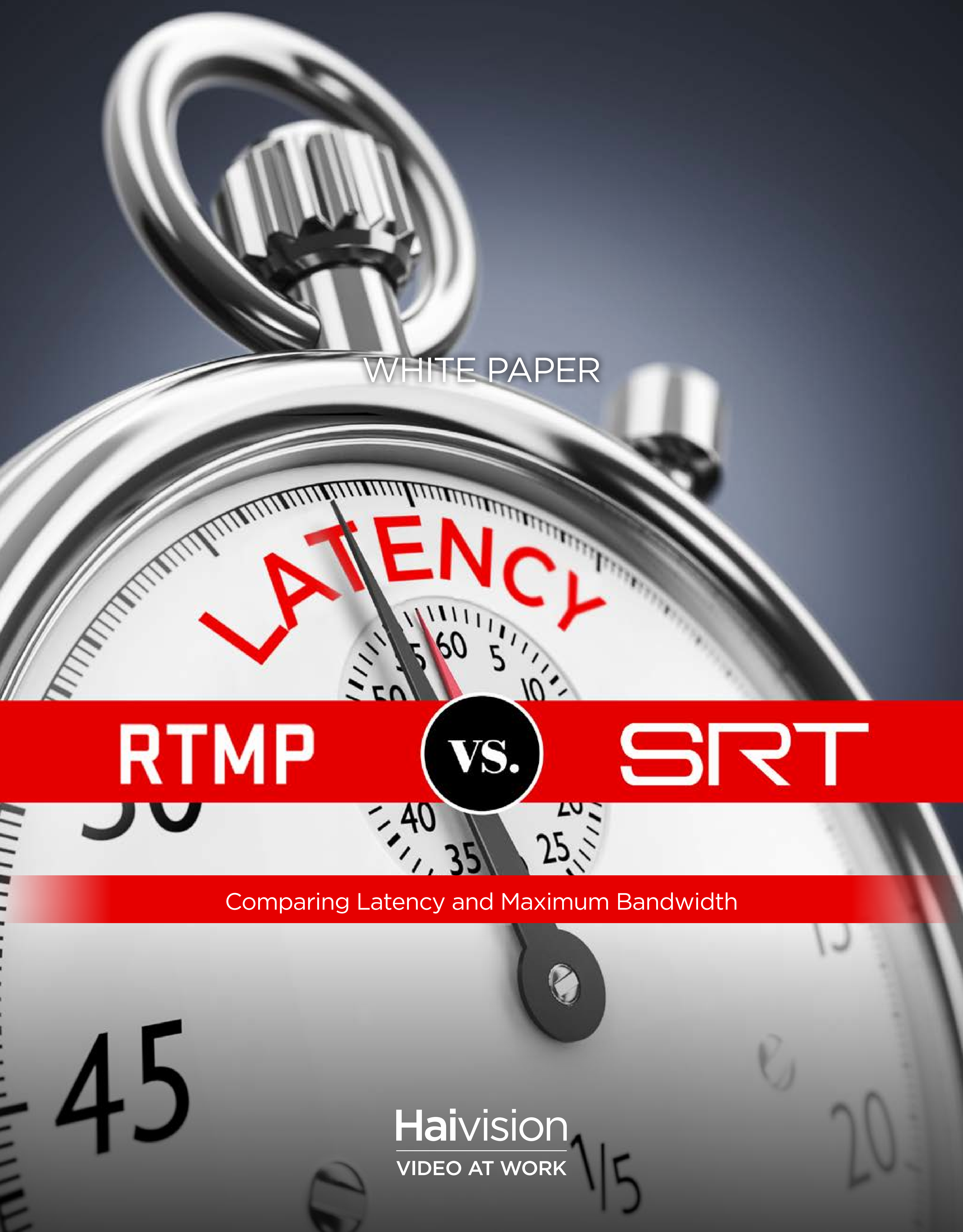


WHITE PAPER

RTMP vs. SRT

Comparing Latency and Maximum Bandwidth

Haivision
VIDEO AT WORK

TABLE OF CONTENTS

RTMP vs. SRT: Comparing Latency and Maximum Bandwidth

INTRODUCTION

For those looking to stream live video over IP with low end-to-end latency, there's a limited selection of transport protocols to choose from. This is especially true when using the public internet as a transport medium as it presents a number of challenging obstacles to overcome such as packet loss and jitter. In this white paper, we set out to compare and evaluate the merits of two commonly used protocols, **RTMP** and **SRT**.

The **Real-Time Messaging Protocol** (RTMP) is a mature, well established streaming protocol with a reputation for reliability thanks to its TCP-based packet-retransmit capabilities and adjustable buffers.

Secure Reliable Transport protocol (SRT) is an open source protocol pioneered by Haivision which leverages an intelligent packet retransmit mechanism on top of a UDP data flow, along with AES256 encryption.

In this white paper, we will examine how SRT performs compared to RTMP over public networks. We'll explore how much buffer is required, what latency looks like and whether there is a limit on the amount of bandwidth you can use. We'll also answer the question of how far a video stream can travel across the globe before it fails.

COMPARING LATENCY OF RTMP VS. SRT

First, let's start by examining end-to-end latency. End-to-end latency, sometimes referred to as glass-to-glass latency, is the total amount of time it takes for a single frame of video to transfer from the camera to the display. There are many factors which contribute to latency depending on the delivery chain and the number of video processing steps involved. While individually these delays might be minimal, cumulatively they can really add up. They include the encoding, decoding, transport medium (internet, satellite, radio, fiber, etc.), sources (cameras, video players etc.) and display devices.

To learn more about latency, what causes it and why it matters, read our video encoding basics blog post.

