February 15, 2018

Notifications management across devices

Kyle Schumacher

Jason Sanio

Notifications management across devices

ABSTRACT

It is common for an individual user to own multiple electronic devices, e.g., laptop, smartphone, etc. Applications on these devices often provide notifications even when not explicitly opened or accessed by the user. Notifications that have been dismissed by the user on one device may continue to persist as zombie notifications on other devices, leading to a suboptimal user experience. Techniques of this disclosure manage notifications across devices such that notifications are synchronized across devices. With user permission, the techniques enable targeting a notification to an active device and/or a device that is appropriate for the user to act on the notification. A notification hub is implemented as a single source of notifications across a range of applications and services. The notification techniques enable users to have more control and deliver an improved notification.

KEYWORDS

- Notification
- Alert
- Notification synchronization
- Cross-device notification
- Notification management

BACKGROUND

Many users own or operate multiple electronic devices, e.g., smartphone, tablet, smartwatch, laptop, desktop computer, etc. To accommodate usage across devices, applications are often designed to be usable on multiple devices. Moreover, applications often issue notifications even when they are not explicitly opened by a user. Multiple notifications across devices can sometimes lead to a frustrating user experience.

For example, an important notification may appear on a currently inactive device while remaining unissued on the active device. Conversely, a notification that was dismissed or actioned upon by the user on one device can persist on other devices, such that the user has to act upon the same notification across multiple devices. This can lead to a bad user experience, e.g., a user that infrequently uses a tablet is bombarded with stale notifications from events that have already been acted upon from other devices, and in some cases, being forced to wait for the stale notifications to load. There are several reasons for the lack of synchronism and inaccurate targeting of notifications, including differing application and system software that behaves differently across devices.

Further, some notifications are not conveniently actionable on certain devices. For example, a notification for an email that requires a detailed response to be composed is better handled on a device with a larger screen (e.g., laptop or desktop computer) than a device with a smaller or no screen (e.g., smartwatch or other wearable device, smartphone). Some current techniques enable a reminder to be provided to a user that dismisses a notification. However, but this is less than satisfactory for the user, as the user does not know with any precision the next time of use of a device appropriate for the notification.

DESCRIPTION

To improve user experience, it is desirable that a user be able to seamlessly switch back and forth across devices, e.g., pick up a task on laptop at the precise point where they left off on phone. Accurately synchronized and targeted notifications are important for such seamless user experience. Further, application developers benefit from being able to deliver synchronized and targeted notifications.

Techniques of this disclosure synchronize and accurately target notifications across user devices. With user consent and permission, the device currently in use (active device) is determined and notifications are delivered to just that device. Further, when notifications are sent to more than one user device, and the user dismisses, snoozes, or otherwise acts upon a particular notification from one device, the notification is pulled back from other devices.

When a user provides permission, the techniques can determine the currently active user device using machine learning techniques. Machine learning may be applied, e.g., to aggregate and/or user-specific historical patterns of usage; real-time authentication data; device movement; etc. based on data that the user provides permission to use. Further, it is determined if a particular notification is actionable on the currently active device. If the notification is not actionable, the user is prompted to set up a reminder for the notification when the user next uses a device suitable for the notification.

In this manner, the techniques enable provisional dismissal of notifications from a device until action is taken on another device. The techniques utilize a notification server or hub which, with prior user consent and permission, receives and manages notifications from applications installed, services subscribed to, or websites accessed by the user. When an action relating to a notification is taken on a particular device, information relating to the action is synchronized with the notification hub and pushed to other devices of the user.

