



VK Play

**Ultra-low latency
streaming. In pursuit
for the perfect
video delivery.**



Chernikov Kirill

Head of the VK Play Cloud
Client Development Team



About team and project

In cloud gaming
development since
2013

3-6 members
in development team



What is ultra-low latency streaming?

Application type	Latency
Downloading media content	Any
Video on demand (VOD)	1-10 sec.
Video calls	< 500 ms.
Real-time control applications (ultra-low latency applications)	< 100 ms. (50-70 ms.)

What is the perfect video delivery?

01

Minimum
possible delay

02

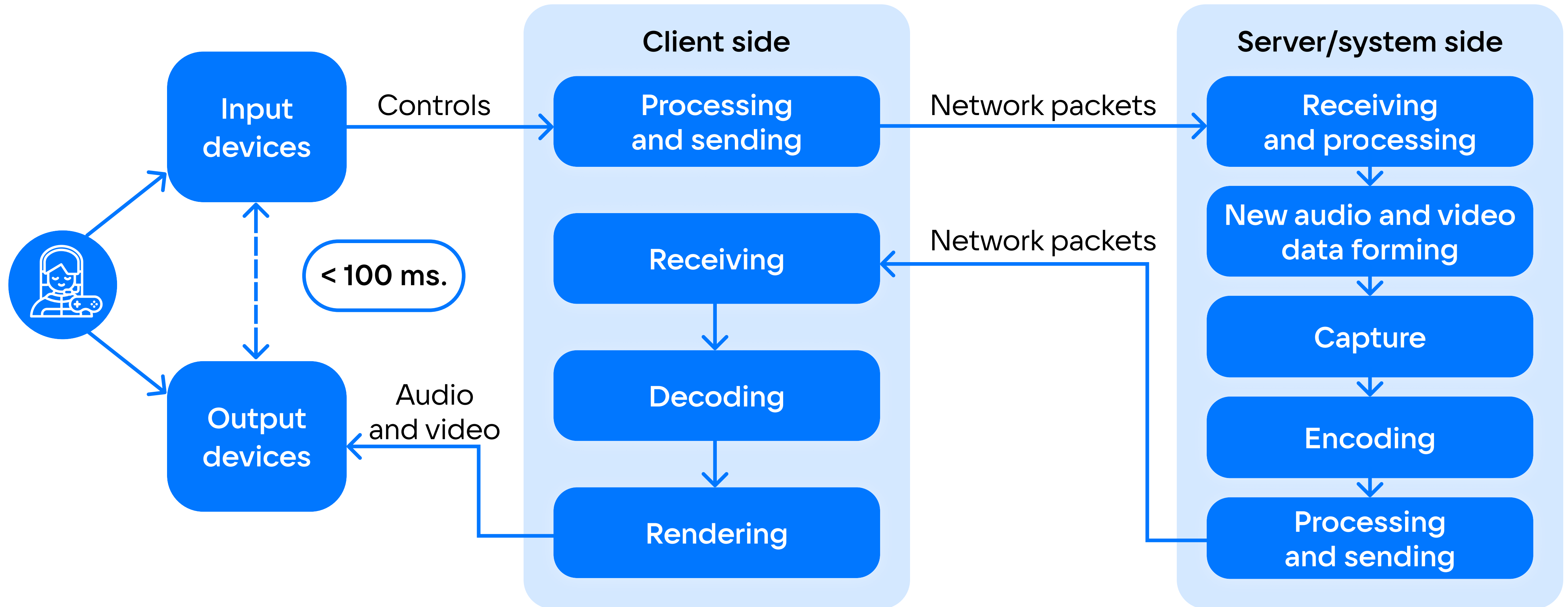
Target video
bitrate or
maximum possible
bitrate

