COMO USAR ONPREMISE SRT SERVER

Introducción

Antes de comenzar a explicar, cómo usar *OnPremise SRT Server*, vamos a hacer una introducción para dejar bien claro, qué es, y para qué ha sido creado.

La motivación para crear *OnPremise SRT Server,* fue facilitar la tarea de crear un despliegue terrestre de repetidores para Radio y/o Televisión, con feedbacks y control en tiempo real de cada uno de sus elementos, usando la red pública de Internet, mediante el protocolo seguro y resiliente **SRT**.

¿Por qué SRT?:
- SRT es un protocolo de código abierto, muy extendido a día de hoy, y presente en la mayoría de software y hardware del mercado (VMix, OBS, VLC, Wirecast, Larix, Kiloview, Haivision, Teradek, Truen …)
- SRT puede configurarse tanto para bajas latencias, como para alta resiliencia a la congestión de red y a la pérdidas de paquetes. Cuando RTMP o HLS no pueden comunicarse en absoluto, SRT aún puede hacerlo con buffer delays inferiores a los 8 segundos. Es ideal donde el RTT (ping) es muy grande (más de 500 ms), o la pérdida de paquetes es superior al 10%.
- SRT proporciona estadísticas del medio por el que se transporta, permitiendo ajustar sus parámetros para optimizar su uso.
- SRT transporta naturalmente MPEG-TS, siendo totalmente agnóstico a los codecs. Puede por tanto transportar MPEG-2, H.264, H.265/HEVC, H.266/VVC, etc. Además de poder transportar SPTS o MPTS. Datos y señalización SCTE, y cualquier cosa que pueda viajar dentro de un Transport Stream.
- SRT permite seguridad punto a punto, mediante cifrado AES de 128, 192 y 256 bits.

Para dicha tarea, *OnPremise SRT Server* ha sido dotado con una arquitectura donde se pueden configurar entradas y salidas IP conectadas fácilmente entre sí. Así pueden haber varias entradas, y cada una de ellas copiar su contenido sobre una o varias salidas a la vez, sin retardo alguno.

Además de las lógicas entradas y salidas SRT, que son la base del servidor, se ha dotado de entradas pull de otros protocolos, con la idea de facilitar la incorporación de canales desde otros servidores que operen en esos protocolos, como UDP (unicast o multicast), RTMP, RTSP, HLS, DASH y HTTP. Esto no significa que sea un servidor RTMP, si no meramente un gateway. que puede recoger señales en esos protocolos de otros servidores. De igual manera se ha aprovechado para añadir, aparte de las salidas SRT habituales, salidas en UDP (unicast y multicast) útiles en algunas cabeceras IP, y RTMP para poder enviar copias a Youtube, Facebook y otros servidores que aún usan ese protocolo para contribución de señal.

OnPremise SRT Server repite exactamente el stream que le entra en cada salida sin modificación alguna, con los mismos PIDs, tablas SDT y EIT, EPGs, etc. No recomprime ni audio ni video, ni los retoca en ningún aspecto, porque esa no es su finalidad, de manera que lo que se envía a él, es lo que va a llegar a todas sus salidas sin modificación alguna.

El panel de *OnPremise SRT Server* representa de un solo vistazo, todos los componentes de la red de contribución y distribución (emisores, receptores, etc), su estado en tiempo real segundo a segundo e información estadística valiosa. Hay absoluto control sobre cada uno de ellos, se puede acceder a sus paneles remotos, se les puede deshabilitar o habilitar con un simple click, o borrar

para siempre, y también editar sus conexiones. A parte, todas las entradas pueden ser monitorizadas mediante un watchdog para saber si llevan el audio silenciado, lo que es útil para detectar fallos en los feedbacks, o en las fuentes, avisando mediante una alerta visual en rojo. En él se puede seguir el rastro tanto de los componentes conectados como de los desconectados, a diferencia de un servidor multimedia tradicional.

El que haya sido orientada para su despliegue in situ (en el propio estudio o cabecera), y el tipo de recursos necesarios para su operativa, hacen que solamente arranque en equipos reales físicos. Aunque la imagen ISO pueda instalarse en una máquina virtual, nunca arrancará en ésta, y es algo que no tenemos previsto que haga ningún momento del futuro. Recomendamos su instalación en equipos de al menos 4 threads CPU de 1.5 GHz o más, con arquitectura de 64 bits y 4 GB mínimo de RAM. Lea el documento HARDWARE_SPA.pdf acerca del hardware recomendado para su instalación.

Cada elemento que participa en la red mediante SRT, establece una conexión punto a punto con el servidor, que es única, totalmente asegurada y monitorizada por separado de todas las demás. Por tanto, dicha conexión la hace mediante un puerto UDP totalmente diferente de cualquier otra conexión. Por tanto, no multiplexamos diferentes conexiones SRT en un mismo puerto UDP, por esta razón, ni tenemos pensado en hacerlo en un futuro, porque ese no es el objetivo con el que se ha creado *OnPremise SRT Server*.

Este software lleva una opción de actualización sencilla. Cuando hay alguna versión nueva, el software se la baja, pero no la aplica hasta que el usuario no pulsa en el botón actualizar. La actualización se puede efectuar sin necesidad de parar la operativa del server. El sistema recarga en caliente, y los equipos reconectan a él en menos de 2 segundos. Las actualizaciones irán corrigiendo fallos y añadiendo nuevas funcionalidades de interés para el objetivo que tiene este software.

OnPremise SRT Server expone un API REST, para que pueda integrarse con otros programas o procesos externos que quieran trabajar con él. Dicho API viene auto documentado con Swagger 2.0, aunque será motivo de otro documento para explicar su funcionamiento.

