Best practices for SIP NAT traversal

By Adrian Georgescu

Traversing NAT (Network Address Translation) is one of the issues that hinder SIP communications. Some background about the problem: With an ideal Internet, all devices would be able to communicate end to end without any intermediaries except routers. This assumes each device has a routable IP address, a public reachable Internet identity. In reality, today many of the devices connected on the Internet are using a NAT function present in the border router. While this stops the Internet to initiate connections to the device (bad for IP telephony or other forms of peer-to-peer communications), it also protects the users against malicious attacks. Using NAT, one may also connect multiple devices to the Internet by only using one public IP address. So NAT has both advantages and disadvantages.

Why doesn't SIP work by default behind NAT? The reason is that many of the communication parameters in SIP are transmitted within the SIP message; such parameters include the IP and port numbers used for signaling and media. A SIP device behind NAT does not know much about how it will be seen from the Internet, it only knows its own IP address and the ports where the SIP application runs. Once communication with the Internet starts, the NAT device translates the private IP:port combination of the SIP device connected on the private NAT interface to a temporary mapping of a public IP:port on the interface connected to the Internet.

How can we solve the problem of traversing NAT in both directions for SIP traffic? SIP traffic consists of signaling and media; they usually travel via different paths and use different port numbers. Both need to be dealt with in a separate manner.

SIP signaling

For a SIP device with a private address, in order to be reachable, it must first initiate a connection to the Internet. This can be done using the SIP Register function, a function that maps into the SIP registrar the current location of the SIP device. When a SIP session is initiated, the SIP Registrar is contacted for finding out the actual Internet address of the device. STUN is a standard designed to help SIP devices figure out how they are seen from the Internet. The SIP device might use the values presented by a STUN server (reachable on the public Internet) in the SIP signaling. Unfortunately, depending on the type of NAT different mappings are opened in the NAT device for each new IP address:port combination. This renders the information provided by the STUN server useless for initiating communication to other addresses than the STUN server address. So STUN cannot provide a 100% safe solution to traverse the NAT.

The NAT problem for SIP signaling can be solved by simply implementing a smart registrar which does not save the contact address as presented by the device in the SIP Register message but rather based on the real IP:port combination the message originates from. Once registered with the SIP Registrar, either the phone or the SIP Registrar must maintain the communication channel open by sending keep-alive packets to the SIP device before the binding expires in the NAT device. The packets could be either SIP packets send by the device or IP packets sent by the Registrar. It

is possible depending on the type of NAT that sending keep-alive packets from outside is not enough, in this case the device inside NAT should take care of this by lowering the Registration interval bellow the NAT binding expiration time (usually a value of 85 seconds will suffice). Now, having a permanent communication path open between the SIP Registrar and the SIP device, it is always possible to ring the device behind the NAT and to start negotiating a SIP session.

The only requirement, which luckily is now available in most of the SIP devices, is to use symmetric signaling, that is the device must send and receive data on the same port number.

OpenSER (http://OpenSER.org) is a SIP Registrar implementation that uses the technique described above in order to maintain the NAT binding open with the device. It always works and requires no STUN capability in the SIP devices.

Media

Media consists of one or multiple streams which are negotiated in the SIP signaling. The Media streams may be added or subtracted to the communication set between SIP devices. As this happens dynamically, one must be able to translate in real time the mappings between the internal and public addresses.

Starting with the SIP Invite message, the SIP devices negotiate a common media. The initial negotiation is performed by SDP (Session Description protocol), a protocol used by SIP to convey information about the media streams (address where the media will be received, codec types, bandwidth and others). The problem is the SDP conveys information about the private IP of the SIP device.

There are two ways to solve this issue. One is improving the SIP devices to be able to negotiate dynamically a communication path for the media even after the initial SIP session has been setup. This can be achieved by ICE (Interactive Connection Establishment), which allows devices to probe for multiple paths of communication by trying to use different port numbers and STUN techniques. If ICE support is present in both devices, there is a good chance the devices can start communication end-to-end without any intermediary media relay.

Depending on the NAT type, the communication might not be able to take place even if by using ICE, in this case a media relay must by used. The major disadvantage of using a media relay is that the media will have to travel via a third party location on the Internet, the quality of the call may be affected by a long round trip time. To circumvent the delay problem the media relay must be placed on the shortest path between the two devices. This can be achieved by having multiple media relays connected close to the customers served by a SIP provider. For an enterprise the media relay should always be located at the border of the network, for roaming uses the media relay location should be chosen depending on subscriber geographical address.

TURN is an IETF standard, which implements media relay for SIP end-points. The approach however is not ideal. It assumes the clients have a trust relationship with a TURN server and request session allocation based on shared credentials. This has scalability issues, requires complex changes in the SIP clients, as TURN protocol is

difficult to implement, has no possibility of distributing the load and complicates the configuration of the SIP user agent.

Another approach, which requires no changes in the SIP devices, is to reuse the trust relationship the SIP device already has with the SIP Proxy. In contrast with how TURN works, the SIP Proxy and not the User agent does the session reservation for the media relay. This has the immediate advantage that the SIP UA does not have to have any TURN capability built in and secondary a database with user credentials does not need to be stored on both the TURN server and the client. Another advantage is the fact that the SIP Proxy has always more clues about where is the best place to assign a media relay for a SIP session than the SIP devices themselves. This allows per call allocation of a media relay session in an optimum place on the Internet and solves the load balancing and scalability of the media relay function.

MediaProxy (http://www.ag-projects.com/MediaProxy.html) is a distributed NAT traversal solution based on the above algorithm. It always works and requires no STUN or ICE capability in the SIP devices.

In this document we analyzed how distributing some functions between SIP Proxy, User Agents and Media relays in a scalable and optimal way can elegantly solve the NAT traversal problem. This approach is opposite to what Session Border Controllers offer; a central point where all SIP signaling and media converge before going to the SIP network. Having Session Border Controllers in the path, the end-to-end SIP communication is segmented, difficult and expensive to upgrade for providing more SIP services that will be available in devices.

So we can summarize some of the best practices for solving NAT traversal in SIP:

1. Use of symmetrical signaling and media in SIP devices, is mandatory.
2. Set Register interval to 85 seconds, or less than the NAT binding (this has implications on the server load as registering is an expensive process. It would be much better if clients could send OPTIONS to the server, which is less expensive, or make the server send stateless OPTIONS to the clients which avoids the need for the clients to be modified to support this pinging).
3. Keep alive the NAT binding from the end device, by sending Options or Register at short intervals, this relieves the SIP Registrar of keeping the connections alive (but requires support in all clients).
4. Implement ICE in SIP devices.
5. Do not use STUN servers; they cannot be relied upon in all scenarios.
6. Distribute any media relays geographically, close to the subscribers.
7. Do not use Session Border Controllers for only for solving NAT problem because it breaks end-to-end SIP communication.