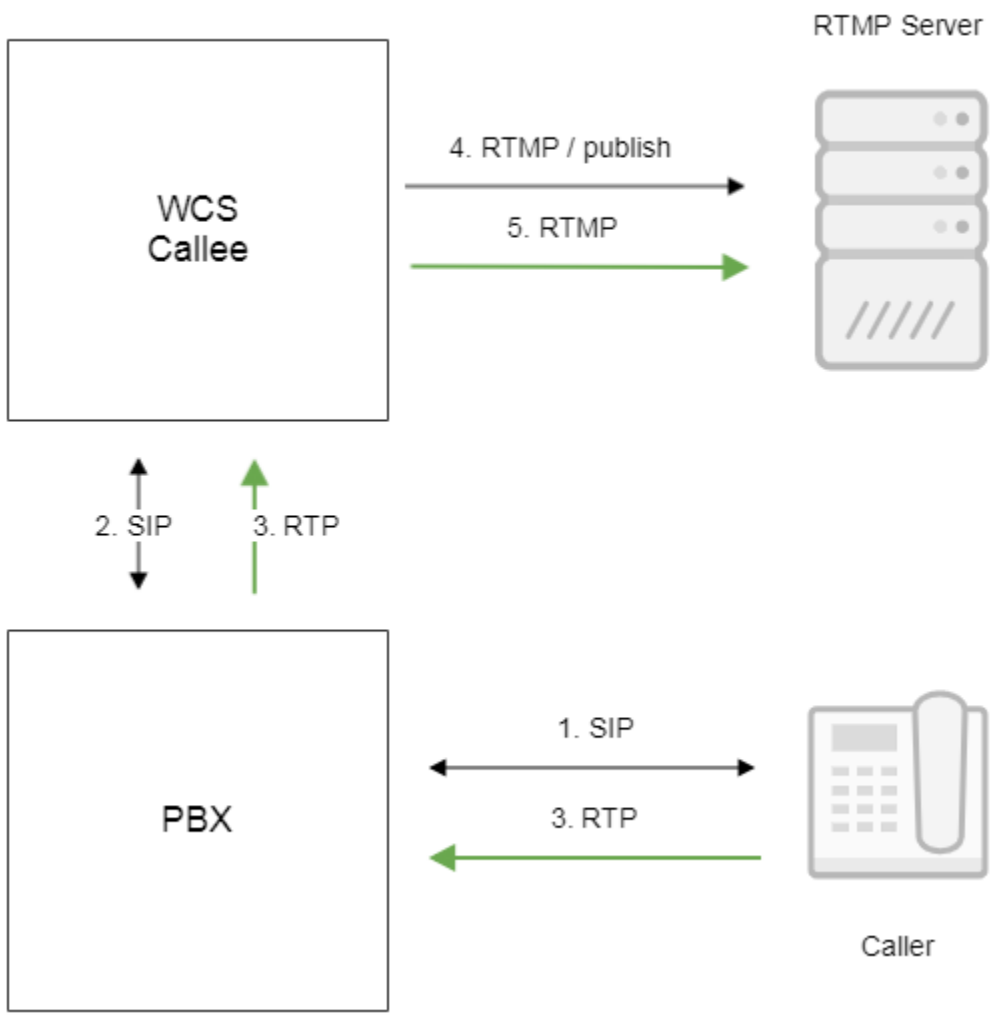
Republishing incoming SIP call to a stream

Overview

WCS 5.2 allows to receive an incoming SIP call from a PBX and republish this call's stream as RTMP to another server (Wowza for example). Also, another stream published on WCS or captured from mp4 file can be injected into the SIP call, so SIP caller can see and hear the injected stream.

To do this, WCS should be configured with SIP trunks as SIP callee. ThenWCS waits for incoming calls from PBX. When SIP call is established, a stream is created from SIP call and republished to target RTMP server.When call is finished, stream is stopped.

