



L I V E V I D E O D R I V E T E S T

NanoPing vs. SRT Connection Bonding

Frame latency and picture glitches of a live 1080p30 camera stream carried over two cellular links from a moving car: nine laps of the same road, measured against a 300 ms application budget.

Version 1 - 25 September 2026

SRT (Secure Reliable Transport) is an open-source protocol originally developed by Haivision. NanoPing is not affiliated with, endorsed by, or sponsored by Haivision or the SRT Alliance. Results reflect the specific implementation (srt-xtransmit / libsrt), configuration and test conditions described in this report and should not be interpreted as representative of all SRT deployments.

Executive summary

A camera in a car streamed 1080p30 video at 3 Mbit/s over two cellular links to a fixed server, nine laps of the same 2.2 km loop: three through NanoPing's bonded tunnel (160 ms jitter buffer) and six through SRT connection bonding. The receiving application has a 300 ms budget: a frame that arrives later, or never, is a gap in the picture. SRT ran three laps at each of two receive delays: 160 ms, its configuration guideline of $4 \times$ the round-trip time, and 270 ms, the most the budget allows once 20 ms of one-way network latency and 10 ms of headroom are set aside.

Pooled over the three laps of each setting, NanoPing had the fewest glitches, the lowest latency and the shortest outages. An interruption is one continuous break in the picture: a run of consecutive frames that arrived after 300 ms or never. NanoPing's picture broke 4 times in three laps, brief hiccups of at most 0.9 s and 1.4 s in total; SRT's broke 136 times at 160 ms and 126 times at 270 ms, with freezes of up to 23 s.

THREE LAPS POOLED	● NANOPING	■ SRT 270 MS	◆ SRT 160 MS
Total outage time (frames late or lost)	✓ 1.4 s	97.7 s	16.3 s
Interruptions (breaks in the picture)	✓ 4	126	136
Glitches (frozen or corrupt frames)	✓ 478	4,790	4,244
Average latency	✓ 162 ms	296 ms	193 ms
99 of 100 frames arrive within	✓ 165 ms	354 ms	350 ms

Three laps pooled per setting; a check marks the best of the three. Glitches: frames the viewer did not see as sent once the 300 ms deadline is enforced - frozen (late or lost, the previous frame held) or corrupt (built on a missing frame) - from a replay of every stream. Outage time: frames late or lost, at 33 ms each. Same car, camera, modems and road for all nine laps; nothing trimmed.

TEST SETUP

How the test was run

A car carried a camera, a small computer and two cellular modems (TDC and Telenor DK). The computer encoded the camera feed and streamed it live to a fixed server through the transport under test. The car drove the same loop for every lap; only the transport changed.

THE STREAM

- 1080p at 30 frames per second, H.264 at 3 Mbit/s, sent as RTP over UDP. Every packet carries its frame's capture time, so the receiver knows each frame's latency exactly: camera to application, transport included.
- The car keeps a reference copy of what it sent, so every received frame can be compared with the original.

THE TWO TRANSPORTS

- SRT connection bonding: srt-xtransmit (libsrt) with the two links in a backup group - one link carries the stream, the other stands by. Three laps at each of two receive delays. 160 ms follows SRT's configuration guideline of setting the latency to $4 \times$ the round-trip time: 4×40 ms on these links. 270 ms is the most the 300 ms application budget allows: 300 ms less the 20 ms one-way network latency and 10 ms of headroom.
- NanoPing: NanoPing's bonded tunnel (plumr 45.0.0), using both links at once, with a 160 ms jitter buffer and a 270 ms timeout. Three laps.

WHAT IS MEASURED

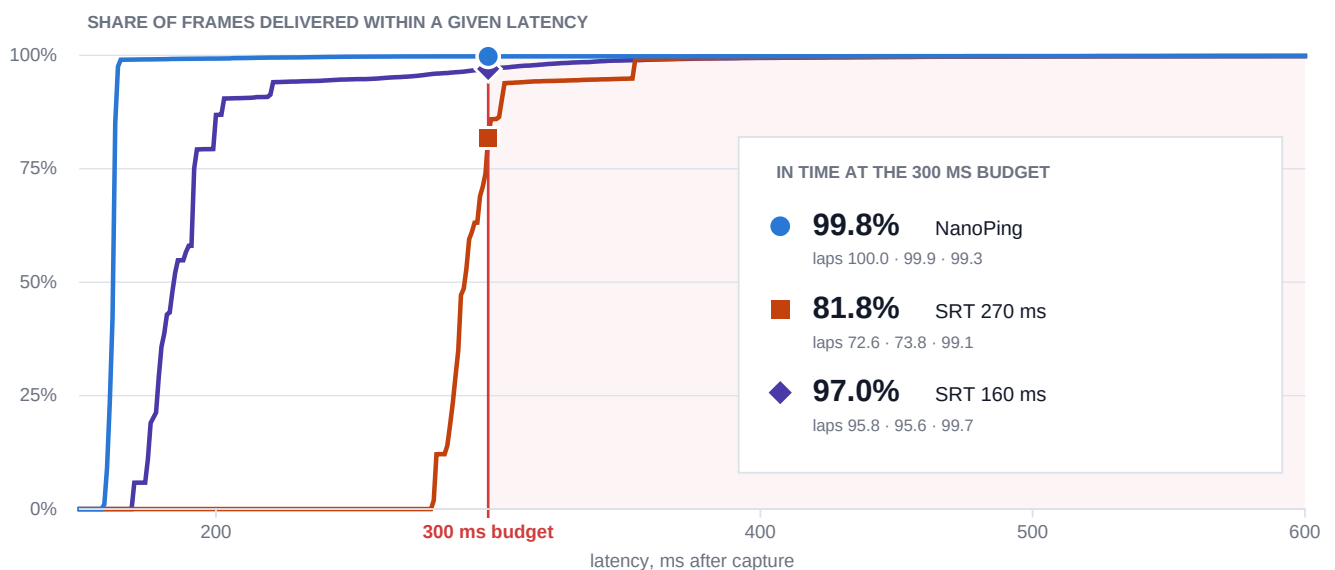
- ✓ The 300 ms budget: a frame is usable if it arrives within 300 ms of capture. Later or never, the viewer sees a gap; outage time adds those gaps up.
- ✓ Glitches: every stream replayed with the deadline enforced - late frames dropped before decoding - and every frame shown compared with the reference. Frozen or corrupt frames are glitches.

