 21:53 atsc3_listener_metrics_ncurses.dSYM
-rwxr-xr-x 1 root root 6905024 jun  5 17:06 atsc3_listener_metrics_ncurses_httpd_isobnff
-rw-rw-r-- 1 luis luis  32656 jun  2 21:53 atsc3_listener_metrics_ncurses_httpd_isobnff.cpp
-rwxr-xr-x 1 root root 6798048 jun  5 17:06 atsc3_mmt_listener_to_http_hls_fmp4
-rw-rw-r-- 1 luis luis  40866 jun  2 21:53 atsc3_mmt_listener_to_http_hls_fmp4.cpp
drwxrwxr-x 3 luis luis   4096 jun  2 21:53 atsc3_mmt_listener_to_http_hls_fmp4.dSYM
-rwxr-xr-x 1 root root 6856296 jun  5 17:06 atsc3_mmt_nfu_monitor
-rw-rw-r-- 1 luis luis  14005 jun  2 21:53 atsc3_mmt_nfu_monitor.cpp
-rwxr-xr-x 1 root root 6624192 jun  5 17:06 atsc3_pcap_stltp_preamble_timing_management_validate
-rwxr-xr-x 1 root root 6640344 jun  5 17:06 atsc3_pcap_stltp_preamble_timing_management_validate_asan
-rw-rw-r-- 1 luis luis   6618 jun  2 21:53 atsc3_pcap_stltp_preamble_timing_management_validate.cpp
-rwxr-xr-x 1 root root 9475080 jun  5 17:06 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer
-rw-rw-r-- 1 luis luis  36220 jun  2 21:53 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer.cpp
-rwxr-xr-x 1 root root 9443616 jun  5 17:06 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer_noasan
drwxrwxr-x 2 luis luis   4096 jun  2 21:53 npu-old
-rw-rw-r-- 1 luis luis   3246 jun  2 21:53 PcapSTLTPVirtualPHYTest.cpp
-rwxrwxr-x 1 luis luis    42 jun  2 21:53 run_atsc3_listener_metrics_ncurses
-rwxrwxr-x 1 luis luis   137 jun  2 21:53 run_valgrind

```

Nota. Las herramientas con texto de color verde indican que se encuentran listas para ser ejecutadas.

Herramienta principal de la biblioteca libatsc3 (atsc3_listener_metrics_ncurses)

La herramienta principal de la biblioteca libatsc3 es *atsc3_listener_metrics_ncurses*, posee una interfaz de usuario desarrollada en *ncurses*. La cual es una biblioteca de programación que provee una API (Interfaz de programación de aplicaciones) que permite al programador escribir interfaces en texto y un *muxed* para transmisiones de ROUTE-DASH y MMT (Jason, 2019).

Esta herramienta trabaja con flujos SLT, MMT o ROUTE-DASH capturados en archivos PCAP (paquetes de captura), en los cuales se encuentra toda la recopilación de tráfico de multidifusión especificando su IP, puerto o id de paquete y como resultado se presenta una depuración detallada de la información de estos flujos de datos y el seguimiento selectivo de los flujos de componentes (Jason, 2019).

Datos de salida de la herramienta principal de la biblioteca libatsc3 (atsc3_listener_metrics_ncurses)

La herramienta principal de la biblioteca libatsc3 entrega la interfaz de usuario que se muestra en la figura 39.

Figura 39

Interfaz de la herramienta atsc3_listener_metrics_ncurses.

```

-----Global ATSC 3.0 Statistics-----
Elapsed Duration : 298.61s
LLS Base Table:
LLS total packets received : 1,194
  + parsed good : 597
  + parsed error : 0
  + SLT packets decoded : 1
  + SLT updates processed : 1
MMP total packets received : 129,412
  + type=0d MMP : 0
  + timed : 0
  + non-timed : 0
  + type=0d Signaling : 0
  + type=0f Other : 0
  + parsed errors : 1,576
  + missing packets : 238,482
ALC total packets received : 0
  + parsed good : 0
  + parsed errors : 0
Non-ATSC3 Packets : 0
Total Multicast Packets RX : 134,460

-----Bandwidth Statistics-----
Bandwidth Calculation Interval: 1,00s
LLS : 31 Kbps, 5 pps, Runtine Duration: 297.51s
MMT : 2,172 Kbps, 447 pps, LLS : 146,208 B, 1,190 pkts
ALC : 0 Kbps, 0 pps, MMT : 79,887,583 B, 128,925 pkts
Filtered: 0 Kbps, 0 pps, ALC : 0 B, 0 pkts
Total : 2,338 Kbps, 463 pps, Total : 85,893,467 B, 133,937 pkts

-----Global ATSC 3.0 Statistics-----
Interval Flow : 239.255.1.1:49152, packet_id: 2, NTP range: 47862.337 to 47863.507 (3084314589 - 3084379930)
packet_sequence_numbers : 2194283 to 2194455 (0x0021791b to 0x002179af)
packet_RX count : 14588
mpu_sequence_number: 6241
missing : 47312
mpu_flow_total: 14588
pkt_seq_num gap : 159
signalling_messages total: 0
Lifetime NTP : 46765.154 to 47863.507 (3064830795 to 3084379930) Loss Pct: 300,000000
packet_seq_numbers : 2132516 to 2194415 (0x00208824 to 0x002179af) max sequence gap: 211
Total packets RX : 14588
missing: 47312

Interval Flow : 239.255.1.1:49152, packet_id: 1, NTP range: 47863.336 to 47864.373 (3084340742 - 3084414519)
packet_sequence_numbers : 2503139 to 2503172 (0x002532ab to 0x00253394)
packet_RX count : 33246
mpu_sequence_number: 6763
missing : 34218
mpu_flow_total: 33246
pkt_seq_num gap : 117
signalling_messages total: 0
Lifetime NTP : 46766.406 to 47864.373 (3064840485 to 3084414519) Loss Pct: 100,000000
packet_seq_numbers : 2436189 to 2503172 (0x00252c06 to 0x00253394) max sequence gap: 137
Total packets RX : 33246
missing: 34218

Interval Flow : 239.255.1.1:49152, packet_id: 2, NTP range: 47862.509 to 47863.465 (3084314349 - 3084379653)
packet_sequence_numbers : 2194312 to 2194364 (0x002179ab to 0x002179ac)
packet_RX count : 14588
mpu_sequence_number: 5935
missing : 47312
mpu_flow_total: 14588
pkt_seq_num gap : 158
signalling_messages total: 0
Lifetime NTP : 46765.058 to 47863.465 (3064830823 to 3084379653) Loss Pct: 300,000000
packet_seq_numbers : 2132465 to 2194364 (0x0020897f to 0x002179ac) max sequence gap: 211
Total packets RX : 14588
missing: 47312

Interval Flow : 239.255.1.1:49154, packet_id: 2, NTP range: 47862.525 to 47863.467 (3084310214 - 3084375372)
packet_sequence_numbers : 2194384 to 2194395 (0x002179ab to 0x002179ac)
packet_RX count : 14591
mpu_sequence_number: 6283
missing : 47312
mpu_flow_total: 14591
pkt_seq_num gap : 158
signalling_messages total: 0
Lifetime NTP : 46765.133 to 47863.467 (3064826405 to 3084375372) Loss Pct: 300,000000
packet_seq_numbers : 2132415 to 2194395 (0x002089af to 0x002179ac) max sequence gap: 159
Total packets RX : 14591
missing: 47311

Interval Flow : 239.255.1.1:49152, packet_id: 1, NTP range: 47863.282 to 47864.319 (3084312008 - 3084396990)
packet_sequence_numbers : 2392289 to 2393481 (0x0025225e to 0x00253393)
packet_RX count : 12445
mpu_sequence_number: 6239
missing : 33938
mpu_flow_total: 33940
pkt_seq_num gap : 155
signalling_messages total: 0
Lifetime NTP : 46765.281 to 47864.319 (3064840720 to 3084396990) Loss Pct: 100,000000
packet_seq_numbers : 2436629 to 2503481 (0x00252b0d to 0x00253393) max sequence gap: 148
Total packets RX : 12445
missing: 33938

Interval Flow : 239.255.1.1:49156, packet_id: 2, NTP range: 47863.083 to 47864.137 (3084321328 - 3084387282)
packet_sequence_numbers : 2194250 to 2194442 (0x002179a9 to 0x002179c1) packet_RX count : 14590
mpu_seq
  
```

La interfaz presenta dos sectores principales, en los cuales se muestra información relevante acerca del flujo ATSC 3.0, como lo es la información de señalización de bajos nivel (LLS), información de la tabla de lista de servicios (SLT), información de metadatos y de los protocolos de transporte utilizado en el flujo ATSC 3.0 (MMTP, ROUTE-DASH), estos sectores están identificados con los siguientes nombres:

- Global ATSC 3.0 Statistics
- Flow ATSC 3.0 Statics

A continuación, se describe cada uno de los campos presentados por la interfaz.

Global ATSC 3.0 Statistics

Aquí se presenta la información elemental del flujo de muestra ATSC 3.0 como es la información relacionada con la señalización de bajo nivel LLS y la información de la tabla de lista de servicios SLT, como se muestra en la figura 40.

Figura 40

Datos de salida de la sección Global ATSC 3.0 Statistics.

