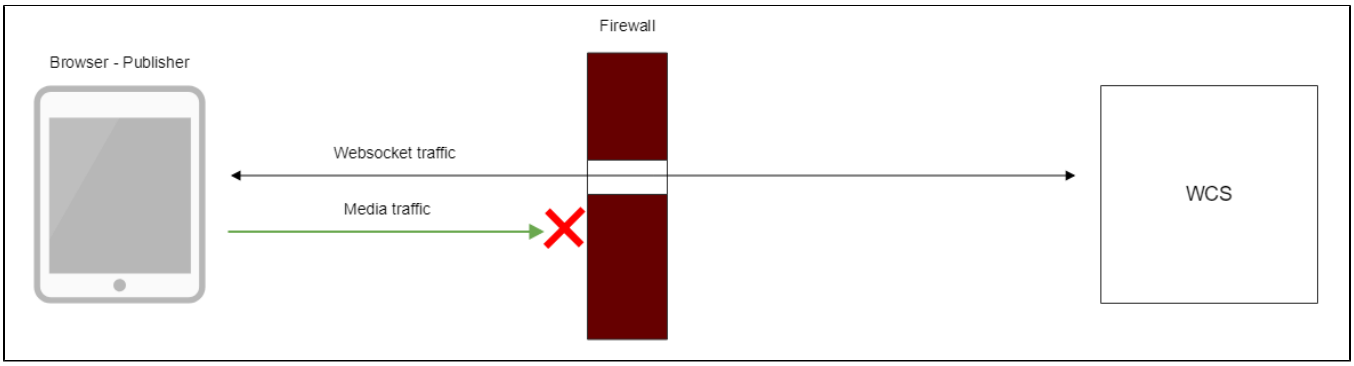
TURN server

Overview

TURN server is used to establish WebRTC connection and transmit media traffic when UDP exchange between client and server is blocked by firewall



There are the following ways to use TURN server with WCS server

1. Using external TURN server, This is suitable for customers with enterprise infrastructure and customers which prefer to separate task performed by servers.

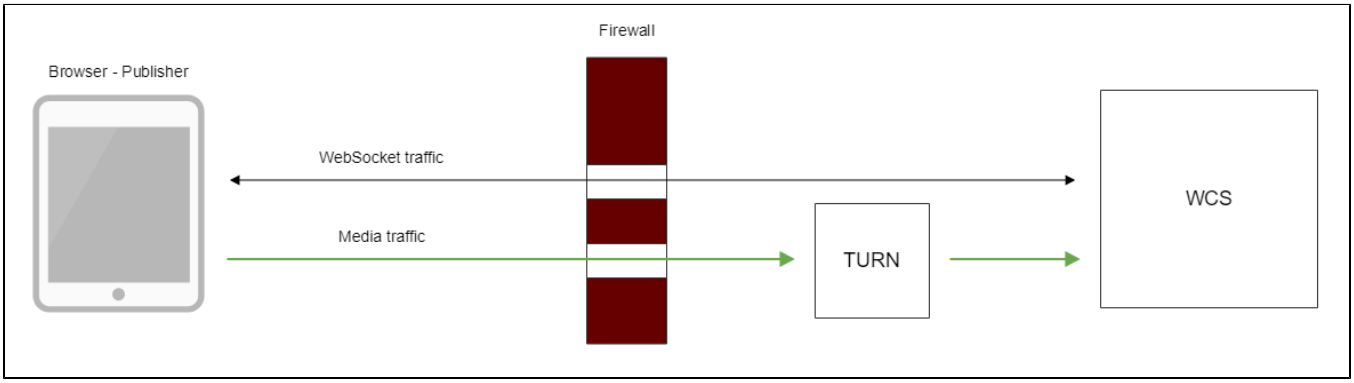
2. Using internal TURN server that embedded to WCS. This is sutable for small office and customer which prefer out-of-the-box solutions.

Supported platforms and browsers

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

Using external TURN server

When external TURN server is used, traffic flows through available ports as follows:



Installing and configuring the TURN server on CentOS 7

1. Download and install turnserver

To compile from sources use the following guide.

2. Create the configuration file turnserver.conf

An example of the turnserver.conf.default configurationb file is in the /usr/local/etc directory. You can rename it to turnserver.conf or create a new file.

