# "Don't Bother Me. I'm Socializing!": A Breakpoint-Based Smartphone Notification System

**Chunjong Park Junsung Lim Juho Kim Sung-Ju Lee Dongman Lee**
School of Computing, KAIST
Daejeon, Republic of Korea
{cj.park, wns349, juhokim, profsj, dlee}@kaist.ac.kr

## ABSTRACT
Smartphone notifications provide application-specific information in real-time, but could distract users from in-person social interactions when delivered at inopportune moments. We explore *breakpoint-based notification management*, in which the smartphone defers notifications until an opportune moment. With a video survey where participants selected appropriate moments for notifications from a video-recorded social interaction, we identify four breakpoint types: long silence, a user leaving the table, others using smartphones, and a user left alone. We introduce a Social Context-Aware smartphone Notification system, SCAN, that uses built-in sensors to detect social context and identifies breakpoints to defer smartphone notifications until a breakpoint. We conducted a controlled study with ten friend groups who had SCAN installed on their smartphones while dining at a restaurant. Results show that SCAN accurately detects breakpoints (precision=92.0%, recall=82.5%), and reduces notification interruptions by 54.1%. Most participants reported that SCAN helped them to focus better on in-person social interaction and found selected breakpoints appropriate.

### Author Keywords
Smartphone notifications; social interactions; interruptions; breakpoints; mobile computing; context-aware computing.

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

## INTRODUCTION
In a world of always-on mobile connectivity and overload of information, a smartphone receives an average of tens to hundreds of push notifications a day [42]. Despite its usefulness in immediate delivery of information, an untimely smartphone notification is considered a source of distraction and annoyance during a social interaction [32, 36]. In fact, even a mere buzz or sound of a notification takes user's attention away [35]. Our preliminary survey (n=224) on smartphone use during a social interaction shows that people particularly feel distracted when notifications arrive in the middle of a conversation. In particular, about 70% of respondents reported that during a social interaction, they start using their smartphones due to external cues such as notifications. Moreover, we found that people use their smartphones more frequently when they are in a casual, informal social setting than in a professional setting. Regardless of the social groups, however, smartphone use during a social interaction is considered inappropriate and many smartphone users think the use hurts, rather than helps, the quality of a conversation [43].

To mitigate undesirable interruptions caused by smartphone notifications, people resort to self-regulation, such as activating silent mode, turning off notifications, and even not bringing smartphones with them [24]. However, such explicit efforts by people are shown to be ineffective due to lack of self-control [40, 45]. Various approaches (e.g., locking the phone [23], warning excessive use [33] and competing against friends [24]) have been proposed to limit smartphone usage. However, they not only require users' explicit effort, but also restrict or restrain users from using the smartphone when needed. Moreover, simply restricting smartphone use could cause users to feel stressed and anxious to check for new notifications [40]. Another thread of research attempts to detect opportune moments in which notifications cause minimal interruptions on the user's engaged task [15, 21, 39]. These approaches reschedule notifications solely based on an individual's activity to minimize interruptions. During a social interaction, however, activities of the group should be considered to detect opportune moments for notifications.

We propose that deferring notifications until an opportune moment, namely a *breakpoint*, would improve the quality of a social interaction. A breakpoint is a term originated from psychology that describes a unit of time in between two adjacent actions [37]. Our intuition is that there exist breakpoints in which notifications do not, if so minimally, interrupt a social interaction. For instance, a pause during a conversation or standing up to go to the restroom might be breakpoint candidates for notifications. By having notifications rescheduled until a breakpoint, users would get minimally interrupted during a social interaction, without having to explicitly regulate their smartphone use. The deferring technique has been


*CSCW '17,* February 25-March 01, 2017, Portland, OR, USA

DOI: http://dx.doi.org/10.1145/2998181.2998189

shown to be effective, as there is a transient reduction on mental effort to receive notifications between two actions [21, 37].

But a challenge remains as to what kind of breakpoints are appropriate and how to detect them. In order to identify an ideal set of breakpoints, we conducted a video survey (n=73), in which respondents were asked to select appropriate moments for notifications as they watched a video clip of a social interaction. Based on the responses, we identified four types of breakpoints that are appropriate for notifications: a long silence (hereinafter *Silence*), a user leaving the table (*Movement*), others using smartphones (*Use*), and a user left alone (*Alone*). *Movement* differs from *Alone* in that a user leaving the table is *Movement*, but not necessarily *Alone* as more than two users might remain at the table. Also, a user may be *Alone* at the table awaiting for others.

We introduce a Social Context-Aware smartphone Notification system, SCAN, that defers smartphone notifications until a breakpoint during a social interaction. SCAN is a mobile application that detects social context using built-in sensors and identifies the four breakpoint types in the background. In order to detect social context of a group, the system works collaboratively with the rest of the group members' smartphones to sense collocated members, conversation, and others' smartphone use. The system then classifies a breakpoint based on the social context and decides whether to deliver or defer notifications.

Our controlled experiment conducted on 10 groups of friends (n=30) shows that SCAN helps users to focus better on the conversation by reducing the number of interruptions by 54.1%. Results show that SCAN accurately classifies breakpoints (precision=92.0%, recall=82.5%) in a controlled social interaction setting, which was measured against ground truth obtained from manually labeling captured videos. From the interviews, we confirm that SCAN defers notifications by 51 seconds on average (min=1s, max=180s), which most participants did not recognize. Most participants appreciated the value of deferred notifications and found the selected breakpoints appropriate. Overall, we demonstrate that breakpoint-based smartphone notification management is a promising approach to reducing interruptions during social interactions.

Our main contributions are as follows:

- We report results from a survey on smartphone use and attitudes toward others' smartphone use during a social interaction for different social groups; people tend to use smartphones more freely in an informal social setting than in a professional setting.
- We identify breakpoints in which smartphone notifications minimally interrupt a conversation during social interactions.
- We build a prototype system, SCAN, that seamlessly manages smartphone notifications by detecting social context and opportune breakpoints.
- We evaluate SCAN during a social interaction in a controlled setting to show its effectiveness in reducing the total number of notification interruptions.

In the remainder of the paper, we first discuss related work. We then report results from a survey and a video study that identify design considerations for detecting opportune moments for notifications. We introduce the SCAN system and report results from a controlled study. Finally, we discuss major design factors for notification management in a social setting, and conclude with future work.

## RELATED WORK

We discuss related work on (i) smartphone use in a social setting, (ii) notification management, and (iii) interruptions.

### Smartphone Use in Social Interactions

As smartphones have become an integral part of our lives, heavy smartphone use is now considered a source of distraction in carrying out daily life activities [3, 30, 32]. Recent studies have shown that smartphone use during a social interaction degrades the quality of the interaction [23, 24, 36]. In fact, a survey conducted by the Pew Research Center reports that people consider smartphone use at public places (e.g., a restaurant, a movie theatre) to be inappropriate [43]. Smartphone users put explicit efforts in limiting the use [24], but these efforts often fail due to lack of self-control [40, 45]. Ko et al. [23] propose a mobile application to help groups focus on a social interaction by locking each others' smartphones. However, restraining users from using their smartphones might cause users to feel anxious about missing important notifications [40]. Moser et al. [36] discuss attitudes toward smartphone use at a dining table and present potential for systems to control smartphone use during a social interaction. We take a similar approach and develop a system to mitigate smartphone use in social interactions by reducing interruptions caused by notifications.

