Dismissed! A Detailed Exploration of How Mobile Phone Users Handle Push Notifications

Martin Pielot
Telefonica Research
Barcelona, Spain
martin.pielot@telefonica.com

Amalia Vradi
University of Macedonia
Thessaloniki, Greece

Souneil Park
Telefonica Research
Barcelona, Spain
souneil.park@telefonica.com

ABSTRACT
We analyzed 794,525 notifications from 278 mobile phone users and how they were handled. Our study advances prior analyses in two ways: first, we systematically split notifications into five categories, including a novel separation of messages into individual- and group messages. Second, we conduct a comprehensive analysis of the behaviors involved in attending the notifications. Our participants received a median number of 56 notifications per day, which does not indicate that the number of notifications has increased over the past years. We further show that messaging apps create most of the notifications, and that other types of notifications rarely lead to a conversion (rates between ca. 15 and 25%). A surprisingly large fraction of notifications is received while the phone is unlocked or the corresponding app is in foreground, hinting at possibility to optimize for this scenario. Finally, we show that the main difference in handling notifications is how long users leave them unattended if they will ultimately not consume them.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
Notifications; Attendance; Interruptibility; Mobile Devices

INTRODUCTION
In 2018, mobile phone notifications are no longer just SMS alerts. They announce all kinds of new content and have become integral means for apps and services to proactively engage with their users. However, notifications alerts are designed to grab a user's attention regardless of the concurrent activity, which not only has effects on task performance in the work context [1, 4, 7, 15, 17, 20, 28, 30], but also negative impact on our well-being [18].

When studying the effect of notifications, and how to mitigate these, it is important to have an understanding what types of notifications are most common, and how mobile phone users typically handle them. A few studies [12, 19, 25, 29] started to shed light on this. For example, in 2013, Pielot *et al.* [25] found that their 15 participants received an average of 63 notifications per day, and largely triaged them within a few minutes.

However, previous reports suffer from potentially inaccurate measuring methods: they either rely on self-reports or they use a measurement API[1] that does not distinguish between an actual notification alert or a notification used to display progress (e.g. "*download 63% complete*"), hence requiring heuristics to clean the data. Further, previous work on actual notifications did not explore in detail the stages in which notifications were attended, such as seeing, triaging, consuming, or dismissing them.

To fill this gap, we analyze a dataset from 278 mobile phone users, who ran a dedicated data-collection application on their mobile phones for an average of 4 weeks during the summer of 2016. The participants were recruited through a specialized agency to reflect the gender and age distribution of the Western European country (Spain) where the study took place[2]. The application logged notification activity through the *NotificationListenerService* API, which was first released with Android OS 4.3 on July 24, 2013 and not widely available in previous studies. This API has two distinct advantages: it allows to log when notifications are removed, which gives better insights into how notifications are handled, and it allows to distinguish between actual notification alerts and notification events that are "mis-used" to continuously display information in the notification bar. The main contributions are:

- updated statistics on the number of notifications that people receive per day from 5 different categories (messaging, group messaging, email, social, non-social) – being the first ever investigation considering group messages as a separate entity and using of a novel API for more accurate measurement;
- a detailed exploration into how notifications are handled, investigating 4 different types of attendance (seen, checked, consumed, removed); and
- the data and the code, appended to this paper, through which we arrived to these insights.

[1] Android's AccessibilityService
[2] In Spain, Android OS has a penetration of about 90%

MobileHCI'18, Sep 03–06, 2018, Barcelona, Spain

DOI: https://doi.org/10.1145/3229434.3229445

RELATED WORK

Iqbal and Bailey [16] define *notification* as a visual, auditory, or tactile *alert* designed to attract attention. In everyday language, the word notification may be used to describe the alert as well as a visual representation that is typically found in a pop-up or a notification center. In this paper, we will use the word *notification* to refer to the actual alert.

What Notifications Do Mobile Phone Users Receive?

In a survey from 2014 [12], the majority of respondents considered themselves to typically receive 20-50 or 50-100 notifications per day. In an *in-situ* log study on mobile phone notifications [25], participants (information workers from several Western European countries) received a medium number of 63.5 of notifications per day. Sahami *et al.* [29] report that the 10 most-popular apps generated a mean number of 44.9 notifications per user and day. In a study on smartphone overuse, Lee *et al.* [19] reported that their participants (Korean colleague students) received an average of 400 notifications, mostly from the KakaoTalk messenger. All of these studies took place in 2013. They reveal that people back then already dealt with dozens of notification alerts every day. On mobile phones, the largest chunk of notifications originate from messaging applications [19, 25, 29], such as SMS, WhatsApp, or Facebook Messenger.

Attending to Notifications - Definition

Notifications are designed to attract attention. They therefore implicitly accept the possibility of interrupting the user during other activities. Depending on the nature of concurrent activities, users may have different attitudes towards such interruptions, and they may be affected differently by them.

Early work by Horvitz and Apacible [13] quantified interruptibility by the amount of money that people would be willing to pay to avoid a disruption in a given context. Fogarty *et al.* [11] defined *interruptibility* (in the context of office workers) in terms of social conventions, *i.e.*, whether it would be appropriate for one person to interrupt another.

In the context of mobile phones, Fischer *et al.* [10] distinguished between *interruptibility* as "*more interesting from a technical systems-oriented perspective, e.g. as a trigger to an action*", and *receptivity* as also considering the (emotional) experience of the interruption. In their follow-up work [9], Fischer *et al.* further define *opportune moments* for interruption delivery as moments that "*minimise the detrimental effects of interruptions*". Pejovic and Musolesi [24] build an interruptibility model that predicts if a recipient will react to an interruption by a notification within a given time interval, and the extent to which the recipient would agree to the statement "*Is this a good moment to interrupt?*".

