

Modeling Network Load of Mobile Instant Messaging: A Modular Source Traffic Generator

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

Abstract—Mobile Instant Messaging (MIM) applications such as WhatsApp transformed human communication by enabling global exchange of various message types, such as text, image, video, or voice, at any time. Network providers are facing a substantial user base and network load which is especially high in group chats where each message needs to be distributed to each member. Due to end-to-end encryption, network operators must obtain knowledge about the communication and the resulting load on the network by other means, which makes it necessary to model the network traffic of MIM. In this work, we therefore present an approach to source traffic modeling for MIM. After identifying the building blocks of a Source Traffic Model (STM) for MIM, we address existing gaps through studies on MIM communication networks, contact proximity, media compression and payload size, as well as media file size distribution. Combining existing literature and our work, we present and implement a modular STM approach which can be used for developing STMs for MIM. Using an exemplary STM, we evaluate the daily network traffic per user. With this, we provide a comprehensive description of MIM in the network researching context and enable consideration of 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, there is still information missing on the communication network and the generated network traffic of typical media sizes, which prevents the development of STMs.

Our goal is to provide an approach, along with an implementation, 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. To achieve this, we analyze the literature to identify existing work and missing aspects towards a STM. For each missing aspect, we collect data to fill these gaps, investigating user contact networks and generated network traffic. Merging existing work and newly collected data, we are able to provide a 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 active contacts of a user, the number of chats and their sizes, as well as on the proximity to other contacts. These valuable insights facilitate comprehensive modeling of communication

Received xx xxxx 202x; revised xx xxxx 202x; accepted xx xxxx 202x.
Date of publication xx month xx; date of current version 04 November 2025.
(Corresponding Author: Fabian Poignée.)

Fabian Poignée, Anika Seufert, Frank Loh, and Tobias Hößfeld are with the Chair of Communication Networks, University of Würzburg, Würzburg, Germany (e-mail: firstname.lastname@uni-wuerzburg.de).

Michael Seufert is with the Chair of Networked Systems and Communication Networks, University of Augsburg, Augsburg, Germany (e-mail: michael.seufert@uni-a.de)

structure with MIM apps and can also assist in the development of improved message and media distribution solutions for chat groups. Additionally, through 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 construct an approach to modular STM generation, with 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 network research. Additionally, we provide and evaluate an implementation of a modular source traffic generator and evaluate the generated contact networks as well as provide a first analysis on the daily generated network traffic per person.

This work presents an extended version of our previous paper [2]. Our previous efforts concentrated on modeling all parts of a STM to present a theoretical approach for MIM STM development. In this work, we extend our contact network modeling to distinguish between active and inactive contacts of a user and provide an explicit fit for the group size distribution. Together with our updated modeling of inter-arrival times (IATs) which allows IATs beyond 24 h, we enable MIM message generation of arbitrary periods. Furthermore, we provide an implementation of a modular source traffic generator and evaluate the generated contact network of different user population sizes, as well as the resulting message and network load for an exemplary user population.

In the remainder, Sec. II presents background and related works. We investigate the MIM contact network in Sec. III, estimate contact 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 our approach to modular STM development and our implementation, as well as the evaluation of the network load for an exemplary STM 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 all the gaps and finally, develop an 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 [3], 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 [4], 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 [5], 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, [6] 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 [7] 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. This methodology of collecting public WhatsApp group chats is also employed by [8], who analyze communication in 141 and 364 public political

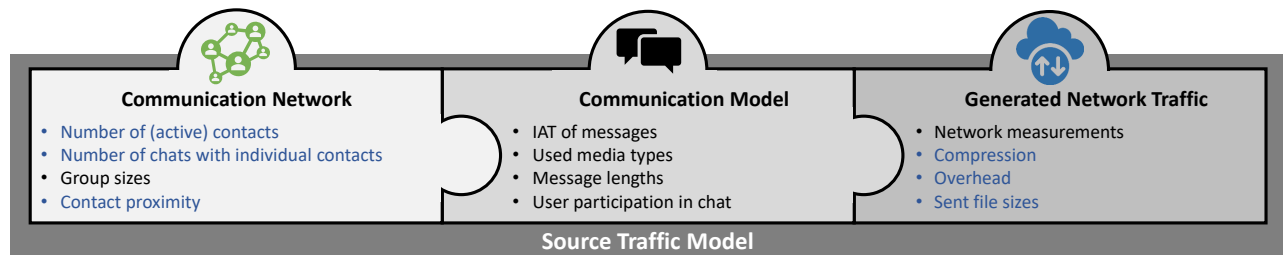


Fig. 1: Components of a STM (black text: available in the literature, blue text: missing in the literature).