```

Global ATSC 3.0 Statistics
Elapsed Duration      : 429,13s
LLS total packets received : 1.680
> parsed good        : 840
> parsed error       : 0
- SLT packets decoded : 1
- SLT updates processed : 1
MMTP total packets received : 182.110
- type=0x0 MPU          : 0
- timed                : 0
- non-timed            : 0
- type=0x1 Signaling   : 0
- type=0x? Other       : 0
> parsed errors        : 5.032
> missing packets     : 363.700
ALC total packets received : 0
> parsed good         : 0
> parsed errors       : 0
Non ATSC3 Packets      : 0
Total Multicast Packets RX : 189.540

LLS Base Table:
lls_table_id      : 1 (0x1 )   group_id      : 1 (0x1 )
group_count_minus1 : 0 (0x0 )   table_version : 57 (0x39 )

SLT: Service contains 4 entries:
service_id       : 16          global_service_id : (null)
major_channel_no : 7          minor_channel_no  : 1
service_category : 1, linear av slt_svc_seq_num   : 0
short_service_name : STRWJLA
broadcast_svc_signaling
sls_protocol      : 2, MMTP
sls_destination_ip_address : 239.255.1.1:49152
sls_source_ip_address : (null)

service_id       : 20          global_service_id : (null)
major_channel_no : 7          minor_channel_no  : 4
service_category : 1, linear av slt_svc_seq_num   : 0
short_service_name : STRMov
broadcast_svc_signaling
sls_protocol      : 2, MMTP
sls_destination_ip_address : 239.255.1.1:49154
sls_source_ip_address : (null)

service_id       : 21          global_service_id : (null)
major_channel_no : 7          minor_channel_no  : 3
service_category : 1, linear av slt_svc_seq_num   : 0
short_service_name : CometTV
broadcast_svc_signaling
sls_protocol      : 2, MMTP
sls_destination_ip_address : 239.255.1.1:49155
sls_source_ip_address : (null)

service_id       : 22          global_service_id : (null)
major_channel_no : 88         minor_channel_no  : 8
service_category : 1, linear av slt_svc_seq_num   : 0
short_service_name : Newsy
broadcast_svc_signaling sls_protocol      : 2, MMTP sls_destina

```

En la parte izquierda de la figura 40, se presenta información general sobre el flujo de muestra ATSC 3.0 como el tiempo de ejecución de la herramienta o el número de paquetes que contienen información de señalización de bajo nivel LLS y el número de paquetes del protocolo utilizado para transportar la muestra ATSC 3.0 (MMTP – ROUTE-DASH), en la parte izquierda se presenta información acerca de información de señalización de bajo nivel LLS y de la tabla de lista de servicios SLT, todas las salidas mencionadas en la figura 40 se describen a continuación.

- ***Elapsed Duration.*** Representa el tiempo de análisis transcurrido, esto permite identificar el tiempo que nuestra herramienta permaneció activada.
- ***LLS total packets received.*** Representa cuantos paquetes tipo LLS están ubicados en la muestra.
- ***MMTP total packets received.*** Permite identificar cuantos paquetes pertenecientes al protocolo de transporte MMT están en la muestra.
- ***ALC total packets received.*** Permite identificar cuantos paquetes de codificación en capas asíncronas están en la muestra.
- ***Non ATSC3 Packets.*** Permite identificar cuantos paquetes que no pertenecen al estándar ATSC 3.0 están en la muestra.

LLS Base table

En esta sección se presentan información base sobre la información de señalización de bajo nivel LLS.

- ***Lls_table_id.*** Valor de 8 bits que permite identificar los servicios entregados por la industria.
- ***Group_count_minus1.*** Valor de 8 bits que permite identificar el número total de diferentes grupos de tablas LLS que se encuentran en el flujo de paquetes ALP transportado por el PLP correspondiente.
- ***Group_id.*** Valor de 8 bits único dentro del flujo ATSC 3.0 que permite la asociar la instancia (estación de televisión).
- ***Table_version.*** Esta salida de información hace referencia al campo *LLS_table_version* que se encuentra en la información de señalización de bajo nivel LLS, su valor es de 8 bits que se incrementará en uno, siempre que los datos de la tabla se identifican mediante una combinación de cambios *LLS_table_id* y *LLS_group_id*. Cuando el valor llega a 0xFF, el valor se establecerá en 0x00.

SLT

En esta sección se presenta información acerca de la tabla de lista de servicios, los atributos se refieren a los descritos en la tabla 11.

- **service_id.** Número que describe el servicio que se está emitiendo dentro del área de difusión.
- **major_channel_no.** Presenta el número del canal principal del servicio que se está distribuyendo.
- **service_category.** Categoría de servicio, codificada según la Tabla 12.
- **short_service_name.** Nombre corto que describe el servicio.
- **global_service_id.** Un URI único a nivel mundial que ha infringido el servicio ATSC 3.0. Este atributo no está presente para los servicios de datos ESG, EAS y DRM.
- **minor_channel_no.** Presentan el número del canal menor del servicio que se está distribuyendo.
- **slt_svc_seq_num.** Versión de la información del servicio SLT para este servicio
- **broadcast_svc_signaling.** Ubicación, protocolo, dirección, información de identificación para la señalización de transmisión, este atributo será único por servicio, es decir su valor no podrá repetirse en otro servicio en una misma transmisión.
- **sls_protocol.** Protocolo utilizado para entregar la señalización de la capa de servicio para este Servicio, codificado en la tabla 12
- **sls_destination_ip_address.** Una cadena que contiene la dirección de destino IPv4 con puntos de los paquetes que transportan datos SLS de difusión para este servicio.

- **sls_source_ip_address.** Una cadena que contiene la dirección de origen IPv4 con puntos de los paquetes que transportan datos SLS de difusión para este servicio.

Flow ATSC 3.0 Statistics

Esta sección presenta la información propia de los protocolos ROUTE-DASH o MMTP que se esté usando en el flujo de muestra ATSC 3.0 que se esté analizando, como se presenta en la figura 41.

Figura 41

Presentación de información propia del flujo ATSC 3.0.

```

Flow ATSC 3.0 Statistics
Interval Flow      : 239.255.1.1:49152, packet_id: 2, NTP range: 46828.009 to 46829.112 (3068962822 - 3069029033)
packet_sequence_numbers : 2145577 to 2145783 (0x0020bd29 to 0x0020bdf7)
packet RX count      : 1542      npu_sequence_number: 6007
missing              : 5081      npu timed_total: 1542
pkt_seq num gap     : 159        signalling_messages total: 0
Lifetime NTP        : 46797.558 to 46829.112 (3066930507 to 3069029033) Loss Pct: 300,000000
packet_seq_numbers  : 2139161 to 2145783 (0x0020a419 to 0x0020bdf7) max sequence gap: 211
Total packets RX    : 1542      missing: 5081

Interval Flow      : 239.255.1.1:49154, packet_id: 2, NTP range: 46828.638 to 46829.052 (3068958354 - 3069024343)
packet_sequence_numbers : 2145478 to 2145683 (0x0020bcc6 to 0x0020bd93)
packet RX count      : 1541      npu_sequence_number: 6049
missing              : 5082      npu timed_total: 1541
pkt_seq num gap     : 159        signalling_messages total: 0
Lifetime NTP        : 46797.369 to 46829.052 (3066924974 to 3069024343) Loss Pct: 300,000000
packet_seq_numbers  : 2139061 to 2145683 (0x0020a3b5 to 0x0020bd93) max sequence gap: 159
Total packets RX    : 1541      missing: 5082

Interval Flow      : 239.255.1.1:49156, packet_id: 1, NTP range: 46828.481 to 46829.536 (3068931555 - 3068997452)
packet_sequence_numbers : 2449821 to 2450022 (0x0025619d to 0x00256266)
packet RX count      : 3488      npu_sequence_number: 6028
missing              : 3395      npu timed_total: 3488
pkt_seq num gap     : 106        signalling_messages total: 0
Lifetime NTP        : 46798.449 to 46829.536 (3066965263 to 3068997452) Loss Pct: 0,000000
packet_seq_numbers  : 2443140 to 2450022 (0x00254784 to 0x00256266) max sequence gap: 118
Total packets RX    : 3488      missing: 3395

Interval Flow      : 239.255.1.1:49156, packet_id: 2, NTP range: 46828.479 to 46828.480 (3068970192 - 3068970201)
packet_sequence_numbers : 2145624 to 2145671 (0x0020bd58 to 0x0020bd87)
packet RX count      : 1543      npu_sequence_number: 6030
missing              : 4922      npu timed_total: 1543
pkt_seq num gap     : 0          signalling_messages total: 0
Lifetime NTP        : 46797.177 to 46828.480 (3066936599 to 3068970201) Loss Pct: 300,000000
packet_seq_numbers  : 2139207 to 2145671 (0x0020a447 to 0x0020bd87) max sequence gap: 159
Total packets RX    : 1543      missing: 4922

Interval Flow      : 239.255.1.1:49152, packet_id: 1, NTP range: 46827.158 to 46828.281 (3068915443 - 3068981784)
packet_sequence_numbers : 2449745 to 2449947 (0x00256151 to 0x0025621b)
packet RX count      : 3267      npu_sequence_number: 6004
missing              : 3623      npu timed_total: 3267
pkt_seq num gap     : 112        signalling_messages total: 0
Lifetime NTP        : 46797.550 to 46828.281 (3066947633 to 3068981784) Loss Pct: 100,000000
packet_seq_numbers  : 2443058 to 2449947 (0x00254732 to 0x0025621b) max sequence gap: 138
Total packets RX    : 3267      missing: 3623