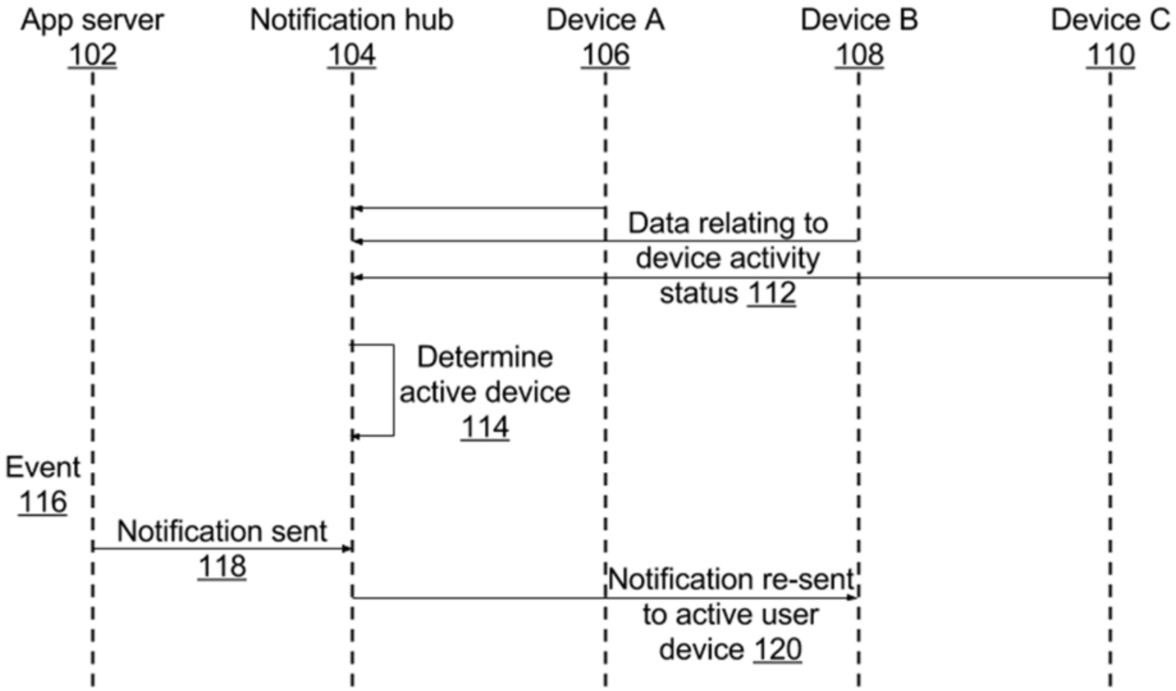
App server
102
Notification hub
104
Device A
106
Device B
108
Device C
110
Data relating to device activity status 112
Determine active device 114
Event 116
Notification sent 118
Notification re-sent to active user device 120

Fig. 1: Targeting a currently active device for sending notifications

Fig. 1 illustrates an example workflow to synchronize across multiple devices for the purposes of targeted notifications, per techniques of this disclosure. Application server (102) provides applications, websites, or other services that the user subscribes to and that periodically send notifications to the user. For example, the application server can be an email application, a travel website, an online calendar or meeting service, etc.

Upon user permission, notification hub (104) receives from the app server notifications addressed to the user. With user permission, the device that the user is currently engaging with is determined (114) based on real time data (112) from various devices (106-110) owned or operated by the user. Users can provide permission for any such data, including for example, logged-in / authentication status to a service on a device (e.g., a user may be logged in to a web-browser on a desktop or other device known to be stationary); machine-learned historical patterns of device usage (e.g., a user may be logged in to a desktop but at certain times of the day is not usually at desk); detection of movement or proximity to human body of a user-owned mobile device; etc. The application server registers (116) an event, e.g., the receipt of a chat message, notice of a delayed flight from a travel website, etc. The application server sends the notification (118) to the notification hub, which in turn pushes the notification to the currently active device (120) and not to other devices owned or operated by the user.