groups. Their focus includes the number of messages per content type, category, and image propagation, as well as the network structure among groups and users. Additionally, usage patterns for 100 WhatsApp users with the aim of predicting demographic characteristics are investigated in [9]. The most detailed MIM communication model so far is from [10], [11], where authors present an extensive data set [12] 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 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 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. Most MIM apps use lossy compression for images and videos. This method reduces the file size by removing some of the data, especially those that are less perceivable to humans. For instance, JPEG compression for images and H.264 or H.265 codecs for videos are commonly used. While this might slightly impair the quality of the media, it significantly decreases the file size and thus the time for upload and download. Unfortunately, for most MIM apps, the compression standard is not publicly available. The researchers in [13] conducted a forensics investigation on video files sent through popular instant messaging applications. A comparison of the parameters of received video files transmitted through Facebook Messenger or Telegram revealed that these files experienced significant degradation in both audio and video parameters. Conversely, when transferred through QQi, a popular MIM app of the Chinese company Tencent, or WhatsApp, the audio and video parameters of the files remained unaffected. As consumer expectations for media quality continue to grow, it is anticipated that users will likely demand media quality in the future that closely resembles the original. Therefore, it is not sufficient to determine approaches to minimize the load on the network at the application level.

Consequently, network providers also need to deal with this increasing load.

In the past, researchers already investigated the traffic generated by MIM. The study conducted by [14] involved the monitoring of network traffic from an average of three million users over a one-month period. Their analysis focused on identifying WeChat traffic and user behavior during video chats through traffic analysis. The researchers revealed daily and weekly usage patterns, noting that 20% of users contributed to 95% of the video call traffic. Additionally, they observed a power-law distribution in calling times, with 96.5% of conversations lasting less than five minutes. Similarly, [15] examined the traffic of 603,000 WeChat users to extract temporal patterns, flow characteristics, and message intervals. They identified nine distinct usage patterns and analyzed the performance of media flows. The landscape changes when examining WhatsApp due to its implementation of end-to-end encryption since 2016. This encryption significantly limits possibilities to investigate user behavior and traffic patterns. In [16], 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 [17], 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 [18]. The authors of [19] 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 [20], [21], 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, as well as the ratio of contacts a user is in active contact with. 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 from the browser extension to the database 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 $E \subseteq \{\{x, y\} | x, y \in V \text{ and } x \neq y \text{ and } \text{chat}(x, y)\}$ 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 [10]. As such, we assume that [10], 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

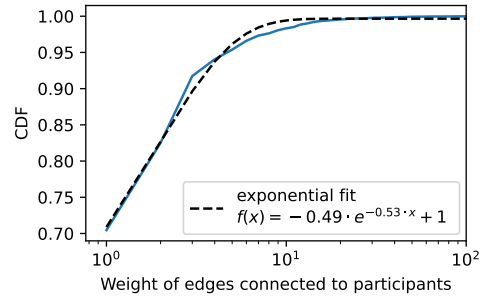


Fig. 2: Number of shared chats with contacts.

and is validated by our data. For completion, we provide the fit obtained from their data set [12], since it contains more chat samples (5956). For group sizes larger than two, i.e., non-dyadic chats, we obtain the fit of the cumulative distribution function (CDF) $F(x) = -0.6559 \cdot e^{-0.508 \cdot x} + 0.9705$ which achieves an R^2 score of 0.94. However, a Kolmogorov-Smirnov (K-S) test between the fit and empirical data rejects the null hypothesis, that is, the hypothesis that the fitted distribution could have generated the empirical data, with $p < 10^{-5}$. 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 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. Additionally, we perform a K-S test which yields $p = 0.3526$. Thus, we do not reject the hypothesis that the data follows the presented exponential distribution. 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 K-S test rejects H_0 with $p < 10^{-5}$. 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 connected 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. In complex real networks, exponential relationships can often be found, for example for the node degree distribution [22]. We find this relationship in our data if we interpret the weight $w(e)$ of the edges in G as the number of multi-edges between to vertices in G' . 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.

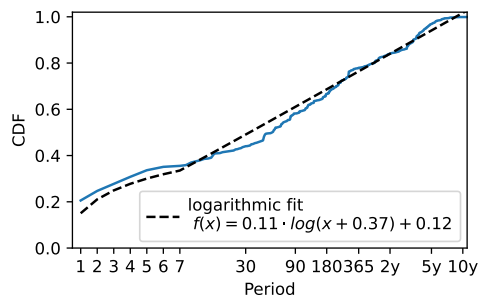


Fig. 3: Average ratio of active contacts.

A reason for the rejection of the null hypothesis in two of the three presented K-S tests is the fact that the sample sizes in those scenarios are large. While the fit presented in Figure 2 achieves a higher R^2 of 0.99 the K-S test rejects H_0 whereas the fit for the number of contacts of a person achieves a lower R^2 of 0.97 and the K-S test fails to reject H_0 . The sample size for the number of contacts of a person is 48, whereas Figure 2 has 23,452 underlying samples. Since this effect of high sensitivity for large sample sizes is well-known [23], [24] we focus on reporting R^2 scores for the regression models on our empirical data.