Interval Flow      : 239.255.1.1:49155, packet_id: 2, NTP range: 46828.609 to 46829.031 (3068962456 - 3069028499)
packet_sequence_numbers : 2145526 to 2145731 (0x0020bcf6 to 0x0020bdc3) packet RX count : 1542      npu_seq#

```

Nota. La información presentada en esta sección está enfocada a el protocolo MMTP hasta la última versión de la biblioteca instalada aún no se encontraba estructurada esta tabla para el protocolo ROUTE-DASH.

Como se muestra en la figura 41 esta sección permite conocer acerca de información propia del protocolo actual que se esté analizando, es así que para el protocolo MMTP, la herramienta de análisis presenta lo siguientes atributos (Jason, 2019):

- Service_id seleccionado
- Cuadro init / moov (estable)
- MOOF (MPU)
- mpu_sequence_number
- Datos de muestra de MDAT por cada packet_id
- Mensaje de señalización MMT para resolución MPT packet_id

Flujos de datos muestra ATSC 3.0 en formato PCAP de transmisiones ATSC 3.0 MMT y ROUTE-DASH en el mercado

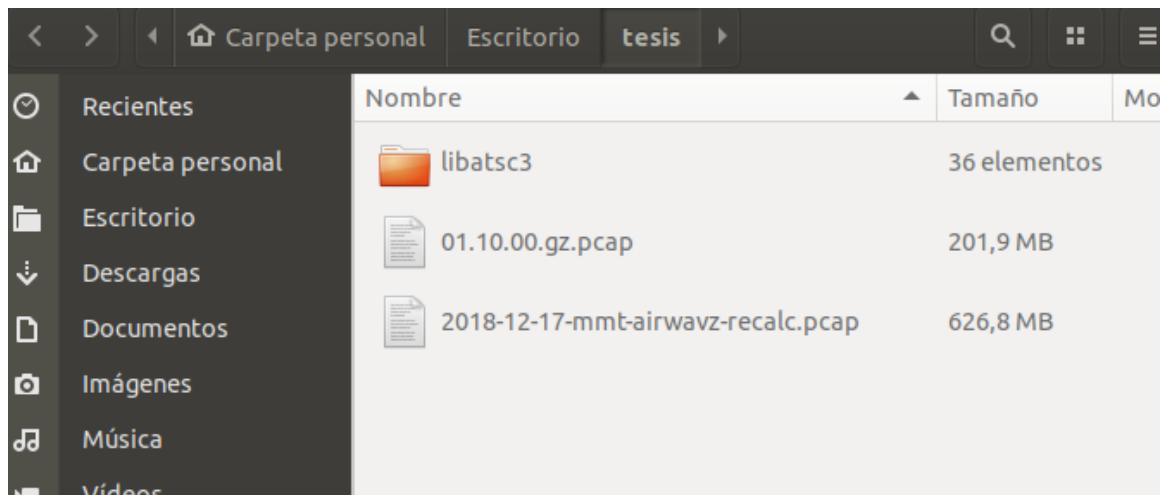
Jason Justman pone a disposición diferentes flujos de datos ATSC 3.0 capturados en formato PCAP de transmisiones reales que se han realizado en el mercado, estás muestra se encuentran alojadas en la plataforma GitHub, los cuales pueden ser descargados y reproducidos fácilmente. Los flujos de datos ATSC 3.0 antes mencionados se encuentran en el siguiente enlace:

<https://github.com/jjustman/atsc-3.0-mmt-pcaps>

Es recomendable que las descargas de estos flujos de muestra sean de forma manual e individual por cada muestra que se requiera analizar, sin utilizar el comando *git clone*, ya que al realizar la clonación de todo el repositorio donde se encuentran los flujos de muestra ATSC 3.0, existe una pérdida de información lo cual no permite su correcta reproducción, además es recomendable que los flujos de muestra a analizar se encuentren en la misma carpeta donde hemos instalado la biblioteca libatsc3 como se muestra en la figura 42.

Figura 42

Flujos de muestra ATSC 3.0 para su análisis.



Para este trabajo se utiliza dos archivos una tanto para el protocolo MMT como para el protocolo ROUTE-DASH, las muestras son las siguientes:

- *2018-12-17-mmt-airwavz-recalp.pcap*
- *01.10.00.gz.pcap*

El primer archivo contiene la información de una transmisión de prueba generada para el testeo del protocolo MMT, realizado en el valle de Hunt, Estados Unidos, el tamaño del archivo es de 626 MB (Jason, 2019).

El segundo archivo contiene la información de una transmisión generada en el mercado en la ciudad de la Vegas, Estados Unidos, el tamaño del archivo es de 201 MB (Jason, 2019).

Reproducción de los flujos de datos muestra ATSC 3.0

Los flujos de muestras ATSC 3.0 que se utilizan se encuentran en el formato PCAP, para su reproducción se utiliza *tcpreplay*. El cual es un conjunto de herramientas de código abierto gratuitas que permiten la edición y reproducción de tráfico de red capturado previamente (paquetes PCAP) (Tcpreplay, s.f.).

Para instalar tcpdump se lo puede realizar con los siguientes comandos:

```
sudo apt-get update
sudo apt-get install tcpdump
```

Una vez instalado tcpdump, se procede a la reproducción de los flujos de muestras, lo que se realiza con el siguiente comando.

```
tcpdump -i name pcap_file
```

donde:

-i: Interfaz de salida de tráfico cliente a servidor / rx / primario.

name: nombre de la tarjeta de interfaz de red, para conocer el nombre de la tarjeta de interfaz de red que cuenta nuestro computador se puede utilizar el comando `sudo ifconfig`, como se muestra en la figura 43.

Figura 43

Interfaces de tarjeta de red disponibles.

```
luis@luis-VirtualBox:~$ sudo ifconfig
[sudo] contraseña para luis:
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::5c1a:8cfb:a00:b984 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:cc:22:70 txqueuelen 1000 (Ethernet)
RX packets 409552 bytes 590011421 (590.0 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 78380 bytes 6934846 (6.9 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Bucle local)
RX packets 1539 bytes 120042 (120.0 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1539 bytes 120042 (120.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Nota. En este caso el nombre de la tarjeta de interfaz de red es `enp0s3`

- `pcap_file`: archivo pcap que se requiere reproducir.

Los comandos para la reproducción los flujos de muestras ATSC 3.0 obtenidos son:

```
sudo tcpreplay -i enp0s3 2018-12-17-mmt-airwavz-recalp.pcap
sudo tcpreplay -i enp0s3 01.10.00.gz.pcap
```

Al finalizar la reproducción del flujo de muestra ATSC 3.0 se obtendrá la información presentada en las figuras 44 y 45.

Figura 44

Finalización de la reproducción del flujo ATSC 3.0 MMTP.

```
luis@luis-VirtualBox:~/Descargas$ sudo tcpreplay -i enp0s3 2018-12-17-mmt-airwavz-recalc.pcap
[sudo] contraseña para luis:
Actual: 518275 packets (618489025 bytes) sent in 530.19 seconds
Rated: 1166537.8 Bps, 9.33 Mbps, 977.52 pps
Flows: 211 flows, 0.39 fps, 518244 flow packets, 31 non-flow
Statistics for network device: enp0s3
  Successful packets:      518275
  Failed packets:         0
  Truncated packets:      0
  Retried packets (ENOBUFS): 0
  Retried packets (EAGAIN): 0
```

Figura 45

Finalización de la reproducción del flujo ATSC 3.0 ROUTE-DASH.

```
luis@luis-VirtualBox:~/Escritorio/tesis$ sudo tcpreplay -i enp0s3 01.10.00.gz.pcap
[sudo] contraseña para luis:
Actual: 140350 packets (199693045 bytes) sent in 231.70 seconds
Rated: 861846.3 Bps, 6.89 Mbps, 605.73 pps
Flows: 33 flows, 0.14 fps, 140338 flow packets, 12 non-flow
Statistics for network device: enp0s3
  Successful packets:      140350
  Failed packets:         0
  Truncated packets:      0
  Retried packets (ENOBUFS): 0
```

Análisis de los flujos de muestra ATSC 3.0 mediante la herramienta libatsc3_listener_metrics ncurses

En esta sección se explicará el proceso a detalle para ejecutar la herramienta *libatsc3_listener_metrics ncurses* de manera correcta, para lo cual como primer paso se procede a abrir dos terminales.

En uno de los terminales se debe ingresar a la carpeta tools, y se despliega las herramientas como se muestra en la figura 46, para esto se utiliza los comandos:

```
cd Escritorio/tesis/libatsc3/src/tools
ls -l
```

Figura 46

Herramientas de análisis proporcionada por la biblioteca libatsc3.

```
Luis@Luis-VirtualBox:~$ cd Escritorio/tesis/libatsc3/src/tools/
Luis@Luis-VirtualBox:~/Escritorio/tesis/libatsc3/src/tools$ ls -l
total 84764
-rwxr-xr-x 1 root root 6629848 jun  5 17:06 atsc3_alc_listener_mde_writer
-rwxr-xr-x 1 root root 6806872 jun  5 17:06 atsc3_alc_listener_mde_writer_asan
-rw-rw-r-- 1 luis luis  15826 jun  2 21:53 atsc3_alc_listener_mde_writer.cpp
drwxrwxr-x 3 luis luis   4096 jun  2 21:53 atsc3_alc_listener_mde_writer.dSYM
-rwxr-xr-x 1 root root 6652712 jun  5 17:06 atsc3_alc_listener_mde_writer_from_stltp_pcap
-rwxr-xr-x 1 root root 6830208 jun  5 17:06 atsc3_alc_listener_mde_writer_from_stltp_pcap_asan
-rw-rw-r-- 1 luis luis  15761 jun  2 21:53 atsc3_alc_listener_mde_writer_from_stltp_pcap.cpp
-rwxr-xr-x 1 root root 6891536 jun  5 17:06 atsc3_listener_metrics_ncurses
-rw-rw-r-- 1 luis luis  21841 jun  2 21:53 atsc3_listener_metrics_ncurses.cpp
drwxrwxr-x 3 luis luis   4096 jun  2 21:53 atsc3_listener_metrics_ncurses.dSYM
-rwxr-xr-x 1 root root 6905024 jun  5 17:06 atsc3_listener_metrics_ncurses_httpd_isobmff
-rw-rw-r-- 1 luis luis  32656 jun  2 21:53 atsc3_listener_metrics_ncurses_httpd_isobmff.cpp
-rwxr-xr-x 1 root root 6798048 jun  5 17:06 atsc3_mmt_listener_to_http_hls_fmp4
-rw-rw-r-- 1 luis luis  40866 jun  2 21:53 atsc3_mmt_listener_to_http_hls_fmp4.cpp
drwxrwxr-x 3 luis luis   4096 jun  2 21:53 atsc3_mmt_listener_to_http_hls_fmp4.dSYM
-rwxr-xr-x 1 root root 6856296 jun  5 17:06 atsc3_mmt_mfu_monitor
-rw-rw-r-- 1 luis luis  14005 jun  2 21:53 atsc3_mmt_mfu_monitor.cpp
-rwxr-xr-x 1 root root 6624192 jun  5 17:06 atsc3_pcap_stltp_preamble_timing_management_validate
-rwxr-xr-x 1 root root 6640344 jun  5 17:06 atsc3_pcap_stltp_preamble_timing_management_validate_asan
-rw-rw-r-- 1 luis luis   6618 jun  2 21:53 atsc3_pcap_stltp_preamble_timing_management_validate.cpp
-rwxr-xr-x 1 root root 9475080 jun  5 17:06 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer
-rw-rw-r-- 1 luis luis  36220 jun  2 21:53 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer.cpp
-rwxr-xr-x 1 root root 9443616 jun  5 17:06 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer_noasan
drwxrwxr-x 2 luis luis   4096 jun  2 21:53 mpu-old
-rw-rw-r-- 1 luis luis   3246 jun  2 21:53 PcapSTLTPVirtualPHYTest.cpp
-rwxrwxr-x 1 luis luis    42 jun  2 21:53 run_atsc3_listener_metrics_ncurses
-rwxrwxr-x 1 luis luis   137 jun  2 21:53 run_valgrind
```

Nota. Se debe validar que el texto de las herramientas se encuentre en color verde y así confirmar que se encuentran listas para ser ejecutadas.

