WHITE PAPER

SRT Open Source Streaming for Low Latency Video

Video contribution using public networks

Haivision

VIDEO AT WORK

SRT Open Source Streaming for Low Latency Video

VIDEO CONTRIBUTION USING PUBLIC NETWORKS

Modern technologies now make it possible to transmit high-quality video in real time across the public internet. A variety of proprietary solutions are available, but these closed systems can require expensive monthly usage fees and are restricted to work only with other compatible devices, usually from a single vendor. SRT (Secure Reliable Transport) is an open source, freely available and flexible specification that performs as well as, or better than, proprietary solutions while providing the ability to work between products from many different manufacturers. SRT users can select the best products for their particular needs while at the same time being able to share streams with many other organizations and avoiding vendor "lock-in."

INTRODUCTION

Years ago, there were very few options available for delivering high-quality video signals for contribution applications over long distances, and all of them were expensive. Satellite connections had to be booked in advance and required relatively expensive terminal gear on both ends of the circuit. Terrestrial connections over dedicated video networks also required reservations, and are still only available in locations where carriers have chosen to deploy interfaces (known as PoPs or Points of Presence). Organizations using these technologies may have felt that they had few alternatives for transporting high-value contribution signals (including feeds from sporting events and live news), because the resulting programs needed to be of the highest quality for broadcasting to thousands or millions of viewers.

Today, many organizations have deployed devices that can transmit high-quality video feeds over the public internet, and this installed base is growing rapidly. These products provide a number of operational advantages for broadcasters, not the least of which is rapid deployment for breaking news stories. Being able to use whatever network connections are available at the site of a story permits live video signals to be transmitted when previously only taped reports could be filed. The flexibility of this technology also eliminates the need to send satellite or microwave equipment to remote broadcast locations, further reducing costs as well as improving flexibility and speeding up dispatch times.

Technologies that transport high-bandwidth, low-latency video streams over unmanaged networks must be able to handle large amounts of packet delay variation (jitter) and be able to recover packets that have been lost in transmission (packet loss). Both of these impairments can be accommodated through error correction or various applications of packet retransmission. Selecting the right implementation of these techniques and providing a product that minimizes delay and maximizes video quality is the key to a successful solution.

A number of companies have developed closed, proprietary systems to address the vibrant market for delivering contribution-quality video over the public internet. Several of these offerings require ongoing monthly payments in exchange for a fixed number of hours of use. Unfortunately, these single-vendor solutions do not allow streams to be sent to or received from other manufacturers' devices. This prevents end-users from exchanging content with other users who don't own the same brand of equipment. Proprietary solutions are also at risk of becoming orphan products, when vendors go out of business or discontinue product lines, as has happened more than once in the recent past.

SRT Open Source

SRT – THE OPEN, INNOVATIVE CHOICE

Secure Reliable Transport (SRT) was developed specifically to address the shortcomings of existing internet-based video contribution systems. It is open source, free, uses the latest technologies, and doesn't require centralized servers to operate. Let's examine each of these aspects in more detail.

An open standard means just that – it is available to anyone who wants to implement it. Not only is SRT open, but it is also open source and royalty-free. This means that the software (technology stack) required to implement SRT is available from a public repository, and there are no fees to incorporate the technology into any device or service. Available on GitHub, the open source SRT protocol uses the Mozilla Public License (MPL 2.0) which provides flexibility to developers, encourages source code contribution to the larger community, and enables integration of SRT in mobile applications.

Widely-used free tools such as VideoLAN's VLC, FFmpeg, GStreamer, OBS and Wireshark now support SRT. The benefits are significant for both technology suppliers and users, greatly simplifying implementation and reducing costs, thereby improving product availability and helping to keep prices low. And, since every implementer uses the same code base, interoperability is simplified.

New technologies are revolutionizing today's video market, as broadcasters migrate away from special-purpose, dedicated devices and services and move instead to use widely-available networks based on IP (internet protocol). SRT builds on this trend by enabling the use of Ethernet and a variety of wireless data networks for transmitting and receiving compressed video signals over virtually any available IP connection that has adequate bandwidth. Lost packets can be easily recovered using SRT's built-in error correction and low-latency data re-sending capabilities.