OnPremise SRT Server es un software que se instala junto con un sistema operativo Linux Debian de 64 bits, y se accede a él mediante un navegador web desde cualquier otro equipo conectado a la red. De esta forma, se puede garantizar una funcionamiento estable y seguro durante años, en un entorno de trabajo continuo 24/7.

En este manual, vamos a explicar cómo crear buenas conexiones SRT entre los diferentes componentes de la red y este servidor. Vamos a trabajar con los componentes más típicos del mercado, y nos enfrentaremos a las situaciones más normales en un estudio o en una cabecera.

Comencemos.

Configuración inicial del sistema

Antes de empezar a montar enlaces, tenemos que establecer la configuración del servidor, una vez se haya instalado el software, conforme a las instrucciones del documento INSTALL_SPA.pdf .

Lo primero será poner una IP local fija al servidor. Es recomendable usar una que esté fuera del pool de DHCP, es decir que no sea de las que asigna el DHCP dinámicamente, y que esté libre. Así por ejemplo, si el DHCP entrega IP desde 192.168.1.100 en adelante, pues podemos elegir una libre por debajo de ese valor, como 192.168.1.68 . El administrador de la red, debe de saber qué parámetros deben configurarse aquí. Para ello, una vez logeados como admin en el panel del servidor, vamos a la sección Network, pulsamos sobre LAN/WLAN Interfaces y ponemos el Mode en Static, la IP y la máscara en Address/Mask, en nuestro ejemplo sería 192.168.1.68/24 y el Gateway que corresponda. También es importante configurar al menos un DNS en la sección LAN/WLAN DNS.
Seguidamente en la misma pantalla, nos vamos a Other Options, y vamos a limitar los puertos UDP que vamos usar en este servidor. Es más que suficiente asignarle un rango de 2000 puertos, como por ejemplo 5000-7000 ó 7000-9000. Para ello ponemos en UDP Ports dicho rango y pulsamos Save.

De esta manera nos aseguramos de que no vamos a crear ninguna conexión SRT Listener fuera de ese rango, ya que el software nos lo indicará.

Para que dichos puertos sean accesibles desde fuera de la red privada, es necesario crear una regla NAT en el router que dirija dicho rango de puertos UDP (no los TCP), a la IP local de nuestro servidor. Si no sabe cómo hacerlo, pídale al administrador de la red que lo haga.

Una vez, toda la sección de Network está configurada, nos vamos a la sección System y pulsamos sobre el botón Reboot, y luego en Logout. El equipo se reiniciará con los nuevos parámetros de red, por lo que en su navegador tendrá que poner la nueva IP fija de su servidor para poder logearse en el mismo.

El último paso es disponer de una dirección fija pública para su conexión a Internet. Si su IP pública no es fija, es conveniente crear una cuenta en No-IP.com o en Dyndns.com, crear un hostname para su conexión de IP dinámica y configurarla en el bloque DDNS. De esta manera su dirección desde el exterior ya no sería una IP que cambia, si no una dirección concreta y fija como por ejemplo: srtlinks.dyndns.org.

El sistema ya esta preparado para crear todos los enlaces que necesitemos, desde dentro y fuera de la red local. Vamos a ello.

Enlaces SRT

A partir de ahora, todo nuestro trabajo va a estar en la sección Streams. Inicialmente estará totalmente vacía con aspecto como éste.

Para empezar a crear nuestra red tenemos que empezar a añadir al menos un INPUT. Esto se hace pulsando en ADD INPUT, que nos abrirá una ventana que tendremos que rellenar con una serie de datos. Pero antes de eso vamos a explicar un poco aspectos generales de esta sección.

Todas las fuentes de video IP, son INPUT, y pueden ser desde un encoder de video u otro servidor que envía señal en SRT, o un servidor de otros protocolos del que recogemos la señal (pull). Sea como fuere, hay una serie de parámetros comunes a todos los INPUT que son:

Protocol: protocolo de comunicaciones usado (SRT en nuestro caso).
Name: servirá para reconocer el origen de la señal, con un nombre descriptivo que se verá en el panel. Si se deja en blanco tomará los valores Channel 1, y así sucesivamente.
Provider: no se verá en el panel, si se deja en blanco tomará el valor TodoStreaming.
Location: no se verá en el panel y es opcional, pero nos puede servir para saber la localización geográfica de nuestra fuente INPUT.
Remote: es también un campo opcional, en él podemos poner la URL http de acceso al panel de control del equipo fuente. En el caso de que rellenemos este campo, aparecerá un icono donde podremos pulsar para ir directos al panel de control de dicho equipo.
Wtdg Timeout: Es un valor numérico entre 0 y 3600 segundos, que indica el tiempo en segundos de silencio de audio que harán saltar la alarma watchdog de este INPUT.
Watchdog: es un checkbox que si lo activamos, activa la alarma descrita anteriormente.

