UNIVERSIDAD NACIONAL DE RÍO CUARTO
FACULTAD DE INGENIERÍA
INGENIERÍA EN TELECOMUNICACIONES

Práctica Profesional Supervisada

Título:

Servicio de distribución de IP TV utilizando el protocolo SRT. Investigación e implementación

Autor:
Luis Nazareno Muñoz

Organización: COLSECOR Lta

Tutor por la Facultad: Enrique Alonso

Tutor por la Organización: Nicolás Ban

Lugar y fecha de realización de la PPS: Río Cuarto-Córdoba Capital. Enero-marzo 2019

Fecha presentación del informe: 17 de abril de 2019

Resumen

El objetivo de la presente práctica fue la transmisión de contenido multimedia en calidad HD desde Colsecor hacia una ubicación remota en la red, utilizando de plataformas de servicio de *streaming* y protocolos de reciente desarrollo que permitan superar los inconvenientes propios de las redes públicas de Internet.

Durante el desarrollo de la misma, se recurrió a conceptos y prácticas que fueron estudiadas durante el cursado de la carrera, entre los que se pueden destacar conceptos referidos a servidores, redes, tráfico de datos, formatos y codificadores de video, entre otros.

El principal objetivo de la práctica fue investigar del protocolo SRT para posteriormente implementarlo en sistemas de distribución, ya sean simulados o no, para comprobar su funcionamiento y viabilidad.

Se recurrió a la plataforma *Wowza Streaming Engine* como agente distribuidor de la señal codificada. A esta también se la testeó en distintos escenarios, a la vez que se detallaron las configuraciones correspondientes a cada escenario.

Como resultado final de la práctica, se obtuvo una transmisión de IP TV en calidad HD estable desde las instalaciones de Colsecor hasta un usuario final utilizando el protocolo SRT, implementado sobre la plataforma Wowza y atravesando redes de uso público.

Para la realización de la presente práctica se utilizaron recursos provistos por Colsecor, así como también recursos del departamento de telecomunicaciones.

Contenido

Objetivos de la práctica

Como se ha dicho con anterioridad, el objetivo principal de la práctica fue la comprobación del funcionamiento de protocolos de reciente desarrollo que permitan transmisiones estables de contenido en alta definición (HD) entre dos instancias remotas, atravesando Internet. Todo esto, utilizando el equipamiento utilizado por Colsecor, además de contar con el asesoramiento por parte de su equipo técnico.

Objetivos secundarios:

- Aplicar conceptos adquiridos en la carrera referidos a transmisión de datos multimediales.
- Establecer un vínculo con el grupo de trabajo de Colsecor y planificar en conjunto el escenario a desarrollar.
- Investigar protocolos de distribución de contenido multimedia con baja latencia.
- Investigar el funcionamiento y configuración de plataformas de servicios de streaming.
- Implementar soluciones factibles para una distribución de IP TV confiable, de calidad y rentable.
- Comprobar el funcionamiento del protocolo SRT a lo largo de una transmisión servidor-usuario final.

Cronograma de trabajo

Detalle de tareas a realizar

1. Incorporación de conocimientos sobre IP TV.

Duración: 15 Hs.

2. Incorporación de conocimientos sobre protocolos SRT, ZIXI y ARQ.

Duración: 15 Hs

3. Análisis de los manuales de los equipamientos utilizados por al utilizar el protocolo SRT.

Duración: 45 Hs.

4. Análisis del Ecosistema propuesto.

Duración: 25 Hs.

5. Configuración del equipamiento.

Duración: 30 Hs.

6. Estudio de las soluciones implementadas actualmente en IP TV. Normas.

Duración: 30 Hs.

7. Análisis de los requerimientos a cumplir en el escenario propuesto.

Duración: 20 Hs.

8. Integración de la solución a un Cabezal DVB-C o IPTV de COLSECOR.

Duracion:30 Hs.

9. Interconexión de cabecera con usuarios dentro del laboratorio.

Duración: 20 Hs.

10. Configuración de receptores de los usuarios.

Duración: 10 Hs.

11. Análisis del tráfico de datos y calidad de servicio.

Duración: 30 Hs.

12. Elaboración del Informe Final.

Duración: 40 Hs.

Cronograma de actividades

Actividades	Semanas											Total de Horas
	1	2	3	4	5	6	7	8	9	10	11	
1	15											
2	15											
3		30	15									
4			15	10								
5				20	10							
6					20	10						
7						20						
8							30					
9								20				
10								10				
11									30			
12										30	10	
Horas Semanales	30	30	30	30	30	30	30	30	30	30	10	310

Introducción[1]

La distribución de señales de televisión mediante el protocolo de Internet, comúnmente denominado IPTV es el desafío actual de las proveedoras de servicios multimedia. En esquemas actuales, las arquitecturas diseñadas para proveer estos servicios están basadas en redes propias y controladas, o bien en el alquiler de enlaces dedicados a empresas mayoristas.

Teniendo en cuenta las tendencias actuales de consumo de contenido multimedia, las plataformas que permiten la distribución de contenidos streaming toman cada vez más relevancia en la estructura de internet.

El principal desafío de estos esquemas es superar los inconvenientes del medio en que se distribuyen estos contenidos, que generalmente es la red pública de Internet que, por su naturaleza, no es un ambiente propicio para transmisiones confiables en tiempo real.

Partiendo de ello, las grandes empresas de distribución de contenido por Internet comenzaron a desarrollar protocolos que sean capaces de hacer frente a estos inconvenientes. En el caso de Colsecor, ya posee las plataformas necesarias para la implementación del protocolo SRT. Pero dado lo reciente de estos desarrollos, la empresa aún no tenía ninguna implementación utilizando SRT, con lo cual esta práctica representó la primera transmisión de SRT atravesando con éxito toda su red. Esto es: desde el origen del contenido hasta el usuario final.

Como parte del convenio vigente entre Colsecor y la FI-UNRC la primera facilitó recursos que fueron utilizados en desarrollo de la presente PPS; léase acceso a las plataformas de contenidos y a los contenidos propiamente dichos como también espacio físico en sus instalaciones de la ciudad de Córdoba.

Acerca de la empresa

La Cooperativa de Provisión y Comercialización de Servicios Comunitarios de Radiodifusión COLSECOR Ltda, es una Cooperativa de segundo grado integrada por otras cooperativas y Pymes de todo el país, que prestan servicios de telecomunicaciones, entre los cuales están diversos servicios de TV.

[1] En Anexos se incluye un glosario con conceptos mencionados a lo largo del presente informe.

El principal rol de esta entidad es la integración de dichas Cooperativas a los fines de lograr economías de escala que esta disciplina requiere y que, de no existir este tipo de asociaciones, sería imposible de lograr en localidades medianas y pequeñas, en donde se encuentra la inmensa mayoría de las que la componen.

Entre algunos aspectos que se pueden destacar en sentido de lo antes dicho se puede mencionar la contratación, de manera mayorista de las señales de TV a los correspondientes dueños de contenido.

Situación actual de la distribución de señales de video de Colsecor.

Actualmente se utilizan señales de video de calidad estándar (SD), que son captadas por una antena y un receptor. La señal es inyectada en un *Encoder*, que permite obtener a la salida IP-TS.

Esta salida es una salida con características de *multicast*. Se configura el *router* del transmisor para que admita contenido multicast y se envía hacia Internet utilizando un túnel GRE entre los routers de borde. El router receptor se encarga de hacer llegar el contenido multicast al *Set Top Box* del usuario final.

Este esquema, ilustrado en la Figura 1.1, solo permite distribuir contenido en calidad standard, requiere de grandes anchos de banda y es muy vulnerable a pérdida de paquetes, *jitter* y *delay*.

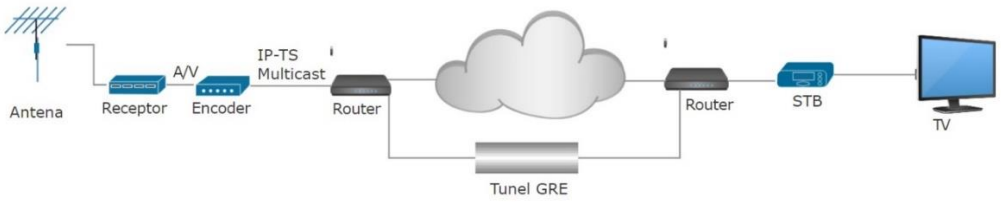


Figura 1

En un segundo esquema utilizado por Colsecor, que se muestra en la Figura 1.2, se comienza a utilizar la plataforma Wowza Streaming Engine, con el objetivo de lograr mejoras con respecto a parámetros tales como perdida de paquetes, jitter y retardo. El inconveniente es que no permite una mejora en la calidad del contenido enviado.

Se utiliza el protocolo *HTTP Live Streaming*, también conocido como HLS, desarrollado por Apple Inc. Este divide los paquetes Transport Stream en fragmentos de igual longitud, denominados chunks. (*Ver Anexo HLS*)

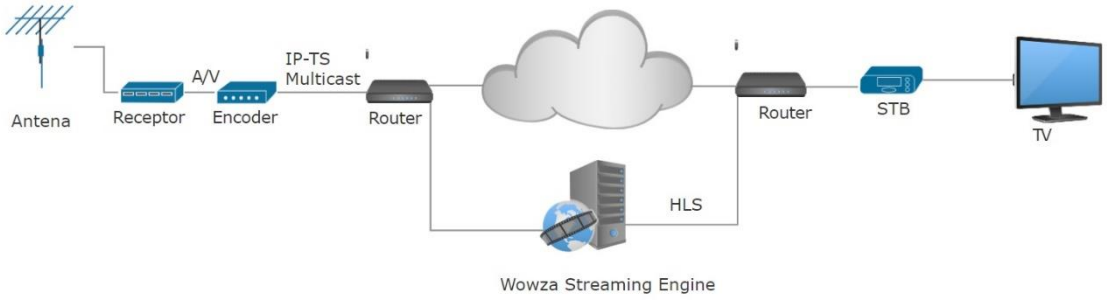


Figura 2

Teniendo en cuenta los inconvenientes planteados, se propuso la implementación de un nuevo protocolo de transporte que sea confiable y seguro para transmisiones de video.

Implementación

Con todo lo expuesto anteriormente, y siguiendo la línea de progresión planteada en el plan de trabajo, se analizaron las distintas propuestas que pudieran dar solución a los inconvenientes de las redes públicas, esto es, tomar conocimiento de los protocolos de transporte utilizados en la actualidad, entre estos SRT, ZIXI y ARQ.