C. Modeling MIM Contact Activity

Even though MIM facilitates messaging between all persons in their contact network, the number of contacts a person is in active contact via MIM is lower. We are able to account for activity between persons with the information about the last message of a chat from our user study. The relationship between a participant and one of their contacts is deemed active within a given period if a chat connecting them was active, i.e., had at least one message, during that period. With this, we can obtain the average ratio of active contacts of a person for any period. The corresponding CDF is shown in Figure 3 for the range from one day up to more than ten years. Within the last 24 h, our participants have been in active contact with on average 20.57% of their contacts. Increasing that period to a week, only increases the ratio of active contacts to 35.54% of all contacts, which shows that users only communicate with a fraction of their contacts regularly. The relationship follows a logarithmic distribution for which we obtain a fit which achieves an R^2 of 0.97. Unfortunately, it is impossible to determine whether there is still an existing social connection between contacts that have not been in active contact for years. On the other hand, [10] confirms that chats exist without activity for multiple years before resuming activity. Thus, taking into account the contact activity is required when modeling MIM communication of a given period.

IV. ESTIMATING MIM CONTACT PROXIMITY

Since social networks and their communication span globally, an important element in modeling MIM communication is the consideration of the proximity of chat partners to, for

example, estimate how much traffic can be handled locally and thereby reduce the overall network load.

Thus, contact proximity is tackled by our study in the following.

A. User Study Questionnaire on WhatsApp Contact 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 in 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 [4]. 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 contact proximity since, unfortunately, more precise methods are highly privacy invasive and have poor practical feasibility.

B. Proximity Questionnaire Evaluation

Figure 4 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 a 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,

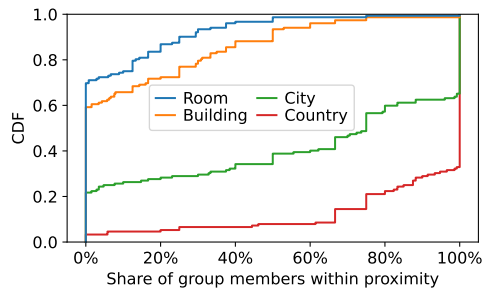


Fig. 4: Reported contact proximity at time of last message.

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.

C. Leveraging Contact Proximity

Commonly, due to the server-client architecture of most MIM applications, all traffic is transmitted to a central server and is distributed among all group members afterwards. Thus, even short-distance communications load the core network. In order to maximize the leverage that contact proximity has on network load reduction, architectural changes are necessary. For example, 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.

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 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, supersampling anti-aliasing. Thus, these data sets are 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 MB to 21 MB is provided by [25]. Additionally, we measure 30 high-quality images from a DSLR camera [26], along with native photos, i.e., smartphone camera images and screenshots, generated on our test device. From [27], we select 30 videos of 5 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 FPS and 60 FPS are investigated. Text messages of varying lengths and content are generated using text generators from [28] and [29]. 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 extend the scope to discuss the general image compression behavior of the different MIM apps. Therefore, Figure 5a 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

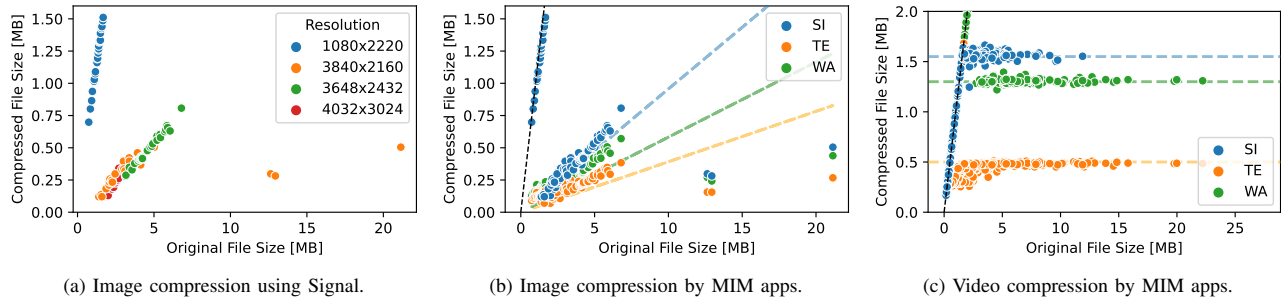


Fig. 5: Image and video compression using MIM applications (SI: Signal, TE: Telegram, WA: Whatsapp).

any compression. All other images receive some form of compression which yields an output file size below 1 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 [30], which iteratively downscales images until the resulting file size is below a certain threshold. Images below the file size threshold of 1.5 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 x 2220 screenshots have a fixed resolution of 996 x 2080, 4k images are compressed from 3840 x 2160 to 2048 x 1152, the DSLR images from 3648 x 2432 to 2048 x 1365, and the photos from the internal camera from 2048 x 1536 to 1280 x 960 pixels. In Figure 5b, 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 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 [30]. Generally, the compression of most images remains below 1.5 MB, independent of their original file size.

