Traffic Characterization of Instant Messaging Apps: A Campus-Level View

Sina Keshvadi
University of Calgary
Calgary, AB, Canada
sina.keshvadi1@ucalgary.ca

Mehdi Karamollahi
University of Calgary
Calgary, AB, Canada
mehdi.karamollahi@ucalgary.ca

Carey Williamson
University of Calgary
Calgary, AB, Canada
carey@cpsc.ucalgary.ca

***Abstract*—Over the past decade, Instant Messaging (IM) apps have become an extremely popular tool for billions of people to communicate online. In this paper, we use a combination of active and passive measurement techniques to study one week of IM app traffic on a large campus edge network. Despite the challenges of end-to-end encryption, user privacy, NAT, DHCP, and high traffic volumes, we identify the key characteristics of four popular IM apps: Facebook Messenger, Google Hangouts, Snapchat, and WeChat. The main observations from our study indicate a rich ecosystem of IM apps, many of which exhibit strong diurnal patterns, complex user interactions, and heavy-tailed distributions for connection durations and transfer sizes. Collectively, these four IM apps contribute about 650 GB of daily traffic volume on our campus network.**

I. Introduction

Hand-held mobile devices have experienced tremendous growth in popularity in recent years, and this growth is expected to continue in the future [3]. Smartphones and tablets have improved in computational power, memory, display, and connectivity. Unlike traditional cell phones, they run modern operating systems that enable them to support many different mobile applications (*apps*). This evolution has made mobile functionality comparable to that offered by desktop computers or laptops. Cisco predicted that mobile Internet traffic would grow at a Compound Annual Growth Rate (CAGR) of 46% from 2017 to 2022, and exceed 77% of overall Internet traffic by 2022 [3].

The popularity of mobile devices along with Instant Messaging (IM) apps have changed the way that people interact and communicate. In 2012, IM apps overtook the Short Message Service (SMS) operated by cellular network carriers in terms of messages sent per day [2]. Recent years have witnessed a fast-growing trend of using mobile IM apps such as WhatsApp, Messenger, and WeChat, to the extent that more and more human interactions now take place in the digital world. With such a large user base at stake, IM apps have added many new features to their services to provide a competitive edge. For instance, WeChat offers video messaging, online payments, localization services, game playing, and more.

Although IM apps have gained immense popularity, there have been relatively few network traffic characterization studies of these apps. One reason might be the difficulties in collecting and analyzing the traffic of IM apps, due to end-to-end encryption, customized application-layer protocols, user privacy issues, and rapidly changing features. Another reason could be the mistaken assumption that the traffic volume generated by these apps is relatively small [21]. A systematic study of IM apps could help inform the design, implementation, and performance optimization of future network-based systems.

In this paper, we study IM app traffic on a large campus edge network. Our work is motivated by the capabilities of modern smartphones, their ubiquitous network connectivity, and the increased popularity of new IM apps for sharing large files and multimedia content.

In this study, we focus on characterizing the network traffic from popular IM apps, as viewed from an edge router of the University of Calgary campus network. Our campus community consists of over 35,000 users, including undergraduate and graduate students, faculty, and staff.

The main aim of our study is to find answers to the following questions:

- How much of the campus network traffic is generated from IM apps?
- What are the key characteristics of IM app traffic?
- How are these IM apps similar to and/or different from each other?
- What are the potential performance implications of these apps on an enterprise network?

For our study, we collected campus-level IM traffic for a one-week period (October 7-13, 2019). This observational period from a recent busy semester gives us sufficient data to study hourly, daily, and weekly patterns of IM app traffic, as well as IM usage patterns and heavy-tailed transfer sizes.

The results from our traffic characterization study in a campus environment are of potential value to app developers, network operators, service providers, and protocol designers. App developers can gain insights into actual IM usage on the Internet, and network operators can plan for resource allocation in future networks. Service providers can gain a glimpse of possible future demands of IM app usage, and improve the quality of their services. Finally, protocol designers can see the technical details of app usage for improving current protocols, or designing new ones.

The main contributions of this paper are as follows:

- We describe our active measurement methodology to identify IM app traffic from a single smartphone under test, even when the traffic is encrypted.
- We present our passive measurement methodology for large-scale data collection from a campus edge network.
- We analyze and characterize the traffic generated by IM apps in a large-scale enterprise network.

The rest of this paper is organized as follows. We provide background on IM apps in Section II. Section III describes our active and passive methodologies for data collection and analysis. We present the workload characterization results in Section IV. Section V discusses prior related work. Finally, Section VI concludes the paper.

II. Background

Instant Messaging (IM) is a technology that supports real-time text transmission over the Internet. By 2010, users moved away from IM Web sites to "messaging apps". The recent IM apps provide more advanced features such as transferring documents, files, locations, clickable hyperlinks, games, banking, VoIP/video calls, etc. In the last decade, IM apps have gained tremendous popularity, such that fewer people use text messages anymore.

IM apps have become increasingly popular since they are simple, free, real-time, reliable, and support multitasking. It is estimated that there are billions of IM users who use IM for many activities: free worldwide texting and calling; sharing documents, music, and videos; scheduling face-to-face meetings; and participating in group conversations with family, friends, and colleagues.

