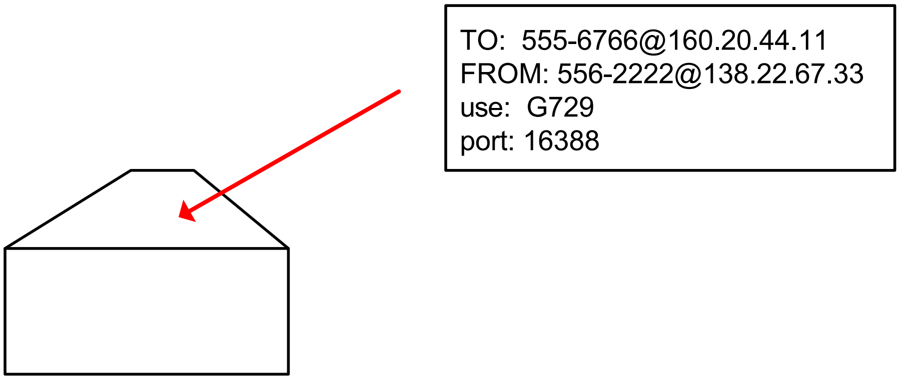
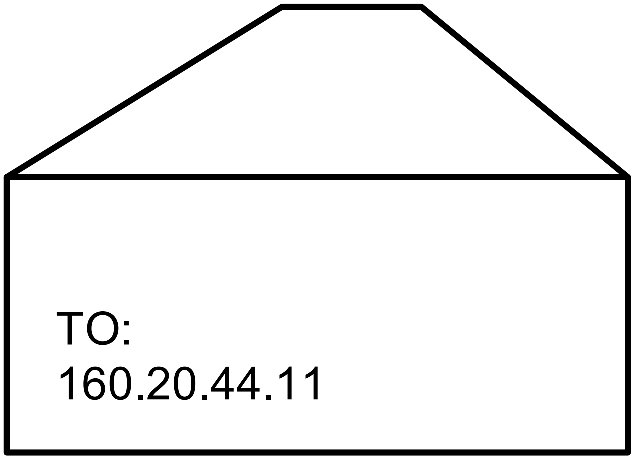
As we have seen in the previous slides, SIP and H323 both use addressing inside their packets to rely information.

Think of an envelope where we place the addresses of telephone/device and call setup information we want to send to the other VoIP user



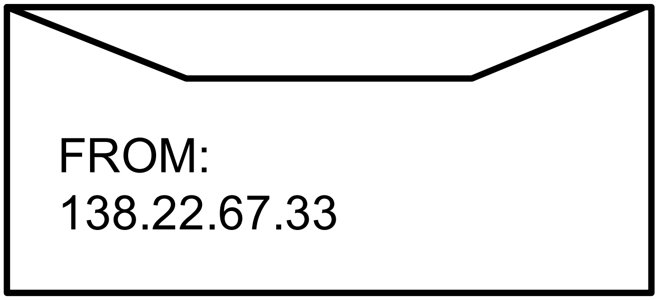
TO: 555-6766@160.20.44.11
FROM: 556-2222@138.22.67.33
use: G729
port: 16388

