# Accessory tools


- Port routing checking
  - nc
  - tcpdump
- jstack tool
- Testing channel bandwidth using iperf


This section describes accessory tools that can be used to manage the WCS server.

## Port routing checking

The WCS server may be behind NAT and as such it will require a port range opened for the external network, for instance, UDP 31000-32000.
This means that a UDP packet sent from the external network to the port in that range should reach the server where WCS is placed.

Hence, we have a simple test. Send a UDP packet from outside using nc and receive it on the server using tcpdump.
If the packet reached, the port is open.

### nc

```
echo -n "hello" | nc -4u -w1 wcs1.com 31000
```

or for Debian:

```
echo -n "hello" | nc -u -w1 wcs1.com 31000
```

This command sends a simple UDP packet in the given direction.

### tcpdump

```
tcpdump udp port 31000
```

This command makes the server listen for a particular port and immediately outputs information about packet arrival to the console:

```
17:50:21.932509 IP myhost.39194 > host.31000: UDP, length 5
```

## jstack tool

This is Java utility that provides important information about a Java process and execution threads.
When you run jstack from the console, a brief information about jstack is shown:

```
[root@localhost bin]# jstack
Usage:
    jstack [-l] <pid>
        (to connect to running process)
    jstack -F [-m] [-l] <pid>
        (to connect to a hung process)
    jstack [-m] [-l] <executable> <core>
        (to connect to a core file)
    jstack [-m] [-l] [server_id@]<remote server IP or hostname>
        (to connect to a remote debug server)

Options:
    -F  to force a thread dump. Use when jstack <pid> does not respond (process is hung)
    -m  to print both java and native frames (mixed mode)
    -l  long listing. Prints additional information about locks
    -h or -help to print this help message
```

If the information is not shown or the jstack utility is not found, use the installation instruction to latest version of JDK. After installing jdk you should create a symbolical link to jstack to quickly run it:

```
ln -sf /usr/java/default/bin/jstack /usr/bin/jstack
```

Example:

```
jstack 8888 > jstack.report
```

Here, 8888 is the ID of the Java process.

# Testing channel bandwidth using iperf

A stream published picture quality depends on channel bandwidth between publisher and server, the same for subscriber. Channel bandwidth can be checked usingiperfutility. This program is implemented for all major OS: Windows, MacOS, Ubuntu/Debian, CentOS. iperf in server mode can be installed and running with WCS, that allows to check whole channel bandwith from publisher to viewer.

iperf can be installed on CentOS 7 as follows:

```
yum install iperf3
```

Run iperf in server mode

```
iperf3 -s -p 5201
```

where 5201 is iperf port for testing client connections

On client side iperf can be launched as follows:

1. To test upload channel bandwith via UDP (Windows example)

```
iperf3.exe -c test2.flashphoner.com -p 5201 -u
```

Where

- test2.flashphoner.com - WCS server
- 5201 - iperf port to connect

The result of the command above should look like this:

```
Connecting to host test2.flashphoner.com, port 5201
[  4] local 192.168.0.195 port 51502 connected to 95.191.131.65 port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec  4.50 MBytes  37.7 Mbits/sec
[  4]   1.00-2.00   sec  5.50 MBytes  46.1 Mbits/sec
[  4]   2.00-3.00   sec  5.62 MBytes  47.2 Mbits/sec
[  4]   3.00-4.00   sec  4.00 MBytes  33.6 Mbits/sec
[  4]   4.00-5.00   sec  3.62 MBytes  30.4 Mbits/sec
[  4]   5.00-6.00   sec  4.62 MBytes  38.8 Mbits/sec
[  4]   6.00-7.00   sec  4.75 MBytes  39.8 Mbits/sec
[  4]   7.00-8.00   sec  2.75 MBytes  23.1 Mbits/sec
[  4]   8.00-9.00   sec  2.25 MBytes  18.9 Mbits/sec
[  4]   9.00-10.00  sec  3.62 MBytes  30.4 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-10.00  sec  41.2 MBytes  34.6 Mbits/sec                  sender
[  4]   0.00-10.00  sec  41.2 MBytes  34.6 Mbits/sec                  receiver

iperf Done.
```

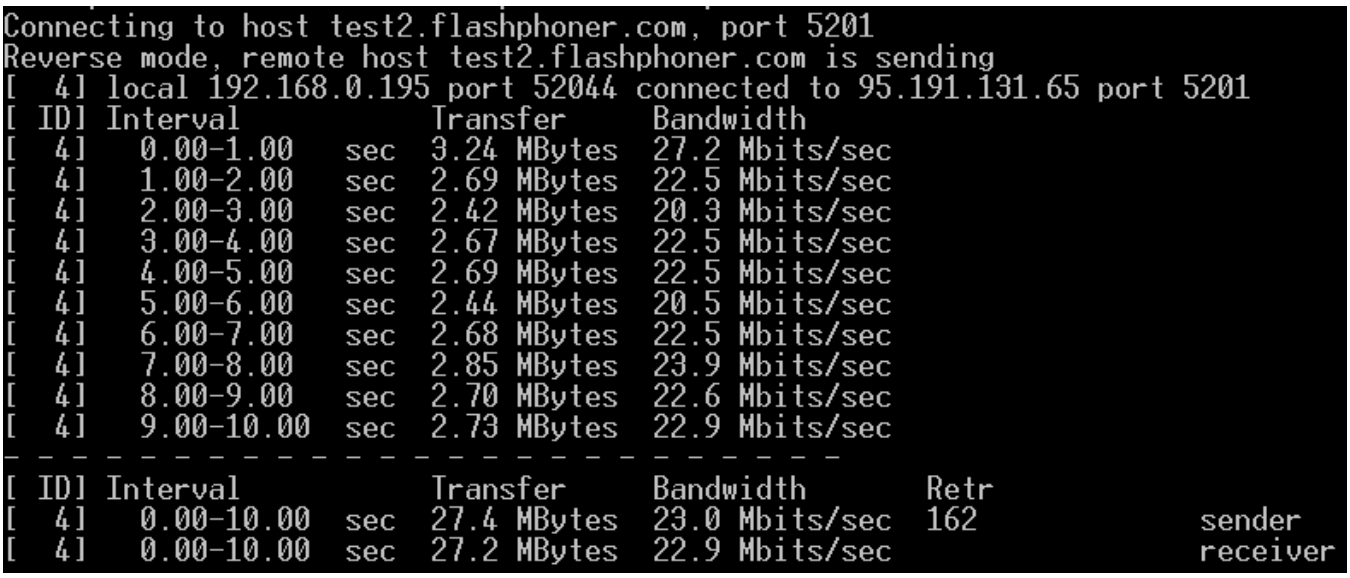
2. To test download channel bandwidth via UDP

```
iperf3.exe -c test2.flashphoner.com -p 5201 -u -R
```

Where

- test2.flashphoner.com- WCS server
- 5201 - iperf port to connect

The result of the command above should look like this:




By default, iperf tests the channel for 10 seconds. This interval should be increased, for example, to 120 second

```
iperf3.exe -c test2.flashphoner.com -p 5201 -u -t 120
```

The upload channel bandwidth test via UDP result shows the maximum video publishing bitrate without packet losses. In the sample above bitrate should be limited with 1000 kbps, on server side for example

```
webrtc_cc_max_bitrate=1000000
```

Note that iperf major versions on server and on testing client should be the same. Today version 3 is actual, but ther is also version 2 in repositories.