In this paper, we study four popular[1] IM apps: Facebook Messenger, Google Hangouts, Snapchat, and WeChat. Below, we briefly review these four IM apps.

A. Facebook Messenger

Facebook Messenger[2] (known as Messenger) is an IM app developed by Facebook in 2011 to provide a chatting platform for Facebook users. Since 2012, users could sign up for Messenger without having a Facebook account. When chatting with other Facebook friends, mobile Facebook users are required to use Messenger. Facebook Messenger has 1.3 billion monthly active users, and is the second-most popular IM app worldwide [18].

In addition to voice and video call features, users can send messages and exchange photos, stickers, audio, videos, and files, as well as send payments, share locations, chat with businesses, and play HTML5-based games with friends [19]. The maximum file size that can be shared is 25 MB.

[1]Unfortunately, we did not study WhatsApp, which is the most popular IM app. Since WhatsApp uses P2P connections for video calls and file transfers when both parties are online, we could not collect this traffic on our campus network due to its technical challenges and user privacy concerns.

[2]www.messenger.com

B. Google Hangouts

Google Hangouts is an IM app developed by Google that integrates features from both Google+ Messenger and Google Talk. Hangouts supports chat conversations, phone calls, sending multimedia content, and group meetings. Conversations can include up to 150 people, and video calls can include up to 10 people simultaneously.

At the time of writing this paper (June 2020), Google has recently converted Hangouts into two separate services called Meet and Chat. These services are aimed at bringing teams together during the COVID-19 pandemic (e.g., it now enables meetings with up to 250 people). Our data collection period pre-dated these recent changes.

C. Snapchat

Snapchat experienced rapid and unprecedented growth in the early history of mobile IM applications. It grew from 2 million Snapchat images per day in May 2012 to 6 billion images/videos per day in November 2015. In addition to text/audio/video conversation, there have been many innovative features that made Snapchat a unique IM service, especially for the younger generation. However, with the emergence and popularity of Instagram and Tik Tok, and also after a heavily criticized redesign, Snapchat has been declining in popularity recently at a rate rivaling that of its meteoric rise.

D. WeChat

WeChat[3] is an IM app developed in China and released in 2011. With 1.1 billion monthly active users, it is the third-most popular IM app worldwide [18].

In addition to text/video/picture messaging, WeChat provides moments (similar to stories in Facebook), mapping and localization services, game playing, E-wallet and online payment services, and dating. The maximum file size limit is 25 MB. WeChat is available for mobile, desktop, and web platforms.

III. Data Collection and Methodology

In this section, we describe our methodology to collect and analyze the large-scale network traffic that we used in this study. The most common approaches for network traffic measurement can be categorized into active and passive methods. First, we developed an active measurement technique to identify IM app traffic for a single smartphone under test. Second, we conducted passive measurements to collect and analyze a large volume of IM network traffic for this study. Here we explain these two phases in progressive detail.

A. Active Measurement

Active measurement refers to injecting traffic into the network for the purpose of measurement. To study IM network traffic generated from a particular mobile app, we experimented with a single mobile device in a controlled testbed environment.

[3]www.wechat.com

Active measurement brings several advantages to our study. First, it indicates the domain names, IP addresses, and port numbers that an application uses. Second, it identifies the traffic patterns associated with different interactions like login, downloading, uploading, streaming, and logout. Third, it provides an overview of generated TCP/UDP connections and the application's traffic behaviour. Fourth, it enables us to study application behaviour in different settings like different devices, operating systems, and network connection types.

However, there are some challenges with active measurement. First, the capturing mechanism should not interfere with application performance. Second, many applications, and in particular IM apps, use end-to-end encryption to secure their connections. Developers use encryption mechanisms to protect user privacy and also to conceal advertising and tracking connections. Finally, there is traffic from other smartphone apps that needs to be filtered (removed) from the analysis.

B. Passive Measurement

Passive measurement refers to capturing traffic generated by other users and applications in the network. It consists of observing traffic omnisciently at an observation point to extract traffic information and performance metrics. Such data can be collected at end hosts or at intermediate nodes within the network, providing insights into the activities at a node or on a network link.

Large-scale traffic measurement faces many challenges. First, it requires having network measurement facilities that are capable of capturing all traffic passing through the campus network. Second, it requires a massive storage infrastructure to store traffic data. Third, we need a mechanism to retrieve and analyze several terabytes of raw data. The final challenge relates to ethical considerations and user privacy. In the next subsection, we explain how we address these many challenges.

C. Data Collection Details

Here, we describe the details of our active and passive measurement approaches.

1) Active Measurement Phase: Our active measurement methodology captures the network traffic generated from an IM app running on a specific smartphone under test. We conducted the active measurement on both Android and iOS platforms. All measurements were done on a Google Pixel smartphone (Quad-Core 2.15 GHz CPU, 4 GB RAM) running Android 8.0, and an iPhone 7s (Quad-Core 2.70 GHz CPU, 3 GB RAM) running iOS 13.1. We used a separate PC (Core i7, 8-core, 3.6 GHz CPU, 8 GB RAM) running Ubuntu 18.04 Linux to capture all incoming and outgoing network traffic from the smartphone. Both the PC and the smartphone were associated to the same WiFi access point.