SRT allows connections to be made directly between signal sources and their destinations. This is in contrast to many existing video transmission systems that require a centralized server to collect signals from remote locations and redirect them to one or more destinations. Architectures based on a central server have a single point of failure that can also become a bottleneck during periods of high traffic. Transiting signals through a hub also increases end-to-end signal transit times and potentially doubles bandwidth costs by requiring two links to be implemented: one from the source to the central hub and another from the hub to the destination. By using direct source-to-destination connections, SRT can reduce delay, remove the central bottleneck, and lower network costs.

Since being developed over four years ago, media and non-media companies alike have moved towards widespread deployment of SRT. For example, ESPN has deployed SRT-equipped devices to 14 collegiate athletic conferences that have been used to produce more than 2,200 events via low-cost internet connections, in place of using traditional satellite uplink services that would have cost $8-9 million. Microsoft uses SRT for hundreds of events, including their annual partner conference, Microsoft Inspire. By using SRT, Microsoft is able to produce professional-quality, live videos using low-cost internet connectivity at hotels and conference centers in place of expensive leased data lines or dedicated video circuits.

Furthermore, Eurovision, operated by the European Broadcasting Union (one of the largest associations of national broadcasters in the world), implemented SRT as part of their video contribution service, offering European broadcasters low-latency live video transport using internet connections. In another example, MediaKind adopted SRT for use in their distributed cloud contribution system that is poised to change the way that broadcasters collect and process live content.

SRT TECHNICAL DESCRIPTION

Whenever signals are sent from one location to another, they are subject to bit errors and packet loss. For high performance video signals, these network faults can lead to significant degradations, even for loss rates that are commonly encountered in transit across the public internet. Figure 1 shows how severely a sample video signal is degraded by a relatively low packet loss rate.

Figure 1: *This image is divided into four quadrants to illustrate the impact of errors and error correction systems on video signals. The source video signal is shown in the lower left. The upper left quadrant shows the video degradation caused by a two percent packet loss rate. The two images on the right half of the illustration show how SRT can eliminate video impairments for two commonly used video codecs at two different bit rates.*

Two techniques are commonly used to correct these network faults – Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ). Both of these have been used in modern video transmission. Figure 2 shows how both techniques impact the packet flow over a network connection.

