UNIVERSITÄT AUGSBURG
Universitätsbibliothek

OPEN ACCESS
OPUS AUGSBURG

Fitting the puzzle: towards source traffic modeling for mobile instant messaging

Fabian Poignée, Anika Seufert, Frank Loh, Michael Seufert, Tobias Hoßfeld

Angaben zur Veröffentlichung / Publication details:

Poignée, Fabian, Anika Seufert, Frank Loh, Michael Seufert, and Tobias Hoßfeld. 2024. "Fitting the puzzle: towards source traffic modeling for mobile instant messaging." In *15th International Conference on Network of the Future (NoF 2024), Castelldefels, Spain, 2-4 October 2024*, edited by Toktam Mahmoodi, Raul Muñoz, Thi-Mai-Trang Nguyen, and Sebastian Troia, 229–37. Piscataway, NJ: IEEE. https://doi.org/10.1109/NoF62948.2024.10741510.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

Fitting the Puzzle: Towards Source Traffic Modeling For Mobile Instant Messaging

Fabian Poignée*, Anika Seufert*, Frank Loh*, Michael Seufert †, Tobias Hoßfeld*
* *University of Würzburg, Chair of Communication Networks*, Würzburg, Germany
{fabian.poignee|anika.seufert|frank.loh|tobias.hossfeld}@uni-wuerzburg.de
† *University of Augsburg, Chair of Networked Embedded Systems and Communication Systems*,
Augsburg, Germany, michael.seufert@uni-a.de

***Abstract*—Mobile instant messaging (MIM) via applications such as WhatsApp transformed human communication by enabling the exchange of various different message types, such as text, image, video, or voice, globally at every time of day. Network providers are confronted with a substantial user base and network load which is especially high in group chats where each message needs to be distributed to each member. Since end-to-end encryption restricts insights into user traffic, it becomes essential for network operators to obtain knowledge about the communication and the resulting load on the network through MIM by other means, which makes it necessary to model the network traffic of MIM. In this work, we therefore present a theoretical approach to source traffic modeling for MIM. Therefore, we identify the building blocks of a source traffic model (STM) for MIM and determine missing pieces. We fill the gaps through studies on MIM communication networks, user proximity, media compression and payload size, as well as media file size distribution. Combining existing literature and our work, we present a theoretical modular STM approach which can be used for developing STMs for MIM. With this, we provide a comprehensive description of MIM in the network researching context and enable consideration MIM in future network design.**

***Index Terms*—mobile instant messaging, traffic modeling, message generation, contact network, traffic measurement**

I. Introduction

The ubiquitous nature of the Internet illustrates the dynamic interaction between technology and human behavior. It transforms our daily life while user adoption also drives technological advancements. One application area that clearly reflects this change is Mobile Instant Messaging (MIM), including apps like WhatsApp, Signal, Telegram, WeChat, and Facebook Messenger. These apps popularize direct conversations, but also group conversations, diverging from traditional one-to-one communication like e-mail or SMS, or one-to-many methods including social media. Consequently, a fundamental change in communication patterns is visible and a general investigation of the impact on the current network and generated load is essential. From the perspective of a network provider, the comprehensive challenge is tackling the huge number of users. According to [1], currently more than every third person worldwide uses MIM, and the usage numbers increase by about 100 million per year. For that reason, only a minor increase of the average individual traffic generated by MIM apps has a significant impact on the global load detected in networks. Furthermore, MIM apps do not only support text messages but also various media types, such as images or video, and asynchronous communication via group chats that additionally stresses networks. This asynchronous communication realized as a delay-tolerant publish-subscribe model multiplies network traffic to each recipient which burdens the network, particularly when large media files are shared.

However, network operators face challenges to understand mechanisms in group-based communication in detail and the resulting traffic generation processes rooted in MIM applications. Network measurement studies to identify all communication specifics have many limitations because of thorough end-to-end encryption. Consequently, extensive data to identify the general impact of MIM on current and future communication networks are not available and in-depth models to predict the way MIM application traffic impacts future network load effectively are currently not existent. To this end, the development of a comprehensive source traffic model (STM) for MIM apps is essential to, among others, model and predict up- and downlink traffic of individual MIM application users, the impact of MIM server traffic on a general wide area network infrastructure, or peak hour traffic on current and future mobile networks. While the communication within a chat has been researched and modeled within literature, missing information on the communication network and generated network traffic of typical media sizes, development of STMs is not possible.

Our goal is to provide a theoretical approach for generating STMs for MIM apps that describes user messaging behavior and the impact on the network as realistically as possible so that research on future networks can accurately take MIM into account. For this reason, we research the literature and identify existing and missing pieces towards a STM. For each missing aspect, we collect data to fill the gaps in the literature. Therefore, we investigate user contact networks, spatial proximity between members of group chats, the impact of media size compression on the amount of network data in different MIM apps, as well as characteristic media file sizes. Merging existing work and newly collected data, we are able to present a theoretical, modular approach for STM generation.

Consequently, the contribution of this work is two-fold. First, after identifying the gaps in the literature on MIM, which include the communication network and generated network

traffic, we bridge those gaps. We detail on the communication network and the contact structure of individual MIM users, the number of chats and their sizes, as well as on the proximity to other users. These valuable insights facilitate comprehensive modeling of communication structure with MIM apps and can also assist in the development of improved message and media distribution solutions for chat groups. Additionally, via in-depth measurement study of message transmissions including image, video, voice, and text content, details about compression effects and the resulting and finally transmitted amount of data are revealed. Secondly, we propose a theoretical and modular approach to STMs, with the insights of our and previous studies, that can be used to create STMs as input for simulations covering MIM. With this, we provide a comprehensive description of MIM that is valuable for networking research.

In the remainder, Sec. II presents background and related works. We investigate the MIM contact network in Sec. III, estimate user proximity in Sec. IV, describe our traffic measurements in Sec. V, and investigate transmitted file sizes in Sec. VI. Based on these insights, we present theoretical STM development in Sec. VII and conclude in Sec. VIII.