However, in many contexts, it may not be sufficient to only consider *whether* the user reacts to a notification, but also *how*. For example, notifications from messaging applications, such as "*okay! see you at 6pm*", do not necessarily require a reaction in form of launching the messaging app and writing a response. Pielot *et al.* [26, 25] therefore introduced the step of *attending* a notification, which is defined as an action that allows to triage a notification and to decide on the appropriate reaction. Once a notification has been triaged, "*potential social pressure, e.g., the need to respond fast, begins to manifest. Further, if the message is unimportant or not urgent, the message can be discarded here*" [26].

Clark [6] (according to [21]) suggested that a user can respond to an interruption in four possible ways: 1. handle it immediately; 2. acknowledge it and agree to handle it later; 3. decline it (explicitly refusing to handle it); or 4. withdraw it (implicitly refusing to handle it). Mehrotra *et al.* [22] divide responses to notifications into two different stages: *seen time* and *decision time*. *Seen time* refers to the time until the user unlocks the screen after having received a notification. *Decision time* refers to the time until the notification is either clicked or dismissed. Turner *et al.* [31] identified four stages of responsiveness in the context of a study where a task-reminder app frequently created notifications to ask user to check off a to-do list item: (1) *Unreachable:* the user did not react to the arrival of the notification, either because they were not interrupted or did not want to triage any notification; (2) *Reachable:* the user interacted with the device (e.g. turned the screen on) to – presumably – triage the notification, but may not have proceeded further; (3) *Engage-able:* the user was reachable and proceeded to access the notification list (which may require them to unlock the device), but may dismiss it or not proceed further; and (4) *Receptive:* the user was engage-able and proceeded to tap on and consume the notification, opening the application.

Attending to Notifications - Previous Findings

Sahami *et al.* [29] report from their large-scale study that if a user clicked on a notification in the notification center, this happened within a median time interval of 30 seconds. Unfortunately, the work does not disclose information about notifications that were dismissed. Mehrotra *et al.* [21] report that their 20 participants clicked on 60% of the notifications within 10 minutes. However, Mehrotra *et al.* did not explore the time-until-click by notification category.

Previous work further shows that how fast people handle notifications depends on the type of the notification. Pielot *et al.* [25] report that message notifications were attended significantly faster than email notifications. On average, notifications from messengers are attended within minutes [2, 8, 12, 26, 29], and people maintain this levels of attentiveness for large parts of their wake time [8]. The reason that notifications from messengers are attended quicker can be explained by the insight that, "*people are assumed to be constantly co-present, and thus, constantly available for conversation*" [3].

GOALS OF THIS RESEARCH

The goal of this research is to fill a number of gaps in existing research. First, while plenty of research investigates notifications and interruptibility, only a handful of works explored which notifications mobile phone users actually receive on a daily bases.

Second, we presume that an increasing number of companies have started the use of push notifications to engage their user base[3]. The preface of a recent symposium of Ambient Notification Environments suggests that "*Direct notifications are on the exponential rise*" [5]. Thus, one open question is to what extent the volume of notifications that people deal with on a daily basis has increased and shifted.

Third, while previous work has shown that most notifications originate from messaging apps, no publication has yet explored the difference between individual- and group messages, and how they are handled.

Forth, previous work usually only explored one type of notification attendance – usually clicking them. What is missing is a more detailed exploration into how notifications are handled in detail and across platforms.

Fifth, previous work had to rely on biased means to collect that information: self reports or Android's *AccessibilityService*. While the *AccessibilityService* allows to intercept a *NOTIFICATION_CHANGED* event, the meaning of this event can be ambiguous. For example, when Google Play updated an app, the *AccessibilityService* would report one *NOTIFICATION_CHANGED* events for each percent in the download process (1%, 2%, 3%, ...). Thus, this method has to rely heavily on manual filtering, with the risk of considering events as notifications that are not perceptible alerts.

This also allows to address the challenge of apps and services sending notifications across platforms. For example, a new email may trigger a mobile phone notification, but may be handled on a different device. In some cases, such as Gmail, handling the notification on another device triggers the notification to be removed from the mobile phone. The events can only be reliably recognized by using the *NotificationListenerService*, which was only introduced in Android 4.3, and that was not available in all previous studies. Without considering such removed events, an analysis might consider a notification as unattended whereas in reality it was attended on a different device.

METHODOLOGY

The dataset was collected in a study which took place in summer 2016. Participants were recruited via a specialized agency. We requested a sample that matches the gender and age distribution of smartphone users in Spain. The only restriction was that people were required to own an Android phone. Android phones account for the large majority ($\sim$ 90%) of the smartphones in Spain.

Data Collection

Participants were required to install a dedicated study application onto their mobile phones. This app was used to collect the data that we analyze in this study. For this analysis, we focus on *notification-posted* and *-removed* events, *screen* events (off, on, unlocks), app launches, and access of the notification drawer.

To capture these events, we asked the participants to grant two special permissions: access to accessibility events and notification events. Events from Android's *AccessibilityService* allow to track app launches and access of the notification drawer . Access to Android's *NotificationListenerService* allows to subscribe to notification-posted and notification-removed events. Screen events (off, on, unlock) where tracked by subscribing to system-wide broadcasts.

To get access to these accessibility- and notification events, users have to visit a dedicated view in the settings and manually grant access for the requesting application. Since accessibility and notification events can contain potentially sensitive information, the informed consent contained a section specifically dedicated to those events. During the setup process, the application itself explained the participants how to grant these two special permissions and automatically sent them to the correct settings view.

Procedure

The study took place from April to July 2016 in Spain. People with interest in joining the study were first directed to the informed consent, which had been approved by the legal department of our institution. The consent form listed all data to be collected in the study and gave details about potential personally-identifiable information. The participants then were taken to an installation guide that explained how to install the mobile application. We ran informal usability tests to ensure that the installation process was fast and easy to understand. The data collection commenced once the app was installed, set-up properly, and once the participants confirmed their agreement with the informed consent from within the app. To protect the identify of the participants, all data was stored in a pseudonymized way.