### Smartphone Notifications and Interruptions

Pielot et al. [42] conducted an in-situ study on 15 smartphone users and observed that people receive 63.4 notifications per day, mostly from messaging and email applications. Another study reported that college students in Korea receive 400 notifications per day, including from messaging apps [30]. Other research reports that people check notifications every few minutes and feel the social pressure to respond quickly to notifications from messaging apps [7, 42, 44].

As notifications take away people's attention from their current task, smartphone notifications are often considered as interruptions. Just a vibration or a short ringing sound is enough to disrupt an ongoing task [47]. Excessive notifications from an application could annoy people and even lead them to delete the application [9]. Other research [34, 41] suggests that people's perception on interruptions caused by notifications differs depending on the type of ongoing tasks, personality, age, application, ringer mode, etc.

There is also evidence that notifications, in form of interruptions, could cause negative impact on people's lives. Interruptive notifications increase the cognitive load [28], making people more prone to error and distractions. They are known to reduce work productivity as it requires additional time and effort to resume a task once they draw attention away [1, 6, 20, 4]. In addition, notifications can cause people to suffer from hyperactivity, inattention, and even ADHD (Attention Deficit Hyperactivity Disorder) [25, 47].

### Mitigating Interruption

Mitigating interruptions has been an active area of research in both desktop and mobile environments. Horvitz et al. [16] show the effectiveness of reducing interruptions in a desktop environment by delaying notifications for one to five minutes. Oasis [21] introduced the idea of deferring notifications until a breakpoint, which is defined as time between when a person finishes an ongoing task and when the person starts a new task [37]. Evaluation showed that deferring email notifications to a breakpoint distracted people less [21].

Breakpoints in a mobile environment have been studied and explored in a few different directions. Ho and Intille explore breakpoints involving various physical activities [15]. The results show that delivering notifications at a transition between two physical activities reduces cognitive load in receiving notifications. Attelia [38] investigates breakpoints in a user's smartphone activities to reduce interruptions while using the smartphone. A follow-up system, Attelia II [39], combines smartphone and physical activity to detect breakpoints in daily life.

### Summary and Our Focus

Previous work has mostly focused on reducing interruptions for individual users. We instead concentrate on social interactions, exploring ways to reduce interruptions caused by smartphone notifications in a small group setting. Breakpoints defined in previous work might not translate directly to a group setting. For example, previous work [15, 39] claims that a transition from walking to sitting is a good breakpoint. However, this might not be true in a social interaction setting. For instance, finding a table and sitting down with friends for a conversation is a walking-to-sitting transition but might not be appropriate to receive notifications. This research attempts to identify a set of breakpoint types acceptable in a social interaction.

## SURVEY: SMARTPHONE USE IN SOCIAL INTERACTIONS

As a preliminary study, we conduct an online survey to understand how smartphone use, as a form of interruption, affects social interactions. Among many forms of social interactions, the survey focuses on mealtime interactions, as the smartphone use while dining is generally considered inappropriate and a violation of etiquette [10, 43]. The goal of this survey is to understand how people use smartphones during a mealtime social interaction, and how their use varies depending on who they are with. We aim to answer the following questions with the survey:

- *Q1. Do people think smartphone use distracts the conversation?*
- *Q2. Do people control or change their smartphone use depending on who they are with?*
- *Q3. What causes people to use smartphones during a social meal?*

To answer the question on different social groups (Q2), we present four groups: (i) a significant other, (ii) family members, (iii) friends and (iv) professional relationships. The survey also asks if a respondent controls smartphone use in front of others and how others' smartphone use makes them feel. Open-ended questions at the end of the survey ask respondents to describe their thoughts and any unpleasant experiences caused by smartphone use during social interactions. We recruited participants through online forums, online social networks, and emails. Participation was voluntary, and there was no financial reward for completing the survey.

A total of 224 (103 male and 121 female) people completed the survey from 14 different countries (Asia: 78.1%, North America: 14.7% and Europe: 7.2%). The age of respondents ranged from 17 to 65 (17-24: 23.2%, 25-34: 53.6%, 35-44: 20.5%, 45-65: 2.7%) and their occupations varied from students to researchers, engineers, artists, designers, teachers, housewives, etc.

### Results

*Q1. Do people think smartphone use distracts the conversation?*

The survey asks to consider smartphone use as a form of distraction in two ways: a distraction caused by the respondent's own smartphone use and a distraction caused by others' smartphone use. Regardless of the group and who the smartphone user is, many respondents (over 73%) feel that smartphone use is a source of distraction during social interactions. We found that gender does not affect the perception of smartphone use during a conversation ($p > 0.05$, Cohen's $d = 0.08$) while age does; respondents in 20s feel less distracted by others' smartphone use than those in 30s and 40s ($p < 0.05$, Cohen's $d = 0.21$). In open-ended questions regarding experiences and thoughts on smartphone use during social interactions, many respondents considered others' smartphone use to be inappropriate, rude, offensive, and making them feel less valued. A respondent wrote *"just want to leave sometimes"* (#8), while another respondent wrote, *"playing a mobile game due to game push notifications makes me feel less important,"* (#182). Another respondent (#19) wrote he felt *ignored* when his significant other used the smartphone.

*Q2. Do people use smartphone differently depending on who they are with?*

Most respondents reported that they control smartphone use in front of others. They control their smartphone use the most when with professional relations (81.2%) and the least when with friends (67.4%). In addition, 56.7% ask for others' permission to use a smartphone when with professional relations, whereas only 28.5% do so when with friends. Respondents reported they use smartphones significantly more frequently when with friends than with professional relations ($p < 0.05$, Cohen's $d = 3.26$).

| Reasons using the smartphone | % of respondents |
|---|---|
| Receiving a call | 73.7 |
| Replying to an incoming message | 73.2 |
| Looking at the notification | 69.2 |
| Using apps to help the conversation | 65.2 |
| Checking the time | 58.0 |
| Sending a message | 43.8 |
| Using apps due to notifications | 35.7 |
| Using apps for other reasons | 37.5 |
| Placing a call | 29.9 |

**Table 1. Activities causing the smartphone use during a social meal.**

*Q3. What causes people to use smartphones during a social meal?*

While many try to control their smartphone use as reported above, respondents still use smartphones during social interactions. Table 1 lists common smartphone-related activities people engage in during a social interaction. It is interesting to note that smartphone use is mainly triggered due to external cues such as incoming calls (73%), messages (73%) and push notifications (69%). About 60% of the respondents use smartphones in response to external cues when they are with friends and the number drops to about 40% when they are with professional relations.

Many respondents were approving of others' smartphone use in emergency situations, such as receiving an urgent phone call, but felt that the use should be avoided otherwise, *"at least during an active conversation."* (#87).

The survey results indicate that people find smartphone use during a social interaction to be distracting. However, many still use their smartphones due to external cues, such as receiving calls, replying to messages, and checking notifications, especially when they are with friends. In the next section, we explore different breakpoints in which the delivery of notifications minimally interrupts the conversation.

## VIDEO EXPERIMENT

We believe that presenting smartphone notifications at the right breakpoint is beneficial for both the smartphone user and the group; the user would get notifications without interrupting the group's ongoing conversation. Completely blocking notifications or smartphone use for an entire social interaction might cause inconveniences such as missing important messages or information, feeling anxious, and increasing habitual checking [40]. Instead, deferring notifications to breakpoints is proven to be effective in reducing interruptions in a work environment or daily activities [21, 39]. We hypothesize that deferring notifications to breakpoints would also be effective during social interactions. Breakpoint identification in prior work has mostly focused on reducing interruptions for an individual, which leaves breakpoints during social interactions largely unexplored. To identify appropriate breakpoints during social interactions, we conducted an online video experiment.