A continuación, se detallan los distintos protocolos que actualmente están siendo implementados por las empresas de servicios de streaming, IPTV y afines:

-SRT

Este protocolo de transporte confiable (SRT, por sus siglas en inglés) fue desarrollado por Haivision, es de **código abierto** y es apoyado por más de 160 empresas de la industria de servicios multimedia. Utiliza el protocolo UDP como protocolo de transporte, y realiza control del enlace mediante mensajes UDT.

-ZIXI

Es una **tecnología patentada** que utiliza técnicas de corrección de error avanzada (FEC) para controla el jitter, perdida de paquetes y la latencia, que provocan errores en la transmisión libre de video. Proporciona una latencia fija y predecible de menos de un segundo para transmisiones en cualquier parte del mundo.

Decisión respecto a protocolo a utilizar

Habiendo analizado detalladamente cada una de estas opciones, y valorando sus ventajas y desventajas, se decidió implementar el protocolo SRT que es de **código abierto**, y que además es compatible con la plataforma Wowza Streaming Engine que ya es utilizada por Colsecor y por la que posee sus licencias correspondientes.

La plataforma Wowza Streaming Engine es capaz de desempeñar funciones tanto de recepción de contenido SRT (*Listener*) o de enviar contenido mediante SRT (*Caller*), con lo cual se diseñó el siguiente escenario como objetivo final de la práctica, teniendo a la plataforma como elemento central en la distribución de los contenidos sobre SRT.

Escenario objetivo de la práctica

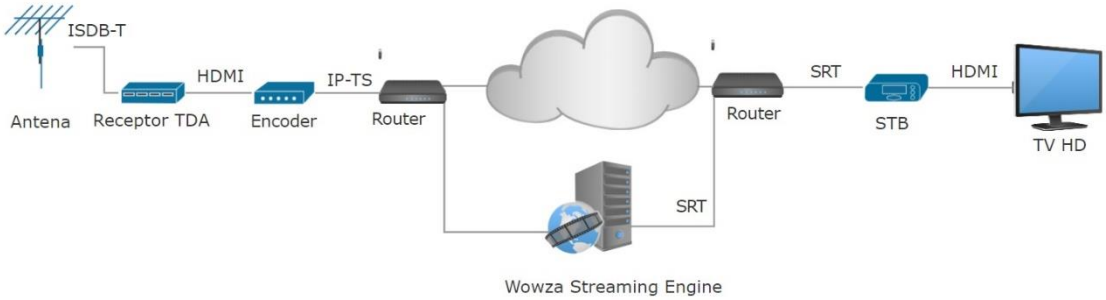


Figura 3

La implementación a la cual se busca llegar es a la mostrada en la Figura 3, que consisten en utilizar el protocolo SRT para atravesar las redes no controladas de Internet. Esto se realizará utilizando la plataforma Wowza Streaming Engine como receptor de los Transport Stream (TS) del cabezal de Colsecor y transmitiendo hacia los usuarios el Canal 12en calidad HD mediante el protocolo SRT.

Cabe destacar que en la PPS se optó por hacer el despliegue de SRT entre las instancias desde la generación del contenido hasta la distribución final al usuario, pero la implementación de interés para Colsecor es solo la parte de distribución del contenido hasta un Nodo distribuidor de contenido cercano al usuario final. Esto en vistas de una próxima implementación del protocolo para la comunicación entre sus nodos. La distribución hasta el usuario final se haría mediante multicast y no por SRT.

La ampliación del escenario se realizó a fines de poder corroborar fehacientemente que el contenido llegara al usuario en la calidad que exigía en el diseño.

Funcionamiento de SRT

Protocolo Secure Reliable Transport (SRT)

El protocolo SRT fue desarrollado por *Haivision* con la finalidad de mejorar la calidad y confiabilidad de las transmisiones de video a través de Internet.

A continuación, se muestra el esquema simplificado de SRT

Figura 4

Como se puede observar en la Figura 4, el contenido a transmitir es ingresado a un SRT Encoder HD, es decir, un codificador HD que soporta SRT. Después de la codificación, el stream poseerá encriptación y capacidad de corrección de errores, y posteriormente será inyectado a la red de Internet público. En el receptor, el flujo de paquetes SRT deberá ser decodificado para su posterior reproducción.

Para visualizar la mejora que se alcanza con el uso de SRT, podemos analizar las siguientes graficas:

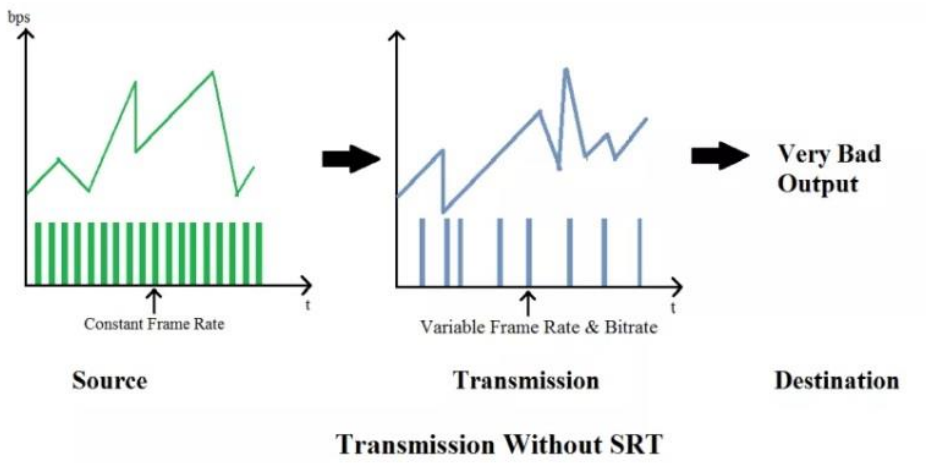


Figura 5

En la Figura 5, se puede observar que en una transmisión a través de Internet pública se producen variaciones en la tasa de la trama, lo cual es sumamente perjudicial para la correcta reproducción del contenido. Estas variaciones son producidas por el Jitter, Pérdida de paquetes y Latencia de la red.

Utilizando SRT, se logran compensar estas distorsiones, lo cual podemos visualizar en la siguiente imagen:

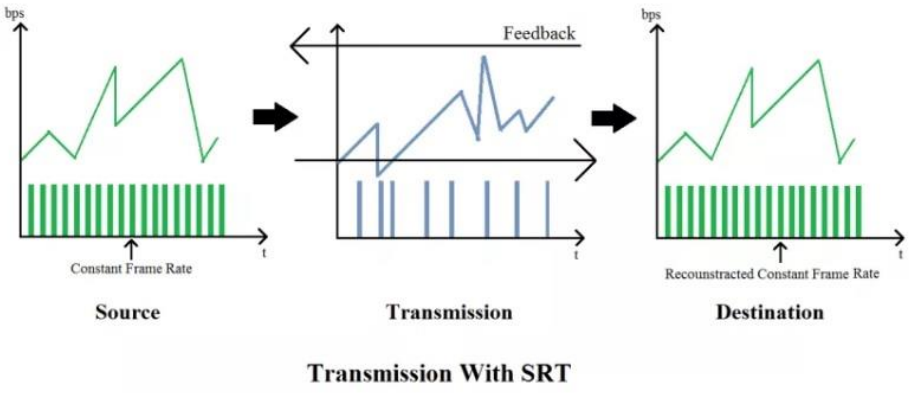


Figura 6

El resultado se observa claramente en la Figura 6, y es que se puede reconstruir la tasa de trama de la reproducción original.

Conceptos Básicos

Fuente y Destino

El protocolo SRT se basa en un tráfico bidireccional de UDP para optimizar los streams de video sobre redes públicas. Hay que tener presente que los datos de video son enviados en una sola dirección, desde el dispositivo la fuente de contenido hacia el destino, y hay un contaste intercambio de información de control entre las dos instancias finales, incluyendo paquetes "keep alive" (si fuera necesario) aproximadamente cada 10 ms, haciendo capaz al flujo SRT de ser restaurado automáticamente después de una conexión perdida.

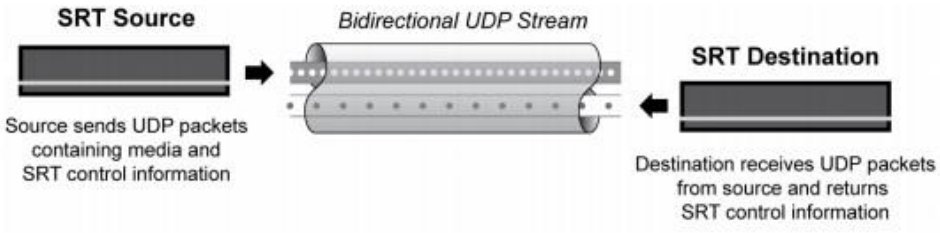


Figura 7

En algunos casos un dispositivo puede actuar como fuente y destino. Por ejemplo, un servidor Gateway podría actuar como destino mientras recibe un stream SRT desde un codificador, y convertirse en fuente para retransmitirlo al decodificador.

Modos de llamado SRT

Para establecer un flujo bidireccional, SRT emplea mecanismos de negociación (Handshake) donde cada dispositivo se identifica a sí mismo como Caller o llamador o como

Listener u Oyente. En otros casos, dos dispositivos pueden negociar simultáneamente una sesión SRT, esto es el modo Rendezvous o de Reunión.

Conocer estos modos de negociación es fundamental para la configuración del flujo SRT a desplegar a lo largo del desarrollo de la práctica.

SRT y Firewalls

En muchas situaciones del mundo real, y particularmente cuando estén involucradas transmisiones sobre Internet, los flujos SRT deberán pasar a través de Firewall tanto en la fuente, destino o en ambos puntos finales. En medida de lo posible, el administrador de red deberá configurar ciertas características en el Firewall, específicamente lo referido a Network Address Translation (NAT) y filtrado de paquetes. Estas configuraciones serán diferentes según el modo en que se configure el dispositivo, es decir, Caller, Listener o Rendezvous.

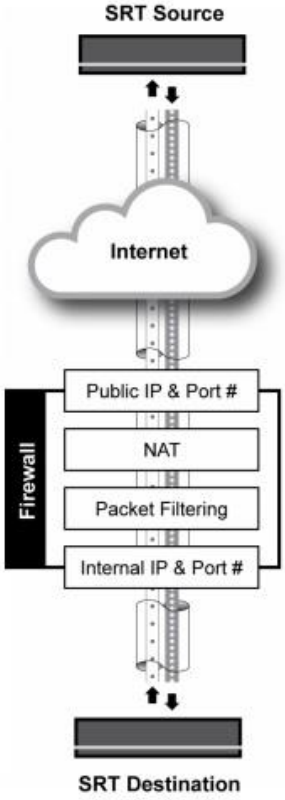


Figura 8

De manera breve, se explicará el funcionamiento del protocolo con la secuencia de paquetes que se intercambian durante la sesión.

