



Websocket server of VAST

Version 1.0.0.5

2019/09/20

© 2019 VIVOTEK Inc. All Right Reserved

VIVOTEK may make changes to specifications and product descriptions at any time, without notice.

The following is trademarks of VIVOTEK Inc., and may be used to identify VIVOTEK products only: VIVOTEK. Other product and company names contained herein may be trademarks of their respective owners.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from VIVOTEK Inc.

TABLE OF CONTENTS

TABLE OF CONTENTS	1
1. OVERVIEW.....	2
2. ARCHITECTURE DIAGRAM.....	3
3. RTSP URL FORMAT	4
4. SAMPLE	5
5. WSS CONNECTION.....	7
6. RESTRICTIONS	9

VIVOTEK CONFIDENTIAL
2019.09.20

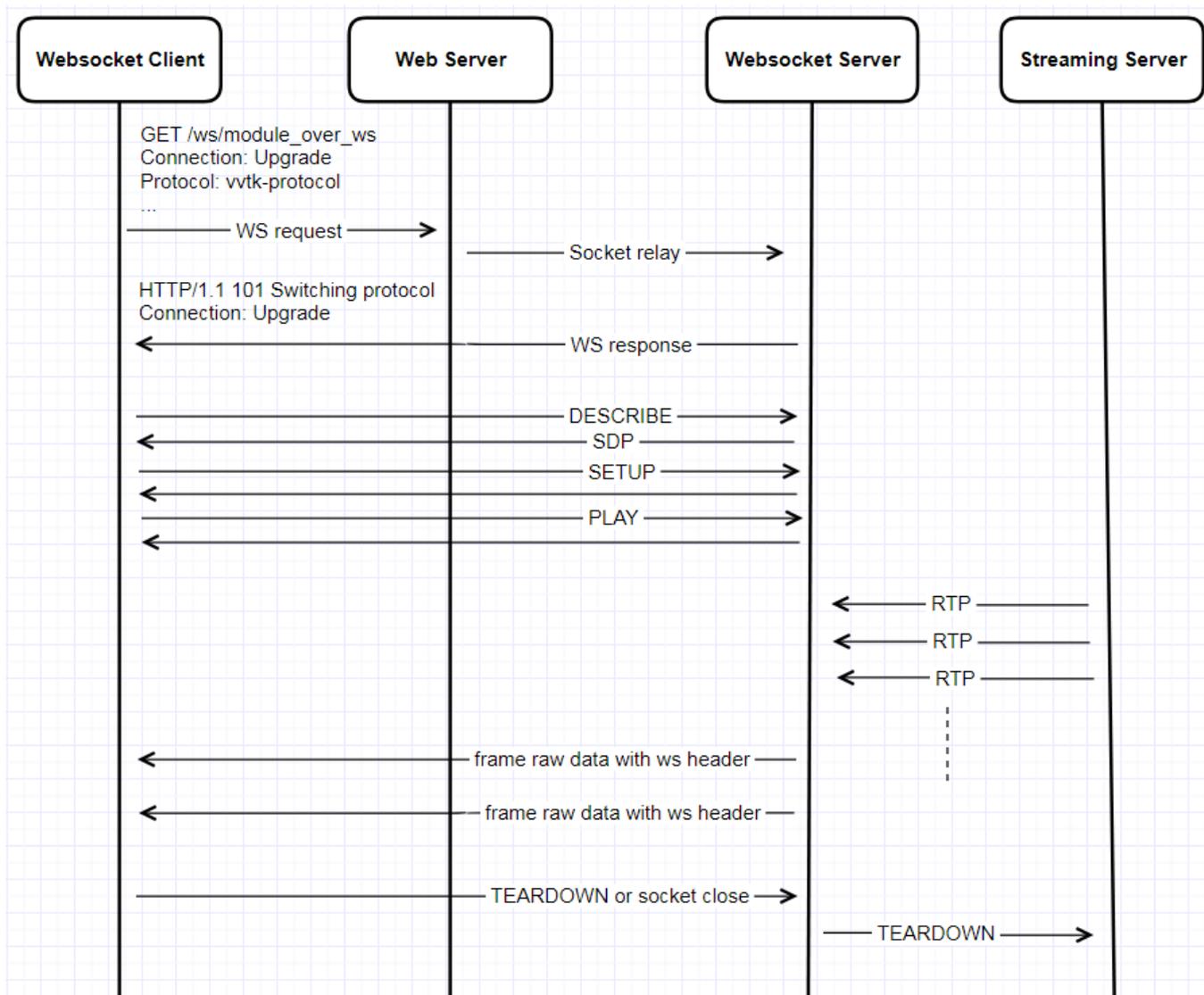
1. Overview

Websocket server can relay streaming from streaming server and doesn't need to create another port to listen connection. Websocket client just needs to send a GET message to http port (VAST default http port is 3454). To specify VIVOTEK format, the GET message must contain "/ws/module_over_ws" and "Protocol: vvtk-protocol".

