6 SRS 4.0 WebRTC to RTMP原理分析

零声学院：音视频高级课程：https://ke.qq.com/course/468797?tuin=137bb271

当前srs版本为srs.4.0.123。

注意和3.x版本函数命名的区别

srs3.x	srs.4.0.123
SrsSource	SrsLiveSource （4.0.112）
SrsRtcStream	SrsRtcSource （4.0.113）
SrsConsumer	SrsLiveConsumer （4.0.114）

把SrsSource改成SrsLiveSource，
把SrsRtcStream改成SrsRtcSource，

把SrsConsumer改成SrsLiveConsumer，
保持SrsRtcConsumer不变。

5 WebRTC2Rtmp测试环境

5.1 启动srs服务器

使用**rtc2rtmp**.conf

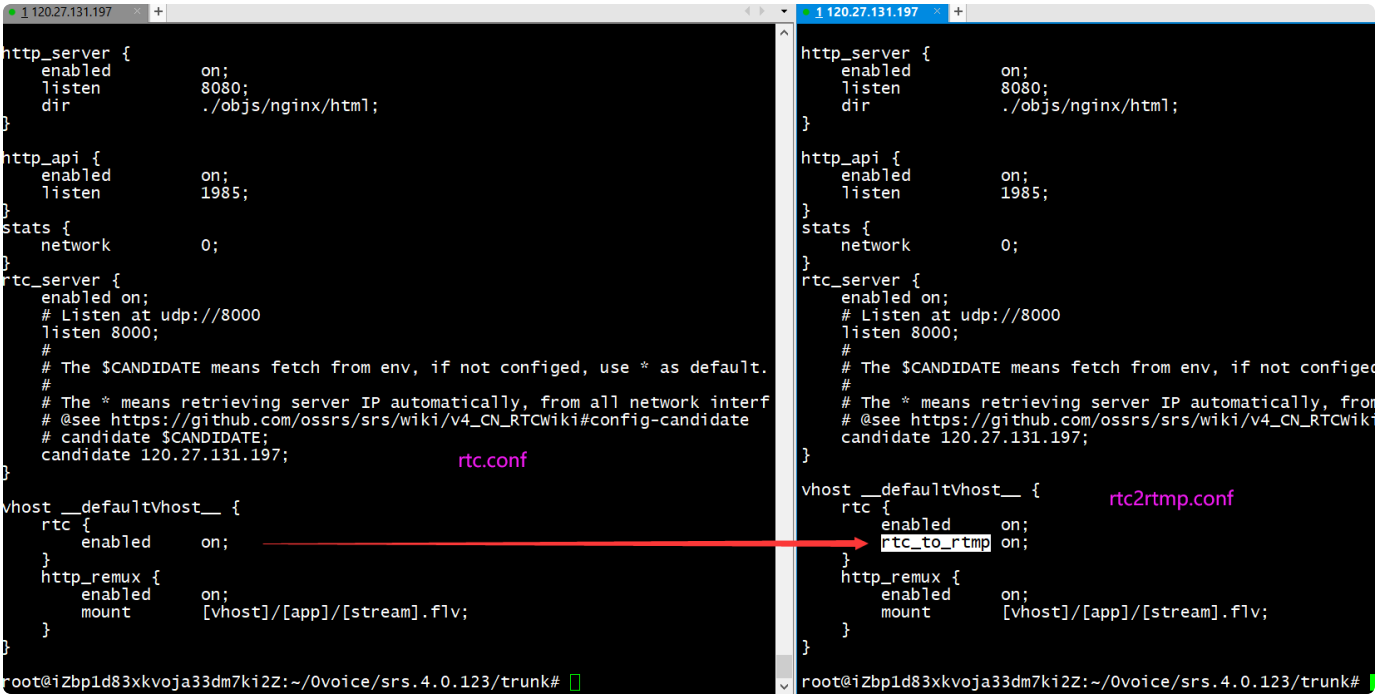
启动服务器

```shell
git clone https://gitee.com/ossrs/srs.git srs.4.0.123
cd srs.4.0.123
git checkout v4.0.123
cd trunk
./configure --gb28181=on  && make
./objs/srs -c conf/rtc2rtmp.conf
```