II. Background and Related Work

To obtain a comprehensive STM for MIM applications, it is important to understand their usage and the traffic generation process. For better understanding and simplified description, we summarize required components in three areas, shown in Figure 1. For each component the figure shows already existing research from the literature in black and content that is not yet available in the literature in blue. Consequently, the goal of our investigation in this work is to fill the gaps and finally, develop a theoretical approach to STM generation.

First of all, it is important to examine the structure of the *communication network* of users within MIM apps. For example, it is important to determine the average number of individual and group chats per user. Furthermore, the total number of communication partners per user is also relevant for a STM. Second, the *communication within (group-) chats* has to be modeled. Factors such as the number of participants in each group and the frequency of message transmissions (inter-arrival time - IAT) as well as its media type need to be clarified here. Finally, the *network traffic caused by MIM apps* has to be modeled. Questions regarding the use of compression techniques for different media types, the network load caused by certain message types and the quantification of message overhead must be addressed. In general, the more detailed each aspect of MIM communication is modeled, the more accurate the STM can be.

A. Modeling the Communication Network

Although MIM is an important research topic, to the best of our knowledge there is only limited research on the communication networks within MIM apps. In [2], the authors present a model designed to characterize the social communication network in MIM apps. To evaluate the effectiveness of the proposed model, simulation experiments are conducted and the results are compared to an authentic real-world graph from Telegram. However, the model focuses on the use of so-called communication channels, and is thus, not generally applicable for generating a comprehensive MIM communications network with individual chats and groups. To fill this gap, in Sec. III, we conduct a user study to gain deep insights in the communication network of MIM users.

Furthermore, it is interesting for network operators to know the geographic distance between people communicating with each other via MIM apps. This information helps to apply strategies to reduce the network load, such as Device-to-Device (D2D) communication or edge caching. In [3], for example, a simulated evaluation of potential traffic savings in MIM apps is conducted. The simulation uses edge caching and D2D communication strategies to transmit messages locally and reduce the load on the mobile network. Their results show that the ratio of locally transmitted messages depends heavily on the proximity of members within a group. However, to the best of our knowledge, the spatial proximity of MIM communication partners at the time of message transmission has never been investigated. This knowledge gap motivates our study in Sec. IV, in which we examine the dynamics of proximity between the communication partners.

B. Modeling the Communication within a Chat

Looking into the communication of individual groups and chats reveals a substantial body of related work. In [4], a survey has been conducted and private chat histories of 243 users are examined to generate initial statistics on chat groups. This allows a characterization of groups and the development of a simple communication model. Building on this, [5] conduct a further analysis of the same data set, refining the model to understand the active participation of users in group chats and the resulting network traffic. The study in [6] involves the collection and analysis of a data set comprising 178 public WhatsApp group chats. This data set includes approximately 45,000 users and 454,000 messages. The evaluation covers metrics such as the number of messages per group and per user, user locations, message content, and language. For the most active groups, the study also examines the number of messages per day. The most detailed MIM communication modelso far is from [7], [8], where authors present an extensive data set [9] comprising 5,956 private WhatsApp chat histories and over 76 million messages from more than 117,000 individuals. They describe and model the characteristics of chat groups and users, discuss the intricacies of communication within these groups, and offer fundamental insights into private MIM communication. We can use their models for our theoretical approach to STMs without further studies in this area.

C. Modeling the Generated Network Traffic

The communication in MIM apps is diverse, allowing the exchange of different media types such as text, voice, images, or videos. To manage the increasing network load caused by

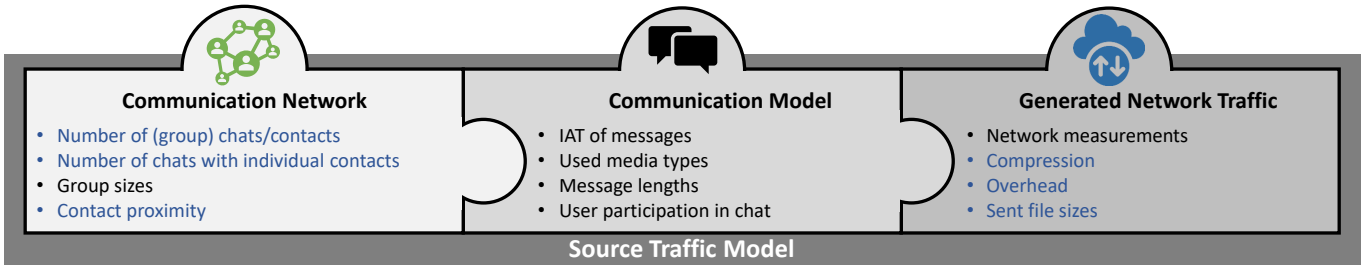


Fig. 1: Components of a STM (black text: available in the literature, blue text: missing in the literature).

the transmission and multiplication of media files in group chats, most MIM apps employ compression techniques. These techniques are crucial to reduce the media file sizes before transmission, thereby limiting both data traffic and network resource consumption. Unfortunately, for most MIM apps, the compression standard is not publicly available.

In the past, researchers already investigated the traffic generated by MIM. In [10], the authors manage to analyze the semantics of encrypted network traffic generated in WhatsApp. Through this analysis, they successfully detect specific app functions such as call termination, missed/rejected calls, and blocked calls. In a different approach, a blind traffic detection technique is proposed [11], capable of differentiating unique WhatsApp calls from encrypted traffic. Furthermore, wiretap data has already been employed to explore methods for determining whether someone is sending or receiving WhatsApp messages at a given time [12]. The authors of [13] use both active and passive measurement techniques to characterize MIM app traffic over the course of a week on the University of Calgary campus network. The study shows that a considerable volume of traffic, on average 650 GB per day, is generated by MIM apps. In [14], [15], an encrypted MIM traffic generation tool is developed with which traffic characteristics of MIM applications are analyzed. Therefore, the authors employ a data-driven approach that utilizes machine learning classification models to analyze and discern encrypted traffic from six distinct MIM apps. Their findings demonstrate the feasibility of distinguishing the behavior of various MIM apps.

