WhatsApp Calling: a Revised Analysis on WhatsApp's Architecture and Calling Service

Juan Martín Vanerio
Universidad de la República, Uruguay
jvanerio@fing.edu.uy

Pedro Casas
AIT Austrian Institute of Technology
pedro.casas@ait.ac.at

ABSTRACT

The use of instant messaging applications in mobile networks has largely increased in recent years, replacing traditional messaging applications (SMS, MMS). WhatsApp is the application of this kind with the greatest market share worldwide; as a consequence, mobile operators are becoming growingly interested in understanding its underlying functioning and growth. In this paper we revise some of the results presented in our previous work on WhatsApp characterization, and extend this analysis to the new WhatsApp calling service. We study how the underlying WhatsApp architecture has changed after its acquisition by Facebook in early 2014, and shed some initial light on the new calling functionality, active since mid 2015. While the main backbone of the WhatsApp service is still hosted by the same cloud provider – SoftLayer, we uncover the usage of Facebook servers to support and implement the calling service.

Keywords

WhatsApp; Calling Service; Network Characterization.

The main author would like to thank Pablo Cuello and Gonzalo Stillo for their contribution to this work.

1. INTRODUCTION

As of February 2016, WhatsApp had approximately one billion users worldwide, and the number of users keeps rapidly growing, specially in developing countries.

The number of WhatsApp messages processed daily exceeds by 50% the total of SMS messages being processed in mobile networks worldwide, and the gap is growing. Mobile operators providing traditional services are affected not only by lower revenues from messaging – and now from voice calls, but also by the changes introduced by the WhatsApp service itself, both in terms of data communication patterns and network radio resources [1,2]. In [2] we presented the first large-scale characterization of the WhatsApp service, targeting both the generated traffic as well as the underlying network infrastructure providing the service. This analysis was conducted through passive measurements collected on a mobile ISP in February 2014, when the number of users was half of todays, and while the company was still not part of Facebook's acquisitions. In addition, the WhatsApp calling service was still not active at the time of that study, thus no characterization of this new functionality is currently available today.

In this paper we revise part of the results presented in [2], relying on active measurements collected one and a half years later. In particular, we study the evolution of the underlying network hosting infrastructure, as well as provide some first results on the characterization of WhatsApp Calling. Similar to [2], our study is based both on controlled lab measurements, as well as on distributed active measurements, performed through a distributed, publicly available active measurements platform called *abongo*[1]. The main results of this analysis as compared to those presented in [2] are summarized as follows:**(i)** The addressing and naming schemes used by WhatsApp to operate the control, text, and multimedia sharing traffic remain unchanged.**(ii)** While the number of identified server IP addresses providing WhatsApp traffic has increased by more than 80%, the WhatsApp text and multimedia sharing still remains a fully centralized service, hosted by the cloud provider SoftLayer at servers located in the US.**(iii)** Facebook servers and Points of Presence (PoPs) distributed worldwide are employed to implement the WhatsApp calling service. This implies a break in the policy of WhatsApp to remain an independent platform, even after being acquired by Facebook.

[1] http://www.abongo.com/

domain whatsapp.net			
sub-domain	**protocol**	**app.**	**IPs**
cX|dX|eX	XMPP (5222/443)	control/text	217
mmiXYZ|mmsXYZ	HTTPS (443)	photo/audio	439
mmvXYZ	HTTPS (443)	video	195
mmx	HTTPS (443)	other	13
cdn|crashlog|etc.	HTTPS (443)	other	18

Table 1: Third level domain names used by `whatsapp.net` and applications.

