

WebRTC, STUN and TURN

Amitesh Madhur

Cisco Systems

Agenda

- ✓ User Media
- ✓ Peer Connection
- ✓ STUN and TURN
- ✓ Setting up STUN and TURN

Agenda

- ✓ User Media
- ✓ Peer Connection
- ✓ STUN and TURN
- ✓ Setting up STUN and TURN

Get User Media

- ✓ `getUserMedia()`
- ✓ Collects video audio input
- ✓ Synchronization of input

```
<video id="me" autoplay></video>
```

```
navigator.getUserMedia = navigator.getUserMedia ||  
    navigator.webkitGetUserMedia ||  
    navigator.mozGetUserMedia;  
  
if (navigator.getUserMedia) {  
    navigator.getUserMedia({video: true}, onSuccess, onError);  
}  
  
window.URL = window.URL || window.webkitURL;  
var me = document.getElementById('me');  
  
function onSuccess(stream) {  
    me.src = window.URL.createObjectURL(stream);  
}  
  
function onError(e) {  
    // error  
}
```

Vendor Prefixes

Chrome \geq 21:

- ✓ `webkitGetUserMedia()`
- ✓ `window.webkitURL.createObjectURL()`

Opera \geq 12:

- ✓ `getUserMedia()`
- ✓ `set video.src`

Firefox \geq 20:

- ✓ `mozGetUserMedia()`
- ✓ `window.URL.createObjectURL()`

IE: not implemented

Agenda

- ✓ User Media
- ✓ Peer Connection
- ✓ STUN and TURN
- ✓ Setting up STUN and TURN

Peer Connection

- ✓ Establish a connection
- ✓ Pass the user media stream
- ✓ Other side gets the stream
- ✓ Add the received stream to <video> tag

Peer Connection

- ✓ Compression and de-compression
- ✓ P2P connection using STUN or TURN
- ✓ Encrypting the data

```
<video id="me" autoplay></video>
<video id="other" autoplay></video>
```

```
peer = new RTCPeerConnection(servers);
peer.onaddstream = gotRemoteStream;
peer.addStream(localStream);

if(host) {
peer.createOffer(callGotOffer, null, {mandatory: {
    OfferToReceiveAudio: true,
    OfferToReceiveVideo: true}});

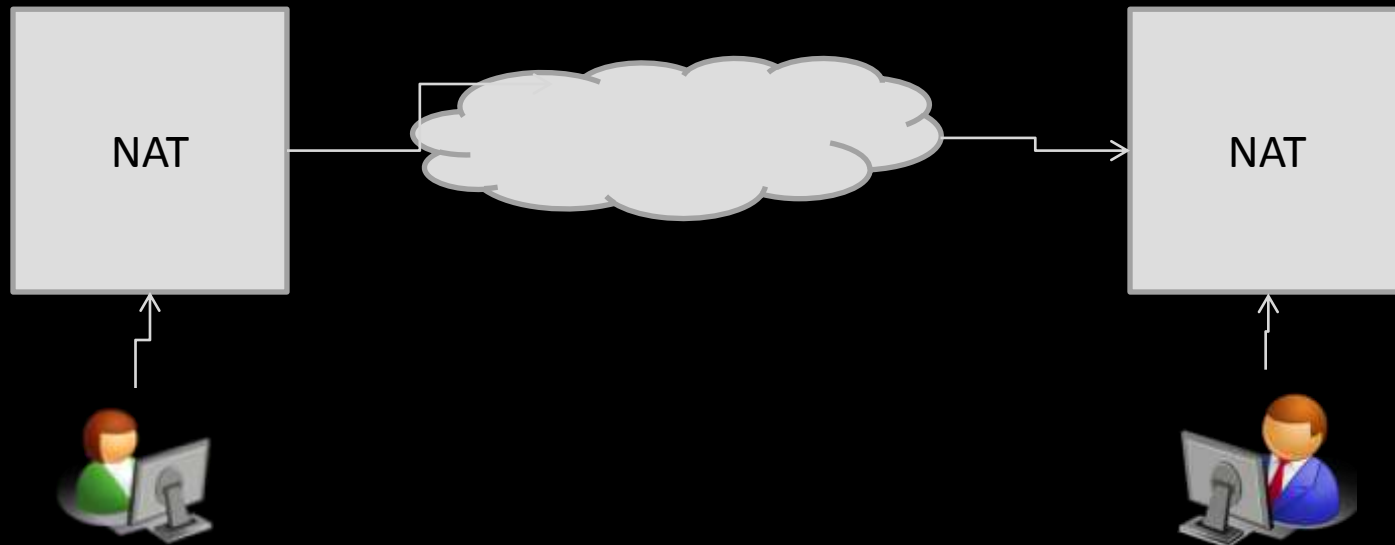
} else {
peer.createAnswer(peer.remoteDescription, callGotOffer);
}
function callGotOffer(sd) {
    peer.setLocalDescription(sd);
}
function gotAnswer(desc) {
    peer.setRemoteDescription(new RTCSessionDescription(desc));
}
function gotRemoteStream(e) {
    attachMediaStream(remoteVideo, e.stream);
}
```