Nevertheless, a simple model which describes traffic generated by sending text, image, video, or audio messages as well as corresponding overhead is missing. Therefore, we present our findings of a thorough measurement campaign regarding the impact of file compression and payload sizes for different media types in WhatsApp, Signal, and Telegram in Sec. V. Furthermore, although the frequency of media transmissions via MIM apps is described in the literature, the media size distributions of the content are unknown. Consequently, we conduct our study in Sec. VI to close this gap.

III. Investigation of a MIM Contact Network

To fill the gap about the comprehension of communication structures and the underlying social network within MIM, we conduct a user study to obtain information about the number of chats and contacts of a user, and how many chats users share with each contact. This information is essential to accurately model the number of chats for a single user and the degree of message replication for a STM.

A. User Study for a Contact Network via Browser Extension

For this, we designed and developed a Google Chrome browser extension as a data collection tool for the web client of the MIM app WhatsApp. Upon logging into `web.whatsapp.com` via a QR code from a smartphone, users can enter an e-mail address into a text field and run our extension. The extension then scrolls through the list of all chats from a user. For each chat, the date of the last message and the contacts within direct chats or group chats are extracted. Prior to any data transmission to the server, an anonymization process takes place, to ensure the confidentiality of both chat and contact names. Despite the anonymization, contacts of a user can still be identified across the user's chats via the inserted aliases, thereby enabling the generation of a social network based on the user's contacts. We define this social network as a weighted graph denoted by $G = (V, E)$ where $v \in V$ represents a person and $E \subseteq \{\{x, y\} | x, y \in V$ and $x \neq y$ and chat$(x, y)\}$ describes the set of edges where two persons x and y are connected if they share a chat. Furthermore, we set the weight $w(e)$ of an edge between two persons to the number of shared chats, since it is possible to have a direct chat and multiple group chats which include the same person. We collected data with our extension from students on a university campus in February 2023. Consequently, the demographics of our participants is strongly biased towards young and well-educated people. In return for participation, our test group received statistics about their contact network. After validation against double entries in our data set, we obtain 48 networks for further usage.

B. Modeling MIM Contact Structures

From the contact data obtained via our browser extension, we model contact structures in the following. Results show an average of 144.13 chats for a single person of which 40.41 % are group chats. The share of direct chats is 30 % less than in the data set from [7]. As such, we assume that [7], due to their manual submission of chats, suffered from a participation bias, whereby disproportionately many group chats have been submitted to their analysis system. However, the group size distribution for chats with more than two members is similar and validated by our data.

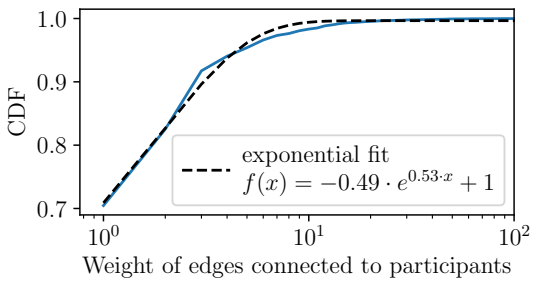


Fig. 2: Number of shared chats with contacts.

Next to the size of chats, the number of contacts of a user is essential. While one participant has only seven contacts, the maximum number of contacts in our data is 2,962. On average, a person is connected to 489.6 other users. The median is at 316 contacts and the 90 % percentile is at 1,060. We are able to model the cumulative distribution function (CDF) for the number of contacts of a single person via an exponential fit $F(x) = 1 - e^{-0.0021 \cdot x}$ with an R^2 score of 0.97.

For a deeper understanding of shared chats with each contact, we construct the social graph for each participant and examine the weight of each outgoing edge, which corresponds to the number of shared chats. Figure 2 shows the CDF of the weight of edges connected to a participant and the exponential fit which achieves an R^2 of 0.99. The minimum edge weight is one and occurs for 70.46 % of all contact relationships. Thus, the prevalent relation between two MIM users is via only a single chat. One reason for this result could be that users are often added to a group chat for, e.g., a social event. There, they are unfamiliar with many chat members, and consequently, have no other shared chats. For 91.74 % of all contact relationships, the weight is three or less, showing that in most cases, two persons share very few chats. However, the tail of the distribution is long and the maximum value in our data set is a person that shares 198 chats with another person. Overall, we see many users with a moderate number of connections and some users that are highly interconnected with others which is typical for a social network.

IV. Estimating MIM User Proximity

Since social networks and their communication span globally, an important element to model MIM communication is the consideration of the proximity of chat partners to, for example, estimate how much traffic can be handled locally. Thus, user proximity is tackled by our study in the following.

A. User Study Questionnaire on WhatsApp User Proximity

In all of our user studies presented in this work, participants were also invited to fill an online questionnaire about their proximity to their WhatsApp contacts when sending or receiving messages. Furthermore, if participants did not want to share their data within one of our studies, they were given the option to only participate in our questionnaire. In total, we received 152 filled forms. Since users may be spread across a variety of distances when sharing messages, the impact on the network is different. For example, communication across the globe requires routing between multiple autonomous systems and their network providers, while communication within the same building involves fewer network hops. Furthermore, the results of our questionnaire are important for analysis of proximity-leveraging technologies such as caching and D2D communication, contributing valuable insights that can enhance future works, but also existing contributions in this domain [3]. Thus, participants were asked to reflect on their most recent group message and estimate the proximity to fellow group members during that interaction. Response options range from within the same room to within the same building, city, or country. This allows to get an estimate on user proximity since, unfortunately, more precise methods are highly privacy invasive and have poor practical feasibility.