Video Compression Analysis: Figure 5c shows the compression of 5 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 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 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 s, this results in a file size of 1.3 MB, depicted by the green dotted line, or a bit rate of 2.08 Mbit/s. This result is in line with previous findings from the literature [10]. 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 MB or a bit rate of 2.48 Mbit/s. The threshold for Telegram is lower at 0.5 MB or a bit rate of 0.8 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 6a 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 kB for Signal and 1.84 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 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 kB for videos transmitted with Signal is obtained. This is more than the 6.56 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 kB which is significantly more than the compressed

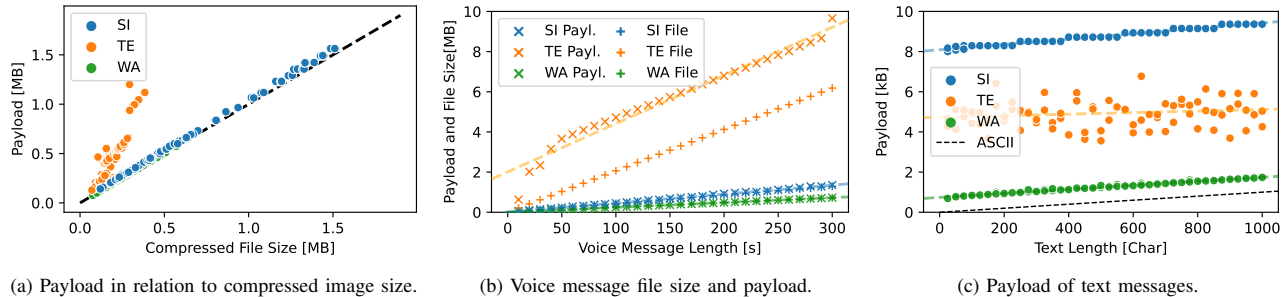


Fig. 6: Payload of images, voice, and text messages (SI: Signal, TE: Telegram, WA: Whatsapp).

file size for some data points.

Voice Message Analysis: Next, we analyze voice messages in more detail in Figure 6b. 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 kB for Signal, 6.11 kB for WhatsApp, and 2.53 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 6a, we observe a predominantly linear increase. With linear regression, we obtain equations composed of two 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 6c 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 an 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 [10]. 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 6.

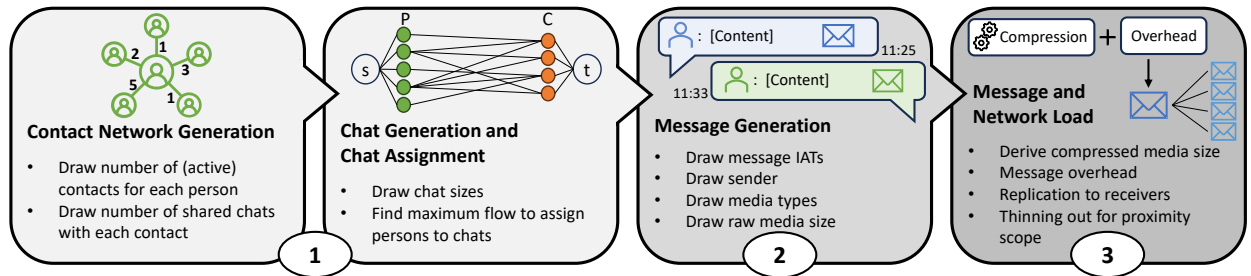


Fig. 7: Steps of STM generation.

VII. MODULAR STM GENERATION

In the following, we not only explore the theoretical foundations of modular STM development, but also present an implementation of a modular STM generator, which can be used to generate concrete STMs and their MIM traffic. 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.

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 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 STM generation process to generate MIM traffic consists of three steps, shown in Figure 7: (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 [10], could be used. Furthermore, this could be extended to the likelihood of each message type following any other message type, as provided in our implementation. 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 [12].

To adapt our approach for different messaging platforms, for example for a MIM application with different compression schemes, only the compression module would need to be adapted. Thus, the approach is MIM app-agnostic. Further-

more, the process is future-proof since changes, such as new media types or a change in contact structures, only require the affected modules to be updated based on future data. For data assessment and modeling, methodology is available in this and related work. In order to facilitate the applicability of STMs, we provide our approach to modular STM generation in the following. Moreover, we present a Python implementation of a modular STM generator in a publicly available GitHub repository¹ that can be adapted to cater for the necessary granularity for different use cases. We evaluate the generated contact networks of our generator and generate an exemplary STM to analyze the daily generated traffic per user.