Below is an example of the minimum configuration file:

```
fingerprint
lt-cred-mech
user=username1:password1
realm=flashphoner.com
cert=/usr/local/etc/turn_server_cert.pem
pkey=/usr/local/etc/turn_server_pkey.pem
pkey-pwd=qweasd
```

a) As seen from this example, operation of the TURN server requires a certificate and a private key.

If the TURN server is installed on the same server as the WCS server, you can use certificates of the WCS server.

If the TURN server is installed on another server, you can use openssl to generate a certificate and a private key:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/local/etc/turn_server_pkey.pem -out usr/local/etc
/turn_server_cert.pem -days 365
```

b) REQUIRED: the configuration file of the TURN server must enable authorization, and users for authorization must be specified (the first three lines of the configuration file example).

3. Start turnserver

```
turnserver
```

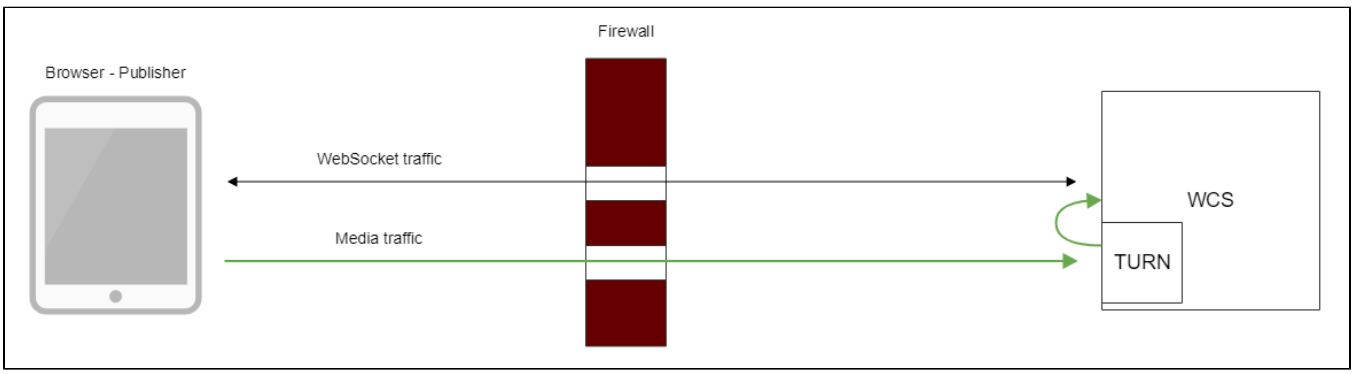
Connection using external TURN server

When you create a session using the Flashphoner API you need to pass the mediaOptions parameter where the coordinates of the TURN server are specified. Below is an example of creating such a session:

```
Flashphoner.createSession({urlServer: url, mediaOptions: {"iceServers": [ { 'url': 'turn:192.168.0.1:3478?
transport=tcp', 'credential': 'password1', 'username': 'username1' } ], "iceTransportPolicy": "relay"}})
```

Using internal TURN server

When internal TURN server is used, traffic flows through available ports as follows:



Client browser establishes TCP connection to internal TURN server port (3478 by default), then internal TURN server connects to local WCS interface through UDP ports from specified range (36001-37000 by default). Thus, for client media traffic flows through TCP tunnel, and for WCS server nothing is changed, traffic flows via UDP ports.

The side effect is that all media traffic between client and server goes through TCP, this may seriously improve quality of HD translations with high bitrate.

Internal TURN server configuration

Internal TURN server should be set up using the following parameters inflashphoner.propertiesfile:

Parameter	Default value	Description
turn_ip	-	Internal TURN server IP address (by default is the same as ip)
turn_ip_local	-	Internal TURN server local IP address used for port binding (by default is the same as ip_local)
turn_port	3478	Internal TURN server TCP port
turn_password	coM77EMrV7Cwhyan	Internal TURN server password (username is `flashphoner`)
turn_media_port_from	36001	Beginning of media UDP ports range for WebRTC connection between TURN and WCS
turn_media_port_to	37000	End of media UDP ports range
turn_media_ports_auditor_interval	5000	Audit interval for busy and free ports, in milliseconds
turn_media_ports_auditor_max_attempts	3	Number of audits to make sure freed port is not bound
turn.server_channel_receive_buffer_size	1048576	Receive buffer size in bytes
turn.server_channel_send_buffer_size	1048576	Send buffer size in bytes

Connection using internal TURN server

When you create a session using the Flashphoner API you need to pass the mediaOptions parameter where the coordinates of the internal TURN server are specified:

```
Flashphoner.createSession({urlServer: url, mediaOptions: {"iceServers": [ { 'url': 'turn:test.flashphoner.com:3478?transport=tcp', 'credential': 'coM77EMrV7Cwhyan', 'username': 'flashphoner' } ]}})
```

If UDP ports are **not** blocked by firewall, browser may establish WebRTC connection through UDP. In this case, `"iceTransportPolicy": "relay"` parameter should be set in`mediaOptions`

```
Flashphoner.createSession({urlServer: url, mediaOptions: {"iceServers": [ { 'url': 'turn:test.flashphoner.com:3478?transport=tcp', 'credential': 'coM77EMrV7Cwhyan', 'username': 'flashphoner' } ], "iceTransportPolicy": "relay"}})
```

for media traffic go through TURN server.

Quick manual for testing

1. For test we use:

- WCS with active embedded TURN server
- Firewall Traversal Streamingweb application in Chrome browser
- iptables to block UDP ports on server

2. Block UDP posrt on external network interface of WCS server

```
iptables -i ens192 -I INPUT -m udp -p udp --dport 0:65535 -j DROP
```

Where ens192 is external network interface of WCS server.

If this step is successful, go to 3b, if ports cannot be blocked, go to 3a

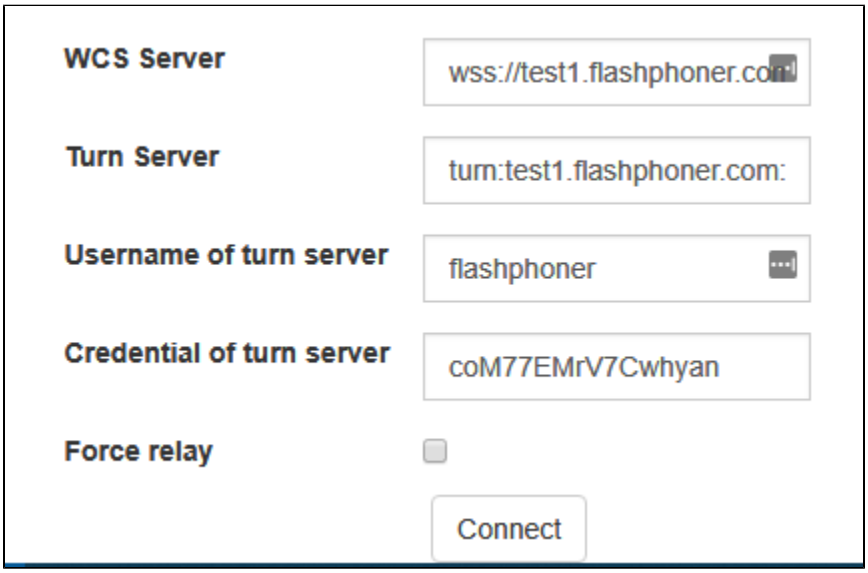
3. Open Firewall Traversal Streaming application, set the following to`TURN server`