B. Proximity Questionnaire Evaluation

Figure 3 shows the CDF of the reported chat member proximity with respect to the share of participants at the time of the last message within their latest group chat. The x-axis displays the share of members within the proximity based on the reported group size. The different proximities describe whether a person was in the same room, building, city, or country. Note that the proximities are displayed as inclusive. That means that a person which is reported to be in the same room is automatically also in the same building, city, and country. For example, the CDF in red starts at an x-axis value of 0 % and at an y-value of 3 %, showing that in 3 % of the groups 0 % of the other members were within the same country. This means for those groups all other members were in a different country. Then, the CDF increases and in 10 % of the groups only 66 % of the group members are within the same country. On the far right, it is visible that after 33.6 %, i.e., for the remaining 66.3% of groups, all members are situated in the same country at the time of the most recent message. Similarly, for 34.2 % of the reported groups, all members are located within at least the same city, as denoted by the green line. Additionally, the orange line, representing the category “same building”, indicates that 41.8 % of the time, at least one other member is in the same building or in closer proximity to the participant. Furthermore, 29.6 % of the time at least one other group member is in the same room with the participant at that moment, as shown in blue. At the median, no other person is in the same building or closer, but 75 % of all members are located in the same city.

Although users are frequently geographically distant from each other and use MIM communication to stay connected, there is evidence that a considerable amount of messaging traffic could be handled locally without the usual server-client structure. Commonly, all traffic is transmitted to a central server and is distributed among all group members afterwards. In contrast, by using local forms of communication, such as D2D, content delivery networks (CDNs), or other forms of caching, e.g., at the local router or base station, the backbone network load could be reduced during message distribution.

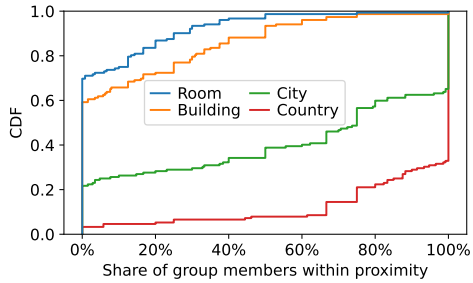


Fig. 3: Reported user proximity at time of last message.

V. Network Traffic Measurements

To develop a comprehensive STM, knowledge about only user connections is insufficient. It is also crucial to obtain an accurate comprehension of the traffic volume generated by a particular media file, requiring examination of both compression ratios and the total payload transmitted to the server. We define the network payload of a message in this work as the sum of bytes within the payload field of all network packets that belong to a single MIM application message and ignore header fields. The following section presents measurements and evaluation thereof for different media types, focusing on file compression and payload size.

A. Measurement Methodology

Our measurements focus on modeling data volumes for three widely-used MIM applications, namely WhatsApp, Signal, and Telegram, along with the most prevalent message types, including text, images, videos, and audio. Our measurements are conducted on a Google Pixel 3a running Android version 12 SP2A.220505.008. Our testbed further consists of a PC with a Linux operating system and a second Android device that is only used to receive the messages. This allows us to access the messages on the receiver side. The PC is connected to the Internet via Ethernet on a $1\,\mathrm{Gbit/s}$ connection and provides a WiFi connection to the phone. As such, we monitor and capture traffic between the sending phone and the MIM servers. We utilized the latest application versions at the time of measurement in March 2023: WhatsApp 2.23.6.9, Signal 6.10.9, and Telegram 9.4.9.

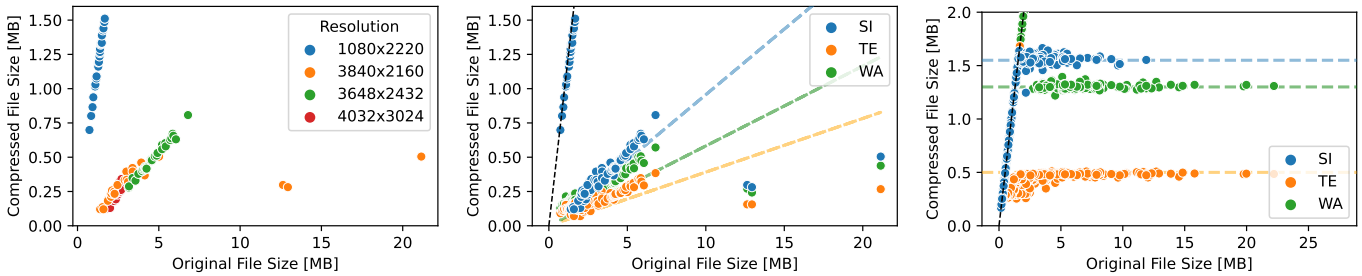
For each media type, the data used for transmission is described in the following. For our measurement study, we employ data sets from literature which have been used in compression and in, what could be called its reversal task, i.e., supersampling anti-aliasing, and are thus, suited to explore the compression algorithms of the different MIM apps. A data set featuring diverse 4K resolution images encompassing nature, people, animals, and faces, with file sizes ranging from $1.3\,\mathrm{MB}$ to $21\,\mathrm{MB}$ is provided by [16]. Additionally, we measure 30 high-quality images from a DSLR camera [17], along with native photos, i.e., smartphone camera images and screenshots, generated on our test device. From [18], we select 30 videos of $5\,\mathrm{s}$ duration in 4K resolution at 60 frames per second (FPS), since they provide a script to generate lower resolutions, i.e., 360p, 520p, 720p, 1080p, and 1440p, as well as lower frame rates. In this work, 30 and 60 FPS are investigated. Text messages of varying lengths and content are generated using text generators from [19] and [20]. Finally, as voice messages must be generated during the measurement for each MIM application, the automated testbed initiates the playback of a podcast via PC speakers before recording the voice message.

We are able to filter the traffic via reverse DNS lookup to only include packets between the MIM servers and our test device. However, due to end-to-end encryption, the measurements can include traffic, which is not directly related to message sending, e.g., key exchanges, online status updates, or typing notifications. We consider this as a minor impact for our measurements, since media files largely outweigh any other kind of application traffic and minimize the number of those effects by sending the media files from the file explorer via the share context to minimize application interaction. Note that we are also able to inspect the downloaded media since we receive all messages on the other smartphone in our lab.