To intercept the network traffic, we set up a man-in-the-middle (MITM) proxy to pass the smartphone traffic through the desktop machine. We installed MITMproxy 5.0 [4] and Wireshark[4] on the desktop PC to capture full packet traces of Internet traffic activities. Prior to running experiments, we performed a factory reset on the smartphone, and we removed all extraneous apps that we were able to delete. Then we installed the MITMproxy CA Certificate on the smartphone and configured the smartphone proxy to the IP address of the desktop PC. We disabled Internet access for any other remaining mobile apps.

We manually tested all of the features of each IM app during a 10-minute experimental session. We repeated this process on several different days to make sure that the destination IP addresses were consistent. During each test, the smartphone traffic was passed through the MITMproxy and Wireshark, and all HTTP/HTTPS flows and packet-level traffic were collected.

In addition to our four target IM apps, we conducted active measurements on other related apps from the same vendor. For example, we collected data from other Facebook apps to determine if Facebook Messenger was using distinct IP addresses. We observed that all of the IM apps under study mainly made connections to distinct IP addresses. However, we observed that all Facebook apps (i.e., Facebook, Instagram, and WhatsApp) connect to analytics servers like the `graph.facebook.com` service. However, the number of such connections was negligible.

From the active measurements, we obtained a list of the main IP addresses used by each IM app.

2) Passive Measurement Phase: The dataset for our study was collected using passive network traffic measurement at the University of Calgary main campus, where the edge routers connect the campus network to the Internet. Our edge network is used by about 32,000 undergrad/grad students and 3,000 faculty/staff.

For our passive measurements, we used an Endace[5] DAG (Data Acquisition and Generation) Packet Capture Card installed at the router that connects the University of Calgary campus network to the Internet. Due to storage limitations and user privacy concerns, our data collection mechanism stores the network traffic only as connection-level summaries, and does not record network packet payloads. First, the DAG card sends the captured traffic to a set of Bro (Zeek) worker nodes[6] that are running on our monitoring system. Then Bro summarizes each connection using a one-line entry in its connection log. This entry contains many fields, including a timestamp, the IP addresses and ports of source and destination, the TCP connection duration, the TCP connection state, and number of packets and bytes sent and received on each TCP connection [9].

To analyze the captured data, we use Vertica[7]. Vertica is an analytic database management tool that is designed to handle large volumes of data. It is optimized for big data analytics and enables very fast SQL query performance using parallelization. Once our Bro data is loaded, we use SQL queries to perform our traffic analyses.

[4]www.wireshark.org

[5]https://www.endace.com

[6]https://zeek.org

[7]https://www.vertica.com

D. Ethical Considerations

Ethical considerations in network traffic measurement are paramount. Our network monitoring facilities are installed in a secure data center and have restricted access. The permission to collect network traffic is obtained via ethics review from the University of Calgary, and in cooperation with IT. Our study is related to the device-generated traffic, not human-generated traffic. When a device connects to the campus network, it typically receives a temporary IP address from DHCP, which effectively makes the user anonymous. However, it is our duty as researchers to ensure that the traffic collection is used only for suitable research purposes, and handled in an appropriate manner such that user privacy is respected.

IV. Measurement Results

In this section, we present the traffic measurement results from our campus edge network for the four chosen IM apps.

A. Traffic Profile

Figure 1 presents a graphical overview of the daily traffic patterns for IM apps. Both graphs are time series plots, showing one week of activity, with the five weekdays (Monday to Friday) followed by the two weekend days (Saturday and Sunday). The top graph represents the number of TCP connections that were initiated by each app in each one-hour interval during the week. The bottom graph shows the corresponding plot for data traffic volume in bytes. Note that the latter graph uses a logarithmic scale on the vertical axis.

The most noticable observation from both graphs is a strong diurnal pattern for IM app usage on our campus. As expected, IM activities seen by our campus network monitor are largely driven by human presence, with strong peaks during the normal working hours for our campus community, and then lighter traffic in the late evening or early morning hours. There is also an obvious reduction in activity on weekends since fewer people are on campus then. Interestingly, WeChat usage is the lowest among the four IM apps during normal working hours on weekdays, but highest among the four IM apps on evenings and weekends. This pattern might indicate more usage among students in the campus residences, or for IM activities with other users in different time zones (e.g., China).

In Figure 1, there is no direct correlation between the number of TCP connections for a specific IM app in the upper graph and its corresponding data traffic volume in the lower graph. For example, Messenger has the most TCP connections on weekdays, but the lowest traffic volume. This observation suggests that Messenger is primarily used to send short text messages rather than sharing files or videos.

A final observation from Figure 1 is the relatively weak correlation between the number of connections and the data traffic volume for all IM apps, especially on weekends. The number of connections declines on the weekend for most IM apps, but the traffic volume does not always decline proportionately. This observation suggests that some IM users (e.g., WeChat) tend to exchange larger objects (like multimedia content) on weekends.