Participants

Participants were recruited via a specialized recruitment agency, that we tasked to provide a representative sample in terms of demographics of (Android) smart phone users in Spain. Over 500 people joined the study. For this research, we consider data from 278 participants who contributed data for at least 10 days and who had a phone with a minimum Android version of 4.3 (Jelly Bean), which is necessary, because the *NotificationListener* API was introduced with this update. The compensation was 10 EUR.

The ages of the 278 participant ranged from 18 to 66 years (M = 37.71, SD = 11.07). 146 (52.52%) participants were female, 132 (47.48%) were male. The mean number of participation days was 26.59 (Mdn = 26.5, SD = 9.57).

DATA PROCESSING

Filtering Notification-Posted Event Bursts

To ensure that we only analyze notifications events that are perceived as notification alerts by the users, we had to filter the data set. The reason is that notification-posted events reported by Android's *NotificationListenerService* do not necessarily equate to a perceptible notification alert according to the definition of Iqbal and Bailey [17]. Sometimes, the operating system creates several notification-posted events in order to

[3]https://techcrunch.com/2015/04/21/notifications-are-the-next-platform/

construct a compound notification, while at the same time, the user only perceives one alert. For example, if there is an unhandled WhatsApp message, and another message from the same contact arrives, the OS will internally generate one posted event for the new message and a second posted event that groups those two unread messages into a box with the title "two unread messages".

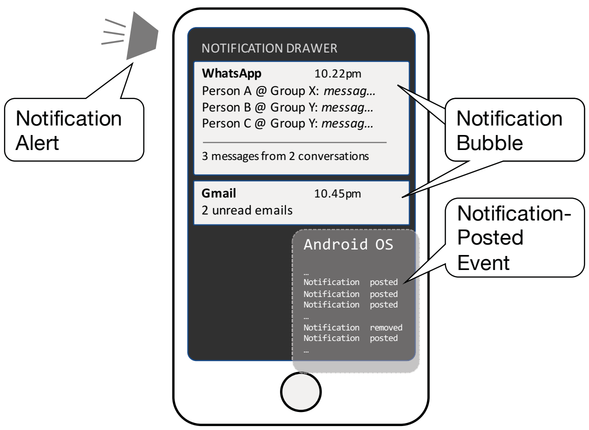

Figure 1: Notifications on iOS.

Group messages can even lead to a whole series of events for what the user perceives as a single notification alert. For example, if – as illustrated in Figure 1 – Person C just sent a WhatsApp message to Group Y, there is already one unread message from Person A in Group X, and another unread messages from Person B in Group Y, the operating system may generate 4 notification-posted events: one to create a new "box" to encompass unread messages from both groups (usually entitled "WhatsApp"), two more events to re-visualize the old notifications from Persons A and B, and a final notification-posted event to add the new notification from Person C. We will refer to this series of notification-posted events as *bursts*.

To filter out these bursts, we apply the following cleaning steps: First, we remove all notifications with the title "two unread messages", including all respective translations that we found in our data set. Second, we mark notification-posted events that occur within the same second as *bursts*. We empirically determined that the last posted event usually contains the notification alert that the user perceived. Thus, we removed all but the last posted events from each burst.

The resulting dataset contains 794,525 notifications.

Notification Categories

Some previous works investigated interruptibility without considering the type of the notification. However, Pielot *et al.* [25] found that people react quite differently to different types of notifications. Thus, we split notifications into several categories, which we explain in the following (Table 1 lists the 5 most-frequent apps for each of the categories):

- **Messaging** denotes the category of notification from messengers, *i.e.*, communication app that allow to send (short) messages to contacts. WhatsApp is the most common messenger in our data set. The rationale of this category is that (1) messages are the most frequent notifications [29, 25], (2) they are usually targeted directly at the receiver and thus usually highly relevant, and (3) that one-to-one messaging is associated with the highest pressure to respond in a timely fashion.

- **Group Messaging** denotes notifications from a messenger which was sent to a group of users. An example is a WhatsApp message received to a group. The rationale is that the user is no longer the prime recipient of the message. Thus, we hypothesize that the social dynamics of group messaging are different than those of individual messaging.

- **Email** denotes the category of notifications from email apps. The most prominent source of email notifications is Android's built-in email client. While emails, too, are text sent to one or several recipients, the cultural norms around emails differ dramatically from messaging. Emails are not always relevant for the recipient, and email senders rarely expect responses within minutes, as compared to messaging.

- **Social** denotes the category of notifications from social networks. The most prominent source of social notifications are Twitter, Instagram, and Facebook. The reason to create the category is to cluster notifications that are created by human activity, such as *Likes* or *Retweets*, but do not require a reaction by the user.

- **Non-Social** denotes the category of notifications from apps that do not facilitate the interaction between people, such as Dropbox or security apps.

The rationale for creating our own set of categories instead of using Google Play's categories was that the majority of the notifications in our dataset would be grouped into the *communication* category, which we consider too broad for our study.

Assignment to Categories

We received notifications from 796 different applications. To assign each app to one of the categories described above, we applied the following steps:

(1) create a set of all apps observed in the data set (uniquely identified by the application package); (2) for each app, try to obtain the category assigned by Google's Play Store;[4] (3) count the number of occurrences of each app – uniquely identified by the Java package name; and (4) from most to least frequently-observed app: assign a mapping from the Play Store category to our own category taxonomy (e.g. Entertainment –> NonSocial) and if that is not feasible, manually assign a category to the unique identifier (e.g. *com.facebook.orca* –> Messenger, *com.facebook.katana* –> Social).

We repeated these steps until 99.7% of the apps had a category assigned. The remaining apps were removed from the analysis.