2. METHODOLOGY

To study the network architecture and addressing schemes used by WhatsApp, we rely on a combination of different measurement techniques. Firstly, given that WhatsApp uses encrypted communications, we rely on the manual inspection of actively generated WhatsApp traffic at end devices, observing the resulting traffic at an intermediate gateway. Given the heavy usage of DNS traffic to address services in WhatsApp [2], we especially focus on the analysis of every DNS request generated by the devices. Similar to [2], our measurements revealed that WhatsApp servers are still associated to the same domain name `whatsapp.net`. The set of Fully Qualified Domain Names (FQDNs) obtained by generating different types of traffic form the basis of the dataset used in the subsequent analysis. The list of FQDNs generated by WhatsApp is complemented with other FQDNs found in public lists of different nature, e.g., from network security lists such as [3]. Secondly, based on the complete list of identified FQDNs, we resolve the IP addresses corresponding to each of the observed FQDNs, using both local DNS resolvers (vantage point is located in Montevideo, Uruguay) as well as DNS resolutions performed from 11 worldwide distributed probes, using the *abongo* distributed platform. The latter is performed to avoid biased results introduced by geolocalization of the requester. To double check that the obtained IP addresses belong to WhatsApp, we verify that the reverse DNS resolutions point to the `whatsapp.net` domain. For each verified IP we then determine the AS number – using the RIPE IP2AS service, as well as the number of hops and RTT from each of the 12 vantage points (1 local + 11 distributed). We use these measurements to improve the geo-localization of the obtained IP addresses, by triangulation [4].

3. ADDRESSING & BACKBONE

We verified that WhatsApp still uses exactly the same addressing and naming scheme reported in [2] for control, text and multimedia-exchange traffic - we shall refer to these services as the “standard” WhatsApp services. All WhatsApp communications are directed against servers in the `whatsapp.net` domain. Tab. 1 reports the observed third level domain names used by `whatsapp.net`, the communication protocol and the corresponding applications. As reported in [2], there are dedicated servers for control and text messages, as well as for each of the specific types of multimedia sharing (photos, audio recordings and videos). There are also other servers with specific functionality for logging application problems (platform-crash) and other unidentified functionalities (e.g., cdn, bin-short, etc.). When a text message is sent, only cX|dX|eX group servers participate; when multimedia is sent, the same group is involved, in conjunction with mmsXYZ and mmiXYZ group servers for photos and audio, and mmvXYZ for video. We also verified that cX|dX|eX servers perform all the registry functions for users, ultimately becoming the main control servers.

# hops	# IPs	min RTT (ms)
11	108	161
13	320	158
14	154	158

Table 2: RTTs and number of hops toward SoftLayer IPs (vantage point is in Uruguay).

As for the WhatsApp calling service, we observed that signaling is encrypted and transmitted via the main control channels, cX|dX|eX. As part of the signaling, IP addresses of Interactive Connectivity Establishment (ICE)-like servers are observed, which are then used by the end devices to establish point-to-point, real-time voice communications.

We also analyzed the underlying network infrastructure. We identified 700 different IP addresses for WhatsApp servers, which represents an increase by more than 80% w.r.t. the 386 server IP addresses found in [2]. This indicates that the service continues to growth in the number of IP addresses dedicated to its operation. Tab. 1 additionally shows the number of server IPs observed for each type of application (note that some IPs are used by more than one single service). From the local vantage point located in Uruguay we could only observe 563 of those 700 IP addresses (i.e., only 80% of the servers), suggesting that some kind of geolocalization-aware mechanism could be in place. This was not the case in [2], where the same set of IPs was obtained, regardless of the geographical location of the requester. We are digging deeper into this as part of our ongoing work, as the number of vantage points we used was relatively small as compared to [2]. Almost all of the observed IPs belong to the same AS 36351 – SoftLayer, as previously identified in [2]. However, IPs related to the WhatsApp calling service belong to another AS – AS 32934, belonging to Facebook. This is a major difference w.r.t. [2] and as compared to the situation before WhatsApp acquisition by Facebook; indeed, it shows how Facebook relies on its own overlay network to enhance the performance of WhatsApp, particularly for the case of real time communications. To assess the location of the SoftLayer IPs, we rely on both geo-localization data – using RIPE Geolocation Finder (https://apps.db.ripe.net/geolocation-finder) and Max-

Mind (https://www.maxmind.com)) services, and active traceroute and ping measurements, performed from the local vantage point. IPs are located in data-centers at exactly the same locations as they were before in [2], i.e., Dallas and Houston, in the US. To further verify this information, Tab. 2 reports the min RTTs to some of these IPs, grouped by number of hops. The minimum RTT reflects the propagation delay, thus it provides a good indication of where servers could be located. Relying once again in geo-location services, we verified that the obtained results perfectly match the latencies observed to other servers located in the same cities, confirming that WhatsApp standard services are still fully centralized at the US.

WhatsApp Calling IPs hosted by Facebook servers are distributed in several places around the world, using Facebook's global peering and distributed overlay platform, with several PoPs distributed worldwide [5].