特别需要注意：**./objs/srs -c conf/rtc2rtmp.conf**

记得修改

```
rtc_server {    enabled on;
    # Listen at udp://8000
    listen 8000;
    #
    # The $CANDIDATE means fetch from env, if not configed, use * as default.
    #
    # The * means retrieving server IP automatically, from all network interfaces,
    # @see https://github.com/ossrs/srs/wiki/v4_CN_RTCWiki#config-candidate
    # candidate $CANDIDATE; 在云服务器为外网ip
    candidate 120.27.131.197;
}
```



5.2 WebRTC推流

http://120.27.131.197:8080/players/rtc_publisher.html

因为我们现在使用使用ip地址进行测试，没有使用https+域名的方式，所以在使用WebRTC时需要修改Chrome的启动参数。

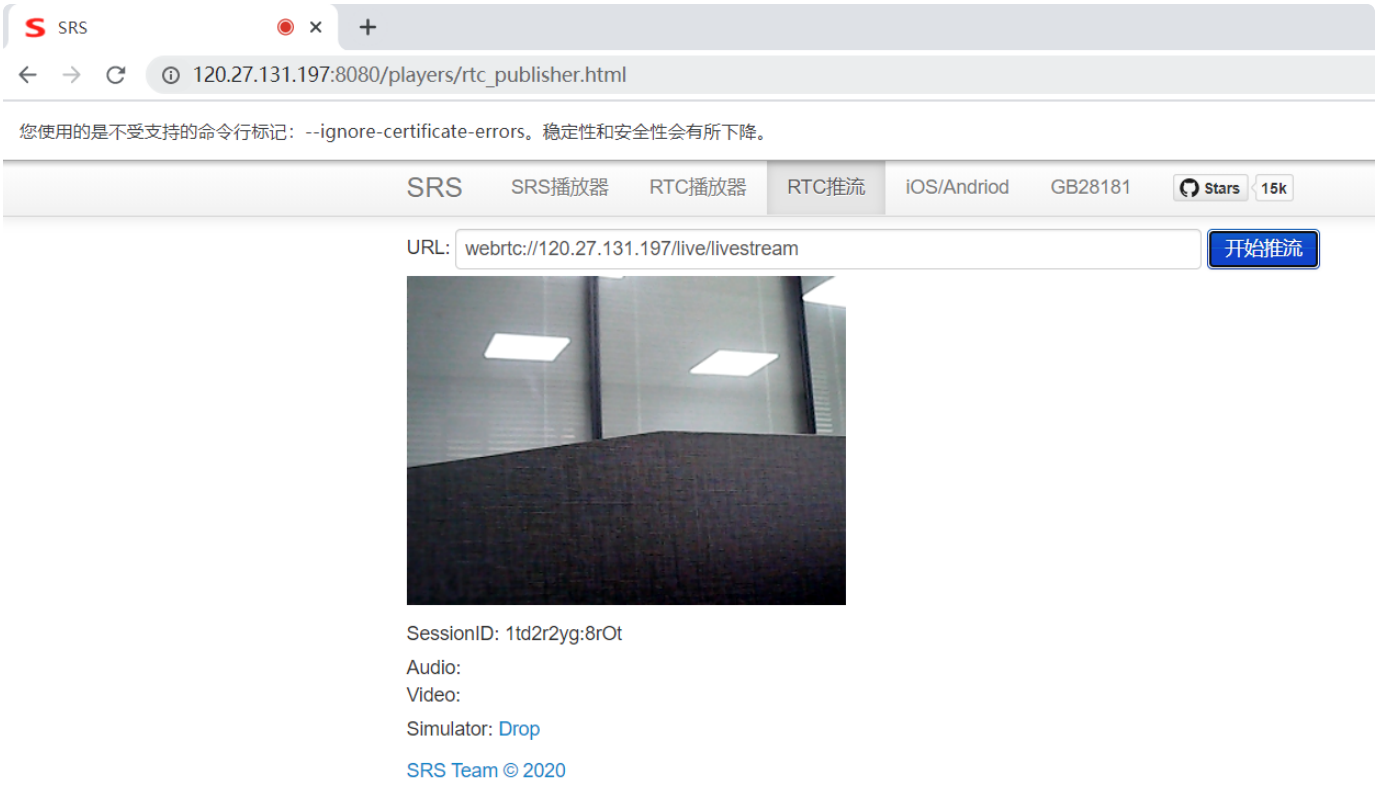
在使用Chrome浏览器推流时会报：TypeError: Cannot read property 'getUserMedia' of undefined 错误，这个错误主要是https证书问题。