[4]Not all apps on a phone are available on Google Play. For those apps, it is not possible to retrieve a category from Google Play

(a) Messaging

App Package Name	#	% (All)	% (Msg)
com.whatsapp	312239	39.3	89.5
com.facebook.orca	9710	1.2	2.8
org.telegram.messenger	8697	1	2.5
com.android.mms	5266	0.6	1
com.google.android.talk	3165	0.3	0.3

(b) Email

App Package Name	#	% (All)	% (Eml)
com.google.android.gm	40209	5	43.1
com.android.email	26433	3.3	28.3
com.microsoft.office.outlook	8544	1	9.2
com.samsung.android.email.provider	3720	0.4	4
com.yahoo.mobile.client.android.mail	3501	0.4	3.7

(c) Social

App Package Name	#	% (All)	% (Scl)
com.twitter.android	20014	2.5	39.8
com.facebook.katana	16858	2	33.6
com.instagram.android	8646	1	17.2
com.facebook.lite	1755	0.2	3.5
com.pinterest	776	0.1	1.5

(d) Non-Social

App Package Name	#	% (All)	% (NScl)
com.cleanmaster.mguard	2803	0.4	4.9
com.dropbox.android	2592	0.3	4.5
com.cleanmaster.security	2266	0.3	4
com.google.android.youtube	1569	0.2	2.7
com.antivirus	1548	0.2	2.7

Table 1: Top-5 Applications per Notification Category

Excluded Notification Categories

We exclude notification events from apps that cannot be considered notification alerts according to the definition of Iqbal and Bailey [17]. For example, Google's QuickSearch Box generates an internal notification event whenever the user returns to the home screen. However, no visible, audible, or otherwise perceptible alert is created. We further excluded notifications from alarm clocks or calendar reminders, as we found that users typically do not attend them by opening the app.

WhatsApp Group Notifications

Group messages are perceived differently from individual messages. In particular, they are more likely to be found a source of annoyance [27] and we hypothesize that there might be less pressure to respond timely to them.

Since there is no easy way to split messaging notifications into individual- and group messages, we built a simple heuristic to recognize group message notifications from the structure of the notification title. We focus our heuristic on WhatsApp messages only, because they account for 89.5% of the messaging notifications. In comparison, the next-most prevalent app, Facebook Messenger, is responsible for only 1.6% of the messaging notifications.

To identify group messages, we made use of the fact that titles from group notifications on WhatsApp either form the pattern “Message @ Group” or the pattern “Group: Message”. This heuristic split the 557,080 WhatsApp notifications into 312,239 individual- and 244,841 group messages.

RESULTS | NOTIFICATION VOLUME

In this section, we investigate number of notifications that our participants received on average per day. Please note that the data set and the scripts that produced the presented analysis can be found in the added material submitted alongside this paper.

To ensure that the daily statistics are not biased by individual users, we compute the average in two steps: first, for each user individually, we compute the mean number of notifications received per day for each of the five categories. Second, we compute the median over all the users for each of the category. When not considering the notification categories, this calculation indicates that the average user receives 56 notifications per day.

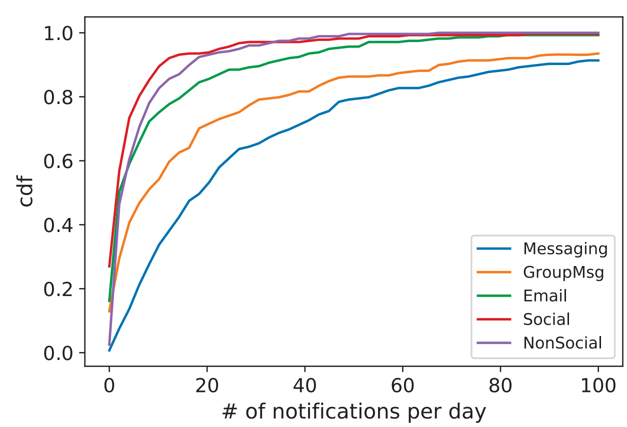

Figure 2: Notification volume of categories per user and day.

Figure 2 visualizes the average daily notification volume per category[5]. We used inferential statistics to establish whether the differences in the daily number of notifications are significant. A Levene's test revealed that variances are not sufficiently equal [$W(4,1386) = 29.92, p = 0.000$], thus we used non-parametric tests. Our participants received approximately a daily median of 18 messaging notifications, 7 group message notifications, 2 notifications from non-social apps, 2 email notifications, and 1 notifications from social networks.

A Friedman test revealed a significant effect of the notification category on the daily notification volume [$\chi^2 = 313.02, p < 0.001$]. Pairwise comparisons were done by Wilcoxon Signed-Rank tests against a Bonferroni-corrected alpha of $p < 0.005$.

[5]Please note that due to the averaging method, the sum of the daily notifications per category do not necessarily match the daily number of notifications computed without considering the notification category

Authors	Notifs	Msgs	Grp	Email	Social	NonSocial	Unit	n	Year	Measurement
Lee *et al.*	400	-	-	-	-	-	M	95	2013	AccessibilityService
Sahami *et al.*	44.9	27.5	-	8.7	4.7	4	M	40,000	2013	AccessibilityService
Pielot *et al.*	63.5	16	-	26.5	3.5	6	Mdn	15	2013	AccessibilityService
Gallud *et al.*	20-100	-	-	-	-	-	Range	114	2014	Self-Report
This work	56	18	7	2	1	2	Mdn	278	2016	NotificationListener

Table 2: Comparison with previous work.

Category	Mdn	M	SD	Comparision	W	p	δ
Messaging	18	36.6	51.9	GroupMsg	12679	0.000	0.303 (small)
				Email	6570	0.000	0.587 (large)
				Social	1893	0.000	0.748 (large)
				NonSocial	3410	0.000	0.653 (large)
GroupMsg	7	27.1	58.6	Email	10922	0.000	0.266 (small)
				Social	5552	0.000	0.426 (medium)
				NonSocial	8772	0.000	0.269 (small)
Email	2	9.7	18.5	Social	11844	0.000	0.169 (small)
				NonSocial	18025	0.412	0.068 (negligible)
Social	1	5.3	16.2	NonSocial	13606	0.000	0.255 (small)
NonSocial	2	6.1	9.3	-	-	-	-

Table 3: Descriptive and inferential statistics of the average number of notifications received per user per day.

