[Skip to main content](https://lms.alnafi.com/xblock/block-v1:alnafi+DCCS102+2025_DCCS+type@vertical+block@497afc6cf0714c5b8cb3377ccbc61c7a?exam_access=&recheck_access=1&show_bookmark=0&show_title=0&view=student_view" \l "main)

Ref <https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/>

**How VoIP apps like Skype, WhatsApp, and Facebook Messenger work**

* [Adeolu Owokade](https://www.routerfreak.com/author/adeolu/)

Share on facebook Share on twitter Share on linkedin Share on whatsapp Share on telegram

TABLE OF CONTENTS

* [Voice over Internet Protocol (VoIP)](https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/#elementor-toc__heading-anchor-0)
  + [How does VoIP work?](https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/#elementor-toc__heading-anchor-1)
  + [VoIP on the Public Internet](https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/#elementor-toc__heading-anchor-2)
  + [Case study: Skype Voice Call](https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/#elementor-toc__heading-anchor-3)
* [Challenges of VoIP over the Public Internet](https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/#elementor-toc__heading-anchor-4)
  + [Quality of Voice call](https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/#elementor-toc__heading-anchor-5)
  + [ISP/Telecoms blocking or rate limiting](https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/#elementor-toc__heading-anchor-6)
* [Conclusion](https://www.routerfreak.com/how-voip-apps-skype-whatsapp-facebook-messenger-work/#elementor-toc__heading-anchor-7)

I went on holiday recently and visited another country 3000 miles from mine. Of course, this means I had long left the reach of my mobile network operator. Due to roaming and partnerships between mobile network operators in different countries, I had the option of making and receiving voip calls using my home phone number without having to buy a new SIM card in the destination country. However, I had no intention of doing that. Roaming rates are ridiculously expensive and in my case, **up to 15 times** the normal rate I pay for calls when at home!

In the not so distant past, I did not necessarily have a choice apart from roaming or just not receiving/making calls when on holiday. But not in today’s world. With [WhatsApp](https://www.whatsapp.com/) installed on my smartphone and [over 1.5 billion people using it monthly](https://techcrunch.com/2018/01/31/whatsapp-hits-1-5-billion-monthly-users-19b-not-so-bad/) around the world, I did not have a problem reaching people as long as I was connected to the Internet (via Wi-Fi). For those who may not be familiar with this app, it is basically a messaging and voice app like [Skype](https://www.skype.com/en/home/) or [Facebook Messenger](https://www.messenger.com/) except that all you need to connect is your mobile phone number (no username or email address required).

**Side talk**: I still find it weird making professional/formal calls using WhatsApp because I consider it a Social app. Nevertheless, the lines are increasingly getting blurred as so many clients even reach out to me via WhatsApp.

In this article, we are going to discuss how apps like WhatsApp, Skype, and Facebook Messenger that allow you to make voice calls over an (Internet) data connection work, the challenges that such apps face, and how some of those challenges are overcome.

**Voice over Internet Protocol (VoIP)**

If you have been in the Networking industry for even a little while, then you have probably heard about networks that support both data and voice traffic. For example, instead of the old way of laying phone cables to create a dedicated voice network inside an organization, the networks of today are now built to carry all kinds of traffic including data and voice. This has brought about the term “VoIP” (i.e. Voice over IP) because voice (and other forms of multimedia) can now be delivered over an IP packet switched network. So instead of buying traditional phone systems, companies now buy IP phones which can be connected to the network either via Wi-Fi or using a LAN (RJ-45) cable.

While many network engineers are familiar with VoIP on internal networks, they have not given much thought to the fact that VoIP is also used outside internal networks – the Internet being the most common example. How else will you classify the call feature in WhatsApp, Facebook Messenger, Skype, Telegram, etc.? Of course these apps do more than just VoIP, but their call features fall squarely under the VoIP category.

**Note**: Even I never considered these apps as offering VoIP capability so no judgment.

**How does VoIP work?**

So how does VoIP work? In VoIP, there are two main parts that need to be solved:

1. Setting up the call i.e. Call signaling.
2. Carrying the actual voice traffic back and forth between the caller and the receiver.

Call signaling basically means establishing the connection between the caller and the receiver. You can think of this as something similar to the TCP three-way handshake. Another example if you wish is how the bike riders in a president’s motorcade clear the way for the president’s car but do not actually carry the president on their bikes. There are several well-known call signaling protocols like Session Initiation Protocol ([SIP](https://tools.ietf.org/html/rfc3261)) and [H.323](https://blogs.technet.microsoft.com/meacoex/2011/07/11/sip-vs-h-323/).