```
turn:test.flashphoner.com:3478?transport=tcp
```

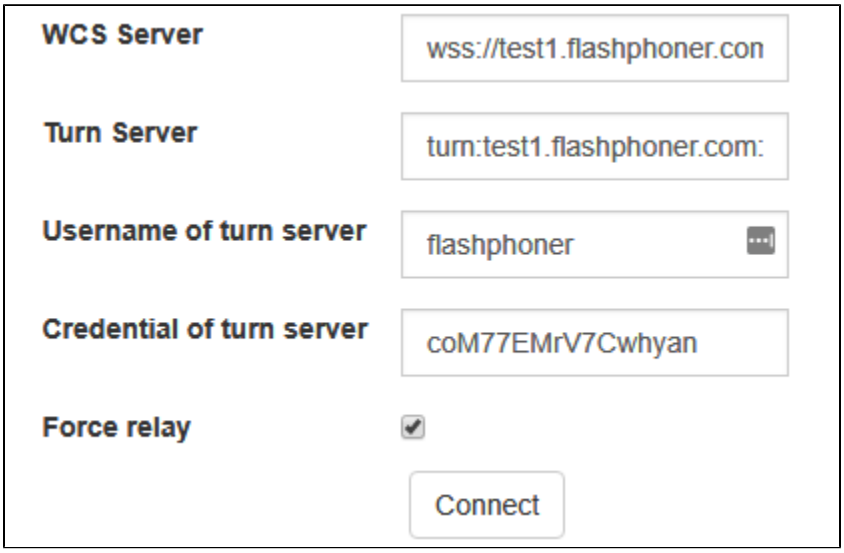
Where

- test.flashphoner.com is WCS server hostname
- 3478 internal TURN server port

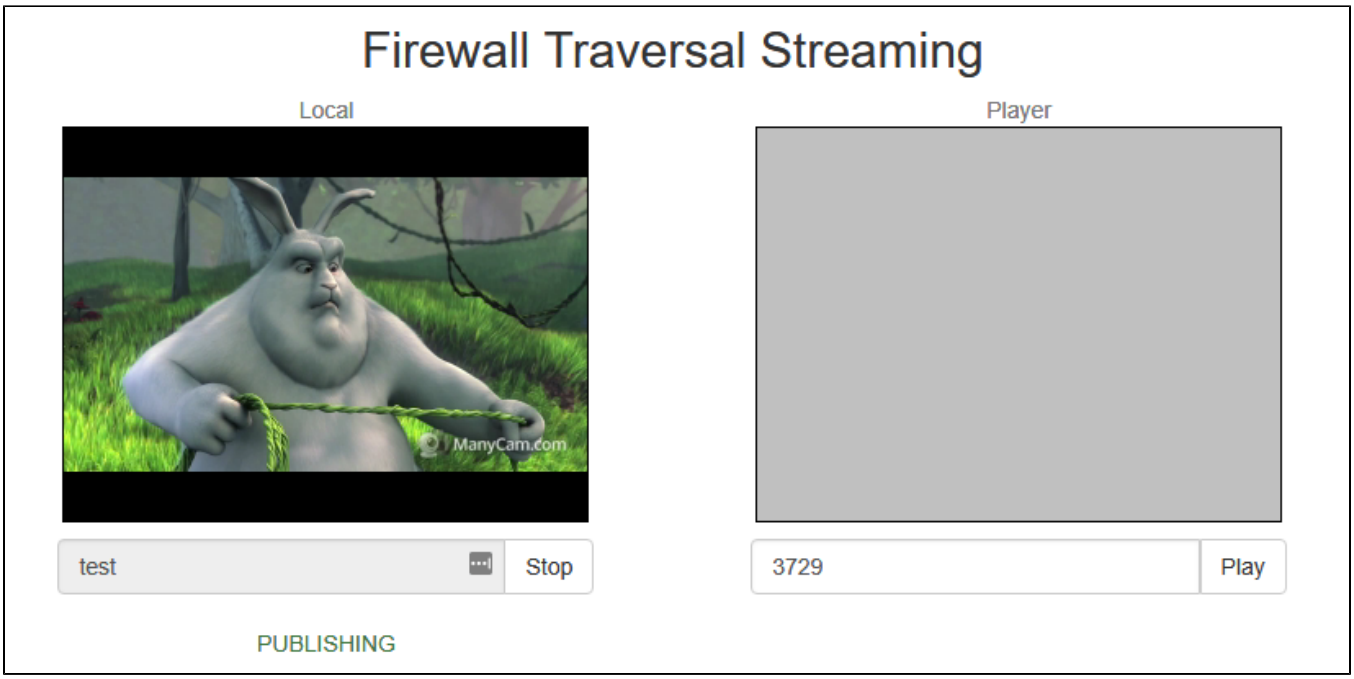
a) if UDP ports on WCS server are not blocked, uncheck`Force relay`



b)if UDP ports on WCS server are blocked, check`Force relay`



4. Press`Connect`, enter stream name`test`and press`Publish`. Stream publishing starts through internal TURN server.



Known issues

1. The Microsoft Legacy Edge and Mozilla Firefox browser does not connect via the TURN server

Symptoms: publishing/playing do not work when trying to connect through the TURN server

Solution: use Chrome browser or its successors.