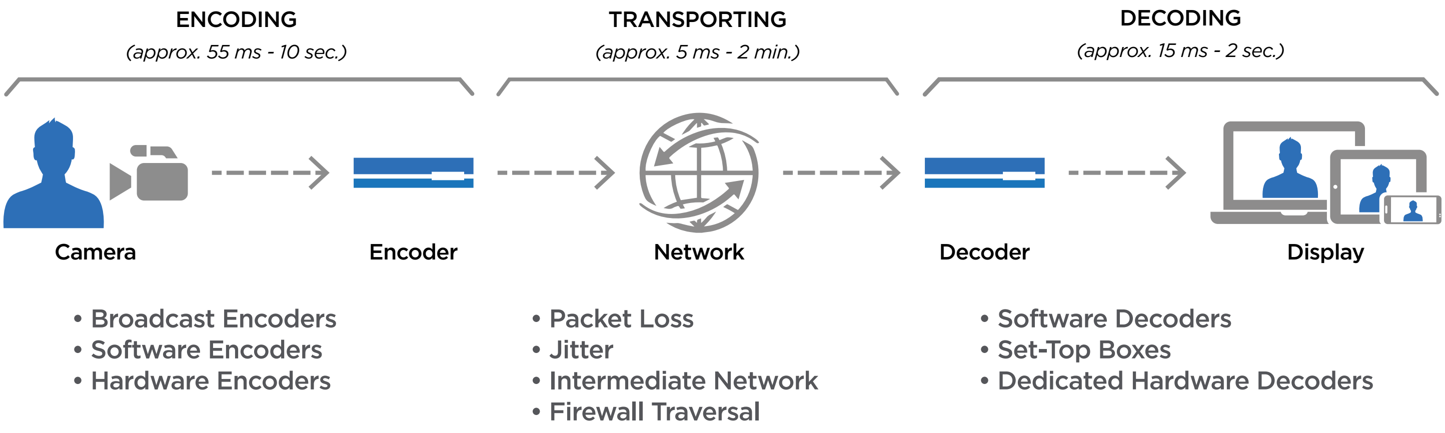


Figure 1: *Components of end-to-end latency*

The focus of this white paper is to measure the impact of using either RTMP or SRT on end-to-end latency. In order to achieve comparable results, the same devices with exactly the same settings were used during testing and the only variable was switching between RTMP and SRT protocols. In some instances however, configurations had to be changed in order to get a stable RTMP stream, and these will be explained in further detail.

TEST SETUP

The following diagram shows an overview of the test setup.

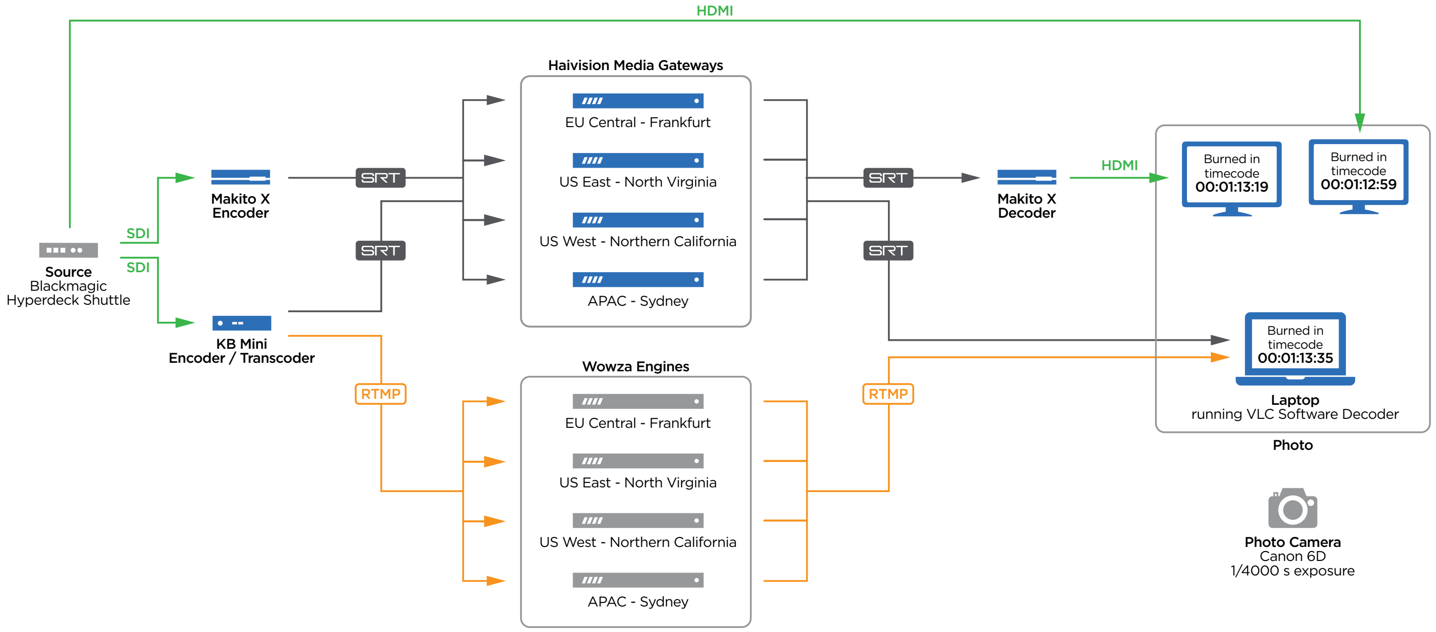


__Figure 2:__ Overview of test setup (all components)

The test setup was designed to be simple and easy to replicate with no special testing equipment required. A still camera was used to capture a single image of two screens side by side displaying a photograph of a single video frame and burnt-in timecode. One of the screens was connected directly to the source and the other to the decoder output so that it was possible to clearly see the round trip end-to-end latency.

In order to capture accurate results using this test setup, the source and destination had to be in the same location. It's important to note that for the purposes of this test, we measured round trip, end-to-end latency. This includes: the encoding of the video signal, the time needed for the stream to travel to one destination and returning to the location of origin, the decoding of the video signal and finally the display latency and buffering of involved servers, software players or hardware decoders.

Source

A Blackmagic Hyperdeck Shuttle video recorder was used as the video source, directly feeding the first screen, and a Haivision KB and Makito X video encoder were used in parallel. The video had a burnt-in timecode to enable identification of each single frame. The video asset was playing at 720p resolution with 60 frames per second. Each single frame of video was visible for 16.67 milliseconds.

Screens

To be able to take a photo which directly displayed the time needed for encoding, streaming and decoding, multiple screens were used. To ensure that the screens did not differ in latency, all displays were connected to the source in parallel and a photo was taken. All screens showed exactly the same frame in the same moment of time. This test was repeated with both screens connected to the laptop, to ensure the laptop internal display was not slower or faster than the screens used.