03

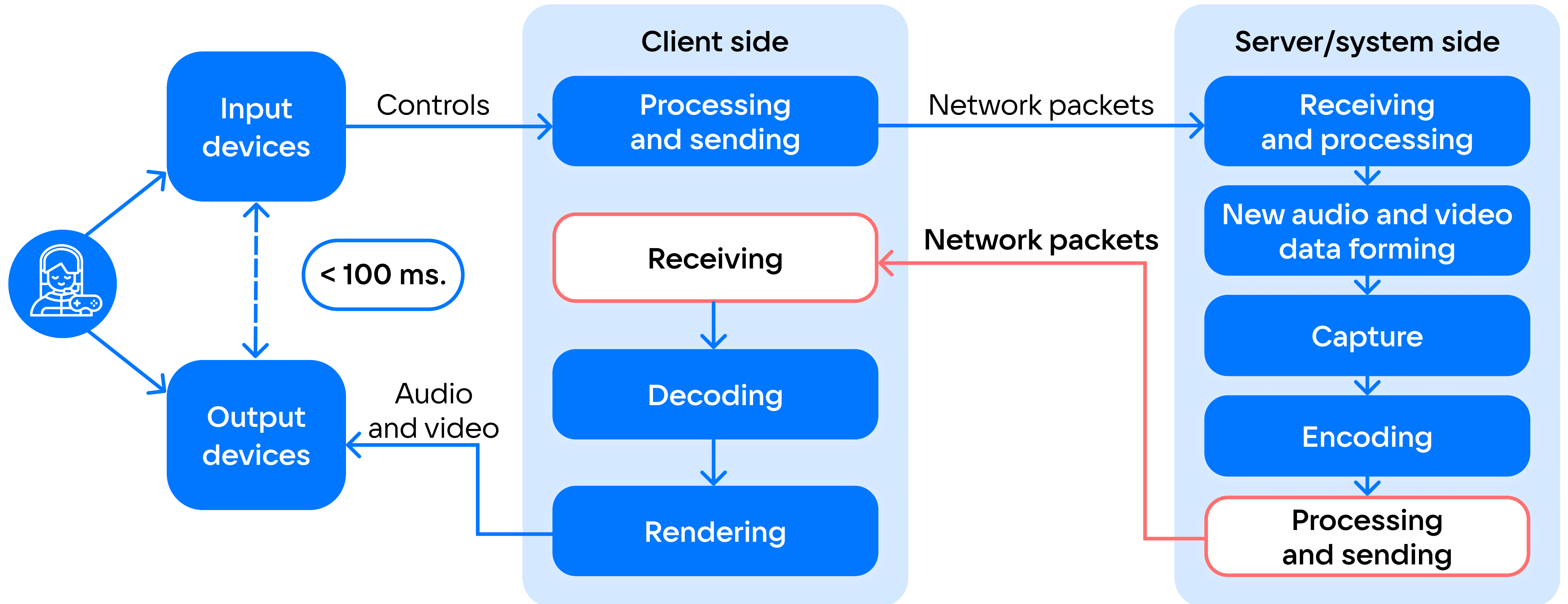
No decoding
problems

Ultra-low latency application scheme

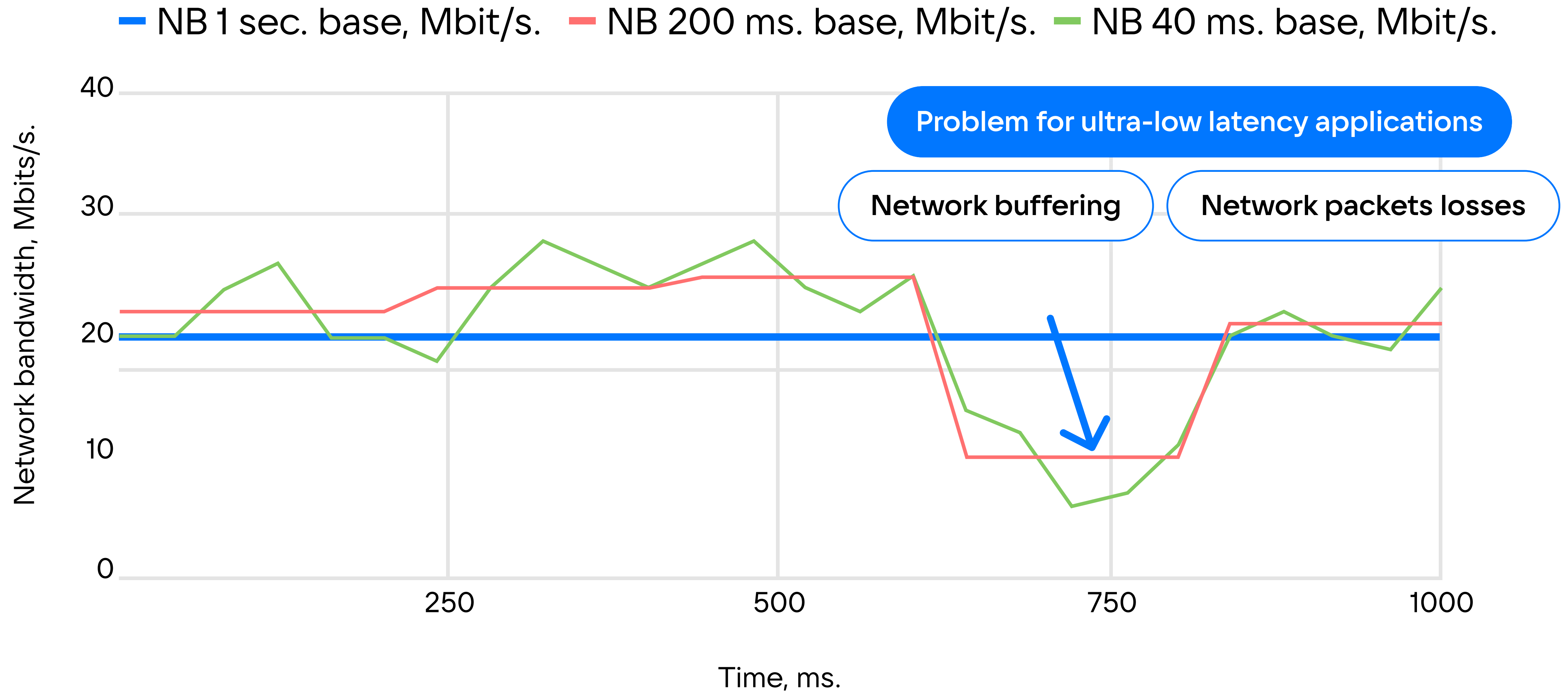
Cloud gaming example



Video, network and key stages



Network bandwidth (NB) estimation



Methods of dealing with network problems

- 1 ABR (adaptive bitrate algorithm)**
 - Network bandwidth estimation
 - Video sending management
- 2 BBR (bottleneck bandwidth and RTT)**
 - Network buffering estimation
 - Network packets losses prediction
- 3 Right video encoding parameters**
 - Reducing the peak load on the network

See for details VideoTech 2022:
Ultra-low latency. Принципы
и механизмы передачи видео
в Cloud Gaming



Methods of dealing with network problems

4 Pacing

- Reducing the peak load on the network
- Video sending management

5 FEC (forward error correction)

- Recovery of network packets losses

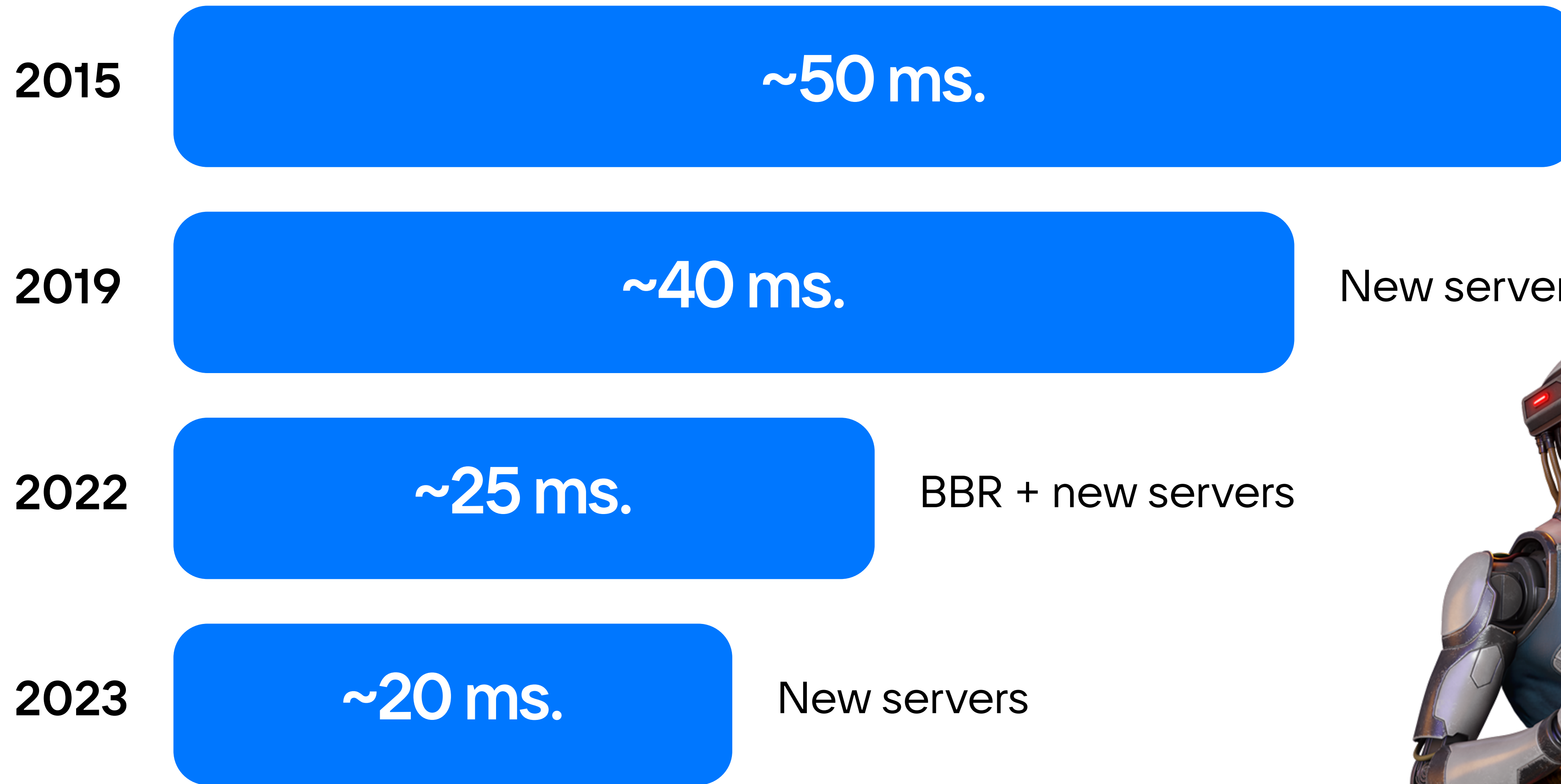
6 Network packets retransmit

- Recovery of network packets losses

See for details VideoTech 2022:
Ultra-low latency. Принципы
и механизмы передачи видео
в Cloud Gaming