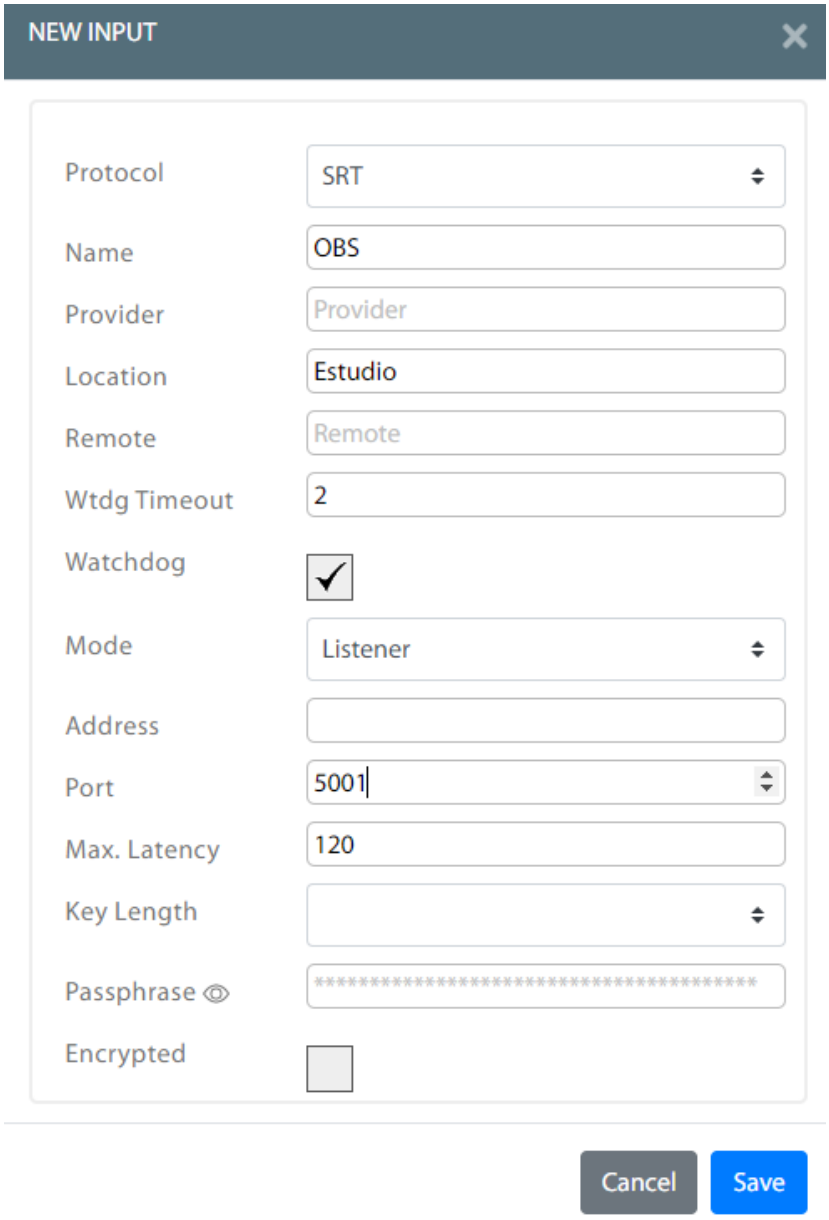
El resto de campos de un INPUT son propios de cada protocolo. Vamos a explicar en primer lugar el protocolo SRT, que es la base de este servidor.

SRT tiene 3 modos (caller, listener y rendezvous). En este manual solamente trataremos los dos primeros. Siempre en un enlace punto a punto SRT una parte será caller y la otra listener. Por simplicidad la parte del servidor INPUT será Listener y escuchará en uno de los puertos UDP del rango configurado en la sección anterior. Por sencillez podemos usar 500x donde x será el número del INPUT. Por ejemplo 5001 para el INPUT 1. Vamos ir viendo cada campo del protocolo SRT por separado:

Mode: en el servidor será siempre Listener, salvo que queramos conectarnos con otro server que tiene un SRT Listener, en cuyo caso tendremos que hacerlo mediante un Caller.
Address: en el caso de usar un SRT Listener, podemos dejarlo vacío ya que este se ignorará, puesto que el servidor escuchará dicho puerto en todas sus interfaces de red. En el caso de usar un SRT Caller, tendremos que poner la IP del equipo que nos espera escuchar con el SRT Listener.

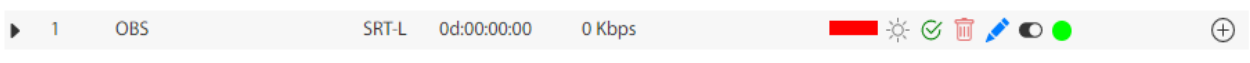
Port: es el puerto UDP en el que escuchamos como Listener, o en el que nos escucha otro server si contamos mediante un SRT caller a éste.
Max. Latency: es la latencia máxima que deseamos en dicho enlace punto a punto. Puede ser un valor entre 20 y 8000 milisegundos. Explicaremos más a fondo este parámetro más adelante.
Key Length: en el caso de que queramos cifrar la comunicación entre ambos puntos, tendremos que definir la longitud del cifrado AES (16, 24 o 32) además del resto de parámetros que siguen en ambos puntos.
Passphrase: es una clave de entre 10 y 79 caracteres a usar para el cifrado AES en ambos puntos.
Encrypted: tendremos que activarlo si queremos cifrar la comunicación entre ambos puntos.

Vamos a crear un INPUT en el que entrará la señal de video de un OBS que conectará con nuestro servidor por SRT. Para ello hacemos clic en ADD INPUT en el panel del servidor, y rellenamos los valores de esta forma:



En Name ponemos OBS para identificar el origen del video en dicho INPUT, aunque podríamos haber puesto cualquier otro nombre descriptivo que nos ayude a identificarlo fácilmente en el panel.

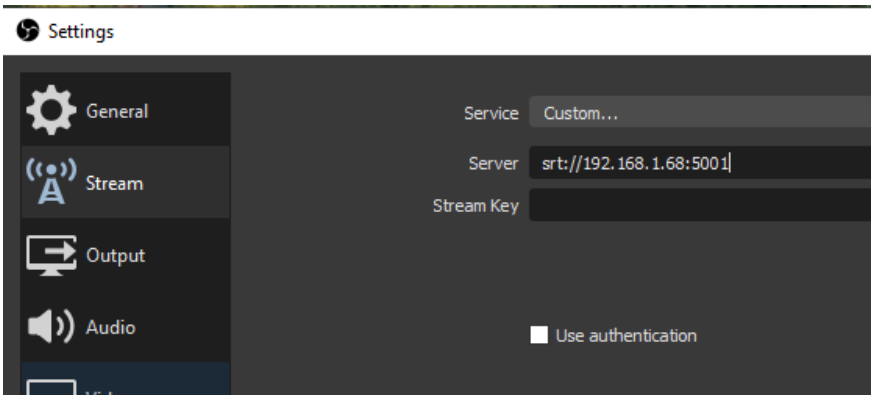
Provider lo dejamos en blanco. Location ponemos que está en el Estudio, aunque podríamos dejarlo en blanco, ya que es opcional. Wtdg Timeout lo ponemos en 2 segundos, y activamos el Watchdog o detector de silencios de audio.
Por sencillez ponemos el Mode en Listener, no hace falta poner el Address como explicamos anteriormente. Escucharemos en el puerto UDP 5001, usando el criterio 500x para INPUT donde x es el número del mismo. En Max. Latency dejamos el valor por defecto de 120 ms, que posteriormente evaluaremos con más detalle. Y no vamos a cifrar la conexión por simplicidad. Hacemos clic en Save, y aparecerá algo así en el panel.