解决办法：先把chrome完全退出，右击桌面上chrome的快捷键，点击属性，在目标一栏添加如下内容，记着有个英文空格，IP换成自己的。
在目标加上以下参数（IP地址换成自己的）：

```
--ignore-certificate-errors --allow-running-insecure-content  --unsafely-treat-insecure-origin-as-secure="http://120.27.131.197:8080"
```

然后重新打开chrome，输入自己的地址。

http://120.27.131.197:8080/players/rtc_publisher.html



5.3 客户端使用ffmpeg进行rtmp拉流

比如：

```shell
ffplay rtmp://120.27.131.197/live/livestream
```

6 WebRTC2Rtmp协议转换

WebRTC转RTMP：

- 编码，涉及到Opus转AAC
- 协议，RTP转RTMP封装

SrsRtmpFromRtcBridger 将rtc source转成rtmp，什么时候会被调用。

SrsLiveSource::SrsLiveSource

视频包分发：

- srs_error_t SrsRtmpFromRtcBridger::packet_video_rtmp(const uint16_t start, const uint16_t end)
- srs_error_t SrsRtmpFromRtcBridger::packet_video_key_frame(SrsRtpPacket* pkt)

音频转码：

- srs_error_t SrsAudioTranscoder::transcode(SrsAudioFrame *in_pkt, std::vector<SrsAudioFrame*>& out_pkts)

音频包分发：

- srs_error_t SrsRtmpFromRtcBridger::trancode_audio(SrsRtpPacket *pkt)

SrsRtcPublishStream 发布流
SrsRtcPlayStream

7 WebRTC2Rtmp逻辑分析

架构

看上课演示框图。
https://www.processon.com/view/link/60f047bf1efad40732081590，密码：v3v3

- 什么时候创建**publisher**
- **什么时候创建SrsLiveSource**
- **什么时候创建**SrsRtmpFromRtcBridger

SrsRtcSource -> SrsRtmpFromRtcBridger -> SrsLiveSource

来自publish**请求**

```
srs_error_t SrsRtcServer::listen_api(){
    srs_error_t err = srs_success;

    // TODO: FIXME: Fetch api from hybrid manager, not from SRS.
    SrsHttpServeMux* http_api_mux = _srs_hybrid->srs()->instance()->api_server();// 默认1985的端口

    if ((err = http_api_mux->handle("/rtc/v1/play/", new SrsGoApiRtcPlay(this))) != srs_success) {
        return srs_error_wrap(err, "handle play");
```

```
    }

    if ((err = http_api_mux->handle("/rtc/v1/publish/", new SrsGoApiRtcPublish(this))) !=
srs_success) {
        return srs_error_wrap(err, "handle publish");
    }

#ifdef SRS_SIMULATOR
    if ((err = http_api_mux->handle("/rtc/v1/nack/", new SrsGoApiRtcNACK(this))) !=
srs_success) {
        return srs_error_wrap(err, "handle nack");
    }
#endif

    return err;
}
```

断点

SrsLiveSource::SrsLiveSource何时创建RTMP source

#0 SrsLiveSource::SrsLiveSource (this=0x55555621d100) at src/app/srs_app_source.cpp:1847#1 0x00005555556edabe in SrsLiveSourceManager::fetch_or_create (this=0x5555560af900, r=0x555556210d70, h=0x5555560b4a18,
    pps=0x55555620a358) at src/app/srs_app_source.cpp:1729
#2 0x00005555557dffce in SrsRtcPublishStream::initialize (this=0x555556219c40, r=0x555556210d70,
    stream_desc=0x555556213e00) at src/app/srs_app_rtc_conn.cpp:1078
#3 0x00005555557ef061 in SrsRtcConnection::create_publisher (this=0x555556211bb0, req=0x555556210d70,
    stream_desc=0x555556213e00) at src/app/srs_app_rtc_conn.cpp:3488
#4 0x00005555557e44c5 in SrsRtcConnection::add_publisher (this=0x555556211bb0, ruc=0x55555620ae60, local_sdp=...)
    at src/app/srs_app_rtc_conn.cpp:1880
#5 0x000055555581bf37 in SrsRtcServer::do_create_session (this=0x5555560b50d0, ruc=0x55555620ae60, local_sdp=...,
    session=0x555556211bb0) at src/app/srs_app_rtc_server.cpp:501
#6 0x000055555581bdba in SrsRtcServer::create_session (this=0x5555560b50d0, ruc=0x55555620ae60, local_sdp=...,