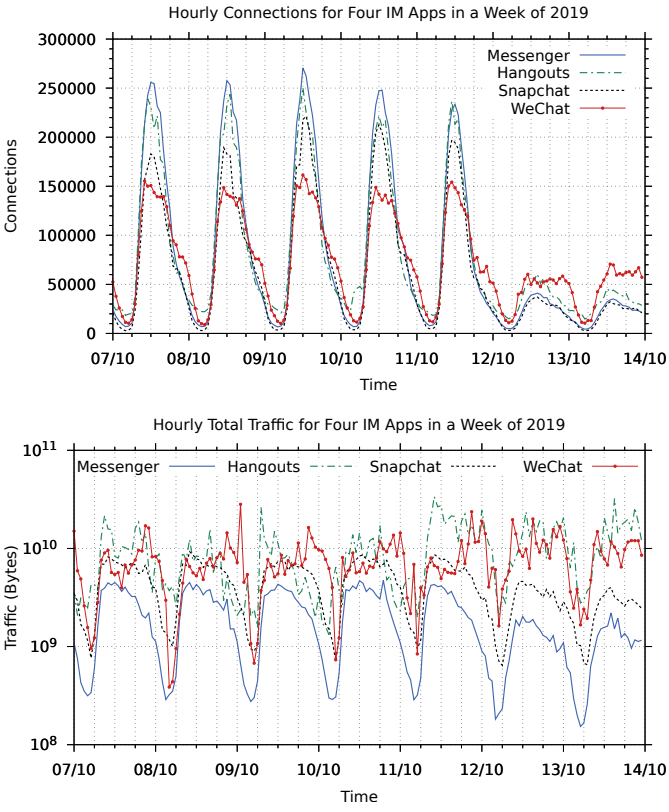

Fig. 1. Traffic Profile for IM Apps (October 7-13, 2019)

B. Traffic Volume

Table I provides a statistical summary of our week-long dataset. This table shows the number of connections, mean duration of connections, sent/received bytes, and the number of distinct client IP addresses and subnets observed for the IM apps on our campus network.

Table I shows that three of the IM apps (Messenger, Hangouts, and WeChat) had comparable numbers of TCP connections (12.1-13.4M), while Snapchat had fewer (10.8M). Whether this reflects the relative popularity of IM apps, or their communication architecture, is hard to say. However, the table shows clear differences in the mean duration of connections. Google Hangouts has the highest mean duration (112 seconds), followed by Facebook Messenger (92 seconds), WeChat (46 seconds), and Snapchat (21 seconds), respectively. The mean connection durations are influenced, of course, by default timeouts for persistent TCP connections, and by extreme connection durations in the tail of the distributions.

In our collected active measurement data, we observed that the IM apps typically use HTTP methods like GET and POST, as is the norm in Web-based conversational applications. As a result, we expected that the inbound and outbound data volumes of IM traffic should be almost balanced. However, there are a lot of differences across the set of IM apps, as

TABLE I
OVERVIEW OF EMPIRICAL DATASET FOR IM APP TRAFFIC (UNIVERSITY OF CALGARY, OCTOBER 7-13, 2019)

Item Description	IM app	Mon Oct 7	Tue Oct 8	Wed Oct 9	Thu Oct 10	Fri Oct 11	Sat Oct 12	Sun Oct 13	Total
TCP Connections	Messenger	2,568,678	2,491,317	2,603,869	2,504,171	2,087,742	562,818	489,681	13.3 M
	Hangouts	2,446,284	2,381,125	2,433,521	2,404,374	2,142,826	845,388	731,959	13.4 M
	Snapchat	1,869,472	1,886,089	2,201,279	2,109,891	1,782,527	513,141	445,273	10.8 M
	WeChat	2,072,794	1,974,731	2,067,017	1,982,426	1,879,129	1,035,352	1,110,427	12.1 M
Mean Duration	Messenger	92.1 s	91.6 s	93.4 s	89.8 s	92.3 s	93.3 s	92.7 s	92.2 s
	Hangouts	104.9 s	106.4 s	105.3 s	103.3 s	112.0 s	125.3 s	126.3 s	111.9 s
	Snapchat	49.8 s	49.0 s	42.8 s	42.3 s	42.6 s	46.9 s	47.1 s	45.8 s
	WeChat	21.1 s	21.3 s	21.0 s	21.2 s	20.9 s	21.0 s	19.8 s	20.9 s
Bytes Sent (GB)	Messenger	15.6	16.2	14.2	17.4	14.5	8.8	8.6	95.3 GB
	Hangouts	144.1	99.6	72.6	98.6	103.0	82.3	82.4	682.6 GB
	Snapchat	27.3	26.0	27.4	29.0	28.3	19.4	16.9	174.3 GB
	WeChat	19.3	16.5	19.7	23.5	23.8	37.2	27.9	167.9 GB
Bytes Recd (GB)	Messenger	38.8	37.3	37.4	39.3	35.4	15.9	14.4	218.5 GB
	Hangouts	59.2	51.2	75.3	59.5	215.1	194.9	209.9	865.1 GB
	Snapchat	86.3	89.4	82.5	90.4	83.2	45.5	37.2	514.5 GB
	WeChat	138.2	121.1	149.9	140.8	147.2	195.6	152.7	1.0 TB
IP Addresses	Messenger	3,050	2,985	3,046	2,911	2,768	2,009	1,934	4,054
	Hangouts	4,525	4,522	4,502	4,497	4,132	3,030	2,966	5,469
	Snapchat	1,993	1,968	1,934	1,866	1,775	1,272	1,170	2,882
	WeChat	1,348	1,409	3,180	1,371	1,370	1,093	1,085	5,053
IP Subnets	Messenger	117	116	116	113	115	67	63	126
	Hangouts	142	146	147	146	144	129	125	154
	Snapchat	44	45	43	44	44	33	28	56
	WeChat	145	149	256	140	132	98	104	256

