

DESCRIPCION DEL SERVICIO SMART-SRT

SmartSRT es un servicio ofrecido por **TodoStreaming**, basado en el uso del protocolo **SRT** de Haivision®, pero creado para redes de distribución punto a multipunto. Como tal, sus puntos fuertes son los siguientes:

- **Alta calidad.**- Calidad de imagen profesional de hasta 16 Mbps (hasta 4K UHD TV)
- **Bajo coste.**- Reemplazando las costosas redes de satélite o los rígidos MPLS, por la más asequible y flexible red de Internet.
- **Flexibilidad.**- Permite el envío desde cualquier parte del planeta con acceso a la red de Internet.
- **Baja Latencia.**- Permite comunicaciones rápidas con UDP y la confiabilidad del TCP.
- **Codec agnóstico.**- Admite cualquier encapsulado en TS, bien sea con MPEG-2, H.264, HEVC, AV1 o cualquier codec futuro.
- **Seguro.**- Cada conexión con el server va cifrada en AES, para que el contenido viaje seguro por Internet.

El servicio está orientado para el punto a multipunto, de manera que un solo emisor, desde cualquier parte del planeta, pueda enviar contenidos de alta calidad en tiempo real a varios puntos remotos, con una latencia inferior a 5 segundos, de manera segura y estable en el tiempo.

Llegó el momento de enterrar los viejos protocolos de streaming basados en TCP (RTMP, HLS, RTSP) que tantos cortes sufren en el Internet público, y que nos hicieron aceptarlos como algo cotidiano e irremediable para siempre.

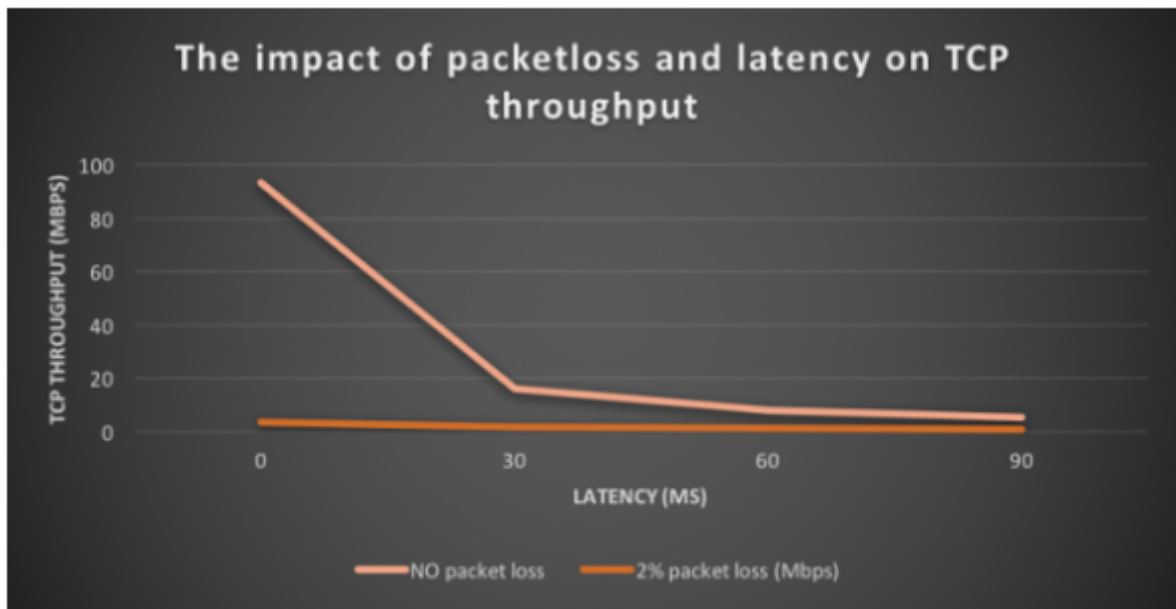
La base teórica de todo esto

Es importante conocer la base de nuestros problemas, si queremos encontrar una solución viable a los mismos. Una de las preguntas más frecuentes que hemos escuchado en estos 10 últimos años de servicios de streaming era, ¿ por qué si tengo 12 Megas de bajada, se me cortan las bajadas de 1.2 Mbps (1200 kbps), más de 5 veces al día, sino más ?. Esa pregunta va encerrada en una sensación de impotencia y frustración absoluta. ¿ La culpa es de mi proveedor de acceso a Internet ?. Tu proveedor te da una dirección de su red donde con tu navegador puedes hacer un test de velocidad. El ping que te da suele ser inferior a los 20 ms, y la velocidad de bajada es de más de 10 Mbps. ¿Donde esta el problema entonces?, si buscas hacer un test de velocidad en cualquier servidor de fuera de la red de tu proveedor, verás que el resultado puede bajar bastante, tanto como tu ping se va estirando a la vez.