Operation flowchart



1. SIP caller makes a call to SIP trunk number on PBX
2. PBX forwards a call to WCS as callee

3. WCS receives media data and creates a stream
4. WCS connects to RTMP server
5. WCS publishes SIP call stream to RTMP server

How to get this working

WCS configuration

On WCS side, the following parameter should be set in flashphoner.properties file

```
sip_add_contact_id=false
```

Also, SIP trunk should be set up in WCS_HOME/conf/sip_trunk.yml file as follows:

```
trunks:
 pbx_t0:
  localPort : 40000
  proxyIp : pbx_address
  remotePort : 5060
  url : rtmp://rtmp_server:1935/live
  visibleName : CUSTOM_NAME
  sdp : |
   v=0
   o=10009 2469 1555 IN IP4 0.0.0.0
   c=IN IP4 0.0.0.0
   t=0 0
   m=audio 7270 RTP/AVP 96
   a=rtpmap:96 opus/48000/2
   a=recvonly
   m=video 9202 RTP/AVP 96
   a=rtpmap:96 H264/90000
   a=fmtp:96 profile-level-id=42801F
   a=recvonly
  sdpParams :
   - b=AS:2000
   - b=RS:50
   - b=RR:100
```

Where

- pbx_t0 – WCS SIP trunk name
- localPort – port to receive incoming SIP calls
- proxyIp – PBX address
- remotePort – PBX port to register
- url – RTMP server URL to republish SIP call stream
- visibleName - name to display to caller, this name is sending to PBX when register
- sdp – SDP to send to PBX in 200 OK response
- sdpParams - parameter to insert to SDP for SIP call media bitrate and channel bandwidth management

WCS supports both TCP and UDP transport for SIP calls, so it listens for incoming both TCP and UDP port defined by localPort parameter.

By default, RTMP stream name will be rtmp_0123456, where 0123456 is callee number. To remove a prefix, set the following parameter in flashphoner.propertiesfile

```
rtmp_transponder_stream_name_prefix=
```

PBX configuration

On PBX side, SIP trunk should be set up to redirect SIP calls to WCS. Calls should be redirected to the port defined in WCS_HOME/conf/sip_trunk.yml file (40000 in example above).

For example, OpenSIPS can be set up to route calls as follows:

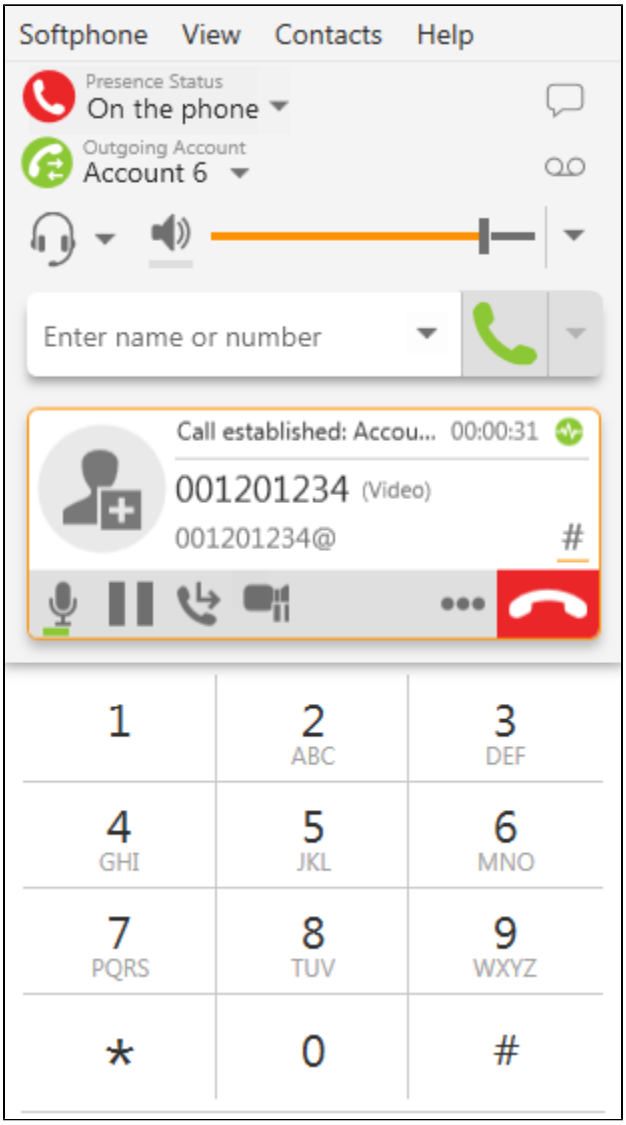
```
route{
        ...
        #WCS Sip trunk routing, 00 prefix + XX for server number (e.g. WCS1 => 01) + X for trunk number
        if ($rU =~ "^00050[0-9]+$") {
            # WCS5 address and port
            rewritehostport("192.168.1.5:40000");
            route(relay);
        }
}
```