All differences (see Table 3) were statistically significant – the only exception being that no significant difference between the volume of email and non-social notifications was found.

Table 2 shows how our statistics on the average number of notifications compare to earlier studies. The average number of daily notifications is in range with other studies, apart from the study by Lee *et al.* which was conducted amongst college students in South Korea. Another notable difference is that the amount of daily email notifications is much lower than in the study by Pielot *et al.*, which can be explained by the fact that their sample primarily consisted of information workers, who might be more likely to enable email.

RESULTS | NOTIFICATION ATTENDANCE

Definitions

Prior work on in-situ notifications typically only explored a single form of attendance, such as opening the app [21, 29] or triaging the notification [25]. We agree with Turner *et al.* [31] that this does not sufficiently reflect the underlying complexity of notification handling. In our study, we therefore treat attendance as a process that involves multiple actions and unfold the actions that can be practically measured. We elaborate on each action below:

Shown: when the screen of the device turns on while notifications are pending, the user might already notice the notification. However, since we do not know whether the user was actually looking at the screen when the notification arrived, this is not guaranteed. Hence, we do not consider this event in our analysis.

Seen: when a user unlocks the device, the user at minimum should notice the icons of applications that have notifications pending in the top-left corner of the screen. The user becomes aware that she or he will have to handle the notification ultimately. Thus, when the device is unlocked, we consider pending notifications as *seen*. If the screen was unlocked when the notification arrived, the notification is considered *seen* immediately.

Checked: when a user opens the notification drawer, it typically discloses parts of the notification, such as the sender & subject of an email or the sender & first part of a message. This information gives the user a better idea about the notification and how to handle it. At this stage, we consider the notification *checked*.

Consumed: when the user opens the app that corresponds to a pending notification, we consider the notification *consumed*. If the notification was removed before the app was opened, we do not consider the notification as consumed, and will not consider it *consumed*. Further, if the app was in foreground when the notification arrived, we exclude it from this observation for two reasons: there is a consumption in progress but it is not triggered by the notification and the way in which the notification is ultimately consumed – if at all – can vary greatly between apps.

Removed: we consider a notification *removed* if Android's *NotificationListenerService* fires a notification-removed event. Several actions may trigger this event, including (1) manually discarding the notification, (2) the notification timing out, or (3) consuming the notification on a different device. Please note that we ignore notifications that were consumed before the removed event was registered.

Notification Conversion Rates

The ultimate goal of notifications is to alert people to content provided via the app. Oftentimes, this content needs to be *consumed* usually by opening the app. In marketing research, this is often referred to as *conversion*. It is important to understand the conversion rate, *i.e.*, which fraction of notifications lead to a conversion, as it tells something about the relevance of the notification and whether it should have been sent in the first place. Please note that we consider a notification consumed if the app is opened before the notification is removed.

Table 4 shows the conversion rates per notification category. About two-third of the messaging-related notifications lead to conversions, whereas the conversion rates for the other categories are notably lower. In particular, only a bit more than 15% of Email and NonSocial notifications led to opening the app. This is an important indication for the importance and relevance of message notifications compared to notifications from other types of apps.

Category	Conversion Rate
Messaging	63.67%
Group Messaging	65.81%
Email	15.47%
Social	26.55%
NonSocial	16.19%

Table 4: Conversion Rates per Notification Category.

Immediate Attendance

Table 5 shows of notifications that were immediately seen, checked, or consumed according to our definition. For example, notification are considered immediately seen if the screen is unlocked. 20 to 35% of the notifications were received while the phone was already unlocked ('seen'). Only very few notifications (0.1% to 0.3%) arrived while the notification drawer was in foreground. We presume that the notification drawer is hardly opened so the likelihood of a notification arriving while it is open is very low. Finally, between 4.2% and 23% of the notifications arrived while the corresponding app was in foreground. The number is particularly high for individual- and group message, which intuitively makes sense as this might happen during chats.

Category	Seen	Checked	Consumed
Messaging	32.7%	0.3%	23%
GroupMsg	26.9%	0.2%	21%
Email	20.7%	0.10%	4.2%
Social	32.6%	0.1%	10.5%
NonSocial	35.6%	0.3%	8.5%

Table 5: % of immediately-attended notifications

Notification Attendance Flow

Figure 3 visualizes the attendance to the notifications of different categories through Sankey diagrams. Please note that we only consider notifications that were not immediately seen, checked, or consumed. The rationale is that we are interested in how notifications grab the user's attention while the user is focused on other things.

We chose Sankey diagrams in order to provide a holistic view of the attendance processes. Attendances can be viewed as the transition of the notifications' state between above's definitions. Sankey diagrams intuitively represent the different types of state transitions and their prevalence. The Sankey diagrams of Figure 3 aggregates notification states at four points in time: 2, 5, 10, 30 minutes since the arrival. Since previous work [25, 29] agrees that notification attendance follows a long-tail distribution, we put higher resolution on the early stage of the notifications. The complete measurements we made throughout the full lifespan of the notifications can be found from the submitted data set.

Alongside the flow analysis, we investigate whether the overall attendance times differ significantly. Since the attendance time scores follow a heavy-tail distribution, we used non-parametric statistics to check for significant differences: we used a Kruskal-Wallis followed by Bonferoni-corrected, pairwise Mann-Whitney U Tests. The full results of the analysis are part of the appendix. In this paper, we only talk about differences that are significant and have a non-negligible effect size (quantified by Cliff's Delta).