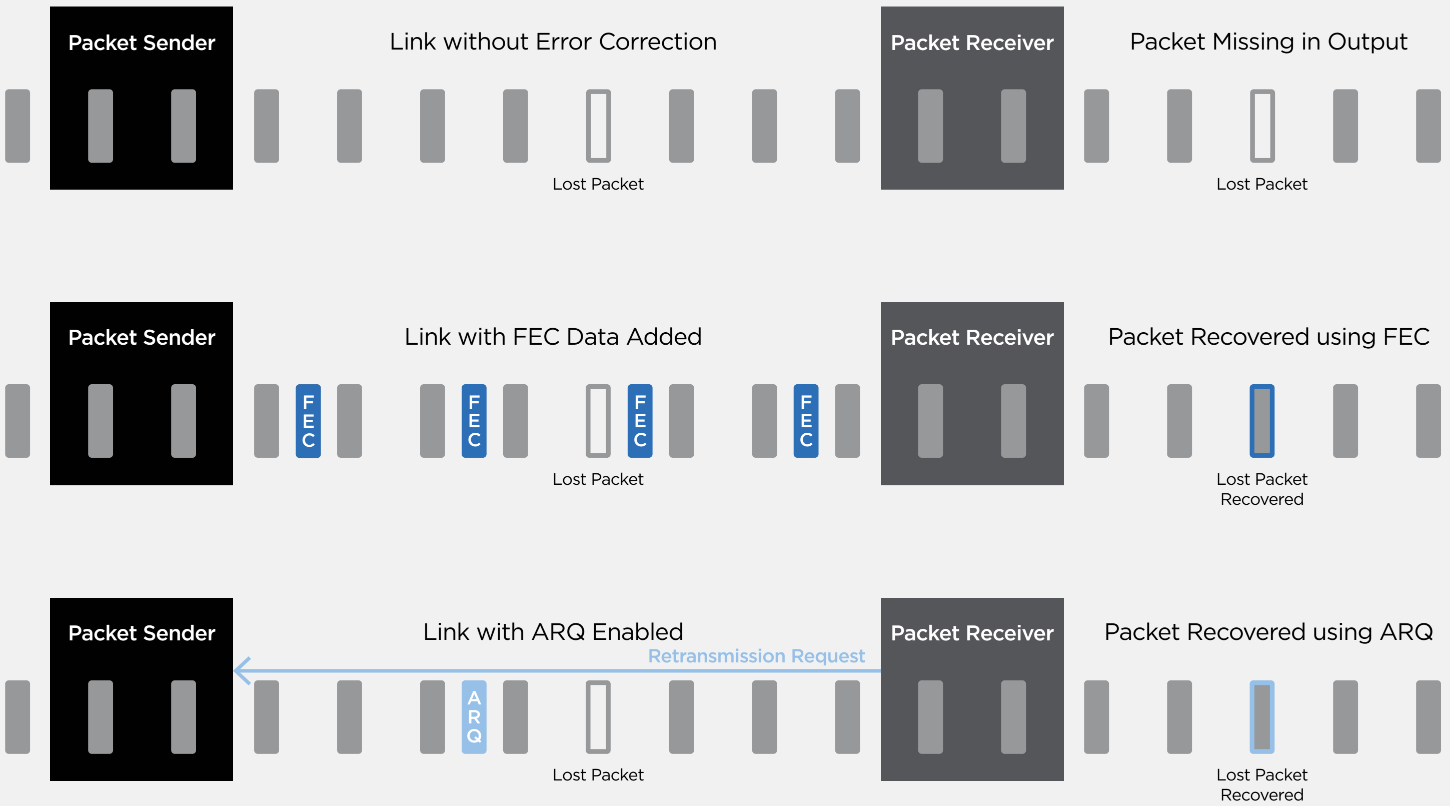

Figure 2: *Comparison of three packet delivery mechanisms. At the top is an uncorrected stream, which generates an error in the output signal whenever a packet is lost. In the middle is a solution using Forward Error Correction (FEC) which adds a constant amount of extra data to the stream which can be used to recreate lost packets (up to a limit). At the bottom is a solution using Automatic Repeat reQuest (ARQ), where the sender retransmits lost packets upon request from the receiver, thus avoiding the constant bandwidth consumption of FEC.*

FEC operates by adding extra data to a transmitted signal that provides information which a receiver can use to recover corrupted or missing data. This technique only works on systems that can support the extra bandwidth required for the FEC data and ones that can withstand signal interruptions which can occur when network error rates burst above a threshold. One drawback of FEC-based systems is that the FEC packets are sent whether they are needed by the receiver or not, so if the data connection is working well then the bandwidth used for the FEC packets is wasted. A second drawback is that the maximum packet loss that can be tolerated is related to and limited by the amount of bandwidth allocated for the FEC data.

ARQ works by establishing a two-way connection between the video source and destination. Each outbound data packet is given a unique sequence number, and the receiver uses these to determine whether all of the incoming packets have been correctly received in the right order. If packets are lost within the network, the receiver can create a list of the sequence numbers of the missing packets and automatically transmits a request back to the sender to have them retransmitted. In the case of networks with high error rates (as can be found on the internet at certain times of day or when failures occur), this process can be repeated multiple times. ARQ requires caching at the sending location (to temporarily store packets in case they need to be retransmitted) and a buffer at the receiving location to rearrange the packets into the correct order before they are sent along to the video decoder or other receivers.

SRT uses ARQ, primarily because it can handle the types of errors that are most common on the internet. There, losses mainly consist of random bursts of lost packets. With ARQ, these errors can easily be fixed through a simple retransmission by the sender of any packets that did not arrive at the receiver. If packets containing bit errors arrive at the receiver, they are treated like missing packets and the sender is asked to retransmit them. As an added benefit, SRT provides high-resolution timestamps for each packet to allow the timing of media streams to be precisely reproduced at the receiver's output. This helps ensure that downstream devices will be able to properly decode video and audio signals.