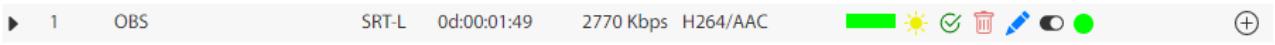
De izquierda a derecha vamos a describir cada elemento visual con detalle.

El triángulo negro de la izquierda, se usa para expandir o colapsar los OUTPUT que copian este INPUT. Esto quedará más claro cuando hayamos creado al menos un OUTPUT más adelante. Después le sigue el número del INPUT, que en este caso por ser el primero es 1. Seguido viene el nombre que hemos puesto a este INPUT para distinguirlo en el panel, en este caso OBS. Después se indica el protocolo que va a usar el servidor para este INPUT, que en nuestro caso es SRT-L de SRT Listener. Luego viene el tiempo de conexión sin interrupciones, que de momento es 0, ya que aún no hay transmisión alguna. Le sigue el bitrate en Kbps de la transmisión. Seguidamente hay una barra de color que indica el estado de la conexión. En rojo significa que no hay conexión, en verde que la conexión es buena, y en colores intermedios amarillo o naranja si es de baja o muy baja calidad. Dejando el puntero sobre dicha barra, se muestra una ventana emergente con valores informativos de la conexión y/o de la fuente. Luego hay una bombilla que estará apagada si no hay conexión establecida, o encendida si la hay. Después hay un checked que es un botón que nos permite desactivar este INPUT, de manera que aunque hubiera conexión, esta se interrumpiría, hasta que volviéramos a pulsar sobre él para restablecerla. Luego viene un icono para borrar para siempre este INPUT. Si lo pulsamos por error, todavía podemos pulsar Cancelar en la ventana de aviso emergente que sale para avisarnos de las consecuencias de borrar esta entrada. Seguido viene el icono de edición (un lápiz), que nos permite cambiar los valores que queramos de este INPUT. Luego viene un switch para desactivar o activar el watchdog de audio, y al lado un led que estará de color negro, si el watchdog está desactivado, en verde si está activado y no ha saltado ninguna alarma de silencio, o en rojo si ha saltado la alarma de silencio, y no ha vuelto aún el sonido. Totalmente a la derecha hay un botón para añadir un OUTPUT a este INPUT a donde enviar la señal que entra por INPUT.

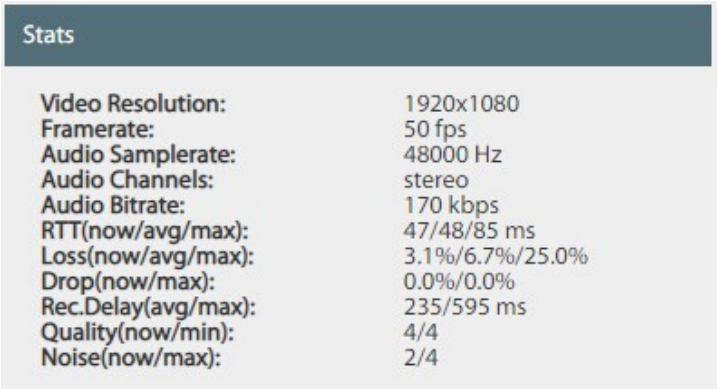
Vayamos a OBS para enviarle video a este INPUT del Server. OBS usa una URL para describir su conexión SRT tipo caller, así como lo hacen otros softwares como VLC.(srt://IP_del_Server:puerto) En nuestro ejemplo la IP del server era 192.168.1.68, y el puerto Listener que hemos usado el 5001, por lo que quedaría así:



Al comenzar a enviar el streaming, veremos algo como esto:



El analizador de entrada se pondrá en funcionamiento tras un tiempo y reconocerá el codec de video y audio usado (H264/AAC en nuestro ejemplo), y analizará los silencios de audio.
Si nos ponemos sobre la barra de color verde, podremos ver la información Stats, que se verá como esto:



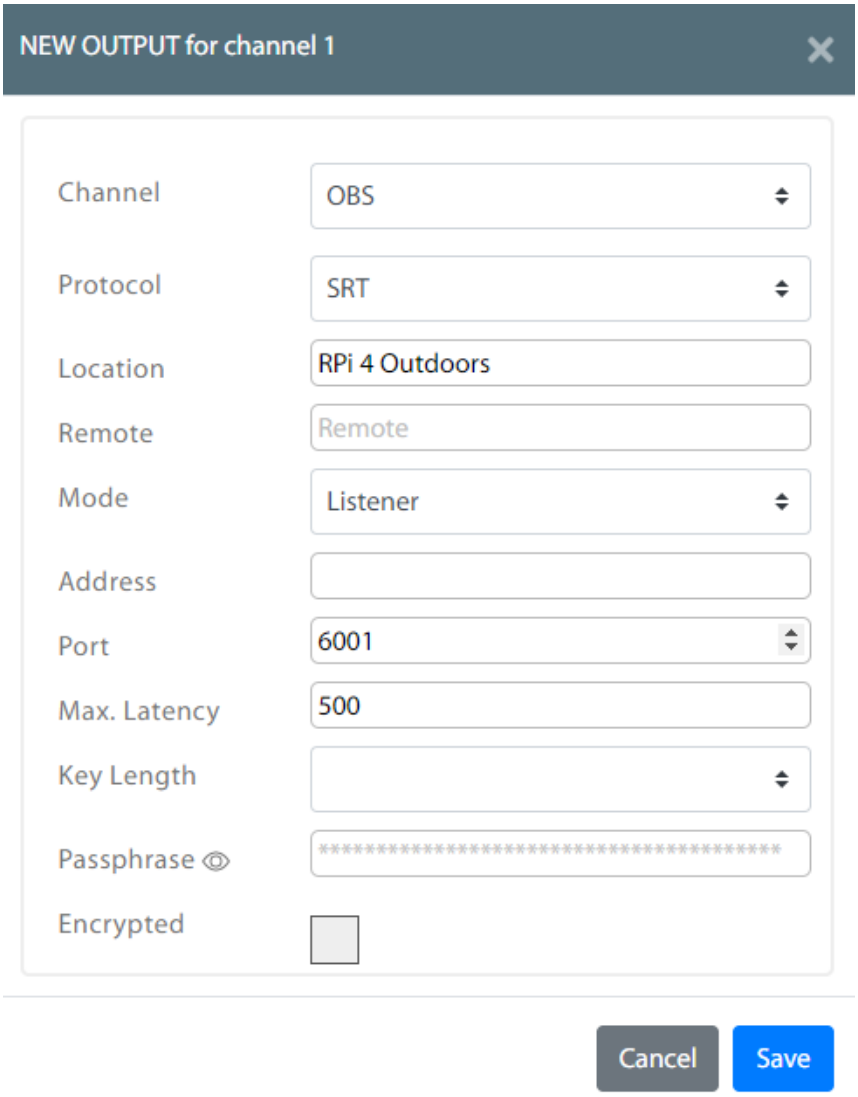
Los valores estadísticos de la conexión SRT son creados cada segundo, y los explicamos a continuación:
RTT: es el ping del segundo actual, el ping promedio de la última hora y el ping máximo de la última hora.
Loss: nos dice la pérdida de paquetes en el último segundo, el promedio en la última hora y el máximo en la última hora. Esto nos indica el ruido que hay en la red, y de estas pérdidas la gran mayoría de paquetes será recuperado por SRT si tiene el suficiente buffer delay.
Drop: nos dice los paquetes irrecuperables en el último segundo, y en máximo en la última hora. Como se puede ver en esta captura, a pesar del loss tan importante, todos los paquetes se han recuperado, y la imagen no se ha visto afectada con cuadros de colores o manchas.
Rec. Delay; es el buffer delay recomendado de acuerdo al loss y el RTT medidos. Es recomendable poner el Max. Delay del SRT por encima del valor máximo indicado por este parámetro, si lo que nos importa es la calidad y no la latencia. En esta captura un buen valor sería 600 ms, en vez de los 120 ms que usamos por defecto.
Quality y *Noise* son valores cualitativos de la calidad de imagen, de menor 0, a mayor 4 , como del ruido de la conexión, de menor 0, a mayor 4.

Por lo tanto, el valor más importante aquí es el máximo de Rec. Delay que nos ayuda a establecer el valor de Max. Delay más adecuado a la conexión. Por supuesto que si nos da igual tener una latencia de poco menos de 10 segundos, y queremos ser prácticamente inmunes al ruido en la red, podemos usar el valor máximo de 8000 ms (8 segundos).

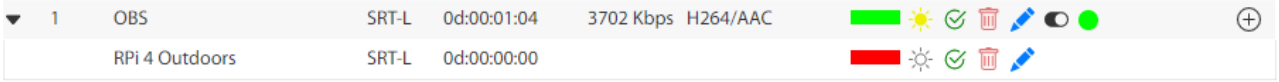
Una vez tenemos funcionando un INPUT podemos crearle uno o varios OUTPUT creando enlaces SRT con equipos a donde queramos enviar la señal de entrada, como podría ser a un repetidor, a otro estudio, etc.

Vamos por tanto a crear un OUTPUT para conectar un Raspberry Pi Player esta vez. Aquí igualmente el servidor creará un OUTPUT con SRT Listener, y el rpiPlayer lo hará como SRT Caller conectando con la IP del servidor y el puerto donde éste escucha. Supongamos que el rpiPlayer está fuera de la red local, por lo que para él la IP del servidor será la IP pública en Internet. En nuestro caso vamos a suponer que es 51.107.4.67 por poner un ejemplo concreto.

Vamos al panel del servidor, y pulsamos en el icono de la derecha del INPUT OBS creado anteriormente, para añadir un OUTPUT nuevo a este INPUT.
Se abre una ventana con campos que ya hemos descrito anteriormente. Recomendamos rellenar al menos Location, y el protocolo como SRT.

