

# Тестирование пропускной способности канала публикации

Начиная со сборки WCS [5.2.1972](#) и сборки SFU SDK [2.0.273](#) добавлена возможность тестирования пропускной способности канала публикации от клиента до сервера. Тестирование производится с использованием WebRTC data channels, без публикации медиапотока.

При входе в SFU комнату устанавливается WebRTC соединение с сервером

```
await state.room.join(state.pc, null, {});
```

После этого становится доступен объект `BitrateTest`

```
const bitrateTest = state.room.getBitrateTest();
```

Метод `BitrateTest.test()` запускает тестирование канала публикации. Метод принимает на вход максимальную длительность теста в миллисекундах, и возвращает Promise, который разрешается по окончании теста. Тест отправляет пакеты по WebRTC data channel, получает от сервера отчеты о том, какой объем данных получен успешно, и вычисляет битрейт передачи данных. Функция, вызываемая при разрешении Promise после окончания теста, получает на вход последнее измеренное значение битрейта в кбит/с

```
bitrateTest.test($("#" +  
bitrateTestState.durationId()).val()).then((bitrateKbps) => {  
  stateSelector.text("Test is finished, last measured bitrate: " +  
  bitrateKbps + " kbps");  
  ...  
});
```

Объект `BitrateTest` позволяет прослушивать событие `onStatusUpdate`, обработчик этого события получает на вход текущее измеренное значение битрейта в кбит/с. Это позволяет отобразить прогресс измерения на странице

```
bitrateTest.setListener({  
  onStatusUpdate(bitrateKbps) {  
    stateSelector.text("Current bitrate: " + bitrateKbps + " kbps");  
  }  
});
```

Полный код функции примера SFU Bitrate Test, запускающей тестирование канала

## code

```
const startBitrateTest = async function (state) {
  if (state.room) {
    await state.room.join(state.pc, null, {});
    const stateSelector = $("##" + state.currentStateId());
    stateSelector.attr("style", "display:inline-block;margin-left:
10px");
    try {
      const bitrateTest = state.room.getBitrateTest();
      state.setBitrateController(bitrateTest);
      bitrateTest.setListener({
        onStatusUpdate(bitrateKbps) {
          stateSelector.text("Current bitrate: " + bitrateKbps + "
kbps");
        }
      });
      bitrateTest.test($("##" +
bitrateTestState.durationId()).val()).then((bitrateKbps) => {
        stateSelector.text("Test is finished, last measured bitrate:
" + bitrateKbps + " kbps");
        state.setBitrateController(null);
        onStopClick(state);
      });
    } catch (e) {
      if (e.type === constants.SFU_ROOM_EVENT.OPERATION_FAILED) {
        onOperationFailed(state, e);
      } else {
        console.error("Failed to start bitrate test: " + e);
        setStatus(state.errInfoId(), e.name, "red");
        onStopClick(state);
      }
    }
  }
}
```