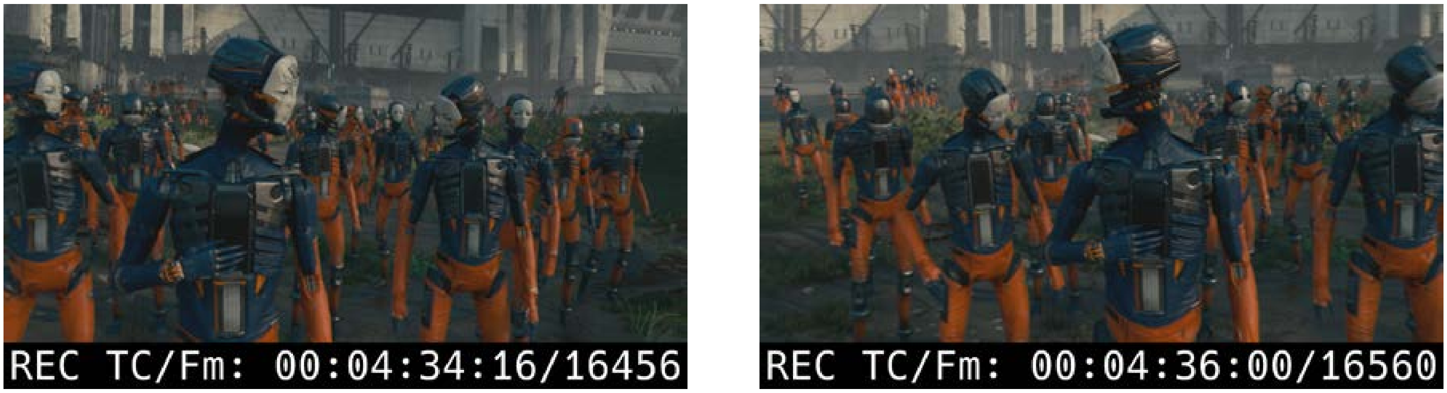


Figure 3: *Screens to show source and decoder output side by side (timecode delta = latency)*

Video Encoder

The baseband video signal was encoded by a Haivision KB mini video encoder (software version 5.4), capable of generating both RTMP and SRT streams in parallel. Both outgoing streams used the same video and audio process. Differences in outgoing bitrate were dependent on the individual protocol overhead used by RTMP and SRT.

The following encoder settings were used:

Resolution	1280 x 720 pixel
Video frame rate	60 fps
Video bitrate	2136 kbps
Video codec	H.264 main profile
GOP size (fixed)	2000 ms
Video entropy	CAVLC
Audio bitrate	192 kbps

Table 1: *Encoding settings*

The total outgoing bitrate adds up to approximately 3Mbps including protocol overhead.

In addition, a Haivision Makito X video encoder and decoder (firmware version 2.3) were used. These are hardware accelerated devices which allow for much lower latency compared to software based encoding/decoding processes. The encoding profiles for both used the same parameters as the KB software encoder.

Video Decoder

As the ability to easily reproduce the test setup was a key focus, the software decoder chosen was VLC player, one of the most widely used video players for both industry and home use. Highly versatile and extremely popular, VLC decodes a wide range of formats including both RTMP and SRT streams.

In this test, a clean VLC install with standard settings was used. The standard cache level of 250 milliseconds worked for most of the tests, but had to be changed in some instances. Details on this can be found in the section: Timestamps, Cache and Buffers.

Wowza Server

As a target for the RTMP streams, a Wowza Streaming Engine (Version 4.7.5 - build 21763) was used. The Wowza Streaming Engine was hosted on AWS instances located in the same AWS data centers as the Haivision Media Gateways, which were acting as stream targets for the SRT streams.

The decoder (VLC) pulled the RTMP stream back to the location of the encoder from that particular server.

The Wowza Streaming Engine also supports SRT, but at the time of testing, its SRT implementation was using a very old version of the protocol and did not represent the full potential of current SRT versions. In addition, at the time of testing Wowza's Streaming Cloud did not have an SRT implementation.

The round trip time (RTT) to a Wowza Server and to a Haivision Media Gateway hosted in the same AWS data center never differed by more than 2 milliseconds. In most tests the total RTT was identical.

Haivision Media Gateway Server

Similar to the RTMP setup, a Haivision Media Gateway (Version 3.0.1) hosted on AWS instances was used as the end point for SRT streams.

Within AWS, the Haivision Media Gateway is available as software as a service and can be spun up in AWS data centers of your choice.

The input, as well as the output, for the SRT streams was configured as SRT listener ports with a defined latency buffer. This buffer size depends on the RTT of the connection between endpoints. More details on this can be found in the section entitled: Timestamps, Cache & Buffers.

Internet Connection

The uplink to the internet was established by a Fritzbox 7590 DSL modem with 100 Mbps downlink and 42 Mbps uplink. The performance of the connection was monitored at all times to ensure the link was not saturated by other applications.

Timestamps, Cache and Buffers

A main difference between RTMP and SRT is the absence of timestamps in the RTMP stream packet headers. RTMP only contains the timestamps of the actual stream according to its frame rate. The individual packets do not contain this information, therefore the RTMP receiver must send each received packet within a fixed time interval to the decoding process. To smooth out differences in the time it takes for individual packets to travel, large buffers are required.

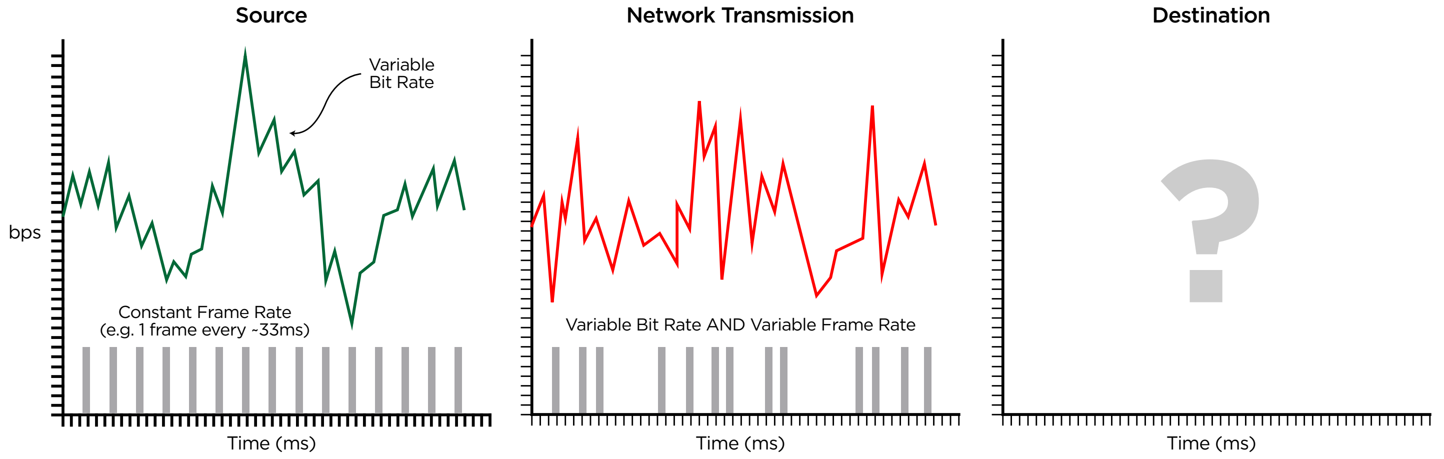


Figure 4: *Input signal characteristics do not match network transmit characteristics*

SRT on the other hand, includes a timestamp for each individual packet. This enables the recreation of signal characteristics on the receiver side and dramatically reduces the need for buffering. In other words, the bit-stream leaving the receiver looks exactly like the stream coming into the SRT sender.