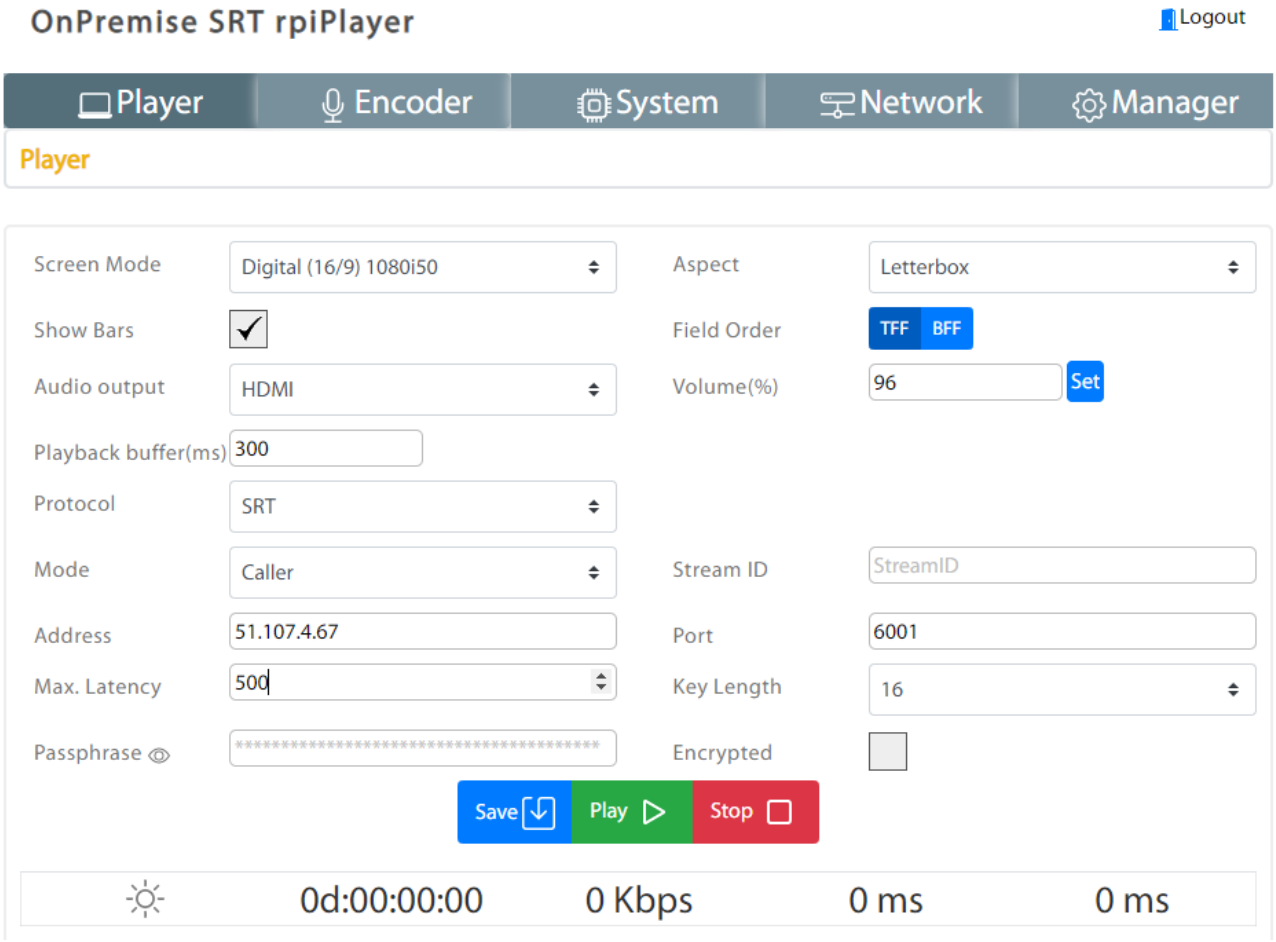


Solamente comentaremos que valor del Port, que para simplificar su elección vamos a usar 600x para los OUTPUT numerados incrementalmente conforme los vamos creando (aunque esto no es obligatorio hacerlo así). Si por casualidad usamos un puerto que ya está siendo usado en otra conexión Listener del servidor, el sistema nos avisará para que lo cambiemos por otro, por lo que no hay que temer nada al respecto. Pulsamos en Save, y tendremos que pulsar en el icono del triángulo negro del INPUT para que se vea el OUTPUT recién creado.



Vemos componentes muy similares a los del INPUT, a excepción del bitrate o de los codecs de audio y video que no son mostrados, ya que al hacer copias exactas del INPUT en el OUTPUT, son exactamente los mismos. Tampoco se muestra el watchdog de audio, por la misma razón.

Vamos ahora a configurar el rpiPlayer, que actuará como SRT Caller para conectarse con el OUTPUT creado en el servidor.



Elegimos SRT en Protocol, Mode en Caller, en Address ponemos la IP del servidor visto desde Internet, y el puerto en el que nos escucha el OUTPUT al que nos vamos a conectar. La latencia no tiene porqué ser la misma en ambos lados de la conexión. De hecho SRT establecerá el mayor de ambos valores. Una estrategia es hacer que sea el servidor quien mande sobre ese valor, para lo que ponemos en los equipos externos el valor mínimo límite que queramos, como por ejemplo 120 ms. De esa manera, el delay será como mínimo 120 ms, o cualquier valor mayor que establezca el servidor.

Como único comentario específico de rpiPlayer y de VMix, es el valor del playback buffer, que establece el buffer que el reproductor toma para evitar dar tirones si el flujo de datos varía debido al jittering o cambio de ping en la red. Un valor seguro para video es 300 ms, y para solo audio es de 1000 ms. VLC por defecto usa 1000 ms.

En cuanto a Address, que era nuestra IP pública en Internet, también podríamos haber puesto nuestro host DDNS, en caso de que nuestra IP no fuera fija.

Al igual que el INPUT, cada OUTPUT arrojará sus propias estadísticas en su conexión SRT que nos permitirán de igual manera ajustar el valor idóneo del Max. Delay para nuestro tipo de uso.

Se pueden añadir todos los OUTPUT que queramos a un mismo INPUT, y se pueden crear tantos INPUT como el hardware del servidor pueda manejar. Mediante los controles podemos editar sus

valores, activarlos o desactivarlos a nuestro gusto, y borrar los que no vayamos a usar más. No obstante es recomendable, desactivar en vez de borrar, ya que una vez borrado un elemento, habrá que volverlo a crear desde el principio, y es mucho más sencillo editar uno preexistente a crear uno de nuevo, pero eso queda a gusto del usuario.