RESULTS

Frame latency against the 300 ms budget

The application waits 300 ms for each frame and then moves on. What counts is how many frames made it in time, and how long the gaps were when they did not.

FRAMES IN TIME		OUTAGE TIME OVER THREE LAPS		INTERRUPTIONS	
● NanoPing	99.8%	● NanoPing	1.4 s	● NanoPing	4
■ SRT 270 ms	81.8%	■ SRT 270 ms	97.7 s	■ SRT 270 ms	126
◆ SRT 160 ms	97.0%	◆ SRT 160 ms	16.3 s	◆ SRT 160 ms	136



Share of frames delivered within a given latency, three laps pooled. NanoPing's curve is a near-vertical step at 162 ms because almost every frame arrives at the same moment; at the 300 ms budget 99.8% are in time. SRT at 270 ms sits at 280 ms at best, so any disturbance pushes it past the budget; SRT at 160 ms starts at 169 ms, but 3.0% of its frames arrive late or never.

THREE LAPS POOLED

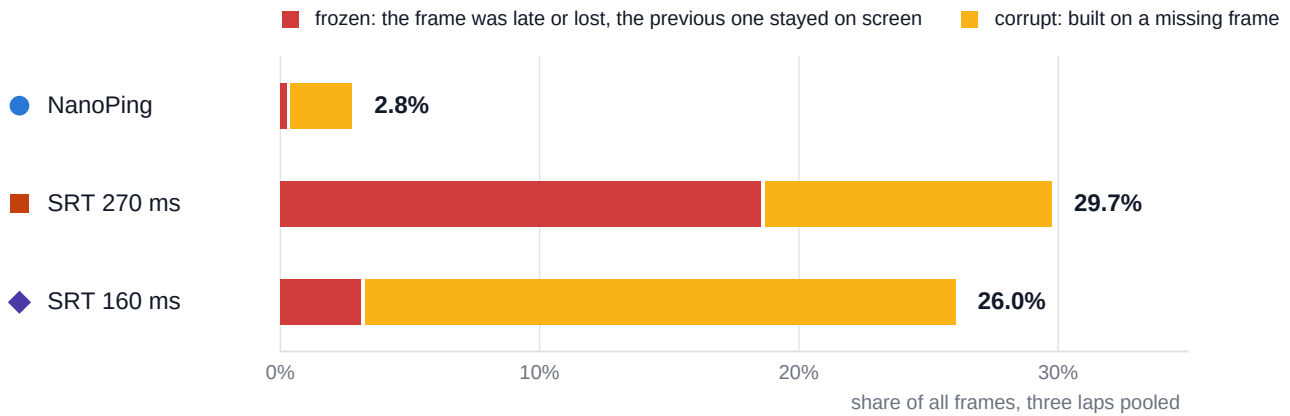
	FRAMES IN TIME	WORST LAP	OUTAGE TIME	INTERRUPTIONS	LONGEST
NanoPing	99.8%	99.3%	1.4 s	4	0.87 s
SRT 270 ms	81.8%	72.6%	97.7 s	126	22.8 s
SRT 160 ms	97.0%	95.6%	16.3 s	136	0.83 s

Interruptions: stretches of consecutive late or lost frames. Outage time: their total, at 33 ms per frame. Longest: the longest single interruption.

RESULTS

Glitches under the 300 ms deadline

A glitch is a frame the viewer did not see as it was sent. Every stream was replayed with the deadline enforced - late frames dropped before decoding, the previous frame held in their place - and every frame shown was compared with what the car sent.



THREE LAPS POOLED

	FROZEN	CORRUPT	GLITCHES	SHARE OF FRAMES
NanoPing	45	433	478	2.8%
SRT 270 ms	2,986	1,804	4,790	29.7%
SRT 160 ms	509	3,735	4,244	26.0%

The stream has a keyframe only every 10 s, so one dropped frame corrupts the frames that depend on it until the next keyframe. NanoPing dropped far fewer frames (45 against 509 and 2,986), so far fewer were corrupted.

Conclusion

Measured the way a receiving application measures it - did the frame arrive in time or not - NanoPing delivered 99.8% of frames within the 300 ms budget over three laps of the same loop, against 97.0% for SRT at 160 ms and 81.8% at 270 ms. Its outages were 4 hiccups totalling 1.4 s, none longer than 0.9 s; SRT's ran to 16.3 s and 98 s, with freezes of up to 23 s.

Replayed with the deadline enforced, the viewer's picture was frozen or corrupt for 2.8% of NanoPing's frames, against 26.0% and 29.7%. With a keyframe only every 10 s, every dropped frame costs the frames that depend on it - and NanoPing dropped far fewer.

The same car, the same road and the same links for all nine laps. The road was the same; the outcomes were not.