Tencent 腾讯 START云游戏

nsdi '26

NSDI 2026 • Tencent START cloud gaming

From Source to Solution: Tackling Packet Losses in Large-scale Cloud Gaming Systematically and Precisely

Loss Avoidance, Detection, and Recovery

Jing WANG*, Xiao KONG*, Yunzhe NI, Nian WEN, Jiaxing ZHANG, Congcong MIAO, Honghao LIU†

Tencent Inc.

Packet loss becomes a user-visible stall.

66.2%

of frames above `100 ms` are loss-affected

Once a frame is lossy, recovery dominates the remaining delay.

wired	71.6%
Wi-Fi	58.3%

Loss is not just a packet-level metric. It is a Quality-of-Experience problem.

WHY THE EVIDENCE IS CREDIBLE

START云游戏

Our production stack is measurable and replaceable end to end.

01

Protocol

TCP vs. RUDP (mimics the kernel TCP stack)

→

02

Congestion control

same deployment, changed logic

→

03

Loss detection

receiver-side NACK vs. sender-side SACK

→

04

Recovery

RTX, WebRTC-FEC, Hairpin

~200,000 hours
production traces

Controlled substitutions
not isolated anecdotes

Real user traffic
wired and Wi-Fi cloud gaming

We can challenge the conventional stack one assumption at a time.

THE CONVENTIONAL STACK

START云游戏

The stack is reasonable. The operating envelope changed.

WebRTC	UDP	Pudica NSDI'24	NACK	FEC
real-time media stack	transport path	cloud-gaming congestion control algorithm	receiver feedback	redundant recovery

GCC is the common WebRTC congestion control algorithm; our production baseline already uses Pudica.

TRADITIONAL REAL-TIME VIDEO

< 2 Mbps

< 500 ms, reasonable defaults for the original setting

vs

CLOUD GAMING TARGET

> 20 Mbps

tail above 100 ms visibly hurts interaction

Production evidence says the old defaults need to be re-examined.

START云游戏

The network is not UDP-neutral.

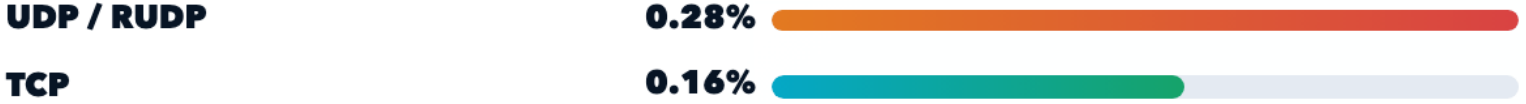
COMMON DEFAULT

UDP should be the low-latency path.

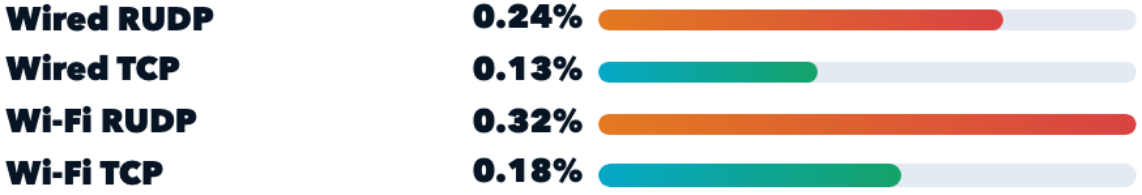
Production traces show the path itself changes loss.

44.8% Lower

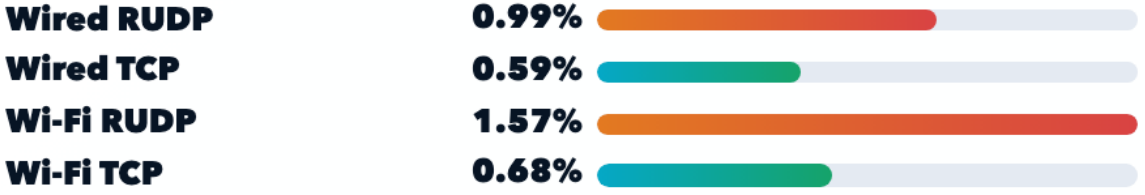
packet loss for TCP than UDP/RUDP (Reliable UDP) under the same logic



