

Jitter buffer and frames collection in stream published

Usually the frames received from a publishing client in media stream are big enough and may be fragmented to a number of network packets. There is a special jitter buffer to collect such frames and pass them to the server engine to play, record, HLS cut etc.

Since WCS build [5.2.1395](#) a more strict jitter buffer implementation was added. The strict jitter buffer awaits for key frame receiving from a publisher and does not pass an incomplete frames to the server engine. It may look like video freeze until key frame is received and collected if stream publisher channel is bad, and buffer capacity is not enough to collect the frame from all its parts which are not received in time due to packet losses. The strict jitter buffer can be enabled by the following parameter

```
use_strict_jitter_buffer=true
```

Jitter buffer capacity must be set. A minimal recommended capacity to smoothly play WebRTC streams published with resolution up to 1080p and bitrate up to 2000 kbps using channel with 2% packet loss is 15:

```
jitter_buffer_capacity=15
```

When using a higher publishing bitrate, or if packet losses are huge, it is recommended to increase the capacity, for example, to 30

```
jitter_buffer_capacity=30
```

Since WCS build [5.2.1988](#) the following parameter is available

```
jitter_buffer_strictness=DEFAULT
```

The parameter accepts the following values:

- **DEFAULT** - strict jitter buffer is disabled
- **TOLERANT** - the strict jitter buffer awaits for key frame receiving from a publisher, with a soft key frame detection
- **STRICT** - the strict jitter buffer awaits for key frame receiving from a publisher, with a strict key frame detection (as like as `use_strict_jitter_buffer=true`)