When the call has been set up, a different set of protocols are then used to carry the actual voice data between the caller and receiver. The most famous protocol used for this function is the Real-time Transport Protocol (RTP) which runs over UDP.

**Note**: It makes sense to use UDP instead of TCP to carry voice packets because of speed. You don’t want to send a voice packet and wait for the receiver to first acknowledge.

Another important part of a VoIP implementation is how the speaker’s voice is converted to a digital signal, compressed, and transmitted over the network. On the other end, it is converted back from the digital form to analog. This is made possible by the use of **Codecs** (Coder-Decoder). Common [voice codecs](https://www.lifewire.com/voip-codecs-3426728) include G.711 and G.729.

**VoIP on the Public Internet**

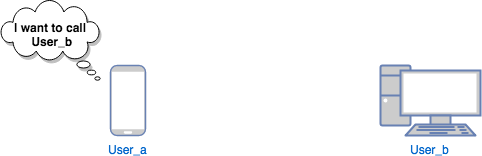
VoIP apps using the public Internet as their base of operation also work similarly to what we have described above. In terms of Call signaling, VoIP apps like Skype and WhatsApp use their own proprietary protocols. For example, WhatsApp previously used a [version](http://www.fit.vutbr.cz/research/pubs/index.php?file=%2Fpub%2F10979%2FWhatsApp.pdf&id=10979) of the Extensible Messaging and Presence Protocol (XMPP), but it [seems they have moved](https://news.ycombinator.com/item?id=13219925) to their own protocol now. Most of them also use some form of RTP/UDP to carry the voice packets. In terms of voice codes, the last known voice codec used by Skype is called [SILK](https://tools.ietf.org/html/draft-vos-silk-01) while WhatsApp and Facebook Messenger use [Opus](https://opus-codec.org/), a variant of SILK.

**Note**: Most of these VoIP apps like Skype and WhatsApp use encryption to protect communication and this makes it difficult to really probe into the protocols in use.

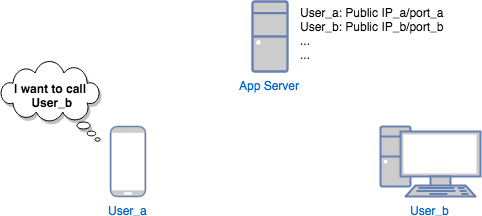
However, with apps working over the Internet, there are additional challenges that must be solved:

1. How will calls be set up between users that can be in any location in the world?
2. Many users of these apps connect to the Internet via a private network (Wi-Fi or 3G/4G) which means that Network Address Translation (NAT) is probably in use. Since communication over the Internet requires a public IP address, how will two-way communication be set up between the calling parties?

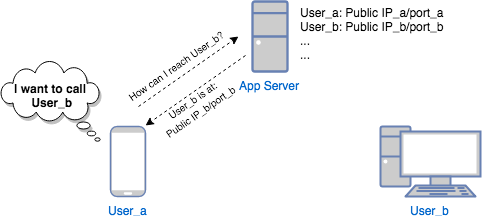
We can use a scenario to explain how these apps solve both challenges. Imagine a User\_a wants to make a VoIP call to User\_b using WhatsApp/Skype/Messenger. User\_a only knows the username/phone number of User\_b but it doesn’t know her public IP address.



This means there needs to be a central repository that stores the mapping between username/phone number to the public IP address on which that user was seen. I can assume that this list will need to be constantly updated as mobile users move around a lot.



So in the simplest case, User\_a will reach out to one of the servers of the app he is using asking for the public IP address of User\_b. The app will check its database for this information and send back to User\_a. User\_a can then use this information to open a session to User\_b (assuming a peer-to-peer model).



Of course, the scenario described above is overly simplistic. Most devices are behind a network performing NAT which means two things:

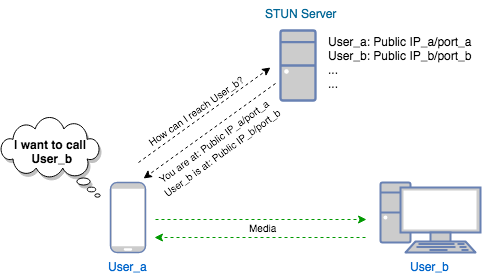
1. Those devices do not even know ***their own*** public IP addresses.
2. Even if problem 1 is solved, they may not be able to initiate a connection to the other user (e.g. firewall blocking).

The most common form of [NAT](https://www.routerfreak.com/nat-cisco-asa/) for users behind a private network that need to access the Internet is something called Port Address Translation (PAT). In this form of PAT, the private IP address and source port of a connection are translated to a public IP address and port on the Internet.