The envelope is addressed to the other user

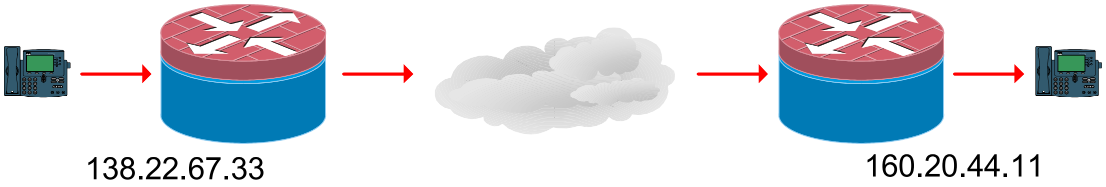


TO:
160.20.44.11

and the return address is on the envelope



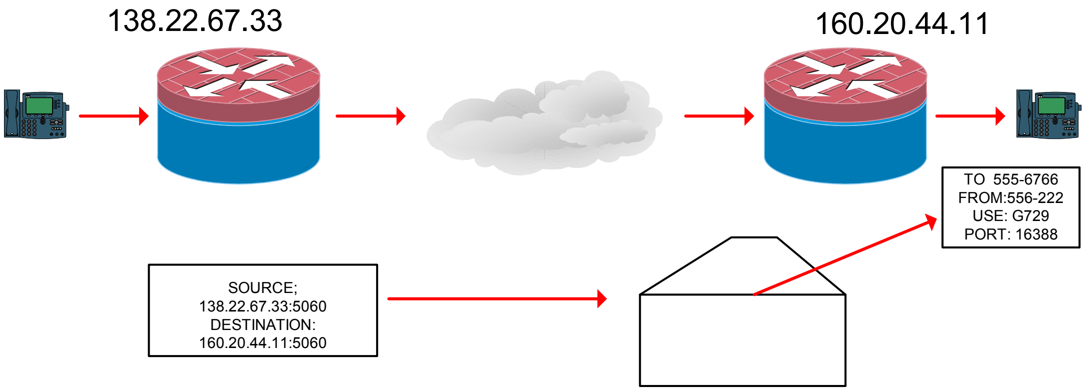
and sent via the routing to the address on the envelope…. the same way an IP packet is constructed with the SIP or H323 content



All works fine as long as the return address on the envelope are the same as the addresses inside the envelope. Both telephones can route traffic to each other.

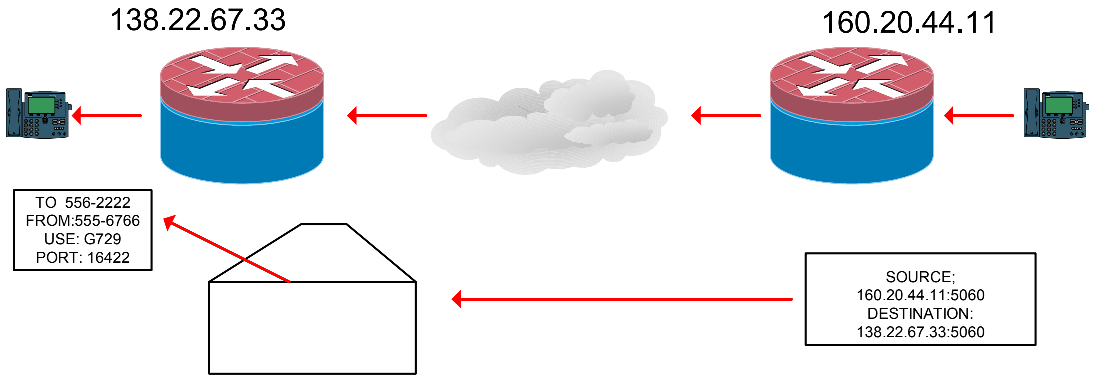
In more detail, for example in SIP the call setup would use port 5060 (at default) so each end knows to open the 5060 envelope and read the contents to setup the audio stream ;