Testing

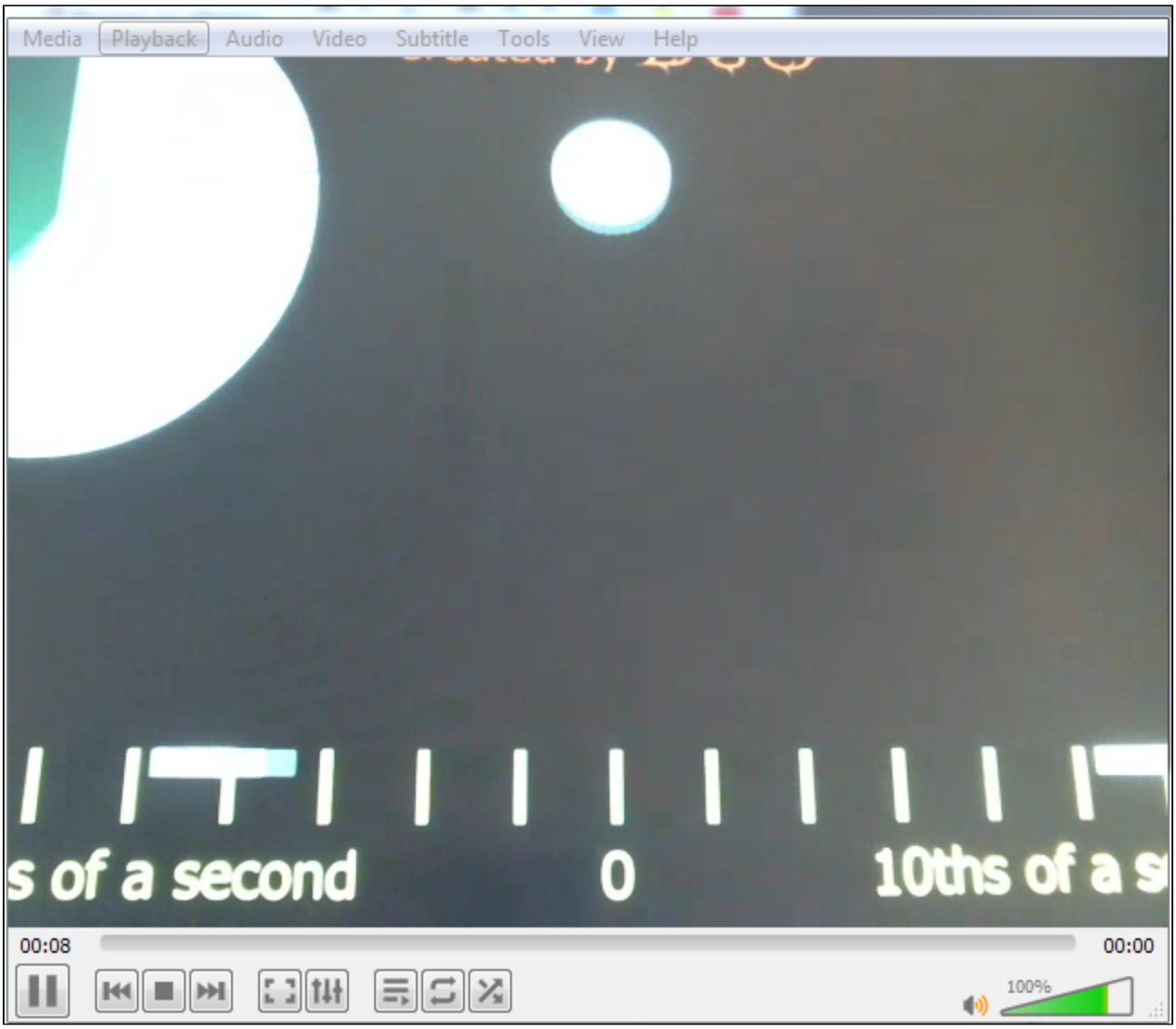
For test we use:

- WCS server
- PBX server
- Softphone to make a call
- RTMP server
- VLC to play RTMP stream

1. Open softphone, connect to PBX server, make a call to a callee number defined in PBX SIP trunk setup, for example 001201234



2. In VLC player open network stream rtmp://rtmp_server:1935/live/rtmp_001201234



3. Send /call/inject_stream/startup query to inject stream from a local file

Method
POST

Request URL
http://p16.flashphoner.com:8081/rest-api/call/inject_stream/startup

SEND

Parameters

Headers | Body | Variables

Body content type
application/json

Editor view
Raw input

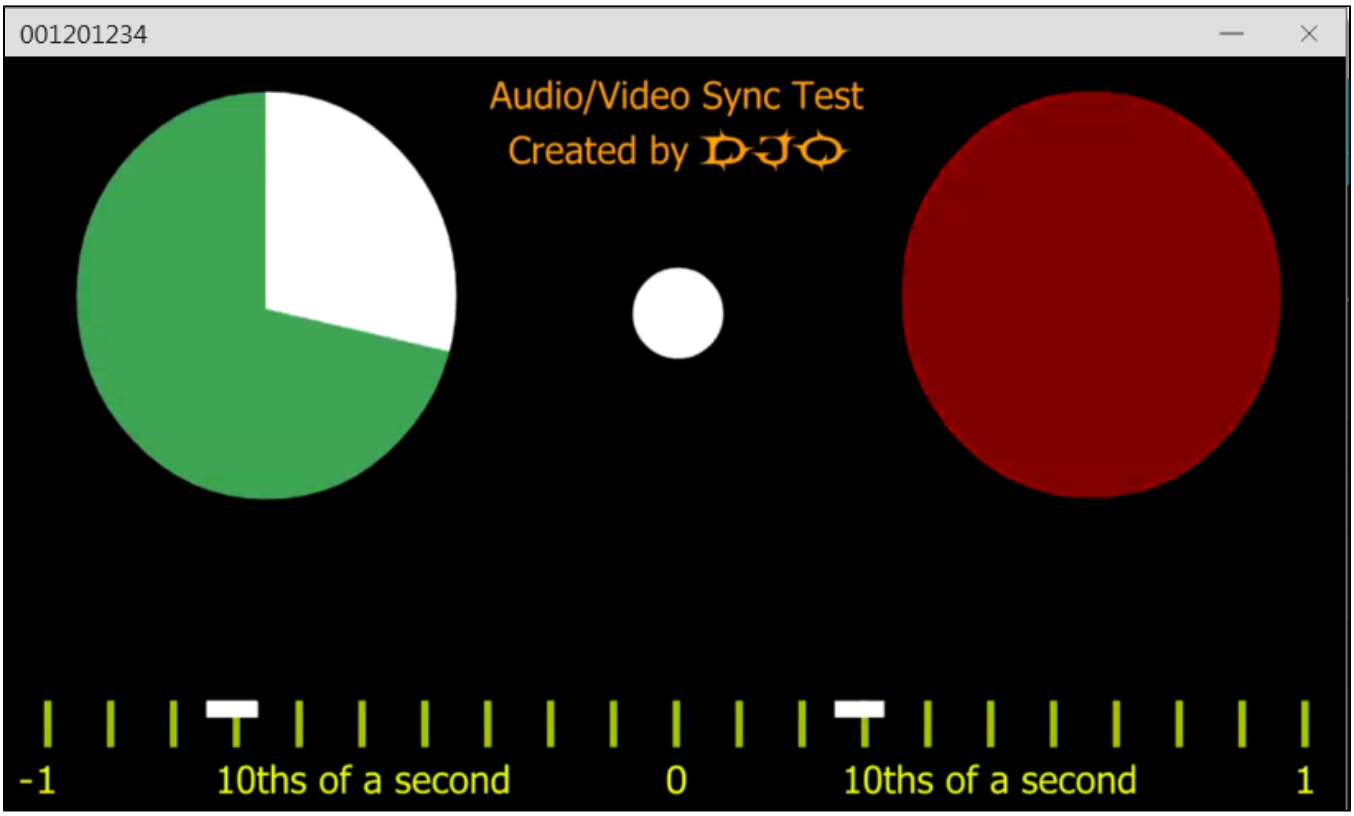
FORMAT JSON MINIFY JSON


```
{
"callId":"YTI2MmYyODg3N2Q2OGZiYWNhZGQ1OTU5NTA3MjdkMzg",
"streamName":"vod-live://test.mp4"
}
```