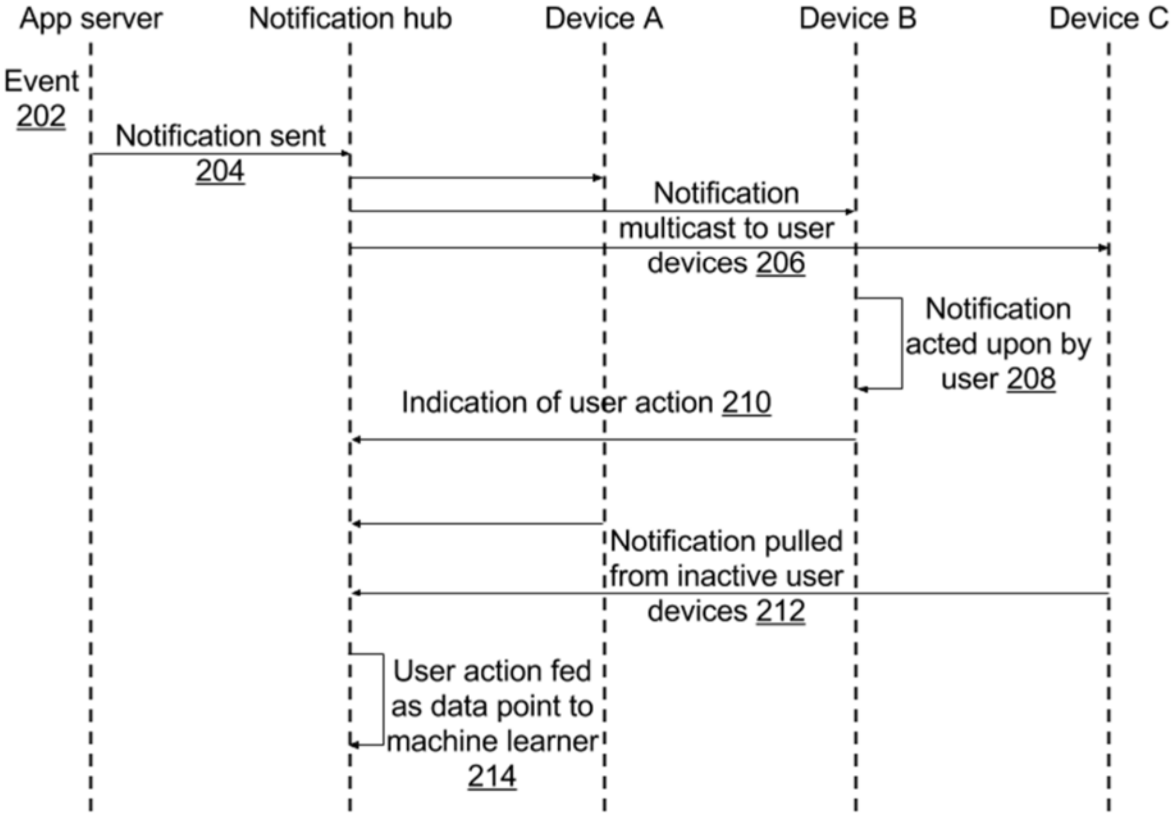
App server
Notification hub
Device A
Device B
Device C
Event 202
Notification sent 204
Notification multicast to user devices 206
Notification acted upon by user 208
Indication of user action 210
Notification pulled from inactive user devices 212
User action fed as data point to machine learner 214

Fig. 2: Workflow to synchronize a notification across multiple devices

In certain situations, it is possible that the correct device to send notifications to is not identified based on the available user data. Fig. 2 illustrates an example workflow to synchronize notifications in such situations. Upon occurrence of an event (202), the application server sends a notification for the event to the notification hub (204). If the active device for the user is not determined, the notification hub multicasts the notification (206) to devices known to be owned or operated by the user.

When the user sees the notification on a particular device, the user may act upon the notification (208). User action could comprise, e.g., dismissal or snoozing of the notification, or other actions based on the notification. With user permission, the notification hub receives indication of the user action (210) and pulls the notification back from other devices (212). In this manner, no presentation of stale notifications, e.g., notifications that were previously dismissed or acted upon, is made to the user at the other devices. When the user provides consent, user actions on the notification can be provided as data points to a machine learning model and used to refine understanding of user behavior (214). For example, when such training data is provided, the model can learn a correlation between work-related email notifications that arrive on a weekend and a user action to snooze such notifications, e.g., until the next work day.

The below examples illustrate the improved delivery of notifications per the techniques of this disclosure. In these examples, the user has enabled access to user data for the specific purpose of management and deliver of notifications.

Example 1. *To-be-actioned resource unavailable on device*: A user receives a chat notification on a smartphone about an incoming chat message that includes a request for a document. With user permission, it is determined that the requested document is not available on the smartphone or an accessible online file-storage service. Further, it is determined that the document is in fact available on local storage of another user-operated device. Based on the context of the notification, and determination of the location of the document (and that the document is not available on the active device), it is determined that the chat notification should be delivered again when the user is active on another device, e.g., that stores the requested document.

Example 2. *Action that requires deliberate composition*: A smartphone user receives a notification about an email that is likely to require a lengthy, thoughtful response. Composing such a response is inconvenient on a small-screen device, e.g., that lacks a physical keyboard. Based on the context of the notification, and on usage patterns, it is inferred that the notification should be provided again when the user is active on a suitable device, e.g., a desktop or laptop computer.

Example 3. *Action that requires a specific time or place*: The user receives a work-related email notification on a weekend. Based on usage patterns, it is determined that the user typically responds to such notifications on the immediately following weekday from a work device. The notification is provided again on the work device when the user activates the work device on the following weekday.