psession=0x55555620a7c8) at src/app/srs_app_rtc_server.cpp:483
#7 0x0000555555834af7 in SrsGoApiRtcPublish::do_serve_http (this=0x5555561b7710, w=0x55555620b500, r=0x55555620d280, res=
    0x55555620e0c0) at src/app/srs_app_rtc_api.cpp:413
#8 0x00005555558338ba in SrsGoApiRtcPublish::serve_http (this=0x5555561b7710, w=0x55555620b500, r=0x55555620d280)
    at src/app/srs_app_rtc_api.cpp:287
#9 0x0000555555697a5d in SrsHttpServeMux::serve_http (this=0x5555560b4780, w=0x55555620b500, r=0x55555620d280)
    at src/protocol/srs_http_stack.cpp:730
#10 0x000055555569884e in SrsHttpCorsMux::serve_http (this=0x5555561fa8f0, w=0x55555620b500, r=0x55555620d280)
    at src/protocol/srs_http_stack.cpp:878
#11 0x000055555577b1e4 in SrsHttpConn::process_request (this=0x5555561da6c0, w=0x55555620b500, r=0x55555620d280, rid=1)
    at src/app/srs_app_http_conn.cpp:250
#12 0x000055555577ae29 in SrsHttpConn::process_requests (this=0x5555561da6c0, preq=0x55555620b5d8)
    at src/app/srs_app_http_conn.cpp:223
#13 0x000055555577a9af in SrsHttpConn::do_cycle (this=0x5555561da6c0) at src/app/srs_app_http_conn.cpp:177
#14 0x000055555577a3a4 in SrsHttpConn::cycle (this=0x5555561da6c0) at src/app/srs_app_http_conn.cpp:122
#15 0x000055555571380a in SrsFastCoroutine::cycle (this=0x5555561fa950) at src/app/srs_app_st.cpp:270
#16 0x00005555557138a6 in SrsFastCoroutine::pfn (arg=0x5555561fa950) at src/app/srs_app_st.cpp:285
#17 0x000055555583bc48 in _st_thread_main () at sched.c:363

SrsRtmpFromRtcBridger::SrsRtmpFromRtcBridger 何时创建RTC2RTMP桥接

#0 SrsRtmpFromRtcBridger::SrsRtmpFromRtcBridger (this=0x55555621e3c0, src=0x55555621d100)   at src/app/srs_app_rtc_source.cpp:1256
#1 0x00005555557e00f8 in SrsRtcPublishStream::initialize (this=0x555556219c40, r=0x555556210d70,
    stream_desc=0x555556213e00) at src/app/srs_app_rtc_conn.cpp:1091