Messages are Attended Fastest

The attendance-flow analysis (Fig. 3) shows that a larger portion of the individual messages are attended, *i.e.*, either 'Consumed', 'Removed', 'Checked', or 'Seen', more quickly than the notifications of other categories: in Figure 3 (a), the flows to the state 'Unattended' are much thinner than the same flows in other diagrams. The statistical analysis confirms that messages are always amongst the fastest group to be attended.

Group Message are Attended Slower than Individual Messages

A related observation is the difference of group messages from individual messages: group messages tend to remain unattended in contrast to individual messages although they come from the same application. The statistical analysis confirms that group messages are seen, checked, and consumed significantly later than individual messages.

Messages Have Highest Consumption Rate

The flow diagrams confirm that messages, regardless of individual or group, are more frequently attended by opening the application (they are 'consumed') than the other three categories of notifications. The thin flows of the other three categories leading to the state 'consumed' in the diagrams reveal that the notifications rarely attract the users to open the app shortly after it was posted.

Few Differences in Time Until Checked or Consumed

Surprisingly, we did not find significant differences between how fast notifications are checked or consumed (if they are consumed). The only exceptions are that (1) emails were checked less quickly than notifications from all other categories, and (2) group messages were checked slower than message- and non-social notifications.

NonSocial Notifications are Quickly Dismissed

While Email, Social, and NonSocial notifications are similar in terms of consumption rates, NonSocial notifications stand out from the other two categories in terms of attendance times. Interestingly, a greater portion of NonSocial notifications are

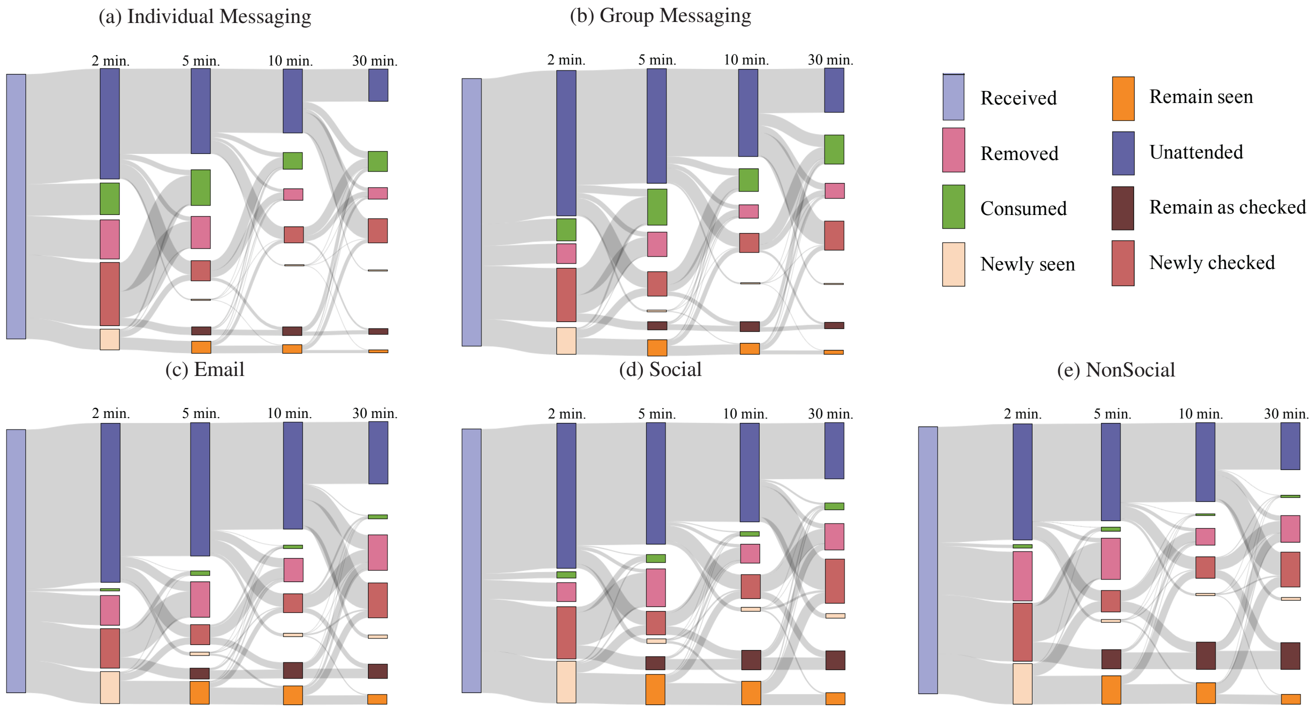

Figure 3: Sankey diagram-based visualization of notification-attendance flow 2, 5, 10, and 30 minutes since reception.

attended quickly. As shown in Figure 3 (e) the ratio of notifications attended within 2 minutes are in fact comparable to the amount of individual messages. However, the majority of them is attended by 'removing' them. Thus, NonSocial notifications tend to be dismissed quickly.

Email and Social Notifications Take Longest to be Attended
In contrast to NonSocial notifications, notifications from email- and social apps take comparably longer until they are seen, checked, or removed. In particular, email notifications are attended significantly slower apart from consumption. This means that users tend to leave notifications from these types of apps unattended for longer period of times.

DISCUSSION

Notification Volume

Our participants received an average of 56 notifications per day. We confirm previous findings [12, 19, 29, 25] that the majority of mobile phone notifications originate from messaging applications. Advancing previous work, we explored individual- and group messages separately. We expected that group message notifications would be more frequent, since they involve a lot of people. However, we found that the majority of messaging notifications are from 1-to-1 chats. Nevertheless, this does not necessarily mean that there are fewer group message. Another explanation is that our participants muted group chats.

The average participant did not receive many notifications from email-, social-, and non-social apps. The low number of emails is an interesting contrast to previous works [25, 29] where email notifications were much more common. We hypothesize that the main difference lies in the sampling of the participants: they way that those previous works sampled their participants was more likely to attract power users and information workers, which may use email a lot more than the average mobile phone user. Further, the low number of non-messaging notifications does not mean that such apps are not trying to notify the user. An alternative explanation is that users have begun to disable notifications from these apps.