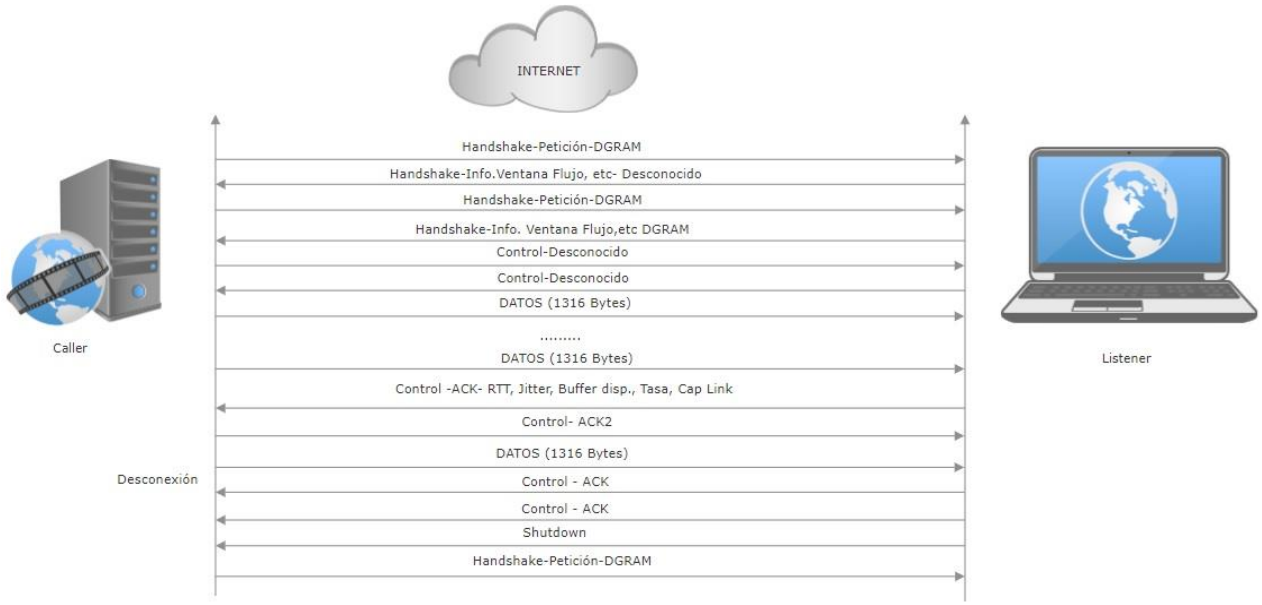


Figura 9

Esto se puede constatar con los paquetes que fueron capturados durante el desarrollo de una sesión SRT implementada en la etapa de testeo.

13809	34.662867140	192.168.0.208	192.168.0.203	UDT	106	UDT type: handshake	
13837	34.700894553	192.168.0.203	192.168.0.208	ICMP	134	Destination unreachable (Port unreachable)	
13868	34.743330302	192.168.0.200	192.168.0.208	UDT	106	UDT type: handshake	
13869	34.743423188	192.168.0.208	192.168.0.200	UDT	106	UDT type: handshake	
13870	34.753330440	192.168.0.200	192.168.0.208	UDT	106	UDT type: handshake	NEGOCIACIÓN
13895	34.862305389	192.168.0.208	192.168.0.200	UDT	62	UDT type: keepalive	
13923	34.904536612	192.168.0.208	192.168.0.200	UDT	70	UDT type: Unknown Control Type (7fff)	CONTROL
13924	34.904638864	192.168.0.208	192.168.0.200	UDT	1374	UDT type: data seqno: 0 msgno: 1	
13948	34.907879578	192.168.0.200	192.168.0.208	UDT	70	UDT type: Unknown Control Type (7fff)	
13957	34.919658030	192.168.0.208	192.168.0.203	UDT	106	UDT type: handshake	
13959	34.955471963	192.168.0.208	192.168.0.200	UDT	1374	UDT type: data seqno: 1 msgno: 2	DATOS
13960	34.959516521	192.168.0.200	192.168.0.208	UDT	86	UDT type: ack seqno: 2	
13961	34.959554838	192.168.0.208	192.168.0.200	UDT	62	UDT type: ack2	INFORME Y CONFIRMACION
14025	35.108245905	192.168.0.208	192.168.0.200	UDT	1374	UDT type: data seqno: 2 msgno: 3	
14029	35.120303079	192.168.0.200	192.168.0.208	UDT	86	UDT type: ack seqno: 3	
14033	35.120589048	192.168.0.208	192.168.0.200	UDT	62	UDT type: ack2	
14060	35.160407603	192.168.0.208	192.168.0.200	UDT	1374	UDT type: data seqno: 3 msgno: 4	
14061	35.167886801	192.168.0.208	192.168.0.200	UDT	1374	UDT type: data seqno: 4 msgno: 5	
14062	35.170681663	192.168.0.200	192.168.0.208	UDT	86	UDT type: ack seqno: 4	
14063	35.170878924	192.168.0.208	192.168.0.200	UDT	62	UDT type: ack2	

Figura 10

Como puede apreciarse en la *Figura 10*, el servidor, en este caso alojado en el dispositivo asociado a la IP 192.168.0.208 es el que envía las invitaciones a los dispositivos para enviarles el contenido. El envío de estos paquetes, en este esquema, es una capacidad solo disponible para el que sea configurado como Caller. El Listener es el que debe contestar a dicha invitación. Existen otras modalidades de funcionamiento donde no está definido un Caller y un Listener, sino que ambos son capaces de iniciar el proceso de negociación.

Una vez levantada la instancia de recepción en el receptor, se inicia el proceso de negociación. Luego de esto, el receptor envía mensajes de control, a la vez que el transmisor

comienza el envío de datos. En este caso, el transmisor envía el contenido, y el receptor recibe la señal, pero además envía paquetes de control hacia el transmisor para que este adecue el envío de paquete según las capacidades del receptor, que serán indicadas en estos paquetes de control.

En caso de desconexión por parte del receptor, este envía el paquete de apagado o (Shutdown) que termina la sesión. En caso de caída del servicio, el transmisor detecta la falta de ack y da por caída la sesión

Funcionamiento Wowza Streaming Engine

Presentación de la plataforma

En el desarrollo de la práctica se dispuso de servidores de streaming Wowza, que es una plataforma de streaming ampliamente utilizada por grandes corporaciones abocadas a la transmisión de datos multimedia a través de Internet.

Wowza Streaming Engine es un servidor de software de contenido multimedia unificado desarrollado por Wowza Media Systems. Este servidor es usado para hacer transmisiones en vivo y de video bajo demanda, audio, aplicaciones de internet embebidas sobre redes IP hacia computadoras personales, teléfonos móviles, IPTV set-top-boxes, TV conectadas a internet, consolas de video juegos y otros dispositivos conectados a la red.

Figura 11

Protocolos de transmisión

Esta plataforma permite transmitir contenido hacia una amplia gama de clientes y dispositivos de manera simultánea, incluyendo reproductores Adobe Flash player, Microsoft

Silverlight, Apple QuickTime y dispositivos iOS, Celulares, set-top-box IPTV (Amino, Enseo, Roku, Streamit y otros), y consolas de videojuego Wii, Xbox y PS3.

Es compatible con los estándares de transmisiones streaming. Respecto a las salidas, incluye RTMP (y sus variantes RTMPS, RTMPT, RTMPE, RTMPTE), HDS, HLS, MPEG DASH, RTSP. Smooth Streaming, y MPEG-TS (unicast y multicast), WebRTC. Se puede ingresar contenido al servidor mediante RTP, RTSP, RTMP, MPEG-TS (unicast y multicast) y transmisiones ICY (SHOUTcast/Icecast). Además, soporta transmisiones entrantes vía RTSP a protocolos WOWZ desde celulares iOS y Android que ejecuten su aplicaion GoCoder.

Configuración de un streaming SRT en Wowza

Si utilizamos la PWSE como Source deberemos configurar el Destination, es decir el receptor. A continuación, se detalla el menú de configuración.

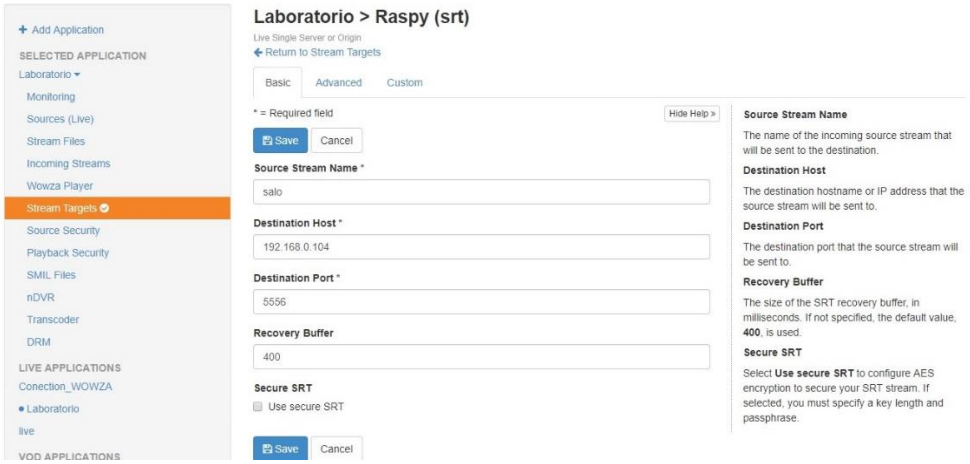


Figura 12

Como puede observarse, se configuró la dirección IP y el puerto del receptor. Además, se configuran el contenido a distribuir y el buffer de recupero. Si se desea, se puede habilitar la encriptación del stream.

Implementaciones

Teniendo siempre en mente el paradigma cliente-servidor, se plantearon distintos niveles de complejidad en el armado del escenario, esto en virtud de permitir un completo entendimiento del protocolo en cuestión, además de aislar los inconvenientes y problemas que fueran surgiendo en cada uno de estos escenarios.

Esta evolución en la complejidad de los escenarios fue de la mano con una evolución o cambio de las herramientas utilizadas, que, si bien cumplieron funciones similares, tenían distinto grado de complejidad y rendimiento. Estas se detallarán en cada caso.

Dado que la plataforma Wowza Streaming Engine iba a ser el núcleo de las implementaciones, desde las más básicas hasta las finales, se debió investigar y estudiar detenidamente su funcionamiento.

Herramientas utilizadas

-stransmit: Es el script que se crea al compilar los repositorios de SRT, obtenidos de GitHub. Es el que gestiona la conexión SRT, la crea, mantiene y destruye.