<chrome://webrtc-internals/>

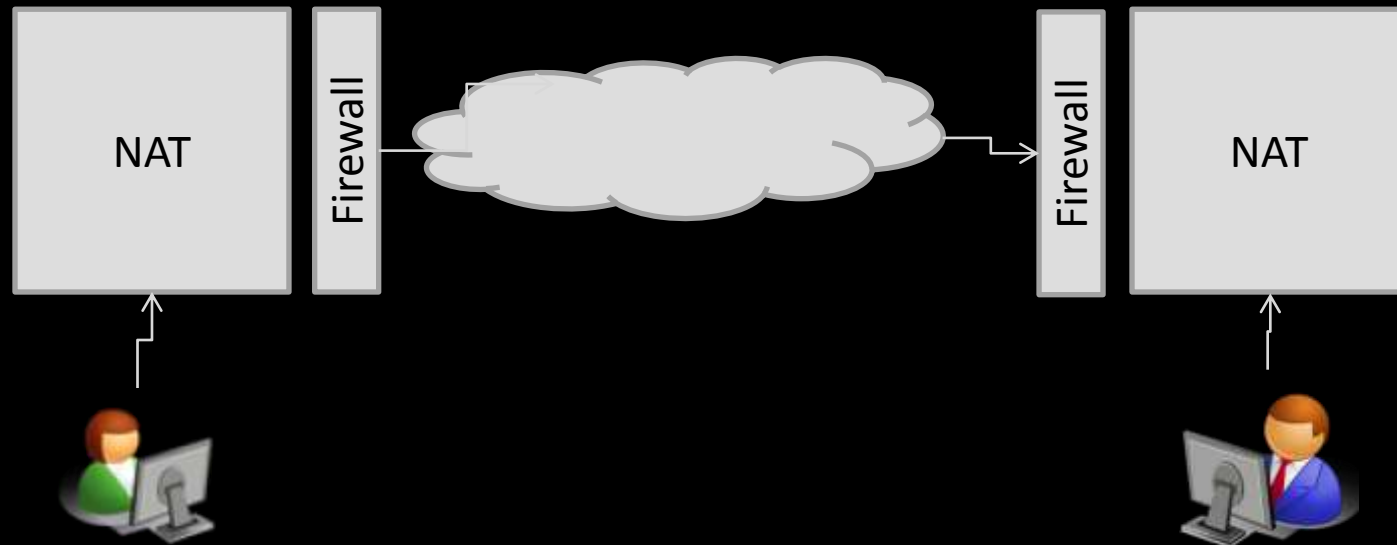
Agenda

- ✓ User Media
- ✓ Peer Connection
- ✓ STUN and TURN
- ✓ Setting up STUN and TURN

Need for STUN



Need for TURN



STUN and TURN

```
peer = new RTCPeerConnection(servers);
```

STUN and TURN

```
var chrome, stun, turn, servers = {}, peer;
chrome = parseInt(navigator.userAgent.match(/Chrom(e|ium)\/([0-9]+)\./)[2]);

stun = {url: 'stun:10.0.0.7:9999'};
turn = {url: 'turn:amitesh@10.0.0.7:3333', credential: 'pass'};

if (isChrome && chrome >= 28) {
    turn = {
        url: 'turn:10.0.0.7:3333',
        credential: 'madhur',
        username: 'amitesh'
    };
}

servers.iceServers = [stun, turn];
peer = new RTCPeerConnection(servers);
```


Public STUN

stun.l.google.com:19302

stun1.l.google.com:19302

stun2.l.google.com:19302

stun3.l.google.com:19302

stun4.l.google.com:19302

Agenda

- ✓ User Media
- ✓ Peer Connection
- ✓ STUN and TURN
- ✓ Setting up STUN and TURN

STUN on Linux

✓ stunprotocol.org

STUNTMAN

Open source STUN server software

About

STUNTMAN is an open source implementation of the [STUN protocol](#) (*Session Traversal Utilities for NAT*) as specified in RFCs [5389](#), [5769](#), and [5780](#). It also includes backwards compatibility for RFC [3489](#). Source code distribution includes a high performance STUN server, a client application, and a set of code libraries for implementing a STUN client within an application.

The current C++ code base compiles using gcc/g++ for most UNIX distributions including Linux, MacOS, BSD, and Solaris. A Windows port via Cygwin is available now.

Stunman is open source software distributed under [the Apache 2.0 License](#).

Download

Latest stable release: version 1.2.5



Unix/Linux sources: [stunserver-1.2.5.tgz \(click here\)](#)



Win32 binaries: [stunserver_win32_1_2_5.zip \(click here - redirects to SourceForge\)](#)

You may also peruse the source files individually, access previous versions, file bugs, and find other resources at the project's [GitHub site](#).