When sending WS request via GET method, it needs to carry authentication information which is your username and password for the VAST server. After authentication is accomplished, you should send a user-specified RTSP URL on DESCRIBE message to Websocket server. After RTSP handshake completed, client will receive the frame raw data.

VIVOTEK CONFIDENTIAL
2019.09.20

2. Architecture Diagram



3. RTSP URL format

As mentioned previously, user should send a user-specified URL on DESCRIBE message to Websocket server after authentication is accomplished. The RTSP URL format is the same as VAST server.

The URL format is

`ws://VAST_Account:VAST_Password@VAST_Server_IP:VAST_Server_HTTP_Port/Media/Live/Normal?camera=C_x&streamindex=x.`

And here are examples about the format of live and playback.

Live URL is

`ws://username:password@172.18.110.88:3454/Media/Live/Normal?camera=C_1&streamindex=1,`
where C_1 means connection 1, streamindex = 1 means streaming index is 1.

Playback URL is

`ws://username:password@172.18.110.88:3454/Media/Database/Normal?HNAME=C_1&STIME=20190507_100000_000,` where HNAME=C_1 means connection1, STIME=xxx means playback start time(default use UTC time). User can add parameter “&LOCTIME=1” after URL to use local time for streaming. Here is the request URL:

`ws://username:password@172.18.110.88:3454/Media/Database/Normal?HNAME=C_1&STIME=20190507_180000_000 &LOCTIME=1.`

4. Sample

The sample web page in “dist” folder shows how to test Websocket Server.

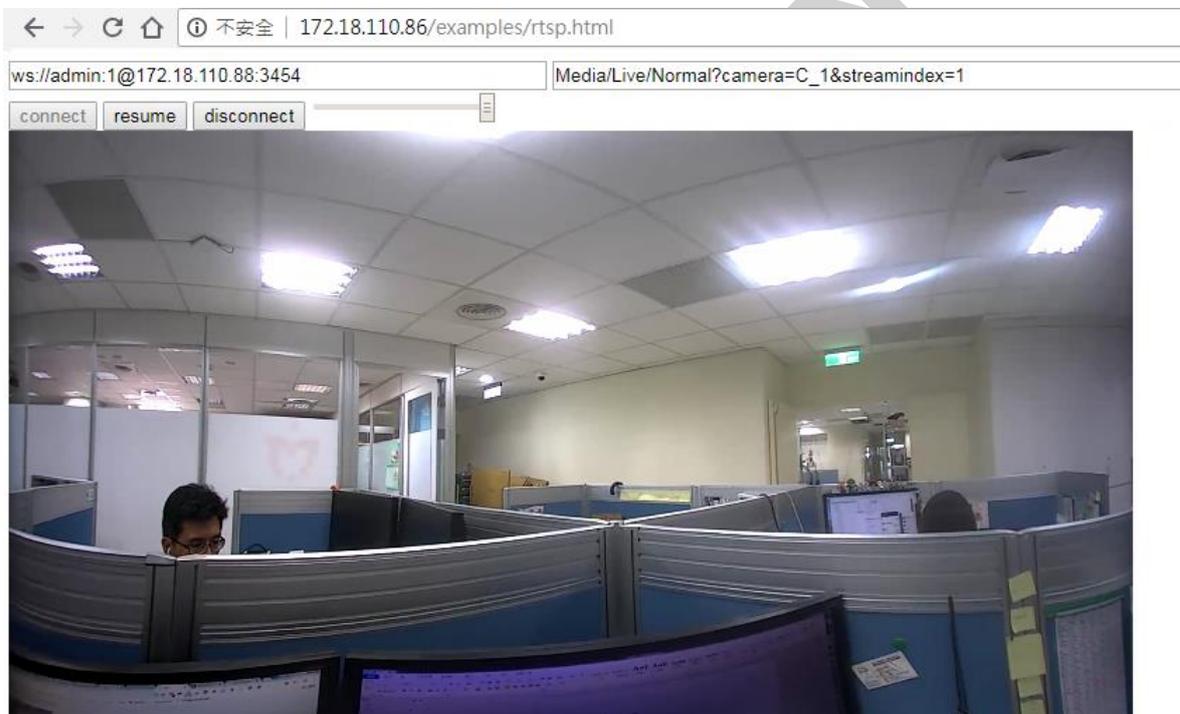
You have to do the following steps to run the sample application.