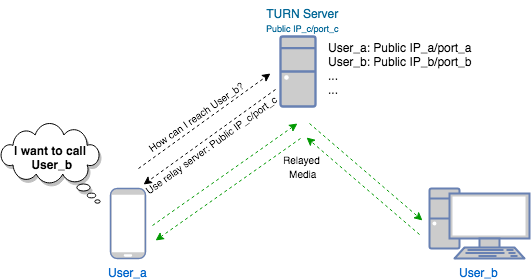
In most cases, a user device does not need to worry about how or where the PAT is done – it just needs to send packets and the NAT device will handle translation. However, in a case where peer to peer (P2P) connection has to be established (like Skype), the user device must be able to inform the other peer on which IP address/port with which to terminate the connection.

Since you cannot give what you don’t know (and giving your private IP/port is useless to the other peer), the device must be able to figure out what public IP and port are being used for the connection. To achieve this, we turn to something called **STUN (Session Traversal Utilities for NAT)**. The concept is simple – the device will open a connection to the app’s STUN server, and that server will reply the device with the IP address and port which the connection came on.

In most cases, the STUN server will also reply with the public IP/port of the other peer that you want to call. With this information, the calling party can now send a message directly to the receiver and set up the call. This is how Skype actually works.



How about if the calling party tries to initiate a connection to the receiver and a firewall blocks that connection? Or say the NAT on the peer’s side does not allow a connection to be made reliably to that peer (e.g. symmetric NAT where the ports keep changing randomly)? In cases like this, we have to fall back to **TURN, Traversal Using Relay NAT**. Basically, each peer establishes a connection to the TURN server and the server **relays** the voice packets between the peers. From this [research paper](http://www.fit.vutbr.cz/research/pubs/index.php?file=%2Fpub%2F10979%2FWhatsApp.pdf&id=10979) written in 2015, it seems WhatsApp was using the TURN method.

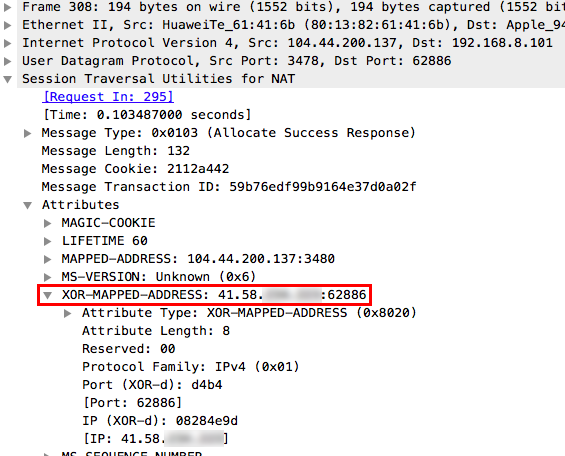


**Note**: While it is not exactly clear what technology Facebook Messenger is currently running, the same concepts described above apply.

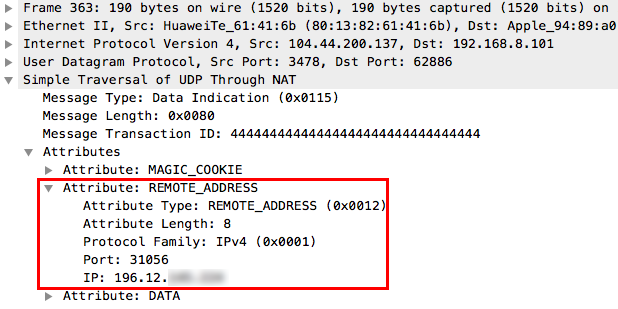
**Case study: Skype Voice Call**

Skype is a P2P application that relies on the STUN server scenario we described above. I made a Skype call on my laptop while capturing packets using Wireshark. Although many of the packets were encrypted, there is a lot we can gather from the capture.

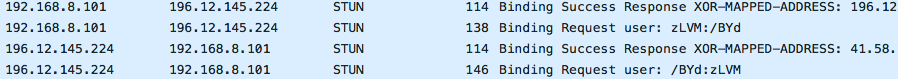
First of all, there is a packet between my laptop and a STUN server (104.44.200.137) where the STUN server tells my device its public IP address and port.



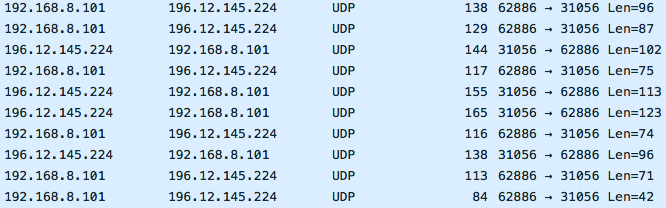
I also have a packet where the STUN server tells me about the public IP/port of the user I wanted to call:



After a couple of messages, the peers now established a connection directly to each other and exchanged some messages:



And after this, the fun part: voice packets following between the peers. Of course, these packets were encrypted but I can only assume (fairly confidently) that those were the voice packets:



**Challenges of VoIP over the Public Internet**

Using VoIP apps over the Internet is not all rosy and there are many issues associated with using such tools. We will look at two of such challenges here and also discuss how these services go about overcoming those challenges.