All in all, our findings show that apart from messaging, mobile phone users do not seem to be "flooded" by notifications from many different apps. Compared to prior work (Table 2), our findings do not confirm that "*Direct notifications are on the exponential rise*" [5]. Instead, our findings confirm that one of the primary functions of smart phones has stayed the same since the success of SMS: stay in touch with people through messaging.

Conversion Rates

Conversion rates (or click rates) indicate the probability that a notification will trigger the desired action: opening the corresponding app. Mehrotra *et al.* [22] report a *click rate* for notifications is 62.52%. This corresponds well to our conversion rates of 63.67% and 65.81% of individual- and group notification, respectively. However, our study also reveals that notifications from non-messenger apps are consumed at much lower rates (Email: 15.47%, Social, 26.55%, NonSocial: 16.19%). This indicates that notifications from non-messaging apps are comparably ineffective. One explanation might be that they are attended on different devices – which highlights the opportunity to suppress notifications on the mobile phone when such attendance can be predicted.

Further, it might be considered surprising that even the conversion rates of messaging notifications are clearly below 100%, as the majority of messages should receive a response. We hypothesize two explanations: some messages do not require a response (e.g. "ok!") and can therefore be dismissed without issues. Further, we might be seeing that responses are made not from within the app, but for example, interactive notifications or web interfaces.

Immediate Attendance

While notifications are designed to draw attention to content outside of the user's focus, it still happens that notifications are fired while the user is already using the phone or the corresponding app. We found that 21-23% of the message notifications are received while the phone is already in foreground. These numbers make sense as messaging sometimes happens as chat. That is, users do not turn off the phone between turns by wait with the chat window opened.

We further found that between 20 and 35% of the notifications arrive while the phone is unlocked. These numbers show the potential of mitigating disruptions by improving the timing when the phone is already in use. For example, the phone could wait with posting the notification until the user is detected to be at a *breakpoint* between two tasks (e.g. closing the current app) [14, 23].

Notification Attendance

Previous work that explored how notifications of different category were handled found that message notifications are triaged [25] and clicked [29] fastest. Our results confirm that notification from messaging apps are on average amongst those to be seen, checked, consumed, and removed fastest. However, thanks to the fact that we explored different types of attendance separately, we found message notifications do not stand out in all types of attendance.

Regarding the time until a notification is checked, we did not find significant difference between the categories. Only email notifications were checked significantly later than the other notifications.

Similarly, if our participants decided to consume the notification, there were no significant differences in the time until the consumption. Only two small effects in the pair-wise comparison hint that group messages might be consumed slower than the other types of notifications. This indicates that if the decision was made to consume a notification, the type of the notification is not the deciding factor to determine the time until it is consumed. We hypothesize that instead, external factors are what predict when a notification can finally be consumed.

The main differences between how notifications from different categories are handled are the time until they are seen and removed. Notifications from messengers and non-social apps were seen fastest, notifications from email apps slowest. The difference in the seen time shows that our participants either did not have audible email notifications, or the alert did not prompt them to check the phone.

Similarly, we find significant differences in the time until notifications were removed if our participants decided to remove them. Notifications from non-messaging apps were removed significantly later than those of messaging apps. This indicates that participants take more time to arrive to the decision to dismiss a non-messaging notification. We hypothesize that there are two factors: there is less social pressure to respond immediately, and dismissing a notification is the harder choice to make, because it ultimately means to accept the possibility of missing out on something.

CONCLUSIONS

We analyzed 794,525 notifications from 278 mobile phone users and how these notifications were handled. We did not find that the daily volume of notification has risen compared to past studies. Individual and group message notifications remain the most important notifications. Other types of notifications were largely ineffective, get removed quickly, or are left pending for a long time.

Research on mitigating the disruptiveness of notifications should therefore focus on notifications from messaging applications. Further, while notifications are designed to grab attention while the user is busy with other things, we found that a significant portion of notifications arrives while the phone is already unlocked, indicating that it makes sense to specifically target this scenario, *e.g.*, by delaying them to breakpoints in the interaction. Finally, as non-messaging notification were largely ineffective, we need to investigate how to make them more useful and relevant for users again.

REFERENCES

1. Piotr D. Adamczyk and Brian P. Bailey. 2004. If Not Now, when?: The Effects of Interruption at Different Moments Within Task Execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 271–278. DOI: http://dx.doi.org/10.1145/985692.985727

2. Agathe Battestini, Vidya Setlur, and Timothy Sohn. 2010. A Large Scale Study of Text-messaging Use. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. ACM, New York, NY, USA, 229–238. DOI: http://dx.doi.org/10.1145/1851600.1851638

3. Jeremy Birnholtz, Jeff Hancock, Madeline Smith, and Lindsay Reynolds. 2012. Understanding unavailability in a world of constant connection. *interactions* 19, 5 (2012), 32–35. DOI: http://dx.doi.org/10.1145/2334184.2334193

4. Jelmer P. Borst, Niels A. Taatgen, and Hedderik van Rijn. 2015. What Makes Interruptions Disruptive?: A Process-Model Account of the Effects of the Problem State Bottleneck on Task Interruption and Resumption. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2971–2980. DOI: http://dx.doi.org/10.1145/2702123.2702156