We designed an online video experiment in which participants are asked to decide whether a specific point during a social interaction shown in a video clip is appropriate for receiving notifications. A 3-minute long video clip plays a situation where a group of three friends are having a casual dinner at a pizzeria. We recorded and added various breakpoint candidates to the clip, such as waiting for a friend, talking with each other, one person going to a restroom, eating, a person using a smartphone, placing an order, and silence. These candidates are selected from activities that frequently occur at a typical restaurant.

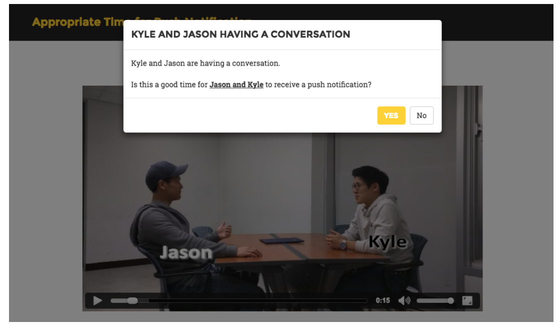


**Figure 1. An online video experiment to identify breakpoints during a social interaction.**

| Situation | % of participants |
|---|---|
| Alone | 98.6 |
| Someone leaving the table | 90.4 |
| Friends using smartphones | 84.4 |
| A long silence | 70.8 |
| A short silence | 63.1 |
| Eating | 47.2 |
| Talking to the waiter | 46.6 |
| Conversation | 24.2 |

**Table 2. Situations selected by video experiment participants as viable breakpoints during a casual social interaction.**

Each breakpoint candidate appears in the video at least once. Designed as a webpage, the video player pauses upon reaching a breakpoint candidate, and asks the participant to answer whether the presented breakpoint candidate is appropriate for receiving notifications (Figure 1). To prevent participants from skipping or providing repetitive responses, the video player control is blocked and hidden when questions are displayed.

### Results

We advertised the video experiment link via email, online social networks, and school communities. A total of 73 participants completed the experiment. As shown in Table 2, only 24% of the participants responded that receiving notifications in the middle of a conversation is appropriate. Breakpoints that a majority of participants found appropriate (over 70%) are: (i) when there is no conversation for five or more seconds (*Silence*), (ii) when a person in the group leaves the table (*Moving*), (iii) when a person is left alone and waiting for friends (*Alone*), and (iv) when others at the table use smartphones (*Use*). We consider these breakpoints as our target breakpoints when designing a system that effectively defers notifications to breakpoints during social interactions.

## DESIGN CONSIDERATIONS

After analyzing the results from the survey and the video experiment, we identified several design considerations for a breakpoint-based smartphone notification management system.

- *Notifications at social interactions:* In the preliminary survey, over 73% of respondents reported that the use of smartphone distracts the social interaction. The survey result tells us that casual interactions with friends are more likely to get distracted by a relatively heavier smartphone use. We also found that smartphone notifications interrupt the ongoing interaction and drive people to use their smartphones. Therefore, managing smartphone notifications seems necessary to reduce interruptions.

- *Breakpoints at social interactions:* Based on our video experiment, we identified a set of breakpoints that people consider appropriate to receive smartphone notifications. Respondents found it appropriate to receive notifications when there is little interaction with the others (e.g., silence, others using smartphones). In order to deliver notifications at appropriate breakpoints, the system must distinguish and identify such contrasting situations.

- *Social context detection:* Social context during an interaction must be detected to accurately identify breakpoints during social interactions. Social context includes not only user's own voice, physical activity and smartphone use status, but also the presence of others, their voice, and smartphone use status. As the social context relies on information from multiple smartphones in the group, inter-device communication is necessary. There are various ways smartphones can communicate with each other, such as cellular networks, Wi-Fi, and Bluetooth. An appropriate networking protocol should be chosen for exchanging context information.

- *Breakpoint classification:* A breakpoint must be correctly identified from the social context. The social context comprises various features describing the current situation and the features might be inter-dependent on one another. Moreover, a breakpoint is represented as a set of feature values, which might not be easily identified using a rule-based approach. As a result, the system could benefit from machine learning techniques to accurately classify breakpoints.

## SCAN: SOCIAL CONTEXT-AWARE SMARTPHONE NOTIFICATION MANAGEMENT SYSTEM

We propose SCAN, a social context-aware smartphone notification management system. SCAN improves the quality of social interactions by reducing interruptions resulting from smartphone notifications and delivering notifications at appropriate breakpoints. Developed as an Android application, our SCAN prototype detects social context and identifies four types of breakpoints as identified from the video experiment: (1) *Silence*: when there is no conversation for five or more seconds, (2) *Moving*: when a person in the group leaves the table, (3) *Alone*: when a person is left alone and waiting for friends, and (4) *Use*: when another person at the table uses the smartphone.

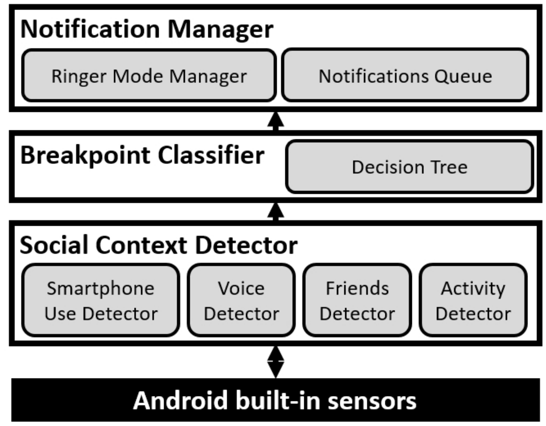


**Figure 2. SCAN system architecture. It consists of three main components and leverages built-in smartphone sensors to detect social context.**

### Prototype Design

SCAN seamlessly identifies breakpoints using built-in smartphone sensors such as microphone, step detector, inertia measurement unit (IMU) and Bluetooth Low Energy (BLE). As shown in Figure 2, SCAN has three key components: a social context detector, a breakpoint classifier, and a notification manager. We now describe each component in detail.

#### *Social Context Detector*

We define social context as a group of features that describe the current situation of a social interaction. Detecting social context is important, as incorrect context detection could result in a misclassification of a breakpoint. For instance, misclassifying a group actively engaged in a conversation as a breakpoint would cause notifications to interrupt users. The current SCAN prototype uses social context information to infer the following: (1) Is the user with a group? (2) Is there an ongoing conversation? (3) Is the user physically moving? and (4) Is the user using the smartphone?

*Presence of others*: SCAN periodically scans and advertises BLE beacons to detect others' presence and advertise its own presence. A beacon is chosen as a communication medium to exchange context data as it does not require explicit actions (e.g., pairing or connecting to an access point) from the user and is energy-friendly [12]. In addition, using a beacon allows us to form an intra-network for a social group and thus eliminates the need for an external server.