Daily average RTT



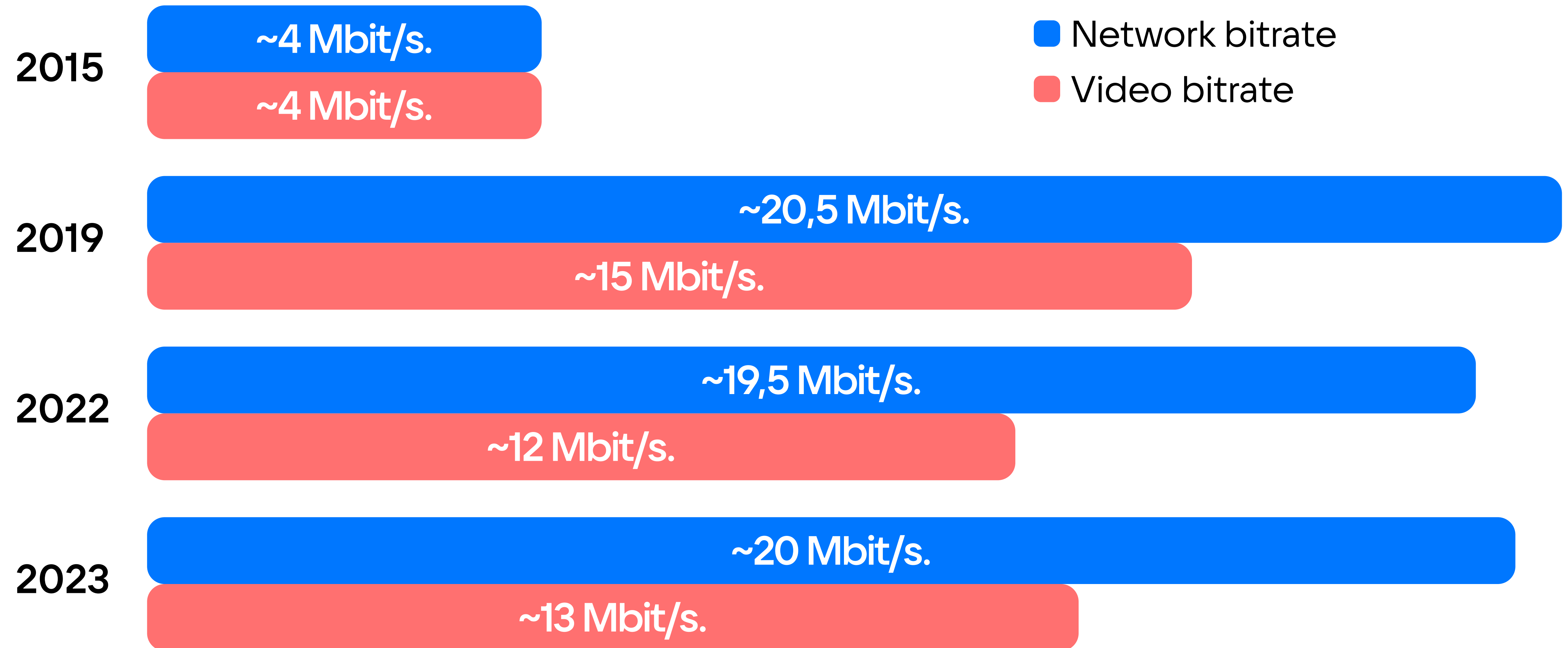
New servers

BBR + new servers

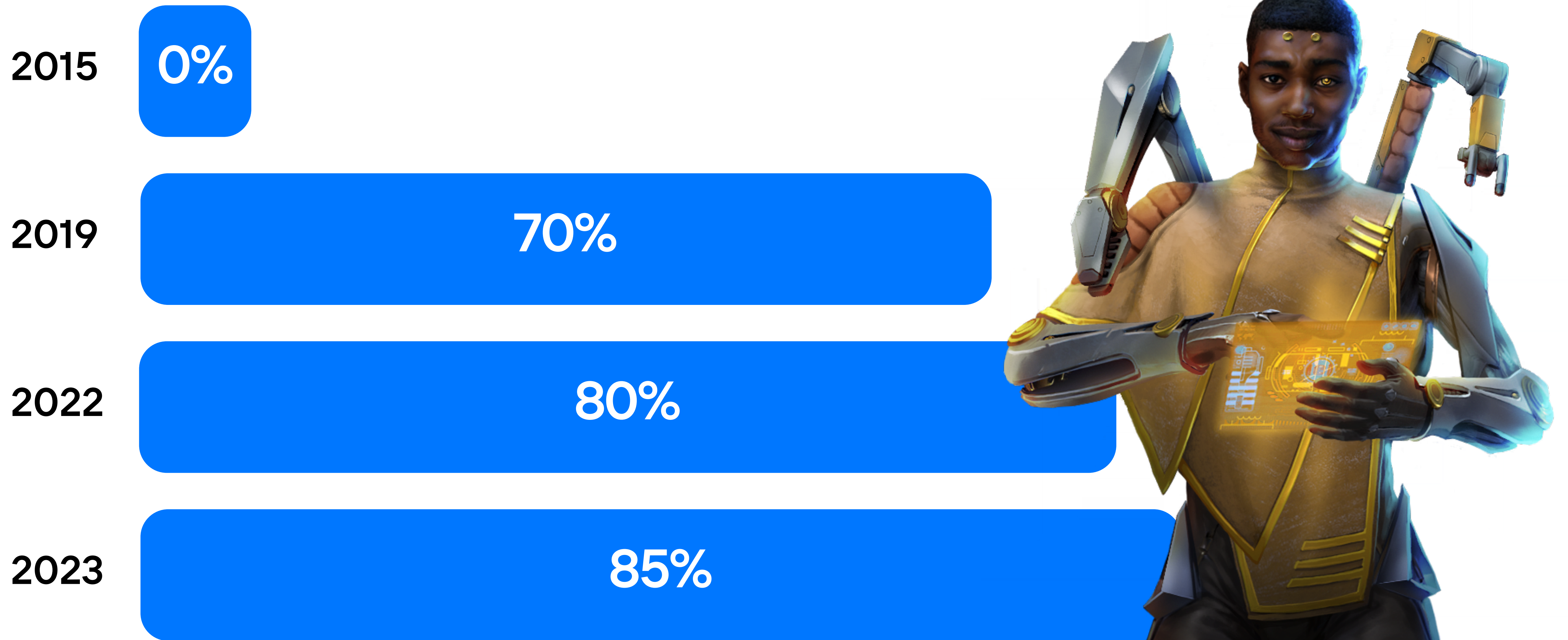
New servers



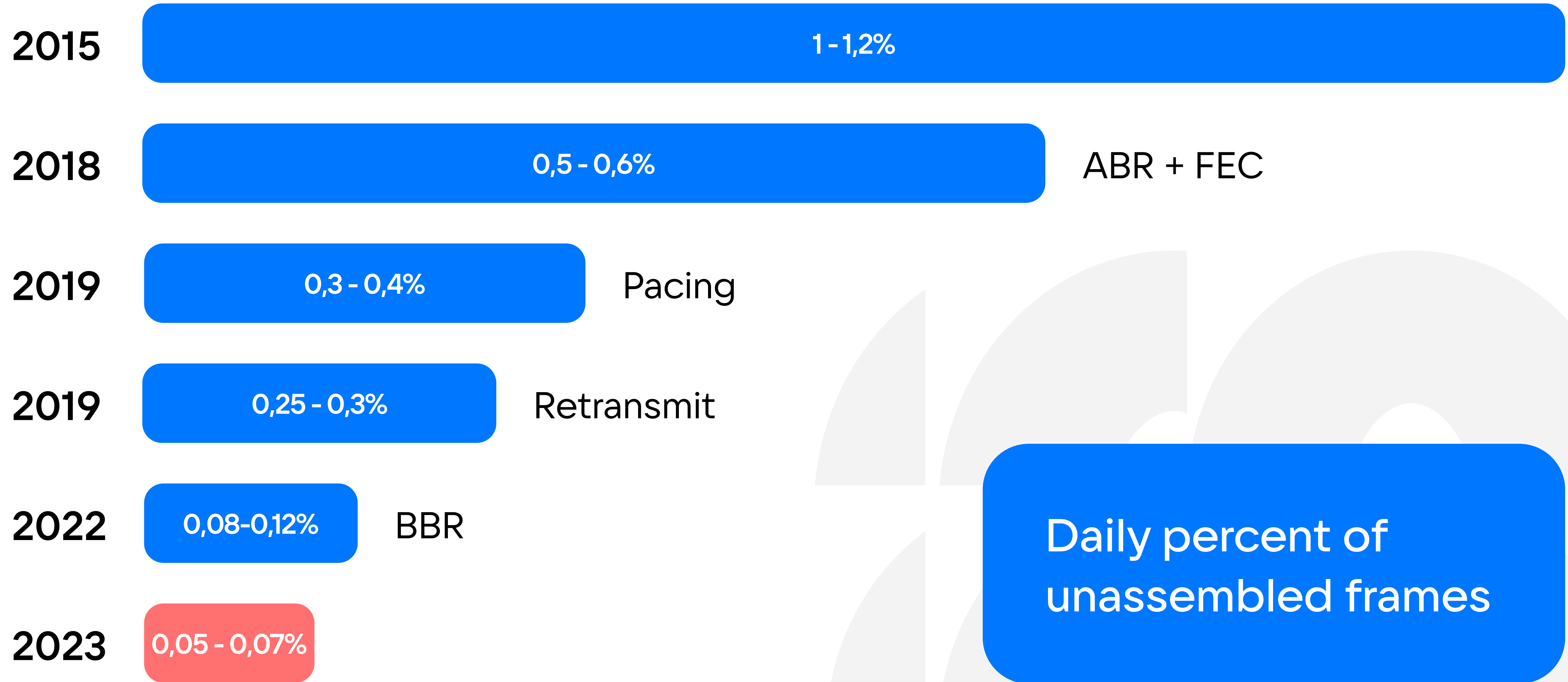
Average network and video bitrates