-ffmpeg: Paquete de software libre que permite la grabación, conversión y streaming de audio y video-

-ffplay: Herramienta del paquete ffmpeg que se utiliza para la reproducción de contenido multimedia.

-netem: Simulador de red. Utilizado para emular condiciones de perdida de paquetes, jitter y latencia.

-Wowza Streaming Engine: Plataforma multiprotocolo de contenido multimedia.

Práctica de laboratorio

El primer paso luego del análisis teórico del funcionamiento del protocolo fue realizar las primeras pruebas en laboratorio.

Primera Experiencia con Wowza Local

Se decidió comenzar con la estructura más sencilla posible del esquema planteado, es decir, un servidor y un cliente.

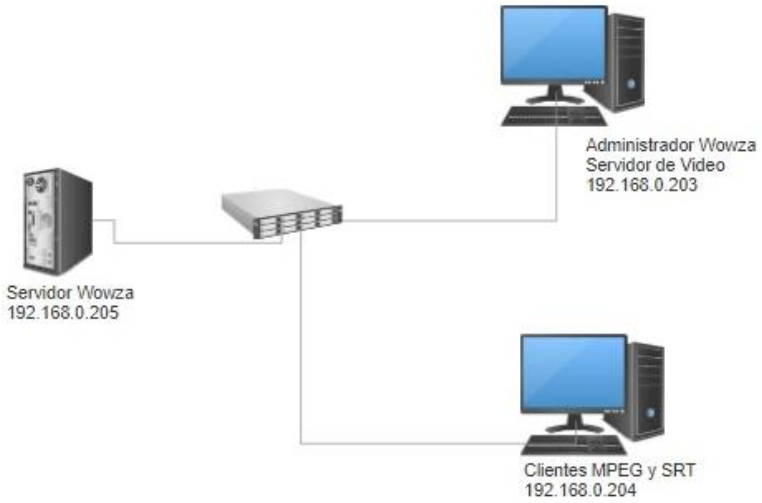


Figura 13

El escenario planteado se utilizó para hacer la implementación más sencilla posible.

Elementos presentes en el escenario

Servidor de Video Streaming: Se utilizó una PC con Windows, que dispone de la herramienta FFMPEG, y se configuró para que subiera contenido en rtmp al Servidor Wowza Local. El servidor de Video Streaming dispone de videos en alta calidad para transmitir. Es quien inyecta el contenido multimedia al sistema.

Servidor Wowza: Alojado en una PC con Ubuntu 18, es el encargado de recibir el contenido en rtmp y enviar el contenido en el protocolo SRT. Su administración se realiza mediante una interface web.

Receptor SRT: Se utilizó una PC con Ubuntu 18, y se requirió de la obtención se los repositorios de Haivision en GitHub para la compilación y posterior ejecución de herramientas stransmit. Además, dispuso del paquete FFmpeg, por la necesidad de la característica FFplay.

Esta implementación se realizó con la finalidad de lograr lo siguiente:

- Toma de contacto con la plataforma Wowza
- Configuración de las instancias participantes
- Conexión mediante el protocolo SRT entre plataforma y receptor
- Comparar rendimiento de SRT contra MPEG-TS

Desarrollo de la experiencia

Como primera experiencia, se configuró todo en el entorno de una red local, en donde se alojaron tanto el servidor de videos, la plataforma de streaming y el receptor.

Para la generación de video se utilizó el software FFmpeg en el sistema operativo Windows, y se utilizó un video en alta calidad. Las comandos e instrucciones necesarias para la configuración del servidor de streaming se han adjuntado en la sección Anexos, y se muestra en la figura #.

El contenido fue enviado a la plataforma Wowza Streaming Engine (de ahora en mas PWSE), y esta fue la encargada de actuar como Llamador (Caller) del receptor.

Toda la configuración y administración de la plataforma se realiza mediante una interface web similar a la mostrada en la figura #.

Para simular condiciones de congestión, se utilizó la herramienta NetEm. Lo que se hizo fue crear un script que permitiera modelar la red con parámetros tales como perdida de paquetes, jitter y latencia. Se configuró de manera tal que exista un 2 % de pérdida de paquetes, una latencia de 150 ms, con una variación que responde a la ley gaussiana de distribución.

Dado que la PWSE recibía el contenido del servidor, se debía configurar la transmisión del contenido desde el Servidor de Video y la PWSE. El envío de contenido a la PWSE no es irrestricto, se requiere conocimiento de la aplicación a la que esta se apuntó, además de una autenticación con el nombre y contraseña de la sesión.

Configuraciones

La configuración se puede observar en la *Figura 14*.

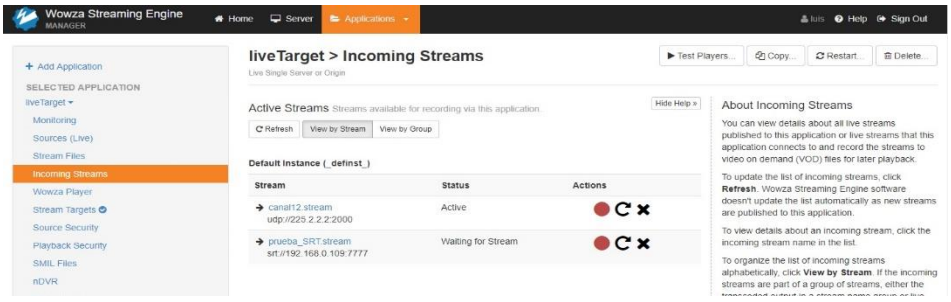


Figura 14

Una vez que se dispone el contenido en *Incoming Steaming*, se puede corroborar que el contenido está llegando a la plataforma de forma correcta.

Configuración de un streaming MPEG-TS

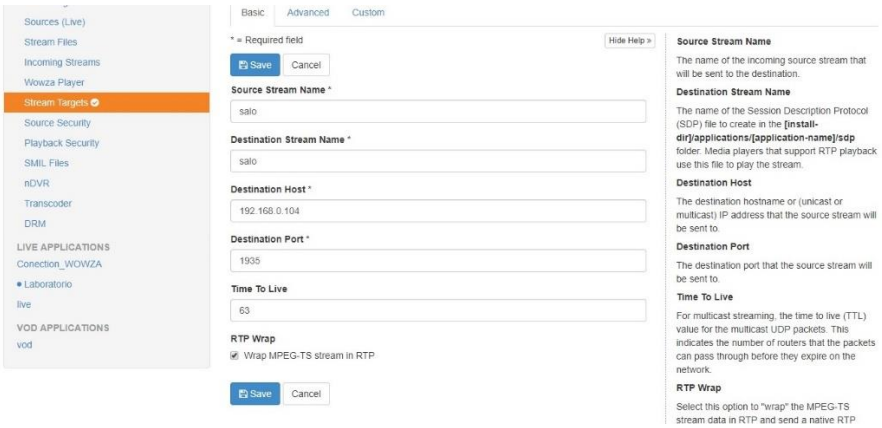


Figura 15

De manera similar, se configuró un *TargetStream* MPEG-TS para comparar el rendimiento antes condiciones desfavorables, que fueron emuladas mediante *Netem*.

La configuración utilizada fue la siguiente:

sudo tc qdisc change dev enp0s3 root netem delay 200ms 40ms 25% loss 15.3% 25% duplicate 1% corrupt 0.6% reorder 5% 50%

Configuración de un streaming SRT

Se configuró la transmisión mediante protocolo SRT como se observó en la figura #. Como se comentó anteriormente, la conexión es unicast, con lo cual se debe configurar un destino, con IP y puerto, así como también se especificó que contenido se debe enviar. Posterior a esto, hay que habilitar el destino.

Con esto, la PWSE comenzó a emitir paquetes de negociación hacia el receptor.

Configuración Receptor SRT

En el receptor se descargaron y compilaron las herramientas provistas por Haivision (ver Anexos). Hecho esto, se utilizó stransmit para la recepción de SRT y se configuro un socket udp para la reproducción, como se detalla a continuación:

Cliente SRT:

Se debe situar en la carpeta srt-master, que la carpeta donde se compilaron las herramientas y ejecutar:

stransmit srt://:5556 udp://127.0.0.1:6666

donde 5556 es el puerto al que se le apunta desde el servidor. Posteriormente, se traduce hacia un socket de salida, que es configurado por el receptor.

ffplay udp://127.0.0.1:6666

Posteriormente, mediante la herramienta ffplay se reproduce el contenido recibido desde la PWSE, y que fue direccionado al socket de salida en la IP local.

Las salidas de los reproductores de video serán analizadas en la sección resultados.

Escenario utilizando la PWSE de Colsecor y receptor Linux genérico.

Una vez que se analizó y comprobó el funcionamiento del protocolo SRT en un entorno local, se estuvo en condiciones de aumentar la dificultad del escenario, utilizando la PWSE de Colsecor, atravesando Internet.

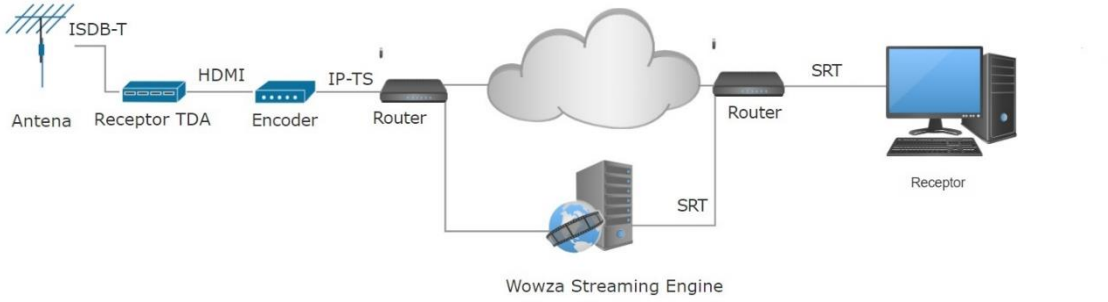


Figura 16

Elementos presentes en el escenario

Servidor de contenido: En este caso se utilizaron señales de canal 12 de córdoba que ya estaban cargadas en la PWSE.

Servidor Wowza: Esta montado en un PC Debian 8 "Jessie", alojado en las instalaciones de Colsecor Lta. de la Ciudad de Córdoba. Recibe la señal de canal 12 HD por medio de transmisión multicast en cabecera y fue el encargado de enviar el contenido en el protocolo SRT. Su administración se realiza mediante una interface web.

Receptor SRT: Se utilizó una PC con Ubuntu 18, y se requirió de la obtención se los repositorios de Haivision en GitHub para la compilación y posterior ejecución de herramientas stransmit. Además, dispuso del paquete FFmpeg, por la necesidad de la característica FFplay.