#2 0x00005555557ef061 in SrsRtcConnection::create_publisher (this=0x555556211bb0,
req=0x555556210d70,
    stream_desc=0x555556213e00) at src/app/srs_app_rtc_conn.cpp:3488
#3 0x00005555557e44c5 in SrsRtcConnection::add_publisher (this=0x555556211bb0,
ruc=0x55555620ae60, local_sdp=...)
    at src/app/srs_app_rtc_conn.cpp:1880
#4 0x000055555581bf37 in SrsRtcServer::do_create_session (this=0x5555560b50d0,
ruc=0x55555620ae60, local_sdp=...,
    session=0x555556211bb0) at src/app/srs_app_rtc_server.cpp:501
#5 0x000055555581bdba in SrsRtcServer::create_session (this=0x5555560b50d0,
ruc=0x55555620ae60, local_sdp=...,
    psession=0x55555620a7c8) at src/app/srs_app_rtc_server.cpp:483
#6 0x0000555555834af7 in SrsGoApiRtcPublish::do_serve_http (this=0x5555561b7710,
w=0x55555620b500, r=0x55555620d280,
    res=0x55555620e0c0) at src/app/srs_app_rtc_api.cpp:413
#7 0x00005555558338ba in SrsGoApiRtcPublish::serve_http (this=0x5555561b7710,
w=0x55555620b500, r=0x55555620d280)
    at src/app/srs_app_rtc_api.cpp:287
#8 0x0000555555697a5d in SrsHttpServeMux::serve_http (this=0x5555560b4780,
w=0x55555620b500, r=0x55555620d280)
    at src/protocol/srs_http_stack.cpp:730
#9 0x000055555569884e in SrsHttpCorsMux::serve_http (this=0x5555561fa8f0,
w=0x55555620b500, r=0x55555620d280)
    at src/protocol/srs_http_stack.cpp:878
#10 0x000055555577b1e4 in SrsHttpConn::process_request (this=0x5555561da6c0,
w=0x55555620b500, r=0x55555620d280, rid=1)
    at src/app/srs_app_http_conn.cpp:250
#11 0x000055555577ae29 in SrsHttpConn::process_requests (this=0x5555561da6c0,
preq=0x55555620b5d8)
    at src/app/srs_app_http_conn.cpp:223
#12 0x000055555577a9af in SrsHttpConn::do_cycle (this=0x5555561da6c0) at
src/app/srs_app_http_conn.cpp:177
#13 0x000055555577a3a4 in SrsHttpConn::cycle (this=0x5555561da6c0) at
src/app/srs_app_http_conn.cpp:122
#14 0x000055555571380a in SrsFastCoroutine::cycle (this=0x5555561fa950) at
src/app/srs_app_st.cpp:270
#15 0x00005555557138a6 in SrsFastCoroutine::pfn (arg=0x5555561fa950) at
src/app/srs_app_st.cpp:285
#16 0x000055555583bc48 in _st_thread_main () at sched.c:363

#17 0x000055555583c4e4 in st_thread_create (start=0x555555713886
<SrsFastCoroutine::pfn(void*)>, arg=0x5555561fa950,
---Type <return> to continue, or q <return> to quit---
    joinable=1, stk_size=65536) at sched.c:694

SrsRtmpFromRtcBridger::trancode_audio

#0 SrsRtmpFromRtcBridger::trancode_audio (this=0x55555621e3c0, pkt=0x555556219450) at src/app/srs_app_rtc_source.cpp:1337#1 0x0000555555824095 in SrsRtmpFromRtcBridger::on_rtp (this=0x55555621e3c0, pkt=0x555556219450)
    at src/app/srs_app_rtc_source.cpp:1322
#2 0x00005555558213d3 in **SrsRtcSource::on_rtp** (this=0x5555560b4bd0, pkt=0x555556219450)
    at src/app/srs_app_rtc_source.cpp:642
#3 0x0000555555829d8d in SrsRtcAudioRecvTrack::on_rtp (this=0x55555621a5d0, source=0x5555560b4bd0, pkt=0x555556219450)
    at src/app/srs_app_rtc_source.cpp:2357
#4 0x00005555557e174e in SrsRtcPublishStream::do_on_rtp_plaintext (this=0x555556219c40,
    pkt=@0x7ffff7fdf6e8: 0x555556219450, buf=0x7ffff7fdf700) at src/app/srs_app_rtc_conn.cpp:1327
#5 0x00005555557e156b in SrsRtcPublishStream::on_rtp_plaintext (this=0x555556219c40,
    plaintext=0x5555561ca570 "\220\357\rF\366\251\303D\330GY", nb_plaintext=102) at src/app/srs_app_rtc_conn.cpp:1300
#6 0x00005555557e1237 in SrsRtcPublishStream::on_rtp (this=0x555556219c40,
    data=0x5555561ca570 "\220\357\rF\366\251\303D\330GY", nb_data=112) at src/app/srs_app_rtc_conn.cpp:1267
#7 0x00005555557e5dd0 in SrsRtcConnection::on_rtp (this=0x555556211bb0,
    data=0x5555561ca570 "\220\357\rF\366\251\303D\330GY", nb_data=112) at src/app/srs_app_rtc_conn.cpp:2148
#8 0x000055555581b76f in SrsRtcServer::on_udp_packet (this=0x5555560b50d0, skt=0x7ffff7fdfb70)
    at src/app/srs_app_rtc_server.cpp:419
#9 0x00005555557b86ac in SrsUdpMuxListener::cycle (this=0x5555560fdec0) at src/app/srs_app_listener.cpp:636
#10 0x000055555571380a in SrsFastCoroutine::cycle (this=0x5555561b7500) at src/app/srs_app_st.cpp:270
#11 0x00005555557138a6 in SrsFastCoroutine::pfn (arg=0x5555561b7500) at src/app/srs_app_st.cpp:285
#12 0x000055555583bc48 in _st_thread_main () at sched.c:363