200 OK 245.96 ms

DETAILS

4. Injected file contents is displayed in softphone video window



5. Send /call/inject_stream/terminate query to stop stream injection

Method
POST
Request URL
http://p16.flashphoner.com:8081/rest-api/call/inject_stream/terminate
SEND

Parameters

Headers | Body | Variables

Body content type
application/json
Editor view
Raw input

FORMAT JSON MINIFY JSON


```
{
"callId":"YTI2MmYyODg3N2Q2OGZiYWNhZGQ1OTU5NTA3MjdkMzg"
}
```


200 OK 212.50 ms
DETAILS

Custom SIP messages listener implementation

In some cases, additional handling of incoming SIP messages is required. To do this, a custom Java class implementing `ISipMessageListener`should be developed to catch incoming SIP messages and handle them as needed.

Look at the example to add port to Request URI of INVITE request if port is not set by callee. Here is the class source code:

customSipMessageListener.java

```
package com.customListener;

import com.flashphoner.sdk.sip.ISipMessageListener;
import com.flashphoner.server.client.IClient;
import gov.nist.javax.sip.address.Authority;
import gov.nist.javax.sip.address.SipUri;
import gov.nist.javax.sip.message.SIPMessage;
import gov.nist.javax.sip.message.SIPRequest;
import gov.nist.javax.sip.stack.MessageChannel;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import javax.sip.message.Request;

public class customSipMessageListener implements ISipMessageListener {

    private static Logger log = LoggerFactory.getLogger("customSipMessageListener");

    @Override
    public void processMessage(SIPMessage sipMessage, IClient client, MessageChannel channel) {
        if (sipMessage instanceof SIPRequest) {
            SIPRequest request = (SIPRequest) sipMessage;
            String method = request.getRequestLine().getMethod();
            if (Request.INVITE.equals(method)) {
                Authority authority = ((SipUri)request.getRequestURI()).getAuthority();
                int port = authority.getPort();
                if (port <= 0) {
                            if (log.isDebugEnabled()) {
                                     log.debug("Inject port " + channel.getPort());
                            }
                    authority.setPort(channel.getPort());
                }
            }
        }
    }
}
```

Make the folder structure in home directory on server

```
mkdir -p com/customListener
```

Copy the class source code to the folder created and compile it

```
javac -cp "/usr/local/FlashphonerWebCallServer/lib/*" ./com/customListener/customSipMessageListener.java
```

Pack the class compiled to the jar file

```
jar cf customSipMessageListener.jar com/customListener/customSipMessageListener.class
```

Copy jar file to server folder

```
cp customSipMessageListener.jar /usr/local/FlashphonerWebCallServer/lib
```

It is necessary to set the class developed to the following parameter inflashphoner.propertiesfile

```
sip_msg_listener=com.customListener.customSipMessageListener
```

then restart server.

Incoming SIP call stream recording

Incoming SIP call streams can be recorded on server. Set the following parameter in flashphoner.properties file to record all the incoming SIP calls:

```
sip_record_stream=true
```

Use REST API query to record a certain SIP call stream.

Note that incoming SIP calls will not be recorded if the following setting is active

```
sip_single_route_only=true
```

Known issues

1. RTMP stream republished from incoming SIP call can be out of sync

Symptoms: SIP call stream is out of sync while playing it from RTMP server

Solution:

a) set the following parameter

```
sip_force_rtcp_feedback=true
```

b) minimize or exclude packet losses in channel between PBX and WCS