En el segundo terminal se debe ingresar a la carpeta donde se encuentren los flujos de muestra ATSC 3.0 y verificar que los archivos se encuentren en dicha locación como se lo muestra en la figura 47, esto se lo puede realizar con los siguientes comandos:

```
cd Escritorio/tesis
ls -l
```

Figura 47

Flujo a ser analizado.

```

luis@luis-VirtualBox:~$ cd Escritorio/tesis/
luis@luis-VirtualBox:~/Escritorio/tesis$ ls -l
total 809308
-rw-rw-r-- 1 luis luis 201938669 jun  6 10:48 01.10.00.gz.pcap
-rw-rw-r-- 1 luis luis 626781449 may 25 18:45 2018-12-17-mmt-airwavz-recalp.pcap
drwxrwxr-x 35 luis luis    4096 jul  4 17:19 libatsc3
luis@luis-VirtualBox:~/Escritorio/tesis$

```

Ya que los flujos de muestra utilizan diferentes protocolos de transporte, su análisis se lo realizará de manera individual empezando con el flujo *2018-12-17-mmt-airwavz-recalp.pcap* correspondiente al protocolo MMT y posteriormente se trabajará con el flujo *01.10.00.gz.pcap* correspondiente al protocolo ROUTE-DASH.

Análisis de flujos de muestra ATSC 3.0 MMT

Como primer paso se debe situar en el terminal que se encuentre en la carpeta *tesis*, la cual es la que contiene los flujos de muestra ATSC 3.0, se procede a reproducir el flujo de muestra *2018-12-17-mmt-airwavz-recalp.pcap*, lo cual se puede realizar con los siguientes comandos:

```
sudo tcpreplay -i enp0s3 2018-12-17-mmt-airwavz-recalp.pcap
```

Posterior se debe dirigir al terminal que se encuentra la carpeta *tools* y ejecutar la herramienta *atsc3_listener_metrics_ncurses* como se muestra en la figura 48, lo que se puede realizar con el siguiente comando:

```
sudo ./atsc3_listener_metrics_ncurses enp0s3
```

Figura 48

Ejecución de la herramienta `latsc3_listener_metrics_ncurses`.

```
luis@luis-VirtualBox:~/Escritorio/tesis/libatsc3/src/tools$ sudo ./atsc3_listener_metrics_ncurses enp0s3
```

Nota. `enp0s3` es el nombre de la interfaz de tarjeta de red por la cual se está reproduciendo el flujo de muestra `2018-12-17-mmt-airwavz-recalp.pcap`.

La figura 49 permite visualizar los dos terminales listos para realizar el uso de la herramienta `atsc3_listener_metrics_ncurses`

Figura 49

Herramientas `tcpreplay` y `atsc3_listener_metrics_ncurses enp0s3` lista para ser ejecutadas.

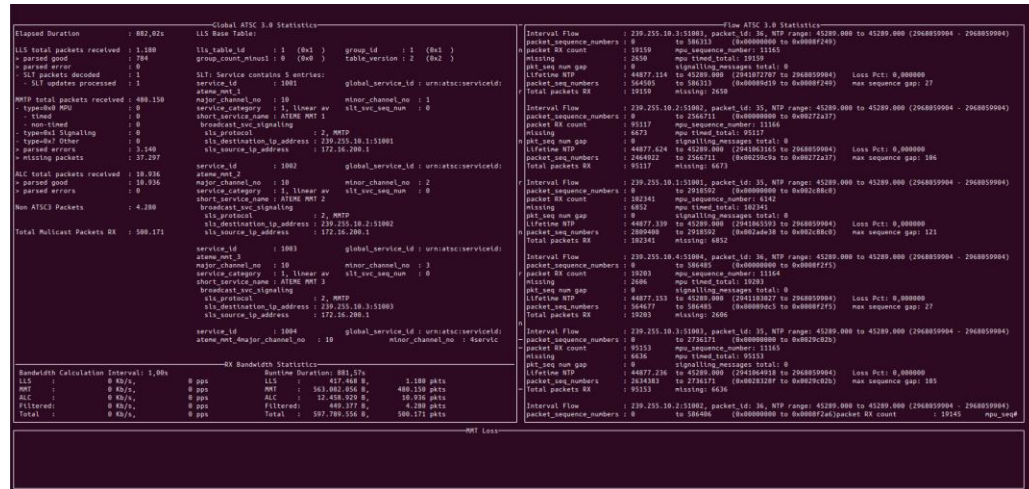
```
luis@luis-VirtualBox:~/Escritorio/tesis/
luis@luis-VirtualBox:~/Escritorio/tesis$ ls -l
total 889388
-rw-rw-r-- 1 luis luis 201938669 Jun  6 10:49 01_10_00_02.pcap
-rw-rw-r-- 1 luis luis 626781449 May 25 18:45 2018-12-17-mt-airwavz-recalp.pcap
luis@luis-VirtualBox:~/Escritorio/tesis$ sudo tcpreplay -i enp0s3 2018-12-17-mt-airwavz-recalp.pcap
[sudo] contraseña para luis:

luis@luis-VirtualBox:~/Escritorio/tesis/libatsc3/src/tools$ ls -l
total 2568352
-rwxr-xr-x 1 root root 6629848 Jun  5 17:06 atsc3_alc_listener_mde_writer
-rwxr-xr-x 1 root root 6886872 Jun  5 17:06 atsc3_alc_listener_mde_writer_asan
-rw-rw-r-- 1 luis luis 15826 Jun  2 21:53 atsc3_alc_listener_mde_writer.cpp
drwxr-xr-x 3 luis luis 4096 Jun  2 21:53 atsc3_alc_listener_mde_writer.dSYM
-rwxr-xr-x 1 root root 6652712 Jun  5 17:06 atsc3_alc_listener_mde_writer_from_stltp_pcap
-rwxr-xr-x 1 root root 6830288 Jun  5 17:06 atsc3_alc_listener_mde_writer_from_stltp_pcap_asan
-rw-rw-r-- 1 luis luis 15761 Jun  2 21:53 atsc3_alc_listener_mde_writer_from_stltp_pcap.cpp
-rwxr-xr-x 1 root root 6891536 Jun  5 17:06 atsc3_listener_metrics_ncurses
-rw-rw-r-- 1 luis luis 21841 Jun  2 21:53 atsc3_listener_metrics_ncurses.cpp
drwxr-xr-x 3 luis luis 4096 Jun  2 21:53 atsc3_listener_metrics_ncurses.dSYM
-rwxr-xr-x 1 root root 6905024 Jun  5 17:06 atsc3_listener_metrics_ncurses_httpd_isobuff
-rw-rw-r-- 1 luis luis 32636 Jun  2 21:53 atsc3_listener_metrics_ncurses_httpd_isobuff.cpp
-rwxr-xr-x 1 root root 6790948 Jun  5 17:06 atsc3_mmt_listener_to_http_hls_fm4
-rw-rw-r-- 1 luis luis 40866 Jun  2 21:53 atsc3_mmt_listener_to_http_hls_fm4.cpp
drwxr-xr-x 3 luis luis 4096 Jun  2 21:53 atsc3_mmt_listener_to_http_hls_fm4.dSYM
-rwxr-xr-x 1 root root 6856296 Jun  5 17:06 atsc3_mmt_nfu_monitor
-rw-rw-r-- 1 luis luis 14005 Jun  2 21:53 atsc3_mmt_nfu_monitor.cpp
-rwxr-xr-x 1 root root 6624192 Jun  5 17:06 atsc3_pcap_stltp_preamble_timing_management_validate
-rwxr-xr-x 1 root root 6640344 Jun  5 17:06 atsc3_pcap_stltp_preamble_timing_management_validate_asan
-rw-rw-r-- 1 luis luis 6619 Jun  2 21:53 atsc3_pcap_stltp_preamble_timing_management_validate.cpp
-rwxr-xr-x 1 root root 9475080 Jun  5 17:06 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer
-rw-rw-r-- 1 luis luis 36220 Jun  2 21:53 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer.cpp
-rwxr-xr-x 1 root root 9443816 Jun  5 17:06 atsc3_srt_rx_stltp_virtual_phy_alc_listener_writer_noasan
-rw-rw-r-- 1 root root 258706813 Jun  6 16:08 debug_log_1623013072_8297
-rw-rw-r-- 1 root root 314837566 Jun  6 16:21 debug_log_1623013760_4476
-rw-rw-r-- 1 root root 10330732 Jun  6 16:26 debug_log_1623014383_6086
-rw-rw-r-- 1 root root 7897268 Jun  6 16:36 debug_log_1623015224_6021
-rw-rw-r-- 1 root root 317191750 Jun  6 17:10 debug_log_1623016660_9590
-rw-rw-r-- 1 root root 952105869 Jun  6 17:32 debug_log_1623017662_2884
-rw-rw-r-- 1 root root 380445895 Jul  3 19:11 debug_log_1623356259_1416
-rw-rw-r-- 1 root root 382420491 Jul  4 17:36 debug_log_1625436333_3848
drwxr-xr-x 2 luis luis 4096 Jun  2 21:53 npu-old
-rw-rw-r-- 1 luis luis 3246 Jun  2 21:53 PcapSTLTPVirtualPHYTest.cpp
-rwxr-xr-x 1 luis luis 42 Jun  2 21:53 run_atsc3_listener_metrics_ncurses
-rwxr-xr-x 1 luis luis 137 Jun  2 21:53 run_valgrind
luis@luis-VirtualBox:~/Escritorio/tesis/libatsc3/src/tools$ sudo ./atsc3_listener_metrics_ncurses enp0s3
```

Una vez se ejecute la herramienta `atsc3_listener_metrics_ncurses`, la información del flujo `2018-12-17-mmt-airwavz-recalp.pcap`, se presentará como lo muestra la figura 50.

Figura 50

Informacion del flujo 2018-12-17-mmt-airwavz-recalp.pcap.



Análisis de resultados de la sección Global ATSC 3.0 Statistics

Para el análisis de esta sección, se la dividirá en dos partes:

- **Parte izquierda.** Donde se presenta información respecto al número total de paquetes de los diferentes protocolos que se encuentran en la muestra ATSC 3.0 2018-12-17-mmt-airwavz-recalp.pcap.
- **Parte derecha.** donde se presenta información de señalización de bajo nivel (LLS) y de la tabla de lista de servicios (SLT).

Como lo muestra la figura 51.