La forma en como la mayoría de tests de velocidad miden, es usando el protocolo HTTP para enviar y recibir un fichero de un tamaño concreto, y cronometran el tiempo transcurrido. De los 2 protocolos de base del transporte que se usan en Internet, llamados UDP y TCP, el más usado por los protocolos de aplicación, como por ejemplo HTTP, es TCP. Este protocolo nos asegura la entrega de cada paquete de datos que se envía por él, esperando que el extremo receptor, envíe una notificación positiva sobre la recepción del mismo. Aunque el mecanismo es más complejo, sirve lo que he explicado para comprender, que cuanto más lejos está el receptor, más tiempo tarda la llegada de la notificación positiva de la recepción, por lo que conforme más lejano estén ambos puntos, menos cantidad de paquetes por segundo se podrán enviar con esta entrega certificada. Para medir la distancia entre 2 puntos, en una red usamos el PING, que nos da el tiempo que tarda un paquete en ir y volver entre ellos. Se suele medir en milisegundos (ms), y a día de hoy las antípodas geográficas de cualquier punto en la Tierra esta a menos de 400 ms.

A la variable distancia, hay que añadir otra variable importante, que es la pérdida de paquetes. Es decir, si un paquete no llega, en TCP debe de ser reenviado dentro de una ventana de tiempo que se va cerrando conforme aumenta la proporción de paquetes perdidos. Esto hace por tanto que la capacidad de envío de paquetes aún se reduzca más debido a esto.

A continuación mostramos una gráfica y una tabla con los valores de la capacidad real (throughput) de TCP en una conexión de 100 Mbps sin pérdida de paquetes (packet loss), y con una tasa del 2% de pérdida (2 paquetes de cada 100). *Extraída de <https://accedian.com/enterprises/blog/measuring-network-performance-latency-throughput-packet-loss/>*



Round trip latency	TCP throughput with no packet loss	TCP throughput with 2% packet loss
0 ms	93.5 Mbps	3.72 Mbps
30 ms	16.2 Mbps	1.63 Mbps
60 ms	8.7 Mbps	1.33 Mbps
90 ms	5.32 Mbps	0.85 Mbps

Como se puede observar, la capacidad de TCP se ve mermada con la distancia (Round Trip Latency o RTT es el PING), pero aún mucho más con la pérdida de paquetes. Cuando un paquete viaja de un punto a otro, pasa por al menos 7 nodos en redes nacionales, unos 12 nodos dentro del mismo continente y unos 20 nodos en zonas más alejadas. Dichos nodos repiten los paquetes conforme llegan. Si el nodo sufre una saturación temporal, puede que lo repita fuera de tiempo, o que simplemente lo descarte por no poder atenderlo. En la tabla y gráficas superiores se evalúa una cantidad típica en Internet, el 2% de pérdida de paquetes (packet loss). Es fácil ya intuir la respuesta a nuestra pregunta. La culpa la tiene el PING (latency en ms, en este gráfico) y la pérdida de paquetes. TCP es la base sobre la que funciona la web HTTP, HTTPS, el streaming RTMP, HLS y muchos otros servicios.

¿Y qué pasaría con UDP?. La otra base de la comunicación en Internet es UDP. En ella funcionan servicios de internet tan importantes como DNS, NTP (servicio de reloj universal), etc. UDP

también envía paquetes de datos, pero no revisa su entrega. En una palabra, si se pierden el servicio no debe verse afectado. Y así es con el reloj universal, que si no llega un paquete, no pasa nada, ya que el reloj de nuestro PC se sincronizará con el siguiente paquete, y si no, con el siguiente. Es evidente que haya o no pérdida de paquetes, el tiempo que tardan los paquetes en llegar al otro punto es la mitad del PING, pero como no hay mecanismo de notificación positiva que retrase el envío de los paquetes siguientes, la velocidad de envío por UDP no es afectada ni por el PING, ni por la pérdida de paquetes. Así que si tienes 12 Mbps, podrás recibir a ese mismo ritmo sin problemas, aunque los paquetes se envíen desde la otro lado del planeta. ¿Y por qué no usamos UDP desde el principio?. Esta claro que es genial si no hay pérdida de paquetes, pero cuando la hay, esto es lo que veremos en el mejor de los casos.