caller sends the envelope to the remote

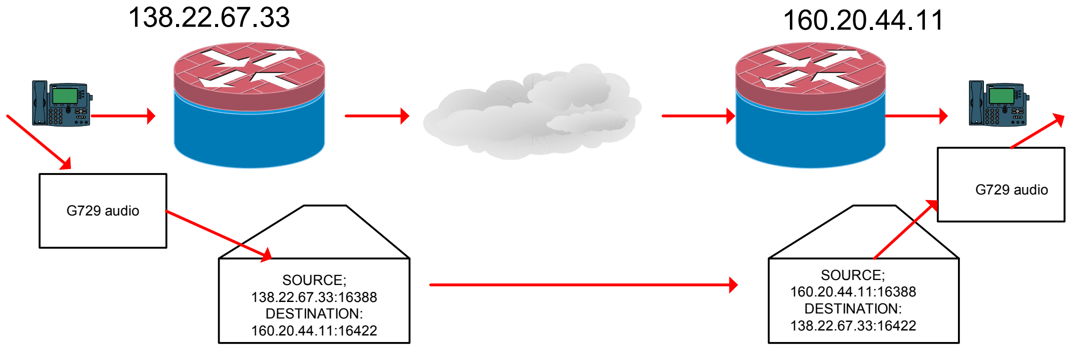


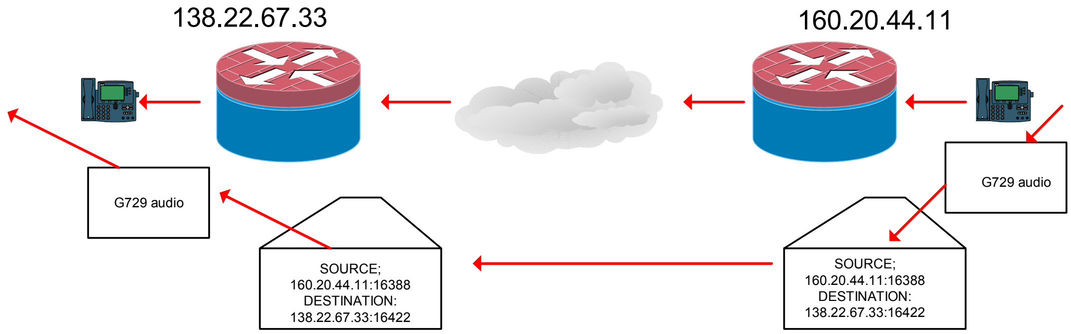
(SIP would send as call setup using port 5060 as default)
(H3232 would send as call setup using port 1720 as default)

remote responds to the caller with its requirements in a 5060 envelope

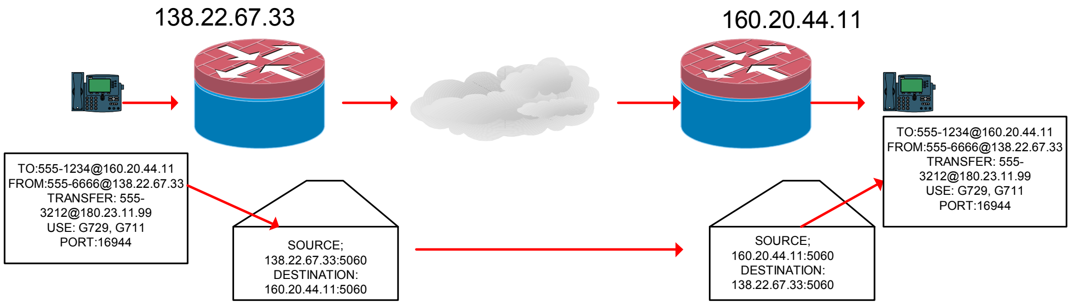


Then the audio stream can begin and the conversation start



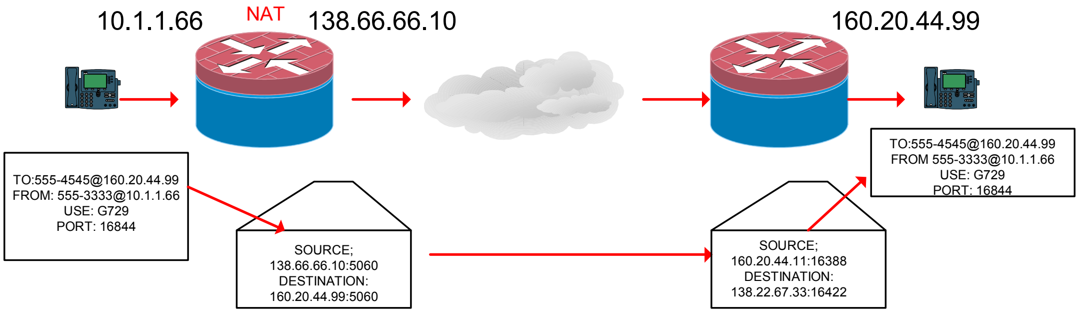


Even for more complex calls with several addresses and options in the envelope…. All works fine, because all the addresses are valid and routable.



Addresses inside the SIP packets are real routable addresses and real port numbers allocated by the telephone systems to carry the audio streams.

Now in the real-world - we have NAT and PAT If we are not using IPv6, we have the problem of limited IPv4 address space and security…. so NAT is used to translate a public address into a private address ;