También es obvio que si se borra un INPUT con varios OUTPUT, éstos se borran a la vez con él. Editando un OUTPUT se puede también cambiar de qué INPUT copia la señal, de manera que no es necesario borrarlo de un INPUT y crearlo de nuevo en otro, si no meramente editarlo y cambiar su Channel o INPUT. De esta manera, podemos desde el mismo servidor, cambiar el contenido que llega a cualquier OUTPUT de esta manera tan rápida y sencilla.

En Youtube, tenemos colgados videos de ejemplo, donde se configuran diferentes softwares usando SRT, en la playlist: https://www.youtube.com/watch?v=W3jWg61VX58&list=PLDyk9LJ27Vpprw-fjRn2uBJ8WUPGIHAH2 .

Para terminar con los enlaces SRT, vamos a tratar el campo StreamID, que hemos dejado siempre en blanco en el modo Caller. Este campo solo será necesario, si el servidor al que se conecta, lo usa para identificar un stream concreto. En nuestros servidores nunca lo vamos a usar, ya que no multiplexamos varios streams en un mismo puerto UDP, no obstante añadimos este campo para ser compatibles con servidores que sí lo hagan, y será el administrador de dicho servidor quien le indicará el StreamID a conectar en su caso.

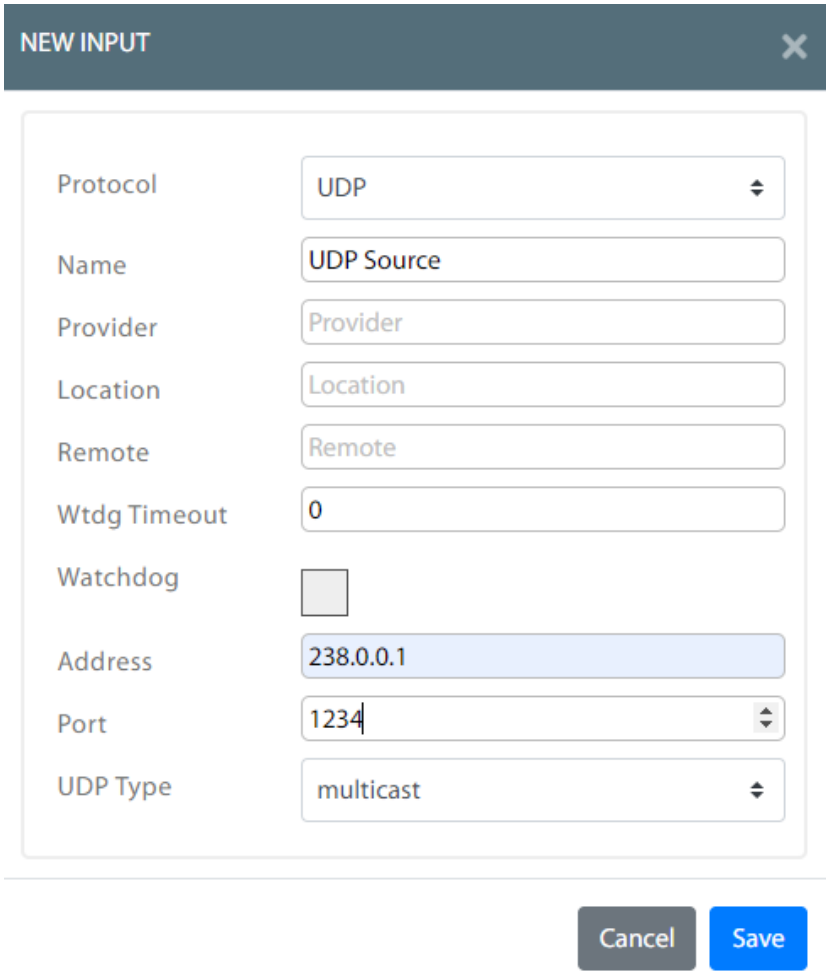
Seguidamente vamos a tratar los enlaces por protocolo UDP, también conocido como MPEG/IP o MPEG-TS.

Enlaces UDP

UDP es un protocolo de transporte que por sí solo no tiene mecanismo para recuperar paquetes perdidos, por lo que solamente lo hemos incluido para casos muy concretos.

En un INPUT el protocolo UDP puede recibir señal de una fuente unicast o multicast. En el caso de que la fuente sea unicast, es como un SRT Listener, que no requiere Address, ya que escuchará dicho puerto en todas las interfaces del servidor. En el caso del multicast, la dirección IP será del tipo multicast (esto es desde 224.0.0.0 hasta 239.255.255.255).

Normalmente los encoders clásicos con salida UDP lo hace en multicast, por ejemplo en direcciones del tipo 238.0.0.1 por el puerto 1234. Un INPUT así se crearía con estos valores.



No volveremos a explicar los campos comunes desde Name hasta Watchdog, ya vistos en los enlaces SRT, y que son comunes a todos los protocolos.

Los OUTPUT por UDP principalmente serán en multicast, para contribución en cabeceras IP, o decoders clásicos que solamente suelen llevar entrada IP en UDP. En el OUTPUT hay solo un campo adicional, llamado TTL, que nos dice la cantidad de routers por los que queremos que pueda pasar como máximo. El administrador de la red multicast le puede dar dicho valor, pero como mínimo debe de ser 1 para que el stream salga a la red local.

Tanto los INPUT como OUTPUT en UDP tienen latencia cero en nuestro servidor.