A. Contact Network Generation and Evaluation

For the first step of our modular STM approach, the contact graph generation, we form a variant of a max-flow problem on a bipartite graph of persons and chats, where the resulting flow provides the assignment of persons to chats. Then, we investigate different problem sizes by using an integer linear program (ILP) to validate our approach for contact network generation purposes.

1) *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. Optionally, the number of contacts can be reduced using the average ratio of active contacts for a given period of interest. 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 share of dyadic chats of 40.41% and the group size distribution for non-dyadic groups which we presented in Section III. 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.

¹The code is publicly available at <https://github.com/linfo3/mim-stm>

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.

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 $G \cap P = \emptyset$, 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. Additionally, to guarantee that each group is completely filled if it is part of the solution, we apply the same kind of constraint for the flow $x_{g,t}$ for each edge from a group g to the sink t as $x_{g,t} \in \{0, c_{g,t}\}$. 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 [31] on their algorithm for max-flow problems. The matching is performed by maximizing $\sum_{j:(s,j) \in E} x_{s,j}$ where x represents the flow between the vertices.

To illustrate the formulation with a small example, consider a scenario with one person who has a total *single contact equivalent* of 4. This person can potentially join several groups. For a group of size $s_g = 3$, the edge capacity from the person to the group is $3 - 1 = 2$, since joining this group would consume a total of two *single contact equivalent*. On the group side, the edge capacity to the sink t is $3 \cdot (3 - 1) = 6$, as all three members of the group establish contacts with the other two members. The additional flow constraints guarantee that an assignment of a person to the group is only part of our solution if there is enough *single contact equivalent* for the person to be assigned to the group and if the group is at full size. Thus, maximizing the flow from source s to sink t is equivalent to maximizing the chat assignments of persons to groups.

2) *Contact Generator Evaluation*: For the implementation of the contact network and chat assignment, we form an ILP using basic flow preservation and our additional constraints in Pyomo [32], [33]. Since ILP solving is expensive, we

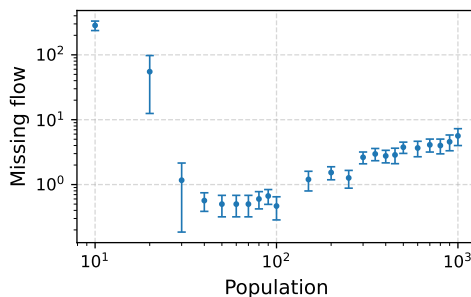


Fig. 8: Missing flow in ILP results.

use the Gurobi Optimizer [34]. To demonstrate the validity of our approach, we generate daily active contact networks, and group mappings for various population sizes from 10 to 1000 persons. For each population size we perform 30 runs, enabling the construction of confidence intervals, and evaluate how close to the optimum our approach is. An optimal solution for the max-flow problem passes all flow from source to sink, while in a suboptimal solution the missing flow results in less connections within the resulting network, since not the full *single contact equivalent* of each person can be used.

Figure 8 shows the missing flow, i.e., the flow value difference to an optimal solution, on the y-axis for the different population sizes. For each population size, mean and 95% confidence intervals are shown. Especially for small problem instances of size 10 and 20, the missing flow, is high with an average of 284.27 (22.9%) and 55.03 (1.91%) missing for those sizes, respectively. The reason stems from the group sizes. If a group size is generated which is larger than the population size, the available positions in the group can not be filled. This problem resolves itself with increasing population size. Note that the absolute missing flow is depicted and keeps below 10 for all populations of size 30 and higher. This lack of flow is less than a single group of four persons, which is equivalent to a flow or *single contact equivalent* of 12. Overall, the ILP achieves satisfactory results and is able to find a optimal solution, i.e., without any missing flow, in 23.19% of our experiments and an absolute missing flow below 10 is acceptable when considering that the populations of size 100 average a total flow demand above 13,500 which increases linearly with population size.

B. Message Generation

The second step, after generating the communication network, is message generation which is executed on chat level. From [10], 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, [10] 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 [12] in multiple intervals for the exponential function $f(x) = \lambda \cdot e^{-(\lambda x - s)}$, obtaining a hyperexponential distribution $f(x) = \sum_{i=1}^4 p_i \cdot \lambda_i \cdot e^{-(\lambda_i x - s_i)}$. The ranges, parameters, and R^2 values for the fit are presented in Table I. It is important to

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 | $4.7541 \cdot 10^{-5}$ | 0.6303 |
| s_i | -0.0886 | -0.1292 | $-9.7290 \cdot 10^{-2}$ | 0.5343 |
| r_i^2 | 0.9920 | 0.9766 | 0.9730 | 0.9855 |
| unit of time | [s] | [s] | [s] | [d] |

emphasize that during the IAT generation, values that are not within the determined range r_i must be omitted. Furthermore, for the parameters in the last column of > 24h, the unit of time is days, since operations in second granularity may lead to numerical errors depending on the programming language.