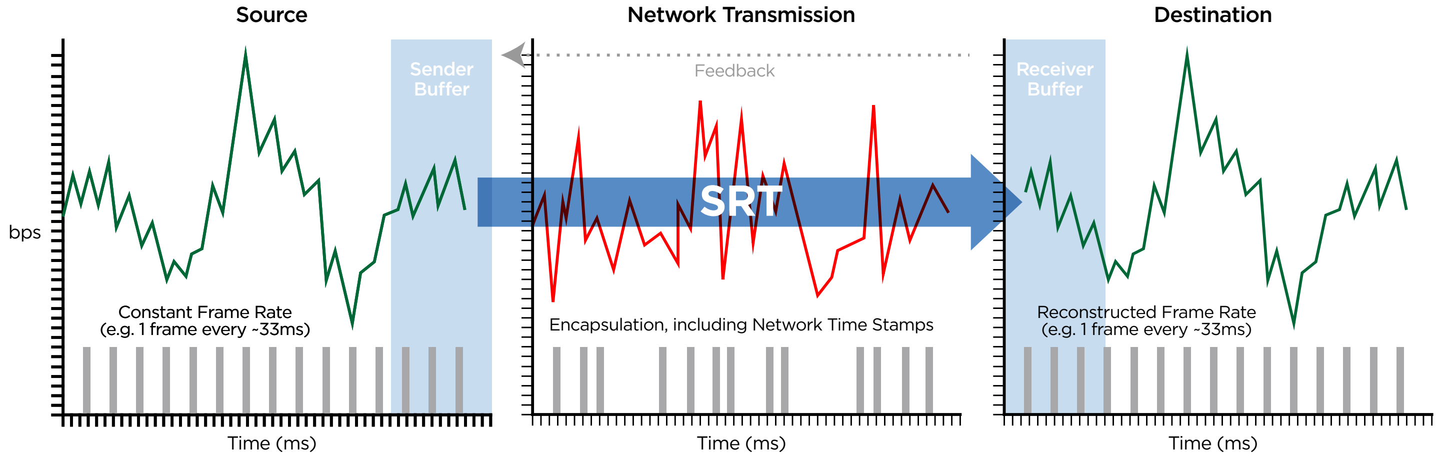


Figure 5: *Recreation of signal characteristics on the SRT receiver side*

Another significant difference between RTMP and SRT is the implementation of packet retransmission. RTMP is based on TCP, which relies on acknowledgments reported by the receiver. However, these acknowledgments (ACKs) are not reported immediately to the sender in order to keep the return traffic low. Only after a sequence of packets has been received, will a report of ACKs or NACKs (negative acknowledgments) be sent back. If there are lost packets within that sequence, the complete sequence of packets (going back to the last ACK) will be retransmitted.

By contrast, SRT can identify an individual lost packet by its sequence number. If the sequence number delta is more than one packet, a retransmission of that packet is triggered. Only that particular packet is sent again, to keep latency and overhead low.

According to Wowza Forums the buffer size calculates as follows:

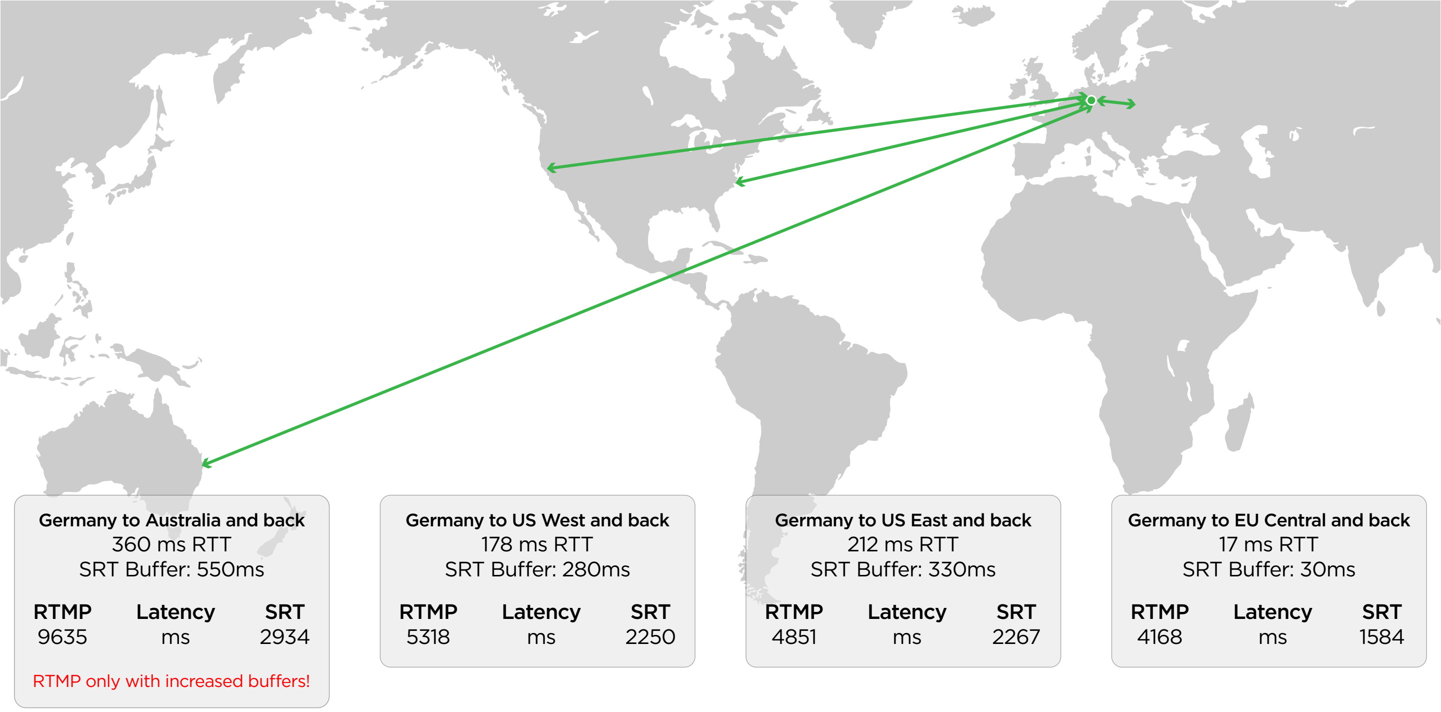
> “Larger SendBufferSize and ReceiveBufferSize will give better throughput on connections where there is a lot of latency (high ping times). The amount of throughput possible in bytes per second must be lower than the buffer size / rtt (round trip time). The size of the buffers determines how much data can be sent before the sending end must stop and wait for a reply from the receiving end. The size used for the connection will be the smaller of the sender’s send buffer or the receiver’s receive buffer. Generally most modern clients will use automatic settings so the send buffer on Wowza may limit the connection speed for high latency connections. By default, Wowza uses 65000 for the send buffer because it is a good average value. This will give you the following performance figures:
>
> 65000bytes / 50ms rtt = 1300000 bytes / sec = 10.4mbps
>
> 65000 / 100ms = 5.2mbps
>
> 65000 / 200ms = 2.6mbps
>
> As you can see, the further away the client player is, the less throughput it will have streaming from your server. Unless your player is a very long way away, the default settings should be fine for the stream and player you have described. You can severely restrict the throughput if you set it too low. The low latency settings are for when the players are very close to the server or the stream bitrates are very low.
>
> 16000bytes / 50ms rtt = 320000 bytes / sec = 2.56mbps
>
> 16000 / 100ms = 1.28mbps
>
> 16000 / 200ms = 640kbps
>
> The receive buffer sizes on Wowza generally only affect publishing of live streams so these should be set to allow your publishing encoders plenty of bandwidth. If you set the sizes too big then you can run into problems when congestion occurs as it will take longer for the server to detect and make adjustments to overcome it. For this reason, you should not set these values too large. If you know roughly how far away your players and publishers are then you will be fine using manual settings.”