Recovery percent of lost packets

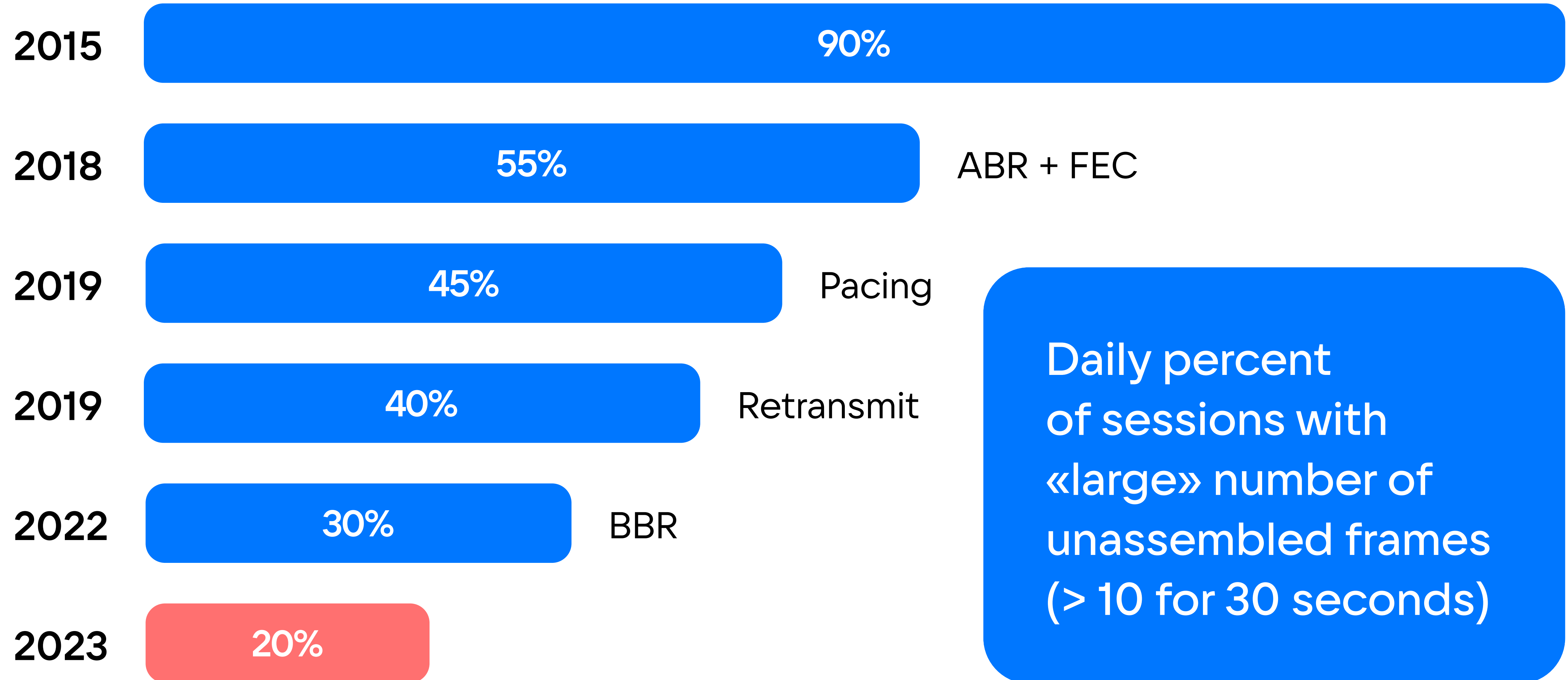


Main key metric



Daily percent of unassembled frames

Second key metric



Daily percent of sessions with «large» number of unassembled frames (> 10 for 30 seconds)