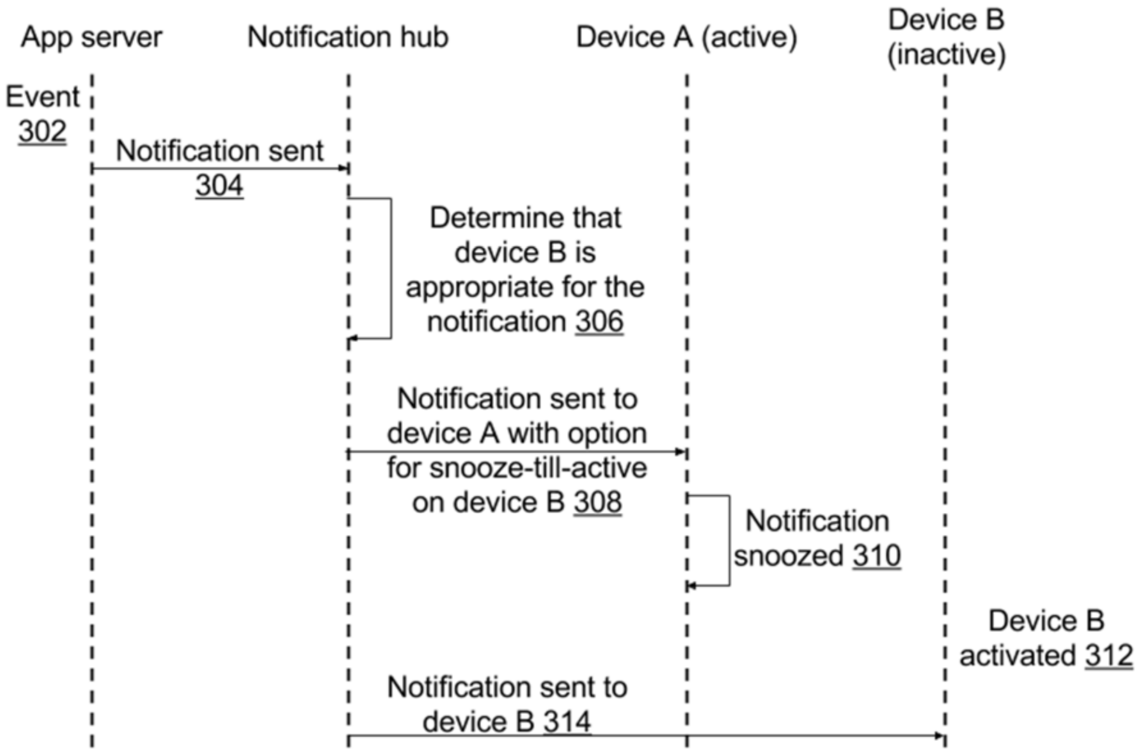
App server
Notification hub
Device A (active)
Device B (inactive)
Event 302
Notification sent 304
Determine that device B is appropriate for the notification 306
Notification sent to device A with option for snooze-till-active on device B 308
Notification snoozed 310
Device B activated 312
Notification sent to device B 314

Fig. 3: Workflow to determine an appropriate device to action a notification

Fig. 3 illustrates an example workflow to determine an appropriate device for the user to take action on a notification, per techniques of this disclosure. The application server experiences an event (302) and sends a notification (304) to the notification hub. It is determined that the active user device (device A) is not suitable for the notification and that a different device, device B that is currently inactive is appropriate (306).

The notification hub sends a notification to the user on device A (308). The notification includes an option to re-notify the user when device B is activated. For example, the option can be provided with a prompt, e.g., “reply or snooze notification until on work computer?” Upon user selection to snooze the notification (310), the notification is removed from device A.

At a later time, it is detected that device B is activated (312). The notification hub re-sends the notification to device B (314). The techniques are also support compound actions, e.g., a user can snooze a notification with the instruction “remind me next week when I am on my tablet.”

The described techniques also conserve battery by minimizing notifications to idle devices.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user’s social network, social actions or activities, user actions and history on a device or applications therein, user’s profession, a user’s preferences, user’s current location, etc.), and if the user is sent content or communications to/from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user’s identity may be treated so that no personally identifiable information can be determined for the user, or a user’s geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

The techniques of this disclosure enable users to dispose a notification synchronously across different devices. Zombie notifications that persist on a device after being dismissed on another device are eliminated. The techniques enable users to snooze notifications until a later time, e.g., when a preferred device is activated, or at a preferred time or place. With user permission, the techniques enable targeting a notification to an active device and/or a device that is appropriate for the user to act on the notification. With user permission, trained machine learning models are utilized to identify usage patterns and context to selectively present notifications and associated options. A notification hub is implemented as a single source of notifications across a range of applications and services. The notification techniques enable users to have more control and deliver an improved notification experience. The techniques also conserve battery by minimizing notifications to idle devices.