To conclude: if you don’t know how far away you are from your ingest point (in terms of RTT) and you are uncertain about the characteristics of your connection: use SRT.

In the tests that were performed, where possible, the default value of 65,000 bytes was used. For the stream to Australia with RTT of 360 ms, however, the buffer was increased to 260,000 bytes in order to achieve a stable stream.

LATENCY TEST RESULTS

RTMP vs. SRT - Latency vs. RTT



Germany to Australia and back
360 ms RTT
SRT Buffer: 550ms

RTMP	Latency	SRT
9635	ms	2934

RTMP only with increased buffers!

Germany to US West and back
178 ms RTT
SRT Buffer: 280ms

RTMP	Latency	SRT
5318	ms	2250

Germany to US East and back
212 ms RTT
SRT Buffer: 330ms

RTMP	Latency	SRT
4851	ms	2267

Germany to EU Central and back
17 ms RTT
SRT Buffer: 30ms

RTMP	Latency	SRT
4168	ms	1584

Figure 6: *Round trip end-to-end latency results*

As expected, the further the distance to the stream target destination, the bigger the impact on end-to-end latency. It is important to note that these numbers are absolute end-to-end latency figures that include: encoding, transport, decoding and display device delay. They are also two way, round trip latencies, so for example, from Germany to Australia and back to Germany.

Route	RTT	VLC Buffer (RTMP / SRT)	Round Trip End-to End Latency	
			RTMP	SRT
Germany → APAC Sydney → Germany	360 ms	2000 / 250 ms	9635 ms	2934 ms
Germany → US California → Germany	178 ms	700 / 250 ms	5318 ms	2250 ms
Germany → US N. Virginia → Germany	212 ms	250 / 250 ms	4851 ms	2267 ms
Germany → DE Frankfurt → Germany	17 ms	250 / 250 ms	4168 ms	1584 ms

Table 2: *Latency test results*

Latency Test Results in Detail

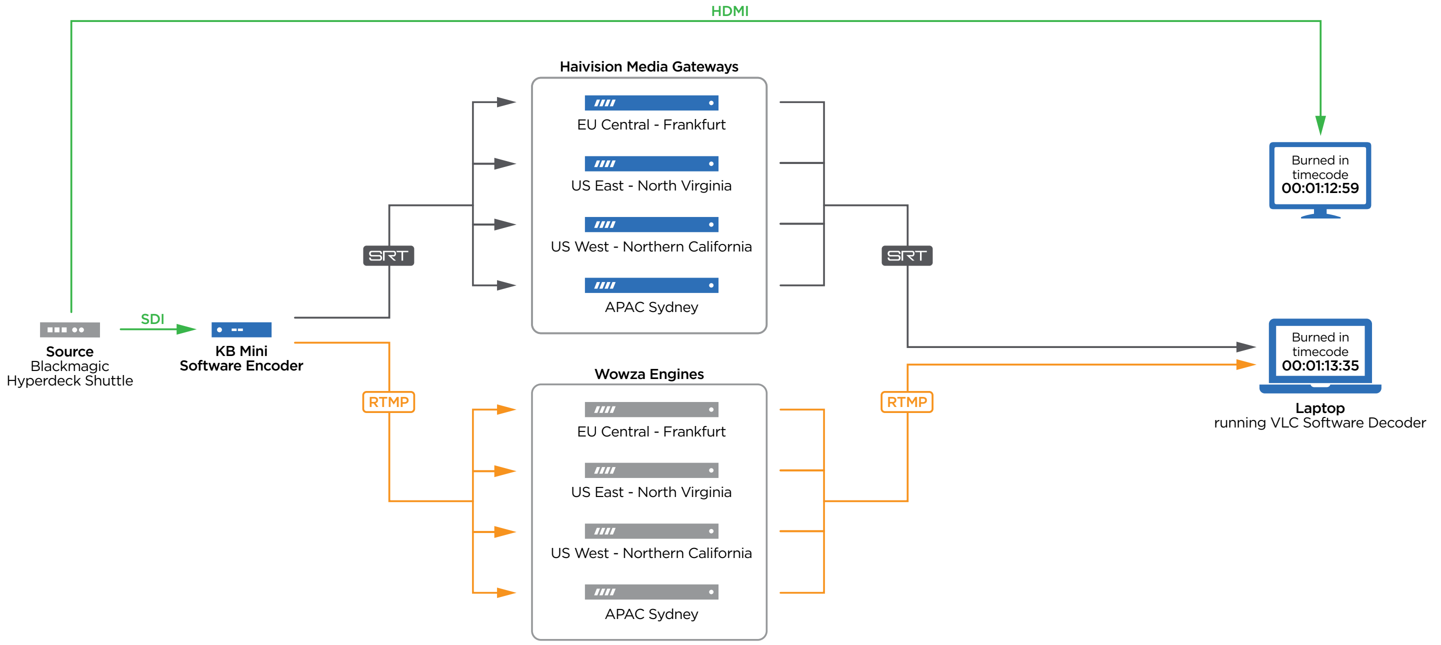


Figure 7: Software encoding / decoding test setup with KB + VLC

The further you are from your stream destination, the longer it takes for the stream to arrive. So it comes as no surprise that streaming from one side of the globe to another, in this case from Germany to Australia, took the most time.