well as on a day-to-day basis. In general, three of the IM apps (Messenger, Snapchat, and WeChat) have asymmetric traffic volumes with 2-6x larger inbound traffic, while one (Hangouts) actually has higher outbound traffic on several of the weekdays. There are two reasons for the latter behaviour. First, we found that Hangouts stores a copy of received multimedia content on the user's Google Photos account by default. This means that for each inbound multimedia object, there will be equivalent outbound traffic. Second, in an academic environment, many faculty members use Hangouts for video conference meetings, which increases the outbound traffic.

A final observation from Table I relates to the number of distinct client IP addresses and subnets on which IM app traffic was seen. These statistics reflect specific features of our campus network, which uses DHCP, NAT, and VPN to support wireless and wired access to the Internet. As shown in Table I, Google Hangouts was used with the most distinct IP addresses. This is not surprising since Google Hangouts is installed on all Android devices by default. In contrast, we see that Snapchat had the fewest distinct IP addresses. This result suggests that Snapchat users are fewer in number, or not fully distributed throughout all areas of the campus network. A possible reason is that Snapchat is used by younger undergraduate students, and is less popular among faculty members and international graduate students. Surprisingly, WeChat was seen from the most /24 IP subnets, suggesting that WeChat users are more widely distributed across the campus than the users of other IM apps.

C. TCP Connection State

We next analyze the TCP connection states reported in the connection logs. Recall that a normal TCP connection starts with an SYN/SYN-ACK handshake, then it sends/receives data, and finally, it ends with a FIN/FIN-ACK handshake that is represented by an SF state. However, not all TCP connections observed in real networks obey this standard implementation.

Table II shows that 63.7% of WeChat's TCP connections had the SF state, while Hangouts had 54.7%, Snapchat had 43.2%, and Messenger had 41.2%. Surprisingly, this trend differs greatly when considering the proportion of data volume exchanged via SF connections. For example, the latter value is highest for Snapchat (85.8%), and lower for the other three IM apps (Messenger 64.9%, Hangouts 61.7%, and WeChat 58.6%). These results are due to differences in app design, as well as the OS implementations of TCP.

The S1, S2, S3, and OTH states represent long-duration connections where the monitor did not observe both the SYN and the FIN from each of the transport-level endpoints. This could be because of network load or because these events straddled across multiple connection logs (each is 3 hours in duration).

For all IM apps, TCP resets by the originator (RSTO) are much more prevalent than resets by the responder (RSTR). We also see that for all IM apps that we studied, the proportion of rejected TCP connections was negligible (i.e., less than 1% for both connections and exchanged data).

D. Transfer Sizes

Next, we investigate the data transfer sizes and durations of the TCP connections established by IM apps. We focus specifically on the tails of these distributions, expecting power-law structures as evidence of heavy-tailed distributions [16].

TABLE II
SUMMARY OF TCP CONNECTION STATES OBSERVED FOR IM APPS (UNIVERSITY OF CALGARY, OCTOBER 7-13, 2019)

TCP State	Connections				Bytes			
	Messenger	Hangouts	Snapchat	WeChat	Messenger	Hangouts	Snapchat	WeChat
SF (Normal SYN-FIN connection)	41.29%	54.79%	43.22%	63.78%	64.87%	61.70%	85.89%	58.62%
S3 (Good conn, but no FIN seen at all)	13.85%	15.99%	7.20%	8.45%	10.83%	14.11%	2.14%	8.62%
RSTO (Conn reset by originator)	11.70%	11.07%	25.26%	10.58%	7.51%	10.53%	7.43%	24.10%
SHR (SYN ACK and FIN,but no SYN)	7.99%	2.94%	1.87%	2.25%	1.42%	0.98%	0.16%	0.25%
RSTR (Conn reset by receiver)	7.82%	2.00%	0.51%	2.99%	6.53%	1.13%	0.13%	2.81%
S1 (Good conn, but server FIN only)	6.07%	5.58%	11.57%	3.30%	3.86%	6.03%	2.26%	1.29%
OTH (Mid-stream traffic (no SYN or FIN))	4.65%	1.73%	4.21%	2.27%	1.25%	1.34%	0.53%	0.73%
S2 (Good conn, but client FIN only)	2.17%	2.94%	1.28%	2.41%	1.50%	2.23%	0.45%	1.40%
RSTOS0 (Failed conn reset by originator)	1.98%	1.37%	4.12%	0.65%	0.81%	0.88%	0.91%	2.04%
SH (SYN and FIN,but no SYN ACK)	0.93%	0.81%	0.30%	0.33%	0.39%	0.47%	0.08%	0.09%
RSTRH (Conn reset by receiver)	0.52%	0.25%	0.05%	0.74%	0.51%	0.54%	0.01%	0.01%
S0 (Saw SYN, but no SYN-ACK at all)	0.52%	0.35%	0.36%	2.11%	0.01%	0.00%	0.00%	0.02%
REJ (Connection ended with a Reject)	0.51%	0.18%	0.05%	0.14%	0.51%	0.06%	0.01%	0.02%
Total (All TCP connections)	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Figure 2 illustrates the log-log complementary distribution (LLCD) plots for transfer sizes (upper graph) and connection durations (lower graph) to visualize the tail behaviors [7]. In the top graph, all four IM apps show evidence of heavy-tailed transfer sizes, with Hangouts and WeChat having some of the largest data transfers. Even Snapchat shows a pronounced tail to its transfer size distribution, despite its lower overall connection activity and traffic volume compared to other IM apps. This result could be because Snapchat users usually share images and short video files (with a 60-seconds limitation). These multimedia objects obviously generate larger data transfers than text messages, but less traffic than video conferencing and long video streaming sessions.