To generate messages for a given chat period and for a number of chats, the STM generator starts a simulation with a maximum simulation time which is the multiple of the period length and the number of chats. Applying inverse transform sampling, iteratively IATs are generated, the current simulation time is assigned as message timestamp, and then, the simulation time is increased by the IAT. After the simulation, each period on the simulation timeline is assigned to one chat and the timestamps are adjusted with modulo operation respectively. For example, to generate IATs within a 24 hour period for 365 chats, a simulation of a single timeline of one year would be performed and afterwards, each chat is assigned the message timestamps of one 24 h interval. Since in such a case, the IATs can be larger than the period per chat, some groups might not be assigned a message timestamp. However, if the underlying contact network is generated by using the active contacts for the given period, then the resulting chats must have at least one message during that period. The STM generator implementation is able to account for this by providing a single message in these cases.

Now, the message is linked to a specific chat member. Our STM generator assigns messages according to the participation distribution from [10], 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. First, a message type has to be drawn. Therefore, we construct a discrete-time Markov chain (DTMC) to obtain the sequence of message types. We obtain the transition probabilities between the media types of text, image, video, and audio from the dataset in [10] to determine the next message type based on the previous type. For the DTMC, an initial state distribution is required to assign the first message type for each chat. We use the stationary distribution of the chain as the initial distribution, which we obtain by constructing the transition matrix and applying the power method as described in [35]. The resulting stationary probabilities for the message types are 93.50% for text, 4.51% for image, 0.56% for video, and 1.43% for audio. Note that we could derive these values directly from the dataset as well. However, providing this functionality in the STM generator facilitates switching to a

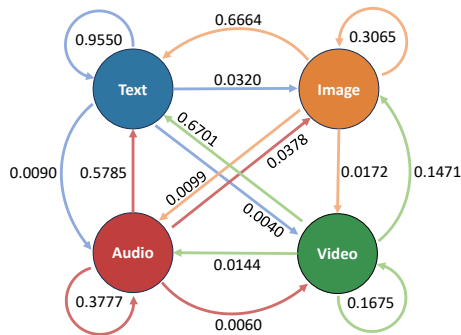


Fig. 9: Discrete-time Markov chain for message types.

different data source for other MIMs or updating with future data. The DTMC with its states and transition probabilities is shown in Figure 9. It can be seen that the probability that the next message is a text message is dominant, since the probability is 95.5% if the previous message was a text message, and is between 57.85% and 67.01% for reverting from another message type. This approach for message type generation is inspired by [5] and their activity model for group-based communication.

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 in Section VI. For our generator implementation, we decide to use the latter. For texts, we derive the UTF-8 byte equivalent directly from the message length distributions in [10] and assume a uniform size from 2–7 B [36] which is a multiple of an ASCII character that needs 1 B. Thus, we can treat each emoji as a multiple of characters and apply our WhatsApp payload and overhead model from the previous section. Note that our generator uses the media file size distributions that were obtained after WhatsApp compression and thus, we only need to apply the Whatsapp-specific overhead for media messages. However, this can easily be changed to support our models for Signal or Telegram.

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. To evaluate the resulting network load in up- and downlink within a given proximity of the sender, the user data presented in Figure 4 can be used to exclude receivers that are not within the network’s scope. For this step, a data-driven approach is used where a sample from our data set is drawn to estimate the network load within each proximity.

D. Generator Traffic Analysis

To provide insight in the average daily traffic volume per person, we present the results of our STM generator for a population of 1000 persons and their active contact network within a 24h period. We generate 30 contact networks and for each contact network we perform 30 runs of message generation. We generate messages for each chat in our network as described. On average, each of our daily active contact

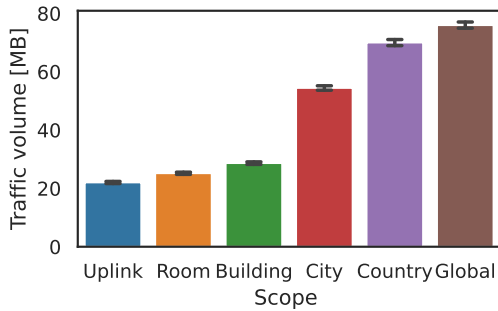


Fig. 10: Average daily traffic volume per person.