5. Lewis Chuang, Sven Gehring, Judy Kay, and Albrecht Schmidt (Eds.). 2017. *Ambient Notification Environments*.
6. Herbert H. Clark. 1999. Using Language. *Journal of Linguistics* 35, 01 (March 1999).
7. Mary Czerwinski, Eric Horvitz, and Susan Wilhite. 2004. A Diary Study of Task Switching and Interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 175–182. DOI: http://dx.doi.org/10.1145/985692.985715
8. Tilman Dingler and Martin Pielot. 2015. I'll Be There for You: Quantifying Attentiveness Towards Mobile Messaging. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 1–5. DOI: http://dx.doi.org/10.1145/2785830.2785840
9. Joel E. Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating Episodes of Mobile Phone Activity As Indicators of Opportune Moments to Deliver Notifications. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 181–190. DOI: http://dx.doi.org/10.1145/2037373.2037402
10. Joel E. Fischer, Nick Yee, Victoria Bellotti, Nathan Good, Steve Benford, and Chris Greenhalgh. 2010. Effects of Content and Time of Delivery on Receptivity to Mobile Interruptions. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. ACM, New York, NY, USA, 103–112. DOI: http://dx.doi.org/10.1145/1851600.1851620
11. James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang. 2005. Predicting human interruptibility with sensors. *ACM Trans. Comput.-Hum. Interact.* 12, 1 (Mar 2005), 119–146. DOI: http://dx.doi.org/10.1145/1057237.1057243
12. Jose A. Gallud and Ricardo Tesoriero. 2015. Smartphone Notifications: A Study on the Sound to Soundless Tendency. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15)*. ACM, New York, NY, USA, 819–824. DOI: http://dx.doi.org/10.1145/2786567.2793706
13. Eric Horvitz and Johnson Apacible. 2003. Learning and Reasoning about Interruption. In *Proc. ICMI '03*.
14. Eric Horvitz, Johnson Apacible, and Muru Subramani. 2005. Balancing awareness and interruption: investigation of notification deferral policies. In *Proc. UM '05*. Springer-Verlag, 5. DOI: http://dx.doi.org/10.1007/11527886_59
15. Shamsi T. Iqbal and Brian P. Bailey. 2008. Effects of Intelligent Notification Management on Users and Their Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 93–102. DOI: http://dx.doi.org/10.1145/1357054.1357070
16. Shamsi T. Iqbal and Brian P. Bailey. 2010. Oasis: A framework for linking notification delivery to the perceptual structure of goal-directed tasks. *ACM Trans. Comput.-Hum. Interact.* 17, 4, Article 15 (Dec 2010), 28 pages. DOI:http://dx.doi.org/10.1145/1879831.1879833
17. Shamsi T. Iqbal and Eric Horvitz. 2010. Notifications and Awareness: A Field Study of Alert Usage and Preferences. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW '10)*. ACM, New York, NY, USA, 27–30. DOI: http://dx.doi.org/10.1145/1718918.1718926
18. Kostadin Kushlev, Jason Proulx, and Elizabeth W. Dunn. 2016. "Silence Your Phones": Smartphone Notifications Increase Inattention and Hyperactivity Symptoms. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1011–1020. DOI: http://dx.doi.org/10.1145/2858036.2858359
19. Uichin Lee, Joonwon Lee, Minsam Ko, Changhun Lee, Yuhwan Kim, Subin Yang, Koji Yatani, Gahgene Gweon, Kyong-Mee Chung, and Junehwa Song. 2014. Hooked on Smartphones: An Exploratory Study on Smartphone Overuse Among College Students. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2327–2336. DOI: http://dx.doi.org/10.1145/2556288.2557366
20. Gloria Mark, Stephen Voida, and Armand Cardello. 2012. "A Pace Not Dictated by Electrons": An Empirical Study of Work Without Email. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 555–564. DOI: http://dx.doi.org/10.1145/2207676.2207754
21. Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 813–824. DOI: http://dx.doi.org/10.1145/2750858.2807544
22. Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1021–1032. DOI: http://dx.doi.org/10.1145/2858036.2858566

23. Tadashi Okoshi, Kota Tsubouchi, Masaya Taji, Takanori Ichikawa, and Hideyuki Tokuda. 2017. Attention and Engagement-Awareness in the Wild: A Large-Scale Study with Adaptive Notifications. In *IEEE International Conference on Pervasive Computing and Communications*.

24. Veljko Pejovic and Mirco Musolesi. 2014. InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*. ACM, New York, NY, USA, 897–908. DOI: http://dx.doi.org/10.1145/2632048.2632062

25. Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014a. An In-situ Study of Mobile Phone Notifications. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 233–242. DOI: http://dx.doi.org/10.1145/2628363.2628364

26. Martin Pielot, Rodrigo de Oliveira, Haewoon Kwak, and Nuria Oliver. 2014b. Didn't You See My Message?: Predicting Attentiveness to Mobile Instant Messages. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3319–3328. DOI: http://dx.doi.org/10.1145/2556288.2556973

27. Martin Pielot and Luz Rello. 2017. Productive, Anxious, Lonely: 24 Hours Without Push Notifications. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA, Article 11, 11 pages. DOI: http://dx.doi.org/10.1145/3098279.3098526

28. Julie Rennecker and Lindsey Godwin. 2005. Delays and Interruptions: A Self-perpetuating Paradox of Communication Technology Use. *Inf. Organ.* 15, 3 (July 2005), 247–266. DOI: http://dx.doi.org/10.1016/j.infoandorg.2005.02.004

29. Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. 2014. Large-scale Assessment of Mobile Notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3055–3064. DOI: http://dx.doi.org/10.1145/2556288.2557189

30. Cary Stothart, Ainsley Mitchum, and Courtney Yehnert. 2015. The attentional cost of receiving a cell phone notification. *Journal of experimental psychology: human perception and performance* 41, 4 (2015), 893. http://dx.doi.org/10.1037/xhp0000100

31. Liam D. Turner, Stuart M. Allen, and Roger M. Whitaker. 2017. Reachable but not Receptive: Enhancing Smartphone Interruptibility Prediction by Modelling the Extent of User Engagement with Notifications. *Pervasive and Mobile Computing* (2017). DOI: http://dx.doi.org/10.1016/j.pmcj.2017.01.011 accepted Jan 31, 2017.