The bottom graph in Figure 2 shows the LLCD plots for TCP connection durations. All four IM apps show a pronounced tail to the distribution, with some connections lasting a few hours. Messenger has some of the longest-lasting connections, while the tail behavior for Snapchat is the lightest among these four IM apps.

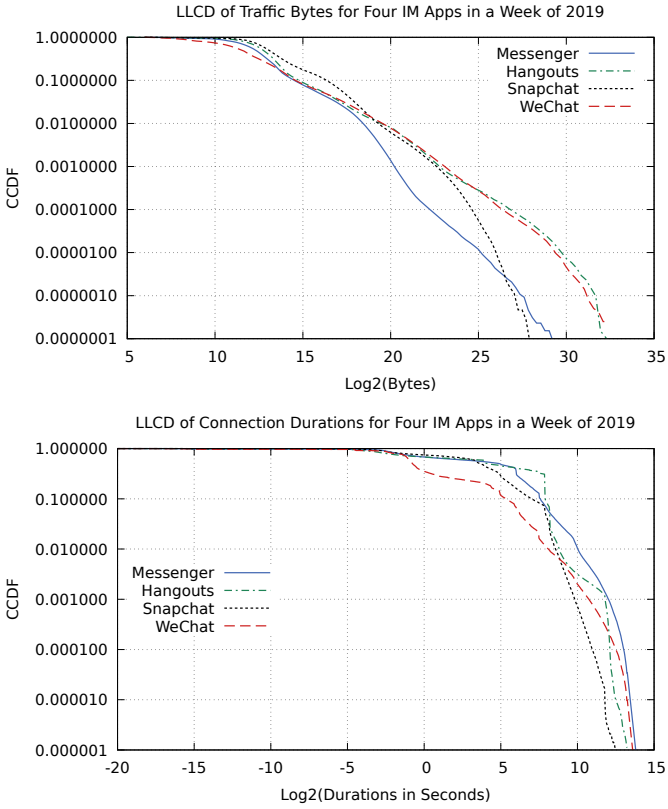

Fig. 2. LLCD Plots of Transfer Size and Connection Duration for IM Apps

E. TCP Throughput

In this subsection, we study the TCP throughput achieved by IM apps. Network throughput (in bits per second) is calculated from the transfer size and duration for each TCP connection. However, for this analysis, we excluded connections that had no application-layer data or lasted for less than 1 second.

Table III shows a statistical summary of the mean throughput for IM apps, on a daily basis, as well as overall. As shown in the table, the mean throughput for IM apps tends to be low, typically 100-300 kilobits per second (Kbps). This may be due in part to the persistent connection timeout values used, or the type of content exchanged by the IM app. For example, Messenger and WeChat, which are primarily used for text messaging, have lower mean throughput than Snapchat and Hangouts, which tend to transfer more multimedia files.

An interesting observation from this table is that for each IM app, the mean throughput remains similar for the first three working days, but then increases slightly on Thursday and Friday. Furthermore, the mean throughput for some IM apps almost doubles on the weekend. As noted earlier in Section IV-A, this indicates that some users of IM apps tend to send/receive larger files on weekends.

Figure 3 illustrates the LLCD plot of IM app throughput for our one-week observation period. The throughput of IM apps is highly variable, with a noticable upper tail. Facebook Messenger has lower throughput than other IM apps. The other three IM apps all have comparable throughput distributions, including the tails.