A beacon contains a unique identifier to represent the user so that other devices become aware of the user's presence. Assuming that every group member has each other's phone number in their contact list, SCAN hashes and includes its own phone number in the beacon's payload. Alternatively, a contact list from a social network can also be used. When a beacon is scanned, the system identifies the beacon originator (i.e., a group member) by comparing the payload to the hashed phone numbers in its contact list. This approach however, might perform poorly in the wild, where friends'

| Data Type | Description |
| --- | --- |
| *activity* | Describes the activity state of this device (e.g., walking, still). |
| *me_using* | Describes whether this device is in use. |
| *other_using* | Describes whether other devices are in use. |
| *me_talking* | Describes whether the user of this device is talking. |
| *other_talking* | Describes whether others are talking. |
| *num_people* | Specifies a total number of people nearby. |

**Table 3. Six types of data captured by SCAN's social context detector. The detected social context is in turn used to classify breakpoints.**

beacons from other tables could be scanned (e.g., in a school cafeteria). While we leave group detection in the wild as future work, we note that existing approaches (e.g., computer vision [2], sound [11] and bluetooth signal [22, 48]) can be leveraged to detect group's presence.

*Conversation*: In order to detect if the user is conversing, we use a built-in microphone to sense human voice by filtering ambient noise and non-voice sound. SCAN extracts the human voice frequency range (60 Hz to 270 Hz) from the raw audio input by applying a band-pass filter. The YIN pitch detection algorithm [8] is then used to detect human voice. The detected voice should be above a certain sound pressure level to ensure that the voice is originated by the smartphone user. The threshold for the sound pressure level was selected empirically as 55 dB. Detecting a conversation in this manner works well in a quiet environment, but more sophisticated methods such as speaker recognition [27] can be used to improve the accuracy in the wild.

A conversation usually takes turns; when one person speaks, others listen [31]. To accurately detect a conversation, the system should know if anyone is talking in the group. SCAN detects a conversation by exchanging the talking status of each user using the beacon.

*Activity*: Among many activities involved in a social interaction, such as dancing, eating, standing up, etc., walking and staying still are the activities of interest for breakpoint detection in SCAN. This is because a *Moving* breakpoint is inferred when a user either walks towards or away from the table. Otherwise, we assume that the user is at the table having a social interaction. On Android, Google Activity Recognition API [19] can be used to identify such an activity change. Alternatively, smartphone's built-in IMU and low-power step sensors can be used, where a heavy fluctuation in sensor values indicate a walking activity [5].

*Smartphone use*: Knowing whether a user is using her smartphone is important in determining a *Use* breakpoint. Smartphone use can be inferred through an interaction between the smartphone and the user. By checking if the screen or touch sensors are on, it is possible to infer whether the user is using the smartphone. As a *Use* breakpoint is represented by other's smartphone use, the detected smartphone use state is shared with the rest of the members at the table.

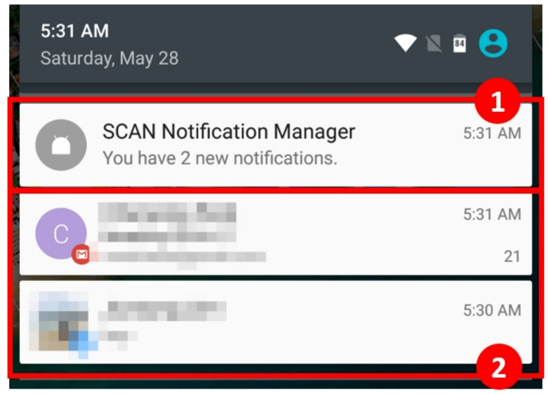


**Figure 3. A screenshot of SCAN. Box #1 shows that SCAN informs a user at a breakpoint that the user has 2 new notifications. Box #2 shows the new notifications that were deferred until the breakpoint.**

In summary, the social context detector generates six types of data, as shown in Table 3, which in turn are used to determine breakpoints. It is important to note that in order to protect user's privacy, SCAN does not store or send collected sensor data to external servers. Moreover, a microphone is used only to detect human voice and process the sound pressure level; the content of the conversation is not analyzed or converted to text.

### *Breakpoint Classifier*

In order to classify whether a moment is a breakpoint, we extract a set of features from the social context, namely *activity*, *me_using*, *other_using*, *me_talking*, *other_talking* and *num_people* (see Table 3). SCAN uses a decision tree for classification as breakpoint detection is a two-class classification since features are non-linearly related and its size is relatively small.

### *Notification Manager*

The notification manager is responsible for deferring and delivering notifications at the right breakpoints. Notifications received at a non-breakpoint are intercepted and queued without alerting the user. Once a breakpoint is detected, the manager batch-delivers all notifications in the queue, using the user's original ringer mode (i.e., alarm, vibrate or silent), as shown in Figure 3. The current SCAN prototype presents the notifications in the order of its original received time, but other sorting techniques could be used.

## Prototype Implementation

The SCAN prototype is an Android application that runs on Android 5.0 or above with a support for BLE. SCAN uses various built-in sensors to detect social context. A microphone records audio samples at a 50% duty cycle per second. Audio samples are processed with the Tarsos Audio Processing Library [46] to determine whether the user is talking. The Google Activity Recognition API [19] is used to determine the physical status of the user. The step counter in Android is used to detect walking, as the Google Activity Recognition API has an indefinite starting time and yields results slowly in the beginning. We use both APIs for consistent and best performance. The Accessibility Service on Android [17] allows SCAN to listen to all UI events and determine whether the user is currently using the smartphone. Exchanging social context information between the smartphones is done via BLE beacons, using the AltBeacon library for Android [49].

For the breakpoint classifier, we first collected 282 social context data from a simulated social interaction and manually labeled data. These data were then used to train a decision tree offline. The classifier is implemented using a J48 decision tree in Weka [13], a machine learning library.

We simulate notification interception by activating the silent mode, so that it neither rings or vibrates when a new notification is received. This is because Android does not permit SCAN to control other applications' notifications due to potential privacy and security concerns [18].

## EVALUATION: CONTROLLED EXPERIMENT

Our evaluation aims to verify whether breakpoint-based notification management is feasible and successfully reduces interruptions caused by smartphone notifications during social interactions. More specifically, objectives of our evaluation are as follows:

- Investigate the existence and distribution of breakpoints.
- Measure the breakpoint detection accuracy of SCAN.
- Evaluate whether SCAN reduces interruptions caused by smartphone notifications, and
- Understand participants' perception of notification management.

To achieve these objectives, we conducted a controlled experiment with groups of friends in a casual dining setting.

### Participants

We recruited ten groups of three friends (female-only group=5, male-only group=3, mixed-gender group=2, n=30, female=17, male=13) from KAIST. Participants are undergraduate and graduate students (mean age=21.8, min age=19, max age=27). We targeted a group of friends in their 20s as this age group is known to use smartphones the most [30], while controlling the least. There was no difference in smartphone use between genders ($p > 0.05$, Cohen's $d = 0.08$). SCAN was installed on participants' smartphones and ran as a background service to manage real-life notifications in a non-invasive manner. As a reward for participation, we provided a free meal of worth around \$15, which was served at a restaurant during the experiment.

Before the experiment, participants were asked to take a pre-questionnaire regarding smartphone usage and push notifications. Over 90% of participants responded that they use their smartphone several times an hour, and place it on a table in a restaurant when dining. Over 90% of participants responded they receive push notifications, including from messaging apps, several times an hour, and over 65% check new notifications within 10 minutes since their arrival.

Figure 4. A Japanese restaurant in which the controlled experiment took place. The experiment was conducted in a confined environment to control the amount and type of notifications participants received during a session.

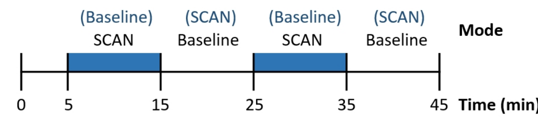


Figure 5. A timeline structure of the experiment. The first five minute is used to allow the participants to get used to the experiment environment. A sequence of mode is alternated for each experiment.