networks consists of 11,885.6 chats and the average number of messages per chat of 39.26 leads to an overall mean IAT of 36.68 min which is close to the mean IAT in [10]. On average each user generates 446.67 messages per day. While sending more than 100 messages on average may seem high to some readers, note that in most chats few users account for the majority of messages. Furthermore, the total daily traffic per user is 75.96 MB which is in line with the Sandvine [37] global internet phenomena report which reports 43 MB WhatsApp traffic volume per user per day for cellular traffic only. So roughly 56.61% of users WhatsApp traffic would occur via mobile networks and the rest via fixed connections, such as Wi-Fi or the desktop application which the authors believe to be a credible estimate. Figure 10 shows the average daily traffic volume per person for the different scopes of uplink only, up- and downlink for the different proximity scopes, as well as the global traffic volume. The black error bars at each bar mark the 95% confidence interval which are small for all scopes. On average, each user sends 22.03 MB of messages to the server. Including the downlink of each message to other users, 25.19 MB of network load occur within the same room as the sender. Those 3.16 MB of traffic could be sent via direct D2D technologies and reduce the network on the route from server to the receivers as proposed in [4]. Looking at the scope of building (28.66 MB) and city (54.43 MB) we see the potential that local caching, e.g., caching at base stations or city-wide caching, could have on the overall network load. Furthermore, we see that more than 90% of traffic originating in a country stays within that country (68.91 MB), calling for decentralized MIM server architectures.

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 contacts 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. Building on these contributions and existing literature, we present a 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. Additionally, we implement a modular STM generator, validate its contact network generation, and make its code publicly available to facilitate future research. A brief analysis of the daily network load per user for a STM from our generator shows the potential for traffic savings by exploiting proximity between contacts. In future work, we plan to investigate research questions related to this potential, i.e., reducing the network load of MIM applications by efficiently managing the network load generated by MIM apps. In particular, we will investigate edge caching to handle the increasing network load of MIM applications for which, enabled by this work, we will use a STM as a basis for our analysis.

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: 2025-03-25. [Online]. Available: <https://www.statista.com/statistics/483255/number-of-mobile-messaging-users-worldwide/>
- [2] F. Poignée, A. Seufert, F. Loh, M. Seufert, and T. Hoßfeld, "Fitting the puzzle: Towards source traffic modeling for mobile instant messaging," in *2024 15th International Conference on Network of the Future (NoF)*. IEEE, 2024, pp. 229–237.
- [3] E. Sahafizadeh and B. T. Ladani, "A Model for Social Communication Network in Mobile Instant Messaging Systems," *IEEE transactions on computational social systems*, 2020.
- [4] 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.
- [5] 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.
- [6] M. Seufert, T. Hoßfeld, A. Schwind, V. Burger, and P. Tran-Gia, "Group-based communication in WhatsApp," in *Networking Conf.* IEEE, 2016.
- [7] K. Garimella and G. Tyson, "Whatapp doc? A first look at WhatsApp public group data," in *Proceedings of the international AAAI conference on web and social media*, vol. 12, no. 1, 2018.
- [8] G. Resende *et al.*, "(Mis) Information Dissemination in WhatsApp: Gathering, Analyzing and Countermeasures," in *The World Wide Web Conference*, 2019.
- [9] A. Rosenfeld, S. Sina, D. Sarne, O. Avidov, and S. Kraus, "WhatsApp usage patterns and prediction of demographic characteristics without access to message content," *Demographic Research*, 2018.
- [10] A. Seufert, F. Poignée, M. Seufert, and T. Hoßfeld, "Share and Multiply: Modeling Communication and Generated Traffic in Private WhatsApp Groups," *IEEE Access*, 2023.