Figura 51

Sección Global ATSC 3.0 Statistics dividida por partes.

```

Global ATSC 3.0 Statistics
-----
Elapsed Duration      : 751,18s

LLS total packets received : 1,220
> parsed good           : 811
> parsed error          : 0
- SLT packets decoded   : 1
  - SLT updates processed : 1

MMTP total packets received : 496,502
- type=0x0 MPU           : 0
  - timed                : 0
  - non-timed            : 0
- type=0x1 Signaling     : 0
- type=0x? Other         : 0
> parsed errors         : 3,242
> missing packets       : 37,854

ALC total packets received : 11,308
> parsed good           : 11,308
> parsed errors         : 0

Non ATSC3 Packets       : 4,539

Total Multicast Packets RX : 517,491

-----
LLS Base Table:
lls_table_id      : 1 (0x1 )   group_id      : 1 (0x1 )
group_count_minus1 : 0 (0x0 )   table_version : 2 (0x2 )

SLT: Service contains 5 entries:
service_id       : 1001      global_service_id : urn:atsc:serviceid:
ateme_mmt_1
major_channel_no : 10      minor_channel_no : 1
service_category : 1, linear av  slt_svc_seq_num  : 0
short_service_name : ATEME MMT 1
broadcast_svc_signaling
sls_protocol     : 2, MMTP
sls_destination_ip_address : 239.255.10.1:51001
sls_source_ip_address  : 172.16.200.1

service_id       : 1002      global_service_id : urn:atsc:serviceid:
ateme_mmt_2
major_channel_no : 10      minor_channel_no : 2
service_category : 1, linear av  slt_svc_seq_num  : 0
short_service_name : ATEME MMT 2
broadcast_svc_signaling
sls_protocol     : 2, MMTP
sls_destination_ip_address : 239.255.10.2:51002
sls_source_ip_address  : 172.16.200.1

service_id       : 1003      global_service_id : urn:atsc:serviceid:
ateme_mmt_3
major_channel_no : 10      minor_channel_no : 3
service_category : 1, linear av  slt_svc_seq_num  : 0
short_service_name : ATEME MMT 3
broadcast_svc_signaling
sls_protocol     : 2, MMTP
sls_destination_ip_address : 239.255.10.3:51003
sls_source_ip_address  : 172.16.200.1

service_id       : 1004      global_service_id : urn:atsc:serviceid:
ateme_mmt_4major_channel_no : 10      minor_channel_no : 4servic
  
```



IZQUIERDA



DERECHA

Nota. Se realiza esta división para desarrollar un mejor análisis de la sección *Global ATSC 3.0*

Parte Izquierda flujo 2018-12-17-mmt-airwavz-recalp.pcap

La duración de análisis de la muestra ATSC 3.0 por la herramienta es de 751,18 segundos, de los cuales obtiene 1220 paquetes de señalización de bajo nivel LLS, 811 paquetes de estos se analizan correctamente obteniendo la decodificación de una tabla de listas de servicios SLT.

La muestra tiene 496502 paquetes pertenecientes al protocolo MMTP en donde un aproximado de 40000 paquetes no se pueden analizar o se pierden.

Existen 11388 paquetes de codificación en capas asíncronas, los cuales en su totalidad fueron analizados y 4539 paquetes de información que no pertenecen al estándar ATSC 3.0.

Todos estos paquetes de información se obtuvieron de una muestra de 517491 paquetes de multidifusión, esto se lo puede observar en la figura 52.

Figura 52

Parte izquierda de la sección Global ATSC 3.0 Statistics.

```

Elapsed Duration           : 751.18s
LLS total packets received : 1.220
> parsed good              : 811
> parsed error             : 0
- SLT packets decoded      : 1
  - SLT updates processed  : 1
MMTP total packets received : 496.502
- type=0x0 MPU             : 0
  - timed                  : 0
  - non-timed              : 0
- type=0x1 Signaling       : 0
- type=0x? Other           : 0
> parsed errors            : 3.242
> missing packets         : 37.854
ALC total packets received : 11.308
> parsed good              : 11.308
> parsed errors            : 0
Non ATSC3 Packets         : 4.539
Total Multicast Packets RX : 517.491

```

Parte derecha flujo 2018-12-17-mmt-airwavz-recalp.pcap

La parte derecha empieza mostrando atributos base acerca de la información de señalización de bajo nivel LLS, la cual se detalla en la tabla 16.

Tabla 17

Información base LLS de la muestra 2018-12-17-mmt-airwavz-recalp.pcap.

Atributo	Valor	Descripción
LLS_table_id	1 (0x01)	El flujo ATSC 3.0 entrega tablas SLT
group_count_minus1	0 (0x0)	En la información del LLS se lleva un solo tipo de tabla
group_id	1 (0x1)	Se asocia LLS_table () con la tabla del LLS_group_id
table_version	2(0x2)	Existen dos cambios en las combinaciones de las tablas <i>LLS_table_id</i> y <i>LLS_group_id</i>

Nota. Las descripciones de los valores, se relacionan a las definidas en la información de señalización de bajo nivel LLS definida para el estándar ATSC 3.0

A continuación, se presenta la información de la tabla de lista de servicios SLT, el flujo de muestra *2018-12-17-mmt-airwavz-recalp.pcap* tiene 5 servicios, como se puede observar en la figura 53.

Figura 53

Servicios del flujo de muestra 2018-12-17-mmt-airwavz-recalp.pcap.

```

SLT: Service contains 5 entries:
service_id      : 1001      global_service_id : urn:atsc:serviceid:ateme_wmt_1
major_channel_no : 10      minor_channel_no  : 1
service_category : 1, linear av  slt_svc_seq_num  : 0
short_service_name : ATEME WMT 1
broadcast_svc_signaling
sls_protocol    : 2, MMTp
sls_destination_ip_address : 239.255.10.1:51001
sls_source_ip_address  : 172.10.200.1

service_id      : 1002      global_service_id : urn:atsc:serviceid:ateme_wmt_2
major_channel_no : 10      minor_channel_no  : 2
service_category : 1, linear av  slt_svc_seq_num  : 0
short_service_name : ATEME WMT 2
broadcast_svc_signaling
sls_protocol    : 2, MMTp
sls_destination_ip_address : 239.255.10.2:51002
sls_source_ip_address  : 172.10.200.1

service_id      : 1003      global_service_id : urn:atsc:serviceid:ateme_wmt_3
major_channel_no : 10      minor_channel_no  : 3
service_category : 1, linear av  slt_svc_seq_num  : 0
short_service_name : ATEME WMT 3
broadcast_svc_signaling
sls_protocol    : 2, MMTp
sls_destination_ip_address : 239.255.10.3:51003
sls_source_ip_address  : 172.10.200.1

service_id      : 1004      global_service_id : urn:atsc:serviceid:ateme_wmt_4
major_channel_no : 10      minor_channel_no  : 4
service_category : 1, linear av  slt_svc_seq_num  : 0
short_service_name : ATEME WMT 4
broadcast_svc_signaling
sls_protocol    : 2, MMTp
sls_destination_ip_address : 239.255.10.4:51004
sls_source_ip_address  : 172.10.200.1

service_id      : 5009      global_service_id : urn:atsc:serviceid:esg
major_channel_no : 0      minor_channel_no  : 0
service_category : 4, esg service  slt_svc_seq_num  : 0
short_service_name : ESG
broadcast_svc_signaling
sls_protocol    : 1, ROUTE
sls_destination_ip_address : 239.255.20.9:52009
sls_source_ip_address  : 172.10.200.1

```

Los atributos que se entregan por cada servicio se detallan en la tabla 17.

Tabla 18

Informacion SLT de la muestra 2018-12-17-mmt-airwavz-recalp,pcap por servicio.

Primer servicio		
Atributo	Valor	Descripción
Service_id	1001	Número establecido por la estación de tv para identificar el servicio transmitido.
Global_service_id	Urn:atsc::serviceid:ateme_mmt_1	URI establecido para el servicio.
Major_channel_no	10	Canal principal para el servicio.
Minor_channel_no	1	Canal menor para el servicio.
Service_category	1, lineaer av	Categoría de servicio lineal.
Slc_svc_seq_num	0	Versión del servicio SLT.
Short_service_name	ATEME MMT 1	Nombre corto para identificar el servicio.
Sls_protocol	2, MMTP	Protocolo para transporte de datos.
Sls_destination_ip_addresses	239.255.10.1:51001	Dirección IP de destino.
Sls_source_protocol	172.16.200.1	Dirección IP de origen.
Segundo servicio		
Atributo	Valor	Descripción

Global_service_id	Urn:atsc::serviceid:ateme_ mmt_2	URI establecido para el servicio.
Major_channel_no	10	Canal principal para el servicio.
Minor_channel_no	2	Canal menor para el servicio.
Service_category	1, lineaer av	Categoría de servicio lineal.
Slit_svc_seq_num	0	Versión del servicio SLT.
Short_service_name	ATEME MMT 2	Nombre corto para identificar el servicio.
Sls_protocol	2, MMTP	Protocolo para transporte de datos.
Sls_destination_ip_addresses	239.255.10.1:51002	Dirección IP de destino.
Sls_source_protocol	172.16.200.1	Dirección IP de origen.
Tercer servicio		
Atributo	Valor	Descripción
Service_id	1003	Número establecido por la estación para identificar el servicio transmitido.
Global_service_id	Urn:atsc::serviceid:ateme_ mmt_3	URI establecido para el servicio.