### Experimental Setting

In order to conduct the experiment in a realistic dining setting, we selected a Japanese restaurant near campus that many participants are familiar with. Figure 4 shows the interior of the restaurant where the experiment was conducted.

We collected video recordings of each group with the consent from every participant. To minimize the effects of a video camera, the camera was installed close to a ceiling so that it is out of the participants' sight. For data analysis purpose, SCAN included a data logging feature to log activities such as smartphone use, received notifications, processed audio data, the number of people detected, and physical activities.

### Design and Procedure

Our controlled experiment used a within-subject design, in which participants used both the baseline (i.e., with no notification management) and SCAN. The baseline used the default Android notification alert setting that generates alerts with sound, vibration, or visual cue as soon as a notification arrives.

Because the experiment involved dining at a restaurant, we ran all sessions during lunchtime or dinnertime. We scheduled the experiment based on each participant group's availability. Participants might behave differently in lunch and dinner settings, given that people's perception toward the mealtimes might differ depending on their cultural background and lifestyle. In order to minimize such differences, we selected the same menu, the same room at the same restaurant, and the same experiment session duration. We controlled when each dish is served, and instructed the participants to dine until we inform them that the session is over.

After signing a consent form, participants were asked to install the SCAN application with our instruction, and have a meal with their friends. They then answered a pre-questionnaire on smartphone usage, notification management, and placement of the smartphone. Each session lasted 45 minutes, which is considered a long enough time to have a social meal in the Korean culture [29]. We discarded data from the first five minutes of each session to allow the participants to get used to the experiment environment. The remaining 40 minutes was split into four 10-minute mini-sessions, where the baseline and SCAN conditions alternated, as shown in Figure 5. The interface order was counter-balanced across groups. This design let participants experience 20 minutes of each interface. Finally, after the meal session, we briefly explained how SCAN works to the participants and conducted a semi-structured interview, followed by a post-questionnaire. Note that we did not explain the exact functionality of SCAN before the experiment to minimize changes in participants' behavior.

In order to control the amount of notifications received across participants in each interface condition, SCAN generates notifications of its own when there is imbalance in total notifications. In determining the amount, we refer to previous studies and surveys: (i) people on average receive around 65 notifications per day [42], (ii) heavy smartphone users receive more than 400 notifications per day [30], (iii) our participants' age group tends to receive more notifications than other age groups [30], and (iv) people receive more notifications during lunchtime and dinnertime [42]. Based on these trends and numbers, we implemented an additional module in SCAN *solely for the experiment*, which generates an artificial push notification when no new notification is received for three minutes. Additionally, previous research shows that notifications from other than messaging, email, and social network apps do not draw immediate attention and have a longer response time [42]. Thus, we designed artificial notifications that send trivial piece of information such as random shopping items and entertainment news headlines, so that the interruption level does not significantly increase due to these notifications.

As for controlling the type of breakpoints participants experience, we ensured that each four breakpoint types happen at least once in each interface condition. As described earlier, the four types are: when a user is alone (*Alone*), when a user leaves the table (*Movement*), when a silence persists for longer than five seconds (*Silence*), and when other members at the table are using their smartphones (*Use*). We assume that *Silence* and *Use* would naturally occur during a session. To trigger *Alone* and *Movement*, we asked one or two participants to leave the room right after placing an order and after they finish eating.

## Results

### Breakpoint Distribution

In order to evaluate the accuracy of SCAN's breakpoint detection, we compared its results against the ground truth. To construct the ground truth on breakpoint appearance and its type, two researchers manually and independently annotated all video recordings. There was a high level of agreement between the two annotators (Cohen's $\kappa$ = 0.85), and conflicts were resolved by discussion.

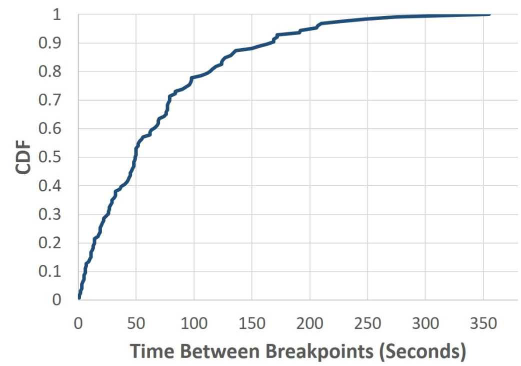


**Figure 6. A cumulative distribution of breakpoint intervals.**

Table 4 shows the distribution of the four breakpoint types, in terms of frequency and duration. *Use* appeared the most (48.3%), followed by *Silence* (46.6%), *Moving* (2.6%) and *Alone* (2.6%).

All participants experienced *Use* during their session. We observe that most participants (28 of 30) used smartphones more than once during the experiment, and a single use lasted about 53 seconds on average. It is interesting to note that the duration of smartphone use varies greatly from as short as a second (e.g., a glance at the phone to check the time) to as long as 962 seconds (e.g., playing a mobile game).

*Silence* was the second most common breakpoint type, which lasted about 8 seconds on average (min=6, max=35). Since the experiment was conducted at a restaurant for 45 minutes per group, *Moving* and *Alone* breakpoints appeared only when a member of the group was instructed to leave the table as part of the experiment design. A total of nine breakpoints each for *Moving* and *Alone* appeared as one group failed to follow the instruction and only a single participant from each group left the table.

We also examine breakpoint intervals, which are defined as the time between a breakpoint and the following breakpoint. Our results in Figure 6 show that most breakpoint intervals (over 80%) were under two minutes. The longest breakpoint interval was 355 seconds. This result indicates that breakpoints of our interest occur every few minutes, which presents opportunities for sending notifications in a fine-grained manner.

### Breakpoint Detection Accuracy

SCAN detects breakpoints in a seamless manner from the detected social context. We evaluate whether SCAN correctly classifies breakpoints from the social context. Comparing against the ground truth acquired from manual labeling, SCAN shows the precision, recall, and F1-score of 92.0%, 82.5%, and 87.0%, respectively (see Figure 7). We believe having a high precision is important to SCAN as an incorrect classification of a breakpoint could lead to undesirable notification delivery at undesirable moments. Recall is perhaps less critical as breakpoints appear quiet frequently as

| | Breakpoint Occurrence | | Breakpoint Duration (sec.) | | | |
|---|---|---|---|---|---|---|
| | Frequency | Ratio | Mean | SD | Min | Max |
| USE | 168 | 48.28 % | 53.43 | 129.47 | 1 | 962 |
| SILENCE | 162 | 46.55 % | 13.18 | 6.47 | 6 | 35 |
| MOVING | 9 | 2.59 % | 101.78 | 52.86 | 49 | 215 |
| ALONE | 9 | 2.59 % | 89.89 | 51.23 | 34 | 198 |

**Table 4. Breakpoint occurrence and duration by each type. The breakpoint occurrence frequency shows the sum of breakpoints appeared during the experiment and the breakpoint duration is shown in seconds.**

| Interface Condition | # Noti. Received | # Noti. Alarmed | Avg. Noti. Deferred | Avg. Time Deferred (seconds) |
|---|---|---|---|---|
| SCAN | 399 | 183 | 2.18 | 51 |
| Baseline | 451 | 451 | N/A | N/A |

**Table 5. Notifications received in each interface condition. On average, SCAN defers 2.18 notifications for 51 seconds. The baseline immediately alarms all received notifications.**

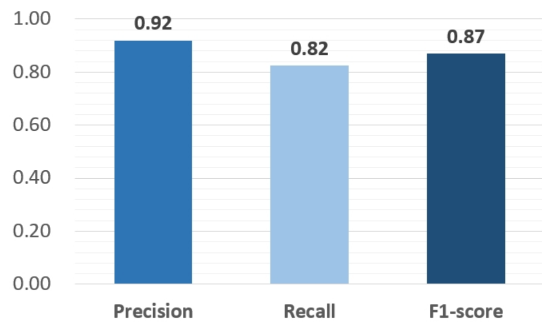


**Figure 7. Precision, recall and F1-score for breakpoint detection of SCAN.**

reported above, which means missed breakpoints only add a short amount of delay to queued notifications.