Germany - Sydney - Germany

In order for stable video and audio streams to reach Sydney from Germany using RTMP, the receive-buffer on the Wowza Streaming Engine had to be increased to 260000 bytes. That's four times the default value of 65,000 bytes. Since the test was based on two way streaming, the receiving buffer of the VLC player also had to be increased from the default value of 250 ms to 2,000 ms. Below these values the stream quality was degraded by both audio and video artifacts or did not even play at all.

Germany - California - Germany

Although the round trip time to California was approximately half of the time compared to that of Australia, RTT was not the only influence on latency. The path back to Germany seemed to experience high jitter which resulted in variances in travel time for individual packets. While streaming from Germany to California worked without a hitch using the default buffer size of 65,000 bytes, the return path required an increase of buffering up to 650 ms in VLC player.

Germany - Virginia - Germany

Somewhat surprisingly, the test results showed that the RTT to Virginia was slightly higher (less than 3 frames or 50 ms) than to California, despite the fact that it's clearly a shorter distance on the map from Germany. From this we can conclude that the shortest geographical path might not always be the fastest. Data travels with the speed of light between data centers and their routers.

Depending on the capacity utilization of the data links, your video signal might not always travel along the shortest internet path, but a faster route instead of a more direct route. Although the RTT was marginally higher than the California test, the link was more stable with less jitter and allowed smaller buffers for RTMP. The video stream was stable with default values - 65,000 bytes buffer on the Wowza Streaming Engine and 250 ms for VLC.

Thuringia - Frankfurt - Thuringia

For the Germany stream test from Thuringia to the AWS datacenter in Frankfurt, the RTT was just 17 ms. The round trip end-to-end latency with RTMP did not decrease much compared to the results for Virginia or California. However, the SRT protocol was able to gain more than a second with the lower RTT compared to the US-based locations.

In these tests, compared to RTMP, SRT was 2.5 to 3.2 times faster.

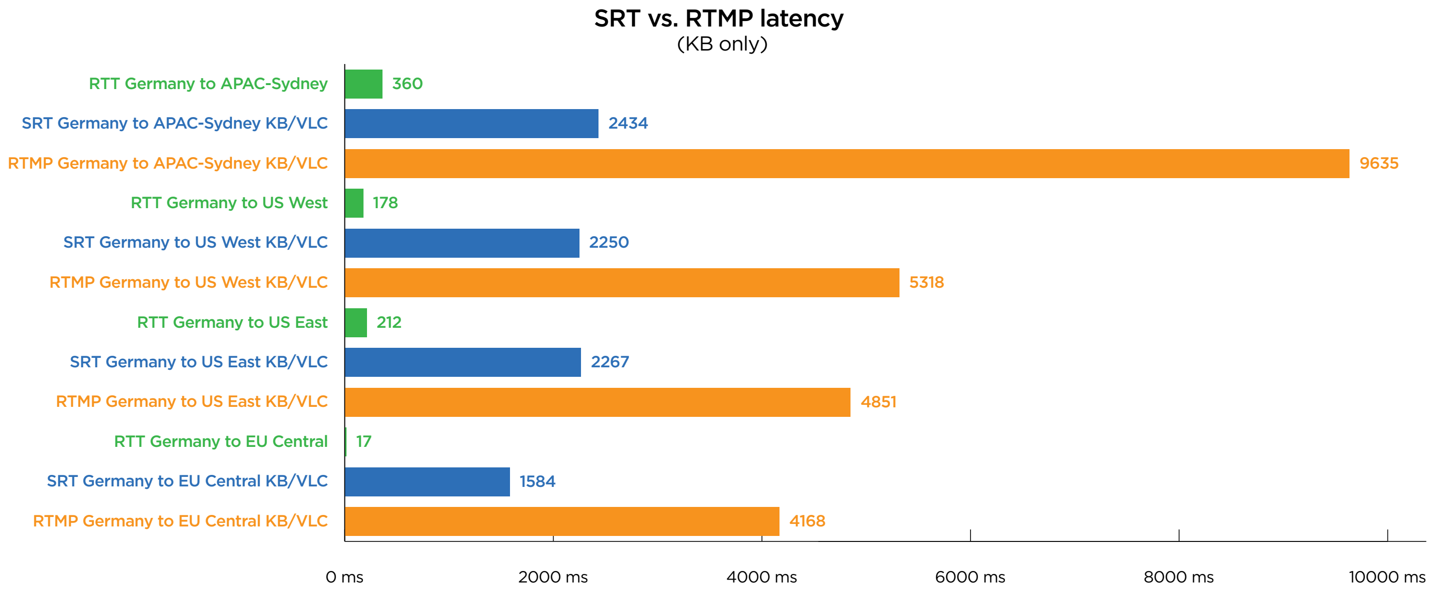


Figure 8: Software encoding / decoding round trip end-to-end latency results

Hardware vs. Software

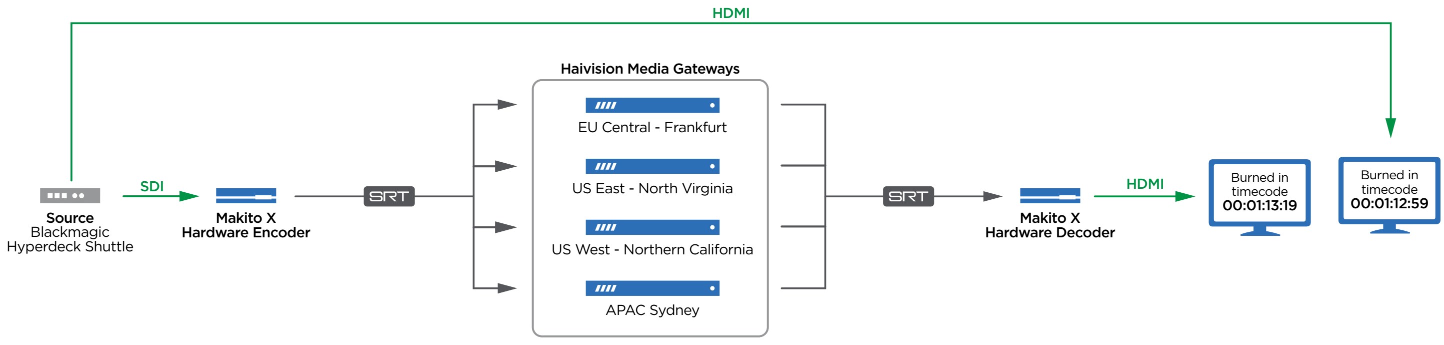


Figure 9: Hardware encoding / decoding test setup with Makito X

So far, we have explored software encoding and decoding and even the fastest results achieved with SRT were still above 1.5 seconds. For use cases where latency is critical, 1.5 seconds is way above the desired threshold for fluid interaction such as live bi-directional interviews.