PACKET LOSS RATE · RELATIVE WITHIN THIS PANEL



LOSSY FRAME RATE · RELATIVE WITHIN THIS PANEL



Source material: paper Figure 5 and Figure 6

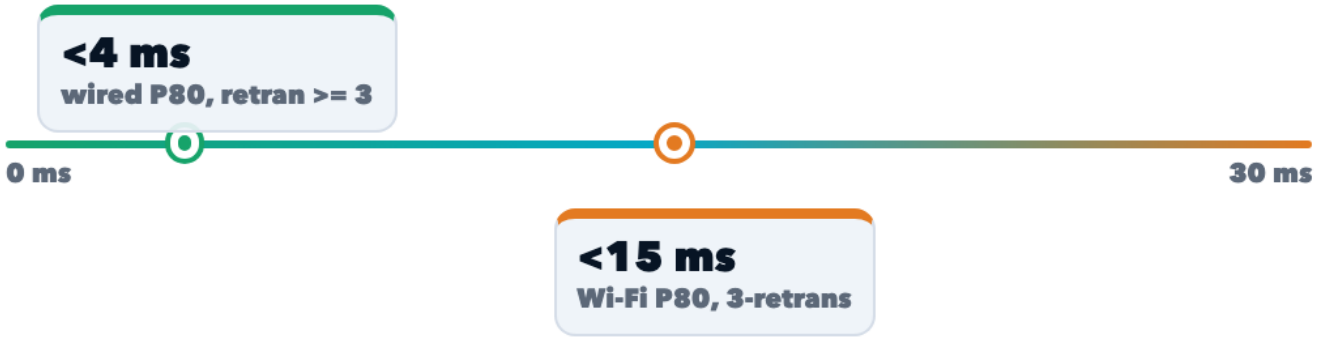
Many losses are created by the transport choice itself.

START云游戏

Low queueing does not mean congestion is absent.

AFTER PUDICA

loss still appears at low queueing delay



Most losses occur before queue delay becomes a strong warning signal.

LOAD DIAGNOSIS

Lower offered load still cuts loss

39.4%

lower loss after a 2 Mbps cap, even with Pudica already deployed

80%
of cases have loss-rate ratio above 1

75%
of cases have in-flight ratio above 1.875

Source material: paper Figure 7, Figure 9, and Figure 35

Delay-based congestion control can still leave a large congestion-loss surface.

Receiver-side NACK detection is too weak for the tail.

loss occurs

receiver ambiguity

late signal

stall risk

Source material: paper Figure 11

FIGURE 11 RESULT

Frames above 100 ms

34.7%

fewer >100ms than NACK

Detection quality directly shapes recovery latency.

FINDING 4 · RECOVERY

START云游戏

FEC helps, but it is not enough by itself.

WHEN FEC IS ENABLED

99.1%

recoverability for FEC-enabled lossy frames

48.4%

of lossy frames were actually FEC-enabled

82.3%

of loss events last fewer than 3 frame intervals

FEC WORKS BEST WHEN LOSS IS

random · sparse · predictable

PRODUCTION LOSS OFTEN IS

short · bursty · hard to predict

Source material: paper Figure 12-14 and Section 3.4

FEC should be targeted after avoid and detect have reduced the problem.

DESIGN DIRECTION

START云游戏

LADR (Loss Avoidance, Detection, and Recovery) turns four measured loss sources into four design choices.

01 • STATIC AVOIDANCE

TCP first

TCP is the default path because production networks are not UDP-neutral.

Keep RUDP (Reliable UDP) only when FEC can pay off.

02 • RUNTIME AVOIDANCE

Pudica + loss-based bitrate control

Loss-Based Bitrate Control, or LBBC, complements delay signals when shallow buffers overflow first.

Use the stricter rate when loss becomes the clearer congestion signal.

03 • FAST DETECTION

Sender-side loss detection

SACK-based detection exposes retransmission loss and tail loss earlier.

Replace late receiver-side NACK repair requests.

04 • TARGETED RECOVERY

Reactive rate limit + FEC

Reactive Rate Limit, or RRL, slows severe loss events; FEC adds redundancy only for hard residual cases.

Recovery handles a smaller and better-timed problem.

Conceptual source: paper Figure 18

Each mechanism is selected by the measured loss source it addresses.

DEPLOYMENT ARCHITECTURE

START云游戏

A TCP-first proxy turns findings into deployable choices.

TCP first
multi-connection pool

Loss-based bitrate control
delay + shallow-buffer loss signals

Sender-side detection
selective acknowledgments

Reactive rate limit + FEC
selective residual recovery

Conceptual source: paper Figure 22

Data, not protocol preference, chooses the default path and recovery scope.

PRODUCTION RESULTS

START云游戏

LADR wins where users feel it: at the frame-level tail.