Every packet sent during an SRT session uses the UDP (User Datagram Protocol) packet format, which offers low-overhead, low-latency packet delivery. Most real-time media transport networks designed for professional (not consumer) applications use UDP because it offers a stable, repeatable packet delivery system with consistent throughput. This is in contrast to the unpredictable, highly-variable nature of TCP connections, as described in the accompanying sidebar "*What's Wrong with HTTP Streaming?*"

WHAT'S WRONG WITH HTTP STREAMING?

Many modern video services, including some real-time signals, are delivered over the public internet using HTTP streaming technology, commonly known as OTT (for Over The Top) video delivery. Examples include Netflix, YouTube, and a variety of broadcast network and sports league streaming services. These services typically use CDNs (Content Delivery Networks) to transmit on-demand and "live" services to thousands or millions of viewers simultaneously.

While it might appear that HTTP streaming could be a viable solution for contribution video applications, there are several drawbacks that become visible upon closer examination. First and foremost is the significant amount of delay that is pervasive in an HTTP streaming implementation – up to 30 seconds of end-to-end delay is not uncommon, due to the many processing steps and multiple buffers along the signal path. With these lengthy latencies, any type of interaction between personnel located on either end of the link is impossible due to the delay.

In addition, the amount of delay is not fixed – when network conditions are good, the delays can drop, but when they are bad, they can increase dramatically. This is due to the nature of the protocol used to transport HTTP – known as TCP or Transmission Control Protocol. TCP requires that all bytes of a stream are completely delivered in their original order. While this might sound like a good way to send video, experience shows that it is not. With video, a few lost bytes can be corrected or at worst ignored in the delivered signal. With TCP, it is not possible to skip over bad bytes; instead, the protocol will continue to retry sending missing data for as long as it takes. This can have a significant impact on viewers, and is the source of many frozen frames and the cause of the dreaded "rebuffering" symbol that appears during crowded network conditions.

A third impact of TCP can be subtle but important to contribution video applications. This is the behavior of TCP to automatically drop packet transmission rates when network congestion occurs. While this behavior is good for reducing overall congestion in a network, it is not appropriate for a video signal, which cannot survive a drop in speed below its nominal bit rate.

Overall, while TCP has a valid role to play in delivering video signals to consumers, the long, variable delays and sudden bit rate decreases make it unsuitable for delivering contribution video signals.

Each SRT connection begins with a handshake and an exchange of capabilities information.

Three different handshake modes are provided for devices to contact each other and set up the necessary data required to send and receive packets, such as IP addresses. The first is Caller mode, where an SRT endpoint tries to connect with a remote device at a known address and UDP port number. The second is Listener mode, where the SRT device continuously monitors incoming traffic to a defined address and port number to await a connection from a Caller device. The third mode is called Rendezvous, where both endpoints simultaneously act as both Callers and Listeners to make it easier to establish a connection through certain types of firewalls. Every handshake requires a two-way confirmation of endpoint identity and passwords through the use of secure cookies before proceeding.

Once the handshake process has completed, the Caller and the Listener exchange their capabilities and configurations. Both ends of the network need to be aware of the overall latency between the two endpoints, so that the correct buffer sizes can be established to handle packet retransmission delays. Connection bandwidth also can be estimated and communicated, to allow the video to be compressed only as much as needed to fit comfortably within the network's capacity. Encryption keys can optionally be exchanged between the sender and the receiver; these are used to scramble the video and audio content within the IP packets using AES 128/192/256-bit encryption so that valuable programming can be sent securely.

SRT CAPABILITIES

Several features differentiate SRT from most of the other video stream delivery formats that are on the market today like RTMP, HLS and MPEG-DASH, including:

Non-Proprietary

SRT is an open source solution that has been integrated into multiple platforms and architectures, including both hardware-based portable solutions and software-based cloud solutions. Because all of the systems rely on the same underlying code base, interoperability is simplified.

Handles Long Network Delays

Because of its flexible, adaptive buffer management system, SRT can work well over connections ranging from a few milliseconds to a few seconds worth of delay, thereby handling anything likely to be found on any private network or anywhere on the global internet.