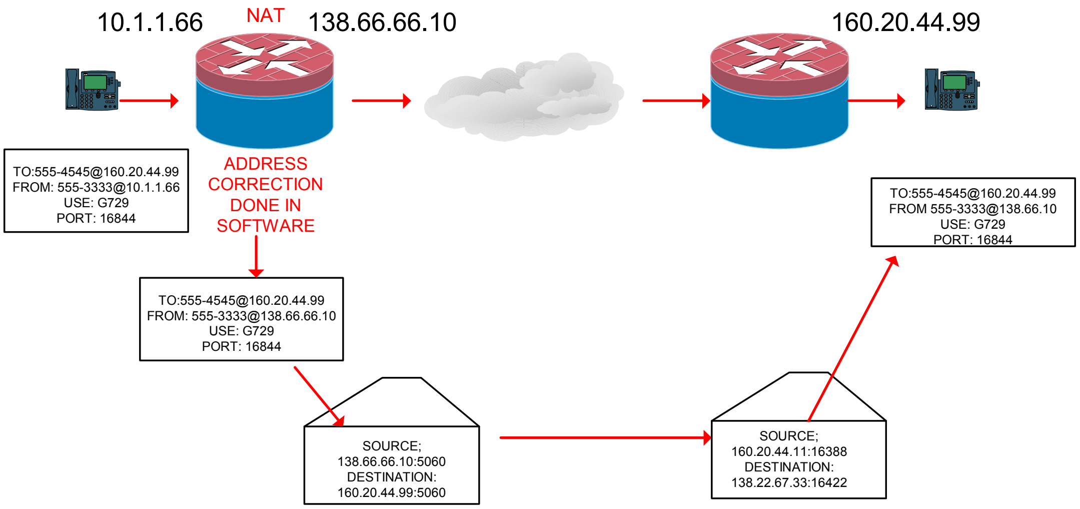
NOTICE that the address on the envelope no longer matches the address inside…. NAT has changed the private 10.1.1.66 address into the outside public address 138.66.66.10 as it forwards the envelope to the remote site.

The remote site can not respond to this request because it does not know where 10.1.1.66 is. Call setup fails.

Simply put ; the address on the envelope is the public address, but the address inside the envelope is the private address. The private address has no meaning outside of the originating site.

When the envelope is opened at the remote end. The requested call setup addresses are private addresses, can not be routed and therefore can not be used…. call setup fails.

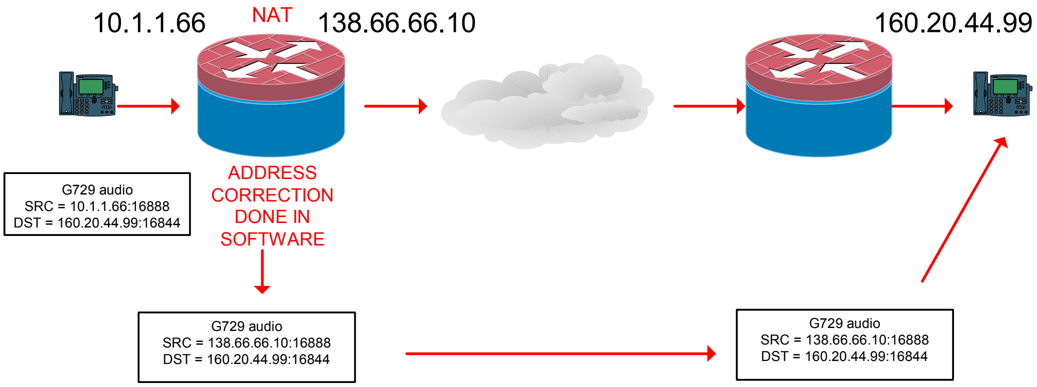
To fix this - there are various methods available, but all result in the public address from the outside of the envelope being copied to the packets inside the envelope…..



Software running in the router, SBC, proxy, gateway or end device uses STUN, ICE or TURN techniques to read the outside 'envelope' address and re-write the internal packet contents replacing the private address with the public address as it is sent out through NAT.

This ensures the remote end receives the envelope with the correctly addressed contents and the call setup can then proceed to the correct addresses.

The same technique has to be applied to the audio stream addresses and port numbers (in each direction) to maintain a translation table and ensure the NAT public to private addresses are re-written in the SIP and RTP audio packets…..

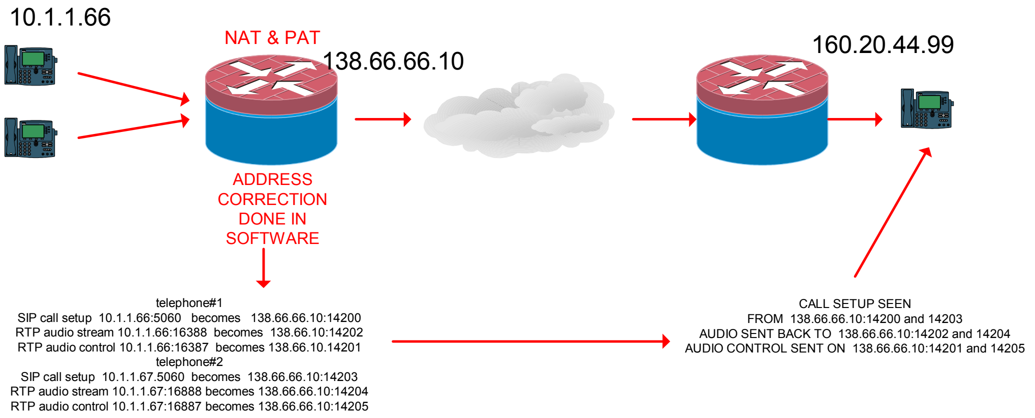


NOW imagine NAT at both ends of the circuit….. twice as many translations, but still an easy solution to handle the problems caused by these one-to-one NAT address translations.