Latency is noticeable during a news broadcast, for example, when a reporter in the US is interviewed by a European TV channel using satellite transmission. Any delays between a question being posed and an answer being provided are always noticeable. Typically interviewees are media trained and often know the question being asked ahead of time so will start answering before the other party has finished the question. With 1.5 seconds latency however, the delay would be extremely obvious.

While SRT helps to bring down latency significantly on the data transport, hardware encoders and decoders, such as Haivision's Makito X Series, will also speed up the video encoding and decoding. The following chart shows the results of a Haivision KB encoder to VLC via RTMP and SRT compared to a Haivision Makito X encoder to decoder via SRT.

The results using the Makito X encoder and decoder with SRT show a dramatic difference. The two-way end-to-end latency to Australia and back decreases from 9.6 seconds to 1.6 seconds when using dedicated hardware encoding and decoding. Within Germany the result is as low as 333 milliseconds for SRT, while RTMP in software needs 4.1 seconds. This is **12 times faster** and can be considered as ultra low latency.

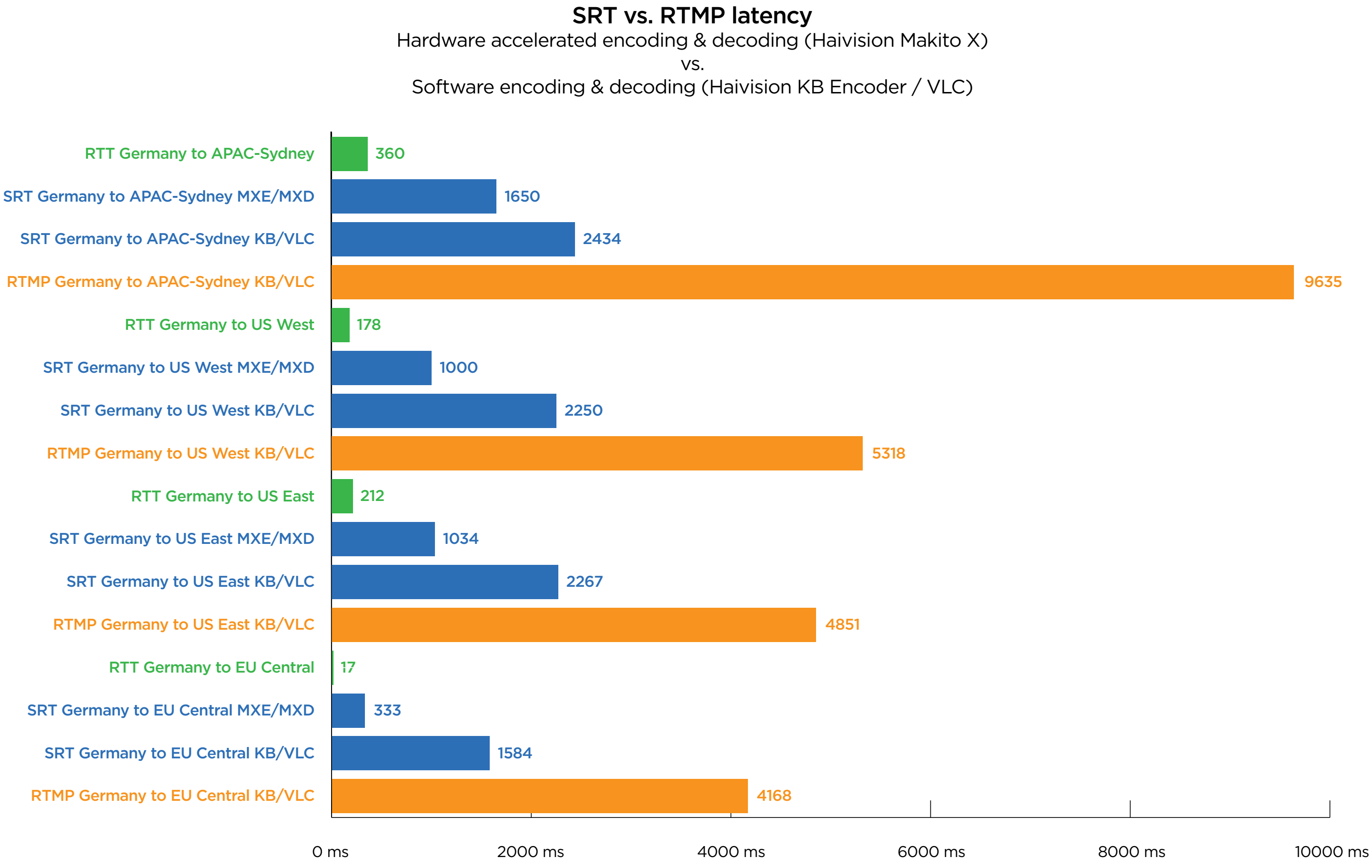


Figure 10: *Hardware encoding / decoding round trip end-to-end latency results*

COMPARING THE MAXIMUM BANDWIDTH OF RTMP AND SRT PROTOCOLS

The test results prove that SRT has a significant impact on reducing video latency. But what about its impact on video quality? An easy way to improve video and audio quality is to simply increase the bandwidth used for streaming. But what is the maximum bandwidth for long distance streams and how far you can get with high bandwidth streams?

Bandwidth Test Setup

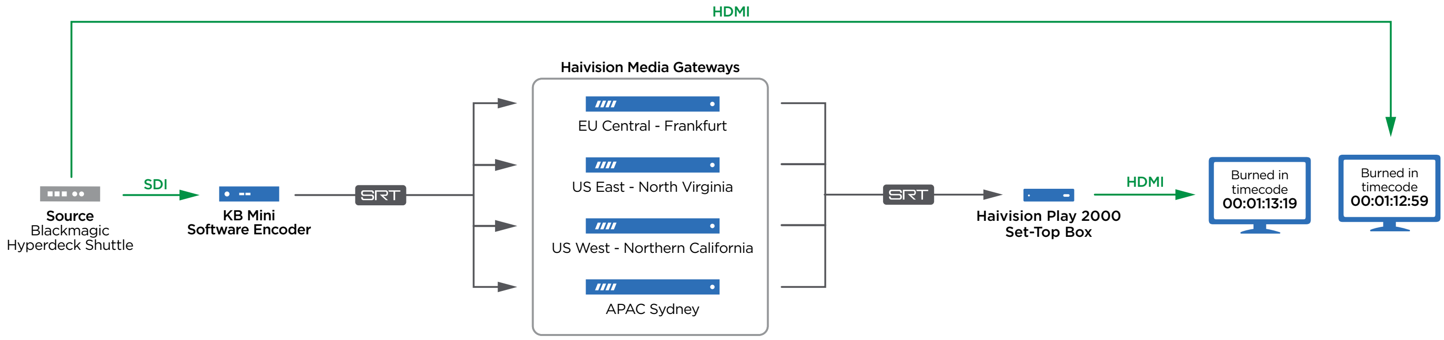


Figure 11: *Bandwidth test setup*

To test high bandwidth streaming, we chose a location with an excellent internet connection, the Microsoft Production Studios in Redmond, Washington. The next major internet node is just 2 hops away and all connected devices had a true 1 Gbps connectivity to the internet.