- [11] 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.
- [12] A. Seufert *et al.*, "WhatsApp Data Set," 2023. [Online]. Available: https://figshare.com/articles/dataset/WhatsApp_Data_Set/19785193
- [13] R. Verma and A. Pathania, "A Study on Video-Files Sent Through Popular Instant Messaging Applications on Smartphones for Forensics Investigation," 2021.
- [14] S. Lin, W. Zhou, and J. Liu, "Network Traffic and User Behavior Analysis of Internet-Based Mobile Messaging Applications: A Case of WeChat," in *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 2016.
- [15] Q. Deng, Z. Li, Q. Wu, C. Xu, and G. Xie, "An empirical study of the WeChat mobile instant messaging service," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2017.
- [16] 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.
- [17] C. Shubha, S. Sushma, and K. Asha, "Traffic Analysis of WhatsApp Calls," in *Int. Conf. on Advances in Information Technology*, 2019.
- [18] 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.
- [19] 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.
- [20] Z. Erdenbaatar *et al.*, "Analyzing Traffic Characteristics of Instant Messaging Applications on Android Smartphones," in *Network Operations and Management Symposium*, 2023.
- [21] —, "Instant Messaging Application Encrypted Traffic Generation System," in *Network Operations and Management Symposium*, 2023.
- [22] W. Deng, W. Li, X. Cai, and Q. A. Wang, "The exponential degree distribution in complex networks: Non-equilibrium network theory, numerical simulation and empirical data," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 8, pp. 1481–1485, 2011.
- [23] G. Nattino, M. L. Pennell, and S. Lemeshow, "Assessing the Goodness of Fit of Logistic Regression Models in Large Samples: A Modification of the Hosmer-Lemeshow Test," *Biometrics*, 03 2020.
- [24] U. Sayili and M. G. Gunver, "A Novel modification approach for the one sample Kolmogorov-Smirnov test in large sample size, journal = Scandinavian Journal of Clinical and Laboratory Investigation," 2025.
- [25] Z. Ruidi, "4K Image Resolution Enhancement Artifacts Database," Apr. 2023.
- [26] 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.
- [27] A. Stergiou and R. Poppe, "Adapool: Exponential adaptive pooling for information-retaining downsampling," *IEEE Transactions on Image Processing*, 2022.
- [28] Wasai LLC, "Lorem Ipsum," 2015, accessed: 2025-03-25. [Online]. Available: <https://loremipsum.io/>
- [29] P. Tracy, "Office Ipsum," 2015, accessed: 2025-03-25. [Online]. Available: <http://officeipsum.com/>
- [30] C. Henthorne, "Signal-Android - Push Media Constraints," 2021, accessed: 2025-03-25. [Online]. Available: <https://github.com/signalapp/Signal-Android/blob/main/app/src/main/java/org/thoughtcrime/securesms/mms/MediaConstraints.java#L45>
- [31] J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the ACM*, 1972.
- [32] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: modeling and solving mathematical programs in python," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 219–260, 2011.
- [33] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, and D. L. Woodruff, *Pyomo—optimization modeling in python*. Springer Science & Business Media, 2021.
- [34] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: <https://www.gurobi.com>
- [35] P. Tran-Gia and T. Hoßfeld, *Performance Modeling and Analysis of Communication Networks*. Würzburg University Press, 2021.
- [36] The Unicode Consortium, "Full Emoji List, v16.0," <https://unicode.org/emoji/charts/full-emoji-list.html>, 2024, accessed: 2025-03-25.
- [37] AppLogic Networks, "Global internet phenomena report," 02 2025. [Online]. Available: https://www.sandvine.com/hubfs/AppLogic_Networks/Collateral/Global%20Internet%20Phenomena%20Reports/GIPR%202025.pdf



Fabian Poignée is a Research Assistant at the Chair of Communication Networks of the University of Würzburg where he is pursuing his PhD. He received his Master's degree in computer science from the University of Würzburg in 2020. His research mainly focuses on Quality of Experience of Internet applications, AI and ML for networks, group-based communication systems, satellite networks, and network digital twins.



Anika Seufert received her Ph.D. in Computer Science from the University of Würzburg, Germany, in 2025. She previously worked as a Research Assistant at the Chair of Communication Networks at the University of Würzburg. Her research interests include user-centric modeling of mobile applications and communication systems, measuring Internet experience from the end-user perspective, and optimizing Quality of Experience.



Frank Loh received the Ph.D. from the University of Würzburg, Germany, in 2024 for his thesis Monitoring the Quality of Streaming and Internet of Things Applications. He is currently with the Chair of Communication Networks, where he leads the Green Communication Networks Research Group. His current research includes performance analysis and investigation, QoS and QoE, and energy and sustainability assessment for current and future networks. He studies a greener way of communication without impacting the user perceived quality.



Michael Seufert (Senior Member, IEEE) is a Full Professor at the University of Augsburg, Germany, heading the Chair of Networked Systems and Communication Networks. He received the Bachelor's degree (2018) in econometrics and the Diploma (2011), PhD (2017), and Habilitation (2023) degrees in computer science from the University of Würzburg, Germany, and holds the First State Examination degree (2011) in mathematics, computer science, and education for teaching in secondary schools. His research focuses on user-centric communication networks, including QoE of Internet applications, AI/ML for QoE-aware network management, as well as group-based communications.



Tobias Hoßfeld is professor at the Chair of Communication Networks at the University of Würzburg, Germany, since 2018. He finished his PhD in 2009 and his professorial thesis (habilitation) "Modeling and Analysis of Internet Applications and Services" in 2013 at the University of Würzburg, where he was also heading the "Future Internet Applications & Overlays" research group. From 2014 to 2018, he was head of the Chair "Modeling of Adaptive Systems" at the University of Duisburg-Essen, Germany. He has published more than 100 research papers in major conferences and journals, receiving several best conference paper awards, 3 awards for his PhD thesis, and the Fred W. Ellersick Prize 2013 (IEEE Communications Society) for one of his articles on QoE. He is member of the editorial board of IEEE Communications Surveys & Tutorials, Springer Quality and User Experience, ACM SIGMM Records and elected chairperson of the ITG/VDE expert group "Communication Networks and Systems" within the German society of Information Technology (ITG).