Suggestions for improvement

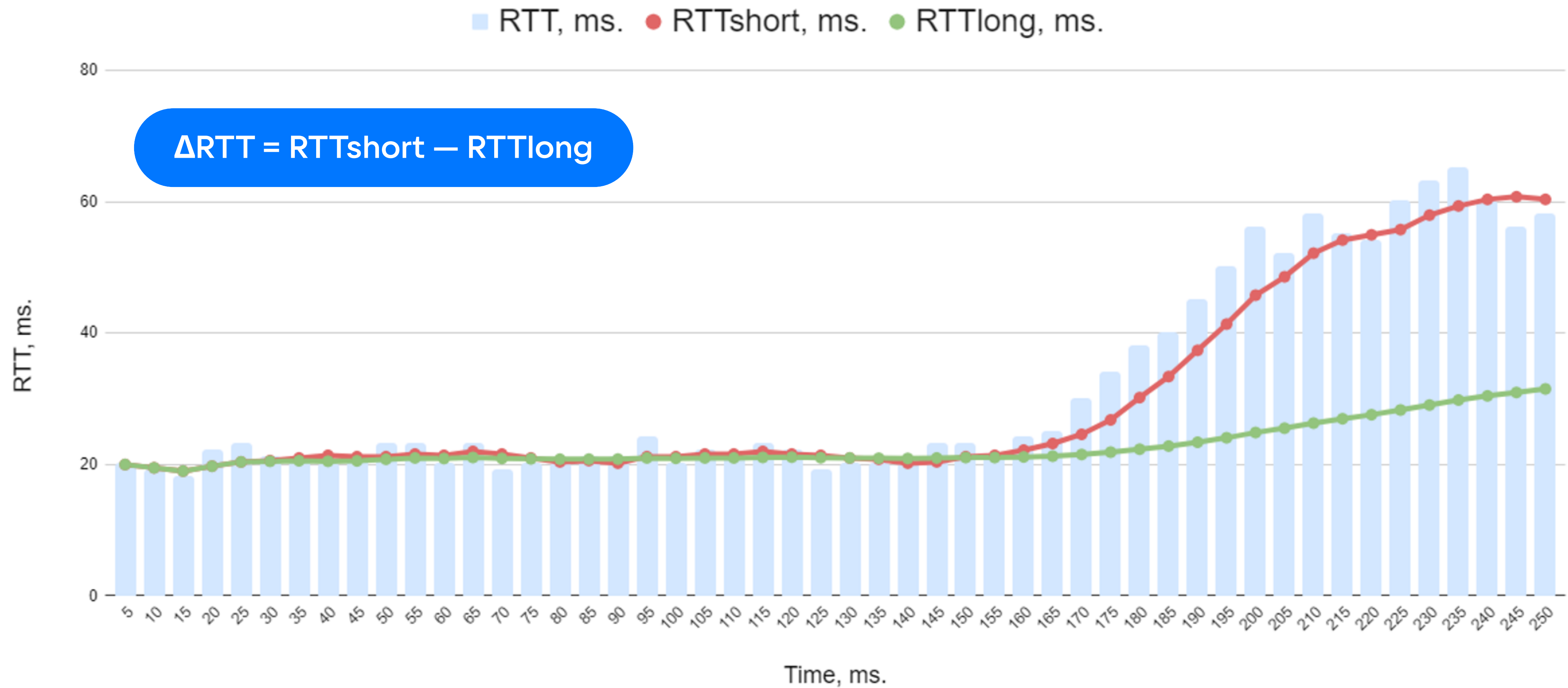
01

Problem situation
memory mechanism
(BBR + ABR)

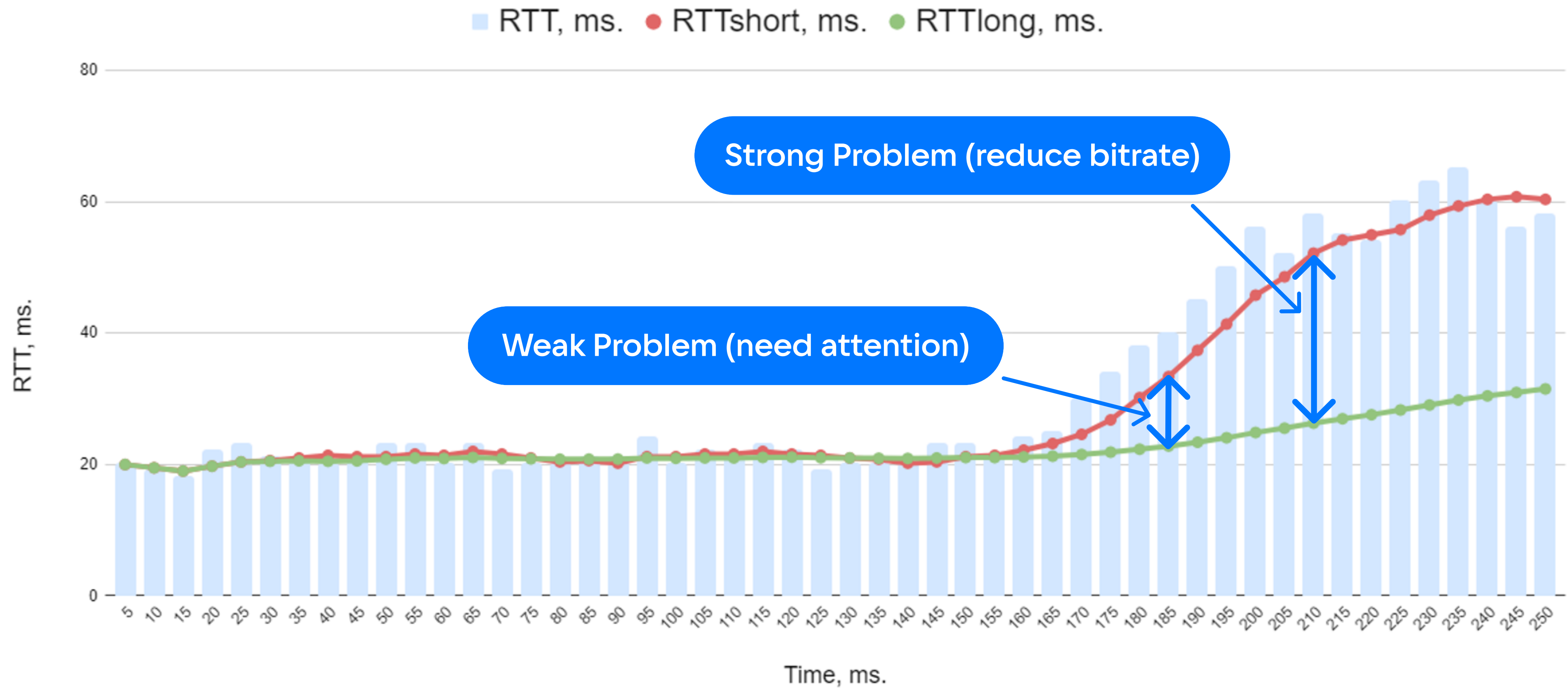
02

Problem ranking
(BBR)