B. Media Network Load Analysis

In the following, we present our measurement results for image and video compression, followed by results for image and video overhead. Afterwards, we provide an analysis of voice messages and conclude with an analysis of text messages.

Image Compression Analysis: First, we look at resolution-specific differences during image compression using Signal before we discuss the general image compression behavior of the different MIM apps. Therefore, Figure 4a shows the compression results for images transmitted with Signal. The original file size is shown on the x-axis, whereas the y-axis depicts the compressed file size. Distinct colors represent varying resolutions prior to compression. The dotted blue line on the left shows screenshots captured at our smartphone and transmitted without any compression. All other images receive some form of compression which yields an output file size below $1\,\mathrm{MB}$. While the relationship seems linear for the majority of the cluster on the left side, outliers in our data set with large file sizes are compressed more, shown by the orange dots on the right. This behavior can be explained by a compression mechanism we identified in the Signal source code [21], which iteratively downscales images until the resulting file size is below a certain threshold. Images below the file size threshold of $1.5\,\mathrm{MiB}$ do not receive a noticeable compression, shown by the top left data points in blue. While Signal keeps the original file format, WhatsApp and Telegram convert images to the JPEG format. Furthermore, after compression, the 1080×2220 screenshots have a fixed resolution of 996×2080, 4k images are compressed from 3840×2160 to 2048×1152, the DSLR images from 3648×2432 to 2048×1365, and the photos from the internal camera from 2048×1536 to 1280×960 pixels. In Figure 4b, the compression of images using different MIM applications is depicted. The result for Signal (SI) is shown in blue, for Telegram (TE) in orange, and for WhatsApp (WA) in green. For the compression results, we formulate a linear fit in the



(a) Image compression using Signal. (b) Image compression by MIM apps. (c) Video compression by MIM apps.

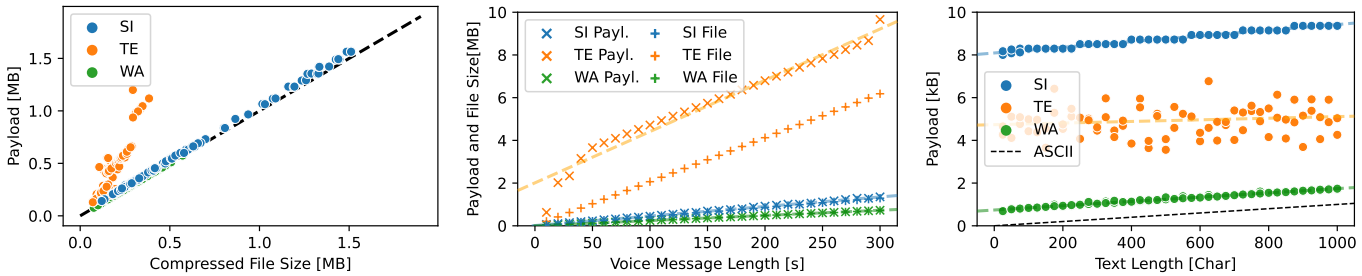
Fig. 4: Image and video compression using MIM applications (SI: Signal, TE: Telegram, WA: Whatsapp).

form of $f(x) = x \cdot c$ with x as the original file size and c as the compression rate for each MIM app, shown by the light lines in the figure. For Signal, we obtain $c = 0.0956$, for Telegram $c = 0.0391$, and for WhatsApp $c = 0.0582$. While a linear trend is evident for typical file sizes, the outliers in our test data set featuring larger file sizes indicate the existence of file size limits enforced by MIM apps, leading to more pronounced compression for these files. This behavior is already noted for Signal [21]. Generally, the compression of most images remains below $1.5\,\mathrm{MB}$, independent of their original file size.

Video Compression Analysis: Figure 4c shows the compression of $5\,\mathrm{s}$ video files for different MIM applications. Note: no results could be obtained for 60 FPS 4K videos for WhatsApp and Telegram, due to lack of support. The dashed black line in the Figure has a slope of one. We see many data points along that line because often videos do not receive any compression, even beyond an original file size of $20\,\mathrm{MB}$, the limit on the y-axis is only used for better visibility of the other results since our maximum non-compressed file size is $27.52\,\mathrm{MB}$. In WhatsApp, most of the 1080p videos are not compressed. While we could not find definitive policies without any source code, it seems that there is a file size threshold for each resolution. If this threshold is exceeded, the videos are compressed to a target bit rate which is the same across all video formats. For WhatsApp and videos of length $5\,\mathrm{s}$, this results in a file size of $1.3\,\mathrm{MB}$, depicted by the green dotted line, or a bit rate of $2.08\,\mathrm{Mbit/s}$. This result is in line with previous findings from the literature [7]. For Signal, a similar behavior is observed. However, only 38,6 % of the videos are compressed because Signal does not compress 60 FPS videos. The target file size obtained from the figure is $1.55\,\mathrm{MB}$ or a bit rate of $2.48\,\mathrm{Mbit/s}$. The threshold for Telegram is lower at $0.5\,\mathrm{MB}$ or a bit rate of $0.8\,\mathrm{Mbit/s}$. Furthermore, Telegram does compress all video resolutions with the exception of 360p.

Image and Video Overhead Analysis: While the compression and the actual size of the media file have the most significant impact on the resulting payload, it is crucial to examine the compressed file size and compare it to the payload. The reason is that each message is transmitted with additional metadata overhead. Such overhead can include, among others, information about the message receivers, timestamps, and potentially other application-specific message overhead information, which we cannot access due to encryption. Thus, the measured payload is always larger than the observed media file size on our receiving test device. Figure 5a depicts the payload on the y-axis in relation to the compressed file size on the x-axis for the different MIM apps. It can be seen that for Signal and WhatsApp, the data points are slightly above the dashed black line which has a slope of one. We obtain an average overhead of $34.11\,\mathrm{kB}$ for Signal and $1.84\,\mathrm{kB}$ for WhatsApp. In addition, the figure shows a different behavior of Telegram. There, significantly more data than the actual file size is sent to the network. The average overhead is $238.20\,\mathrm{kB}$. A potential reason for the overhead could be that multiple representations at, for example, different resolutions are sent to and stored at the server. For video payloads, the same behavior applies, however, we obtain considerably larger overheads. An average overhead of $131.73\,\mathrm{kB}$ for videos transmitted with Signal is obtained. This is more than the $6.56\,\mathrm{kB}$ for WhatsApp, but both achieve a constant small overhead in comparison to the file size. On the other hand, the measurements for Telegram achieve an average overhead of $630.88\,\mathrm{kB}$ which is significantly more than the compressed file size for some data points.