Para ello se crearon mecanismos pasivos para la corrección de errores similares a los usados en el standard DVB, conocido como FEC. Dentro del mismo TS (transport stream) van incluidos los datos para la recuperación de los errores. Este método es suficiente para redes privadas con poco ruido, pero en Internet público es bastante ineficaz. Con el tiempo diferentes empresas han ido creando protocolos de aplicación basados en UDP con recuperación de paquetes usando diferentes mecanismos, y el más común es mediante la notificación de paquetes no entregados (notificación negativa o NAK), que genera mucho menor tráfico desde el receptor hacia el emisor que la notificación positiva del TCP. La cadencia de dichos envíos NAK, y la información que se enviaba en ellos eran las principales diferencias entre ellos. Sin embargo todos fueron patentados y licenciados privativamente (QVidium, Zixi). No fue hasta Marzo de 2017, que Haivision decidió abrir su protocolo SRT al mundo de los desarrolladores open source. Su protocolo llevaba una mejora adicional entre otras, y era el adecuar el timestamp del paquete a las circunstancias de la red en cada momento. La versión que hemos usado nosotros es la 1.2.2 liberada el 20 de Noviembre de 2017 (<https://github.com/Haivision/srt>). Con dicha liberación y la posterior creación de la SRT Alliance, y con la entrada de grandes empresas como Microsoft y Ericson, se ha hecho posible una rápida adopción de este protocolo para el transporte de streaming y otros datos.

SRT es un protocolo definido como punto a punto, basado en UDP con un mecanismo avanzado de recuperación de paquetes perdidos de hasta el 10%, con un impacto de no más del 33% en el incremento del ancho de banda. Esto quiere decir que si tu ancho de banda nominal, es decir el que tu compañía te asegura dentro de su red interna, es de 12 Mbps de bajada, podrías recibir sin problemas más de 8 Mbps ($12 / 1.5$). Y eso es independiente del PING, y de la pérdida de paquetes.

Con SRT podrás enviar mucha mejor calidad de video que con RTMP, y sin tener que sufrir cortes, porque el player consumió todo el búffer guardado, debido a la caída de capacidad producida por la pérdida de paquetes o el incremento del PING durante el tiempo suficiente para vaciar dicho búffer. SRT fue creado con la idea de atravesar distancias muy largas y redes poco confiables, y además hacerlo en el tiempo más bajo posible. Para ello el mecanismo de recuperación de paquetes mediante NAK se realiza con una cadencia de la mitad del PING, por lo que si usas un búffer inferior a 1.5 veces el PING, no das tiempo a que los datos re-enviados por el emisor lleguen a tiempo, y sería como usar UDP crudo, con la visualización de artefactos. Es por eso que es recomendable usar un búfer de al menos 3 veces el PING, si queremos que se recuperen los paquetes. Si vemos artefactos en nuestro video, es que debemos duplicar el búffer para dejar de verlos. Sin embargo si un búffer de digamos 100 ms pasa a ser de 200 ms, significa que la latencia aumentará justo esa cantidad. Tendrás por tanto que decidir si para tu uso específico, prefieres baja latencia o calidad.

SmartSRT es un servicio punto a multipunto, en el que hay un servidor de por medio entre el emisor y los diferentes receptores. Esto es necesario, si vamos a distribuir la señal broadcast a más de un destino, o si queremos distribuirla a un solo punto, pero queremos evitarnos configuraciones de redes dinámicas complejas. Configurar un punto a punto SRT, requiere un cierto adiestramiento y conocimientos de redes. SmartSRT te facilita toda esa configuración de manera automática convirtiéndolo en un mero corta y pega.

Por otro lado el mercado esta repleto de equipos y software encoders que usan aún los protocolos de toda la vida, como RTMP, RTSP y HTTP. Youtube y Facebook mismo, a parte de otros, siguen recibiendo tu señal en vivo por RTMP. Aunque tienen la infraestructura suficiente, como para proporcionarte un punto de ingesta a menos de 20 ms de ti. La idea es poder reutilizar todo ese software y hardware para emitir y recibir SRT. Para ellos necesitaremos instalar en la red local del emisor un software que recoja la señal local RTMP, RTSP, HTTP o UDP y la convierta en una subida SRT. Para ello usaremos un Raspberry Pi. El mismo software también lo podremos usar para recibir la señal SRT del servidor SmartSRT en cada punto receptor.

Con SmartSRT puedes seguir usando tus software y equipos de siempre, además de beneficiarte de las ventajas de SRT.

De hecho, todo usuario que use nuestro software Player for Raspberry Pi, con la version 20181225 o posterior, podrá trabajar con SmartSRT. En el Manual Paso a Paso, se describe el proceso completo de creación y configuración, de una red completa de distribución de TV, sobre SmartSRT en la red pública de Internet.