Esta implementación se realizó con el objetivo de lograr lo siguiente:

- Configuración de las instancias participantes
- Conexión mediante el protocolo SRT entre plataforma y receptor atravesando Internet

Desarrollo de la experiencia

Inconvenientes previos

Aunque conceptualmente consistía en un planteo similar al realizado en el escenario local, el atravesar Internet supuso inconvenientes referidos a la naturaleza univoca de la conexión SRT.

Como la presente practica se desarrolló en el Laboratorio de Comunicaciones y Redes Multimediales, situado en las instalaciones de la Universidad Nacional de Rio Cuarto, en primera instancia no se pudo acceder a la interface de administración de la PWSE de Colsecor, pese a disponer de la dirección IP publica a la cual apuntar. Esto debido a la política de seguridad implementada por la Unidad de Tecnología de la Información (UTI).

Ante este inconveniente, se debieron utilizar herramientas que permitan el acceso remoto a un PC ubicado fuera de las instalaciones de la Universidad Nacional de Rio Cuarto. El PC elegido fue uno ubicado en el domicilio particular del practicante. La manipulación de dicha PC desde las instalaciones de la FI se realizó remotamente mediante la herramienta TeamViewer. Gracias a esto, se pudo acceder correctamente a la plataforma de administración.

En este punto, si bien él se logró subsanar el impedimento de acceso a la administración de la plataforma, aun no era posible la recepción del contenido, por los inconvenientes referidos al direccionamiento Socket a Socket que se mencionó anteriormente. (*Ver sección SRT y Firewalls*)

Como la conexión del domicilio es administrada por un router marca *Mikrotik*, se debió configurar reglas que permitan la traducción del contenido dirigido a la IP publica del router a hacia la dirección IP privada del receptor SRT.

Nateo y Forwarding de puertos

Dada la naturaleza del protocolo SRT, el contenido se puede dirigir solo a una única dirección de destino, con lo cual su despliegue en una red con NAT se hace inviable. Para subsanar este inconveniente se debe disponer de un par de direcciones IP públicas, o en el caso de no ser esto posible, configurar los routers que hacen nateo para que direccionen los paquetes dirigidos a la IP publica hacia la IP privada donde está alojado el receptor o servidor.

En el caso de la práctica, como se comentó anteriormente, se disponía de un router marca Mikrotik, en el cual se realizó la configuración que se detalla en los ANEXOS

Generación del contenido a distribuir

Como se mencionó anteriormente, el contenido a distribuir se obtiene mediante la recepción de señales de TDT (también denominada TDA) en la norma ISDB-T (*Consultar en Anexos*), que son decodificadas y luego son ingresadas a un Pulplexor, que es el encargado de generar los MPEG-TS y publicarlos en un grupo de multicast. Esta señal es la que ingresará a la plataforma y se configura de la siguiente manera:

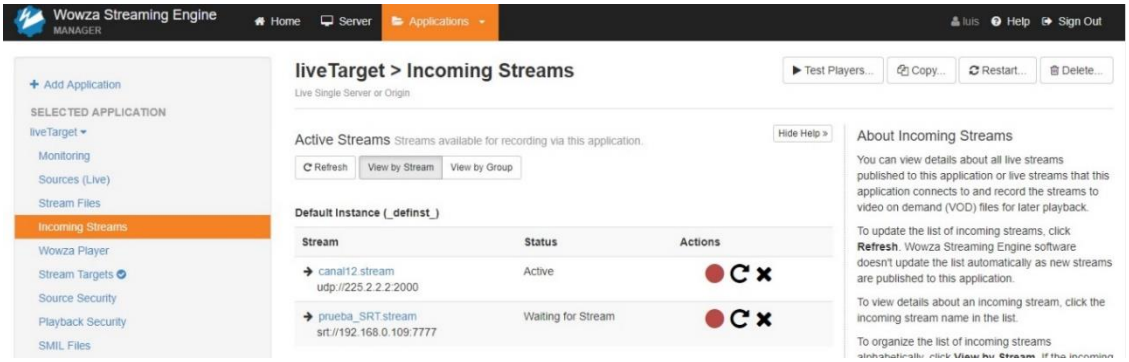


Figura 17

En la PWSE se configura la dirección del grupo multicast como Stream Files, posteriormente se asocia a dicho grupo. Hecho esto, el stream aparecerá en la pestaña Incoming Stream, como se muestra en la *Figura 17*.

Configuración de un streaming SRT

Se configuró el destino SRT en la sección TargetStream, teniendo la precaución de configurar correctamente la IP pública asignada por el ISP al router del domicilio, además de un puerto donde se enviaría el contenido.

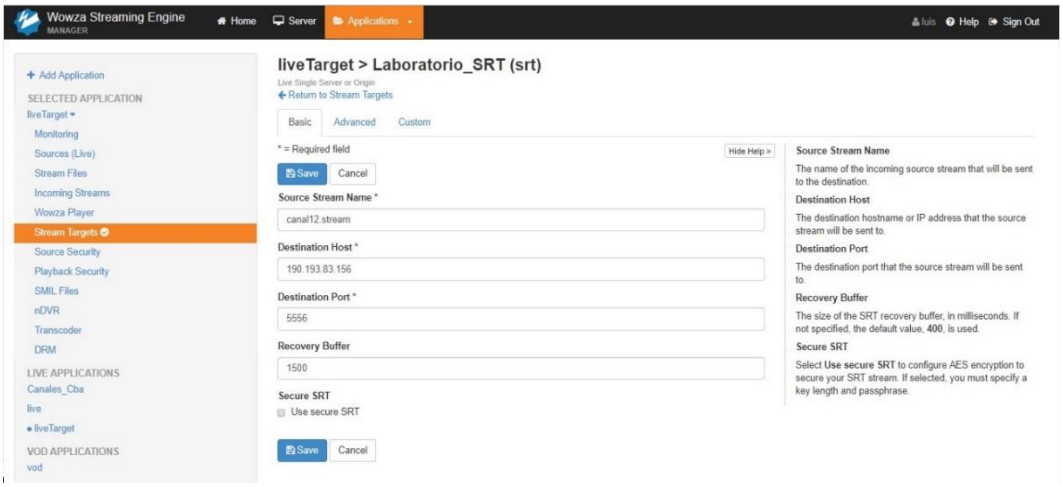


Figura 18

Como puede observarse en la *Figura 18*, además se configuran otros parámetros de importancia, como lo son el Buffer de Recupero y la opción de encriptar la emisión.

Configuración de un streaming MPEG-TS

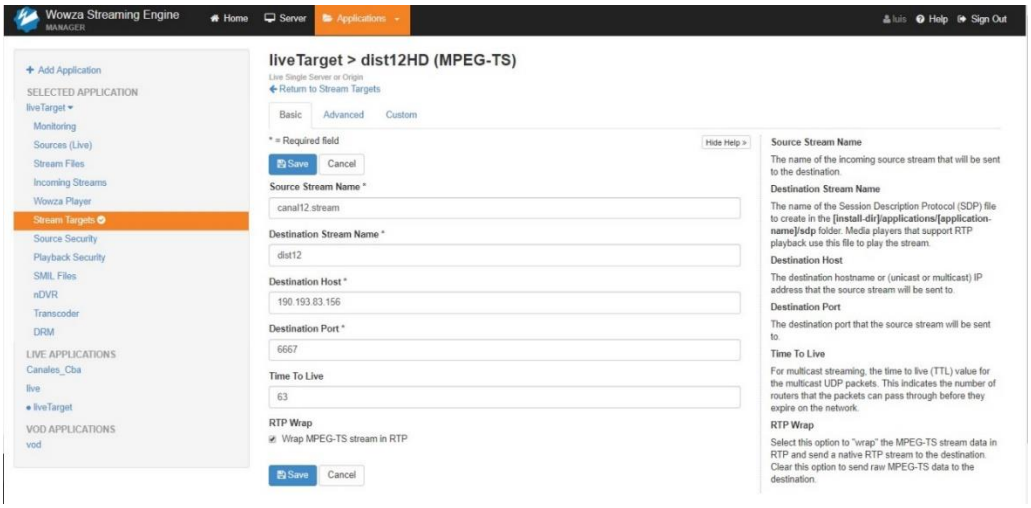


Figura 19

De manera similar a lo realizado al configurar SRT, se configuro el destino del contenido MPEG-TS. En este caso no existe posibilidad de encriptación alguna.

Nota: Cuando de configura la instancia MPEG-TS, no existe negociación entre las partes, solo se envía el contenido.

Configuración Receptor SRT

Finalmente, en el Receptor SRT,que posee idénticas características al utilizado en la implementación local, solo que, tomando en consideración al puerto configurado en la PWSE, en este caso el 5556.

Configuración Receptor MPEG-TS

Para la reproducción de este stream, solamente fue necesario utilizar una instancia de ffplay en la pc, y reproducir el contenido enviado por la PWSE.

>ffplay udp://181.16.201.19:6667

Escenario utilizando la PWSE de Colsecor y receptor montado en STB.

Finalmente, y como escenario objetivo de la práctica, se montaron todos los elementos que están presentes en un esquema habitual de transmisión IPTV implementado por las empresas especializadas en el rubro.

Implementación y escenario desarrollado

El escenario final que se implementó es el siguiente:

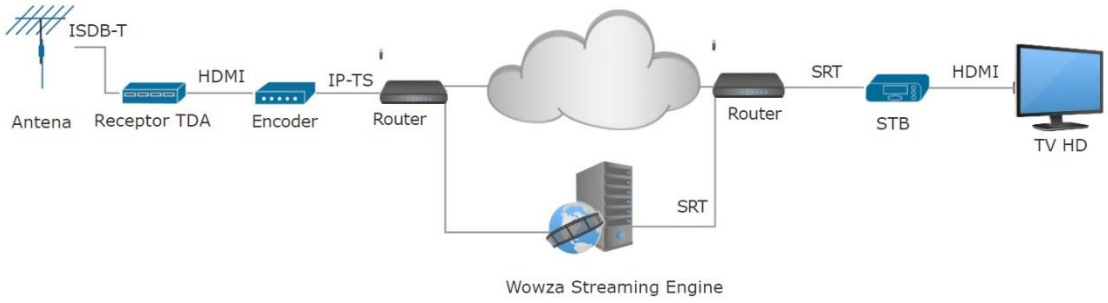


Figura 20

Elementos presentes en el escenario

Servidor de contenido: En este caso se utilizaron señales de canal 12 de córdoba que ya estaban cargadas en la PWSE.

Servidor Wowza: Esta montado en un PC Debian 8 “Jessie”, alojado en las instalaciones de Colsecor Lta. de la Ciudad de Córdoba. Recibe la señal de canal 12 HD por

medio de transmisión multicast en cabecera y fue el encargado de enviar el contenido en el protocolo SRT. Su administración se realiza mediante una interface web.