TABLE III
MEAN THROUGHPUT (KBITS/SEC) OF FOUR IM APPS IN WEEK OF OCTOBER 7-13, 2019

Item Description	Mon Oct 7	Tue Oct 8	Wed Oct 9	Thu Oct 10	Fri Oct 11	Sat Oct 12	Sun Oct 13	Average
Messenger	70.97	69.05	68.09	79.12	80.52	129.85	138.73	78.10
Hangouts	310.48	272.07	267.44	294.17	363.48	550.17	631.02	334.61
Snapchat	230.09	224.11	216.26	235.20	256.59	402.06	393.68	246.63
WeChat	108.19	103.03	105.07	114.17	138.92	243.43	220.91	134.53

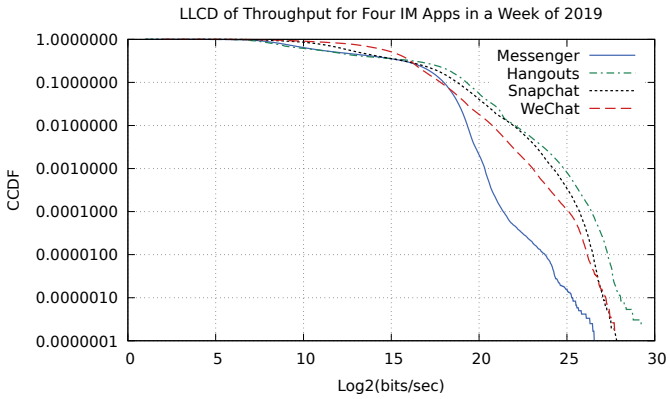

Fig. 3. LLCD Plot of TCP Connection Throughput for IM Apps

F. Performance Implications

As we have seen, IM app traffic can put a significant amount of workload on our campus network. While this workload is not extremely large compared to video streaming applications like Netflix and YouTube (several Terabytes per day), the traffic from IM apps will undoubtedly grow as they offer more bandwidth-consuming features such as video messages, video editing, multimedia stories, online games, and more.

Another important observation was a shift in IM app usage characteristics on weekends (e.g., connections, transfer sizes, throughput). This usage shift might also apply for other social/entertainment applications, such as social networks, video streaming, gaming, and so on.

Observing an average of 650 GB of daily traffic volume from only four chosen IM apps on our campus network was higher than we expected. Considering this traffic and its potential future impact is essential for understanding and improving the performance of our campus network.

V. RELATED WORK

Early works on traffic characterization of IM apps analyzed unencrypted traffic of applications that were originally designed for PC or Web platforms. Xiao et al. [20] studied network traffic generated from AOL Instant Messenger (AIM) and MSN/Windows Live Messenger, from 4000 employees in a large enterprise network. They found that chat messages contributed just a small portion of the overall IM traffic, while the presence status, hints, and other features contributed most of the traffic. They observed that the social network structure of IM users can be characterized by a Weibull distribution. In other early studies, Reust [17] studied AIM, Dickson [6] analyzed Yahoo! Messenger, and Kiley [12] studied popular IM Web sites.

After IM applications migrated to smartphone platforms, these mobile apps have been the subject of numerous studies in network security and digital forensics. These studies mainly aim to identify data privacy leaks in IM apps, and identify user behaviour within encrypted traffic.

Fiscone et al. [8] proposed a side-channel approach to detect the actions of WhatsApp users within encrypted traffic. The proposed approach is able to recognize call termination, missed call, blocked call, and call rejection in unicast and multicast (group call) WhatsApp conversations. Bahramali et al. [1] investigated the encrypted traffic of IM users (mainly Telegram) from an adversary viewpoint, with the goal of identifying (the IP addresses of) the members or administrators of target IM channels. The main idea is to match the flow patterns of channels to the traffic patterns of flows monitored in other parts of the entire network. The proposed attack mechanism can identify the channel's administrators with 94% accuracy when using 15 minutes of Telegram traffic (assuming that the attacker is able to monitor the entire network, much like a totalitarian government). Finally, they proposed IMProxy, a countermeasure approach to decrease the efficacy of the adversary attacks.

There are several prior studies that exploit traffic classification techniques to identify specific IM applications or protocols among the traffic in networks. Lotfollahi et al. [15] proposed a deep-learning traffic classification approach, called Deep Packet. This mechanism is based on a convolutional neural network (CNN), which is a class of deep learning neural networks, to classify encrypted traffic. Their framework reduces the amount of hand-crafted feature engineering and achieves good accuracy, including the identification of IM apps like AIM Chat and Google Hangouts. However, these studies usually rely on lab-generated datasets for the training and classification phases. We have observed that large-scale traffic on a campus network differs greatly from sample traffic generated from a few devices in a controlled lab.

There are a few recent characterization studies on encrypted traffic from IM apps. Deng et al. [5] studied user behaviour in WeChat using two days of packet-level traffic from an urban cellular network in China. They found that WeChat traffic follows a strong diurnal pattern with several bursty spikes, and that media objects account for 70% of traffic. They also used an artificial neural network methodology to cluster users based on 8 properties: active times, active duration, media update, media uploaded/downloaded, messages sent/received, and articles viewed. They found that some users use all

WeChat services, while over 60% of users just use WeChat occasionally.

VI. Conclusion

In this paper, we used a combination of active and passive measurement techniques to characterize one week of IM app traffic on the University of Calgary's campus network. In this study, we focused on one week of empirical connection log data from October 7-13, 2019, to identify the key characteristics of four popular IM apps: Facebook Messenger, Google Hangouts, Snapchat, and WeChat.

Our characterization results show that IM app traffic is strongly driven by human presence on campus, with very clear diurnal and weekly patterns. We observed that WeChat users are also active during the evenings and weekends, and widely spread across our campus area, while Snapchat users tend to be active during weekdays and have the smallest network footprint. WeChat generates the largest inbound data traffic, while Google Hangouts generates the largest outbound traffic. On average, Hangouts had the longest average duration for its TCP connections. All IM apps showed pronounced tails to their distributions for connection duration and transfer sizes.