Supports Multiple Stream Types

Unlike some other solutions that only support specific video and audio formats, SRT is payload agnostic. Any type of video or audio media, or in fact any other data elements that can be sent using UDP, are compatible with SRT.

Supports Multiple Simultaneous Streams

Several different media streams, such as multiple camera angles or alternative audio tracks can be sent over parallel SRT streams that share the same UDP ports and addresses on a single point-to-point link. This can be accomplished while preserving each signal's media format and timing, thereby allowing, for example, an MP4 video signal to share a link with a JPEG2000 stream. This simplifies network configuration and firewall traversal.

Enhanced Firewall Traversal

No modern organization, media-based or otherwise, allows corporate systems to have unrestricted access into or out of the public internet. Firewalls protect private network devices such as PCs and servers from unwanted external connections and attacks. The handshaking process used by SRT supports outbound connections without requiring dangerous, permanent exterior ports to be opened in a firewall, thereby maintaining corporate security policies.

Accurate Signal Timing

Many compressed video signal formats are sensitive to disruptions caused by changes in timing between different elements of the signal. With SRT, each data packet has a high-resolution timestamp assigned by the sender, which can be recovered by the receiver to accurately recreate signal timing relationships, irrespective of network delay variations. In addition, during the handshake process, SRT endpoints establish a stable end-to-end delay profile, eliminating the need for downstream devices to have their own buffers to cope with changing signal latencies.

Central Server Not Required

Some proprietary media transport systems require a centralized server to be used between the sender and the receiver, which can add cost and delay. SRT connections can be made directly between devices, so that a central server is not needed. In addition, if desired, SRT can be deployed using centralized servers and relay points for applications such as cloud-based content collection systems and clip distribution networks where a centralized model is preferred.

Low Recurring Cost

SRT systems are implemented using a free, open source code base, which helps keep cost low for all parties. No royalties, long-term contracts or monthly subscription fees are required for SRT deployments.

API-Based

The SRT technology package is API-based allowing vendors to establish tight, repeatable integrations with their platforms and endpoints.

Open Source Community Adoption

SRT has been adopted by industry leading open source initiatives such as: VideoLAN's VLC, the free and open source cross-platform multimedia player and framework; GStreamer, a foundational streaming engine for compact appliances and mobile devices; Wireshark, the leading network stream analyzer; FFmpeg, the world's most prevalent open source video compression toolkit; OBS Studio, a free and open source tool that allows content creators to broadcast livestreams and recordings around the world; and the Libav open source suite of audio and video processing tools.

Vendor Adoption

Since SRT became available as an open source technology in 2017, hundreds of companies have endorsed the open source project by supporting the SRT Alliance. This consortium of vendors and end users works collaboratively to increase industry-wide awareness and adoption of SRT as a common standard for low-latency video transport over the internet. Highlighted SRT Alliance members include Ateme, Avid, Brightcove, Eurovision, Haivision, Harmonic, Limelight, MediaKind, Microsoft, NetInsight, Telestream, Sencore and Wowza.

End User Adoption

The SRT protocol is used by thousands of organizations globally in a number of applications and markets. End user adopters include Comcast, ESPN, Fox News, Microsoft, NBC Sports, the NFL and Tencent.

CONCLUSION

SRT is an open standard currently supported by an alliance with a large number of active technology suppliers, providing product diversity and avoiding the risks and cost of single-vendor "lock-in." Within this enthusiastic community, many solutions that address a wide range of industry applications now support SRT, including IP cameras, encoders, decoders, gateways, OTT platforms and CDNs. For more information about various suppliers and users of SRT-based technology, visit www.srtalliance.org

As the original developer of the SRT protocol, Haivision is also a founding member of the SRT Alliance and SRT Open Source Project for low-latency video streaming. Haivision provides a number of different technology solutions that address many types of media transport applications. For more information about these products, please visit www.haivision.com or contact info@haivision.com.

If you're interested in learning more about SRT and Haivision's video streaming solutions, contact us.

info@haivision.com | 1-514-334-5445