Receptor SRT: se implementó la distribución OSMC en una placa Raspberry Pi 2.

Para adecuar a los requisitos del sistema, se accedió a dicha placa mediante el servicio SSH. Una vez dentro del sistema, de igual manera que en Ubuntu 18, se procedió a la descarga de los repositorios y a la compilación de las herramientas provista por Haivision en GitHub.

Con este escenario final se pretende dar el cierre a la PPS completando todo el circuito que comprende la transmisión desde el origen a del contenido hasta la reproducción del mismo en el hogar del usuario final. Con lo cual se pretende de esta experiencia lo siguiente:

- Configuración de las instancias participantes hasta el usuario
- Conexión mediante el protocolo SRT entre plataforma y receptor del usuario final atravesando internet

Desarrollo de la experiencia

El principal objetivo de esta última experiencia es lograr alojar el receptor SRT en una plaqueta *Raspberry Pi* con un sistema operativo embebido apropiado, de manera tal que se pueda independizar del uso de una computadora de propósito general. Además, se buscó la menor intervención posible por parte del usuario final.

Configuraciones

Receptor SRT: Como se comentó anteriormente, se alojó en una Raspberry Pi. En este caso, se optó por crear un script en bash para que automatice la configuración y ejecución de los comandos necesarios. Además, se incluyó la instrucción de ejecución de dicho script en el inicio de sistema de manera automática. Esto en búsqueda de hacer que el proceso de conexión no sea tarea del consumidor final del contenido.

Esto fue realizado de la siguiente manera:

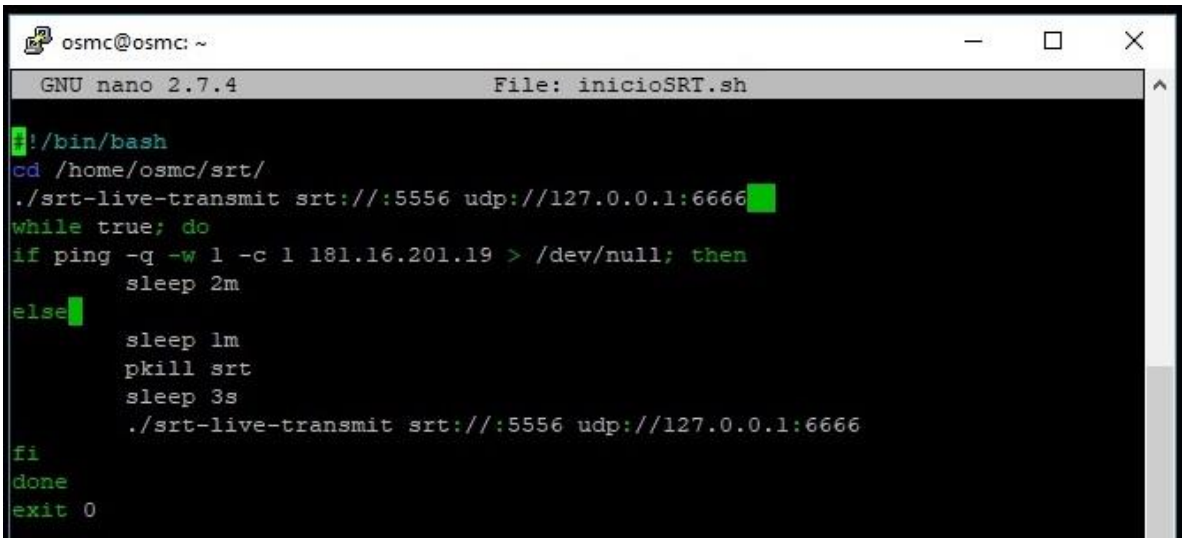


Figura 21

Con la adición de estas líneas, se configuraron los parámetros necesarios para poder detectar la interrupción del servicio de Internet, y en base a eso, poder reiniciar tanto los comandos de conexión como dar de baja los servicios relacionados con el protocolo SRT.

Esto debe ser realizado debido a que, si bien el protocolo es capaz de levantar nuevamente la conexión en caso de que exista un corte de servicio, de la manera que se configuro su funcionamiento en esta plataforma, existe un solo proceso de negociación.

En tanto, en la interface de OSMC, se configuró la carpeta en la cual va a estar disponible el contenido.

Luego de haberse realizado todo el proceso de configuración, el usuario final solo debe seleccionar el archivo correspondiente al contenido que desea ver.

En una actualización posterior, se configuró la reproducción automática del contenido.

Configuración de un streaming SRT

Como es de esperar, como lo que se modificó fue la etapa de recepción, no fue necesario configurar nuevamente la instancia en la PWSE.

Pruebas de funcionamiento del sistema final

A modo de comprobación del correcto funcionamiento del escenario, se dispuso de una pantalla capaz de reproducir contenido HD y a la que se conectó la plaqueta Raspberry Pi.

Receptor alojado en la ciudad de Rio Cuarto: La prueba tuvo una duración de 4 horas, en las cuales se dejó trabajar la Raspberry Pi sin interrupción, reproduciendo el contenido recuperado de la PWSE de COLSECOR.

Dado que se disponía de un control de ancho de banda en el Router Mikrotik en el domicilio, se constató que para la reproducción en HD (720p) se requería a la red un ancho de banda de 7 Mbits de internet.

El objeto de esta prueba fue probar el desempeño del protocolo SRT atravesando Internet y la reproducción en la Raspberry Pi.

Receptor situado en las instalaciones de Córdoba capital: Se dispuso de una pantalla de 32 pulgadas, con capacidad de reproducir contenido en FullHD(1080p). En este caso se dispuso de una Placa Raspberry Pi 3 que se conectó con el TV mediante conexión HDMI.

El sistema se dejó funcionando en condiciones de congestión durante 2 días a modo de testeo, esto para comprobar su robustes.

Con esta prueba se buscó depurar errores de reproducción, así como también el desempeño del protocolo en condiciones de alta congestión y en pruebas de larga duración.

A continuación, se analizarán los resultados de todas las implementaciones.

Resultados

En las siguientes imágenes se analizan los resultados obtenidos al realizar la comparación entre una reproducción que utiliza el protocolo SRT y otra que utiliza MPEG-TS.

Se hicieron pruebas en cada una de los escenarios planteados, con los cuales se obtuvieron los siguientes resultados:

Escenario Wowza Local:

Recordar que esta red ha sido modelada con la herramienta NetEm para emular condiciones de una red pública con congestión, perdida de paquetes, latencia, jitter, etc.

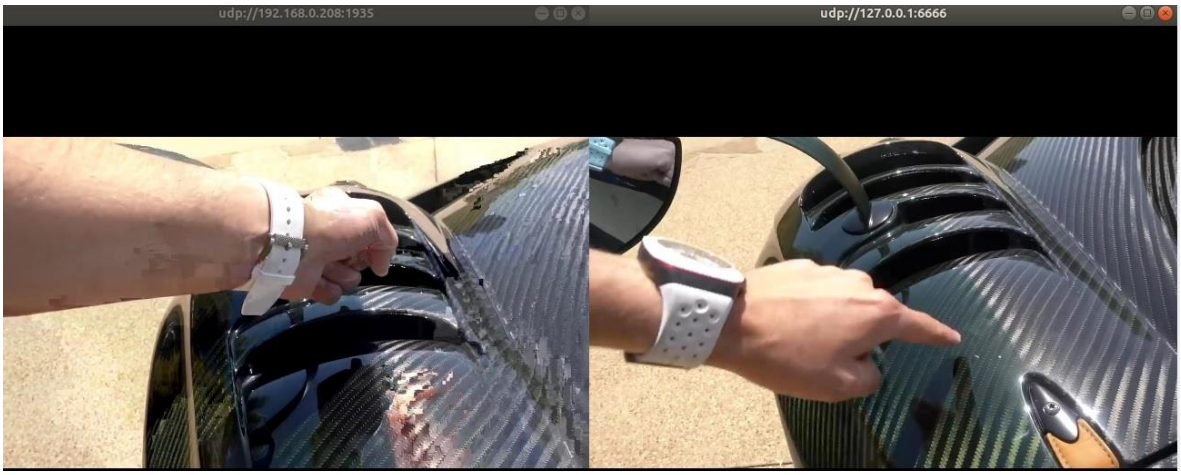


Figura 22

Como puede apreciarse, en idénticas condiciones de red, se aprecian diferencias en la calidad de reproducción, en este caso, fueron leves distorsiones.

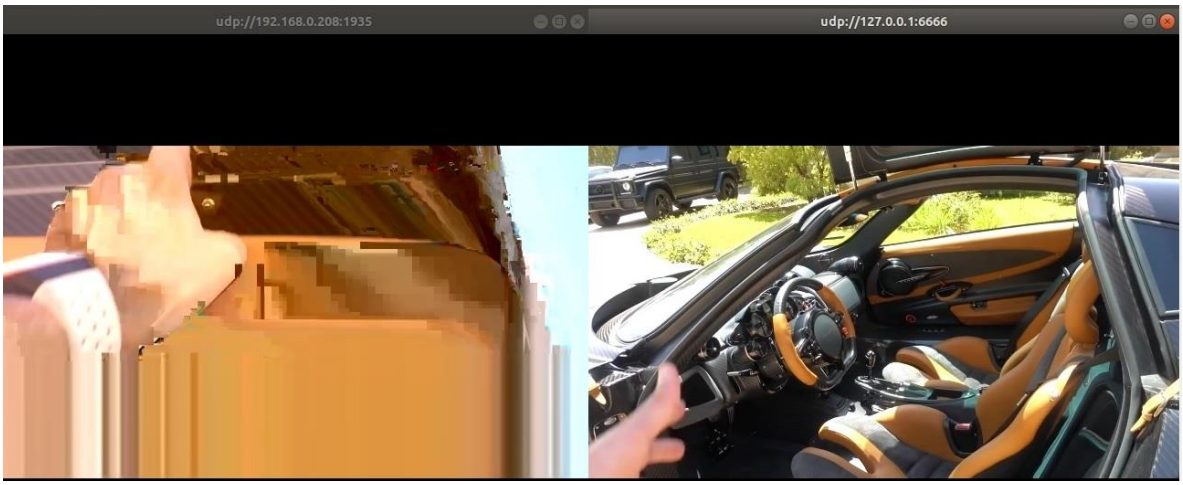


Figura 23

En otros casos más graves, las variaciones en la red provocaron que la reproducción del contenido en MPEG-TS presente errores más importantes, que se visualizan mediante "Pixelado" de las imágenes al punto de corromper totalmente la imagen a reproducir.