All is working again through the NAT translation….. but using NAT on its own only provides a one-to-one mapping of public to private addressing = we still would need a public address for every device inside our networks…. not scalable not real-world.

Hence PAT = port address translation enables the enterprise to share a public address among the hundreds of private addressed devices in the internal network. So all the external packets have the same public address but use a different port number to correlate them with their original private address and original port number.

PROBLEM for SIP/H323 etc.. once again - the address on the envelope will no longer match the address inside the envelope….. but now a much more complex problem to fix…... We now need to keep a database of NAT and PAT and how they relate to each SIP/H323 call, each voice stream and each control flow (and each video stream).



telephone#1
SIP call setup 10.1.1.66:5060 becomes 138.66.66.10:14200
RTP audio stream 10.1.1.66:16388 becomes 138.66.10:14202
RTP audio control 10.1.1.66:16387 becomes 138.66.10.14201
telephone#2
SIP call setup 10.1.1.67.5060 becomes 138.66.66.10:14203
RTP audio stream 10.1.1.67:16888 becomes 138.66.66.10:14204
RTP audio control 10.1.1.67:16887 becomes 138.66.66.10:14205

CALL SETUP SEEN
FROM 138.66.66.10:14200 and 14203
AUDIO SENT BACK TO 138.66.66.10:14202 and 14204
AUDIO CONTROL SENT ON 138.66.66.10:14201 and 14205

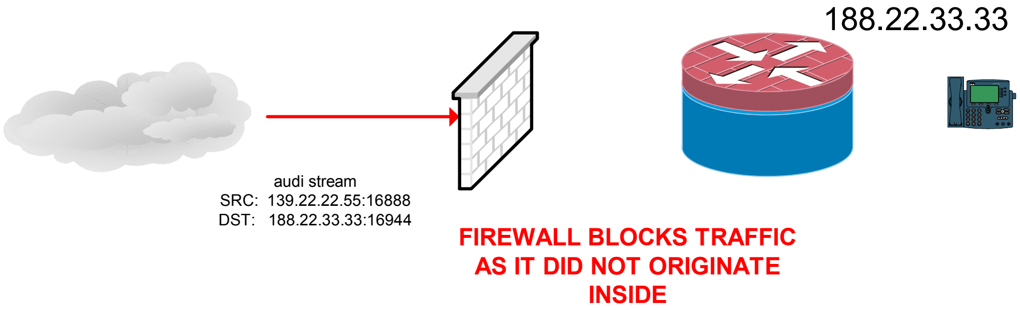
the software has to keep track of which port belongs to which device…. and which audio stream was requested by which end device and make all the necessary changes to the addresses and port numbers inside each envelope…. not too bad until you factor in the features required by the modern telephony systems = transfer, hold, re-direct, conference and three way calling etc…. The software has to keep track of every stream in every flow.

quite complex …. not really we are just beginning ☺

Next - let us factor into the path a firewall or two

One of the jobs of the firewall is to block traffic from the outside that was not originated from the inside.

Here is another problem for our SIP or H323 traffic… In this example our telephone requested RT audio stream to be sent to it on port 16944, the remote telephone requested port 16888.



So call setup functioned correctly, but the audio stream failed because the firewall was not multi-media aware or configured for the audio streams.

The firewall does its job and blocks that audio stream from the remote telephone, because port 16944 was initiated from the outside.

Dirty fix would be to allow the RTP port range through the firewall. This would be an un-secure solution as it opens all the RTP ports to the outside world.

A better solution is to have the intelligence in the firewalls to look into the call setup packets and be SIP and/or H323 aware = For example in SIP - using the SDP packet information to read the required audio ports and open the firewall to those port when the call is made.

Put these scenarios all together an you have a **very basic** understanding of SIP/H323 NAT/PAT and Firewall handling problems you may encounter.

Next we will review the call setup and packet structure of SIP and H323 to understand the call flow setup, redirect, transfer and other features and see how NAT/PAT and firewalls can really screw you up.