Bitrate management when capturing WebRTC stream in browser

Overview

For optimal pucture quality with available channel bandwith video bitrate should be managed while capturing WebRTC stream in browser. WCS server allows to limit minimum and maximum video bitrate for stream published. Audio bitrate is not managed.

Platforms and browsers supported

	Chrome	Firefox	Safari 11	Edge
Windows	+	+		+
Mac OS	+	+	+	
Android	+	+		
iOS	-	-	+	

Settings

The following WCS settings are intended to limit publishing bitrate:

	On browser side (JavaScript)	On server side (flashphoner.properties)
Minimum bitrate limit	`constraints.video.minBitrate`	`webrtc_cc_min_bitrate`
Maximum bitrate limit	`constraints.video.maxBitrate`	`webrtc_cc_max_bitrate`

On browser side, bitrate limits are set in kilobits per second, for example

```
constraints.video.maxBitrate=600
```

and on server side limits are set in bytes per second

```
webrtc_cc_max_bitrate=600000
```

If the limits are set on both sides, browser settings take precedence over server settings.

If browser settings are not defined, server settings are applied,

If none of limits are set, default settings are applied

```
webrtc_cc_min_bitrate=30000
webrtc_cc_max_bitrate=10000000
```

These settings work in most modern browsers and define the REMBbitrate management limits.

How it works

If maxBitrate is set, WCS server will send REMB command to decrease bitrate when this limit is reached.
If minBitrate is set, WCS server will stop sending REMB command to decrease bitrate when this limit is reached.
Therefore, the settings define 3 ranges with its own bitrate management algorithm:

№	Range	Algorithm
1	[0, minBitrate]	WCS stops bitrate management and not send any REMB commands
2	[minBitrate, maxBitrate]	WCS server makes active bitrate management: depending on jitter and incoming traffic uniformity WCS decides to send REMB commands to decrease bitrate. If channel is good, WCS does nothing and bitrate is not decreasing
3	[maxBitrate, ...]	WCS constantly sends bitrate decrease commands until it reduce to maxBitrate

How to enforce bitrate increasing

Now, bitrate increasing can be enforced in Chrome browser only by setting`x-google-max-bitrate`and`x-google-min-bitrate`parameters via SDP hook.
It is impossible to enforce bitrate rising with client and server settings, only bitrate decreasing can be managed.
In this case, Chrome specific settings take precedence if they are set, i.e. `constraints`and server settings will be ignored. Note that Chrome default settings found by experience is

```
x-google-max-bitrate=2500
```

Usage

Bitrate limiting in certain borders can be useful for example when publishing video to Safari browser subscribers. This browser is sensitive to bitrate jumps, in this case picture quality loses until freeze and browser hangs. It is recommended to stabilize bitrate when publishing streams for Safari viewers by setting narrow limits of bitrate change, for example

```
constraints.video.minBitrate=600
constraints.video.maxBitrate=600
```

In this case picture quality in Safari browser will be acceptable depending on channel bandwith and state.

Bitrate rising needs to be enforced when publishsing HD and 4K streams. In this case, Chrome browser is recommended for publishing.