STUN on Linux

- ✓ Run “make”
- ✓ Provides 3 libraries
(stunserver, stunclient and stuntestcode)
- ✓ Run `stunserver --help` or `stunclient --help`
- ✓ By default runs on port 3478

STUN on Windows

✓ stunprotocol.org

STUNTMAN

Open source STUN server software

About

STUNTMAN is an open source implementation of the [STUN protocol](#) (*Session Traversal Utilities for NAT*) as specified in RFCs [5389](#), [5769](#), and [5780](#). It also includes backwards compatibility for RFC [3489](#). Source code distribution includes a high performance STUN server, a client application, and a set of code libraries for implementing a STUN client within an application.

The current C++ code base compiles using gcc/g++ for most UNIX distributions including Linux, MacOS, BSD, and Solaris. A Windows port via Cygwin is available now.

Stunman is open source software distributed under [the Apache 2.0 License](#).

Download

Latest stable release: version 1.2.5



Unix/Linux sources: [stunserver-1.2.5.tgz](#) (click here)



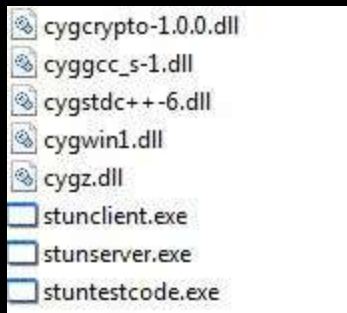
Win32 binaries: [stunserver_win32_1_2_5.zip](#) (click here - redirects to SourceForge)

You may also peruse the source files individually, access previous versions, file bugs, and find other resources at the project's [GitHub site](#).

STUN on Windows

- ✓ Unzip
- ✓ goto command prompt and run server:

*stunserver.exe --mode full --primaryinterface
10.0.0.6 --altinterface 10.0.0.11 --altport 999 --
primaryport 9999 --verbosity 3*



TURN on Linux

✓ Build libevent

```
$ wget https://github.com/downloads/libevent/libevent/libevent-2.0.21-stable.tar.gz
$ tar xvfz libevent-2.0.21-stable.tar.gz
$ cd libevent-2.0.21-stable
$ ./configure
$ make
$ make install
```

TURN on Linux

✓ Download TURN from

<https://code.google.com/p/rfc5766-turn-server>

```
$ tar xvfz turnserver.tar.gz
```

```
$ ./configure
```

```
$ make
```

```
$ make install
```


TURN on Linux

- ✓ Copy the “turnserver.conf” from
`turnserver/examples/etc/turnserver.conf`
to
`/usr/local/etc/turnserver.conf`

- ✓ Changes in turnserver.conf

`listening-port=<new-port>`

`listening-ip=<your ip>`

`user=<user>:<password>`

TURN on Linux

✓ Run turnserver

turnserver -L <ip_address>

IN YOUR JAVASCRIPT

```
var turn;
```

```
turn = {          url: 'turn:<user-name>@<IP>:<PORT>',
                  credential: 'password'
        };
```

```
// for chrome 28 and above
```

```
turn = {
    url: 'turn:<IP-address>:<PORT>',
    username: '<user-name>',
    credential: '<password>'
};
```

TURN on Windows

- ✓ Install Cygwin
- ✓ Make sure to install devel dependencies
- ✓ Build libevent

```
$ wget https://github.com/downloads/libevent/libevent/libevent-2.0.21-stable.tar.gz
$ tar xvfz libevent-2.0.21-stable.tar.gz
$ cd libevent-2.0.21-stable
$ ./configure
$ make
$ make install
```

TURN on Windows

✓ Download TURN from

<https://code.google.com/p/rfc5766-turn-server>

```
$ tar xvfz turnserver.tar.gz
```

```
$ ./configure
```

```
$ make
```

```
$ make install
```

TURN on Windows

- ✓ Issues during ./configure
 - ✓ Missing libpq.a, hiredis, postgresql
(You may ignore them, since these are optional)
- ✓ Issues during make
 - ✓ error: 'struct sockaddr_in' has no member named 'sin_len'

Edit turnserver.version/src/client/na_turn_ioaddr.c and comment line #169// addr->s4.sin_len = sizeof(struct sockaddr_in);

TURN on Windows

- ✓ Run fixing “make” and “make install”
- ✓ Copy the “turnserver.conf” from
turnserver/examples/etc/turnserver.conf
to
/usr/local/etc/turnserver.conf
- ✓ Changes in turnserver.conf

```
listening-port=<new-port>
```

```
listening-ip=<your ip>
```

```
user=<user>:<password>
```

TURN on Windows

✓ Run turnserver

turnserver.exe -a -r 10.0.0.6

IN YOUR JAVASCRIPT

```
var turn;
```

```
turn = {          url: 'turn:<user-name>@<IP>:<PORT>',
                  credential: 'password'
                };
```

```
// for chrome 28 and above
```

```
turn = {
    url: 'turn:<IP-address>:<PORT>',
    username: '<user-name>',
    credential: '<password>'
};
```

Thank you

Demos : <http://webrtc.googlecode.com/svn/trunk/samples/js/demos/>

Twitter: @amiteshawa

Facebook: <facebook.com/amitesh.madhur>

Linkedin: <in.linkedin.com/pub/amitesh-madhur/a/932/499>