Major_channel_no	10	Canal principal para el servicio.
Minor_channel_no	3	Canal menor para el servicio.
Service_category	1, lineær av	Categoría de servicio lineal.
Slit_svc_seq_num	0	Versión del servicio SLT.
Short_service_name	ATEME MMT 3	Nombre corto para identificar el servicio.
Sls_protocol	2, MMTP	Protocolo para transporte de datos.
Sls_destination_ip_address	239.255.10.1:51003	Dirección IP de destino.
Sls_source_ip_address	172.16.200.1	Dirección IP de origen.
Cuarto servicio		
Atributo	Valor	Descripción
Service_id	1004	Número establecido por la estación para identificar el servicio transmitido.
Global_service_id	Urn:atsc::serviceid:ateme_mmt_4	URI establecido para el servicio
Major_channel_no	10	Canal principal para el servicio
Minor_channel_no	4	Canal menor para el servicio
Service_category	1, lineær av	Categoría de servicio lineal
Slit_svc_seq_num	0	Versión del servicio SLT

Short_service_name	ATEME MMT 4	Nombre corto para identificar el servicio
Sls_protocol	2, MMTP	Protocolo para transporte de datos
Sls_destination_ip_addresses	239.255.10.1:51004	Dirección IP de destino
Sls_source_protocol	172.16.200.1	Dirección IP de origen
Quinto servicio		
Atributo	Valor	Descripción
Service_id	1001	Número establecido por la estación para identificar el servicio transmitido.
Global_service_id	Um:atsc::serviceid:esg	URI establecido para el servicio
Major_channel_no	0	Canal principal para el servicio
Minor_channel_no	0	Canal menor para el servicio
Service_category	4, esg service	Categoría de servicio lineal
Slf_svc_seq_num	0	Versión del servicio SLT
Short_service_name	ESG	Nombre corto para identificar el servicio
Sls_protocol	1, ROUTE	Protocolo para transporte de datos
Sls_destination_ip_addresses	239.255.20.9:52009	Dirección IP de destino
Sls_source_protocol	172.16.200.1	Dirección IP de origen

Nota. Los atributos de salida como las descripciones presentadas en la tabla 17, se validan con la información presentada en la tabla de lista de servicios SLT, definido para el sistema ATSC 3.0.

Con ayuda de la biblioteca *libatsc3*, se establece la relevancia de los parámetros tanto de la información de señalización de bajo nivel LLS como de la tabla de lista de servicios SLT utilizados para la transmisión de flujos ATSC 3.0 que contengan servicios o contenidos en los mismos. Como se presenta tanto en la figura 53 como el análisis presentado en la tabla 17, el flujo de muestra se encuentra estructurado con la señalización de estos parámetros.

Análisis de resultados de la sección Flow ATSC 3.0 Statistics

La sección *Flow ATSC 3.0 Statistics* presenta datos por intervalos de flujos definidos en la muestra ATSC 3.0, que se describen para una sesión MMTP que se puede identificar mediante la dirección IP de destino y el número de puerto de destino. Un flujo de paquetes MMTP como lo es la muestra *2018-12-17-mmt-airwavz-recalp.pcap* que se encuentra asociado con los componentes del servicio que transporta, utilizan el *packet_id*, único dentro del alcance de la sesión MMTP principal para su identificación. Las propiedades comunes a todos los flujos de paquetes MMTP y ciertas propiedades de los flujos de paquetes MMTP individuales se presentan en la figura 54.

Figura 54

Información de la sección Flow ATSC 3.0 Statistics del flujo 2018-12-17-mmt-airwavz-recalp.pcap.

```

Flow ATSC 3.0 Statistics
Interval Flow : 239.255.10.1:51001, packet_id: 36, NTP range: 45289.156 to 45289.000 (2968125351 - 2968059904)
packet_sequence_numbers : 0 to 588850 (0x00000000 to 0x0008fc32)
packet RX count : 19808 mpu_sequence_number: 6142
missing : 2711 npu_timed_total: 19808
pkt_seq_num_gap : 0 signalling_messages_total: 0
Lifetime NTP : 44863.000 to 45289.000 (2940141568 to 2968059904) Loss Pct: 0,000000
packet_seq_numbers : 566332 to 588850 (0x0008a43c to 0x0008fc32) max sequence gap: 27
Total packets RX : 19808 missing: 2711

Interval Flow : 239.255.10.3:51003, packet_id: 36, NTP range: 45289.175 to 45289.000 (2968069642 - 2968059904)
packet_sequence_numbers : 0 to 586313 (0x00000000 to 0x0008f249)
packet RX count : 19812 mpu_sequence_number: 11165
missing : 2708 npu_timed_total: 19812
pkt_seq_num_gap : 0 signalling_messages_total: 0
Lifetime NTP : 44863.000 to 45289.000 (2940141568 to 2968059904) Loss Pct: 0,000000
packet_seq_numbers : 563794 to 586313 (0x00089a52 to 0x0008f249) max sequence gap: 27
Total packets RX : 19812 missing: 2708

Interval Flow : 239.255.10.4:51004, packet_id: 35, NTP range: 45289.041 to 45289.000 (2968068766 - 2968059904)
packet_sequence_numbers : 0 to 2736451 (0x00000000 to 0x0029c143)
packet RX count : 98473 mpu_sequence_number: 11164
missing : 6646 npu_timed_total: 98473
pkt_seq_num_gap : 0 signalling_messages_total: 0
Lifetime NTP : 44863.000 to 45289.000 (2940141568 to 2968059904) Loss Pct: 0,000000
packet_seq_numbers : 2631333 to 2736451 (0x002826a5 to 0x0029c143) max sequence gap: 107
Total packets RX : 98473 missing: 6646

Interval Flow : 239.255.10.3:51003, packet_id: 35, NTP range: 45289.037 to 45289.000 (2968115985 - 2968059904)
packet_sequence_numbers : 0 to 2736171 (0x00000000 to 0x0029c02b)
packet RX count : 98402 mpu_sequence_number: 11165
missing : 6716 npu_timed_total: 98402
pkt_seq_num_gap : 0 signalling_messages_total: 0
Lifetime NTP : 44863.000 to 45289.000 (2940141568 to 2968059904) Loss Pct: 0,000000
packet_seq_numbers : 2631054 to 2736171 (0x0028258e to 0x0029c02b) max sequence gap: 105
Total packets RX : 98402 missing: 6716

Interval Flow : 239.255.10.1:51001, packet_id: 35, NTP range: 45289.070 to 45289.000 (2968086138 - 2968059904)
packet_sequence_numbers : 0 to 2918592 (0x00000000 to 0x002c88c0)
packet RX count : 105812 mpu_sequence_number: 6142
missing : 6932 npu_timed_total: 105812
pkt_seq_num_gap : 0 signalling_messages_total: 0
Lifetime NTP : 44863.000 to 45289.000 (2940141568 to 2968059904) Loss Pct: 0,000000
packet_seq_numbers : 2805849 to 2918592 (0x002ad059 to 0x002c88c0) max sequence gap: 121
Total packets RX : 105812 missing: 6932

Interval Flow : 239.255.10.4:51004, packet_id: 36, NTP range: 45289.527 to 45289.000 (2968097719 - 2968059904)
packet_sequence_numbers : 0 to 586485 (0x00000000 to 0x0008f2f5) packet RX count : 19855 mpu_seq#

```

En la tabla 18 se describen los atributos mostrados en la figura 54.

Tabla 19

Descripción de atributos de la sección Flow ATSC 3.0 Statistics.

Atributo	Descripción
Interval Flow	Intervalo de flujo de muestra ATSC 3.0
Packet_id	Identificador de los componentes que se encuentren en el flujo, permitiendo el filtrado de paquetes MMT y sesiones MMT
NTP range	Rango de protocolo de tiempo de red
Packet RX count	Número de paquetes recibidos
Lifetime NTP	Tiempo total del protocolo de tiempo de red
packet_seq_numbers	Numero asignado al paquete
Total packets RX	Número total de paquetes recibidos
mpu_sequence_number	Número asignado al mpu para su identificación

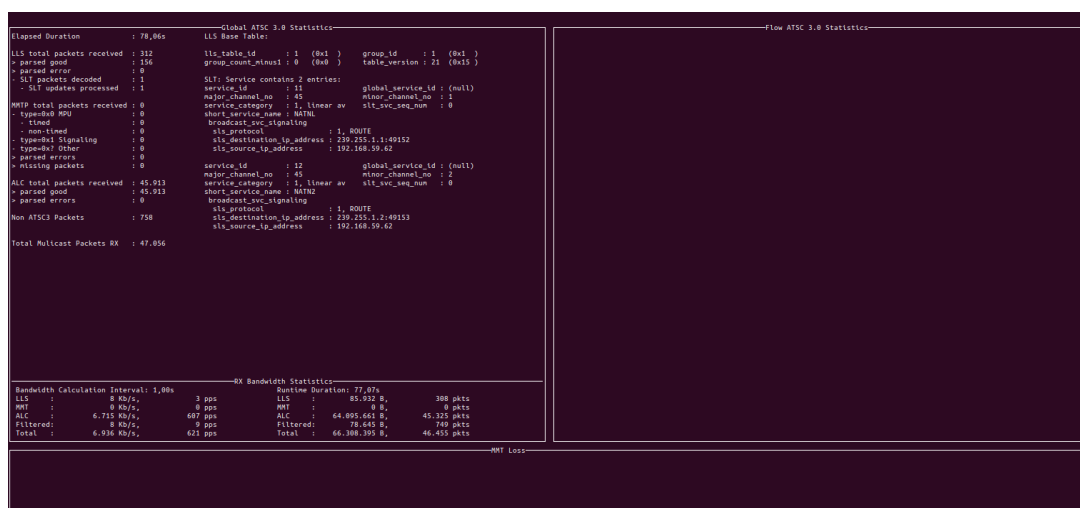
Nota. Los atributos de salida como las descripciones presentadas en la tabla 18, son base para la entrega, adquisición de servicios y contenidos con formato MMT.

Análisis de flujos de muestra ATSC 3.0 ROUTE-DASH

Para el flujo 01.10.00.gz.pcap que es una muestra del protocolo ROUTE/DASH, se realiza el mismo procedimiento descrito en la sección anterior para el flujo de MMT, los resultados obtenidos se muestran en la figura 55.

Figura 55

Informacion del flujo 01.10.00.gz.pcap

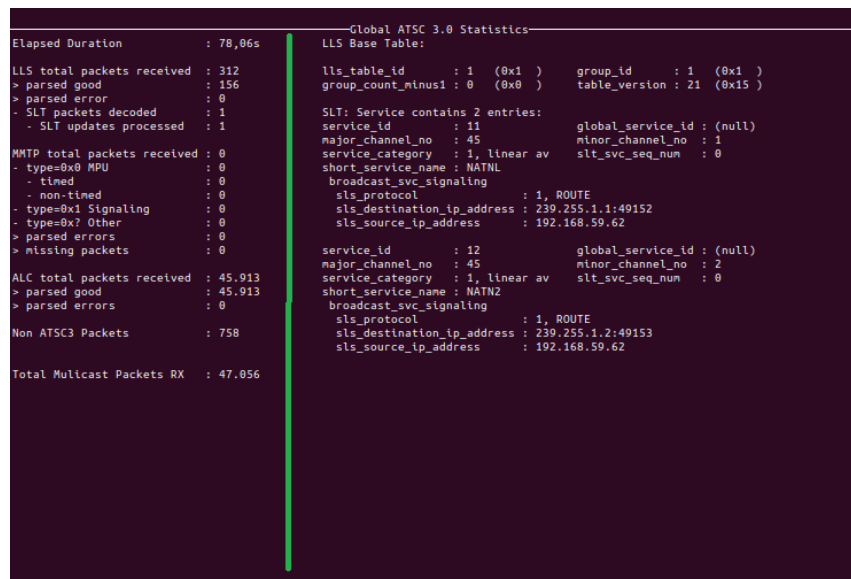


Análisis de resultados de la sección Global ATSC 3.0 Statistics

Para el análisis de esta sección, se la dividirá en dos partes (izquierda, derecha) de la misma manera que para del flujo de muestra MMT, como se muestra en la figura 56.