Voice Message Analysis: Next, we analyze voice messages in more detail in Figure 5b. The figure shows the payload and file size for audio messages based on the message length in seconds. The measured payloads are depicted with the marker x, while the file sizes are depicted with $+$. Both are shown on the y-axis, whereas the message length is depicted on the x-axis. Additionally, a linear regression is performed on the payloads for each MIM app and depicted by the dashed lines. For Signal and WhatsApp, the data points for file sizes and payloads overlap and their symbols combine to stars because of only small differences. The average overhead is $35.47\,\mathrm{kB}$ for Signal, $6.11\,\mathrm{kB}$ for WhatsApp, and $2.53\,\mathrm{MB}$ for Telegram. Again, the overhead during voice message transmission via Telegram is notably higher in contrast to the other apps. However, in comparison to the image compression investigated in Figure 5a, we observe a predominantly linear increase. With linear regression, we obtain equations composed of two



(a) Payload in relation to compressed image size. (b) Voice message file size and payload. (c) Payload of text messages.

Fig. 5: Payload of images, voice, and text messages (SI: Signal, TE: Telegram, WA: Whatsapp).

summands where the message length s is multiplied by the slope, i.e the bit rate, and the intercept represents the estimated message overhead. Consequently, we obtain

$$f_{SI}(s) = s \cdot 35.68\,\text{kbit/s} + 12.46\,\text{kB}, (R^2 = 0.9989) \quad (1)$$

$$f_{WA}(s) = s \cdot 19.36\,\text{kbit/s} + 3.03\,\text{kB}, (R^2 = 0.9999) \quad (2)$$

$$f_{TE}(s) = s \cdot 192\,\text{kbit/s} + 2011\,\text{kB}, (R^2 = 0.9647). \quad (3)$$

Text Message Analysis: Finally, we investigate the behavior when text messages are transmitted with MIM apps. Figure 5c shows the payload of the text messages with respect to the text length, i.e., the number of characters. The black line shows the raw text size if we assume ASCII or UTF-8 encoding for our basic latin characters which typically receive 1 B per character. Again, we perform a linear regression to fit our data. For WhatsApp, the resulting payload is the smallest and increases linearly with the text length. The equation $f(c) = c \cdot 1.33\,\text{kB} + 0.74\,\text{kB}$ describes this relationship for a text with c characters with a R^2 of 0.9879. For Signal, we derive similarly $f(c) = c \cdot 1.33\,\text{kB} + 0.74\,\text{kB}$ ($R^2 = 0.9741$). However, a step-wise payload increase can be seen in the figure. Thus, cipher blocks of fixed lengths are potentially used in the encryption scheme of Signal. In contrast, it is challenging to derive the payload size from the text length using Telegram, as it is manipulated during the transmission process. The resulting average payload size, however, is 4.47 kB.

With these models, the resulting payload can be estimated for a given media or message that shall be transmitted using a MIM app. Despite acknowledging the dynamic nature of MIM app development and their resulting network loads over time, our described methodology is a general approach, easily applicable to measure generated traffic in the present or future, with these applications or others.

VI. Media File Size Investigation

After modeling compression rates and message overheads for the different file sizes, the question about the total transmitted payload using MIM apps in reality remains. Therefore, we conduct a second user study to investigate the file size distributions of media received when WhatsApp is used. This investigation is essential for realistic message size modeling.

A. User Study for WhatsApp Media Sizes

In our WhatsApp media size study on a university campus in June 2023, 51 individuals voluntarily connected their smartphones to a notebook. A concise script has then been employed to extract the file sizes of image, video, and voice files directly from the respective WhatsApp media folder on their Android devices. Thus, we can obtain the file sizes of any received media previously compressed and sent over the network. Participants were informed about the privacy policy that ensures capturing of only anonymized meta-data.

B. Media File Size Modeling

While we did not obtain data on text lengths, this information can be derived from the message length distribution from [7]. For media files, we obtain the actual file sizes which have been sent or received by users for 17,432 videos, 375,820 images, and 168,952 voice messages from our user study in Sec. VI-A. The file sizes (in MB) follow an exponential distribution with $\lambda = 0.159$ for videos ($R^2 = 0.992$), $\lambda = 4.686$ for images ($R^2 = 0.895$), and $\lambda = 11.278$ for voice messages ($R^2 = 0.998$). Those models are an alternative approach to derive the message payload, when combined with our results for the message overhead from Figure 5.