SCAN showed a high precision (>= 80%) on every social group but not as high recall. In particular, SCAN's recall score for one group was below 20%. This is because the group played a mobile game together throughout the social interaction and the Android Accessibility Service did not capture UI events created by the game. As a result, SCAN was unable to detect any of the user's gaming events in terms of social context, thus missing *Use* breakpoints.

### *Reduced Interruption*

SCAN delivers notifications only at breakpoints and queues the notifications until breakpoints. We evaluate SCAN by obtaining the number of reduced interruptions during a social interaction from the total number of interruptions of the baseline. We select the number of notification alarms as a metric for comparison, since prior work shows that a mere buzz notification causes an interruption [47]. We analyze the smartphone event log data to obtain the number of notifications received, alarmed, and queued, and deferred time of notification. Table 5 shows the summary of the results.

Throughout the experiments, there were 399 raw notifications in SCAN and 451 in the baseline interface ($p > 0.05$, Cohen's $d = 0.12$). These aggregate results represent notifications received by 30 participants in their 20-minute sessions using each interface. Notifications from messaging apps account for 57% of the total notifications received, followed by artificial notifications we created for the experiments (28%), SMS/MMS (5%), and others. Phone call is not considered as a notification in this work; a notification generates a brief alarm whereas a phone call continuously notifies people for a longer time than a notification does. Each participant received on average of about 13 and 15 notifications when SCAN and baseline are applied, respectively, and four of them were artificial notifications. On average, SCAN delivered a batch of 2.18 notifications ($sd = 2.22$) at a breakpoint, and deferred notifications for 51 seconds (min=1s, max=180s). Compared with the baseline, SCAN eliminates alarms for 216 (= 399 − 183) notifications, which accounts for 54.1% of the total notification alarms, thus reducing the interruptions caused by notifications.

### *Changes in Smartphone Use and Conversation Duration*

There was no significant change in smartphone use or conversation duration with SCAN. Initially, we assumed that participants would use smartphones less and engage in a longer conversation by having less notification interruptions. However, participants used smartphones similar number of times ($p > 0.05$, Cohen's $d = 0.34$) and the total conversation duration also remained similar ($p > 0.05$, Cohen's $d = 0.35$) between the two interfaces. This might be due to participants' proactive use of smartphones during the interaction. Also, we noticed that many had their eyes on the smartphones even when engaged in a conversation.

## Participants' Perception of Notification Management

Through the post-study interview, we inquired about participants' perception of interruptions and the four breakpoint types, as well as any potential inconvenience caused by deferring notifications.

*Q1. Did participants notice that notifications are deferred? Did they experience any inconveniences caused by deferred notifications?*

Most (25 of 30) participants responded that they did not notice that notifications were deferred. One participant noted, *"I* *used my smartphone as usual and didn't recognize that notifications were delayed at all."* Five participants who noticed deferred notifications mentioned that *"I noticed it because my smartphone vibrated only once but there were multiple new messages (#3)"*, and *"when I checked the notification status bar after the vibration, there was a perceivable time difference (#29)."*

Since SCAN's deferral is unnoticed by most participants, they did not report inconveniences caused by deferred notifications. A participant responded: *"I didn't feel any inconvenience because I didn't even realize this app is deferring my notifications."*(#18) When we asked whether SCAN would cause inconveniences by deferring notifications, most participants said that it is okay to defer notifications for a few minutes. Only one participant mentioned deferring would cause inconvenience, saying that *"I want to be informed with notifications at arrival, and I want to be in control of deciding whether to read or ignore."*(#20) We observed that deferring notifications for a few minutes is acceptable to most participants. The responses support that SCAN's notification deferring time, 51 seconds on average, would not cause inconveniences to most people.

*Q2. Do participants think deferring notifications to breakpoints reduces interruptions?*

Most participants agreed that deferring notifications to breakpoints could help them focus on a conversation. Many mentioned that notification alarms distract them from an ongoing conversation. One specifically said, *"vibrations or ringing sound during a conversation sometimes made me forget what I was about to say to my friends."* A majority of participants responded that many notifications do not require an immediate response, and avoiding alarms in the middle of a conversation would be helpful.

*Q3. What was the participants' perception of the four breakpoint types?*

We explained the four breakpoint types to participants before asking this question. Many participants responded that they could recall getting notification alerts when they went to a restroom, when there was a silence, and when their friends used their smartphone. Participants said it was not distracting to receive notifications when going to a restroom or being alone because they would use their smartphones anyway in these situations. They agreed that a breakpoint triggered when a friend uses the smartphone seems natural, because *"I also briefly check notifications when a person in front of me uses the smartphone."*(#13) But one participant expressed concern that *"it might lead to a prolonged smartphone use."*(#1)

They also agreed that a silence is a good breakpoint, mentioning that *"when I am with my friends, I also experienced a silent moment when all of us briefly check smartphones and come back to the conversation."*(#11) A few expressed different opinions on silence as breakpoints, mentioning that notification alarms during the silence might prevent them from resuming the conversation, and notifications might be even more distracting during silence. One participant also expressed concern on treating smartphone use as breakpoints, commenting that one person's smartphone use might initiate other people's smartphone use and result in less conversation at the table.

In order to confirm whether the four breakpoint types are sufficient, we asked the participants to suggest other possible breakpoint types. All participants responded that the four types covered most of the moments they felt appropriate. Overall, we confirmed that the four breakpoint types are generally acceptable to most participants.

*Q4. Did SCAN help improve the conversation quality?*

In the post interview, 13 participants reported they recall receiving notifications at breakpoints. When asked if deferring notifications to breakpoints helped them focus on the ongoing conversation, 10 of 13 responded positively. One participant commented, *"now that I come to think of it, my smartphone did not vibrate when we were talking, so my attention was at the conversation the whole time."*(#5) The other three participants felt deferring notifications was unnecessary. One commented, *"deferring notification seems to be a good idea, but I usually control myself in checking notifications."*(#7)

## DISCUSSION

### Smartphone Use During Social Interactions

Smartphone use during social interactions is considered inappropriate in general [23, 36]. However, from the controlled experiment, we observed many participants proactively using smartphones to assist interactions at the table. For instance, seven of ten groups used smartphones to take pictures and four groups used them to setup future schedules. This might be one of the reasons behind why our controlled experiment result reflects no significant changes in the amount of smartphone use and conversation duration with and without SCAN. Since SCAN does not explicitly limit the smartphone use, changes in smartphone use are not significant. Furthermore, this might be a unique characteristic of the target age group (20s). Nevertheless, comments from the post-interview tell us that most participants appreciated the value of deferred notifications and found the selected breakpoints appropriate. We believe a further study with more diverse demographic is needed to examine SCAN's effect in smartphone use and conversation length during social interactions.

