Analysing Middlebox Interactions

Seppo Hätönen
shatonen@cs.helsinki.fi

The goal of this analysis is to gather information about different NAT traversal environments with Session Initiation Protocol.

Various middleboxes such as Network Address Translators (NATs) are very common in today's Internet. The main benefit of the NAT is that multiple computers can be placed behind a NAT with the NAT having only a single public IP address. This allows Internet Service Providers and organizations to use their address pool more efficiently and at the same time helps conserving the diminishing IPv4 address space.

Due to the characteristics of the NAT implementations, the NAT works reasonably well with client-to-server type of protocols and applications. Unfortunately, today peer-to-peer type of protocols and applications are starting to be more widespread in the Internet. The NAT causes problems with this kind of connectivity as NATs block all unknown traffic from outside network to the hosts behind the NAT. This means special NAT traversal methods need to be deployed for protocols and applications like Voice over Internet Protocol (VoIP) to make them work.

Traversal Protocols

STUN – Session Traversal Utilities for NAT

- Provides means for hosts to find out about their network environment.
- Allows applications to discover the address and port pair that the NAT has allocated for them on the public side of the network.

TURN – Traversal Using Relays around NAT

- In some cases the only way around a middlebox is relaying all traffic through a server
- A host behind a NAT registers itself to a TURN server
- Packets from the host seem to originate from the TURN server
- Remote hosts connect to the TURN server when they need to communicate with the host behind the NAT

ICE - Interactive Connectivity Establishment

- Provides a framework for different NAT traversal technologies
- ICE uses STUN and TURN for NAT traversal
- Two hosts can use ICE to discover a working address pair for communication

NAT Traversal Test Platform

The goal of this work is to create a test environment to gather data about real-world network statistics and evaluate how well different NAT traversal mechanisms work. The environment consists of SIP server, STUN and TURN servers, a couple of dummy SIP clients behind different NATs and an automated test client. The test client will be easily available and will run its test automatically and after finishing the test, the client will send the results to the test server for analysis.

Test Components

- SIP server
- STUN / TURN server
- Dummy SIP clients behind NAT(S)
- Test client (Linux)

Experimentation

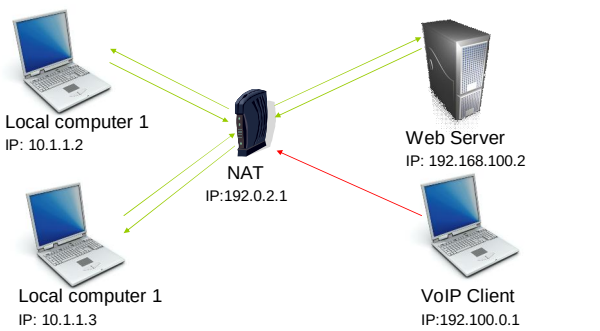
- Connectivity with no traversal mechanism
- Connectivity with STUN
- Connectivity with TURN
- Connectivity with ICE

At the first stage, the client fetches a configuration file from a web server. The configuration file contains information about the username, password, the addresses of the servers to be used, which dummy peer to call etc.

Next the client proceeds with the specified information to register to the given SIP server and make a call to the given dummy client.

For the call to go through, the client tries to use a set of TURN, STUN and ICE mechanism to traverse the NAT.

After the connectivity tests are complete, the client sends results back to the server for analysis.



Source address before NAT	Source address after NAT
10.1.1.2:17240	192.0.2.1:7000
10.1.1.3:28240	192.0.2.1:7001

Figure 1: Network Address Translation

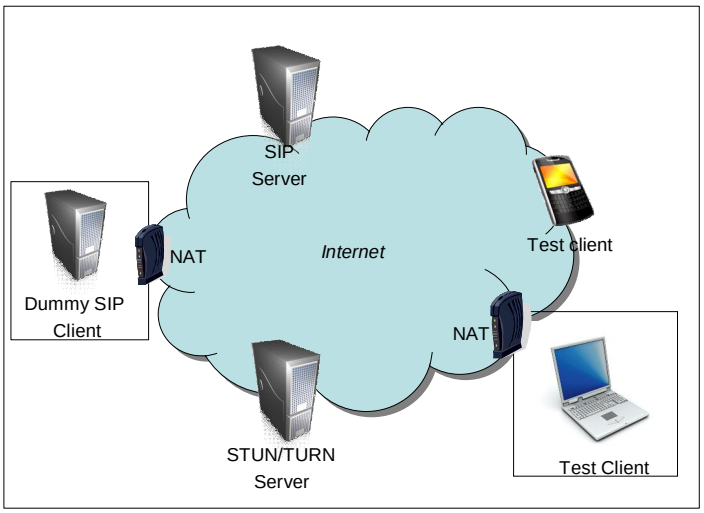


Figure 2: Test setup

WISEciti – Wireless Community Services for Mobile Citizens (2008 – 2010)

http://www.cs.helsinki.fi/group/wiseciti/