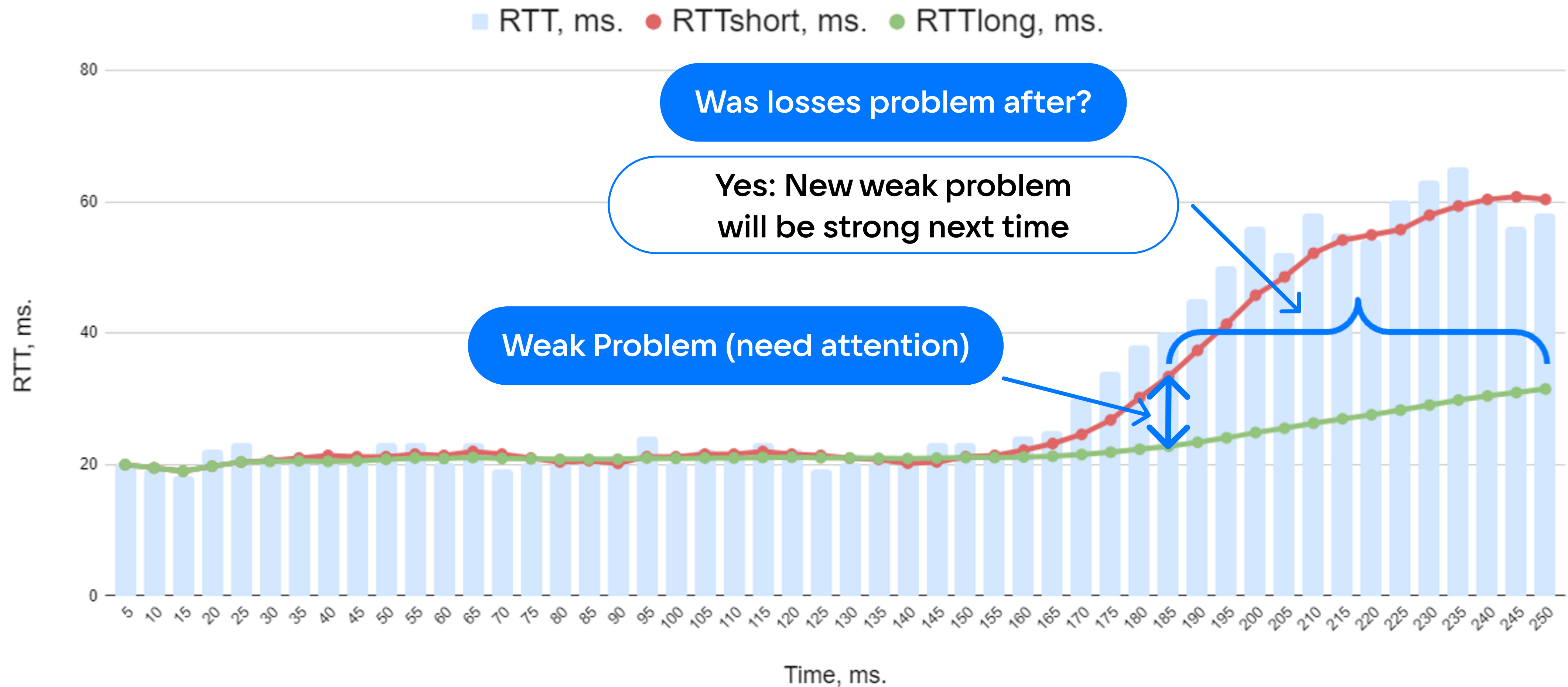
BBR. Network buffering



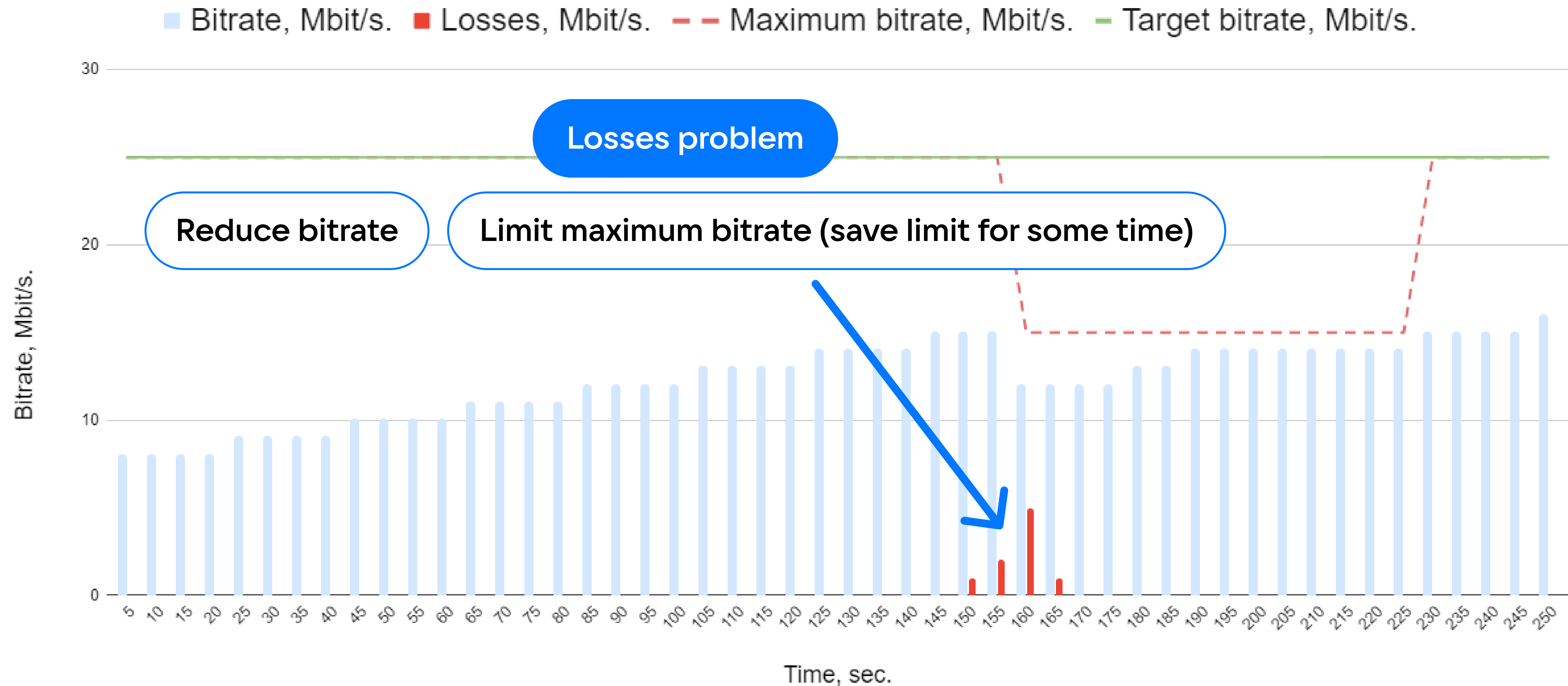
BBR. Problem ranking



BBR. Memory mechanism



ABR. Memory mechanism



Results of improvements

After release:

- Unassembled frames = **~0,06-0,07%** (from 0,08%-0,12%)
- Sessions with «large» number of unassembled frames = **~26-28%** (from 30%)

3 weeks after release:

- Unassembled frames = **~0,08-0,1%**
- -Sessions with «large» number of unassembled frames = **~32-34%**

WHY?

Why?