4. WHATSAPP CALLING SERVICE

The WhatsApp calling service was announced during the first quarter of 2014, but became fully active for Android OS users in April 2015. According to our traffic captures, the service is implemented with a communication technique similar to that used by the conventional Interactive Connectivity Establishment (ICE - RFC 5245) protocol to establish a communication channel between terminals, which might be located behind gateways, therefore requiring NAT translation and address discovering. ICE uses the Session Traversal Utilities for NAT (STUN - RFC 5389) protocol and its extensions. In a nutshell, ICE allows a terminal to discover and communicate its public IP so that it can be reached by others, establishing a real-time communication channel over the shortest path between endpoints.

Differently from the standard services, WhatsApp calling does not rely on DNS to obtain the server IPs to contact. The information required for call setup is encrypted and sent through WhatsApp control channels (i.e., cX|dX|eX servers). In particular, the public IP addresses of the voice call servers – hosted by Facebook, are communicated to the terminals. To establish a call, the calling terminal contacts its current WhatsApp control server through it persistent control channel, which answers with a set of potential public IPs to be further contacted by the caller, and forwards the request internally, through the WhatsApp system. The called terminal receives then the request and the set of public IPs to be used from its current control server. Immediately after, both terminals attempt to establish communication simultaneously with eight different public IP addresses, using the STUN protocol (UDP/3478). We shall refer to these IPs as the "STUN servers". At last, the calling terminal establishes communication with one of the responding STUN servers, which in turn is the one that handles the end-to-end communication establishment between both terminals, through standard RTP and RTCP protocols. Even after successful binding of the two terminals, both endpoints keep trying new connections to possibly achieve shorter paths, as well as to adapt to quality and routing changes.

Using the traffic collected at the local vantage point we managed to identify 17 different Facebook IPss used as STUN servers. Three of these are used in every call establishment (`185.60.216|217|218.2`), whereas the remaining five always vary, probably due to load-balancing. We performed reverse DNS queries on these IPs to obtain further information, and discovered that the resolved domain names include an IATA airport code, indicating the city where each of the servers is located. Interestingly, the three aforementioned server IPs do not have PTR DNS records – we are further exploring the location of the corresponding servers, which seems to be somewhere in Brazil, according to RTTs. IP addresses with PTR DNS records include `31.13.XX.48`, `173.252.114|121.1` and `179.60.192.48` and its FQDN has the form edgeray-shv-01-IATA#.facebook.com. The IATA code indicates the hosting city and # the specific server within the city. Servers are located in cities such as Amsterdam, Los Angeles, Atlanta. Vienna, Milan, Paris and Tokyo among others. To further verify that the cities obtained by the IATA codes are accurate, we performed traceroutes and RTT-based tests from public looking glass servers located in the cities determined by the IATA codes. We confirmed that the location is correct for almost all of the servers. An exceptional case is represented by two servers which should be located in Brazil (according to the IATA codes), but expose a much higher latency than the one expected – around 170 ms instead of 65 ms. Further analysis based on RTT triangulation suggest that both servers with IPs `173.252.114|121.1` are located in southeastern US, possibly near Atlanta. The identified cities are consistent with PoPs in which Facebook provides peering agreements [5]. In terms of traffic, it consists of two unidirectional flows, each of 50 pps with slightly variable bit rate and payload size, at around 45kbps (Ethernet), equivalent to 28.2kbps level audio coding. Mean RTP packet size is 112 bytes, meaning a 58 bytes payload. There are several possible codecs that can achieve these rates, such as G.722.2 or Opus, although a dynamic and unidentified codec is being used at the RTP level. Further analysis is required to determine exactly which sort of codecs are being used.

5. REFERENCES

[1] A. Aucinas et al., "Staying Online While Mobile: The Hidden Costs", in *CoNEXT*, 2013.

[2] P. Fiadino et al., "Vivisecting WhatsApp in Cellular Networks: Servers, Flows, and Quality of Experience", in *TMA*, 2015.

[3] Whatsapp Servers List identified by VirusTotal.com, in https://www.virustotal.com/en/domain/sro.whatsapp.net.

[4] E. Katz-Bassett et al., "Towards IP Geolocation Using Delay and Topology Measurements", in *IMC*, 2006.

[5] PeeringDB, in https://www.peeringdb.com.