Este caso tuvo solo la finalidad de comprobar el funcionamiento de SRT, además de comprobar las ventajas respecto a una transmisión de MPEG-TS estándar en condiciones de red desfavorables.

Escenario Wowza Colsecor

En este caso, la comparación se hizo directamente entre dos instancias alojadas en la misma PC.

Figura 24

La forma de realizar la comparación de los rendimientos de cada protocolo fue no solo integrarlos en una red no administrada y de uso público, sino que además se le aplicaron restricciones en el ancho de banda disponible, a fin de exigir más rendimiento a los protocolos.

Como se observa en la *Figura 24*, el rendimiento de SRT es muy bueno, logrando mantener la calidad, así como también la fluidez de la reproducción.

La transmisión con MPEG-TS puro presenta inconvenientes de reproducción, que son el resultado de la pérdida de paquetes y el Jitter presente en la red, y esto agravado por la limitación de ancho de banda.

La estabilidad de la reproducción del contenido transportado por SRT fue notable, permitiendo una reproducción continua e ininterrumpida, pese a las limitaciones de ancho de banda y condiciones desfavorables antes mencionadas.

Escenario Wowza Colsecor a RP en usuario

Habiendo probado el funcionamiento de la transmisión SRT desde las oficinas de Colsecor hasta el usuario final, solo restaba comprobar si se podía reemplazar la instancia de recepción por una solución que fuera más práctica que la de disponer de una computadora con Linux. Además, como se detalló en la implementación, se automatizo el proceso de conexión y reproducción, haciendo que un hipotético usuario solo deba encender la RP.

- **Receptor alojado en Río Cuarto:**

Los resultados obtenidos fueron los siguientes:

Figura 25

Como puede observarse, la calidad de reproducción fue alta, pese a las limitaciones de Hardware que posee la plaqueta. Además, se realizaron pruebas de caída de enlace, las cuales pudo solventar gracias al script que se detalló en la implementación.

La facilidad de manipulación y de funcionamiento, sumado a la gran calidad de reproducción hicieron que la implementación fuera exitosa. Esto permitió cerrar el circuito desde el servidor hasta la reproducción final del contenido.

Cabe destacar que la elección de la distribución embebida en la plaqueta Raspberry Pi fue un aspecto fundamental, ya que se hicieron pruebas modificando el sistema operativo original, Raspbian, que arrojaron resultados desastrosos en lo referido a la decodificación y reproducción del contenido.

OSMC (Open Source Media Center) es una solución enfocada en la reproducción de contenidos, por lo cual fue apropiada en este caso, y sobre todo porque se requería la reproducción de video en alta calidad.

- **Receptor situado en Córdoba Capital:**

Como es de esperarse, los resultados fueron similares a los obtenidos en el receptor ubicado en Rio Cuarto, por lo cual se le realizaron pruebas para pulir aún más el rendimiento, durante las cuales se monitoreo la utilización de los recursos de la placa. (*Ver Anexo HTop*)

Una vez iniciado el sistema, se discutió con el personal acerca de pequeñas deficiencias en la calidad de la reproducción de contenido en la pantalla. Esto motivó la reconfiguración de la resolución de salida de video. Se cambió la configuración de fábrica para forzar la salida de video a Full HD, es decir 1080p. Este cambio produjo una mejora sustancial en la calidad de reproducción del contenido. Con lo cual se descartó un error en la recepción del protocolo SRT, ya que las pequeñas distorsiones que se detectaron en la pantalla eran debido a cuestiones de resolución del terminal final, y no a errores asociados con la estabilidad del protocolo.

Además, se dejó en funcionamiento durante el transcurso de dos días en las instalaciones de Colsecor y se comprobó que la transmisión no sufrió cortes, y en el caso de que los hubiera sufrido durante algún periodo en el cual no haya sido supervisado, se logró recuperar exitosamente debido al script de reanudación de servicio que se explicitó anteriormente.

Mejoras

Una vez cumplido el objetivo principal de la práctica, que era probar el funcionamiento de una transmisión de una señal de TV desde los servidores de Colsecor hasta el usuario final a través de internet, se planteó la idea a una extensión a lo desplegado. El equipo técnico de Colsecor se mostró muy satisfecho con el resultado, por lo cual continuaran las pruebas con los desarrollos realizados en la PWSE y la RP. En un futuro se planea configurar la RP para que permita sacar por una de sus interfaces una señal MPEG-TS mediante Multicast. Esto con la finalidad de ser utilizado en un nodo de distribución en sus redes de televisión.

Conclusiones

El trabajo asignado fue muy interesante y desafiante por el hecho de que son tecnologías que están en proceso de desarrollo, por lo cual no existe documentación formal sobre los elementos a utilizar.

Si bien se desarrollaron la mayoría de las actividades en la Facultad de Ingeniería, en los días de trabajo que transcurrieron las instalaciones de Córdoba fui rápidamente integrado al grupo de trabajo, y recibí asistencia e instrucción tanto de mi tutor como del resto del personal de Colsecor. Se me explicaron aspectos técnicos del funcionamiento de la empresa, así como también tuve la posibilidad de tomar contacto con el equipamiento que formó parte de la PPS. Fue agradable que sugerencias y recomendaciones que resultaron de la finalización de la práctica fueran tenidas en cuenta, así como también sentir que lo trabajado está siendo analizado como una línea de trabajo para actualizar las tecnologías que utiliza Colsecor actualmente.

Respecto a la experiencia adquirida, fue una instancia de mucho aprendizaje, de investigación sobre tecnologías innovadoras y de formación respecto a la vida laboral. Fortaleció estrategias de identificación y solución de los inconvenientes que se fueron presentando, así como también la consulta directa al trabajo continuo de los desarrolladores de protocolos.

Anexos

Glosario

A continuación, se definen conceptos que serán de utilidad a lo largo del desarrollo de la PPS.

IPTV: Internet Protocol Television, hace referencia a sistemas de distribución por suscripción de señales de televisión de pago usando conexiones de banda ancha sobre el protocolo IP.

Latencia: Es la suma de retardos temporales dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red.

Jitter: Fluctuación del retardo a la variabilidad temporal durante el envío de señales digitales, una ligera desviación de la exactitud de la señal de reloj. El jitter suele considerarse como una señal de ruido no deseada.

Socket: Los sockets de Internet constituyen el mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos apropiados. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.

Streaming: Es la distribución digital de contenido multimedia a través de una red de computadoras, de manera que el usuario utiliza el producto a la vez que se descarga. La palabra retransmisión se refiere a una corriente continua que fluye sin interrupción, y habitualmente a la difusión de audio o video.

Stream: Flujo individual de una transmisión de contenido multimedia.

Multicast: Es un método de envío simultáneo de paquetes a nivel de IP que tan sólo serán recibidos por un determinado grupo de receptores, que están interesados en los mismos. Envío uno a algunos (grupo)

Unicast: El método de transmisión unicast es uno a uno (one-to-one), con este método el envío de datos se realiza desde un único emisor a un único receptor

MPEG-TS: Es un estándar de contenedor digital para la transmisión y almacenamiento de audio y video, entre otros. Es usado en sistemas de distribución como pueden ser DVB, ATSC e IPTV.

Set-Top-Box: Es el dispositivo receptor o decodificador de las señales (analógicas o digitales) de televisión analógica o digital (DTV), para luego ser mostrada o visualizada en el televisor.

FEC: Es un tipo de mecanismo de corrección de errores que permite su corrección en el receptor sin retransmisión de la información original.

Gateway: También conocido como puerta de enlace es el dispositivo que actúa de interfaz de conexión entre aparatos o dispositivos, y también posibilita compartir recursos entre dos o más computadoras.

Firewall: Se trata de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar o descifrar el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios

NAT: Es un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles.

Filtrado de paquetes: Es una herramienta que utiliza el firewall. Consiste en un software que examina la cabecera de los paquetes de datos según van pasando y básicamente decide si descartar o aceptar el paquete en cuestión. Sirve para control y seguridad de la red que se intenta proteger.

Raspberry Pi: Es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo coste desarrollado en el Reino Unido con el objetivo de estimular la enseñanza de informática en las escuelas. Sus recursos son muy limitados en comparación con una computadora convencional.

Bash Scripting: Hace referencia a la creación de secuencias de instrucciones que permitan realizar acciones por consola en sistemas GNU Linux. Permite automatizar acciones del sistema.

Túnel GRE: Generic Routing Encapsulation (GRE) es un protocolo de túnel que fue originalmente desarrollado por Cisco. Se utiliza para encapsular una amplia variedad de protocolos a partir de la creación de un enlace virtual punto a punto.

Configuraciones implementadas

Escenario LOCAL con servidor Wowza local:

En este escenario se utilizaron estas configuraciones:

Servidor de video:

ffmpeg -re -i Downloads/video.mp4 -vcodec libx264 -acodec aac -ar 48000 -strict experimental -f flv rtmp://hola:lucho@192.168.0.205:1935/liveSource2/salo

Cliente MPEG-TS:

ffplay –f mpegts –i udp://192.168.0.205:6667

Escenario Wowza córdoba Linux generico

Configuración SRT Wowza:

Source stream Name: Canal12.stream

Destination Host: 190.193.83.156

Destination Port : 5556

Configuración Receptor:

./stransmit srt://:5556 udp://127.0.0.1:6666

ffplay udp://127.0.0.1:6666

Si se usa VLC:

udp://@:6666

Configuración Router Mikrotik

[admin@MikroTik] ip firewall nat> add action=dst-nat chain=dstnat \
dst-address=190.193.83.156/32 to-addresses=192.168.64.204

[admin@MikroTik] ip firewall nat> add action=src-nat chain=srcnat \
src-address=192.168.0.4/32 to-addresses=190.193.64.204