VII. Theoretical Modular STM Development

Given the general contact structure of MIM users and their messaging behavior, theoretical STMs can be developed by using the results of our conducted studies with the following limitations and assumptions. The description of our modular approach requires independent relationships across messages or users during the modeling process. In reality, message content, social factors, and other events which influence users are factors, but cannot be accounted for in our evaluated and modeled data. Nonetheless, we present an approach to generate and populate MIM chat groups and their messages for 24 h according to our analysis and available data in the following. While our description is rather sophisticated, we believe this level of detail is necessary since the total network load is heavily influenced by the message replication process to all receivers. Moreover, this allows us to model traffic patterns for individual users, for example in access networks. The traffic generation process consists of three steps, shown in

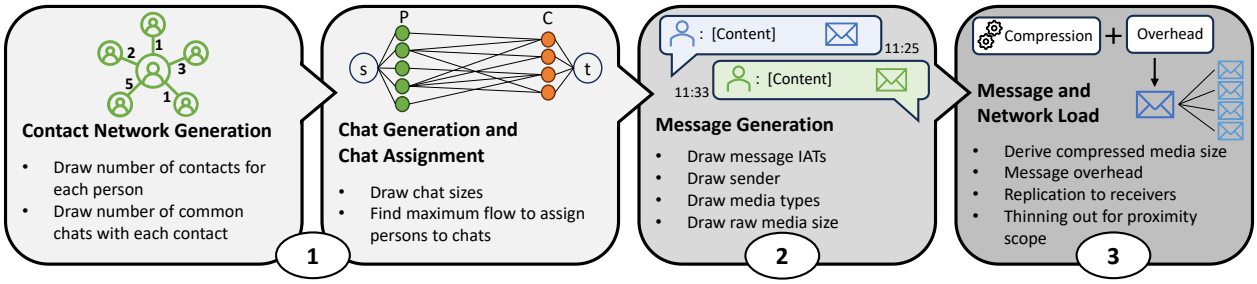


Fig. 6: Steps of the theoretical approach to STMs.

Figure 6: (1) the contact network generation consisting of the generation of the raw contact network of a person and their chats, (2) message generation, and (3) message and network load. With our presented study results, each module is sufficiently modeled. An advantage of our modular approach is that each module can be exchanged for a simpler or even more sophisticated module. For example, instead of choosing a media type for a message from the ratio of media types, the likelihood of consecutive message types, e.g., a media message following a text message [7], could be used. Furthermore, this could be extended to the likelihood of each message type following any other message type. Alternatively, modules can be exchanged by original data for a more data-driven approach. For example, our second module could be exchanged for specific message histories from the data set in [9]. Use cases for such STMs lie in the applicability for network resource planning and simulations to, for example, investigate potential benefits of edge caching or D2D communication.

A. Contact Network Generation and Chat Assignment

For the contact graph generation, we model a bipartite graph. On the left, we generate persons which need to be assigned to chats on the right. Given a fixed population size, we generate a person as a node in our contact graph and assign the number of contacts as edges for that contact from the distribution presented in Sec. III. Furthermore, we assign the number of shared chats with each contact as the edge weight according to the exponential fit in Figure 2. On the other hand, we generate groups with their group sizes according to the data available in [7]. The number of incoming edges represent the group size. Without loss of generality, we generate groups until the number of total incoming edges matches the product of the number of outgoing edges of all persons and their weights. Then, the task is to add persons to the groups while fulfilling the weight constraints for each person as best as possible, since a perfect solution might not be possible. Thus, we use a simplification which allows us to convert our task to a maximum flow problem. Therefore, we explode the weighted edges. This means that an edge with weight of three would be replaced by three edges with a weight of one. Unless persons need to be modeled with their social group, e.g., if they are simulated as a moving group together in a scenario where social group mobility plays a role, it is not important whether a person shares, for example, three chats with the same person or three different persons. Thus, we can arbitrarily connect outgoing edges of the persons, which represent what we call a *single contact equivalent*, to incoming edges of the groups. The maximum flow problem is defined as follows.

TABLE I: Parameters of hyperexp. distribution to model IATs.

$range_i$	0 s – 100 s	101 s – 1 h	1 h – 24 h	> 24 h
p_i	0.7473	0.2019	0.0465	0.0043
λ_i	0.0480	0.0013	$4.7541 \cdot 10^{-5}$	n.d.
s_i	-0.0886	0.1292	$-9.7290 \cdot 10^{-2}$	n.d.
r_i^2	0.9920	0.9766	0.9730	n.d.

Let $N = (V, E)$ be a network with vertices V and edges E. Additionally, there are $s, t \in E$ as the source and sink of our network N, respectively. Furthermore, there are the two subsets $G, P \subseteq V$ and $G \cap P = \varnothing$, where P represents the persons and G the groups. Each person $p \in P, P \subset V$ is connected to the source s via an edge (s, p). Its capacity $c_{s,p}$ is the sum of *single contact equivalent* for that person. Each group $g \in G, G \subset V$ is connected to the sink t via an edge (g, t). Its capacity is $s_g \cdot (s_g - 1)$, where s_g represents the group size of g. From each person to each group exists a single edge, representing the possibility of the person belonging to that group. The capacity of the edge is $(s_g - 1)$, since this is the amount of *single contact equivalent* that would be used for the person if it would join this group. For example, joining a group of size five means that there will be four other members for each of which one *single contact equivalent* is used. Since the membership in a group is a binary relationship, as one is either a member or not, the flow $x_{p,g}$ over an edge (p, g) needs an additional constraint, i.e., $x_{p,g} \in \{0, c_{p,g}\}$, to be either zero or the edge capacity. Other than that, only basic flow conservation constraints are necessary which are found in any max-flow problem literature, e.g., in Edmonds' and Karp's work [22] on their algorithm for max-flow problems. The matching is performed by maximizing $\max \sum_{j:(s,j)\in E} x_{s,j}$ where x represents the flow between the vertices.

B. Message Generation

Message generation is executed on chat level. From [7] we obtain a mean IAT for a given chat size. Optionally, the IAT can be adapted for the hour of the day, since, e.g., during the night less messages are sent. However [7] report that the general IAT follows a beta prime function without a well-defined

mean. To generate IATs with respect to the mean IAT from the chat size, we fit their data from [9] in multiple intervals for the exponential function $f(x) = \lambda \cdot e^{-(s+\lambda x)}$, obtaining a hyperexponential distribution $f(x) = \sum_{i=1}^{3} p_i \cdot \lambda_i \cdot e^{-(s_i+\lambda_i x)}$ in the IAT range up to $24\,\mathrm{h}$ and ∞ for messages above that range. The ranges, parameters, and R^2 values for the fit are presented in Table I. It is important to emphasize that during the IAT generation, values that are not within the determined range r_i must be omitted. Now, the message can be linked to a specific chat member, e.g., according to a participation distribution [7], since not all members are equally active within a chat, but usually few members generate most messages.