Figura 56

Sección Global ATSC 3.0 Statistics dividida por partes.



```

Global ATSC 3.0 Statistics
-----
Elapsed Duration      : 78,06s

LLS total packets received : 312
> parsed good           : 156
> parsed error          : 0
- SLT packets decoded    : 1
- SLT updates processed  : 1

MMT total packets received : 0
- type=0x0 MPU           : 0
- timed                  : 0
- non-timed              : 0
- type=0x1 Signaling     : 0
- type=0x7 Other         : 0
> parsed errors         : 0
> missing packets       : 0

ALC total packets received : 45,913
> parsed good           : 45,913
> parsed errors         : 0

Non ATSC3 Packets      : 758


Total Multicast Packets RX : 47,056


LLS Base Table:
lls_table_id      : 1 (0x1 )   group_id      : 1 (0x1 )
group_count_minus1 : 0 (0x0 )   table_version : 21 (0x15 )

SLT: Service contains 2 entries:
service_id        : 11         global_service_id : (null)
major_channel_no  : 45         minor_channel_no  : 1
service_category  : 1, linear av  slt_svc_seq_num   : 0
short_service_name : NATL
broadcast_svc_signaling
  sls_protocol     : 1, ROUTE
  sls_destination_ip_address : 239.255.1.1:49152
  sls_source_ip_address  : 192.168.59.62

service_id        : 12         global_service_id : (null)
major_channel_no  : 45         minor_channel_no  : 2
service_category  : 1, linear av  slt_svc_seq_num   : 0
short_service_name : NATN2
broadcast_svc_signaling
  sls_protocol     : 1, ROUTE
  sls_destination_ip_address : 239.255.1.2:49153
  sls_source_ip_address  : 192.168.59.62

```


IZQUIERDA


DERECHA

Parte Izquierda flujo 01.10.00.gz.pcap

La duración de análisis de la muestra ATSC 3.0 por la herramienta es de 78,06 segundos, de los cuales obtiene 312 paquetes de señalización de bajo nivel LLS, 156 paquetes de estos se analizan correctamente obteniendo la decodificación de una tabla de listas de servicios SLT.

Existen 45913 paquetes de codificación en capas asíncronas, los cuales en su totalidad fueron analizados y 758 paquetes de información que no pertenecen al estándar ATSC 3.0.

Todos estos paquetes de información se obtuvieron de una muestra de 47056 paquetes de multidifusión, esto se lo puede observar en mayor detalle en la figura 57.

Figura 57

Parte izquierda de la sección *Global ATSC 3.0 Statistics*.

```

Elapsed Duration           : 78,06s
LLS total packets received : 312
> parsed good              : 156
> parsed error             : 0
- SLT packets decoded      : 1
  - SLT updates processed  : 1
MMTP total packets received : 0
- type=0x0 MPU             : 0
  - timed                  : 0
  - non-timed              : 0
- type=0x1 Signaling       : 0
- type=0x? Other           : 0
> parsed errors            : 0
> missing packets         : 0
ALC total packets received : 45,913
> parsed good              : 45,913
> parsed errors            : 0
Non ATSC3 Packets         : 758
Total Multicast Packets RX : 47.056

```

Parte derecha flujo 01.10.00.gz.pcap

La parte derecha empieza mostrando atributos base acerca de la información de señalización de bajo nivel LLS, la cual se detalla en la tabla 19.

Tabla 20

Información base LLS de la muestra 01.10.00.gz.pcap.

Atributo	Valor	Descripción
LLS_table_id	1 (0x01)	El flujo ATSC 3.0 entrega tablas SLT
group_count_minus1	0 (0x0)	En la información del LLS se lleva un solo tipo de tabla
group_id	1 (0x1)	Se asocia LLS_table () con la tabla del LLS_group_id
table_version	21(0x15)	Existen 21 cambios en las combinaciones de las tablas <i>LLS_table_id</i> y <i>LLS_group_id</i>

Nota. Las descripciones de los valores, se relacionan a las definidas en la información de señalización de bajo nivel LLS definida para el estándar ATSC 3.0

En la figura 58, se puede observar la información correspondiente a la tabla de lista de servicios SLT, el flujo de muestra 01.10.00.gz.pcap presenta 2 servicios.

Figura 58

Servicios del flujo de muestra 01.10.00.gz.pcap.

```
SLT: Service contains 2 entries:
service_id      : 11                global_service_id : (null)
major_channel_no : 45                minor_channel_no  : 1
service_category : 1, linear av    slt_svc_seq_num   : 0
short_service_name : NATNL
broadcast_svc_signaling
  sls_protocol      : 1, ROUTE
  sls_destination_ip_address : 239.255.1.1:49152
  sls_source_ip_address  : 192.168.59.62

service_id      : 12                global_service_id : (null)
major_channel_no : 45                minor_channel_no  : 2
service_category : 1, linear av    slt_svc_seq_num   : 0
short_service_name : NATN2
broadcast_svc_signaling
  sls_protocol      : 1, ROUTE
  sls_destination_ip_address : 239.255.1.2:49153
  sls_source_ip_address  : 192.168.59.62
```

Los atributos que se entregan por cada servicio se detallan en la tabla 20.

Tabla 21

Informacion SLT de la muestra 01.10.00.gz.pcap.

Primer servicio		
Atributo	Valor	Descripción
Service_id	11	Número establecido por la estación de tv para identificar el servicio transmitido.
Global_service_id	null	URI establecido para el servicio.
Major_channel_no	45	Canal principal para el servicio.
Minor_channel_no	1	Canal menor para el servicio.
Service_category	1, lineaer av	Categoría de servicio lineal.
Slit_svc_seq_num	0	Versión del servicio SLT.

Short_service_name	NATNL	Nombre corto para identificar el servicio.
Sls_protocol	1,ROUTE	Protocolo para transporte de datos.
Sls_destination_ip_address	239.255.1.1:49152	Dirección IP de destino.
Sls_source_protocol	192.168.59.62	Dirección IP de origen.
Segundo servicio		
Atributo	Valor	Descripción
Service_id	22	Número establecido por la estación para identificar el servicio transmitido.
Global_service_id	null	URI establecido para el servicio.
Major_channel_no	45	Canal principal para el servicio.
Minor_channel_no	2	Canal menor para el servicio.
Service_category	1, lineaer av	Categoría de servicio lineal.
Slit_svc_seq_num	0	Versión del servicio SLT.
Short_service_name	NATN2	Nombre corto para identificar el servicio.
Sls_protocol	1, ROUTE	Protocolo para transporte de datos.
Sls_destination_ip_address	239.255.1.2.49153	Dirección IP de destino.
Sls_source_protocol	192.168.59.62	Dirección IP de origen.

Nota. Los atributos de salida como las descripciones presentadas en la tabla 20, se validan con la información presentada en la tabla de atributos de la tabla de lista de servicios SLT, definido para el sistema ATSC 3.0.

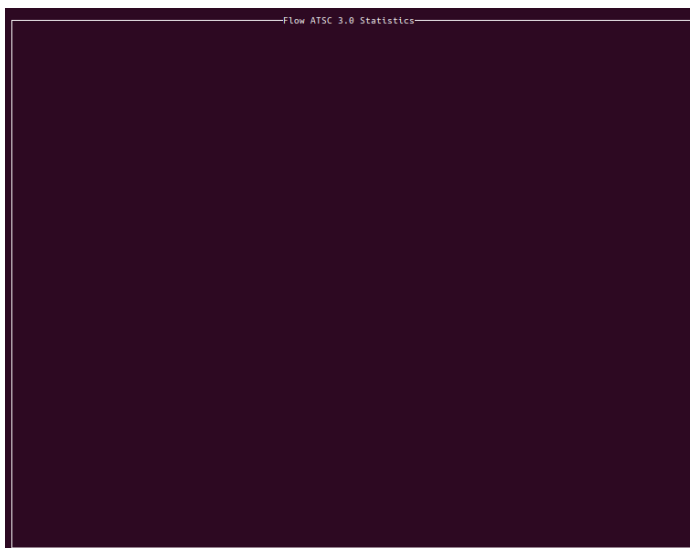
Análisis de resultados de la sección Flow ATSC 3.0 Statistics

Para flujos ATSC 3.0 ROUTE-DASH, la biblioteca *libatsc3*, aún se encuentra estructurando la tabla de información de la sección *Flow ATSC 3.0 Statistics*, por lo que

en la versión actual la biblioteca no presenta ningún resultado como se muestra en la figura 59.

Figura 59

Informacion de la sección Flow ATSC 3.0 Statistics del flujo 01.10.00.gz.pcap.



Capítulo IV

Conclusiones y Recomendaciones

Conclusiones

- El presente trabajo de investigación presento la estructura y los conceptos definidos para la capa de transporte del estándar ATSC 3.0, necesarios para entender las nuevas tecnologías y su correcta interconexión para la realización de esta disertación. Las normas A/330, A/331 y A/324 fueron la base para comprender la forma en que se diseñó el sistema y así conciliar los conocimientos que se necesitan para el desarrollo del proyecto.
- La capa de transporte establecida para el estándar ATSC 3.0 utiliza el protocolo de capa de enlace (ALP) definida en la norma A/330, para la encapsulación de fuentes de datos que pertenezcan a diferentes protocolos, en un solo formato de paquete ALP, utilizando técnicas avanzadas de encapsulación, comprensión y señalización, lo que permite el envío y recepción de información de una manera eficiente entre la capa física y la capa de red, además permite que el sistema sea compatible para trabajar conjuntamente con diferentes tecnologías.
- Se presenta los protocolos MMTP Y ROUTE que utilizan diferentes mecanismos y procesos para la entrega, señalización y transporte de contenido y servicios ATSC 3.0 sobre el protocolo IP hacia receptores ATSC 3.0 a través de redes híbridas (broadcast/broadband) definidos en la norma A/331, el protocolo MMTP se basa en MPEG media Transport (MMT) y utiliza las unidades de procesamiento de medios (MPU) para la entrega y procesamiento de audio, video y servicios,

así como el protocolo ROUTE se basa en MPEG-DASH y utiliza segmentos de medios DASH para la reproducción de audio, video y servicios.

