# NAT Traversal

## About

Some difficulties have been encountered with devices that have poor NAT support. FreeSWITCH goes to great lengths to repair broken NAT support in phones and gateway devices.

In order to aid FreeSWITCH in traversing NAT please see the External profile page.

Some routers offer an Application Layer Gateway feature which can prevent FreeSWITCH NAT traversal from working. See the ALG page for more information, including how to disable it.

## Using STUN to aid in NAT Traversal

STUN is a method to allow an end host (i.e. phone) to discover its public IP address if it is located behind a NAT . Using this method requires a STUN server on the public internet and a client on the phone. The phone's STUN client queries the STUN server for it's own public IP and transmits the information it has received in it's connection information in the SIP packets it sends to the SIP server.

Enable and configure STUN settings on your phone in order correctly to report your phone's contact information to FreeSWITCH when registering. Unfortunately, not all phones have a properly working STUN client.

### STUN servers

This site contains a list of public STUN servers: https://gist.github.com/zziuni/3741933
stun.freeswitch.org is never guaranteed to be up and running so use it in production at your own risk. There are several open source projects to run your own STUN server, e.g. STUNTMAN

## Using FreeSWITCH built-in methods to aid in NAT Traversal

### nat-options-ping

This parameter causes FreeSWITCH to regularly (every 20 - 40s) send an OPTIONS packet to NATed registered endpoints in order to keep the port on the clients firewall open. This is usefull if the initial REGISTER works, but subsequently the phone is no longer reachable, as the clients firewall has closed the port the REGISTER originated from.

Activate this parameter in sip_profiles/internal.xml and restart the sip profile:

**sip_profiles/internal.xml**

```
<param name="nat-options-ping" value="true"/>
```

### NDLB-force-rport

This parameter forces FreeSWITCH to send SIP responses to the network port from which they were received. Use with caution, as it may break things for devices that actually expect to get replies on a different port. See Sofia Configuration Files for more details.

**sip_profiles/internal.xml**

```
<param name="NDLB-force-rport" value="true/safe"/>
```

### apply-nat-acl

You can add the apply-nat-acl param to your profile to force NAT behavior when matching a certain access list

```
<param name="apply-nat-acl" value="rfc1918"/>
```

Possible values of apply-nat-acl using access list: ACL Notes

### sip_sticky_contact

```
<action application="set" data="sip_sticky_contact=true"/>
```

### sip_nat_detected

sip_nat_detected is set to true when NAT is detected. Use it in your dialplan to handle NATted devices differently.

```
<condition field="${sip_nat_detected}" expression="true">
```

### sip-force-contact

The sip-force-contact variable can be used to activate NATHACK / TLSHACK registration, which rewrites the contact IP:port.

**User Directory Example**

```
<user id="100" mailbox="100">
  <params>
    <param name="password" value="1234"/>
    <param name="vm-password" value="4321"/>
  </params>
  <variables>
    <variable name="sip-force-contact" value="NDLB-connectile-dysfunction"/>
  </variables>
</user>
```

### NDLB-connectile-dysfunction

- Rewrites contact IP and port to that of the NAT device by looking at IP address:port info from the packets reaching FreeSWITCH.
- Forces REGISTER expiry to 30 seconds. (Unless sip-force-expires is set)

This is similar to the way Asterisk tries to deal with NAT traversal.

### NDLB-tls-connectile-dysfunction

- Rewrites contact port

### NDLB-connectile-dysfunction-2.0

Needs documentation; apparently rewrites IP and port and adds the `fs_path` parameter in the Contact header.

- Forces REGISTER expiry to 30 seconds. (Unless sip-force-expires is set)

## Using Phone specific features to aid in NAT Traversal

Your phone may allow you to specify your public network information to use when registering and/or a keepalive-mechanism to keep the signalling-port open. Please check with the manufacturer of your phone for the best information.