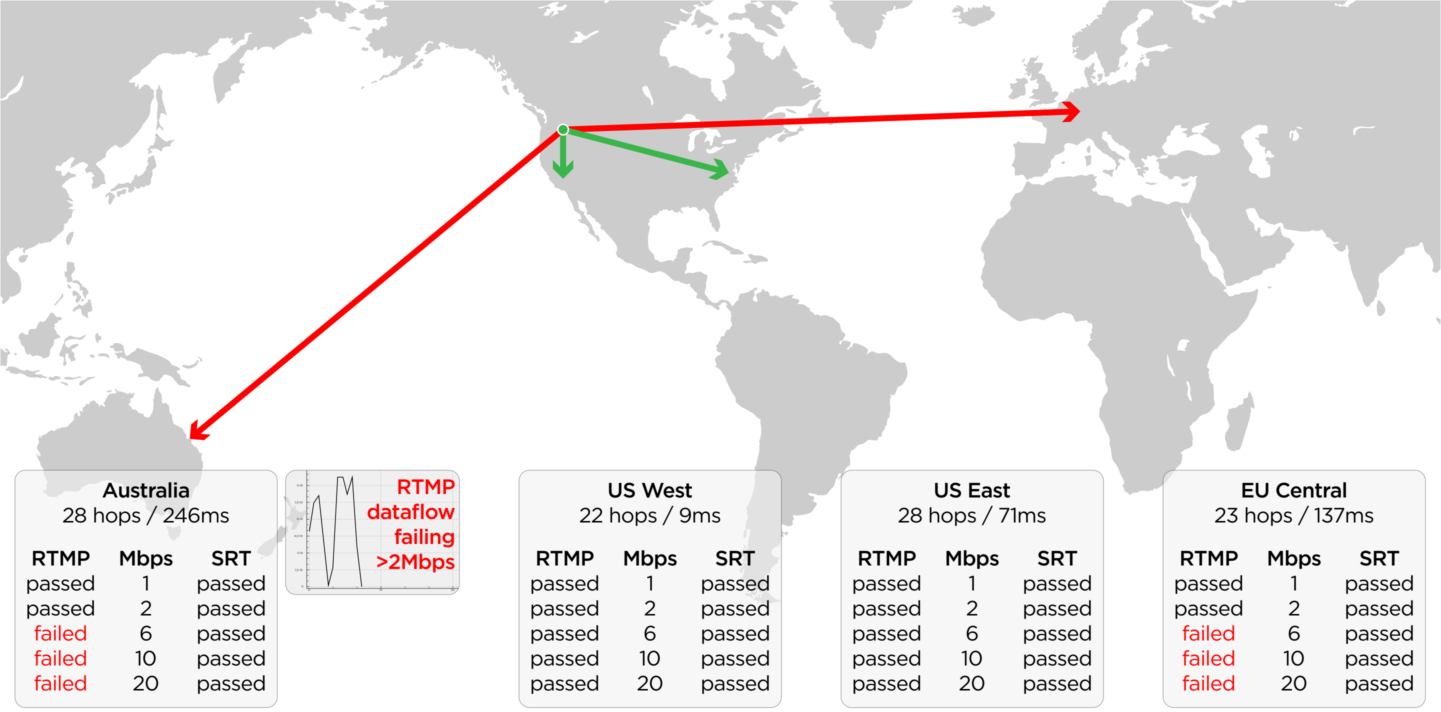
Given that low latency remained the focus of testing, the buffer settings remained unchanged when increasing the bandwidth during the tests. The buffer sizes were tested with a 2 Mbps stream. Once it was stable, tests were conducted using a 1, 2, 6, 10 and 20 Mbps stream.

A Haivision KB 5.4 encoder provided the stream which was sent to a Haivision Media Gateway in AWS data centers in California and North Virginia in the US, Frankfurt in Germany and Sydney, AustraliaUS-east. To judge the quality of the video, the stream was sent back to Redmond via SRT and played out on a Haivision Play 2000 Set Top Box.

Bandwidth Test Results

The test results were conclusive. SRT experienced no issues streaming up to 20 Mbps to any region in the world. RTMP worked well when both the sender and receiver were on the same continent, North America in this case. From Redmond, it was possible to send up to 20 Mbps using RTMP to California or Virginia. Streaming to Europe and Australia, however, was not possible at bitrates above 2 Mbps. With SRT, even higher bitrates would have been possible, but as RTMP had already failed at long distances via public internet at low bitrates, 20 Mbps was a good upper limit for this particular test.

RTMP vs. SRT - How far can you go via public internet?



Australia
28 hops / 246ms

RTMP	Mbps	SRT
passed	1	passed
passed	2	passed
failed	6	passed
failed	10	passed
failed	20	passed

US West
22 hops / 9ms

RTMP	Mbps	SRT
passed	1	passed
passed	2	passed
passed	6	passed
passed	10	passed
passed	20	passed

US East
28 hops / 71ms

RTMP	Mbps	SRT
passed	1	passed
passed	2	passed
passed	6	passed
passed	10	passed
passed	20	passed

EU Central
23 hops / 137ms

RTMP	Mbps	SRT
passed	1	passed
passed	2	passed
failed	6	passed
failed	10	passed
failed	20	passed

Figure 12: *Bandwidth test results*

CONCLUSION

In terms of end-to-end latency and maximum transferable bitrate, SRT far outperforms RTMP. Although these tests could have been conducted in a laboratory environment using tools to add packet loss and high jitter, we purposefully chose to perform them using the public internet. Real world conditions are difficult to replicate in a laboratory and would always be an approximation.

Looking to the future, there's a new Haivision innovation on the horizon, SRT Hub, which will offer an alternative to streaming via public internet. SRT Hub, powered by SRT, is a cloud-based routing service which leverages the scalability and reach of the global Microsoft Azure network. This will minimize the number of hops to travel via the internet to ingest and egress points, driving latency down to even lower levels.

If you're interested in learning more about Haivision's
Emmy® Award winning video streaming solutions, contact us.

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