#13 0x000055555583c4e4 in st_thread_create (start=0x7ffff6e689d8
<__libc_multiple_threads>, arg=0x5b0000006e, joinable=119,
    stk_size=124) at sched.c:694
#14 0x00007ffff6e63c40 in ?? () from /lib/x86_64-linux-gnu/libc.so.6
#15 0x0000000000000000 in ?? ()

SrsRtmpFromRtcBridger::packet_video_key_frame

Breakpoint 5, SrsRtmpFromRtcBridger::packet_video_key_frame (this=0x55555621e3c0,
pkt=0x5555562bc5b0)   at src/app/srs_app_rtc_source.cpp:1441
1441 {
(gdb) bt
#0 SrsRtmpFromRtcBridger::packet_video_key_frame (this=0x55555621e3c0,
pkt=0x5555562bc5b0)
    at src/app/srs_app_rtc_source.cpp:1441
#1 0x0000555555824766 in SrsRtmpFromRtcBridger::packet_video (this=0x55555621e3c0,
src=0x5555562be730)
    at src/app/srs_app_rtc_source.cpp:1410
#2 0x00005555558240ae in SrsRtmpFromRtcBridger::on_rtp (this=0x55555621e3c0,
pkt=0x5555562be730)
    at src/app/srs_app_rtc_source.cpp:1324
#3 0x00005555558213d3 in SrsRtcSource::on_rtp (this=0x5555560b4bd0,
pkt=0x5555562be730)
    at src/app/srs_app_rtc_source.cpp:642
#4 0x000055555582a133 in SrsRtcVideoRecvTrack::on_rtp (this=0x55555621aaf0,
source=0x5555560b4bd0, pkt=0x5555562be730)
    at src/app/srs_app_rtc_source.cpp:2415
#5 0x00005555557e17d3 in SrsRtcPublishStream::do_on_rtp_plaintext (this=0x555556219c40,
    pkt=@0x7ffff7fdf6e8: 0x5555562be730, buf=0x7ffff7fdf700) at
src/app/srs_app_rtc_conn.cpp:1332
#6 0x00005555557e156b in SrsRtcPublishStream::on_rtp_plaintext (this=0x555556219c40,
    plaintext=0x5555561ca570 "\220}\177\221\350\324\305(\314Ce\241\276", <incomplete
sequence \336>, nb_plaintext=43)
    at src/app/srs_app_rtc_conn.cpp:1300
#7 0x00005555557e1237 in SrsRtcPublishStream::on_rtp (this=0x555556219c40,
    data=0x5555561ca570 "\220}\177\221\350\324\305(\314Ce\241\276", <incomplete
sequence \336>, nb_data=53)
    at src/app/srs_app_rtc_conn.cpp:1267
#8 0x00005555557e5dd0 in SrsRtcConnection::on_rtp (this=0x555556211bb0,

data=0x5555561ca570 "\220}\177\221\350\324\305(\314Ce\241\276", <incomplete sequence \336>, nb_data=53)
at src/app/srs_app_rtc_conn.cpp:2148
#9 0x000055555581b76f in SrsRtcServer::on_udp_packet (this=0x5555560b50d0, skt=0x7ffff7fdfb70)
at src/app/srs_app_rtc_server.cpp:419
#10 0x00005555557b86ac in SrsUdpMuxListener::cycle (this=0x5555560fdec0) at src/app/srs_app_listener.cpp:636
#11 0x000055555571380a in SrsFastCoroutine::cycle (this=0x5555561b7500) at src/app/srs_app_st.cpp:270
#12 0x00005555557138a6 in SrsFastCoroutine::pfn (arg=0x5555561b7500) at src/app/srs_app_st.cpp:285
#13 0x000055555583bc48 in _st_thread_main () at sched.c:363