C. Message and Network Load

Then, the final task is to get the message size. A message type has to be drawn from an overall media type distribution. A more sophisticated model may even consider the previous message type. Both options are reported in [7]. The media size can be generated from original file sizes at the sender by using our compression models from the previous section or from the WhatsApp media file size distributions presented earlier. Afterwards, the modeled overhead and payload for the different MIM apps must be accounted for. Finally, to achieve the total network load, the replication due to the number of chat members has to be considered. If the perspective is global, the replication factor is the number of chat members subtracted by one, i.e., the sender. If the scope is within a given proximity, the achieved user data presented in Figure 3 can be used to exclude receivers that are not within the network's scope.

VIII. Conclusion

The ability for global communication at any time of day via Mobile Instant Messaging (MIM) has changed the way people communicate. Sharing text and media messages which are multiplied and sent to each recipient strains the underlying network. However, due to end-to-end encryption, valuable insights for network operators are hard to obtain. To address this issue, we identified the building blocks of a source traffic model (STM) for MIM and scouted existing research to identify missing pieces. We fill the gaps through user studies and measurements that we present in this paper. More precisely, we investigated contact networks to identify the number of chats of a person and the relationship to contacts across multiple chats. Further we obtained an estimate of the spatial proximity of users during messaging. Using network measurements to investigate media compression and message payload, we could identify differences among the used MIM apps and the type of message that leads to different characteristics in generated network traffic. Further, we reported on media file size distributions in MIM. Finally, building on these contributions and existing literature, we presented a theoretical, modular approach to STM generation which can be used for traffic modeling with MIM. With this, we provide a comprehensive description of MIM in the network researching context.

In future work, enabled through our modular approach to STM, we plan to investigate new ways of efficiently managing the network load generated by MIM apps, for example through edge caching or D2D communication.

Acknowledgement

This work was partly funded by Deutsche Forschungsgemeinschaft (DFG) under grants HO 4770/7-1 and SE 3163/1-1, project number: 442413406, as well as SE 3163/3-1, project number: 500105691. The authors alone are responsible for the content.

References

[1] Statista, "Number of Mobile Phone Messaging App Users Worldwide from 2019 to 2025," 2023, accessed: 2024-02-07. [Online]. Available: https://www.statista.com/statistics/483255/number-of-mobile-messaging-users-worldwide/

[2] E. Sahafizadeh and B. T. Ladani, "A Model for Social Communication Network in Mobile Instant Messaging Systems," *IEEE transactions on computational social systems*, 2020.

[3] M. Seufert *et al.*, "Potential Traffic Savings by Leveraging Proximity of Communication Groups in Mobile Messaging," in *International Conference on Network and Service Management*, Rome, Italy, 11 2018.

[4] M. Seufert, A. Schwind, T. Hoßfeld, and P. Tran-Gia, "Analysis of Group-based Communication in WhatsApp," in *International Conference on Mobile Networks and Management*. Springer, 2015.

[5] M. Seufert, T. Hoßfeld, A. Schwind, V. Burger, and P. Tran-Gia, "Group-based communication in WhatsApp," in *Networking Conf.* IEEE, 2016.

[6] K. Garimella and G. Tyson, "Whatsapp, doc? a first look at whatsapp public group data," *arXiv preprint arXiv:1804.01473*, 2018.

[7] A. Seufert, F. Poignée, M. Seufert, and T. Hoßfeld, "Share and Multiply: Modeling Communication and Generated Traffic in Private WhatsApp Groups," *IEEE Access*, 2023.

[8] A. Seufert, F. Poignée, T. Hoßfeld, and M. Seufert, "Pandemic in the digital age: analyzing WhatsApp communication behavior before, during, and after the COVID-19 lockdown," *Humanities and Social Sciences Communications*, 2022.

[9] A. Seufert *et al.*, "WhatsApp Data Set," 2023. [Online]. Available: https://figshare.com/articles/dataset/WhatsApp_Data_Set/19785193

[10] D. L. Fiscone *et al.*, "Network Forensics of WhatsApp: A Practical Approach based on Side-channel Analysis," in *International Conference on Advanced Information Networking and Applications*. Springer, 2020.

[11] C. Shubha, S. Sushma, and K. Asha, "Traffic Analysis of WhatsApp Calls," in *Int. Conf. on Advances in Information Technology*, 2019.

[12] R. Cents and N.-A. Le-Khac, "Towards A New Approach to Identify WhatsApp Messages," in *International Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020.

[13] S. Keshvadi, M. Karamollahi, and C. Williamson, "Traffic characterization of instant messaging apps: A campus-level view," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020.

[14] Z. Erdenebaatar *et al.*, "Analyzing Traffic Characteristics of Instant Messaging Applications on Android Smartphones," in *Network Operations and Management Symposium*, 2023.

[15] ——, "Instant Messaging Application Encrypted Traffic Generation System," in *Network Operations and Management Symposium*, 2023.

[16] Z. Ruidi, "4K Image Resolution Enhancement Artifacts Database," Apr. 2023.

[17] A. Ignatov *et al.*, "DSLR-quality Photos on Mobile Devices with Deep Convolutional Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[18] A. Stergiou and R. Poppe, "Adapool: Exponential adaptive pooling for information-retaining downsampling," *IEEE Transactions on Image Processing*, 2022.

[19] Wasai LLC, "Lorem Ipsum," 2015, accessed: 2024-02-07. [Online]. Available: https://loremipsum.io/

[20] P. Tracy, "Office Ipsum," 2015, accessed: 2024-02-07. [Online]. Available: http://officeipsum.com/

[21] C. Henthorne, "Signal-Android - Push Media Constraints," 2021, accessed: 2024-02-07. [Online]. Available: https://github.com/signalapp/Signal-Android/blob/main/app/src/main/java/org/thoughtcrime/securesms/mms/PushMediaConstraints.java#L99

[22] J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the ACM)*, 1972.