### Breakpoint-based Management

During the experiment, we observed that people use smartphones for different reasons. Most used to respond to notifications, but there were cases when smartphones were used to help the social interaction (e.g., taking a picture, sharing information with others, and playing a mobile game together). Limiting smartphone use, including the use that helps a conversation, might not always improve the quality of a conversation. A recent work [23] limits smartphone use in a group setting and allows smartphone use for a short period of time, reflecting the users' need to use smartphones to help a conversation. We instead propose a breakpoint-based approach, which we believe is a more ambient and softer approach to reducing interruptions. Rather than explicitly limiting smartphone use, SCAN prevents smartphone use due to notifications while allowing users to use it when wanted. This ambient approach has an advantage of reducing interruptions such that the user does not even recognize that notifications are deferred. A potential risk is when context detection or breakpoint classification fails – the system might not explain its behavior clearly, as there is no separate visual UI. We address this issue by calibrating the system for high precision than for high recall, which means when notifications are presented, it is highly likely that it is an opportune moment, although the system might miss some candidate breakpoints.

### Breakpoints During Social Interactions

SCAN prototype is designed to identify four types of breakpoints during social interactions: *Alone*, *Moving*, *Silence*, and *Use*. In interviews after the experiment, most participants reported that they liked the idea of deferring notifications to a breakpoint. One participant commented: *"Giving me a notification alert when I am free seems to be a good idea."*(#29) We intentionally selected the four breakpoint types that are common, universal and context-independent, which span a wide variety of small-group social interactions. At the same time, we acknowledge these are only a small subset of possible breakpoints that can be found. With additional context information, such as hiking in a group, the action of taking a seat to have a break could be considered a good breakpoint. It would be interesting to explore and examine other forms of breakpoints at different social interactions.

### Different Views on Breakpoints

For the four breakpoint types we studied, it is interesting to see that participants have different views on the level of breakpoint appropriateness. During the interview after the experiment, every participant agreed that *Alone* is an appropriate time for notification as there is no one around. On the other hand, five participants were skeptical over receiving notifications at *Silence* or *Use* because the level of interruptions perceived might vary depending on the situation. This implication opens up a new opportunity to apply different deferring and presentation strategies for different breakpoint types, and customize or even personalize which breakpoints the system should register.

### Continuous Sensing Application and Privacy

SCAN uses various sensors in a smartphone to detect breakpoints. Specifically, it utilizes inertia measurement unit, Bluetooth, microphone, and touch screen to sense a user's context. Such continuous sensing is known to be a possible threat to a user's privacy [26, 14]. SCAN attempts to minimize the privacy leakage by instantly discarding the sensed data after classifying the breakpoints. Moreover, it does not store the processed data either in local storage or the cloud. For instance, a microphone is used solely to extract pitch and loudness from the captured audio signals to infer whether a human is speaking. Despite our efforts, privacy is not thoroughly studied in this paper, and future work is needed to address privacy-related issues, especially via a field study.

### Limitations

#### *Study Limitations*

Since SCAN is evaluated in a controlled environment on 30 college students from Korea, the study shows some limitations on demographic diversity and long-term effect. There might be cultural differences in smartphone use, dining manners, and group dynamics. The study also focused on a single age group (20s), and different age groups and social configurations might yield different results. SCAN is evaluated only for a short period of time, so a long-term effect of deferring notifications to breakpoints remains unexplored. An additional study with diverse age groups and cultures could provide complementary insights .

#### *Technical Limitations*

There are some technical limitations that must be overcome for SCAN to be used in the wild. First, a more profound mechanism for group detection is needed. The experiment was conducted in a quiet environment with a single group of participants per session. This setup made it possible to detect the social group using only BLE beacons as no other beacons from nearby tables were scanned. Second, SCAN can improve robustness in detecting social context. Most participants happened to place their smartphones on a table or a floor where microphone and activity detection sensors worked well. However, this might not be the case in the wild, and the system should be able to detect social context in various environments. Lastly, the system must be evaluated in terms of energy efficiency and computation overhead. Efforts have been made to make a lightweight and energy-friendly mobile system (e.g., use of BLE, audio duty cycle, and low-power step detector), but it has not yet been field-tested. The system burden could be further lowered by collecting contextual data from external sensors.

## CONCLUSION AND FUTURE WORK

Smartphone notifications during social interactions take people's attention away from the ongoing conversation. We explore opportune moments, i.e., breakpoints, in which notifications delivery minimally interrupts social interactions. We design and implement SCAN that detects social context using smartphone sensors and identifies breakpoints in the background. To evaluate SCAN, we select four target breakpoints as a long silence, a user leaving the table, others using smartphones, and a user left alone. Results from our controlled experiment and post-interview show that SCAN reduces interruptions caused by notifications in an unnoticeable and non-restrictive manner.

For future work, we plan to explore different breakpoints during different types of social interactions. We also plan to investigate factors, such as the type and size of a social group, that affect the appropriateness of a breakpoint perceived by the smartphone users. After applying more robust techniques for detecting social groups and smartphone placement to SCAN, we will conduct an in-the-wild study. We believe SCAN presents a model for managing notifications in an unobtrusive manner, helping people focus better on the social interaction.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful reviews and constructive comments. We thank Kiseok Kim, Sunwoo Kim, and all KAIST NMSL members for their invaluable help. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No.2016R1A2B4014068) and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (R0115-16-1006, Development of Data Mining Core Technologies for Real-time Intelligent Information Recommendation in Smart Spaces).

## 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. Xavier Alameda-Pineda, Yan Yan, Elisa Ricci, Oswald Lanz, and Nicu Sebe. 2015. Analyzing Free-standing Conversational Groups: A Multimodal Approach. In *Proceedings of the 23rd ACM International Conference on Multimedia (MM '15)*. ACM, New York, NY, USA, 5–14. DOI: **http://dx.doi.org/10.1145/2733373.2806238**

3. Morgan G. Ames. 2013. Managing Mobile Multitasking: The Culture of iPhones on Stanford Campus. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 1487–1498. DOI: **http://dx.doi.org/10.1145/2441776.2441945**

4. Brian P Bailey and Joseph A Konstan. 2006. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in human behavior* 22, 4 (2006), 685–708.

5. Alberto G Bonomi, AH Goris, Bin Yin, and Klaas R Westerterp. 2009. Detection of type, duration, and intensity of physical activity using an accelerometer. *Med Sci Sports Exerc* 41, 9 (2009), 1770–1777.

6. Jelmer P Borst, Niels A Taatgen, and Hedderik van Rijn. 2010. The problem state: a cognitive bottleneck in multitasking. *Journal of Experimental Psychology: Learning, memory, and cognition* 36, 2 (2010), 363.

7. Yung-Ju Chang and John C. Tang. 2015. Investigating Mobile Users' Ringer Mode Usage and Attentiveness and Responsiveness to Communication. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 6–15. DOI: **http://dx.doi.org/10.1145/2785830.2785852**

8. Alain De Cheveigné and Hideki Kawahara. 2002. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America* 111, 4 (2002), 1917–1930.

9. Adrienne Porter Felt, Serge Egelman, and David Wagner. 2012. I've Got 99 Problems, but Vibration Ain'T One: A Survey of Smartphone Users' Concerns. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '12)*. ACM, New York, NY, USA, 33–44. DOI: **http://dx.doi.org/10.1145/2381934.2381943**

10. Jason Gay. 2014. When Is It OK to Check a Smartphone at the Dinner Table? (Jun 2014). **http://on.wsj.com/12iWmq1**

11. Mayank Goel, Brendan Lee, Md. Tanvir Islam Aumi, Shwetak Patel, Gaetano Borriello, Stacie Hibino, and Bo Begole. 2014. SurfaceLink: Using Inertial and Acoustic Sensing to Enable Multi-device Interaction on a Surface. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1387–1396. DOI: **http://dx.doi.org/10.1145/2556288.2557120**

12. Carles Gomez, Joaquim Oller, and Josep Paradells. 2012. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors* 12, 9 (2012), 11734–11753.

13. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18. DOI: **http://dx.doi.org/10.1145/1656274.1656278**

14. Muhammad Haris, Hamed Haddadi, and Pan Hui. 2014. Privacy Leakage in Mobile Computing: Tools, Methods, and Characteristics. *arXiv preprint arXiv:1410.4978* (2014).

15. Joyce Ho and Stephen S. Intille. 2005. Using Context-aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 909–918. DOI: **http://dx.doi.org/10.1145/1054972.1055100**

16. Eric Horvitz, Johnson Apacible, and Muru Subramani. 2005. Balancing awareness and interruption: Investigation of notification deferral policies. In *User Modeling 2005*. Springer, 433–437.

17. Google Inc. 2016a. Accessibility — Android Developers. (2016). **https://developer.android.com/guide/topics/ui/accessibility/index.html.**

18. Google Inc. 2016b. Android Developers. (2016). **https://developer.android.com/index.html**

19. Google Inc. 2016c. Google Activity Recognition API. (2016). **https://developers.google.com/android/**

reference/com/google/android/gms/location/ActivityRecognitionApi

20. Shamsi T. Iqbal and Brian P. Bailey. 2006. Leveraging Characteristics of Task Structure to Predict the Cost of Interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 741–750. DOI: http://dx.doi.org/10.1145/1124772.1124882

21. 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

22. Haojian Jin, Cheng Xu, and Kent Lyons. 2015. Corona: Positioning Adjacent Device with Asymmetric Bluetooth Low Energy RSSI Distributions. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 175–179. DOI: http://dx.doi.org/10.1145/2807442.2807485

23. Minsam Ko, Seungwoo Choi, Koji Yatani, and Uichin Lee. 2016. Lock N' LoL: Group-based Limiting Assistance App to Mitigate Smartphone Distractions in Group Activities. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 998–1010. DOI:http://dx.doi.org/10.1145/2858036.2858568

24. Minsam Ko, Subin Yang, Joonwon Lee, Christian Heizmann, Jinyoung Jeong, Uichin Lee, Daehee Shin, Koji Yatani, Junehwa Song, and Kyong-Mee Chung. 2015. NUGU: A Group-based Intervention App for Improving Self-Regulation of Limiting Smartphone Use. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 1235–1245. DOI: http://dx.doi.org/10.1145/2675133.2675244

25. 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

26. Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. 2010. A survey of mobile phone sensing. *IEEE Communications magazine* 48, 9 (2010), 140–150.

27. Anthony Larcher, Jean-Francois Bonastre, Benoit GB Fauve, Kong-Aik Lee, Christophe Lévy, Haizhou Li, John SD Mason, and Jean-Yves Parfait. 2013. ALIZE 3.0-open source toolkit for state-of-the-art speaker recognition.. In *Interspeech*. 2768–2772.

28. Nilli Lavie. 2010. Attention, distraction, and cognitive control under load. *Current Directions in Psychological Science* 19, 3 (2010), 143–148.

29. Claire Lee. 2014. Risks of Korean eating habits. (jun 2014). http://www.koreaherald.com/view.php?ud=20140627001023

30. Uichin Lee, Joonwon Lee, Minsam Ko, Changhun Lee, Yuhwan Kim, Subin Yang, Koji Yatani, Gahgene Gweon, Kyong-Mee Chung, and Junehwa Song. 2014b. 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

31. Youngki Lee, Chulhong Min, Chanyou Hwang, Jaeung Lee, Inseok Hwang, Younghyun Ju, Chungkuk Yoo, Miri Moon, Uichin Lee, and Junehwa Song. 2013. SocioPhone: Everyday Face-to-face Interaction Monitoring Platform Using Multi-phone Sensor Fusion. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*. ACM, New York, NY, USA, 375–388. DOI: http://dx.doi.org/10.1145/2462456.2465426

32. Yu-Kang Lee, Chun-Tuan Chang, You Lin, and Zhao-Hong Cheng. 2014a. The dark side of smartphone usage: Psychological traits, compulsive behavior and technostress. *Computers in Human Behavior* 31 (2014), 373–383.

33. Markus Löchtefeld, Matthias Böhmer, and Lyubomir Ganev. 2013. AppDetox: Helping Users with Mobile App Addiction. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia (MUM '13)*. ACM, New York, NY, USA, Article 43, 2 pages. DOI: http://dx.doi.org/10.1145/2541831.2541870

34. 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

35. Shalini Misra, Lulu Cheng, Jamie Genevie, and Miao Yuan. 2016. The iPhone effect the quality of in-person social interactions in the presence of mobile devices. *Environment and Behavior* 48, 2 (2016), 275–298.

36. Carol Moser, Sarita Y. Schoenebeck, and Katharina Reinecke. 2016. Technology at the Table: Attitudes About Mobile Phone Use at Mealtimes. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1881–1892. DOI: http://dx.doi.org/10.1145/2858036.2858357

37. Darren Newtson and Gretchen Engquist. 1976. The perceptual organization of ongoing behavior. *Journal of Experimental Social Psychology* 12, 5 (1976), 436–450.

38. Tadashi Okoshi, Jin Nakazawa, and Hideyuki Tokuda. 2014. Attelia: Sensing User's Attention Status on Smart Phones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 139–142. DOI: **http://dx.doi.org/10.1145/2638728.2638802**

39. Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015. Reducing Users' Perceived Mental Effort Due to Interruptive Notifications in Multi-device Mobile Environments. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 475–486. DOI: **http://dx.doi.org/10.1145/2750858.2807517**

40. Antti Oulasvirta, Tye Rattenbury, Lingyi Ma, and Eeva Raita. 2012. Habits make smartphone use more pervasive. *Personal and Ubiquitous Computing* 16, 1 (2012), 105–114.

41. Veljko Pejovic, Mirco Musolesi, and Abhinav Mehrotra. 2015. Investigating The Role of Task Engagement in Mobile Interruptibility. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15)*. ACM, New York, NY, USA, 1100–1105. DOI: **http://dx.doi.org/10.1145/2786567.2794336**

42. Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014. 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**

43. Lee Rainie and Kathryn Zickuhr. 2015. Americans' Views on Mobile Etiquette. Pew Research Center. (August 2015). **http://www.pewinternet.org/2015/08/26/americans-views-on-mobile-etiquette/**

44. 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**

45. Choonsung Shin and Anind K. Dey. 2013. Automatically Detecting Problematic Use of Smartphones. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 335–344. DOI: **http://dx.doi.org/10.1145/2493432.2493443**

46. Joren Six, Olmo Cornelis, and Marc Leman. 2014. TarsosDSP, a real-time audio processing framework in Java. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society.

47. 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.

48. Zhixian Yan, Jun Yang, and Emmanuel Munguia Tapia. 2013. Smartphone Bluetooth Based Social Sensing. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp '13 Adjunct)*. ACM, New York, NY, USA, 95–98. DOI: **http://dx.doi.org/10.1145/2494091.2494118**

49. David G. Young. 2016. AltBeacon. (2016). **https://github.com/AltBeacon/android-beacon-library**