PACKET LOSS

0.049%

average packet loss

20% lower

than UDP-based WebRTC / Hairpin baselines

LOSSY FRAMES

0.151%

frames needing loss recovery

71% / 85% lower

than WebRTC / Hairpin

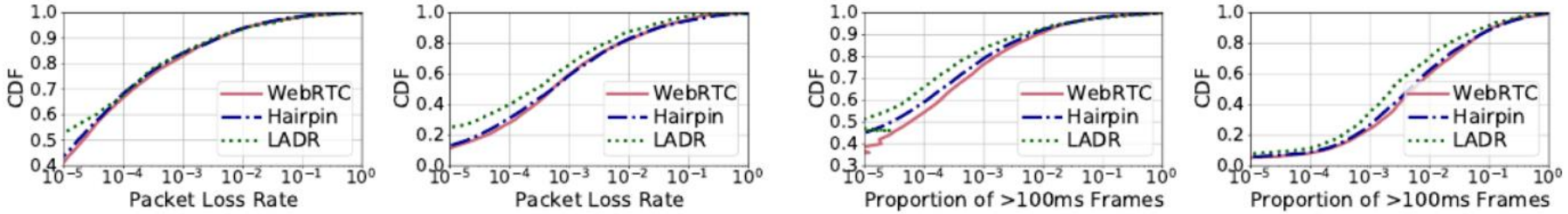
TAIL FRAMES

0.115%

frames above `100 ms`

60%
lower than WebRTC

55%
lower than Hairpin



Source material: paper Figure 21 and Figure 23

The result is not an isolated metric bump; it is a baseline-relative production win.

WHY THE GAINS HAPPEN

START云游戏

The step-by-step breakdown shows where the tail drops.

Default

0.192%

frames above 100 ms

baseline

+ Loss-based bitrate control

0.185%

shallow-buffer congestion signal

−3.6%

+ Reactive rate limit

0.143%

severe-loss pacing and repair priority

−25.5%

Full LADR

0.115%

opportunistic FEC

−40.1%

All reductions are relative to the default system. Raw packet loss slightly rises from `0.039%` to `0.049%` after opportunistic UDP-based FEC, but loss-affected and tail frames still fall sharply.

Source: paper performance improvement breakdown table, average values shown for presentation clarity

The biggest story is problem reshaping, not stronger coding.

CONCLUSION

START云游戏

LADR treats packet loss as a full-pipeline systems problem.

01

Avoid first

Choose the lower-loss path and reduce shallow-buffer congestion before loss becomes visible.

02

Detect earlier

Move loss detection earlier so recovery still has time to protect interactive frames.

03

Recover selectively

Use rate limiting and FEC only for the residual loss patterns where they can help.

The sequence matters: avoid first, detect earlier, recover selectively.

START云游戏

TENCENT START RESEARCH PROGRAM

This paper is one piece of a multi-year attack on cloud-gaming latency.

NSDI '23
AFR
Adaptive Frame Rate
Layer: rendering / decoder · target: stutter

NSDI '24
Hairpin
Differentiated loss recovery
Layer: recovery · target: late frames

NSDI '24
AUGUR
Mobile multipath transport
Layer: multipath · target: Wi-Fi tail

NSDI '24
Pudica
Near-zero queueing congestion control
Layer: congestion control · target: queueing

NSDI '26
BLADE
Adaptive Wi-Fi contention control
Layer: Wi-Fi access · target: packet droughts

NSDI '26
LADR
Loss avoidance, detection, recovery
Layer: loss pipeline · target: source to solution

Public sources: USENIX NSDI pages for AFR, Pudica, Hairpin, AUGUR, BLADE, and LADR.

Special thanks toMr. Honghao Liufor leading our ultra-low-latency video transport research.

Authors across the six START papers

Honghao Liu, Rui Han, Xiao Kong, Tingfeng Wang, Jing Wang, Chenglei Wu, Fengqian Guo, Yunzhe Ni, Nian Wen, Jiafeng Chen, Longwei Jiang, Jianjun Xiao, Changxi Zheng, Congcong Miao, Xin Liu, Xue Wei, Zili Meng, Yuhan Zhou, Shibo Wang, Jiaxing Zhang, Jing Chen, Yixin Shen, Bo Wang, Mingwei Xu, Chenren Xu, Shusen Yang, Venkat Arun, Hongxin Hu, Liying Wang, Cong Zhao, Xuesong Yang, Yuxin Liu, Hancheng Lu, Chang Wen Chen, Yaxiong Xie

The common thread is production evidence: find the latency source, then change the right layer.

nsdi '26