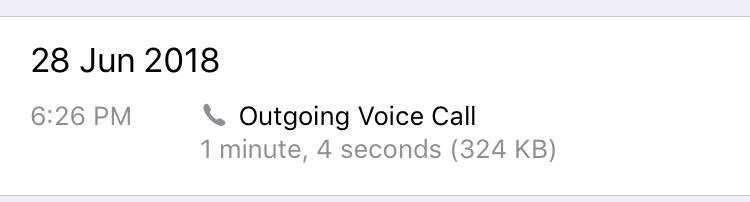
**Quality of Voice call**

While voice packets are very small in size (meaning bandwidth is not usually that much of a huge problem), they do not perform well in networks where delay, latency, jitter, and packet loss are huge. This is why there may be a lag when you are on one of such calls or as WhatsApp users may be familiar with, you get the “Reconnecting” message.

On an internal network, network administrators will usually implement some form of Quality of Service (QoS) such that voice packets are given preference on the network. Unfortunately, the network on which these apps run (the Internet) is under the control of a third-party and there is usually no guarantee of quality.

To put it frankly, there is not much the app creators like Skype and WhatsApp can do in terms of delay and latency. However, they can optimize for bandwidth and possibly packet loss. By compressing the voice packets such that they the packets are not large in size, you can provide a fairly good call quality on low-speed networks (e.g. 2G) and also reduce the chances that the packets will be dropped due to congestion.

From my personal experience with WhatsApp, it uses an average of 300KB of data for every minute of call (without the “Low Data Usage” option turned on). Skype and Google Duo seem to [use more data per minute](https://mybroadband.co.za/news/cellular/266231-voice-calling-price-comparison-whatsapp-vs-facebook-vs-skype.html).



**ISP/Telecoms blocking or rate limiting**

Let me paint a picture for you: I can get 1.5GB of data for about $3 from my mobile network operator. Assuming I use the entire data package to make WhatsApp calls, it means I can talk for over 5000 minutes. On the other hand, the same amount of credit will only give me about 120 minutes of regular phone calls on the same network and this is without international calls!

As you can see, there is more incentive for me to use WhatsApp and other VoIP services than to make regular calls and that equates to [lost revenue](https://www.researchgate.net/publication/276175550_Impact_of_Over_the_Top_OTT_Services_on_Telecom_Service_Providers) for the telecoms service providers. For example, [AT&T blocked Skype on iPhone](https://www.pcworld.com/article/162864/skype_iphone.html?tk=rel_news) back in 2009 and Skype on iPhone could only work with Wi-Fi. In more recent times, telecoms service providers in Nigeria [contemplated blocking these services](http://nigeriacommunicationsweek.com.ng/telecom/telcos-may-block-skype-whatsapp-calls-over-haemorrhaging-revenue) because it was eating deep into their revenue. In fact, [some countries like UAE explicitly block these services](https://www.khaleejtimes.com/technology/no-whatsapp-calls-in-the-uae-authority-clarifies).

Apart from Telecoms service providers fighting to keep their revenue from falling, these service providers may also have to comply with Government rules that require them to block access to these VoIP services from time to time. For example, [Brazil has banned WhatsApp several times](https://qz.com/675070/the-messaging-app-thats-benefiting-from-whatsapps-ban-in-brazil/) because the authorities say WhatsApp did not provide them with data needed for a criminal investigation. The plus side of this ban was that more users downloaded and started using Telegram, a WhatsApp alternative.

Unfortunately, there is not much these VoIP apps can do about this issue apart from hope (and lobby) for favorable policies. The [recent Net Neutrality](https://www.cnet.com/news/net-neutrality-is-now-really-officially-dead-open-internet-congress-now-what/) repeal did not go well in favor of such services since a service provider can now look into the type of traffic and decide to slow it down or charge more.

**Note**: It is believed that many ISPs slow down P2P traffic such as torrent applications so I’m not sure they were entirely keeping to Net Neutrality before this repeal.

**Conclusion**

This brings us to the end of this article where we have looked VoIP and how it works especially on the Internet. We have demystified the voice call features of apps like Skype and WhatsApp (even though we couldn’t see into all the packets because of encryption). We have also discussed challenges with running these apps over a public network like the Internet and see how some of these challenges can be combated.