Our campus with 35,000 members is an example of a large and dynamic enterprise-scale edge network. Our study shows that the number of connections and the generated traffic by IM apps is large (e.g., 650 GB per day on average). Also, we observed that even though many users use IM apps, there is no single dominant IM app. Each serves its own niche of devoted users, providing a rich ecosystem for online communication via IM apps.

Our ongoing work focuses on characterizing other IM apps (e.g., WhatsApp) that use cloud-based hosting, or P2P architectures, with a diverse array of dynamically-determined IP addresses. Such applications are challenging for network traffic analysis and characterization.

ACKNOWLEDGEMENTS

Financial support for this work was provided by Canada's Natural Sciences and Engineering Research Council (NSERC). The authors thank the anonymous ICPE reviewers for their constructive feedback and suggestions on an earlier version of this paper.

References

[1] A. Bahramali, R. Soltani, A. Houmansadr, D. Goeckel, and D. Towsley, "Practical Traffic Analysis Attacks on Secure Messaging Applications", *Network and Distributed Systems Security (NDSS) Symposium*, San Diego, CA, USA, February 2020.

[2] BBC. "Chat app messaging overtakes SMS texts", *Informa says*. 29 April 2013, http://www.bbc.com/news/business-22334338.

[3] Cisco. "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022 White Paper".

[4] A. Cortesi, M. Hils, T. Kriechbaumer, and contributors, "Mitmproxy: A Free and Open Source Interactive HTTPS Proxy", 2010-, https://mitmproxy.org/, [Version 4.0].

[5] Q. Deng, Z. Li, Q. Wu, C. Xu, and G. Xie, "An Empirical Study of the WeChat Mobile Instant Messaging Service", *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 390-395, Atlanta, USA, May 2017.

[6] M. Dickson, "An Examination into AOL Instant Messenger 5.5 Contact Identification", *Digital Investigation*, Vol. 3, no. 4, pp. 227-237, 2006.

[7] D. Feitelson, "Workload Modeling for Computer Systems Performance Evaluation", *Cambridge University Press*, 2015.

[8] G. Fiscone, R. Pizzolante, A. Castiglione, and F. Palmieri, "Network Forensics of WhatsApp: A Practical Approach Based on Side-Channel Analysis", *International Conference on Advanced Information Networking and Applications*, pp. 780-791, Caserta, Italy, April 2020.

[9] M. Haffey, M. Arlitt, and C. Williamson, "Modeling, Analysis, and Characterization of Periodic Traffic on a Campus Edge Network", *IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 170-182, September 2018.

[10] J. Jakobsen and C. Orlandi, "On the CCA (in) Security of MTProto", *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 113-116, Vienna, Austria, October 2016.

[11] P. Jovanovic and S. Neves, "Dumb Crypto in Smart Grids: Practical Cryptanalysis of the Open Smart Grid Protocol", *IACR-FSE*, pp. 428-447, 2015.

[12] M. Kiley, S. Dankner, and M. Rogers, "Forensic Analysis of Volatile Instant Messaging", *IFIP International Conference on Digital Forensics*, pp. 129-138, Boston, MA, January 2008

[13] S. Klenow, C. Williamson, M. Arlitt, and S. Keshvadi, "Campus-Level Instagram Traffic: A Case Study", *Proceedings of IEEE MASCOTS*, pp. 228-234, Rennes, France, October 2019.

[14] M. Laterman, M. Arlitt, and C. Williamson, "A Campus-Level View of Netflix and Twitch: Characterization and Performance Implications", *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pp. 1 8, Seattle, WA, USA, July 2017.

[15] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning", *Soft Computing*, Vol. 24(3), pp.1999-2012, 2020.

[16] A. Mahanti, N. Carlsson, A. Mahanti, M. Arlitt, and C. Williamson, "A Tale of the Tails: Power-Laws in Internet Measurements", *IEEE Network*, Vol. 27, No. 1, pp. 59-64, January 2013.

[17] J. Reust, "Case Study: AOL Instant Messenger Trace Evidence", *Digital Investigation*, Vol. 3, no. 4, pp. 238-243, 2006.

[18] Statista. "Most popular global mobile messenger apps as of July 2019, based on number of monthly active users (in millions)".

[19] Wikipedia, "Facebook Messenger", September 2019. https://en.wikipedia.org/wiki/Facebook_Messenger.

[20] Z. Xiao, L. Guo, and J. Tracey, "Understanding Instant Messaging Traffic Characteristics", *27th International Conference on Distributed Computing Systems (ICDCS '07)*, pp. 1-8, Toronto, Canada, July 2007.

[21] L. Zhang, X. Chao, P. Pathak, and P. Mohapatra, "Characterizing Instant Messaging Apps on Smartphones", *16th International Conference, Passive and Active Measurement (PAM)*, pp. 83-95, New York, NY, USA, March 2015.

[22] Z. Zhang and C. Williamson, "A Campus-level View of Outlook Email Traffic", *Proceedings of ICNCC*, Taipei, Taiwan, pp. 299-306, December 2018.