01

Errors in
calculations and
implementation

OK

02

Users behavior
has changed

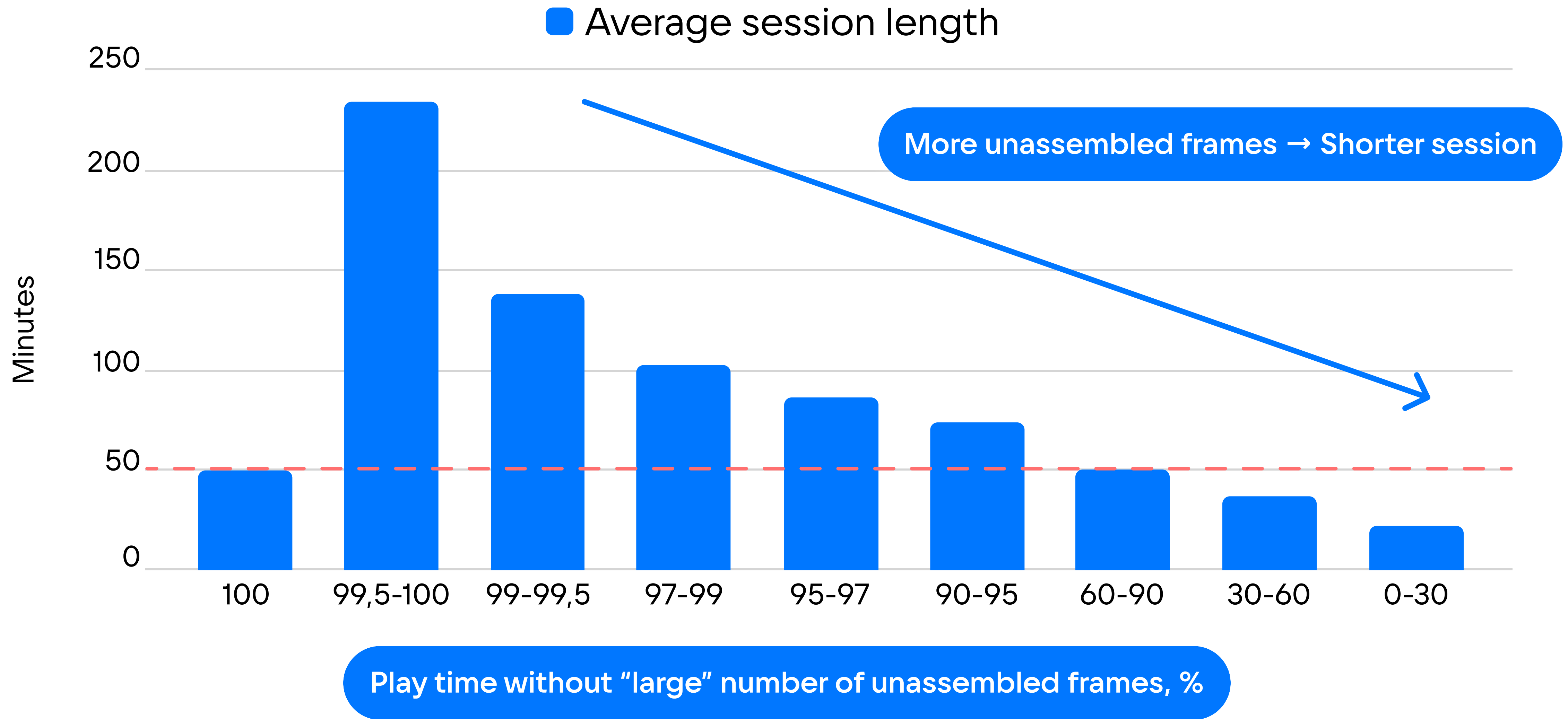
INVESTIGATE

03

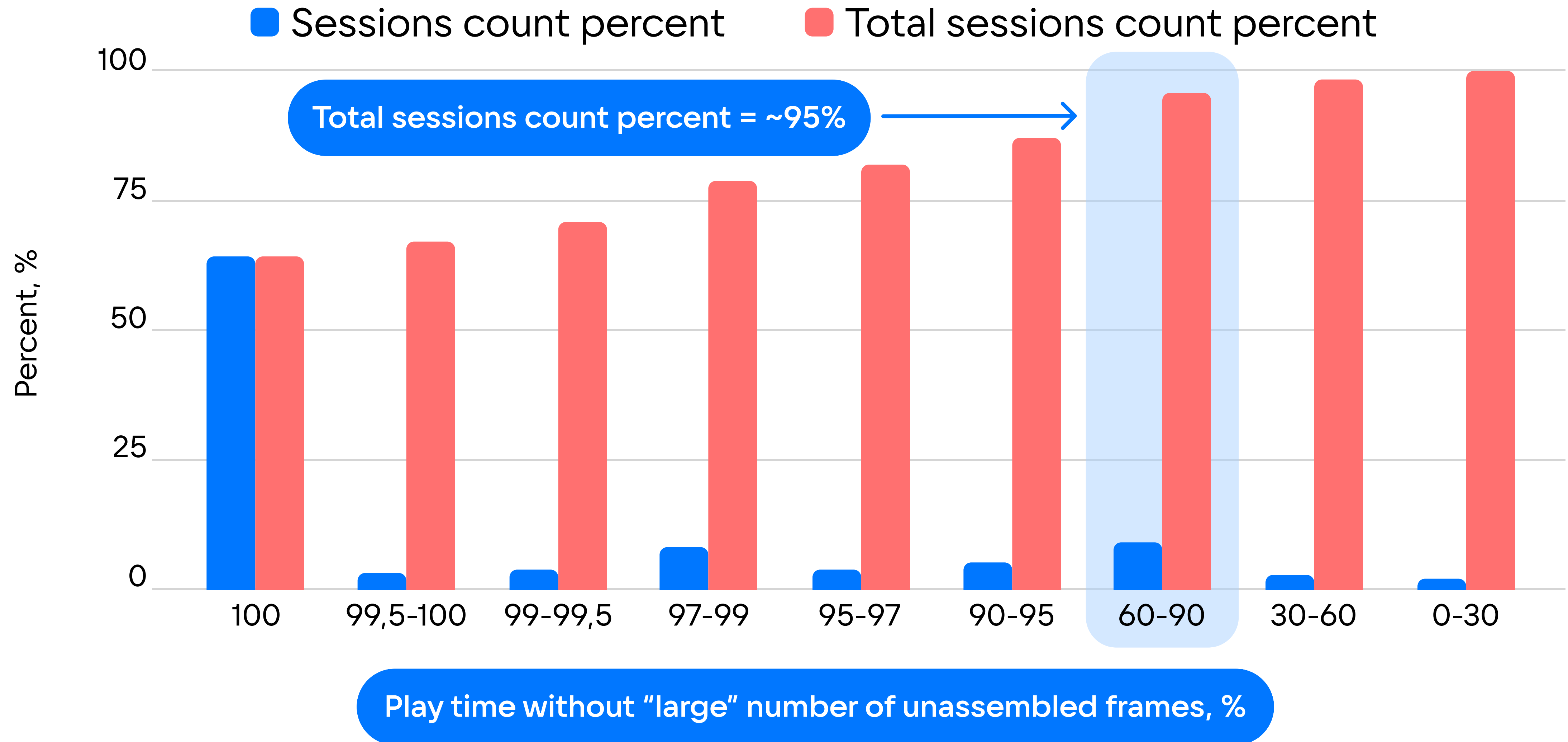
Our metrics
doesn't work
correctly

INVESTIGATE

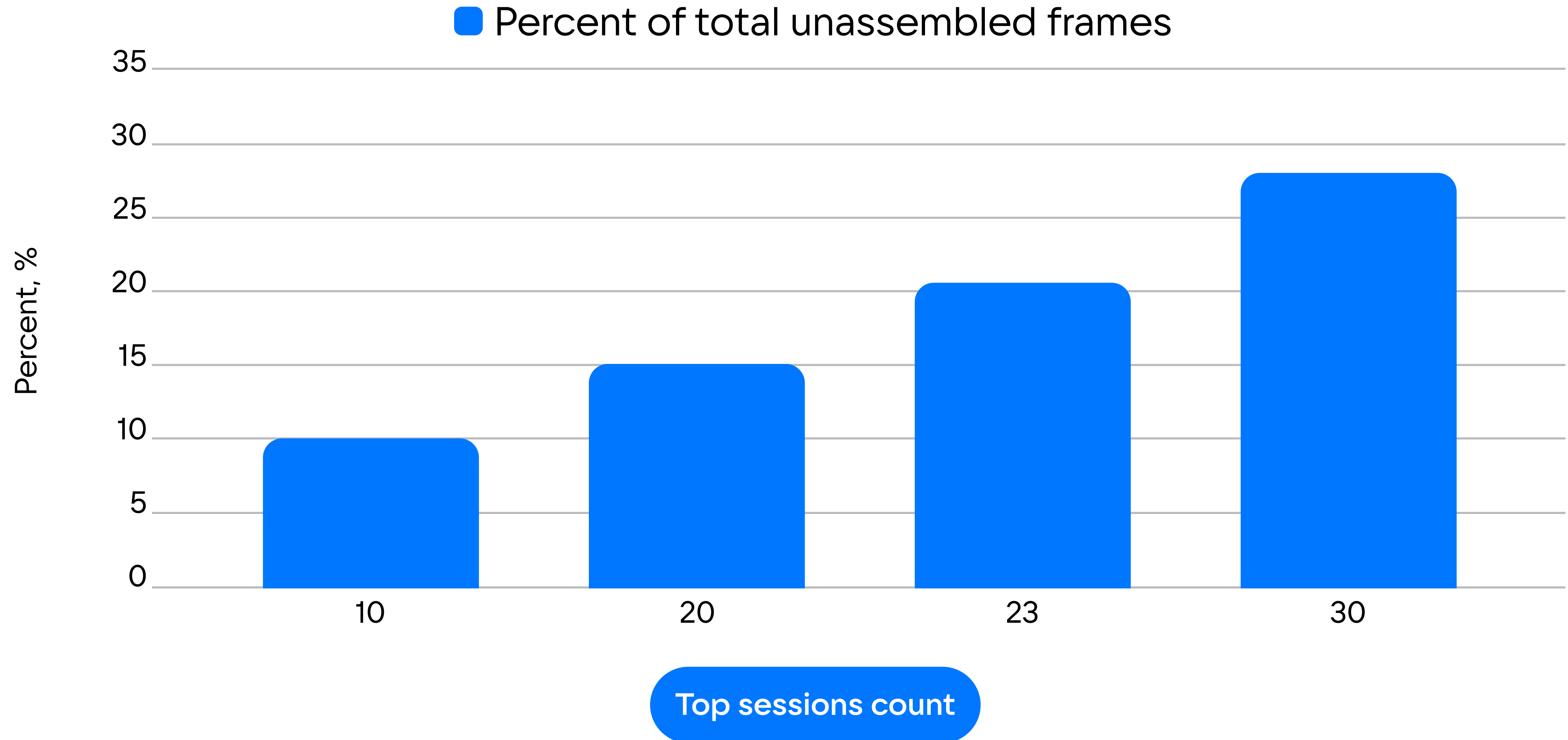
Play time in sessions



Play time in sessions

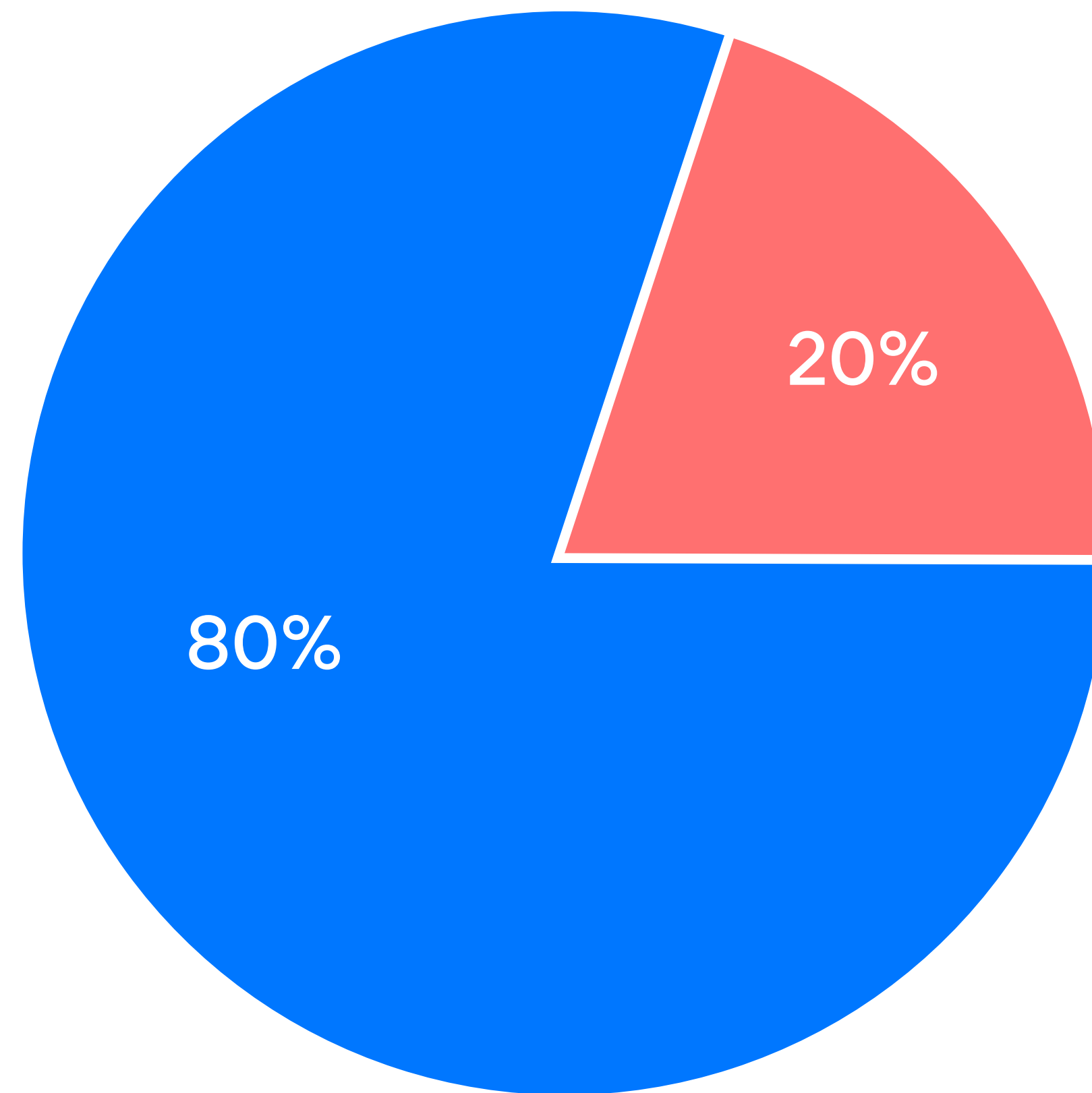


Users behaviour



Top 20 sessions

Unassembled frames reasons



Very low and not uniform throughput

Manual bitrate setting
(set too high bitrate)

Answers for «Why?»

01

Our main and second key metrics need to be corrected

02

User behavior has really changed

03

The proposed algorithms work, but they do not take into account the subjective perception of the user

Metrics in NIC context

	Native	Wi-Fi	Cellular
Unassembled frames	~0,06-0,08%	~0,08-0,1%	~0,4-0,45%
Recovery percentage of lost packets	~80-85%	~80-85%	~45-50%
Average RTT	~15 ms.	~23-24 ms.	~90 ms.
Average network and video bitrates	~22,5 and 16 Mbit/s.	~17 and 9,5 Mbit/s.	~4 and 2,5 Mbit/s.

WebRTC (from box) vs Native solution

	Native	WebRTC
Unassembled frames	~0,08-0,1%	~1-2%
Average RTT	~20 ms.	~40-45 ms.
Average network bitrate	~20 Mbit/s.	~16,5 Mbit/s.

Conclusions

01

ABR should take into account the subjective perception of the user (perception factor of «broken» stream)

02

Working via cellular requires adaptation of ABR

03

WebRTC (from box) not suitable for ultra-low latency applications



VK Play

Thanks
for your attention!



Chernikov Kirill

Head of the VK Play Cloud
Client Development Team
k.chernikov@corp.mail.ru