- La capa de Transporte establecida para el estándar ATSC 3.0 implementa los protocolos conocidos como MMTP y ROUTE-DASH para la entrega de contenido y servicios de manera eficiente basando en la utilización de información de señalización de bajo nivel LLS para filtrar de manera correcta si el paquete pertenece a el protocolo MMTP o ROUTE-DASH, además utiliza la tabla de lista de servicios SLT que es un tipo de instancia de la información LLS para la decodificación y correcta entrega de información de servicios SLS en el receptor.
- Se puede concluir que la tabla de lista de servicios SLT definida para la capa de transporte del estándar ATSC 3.0, tiene la misma función que la de Tabla de Asociación de Programas PAT definida en los sistemas MPEG-2. Ofreciendo un escaneo rápido de canales que permite a un receptor crear una lista de todos los servicios que puede recibir, con su nombre de canal, número de canal, etc., además proporciona información de arranque que permite al receptor descubrir la información SLS para cada uno de los servicios.
- Para la interconexión eficiente entre la capa de Transporte y la capa Física del estándar ATSC 3.0, la norma A/324 establece al protocolo ALPTP, DSCP los cuales utilizan el protocolo CTP para ocultar aspectos del protocolo individual que se esté utilizando, con el objetivo de que cada uno de estos pueda ser tratados de manera similar cuando se enrutan al PLP correspondiente, es así que el protocolo DSTP define un encabezado de información de paquetes tunelizados que permite evitar modificaciones en los paquetes de fuentes de datos (MMT o ROUTE-DASH). Este encabezado proporciona información para ayudar al generador de ALP a enrutar los diversos paquetes al PLP apropiado, el protocolo ALPTP proporciona información de red adicional a los paquetes ALP para su eficiente

transporte por redes IP y el protocolo STLP proporciona un mecanismo para transferencia de paquetes banda base hacia receptores broadcast ATSC 3.0 utilizando diferentes enlaces como microondas, fibra o satélite.

- Se concluye que la estructura de un flujo de capa de transporte del estándar ATSC 3.0, se define por la entrega de contenido y servicios mediante los protocolos MMTP Y ROUTE-DASH, los cuales son encapsulados en un paquete ALP utilizando el protocolo DSTP, y transmitidos por redes IP utilizando el protocolo ALPTP, los cuales serán transmitidos a receptores broadcast ATSC 3.0 utilizando el protocolo STLP.
- Se concluye que la biblioteca libatsc3 es una herramienta eficaz para el análisis del estándar ATSC 3.0, ya que permite desglosar la estructura de un flujo de muestra ATSC 3.0, utilizando la herramienta principal atsc3_listener_metrics ncurses se evidenció la entrega y decodificación de información por medio de la información de señalización de bajo nivel LLS y la tabla de servicios SLT, además de información de metadatos y unidades de procesamiento MMT de muestras reales obtenidas en el mercado.

Recomendaciones

- La bibliografía y literatura es extensa de los diferentes protocolos y técnicas avanzadas que permiten definir la estructura de la capa de transporte para el estándar ATSC 3.0 por lo que se propone las siguientes recomendaciones:
- Extender la información de las tablas SLT, RRT, System Time, AEAT, OnscreenMessageNotification que se encuentran en la información de señalización de bajo nivel LLS, que permiten la entrega de varios servicios para el estándar ATSC 3.0.

- Extender los conceptos de la estructura de flujos de los protocolos DSTP, ALTP Y SLTP que permiten la interconexión de la capa de Transporte con la capa Física.
- El uso de las diferentes herramientas que ofrece la biblioteca libastc3, permitirá el desarrollo de nuevos trabajos de investigación, ya que en la biblioteca libastc3 cuenta con un conjunto de herramientas enfocados a protocolos individuales de la capa de transporte, además que es una gran oportunidad para integrarse en la comunidad de investigación sobre el estándar, ya que el diseñador ha creado un foro para que todos los interesados puedan participar y aportar para el entendimiento del estándar.

Bibliografía

- Advanced Television Systems Committee. (15 de Junio de 2016). ATSC: Standard: A/107 - ATSC 2.0 Standard. *A/107 – ATSC 2.0 Standard. ATSC: Standard: A/107 - ATSC 2.0 Standard. A/107 – ATSC 2.0 Standard.* Washintong. Obtenido de <https://www.atsc.org/wp-content/uploads/2015/11/A107-2015-1-1.pdf>
- Advanced Television Systems Committee. (2019). *ATSC Standard:Link-Layer Protocol (A/330)*. Washington, D.C. Obtenido de <https://muygs2x2vhb2pjk6g160f1s8-wpengine.netdna-ssl.com/wp-content/uploads/2021/04/A330-2019a-Link-Layer-Protocol.pdf>
- Advanced Television Systems Committee. (2020). *ATSC Standard: A/300:2020, “ATSC 3.0 System”*. Washington, DC. Obtenido de <https://muygs2x2vhb2pjk6g160f1s8-wpengine.netdna-ssl.com/wp-content/uploads/2021/07/A300-2021-ATSC-3-System-Standard.pdf>
- Advanced Television Systems Committee. (2020). *ATSC Standard: Signaling, Delivery, Synchronization, and Error Protection*. Washington, D.C. Obtenido de <https://muygs2x2vhb2pjk6g160f1s8-wpengine.netdna-ssl.com/wp-content/uploads/2020/09/A331-2020-Signaling-Delivery-Sync-FEC.pdf>
- Advanced Television Systems Committee. (2021). *ATSC Standard: Scheduler / Studio to Transmitter Link*. Washington D.C. Obtenido de <https://muygs2x2vhb2pjk6g160f1s8-wpengine.netdna-ssl.com/wp-content/uploads/2021/03/A324-2021-Scheduler-STL.pdf>
- Chang, M., & Narváez, J. (2015). *Implementación de un Software que Permite Integrar las Redes Sociales con la TV Digital en Tiempo Real para un Canal de Televisión Desarrollado en Ginga NCL*. Guayaquil. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/10344/1/UPS-GT001266.pdf>

- Chernock, R. (1 de Enero de 2015). *ATSC*. Obtenido de ATSC 3.0: WHERE WE STAND: <https://www.atsc.org/news/atsc-3-0-where-we-stand/>
- Chernock, R., Gómez, D., Whitaker, J., Park, S.-i., & Wu, Y. (2016). ATSC 3.0 Next Generation Digital TV Standard—An Overview and Preview of the Issue. *IEEE TRANSACTIONS ON BROADCASTING*, 62(1), 154-158. Recuperado el 31 de marzo de 2021, de <https://ieeexplore.ieee.org/document/7414448>
- Dolan, M. (4 de Enero de 2016). *ATSC*. Obtenido de ATSC 2.0 – BRIDGE TO 3.0: <https://www.atsc.org/news/atsc-2-0-bridge-to-3-0/>
- GitHub. (2010). *GitHub Desarrollador de software*.
- ITU. (2 de Marzo de 2018). *TU Committed to connecting the world>>SUSTAINABLE DEVELOPMENT GOALS>>STATISTICS*. Obtenido de <https://www.itu.int/en/ITU-D/Spectrum-Broadcasting/DSO/Pages/statistics.aspx>
- ITU. (2 de Noviembre de 2019). *ITU Committed to connecting the world: ITU marks 70 years of television standards work*. Obtenido de ITU Committed to connecting the world: ITU marks 70 years of television standards work.
- Jason, J. (13 de marzo de 2019). *ngbp.org: ATSC 3.0 Open Source Tools, Samples, and Insight for NextGen*. Obtenido de [ngbp.org: ATSC 3.0 Open Source Tools, Samples, and Insight for NextGen.:](https://www.ngbp.org/2019/03/2019-03-10-libatsc3-core-release.html) <https://www.ngbp.org/2019/03/2019-03-10-libatsc3-core-release.html>
- Jung, H., Park, S.-i., Lee, J.-y., Hur, N., Kim, H., & Kim, J. (2019). ATSC 3.0 Channel Bonding Performance in Mobile Channel Environments. *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 1-3. Obtenido de <https://ieeexplore.ieee.org/document/8971866?denied=>
- Kwon, W., Hwang, J., Yang, H.-K., Hwang, S., Takahashi, K., & Michael, L. (2016). The ATSC Link-layer Protocol (ALP): Design and Efficiency Evaluation. *IEEE*

Transactions on Broadcasting, 316-327. Obtenido de <https://ieeexplore.ieee.org/document/7389995>

Lee, B., Yang, K., Park, S.-i., Kim, H., & Soon, C. (2018). A Study on the ATSC 3.0 Service Following. *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 1144 -1147. Obtenido de <https://ieeexplore.ieee.org/document/8539515>

MINTEL. (s.f.). *TDT ECUADOR: Ecuador prepara el escenario para la llegada de la TDT*. Obtenido de TDT ECUADOR: Ecuador prepara el escenario para la llegada de la TDT: <https://tdtecuador.mintel.gob.ec/antecedentes-tdt/>

Song, J., Yang, Z., & Wang, J. (2015). *Digital Terrestrial Television Broadcasting: Technology and System*. Obtenido de <https://ieeexplore.ieee.org/servlet/opac?bknumber=7198538>

Soseman, N. (21 de Julio de 2016). *PLAYOUT BROADCAST & TRANSMISSION THE _____ BRIDGE*. Obtenido de ATSC 3.0 Details Explained, Part 4: <https://www.thebroadcastbridge.com/content/entry/6229/atsc-3.0-details-explained-part-4>

Tcpreplay. (s.f.). *Tcpreplay - Pcap editing and replaying utilities*. Obtenido de Tcpreplay - Pcap editing and replaying utilities: <https://tcpreplay.appneta.com/>

UTI-R. (2016). Manual sobre la implantación de redes y sistemas de radiodifusión de televisión terrenal digital. *Radiodifusión de televisión terrenal digital (DTTB)*. Ginebra.