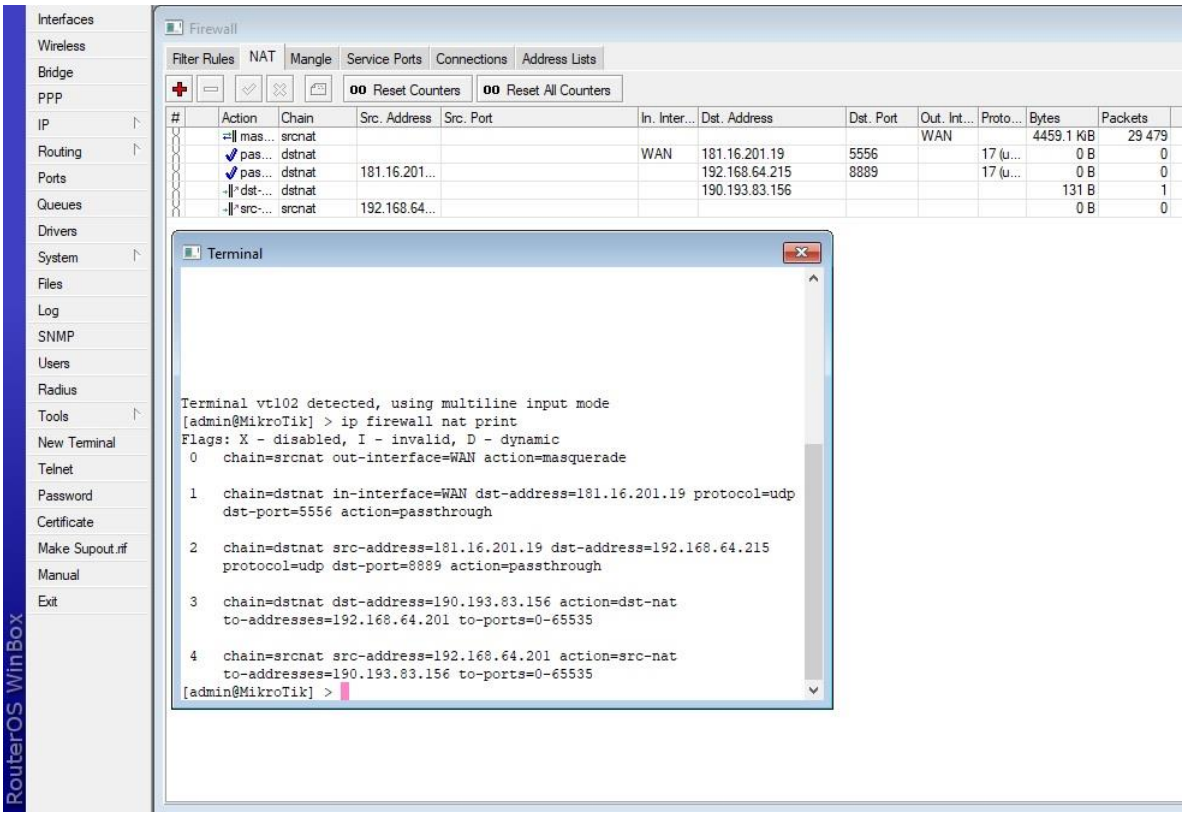
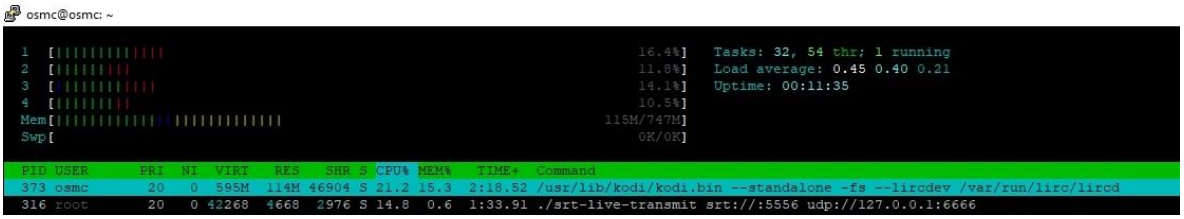


Figura 26

Control de uso de recursos en la RP

Herramienta htop: htop es un sistema de monitorización, administración y visor de procesos interactivo. Es una alternativa más intuitiva, interactiva y funcional del conocido como Top, incluido en sistemas operativos de tipo Unix

Instalación de htop: sudo apt-get install htop



Descripción de HLS

HLS es un protocolo utilizado para enviar audio y video sobre HTTP, soporta bitrate adaptivo y es compatible con la mayoría de los dispositivos. Este protocolo fue desarrollado por Apple y puede ser utilizado tanto para video en vivo como video en demanda.

Características:

Se adapta a las condiciones de la red. Dependiendo el ancho de banda de internet del espectador, se le entregará una calidad o bitrate adecuado.

Además, puede ser entregado desde un servidor tradicional de HTTP.

Y puede ser protegido con encriptación o autenticación. Esto con el fin de proteger derechos de autor.

Flujo de HLS: Primero el archivo de video se debe codificar con compresión de video H.264 y compresión de Audio AAC. Posteriormente se debe seleccionar MPEG2-TS como contenedor de video, esto lo hace normalmente un encoder de video. Una vez codificado el video se realiza un proceso de segmentación. Este proceso realiza cortes del video cada cierta cantidad de segundos, usualmente cada 10 segundos. A estos fragmentos se les conoce como segmentos o *chunks* de video.

Una vez que el archivo se encuentra segmentado se genera un archivo índice. Este archivo contiene la información de los segmentos y en donde se encuentran para que el *player* los descargue. Este mismo proceso se realiza tanto para video en vivo o video en demanda (video almacenado). Cuando se decide encriptar la información, se debe ejecutar un algoritmo para proteger cada uno de los segmentos. Y sólo el *player* que tenga la llave para desencriptarlos podrá ver el contenido. Estas llaves pueden ser estáticas o dinámicas, puede generarse una llave para todos los videos o una llave por video.

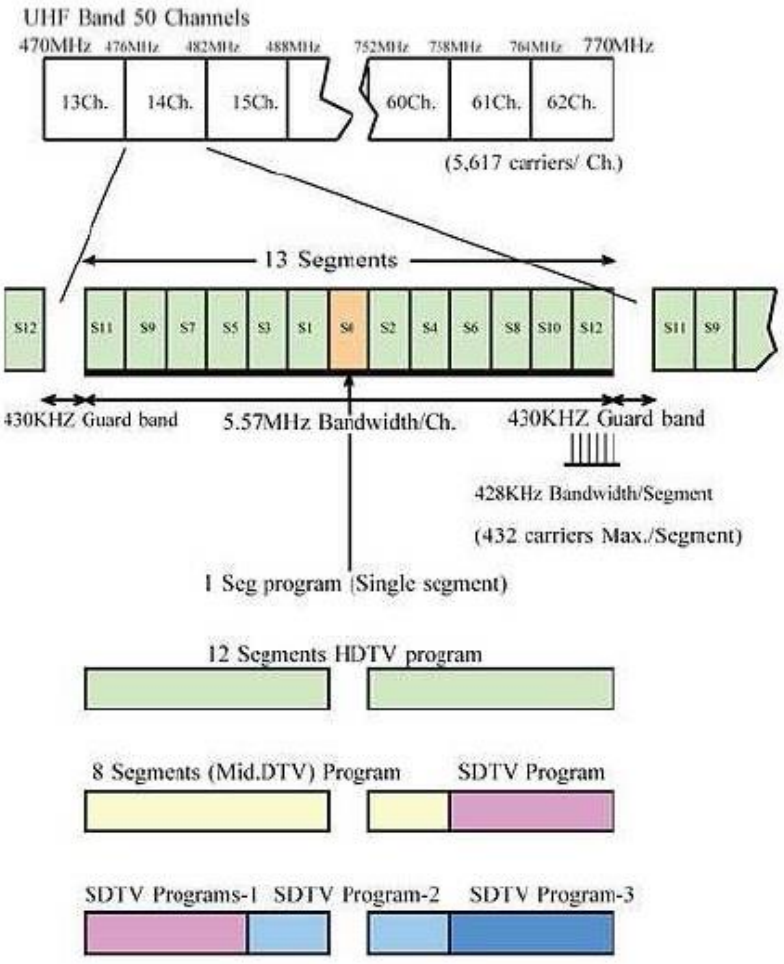
Lo único que hace falta es compartir el URL del archivo índice que es algo como: http://nombre-del-dominio/directorio-del-video/mi-video.m3u8.

Norma ISDB-t

La norma ISDB-t, fue adoptada previamente por Brasil y luego por el resto de los países de sudamérica excepto Colombia. En EEUU existe como norma la ASCT y en Europa la DVB-T, la decisión tiene razones políticas, las otras normas fueron pensadas para servicios pagos de televisión. Acá se privilegió brindar un servicio gratuito y de calidad que además permitiese ver la televisión a través de dispositivos móviles como los celulares.

El ancho de banda actual de un canal (por ejemplo el 7) analógico es de 6MHz, con la tecnología digital ahora en ese ancho de banda vamos a poder tener hasta 5 señales con la misma calidad y mejor (sin lluvias ni fantasmas) que los canales analíogiocos.

Esos 6 MHz se dividen en 13 segmentos de información más otro de "guarda" (separación entre canales). Uno de esos segmentos, el central de la banda, llamado one-seg, transmite la información necesaria para poder ver la televisión en forma móvil, con resolución adecuada para esos dispositivos.



Los formatos de codificación de audio y video son respectivamente el AAC y MPEG2/4 que permiten que en ese ancho de banda de 6MHz viaje más información, siendo posible que se transmitan hasta seis señales de calidad estandar en un canal (también llamado MUX). Como ejemplo en el canal 25 conviven las señales CN23, C5N, telesur, 360TV, Construir y también el one-seg CN23.

Encriptación de SRT

SRT aplica la encriptación y provee la recuperación de errores. Antes de la decodificación (o trascodificación), SRT descifra el stream y permite la recuperación de paquetes perdidos, muy normales en las conexiones a internet.

Los comandos utilizados en el receptor fueron los siguientes:

```
./srt-live-transmit srt://:5556 udp://127.0.0.1:6666&passphrase=1234567890123456
```

Estte comando es asi debido a que en la instancia en la PWSE se habilito la encriptación AES-128, que utiliza claves de 16 caracteres. En este caso, se configura en el parámetro passphrase la palabra clave que desencripta el contenido.

Actualmente no está habilitado para la implementación desde los repositorios de GitHub, pero si en los encoders comercializados por Haivision, como lo es Makito X.

Bibliografía

Página oficial SRT: https://www.srtalliance.org/
Página del impulsor de SRT: https://www.haivision.com/products/srt-secure-reliable-transport/
Repositorios del código: https://github.com/Haivision/srt

Descripción protocolo UDT: https://tools.ietf.org/html/draft-gg-udp-02#page-5

Protocolos de IPTV:https://es.scribd.com/presentation/31392571/Protocols-in-IPTV

Manual de utilización de Wowza: https://www.wowza.com/docs/wowza-streaming-engine-product-articles

Documentación de herramientas SRT:

https://github.com/Haivision/srt/blob/master/docs/stransmit.md

Documentación herramienta FFMPEG:

https://video.stackexchange.com/questions/20495/how-do-i-set-up-and-use-ffmpeg-in-windows

Compatibilidades FFMPEG: https://ffmpeg.org/ffmpeg-protocols.html#srt

Funcionamiento protocolo SRT: https://www.headendinfo.com/srt-secure-reliable-transport/

Automatización de placa Raspberry Pi: https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/#local