Step1: Install a web server like Nginx.

Step2. Copy all files under “dist” folder to /www folder

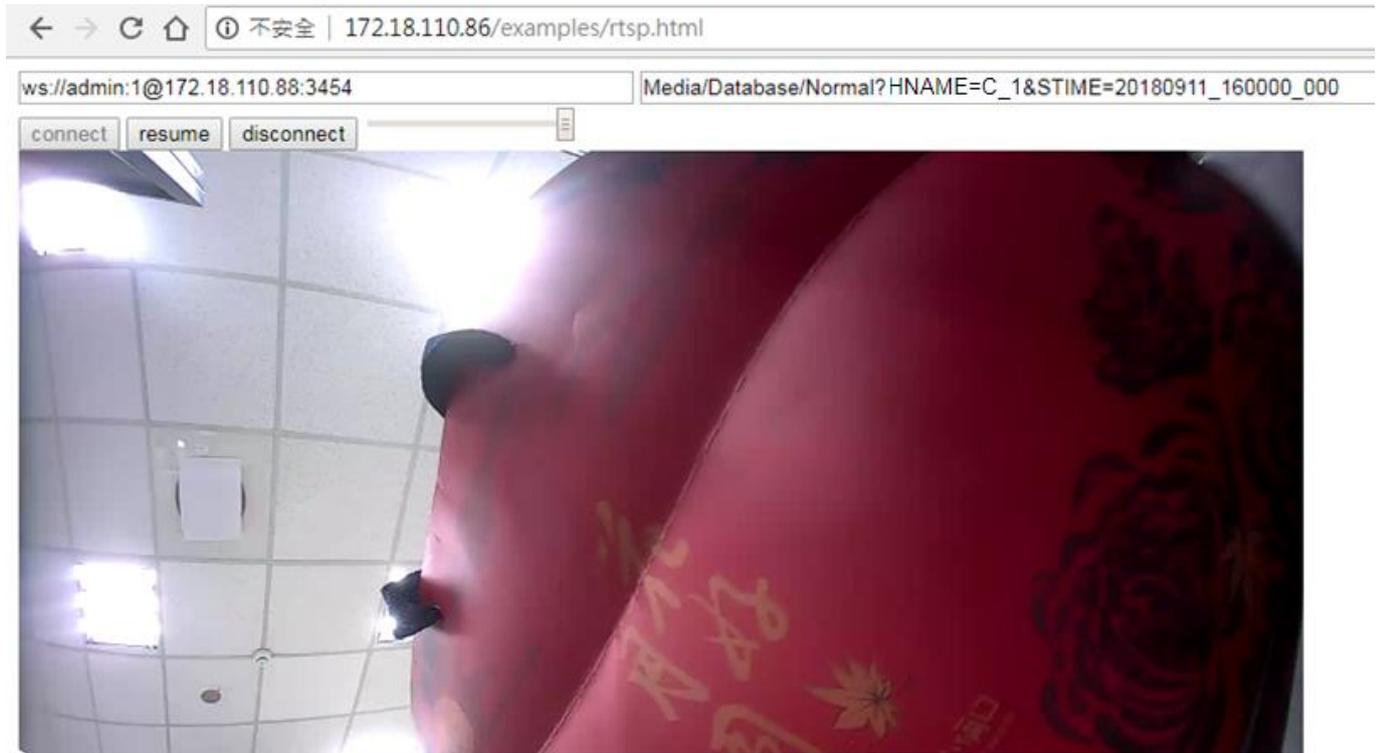
Step3. Open Chrome and type the address as following snapshot shows.

Live:



Websocket server of VAST

Playback:



More Information about web page sample please refer to websocket_api.mht.

VIVOTEK.COM
2019.09.12

5. wss connection

Websocket server supports ws and wss connection and the wss default port is 3443.

The following steps will show you how to connect to websocket server via wss.