Protocolo RTMP

El protocolo RTMP fue creado por Adobe para ser usado en la transmisión de video junto a la tecnología Flash. A día de hoy, Adobe ha abandonado Flash, y ya no se le da soporte. La única razón para que hoy se siga usando RTMP, es que sigue implementado como contribución en muchos servidores de streaming web, pero se espera que con el tiempo sea sustituido por SRT. El standard RTMP solamente admite los codecs VP8 y H.264 de video, lo que limita su uso profesional solo al terreno del H.264 en exclusiva. Al usar en la capa de transporte TCP, su capacidad de envío se reduce conforme crece el RTT (ping) o cuando aumenta la tasa de pérdidas de paquetes (loss), hasta reducirse por debajo de los 800 kbps, donde el streaming de video profesional es totalmente impracticable. Se puede dar el caso, donde una conexión en RTMP no pueda superar los 200 kbps, debido a pings superiores a los 500 ms, y pérdidas de más del 2%, y mientras en SRT en esa misma linea se pueda enviar sin problemas 8000 Kbps.

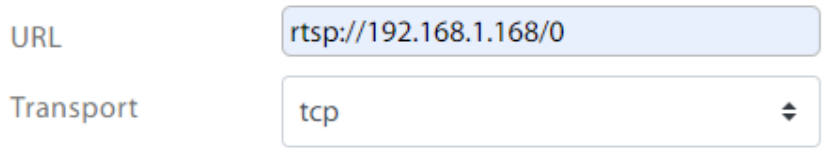
No obstante, para facilitar la incorporación de streams desde servidores que aún solamente pueda servirlo en este protocolo, hemos añadido RTMP pull a INPUT, y RTMP push a OUTPUT para permitir publicar en servicios que aún lo usan como Youtube, Facebook, Twitch y muchos otros.

La URL de un stream RTMP siempre tiene la forma rtmp o rtmps:
rtmp://dominio/app/streamname
La parte del dominio, puede ser un dominio completo, o al IP del servidor RTMP. La parte del streamname en algunos servidores se conoce como clave de emisión o RTMP Key. Y por tanto a la parte anterior se le llama URL o FMS URL.

Esta sería la forma de introducir la URL de un stream RTMP, tanto en un INPUT como en un OUTPUT.

Protocolo RTSP

Es un protocolo muy poco usado a día de hoy, salvo en cámaras IP de vigilancia, y algún que otro encoder. Por ello hemos añadido la opción de hacer pull de este tipo de servidores solo como INPUT. Una URL de rtsp, va toda completa como por ejemplo: rtsp://192.168.1.168/0, y aunque la capa de transporte más típica es TCP, también puede ser UDP o HTTP. Este sería un dato que el servidor RTSP en cuestión debe de ofrecer.



Protocolo HTTP

Es un protocolo que ya conocemos para acceder a cualquier sitio web, tanto sin cifrar HTTP, como con el cifrado SSL, en forma HTTPS. Todas sus URLs comienzan en http:// o https:// .

Hemos incluido 3 tipos de INPUT accesibles desde HTTP, como son:

- HTTP progresivo : típico de servicios como Shoutcast o Icecast, de audio.
- HLS: protocolo creado por Apple, y hoy muy usado en la web. Sus URL acaban en .m3u8
- DASH: protocolo abierto, usado mucho en el VoD. Sus URL acaban en .mpd

Administración de OnPremise SRT Server

Para usar OnPremise SRT Server, se puede hacer con 2 roles bien diferenciados: admin y usuario. El administrador, puede acceder a todo el panel completo y funciones, mientras el usuario solamente a los secciones Streams y System.

Desde la sección Manager, el Admin, puede cambiar su clave, y correo electrónico de recuperación de claves, en caso de olvido. También puede establecer el nombre de usuario y clave del usuario del server, y también podrá bloquear al usuario de manera que todos los streams queden paralizados.

El administrador puede también actualizar el servidor cuando alguna actualización esté disponible. También podrá hacer un Backup de todo el servidor cuando, por ejemplo, se tenga ya funcionando toda la red de distribución. El fichero que se genera lleva la fecha y la hora de creación en el nombre, y es de tipo JSON. Si en algún momento se quiere reestablecer el sistema a una fecha concreta que tengamos guardada, se hacer un Restore de dicho archivo guardado.

IMPORTANTE: *Cada vez que actualice el software, una vez que el server ha recargado (3 segundos) y que estás en la página de login, usa Ctrl+F5 una o dos veces para recargar la webapp del panel en tu navegador. Si no lo haces, estarás usando la versión antigua del mismo que aún está en la caché del navegador.*

La sección License, contiene información sobre la licencia que está en vigor, y la fecha de expiración de la misma. Es importante guardar el identificador de Licencia o LICID en el caso de que necesitemos cambiar de hardware el servidor, y queramos mantener nuestra licencia actual.

Desde el mismo panel, se pueden efectuar las compras y suscripciones a las diferentes licencias disponibles, así como su pago.

Cuando el software es instalado por primera vez en un hardware nuevo, se genera un nuevo LICID con licencia gratuita LICA-G para los próximos 30 días. Si quiere instalar una licencia proveniente de otro hardware en este nuevo equipo, solo tiene que poner el LICID de dicha licencia y hacer clic en Save.

El acceso como Admin, por defecto, es admin para el username, y admin para el password. Si ha cambiado el password del mismo, y lo olvida, podrá recuperarlo introduciendo el e-mail de recuperación en el Login al pulsar el botón “I’ve forgot my login info”.

El acceso como usuario, por defecto, es user para el username, y user para el password. Si ha cambiado el username y/o password del mismo, y lo olvida, podrá también recuperarlo de la misma forma.

Si tiene alguna duda al respecto del uso de OnPremise SRT Server, puede enviarnos un formulario de contacto desde nuestra web: https://todostreaming.eu/ , estaremos encantados de atenderle.

También puede acceder a los videos de formación en Youtube desde la misma página web.