### Polycom

```
<nat nat.ip="4.2.2.2" nat.signalPort="" nat.mediaPortStart="" nat.keepalive.interval=""/>
```

## FreeSWITCH Behind NAT

With FreeSWITCH behind NAT, FreeSWITCH can only bind its ports to a local IP. However when connecting to FreeSWITCH from an external network, the external IP is needed.

With the standard setup users may be able to register phones correctly, however the phones may not be reachable and you may encounter no audio or one way audio when a call is set up.

This is one working example for FreeSWITCH behind NAT:

**vars.xml**

```
<X-PRE-PROCESS cmd="set" data="bind_server_ip=my.domain.com"/>
<X-PRE-PROCESS cmd="set" data="external_sip_ip=stun:stun.freeswitch.org"/>
<X-PRE-PROCESS cmd="set" data="external_rtp_ip=stun:stun.freeswitch.org"/>
<X-PRE-PROCESS cmd="set" data="add_ice_candidates=true"/>
```

host:domain.example.com is another possible value; however this will not toggle the autonat flags.

If FreeSWITCH sits behind NAT with dynamic DNS and stun doesn't work, you should write a script that determines your public IP address, makes the change, and calls reloadxml. This also holds true for the external profile. No special processing happens to determine the IP address before the variable gets passed to the external profile.

host:domain.example.com may be used in places "where you have two interfaces in a box and one is public–facing and one isn't, so one never has to tell the lies."

### sip_profiles

**internal.xml**

```
<param name="ext-rtp-ip" value="$${external_rtp_ip}"/>
```

**external.xml**

```
<param name="ext-sip-ip" value="$${external_sip_ip}"/>
<param name="ext-rtp-ip" value="$${external_rtp_ip}"/>
```



Do not set ext-rtp-ip to a domain name instead of an IP or STUN entry; you will encounter a "SIP/2.0 500 Cannot Get IP Address for Media" error.

By default external_sip_ip and external_rtp_ip are set in vars.xml to use the FreeSWITCH STUN server.

## Restricting Port Range

Some firewall providers provide limited support for port forwarding or virtual servers or whatever they are called by the provider. To help, FreeSWITCH can limit the ports it will use for RTP streams, so you don't have to forward 16,000 ports. You will need enough to handle all media channels coming across the firewall. Keep that in mind.

Specify the lower and upper bounds on port numbers in conf/autoload_configs/switch.conf.xml as follows:

**switch.conf.xml**

```
<param name="rtp-start-port" value="16384"/>
<param name="rtp-end-port" value="16389"/>
```

In the example above, you only need to forward 6 ports, so it will also only allow a maximum of 6 audio channels across that connection, and depending on networking configurations maybe not even that. Be very careful with your expectation level here.

## Distilled Wisdom

Archived for posterity from: [FS-users mailing list]

Kristian Kielhofner <kris at kriskinc.com> Wed Nov 21 07:49:52 MSK 2012

You can rewrite the SDP to anything you want but that doesn't mean you'll get media.

I suggest you do some research on the two main schools for SIP NAT traversal (far end and near end). In a nutshell, far end NAT traversal is implemented by the server. This is the FreeSWITCH default. The SDPs are rewritten to the address of the FreeSWITCH server and passed to the remote endpoint. This causes the media to be relayed at the server between the endpoints. This has the advantage of being universally compatible - regardless of how dumb the endpoint(s) is/are. Disadvantages are many - increased sever load, bandwidth, latency, etc. However, with many endpoints this is the only choice.

Near end NAT traversal relies on advances technologies such as STUN, TURN, and ICE to be supported by the client (and the servers to enable them configured and available). TURN actually blurs the lines between these two strategies, as do various hybrid approaches using local proxies, etc but that's for another day.

Bypass media is incompatible with NAT unless you're using STUN/TURN/ICE in your clients and even then I'm not sure FreeSWITCH will completely and cleanly handle the various ICE/SIP/SDP interactions.

## Related Pages

- General_NAT_example_scenarios
- NAT
- Access Control List (ACL) (Modifying NAT behavior when matching a certain access list)