1. Visit VAST HTTPS server with browser and accept self-signed SSL certificate.



Your connection is not private

Attackers might be trying to steal your information from **10.16.104.15** (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR_CERT_AUTHORITY_INVALID

Help improve Safe Browsing by sending some [system information and page content](#) to Google. [Privacy policy](#)

Advanced



Back to safety

Advanced

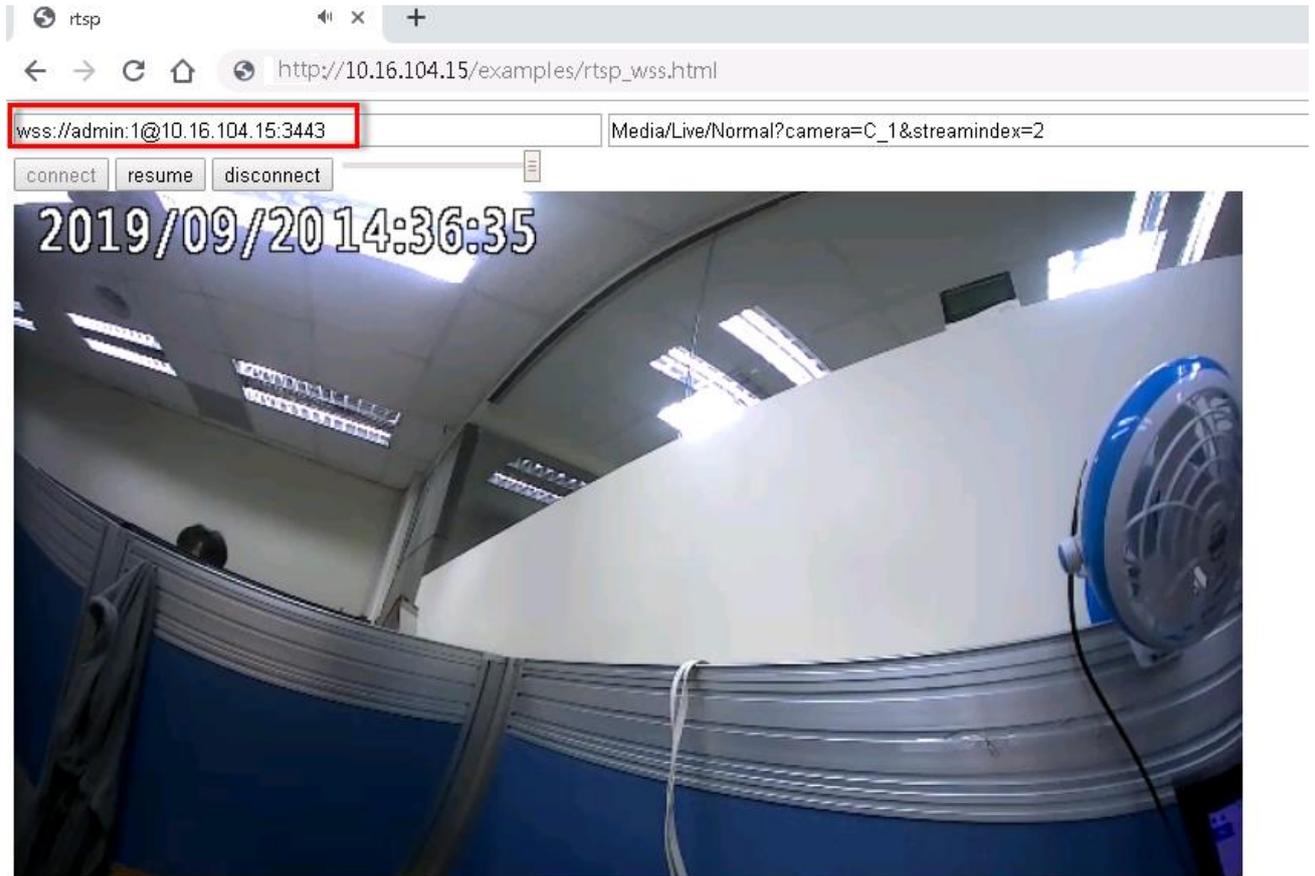
This server could not prove that it is **10.16.104.15**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

[Proceed to 10.16.104.15 \(unsafe\)](#)



WebSocket server of VAST

2. Visit websocket sample webpage and use the protocol **wss**, port **3443**.



6. Restrictions

Websocket server supports all camera supported codecs, but our SDK is base on Media Source Extension so it only supports MP4 ,H264, G711 and G726 due to browser restrctions. Please reference the MDN document for more details.

[https://developer.mozilla.org/en-US/docs/Web/API/Media Source Extensions API#The MSE standard](https://developer.mozilla.org/en-US/docs/Web/API/Media_Source_Extensions_API#The_MSE_standard)

VIVOTEK CONFIDENTIAL
2019.09.20