November 2022

Quick Multi Device App Installation Using a Prediction Engine

D Shin

Quick Multi Device App Installation Using a Prediction Engine

ABSTRACT

When users have multiple devices, one of the devices can act as a host device where an app is downloaded. Upon pairing with a target device, the app is transferred to the target device from the host device. This enables a user to use their preferred device to search for and obtain apps, while also providing the apps on the user's other devices. However, this protocol is impeded by the wait to establish connectivity between the devices and by the time needed for app transfer. This disclosure describes techniques that, with user permission, use predictive signals to generate a dictionary of apps for a user's devices. After the user completes app installation on a host device and the device is paired with a target device, a simple, host-target key-match is used to identify apps in the dictionary that are to be installed on the target device. This approach can reduce installation time and provide an immediate and seamless user experience for app installation.

KEYWORDS

- Application store
- App install
- Mobile apps
- Device ecosystem
- App prediction
- Predictive dictionary
- Prediction engine

BACKGROUND

Many users own multiple devices such as smartphones, smartwatches, smart automobile dashboards, smart speakers, etc. While some devices have direct network connectivity and can directly access online application stores to install apps, such capability is not necessarily available on all devices. Further, to search or browse for apps, a user may prefer the user interface of an application store on a particular device (e.g., a large screen laptop or tablet) rather than a smaller device such as a smartphone or a smart speaker.

To install an app on target devices without browsing for the app on that device, one current mechanism is to use a host device such as a laptop to search an online application store for an app to install on a target device such as a smartphone. The laptop (an intermediate host device) can be used to navigate the application store and download a copy of the desired app for the target device. The downloaded app is then installed on the target device without the user accessing the application store from the target device. Another example mechanism is to install (or download) the app on a particular target device such as a smartphone that can then act as a host device for subsequent installs on other target devices. For example, the app can be installed on an automobile dashboard when the smartphone is connected, e.g., via a wire, via Bluetooth, etc. to the automobile dashboard.

In the above mechanisms, waiting for the connection to install the app from the host to the remaining target devices constitutes a bottleneck. Downloading the app from the host device to the target device requires host-target pairing and can also interfere with the usage or functionality of the host and/or the target devices during installation.

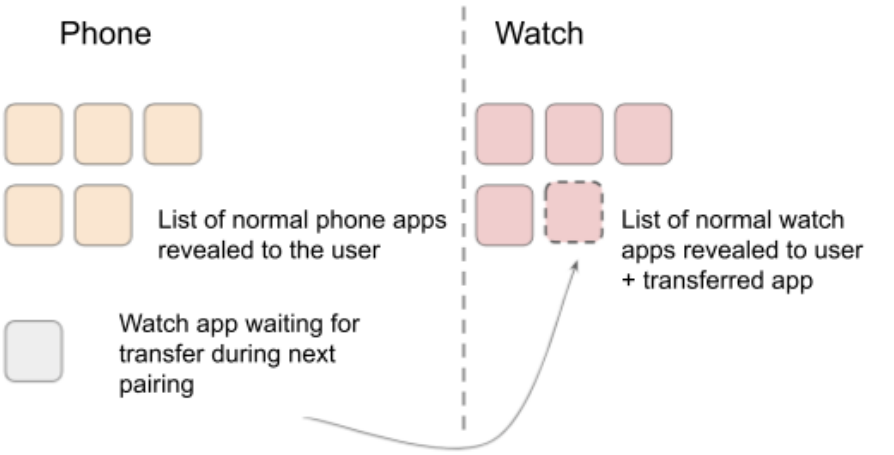

Fig. 1: Conventional multi-device app installation

Fig. 1 illustrates the conventional protocol for multi device app installation with the example of a smartphone and a smartwatch. As illustrated, the phone (a host device) stores the watch app intended for installation on the watch (target device). When the phone and the watch are paired, the app is transferred to the watch.

DESCRIPTION

This disclosure describes techniques to periodically create a dictionary of apps that are common to the multiple devices used by a user. To implement the described techniques, user permission is obtained to access the list of user's devices and apps installed on each device (or a subset of devices permitted by the user. The dictionary of apps is built using predictive signals (accessed with user permission) and is referred to as a predictive dictionary. It is encrypted and is private to the user. When the user completes an app installation journey (for one or more apps) on a host device (as described above), during pairing time, a simple key-match between host and target device is used to identify apps in the dictionary that are to be installed on the target device. This can reduce installation time and provide an immediate and seamless user experience.

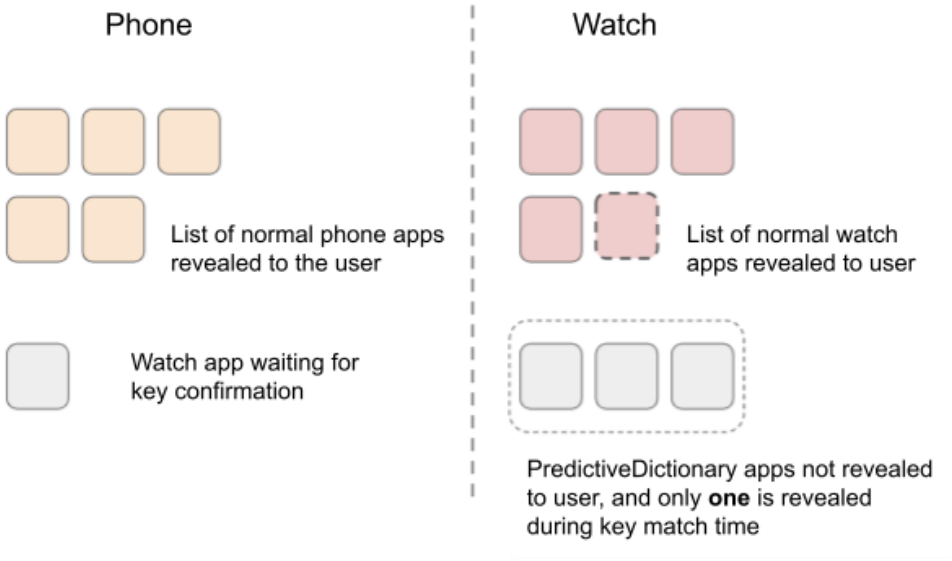

Fig. 2: Multi device app installation using a predictive dictionary

Fig. 2 illustrates multi device app installation based on a predictive dictionary, using the example of a smartphone and a smartwatch. As illustrated, on the phone (the host device), the watch app is not necessarily downloaded; rather, only a key for the app is stored. Prior to app download time, with user permission, the predictive dictionary creates and updates in the memory of the watch (the target device) a set of apps, revealable to the user upon request. The set of apps are identified based on user preference signals (obtained with user permission), such as genre of prior app downloads, current list of apps on user devices, general popularity of apps (e.g., prior distribution), net preferences among friends, etc.

Apps in the predictive dictionary are periodically updated, e.g., once a day. The updating period is tunable. At the start of a period, a predictive engine gathers up to date predictive signals and parses the available signals to generate an updated list of apps. The most recent list of apps can be different from that of the previous period(s). The list of apps is updated on the watch (the target device).

The predictive engine can be implemented using any suitable technique, e.g., a neural network that receives as inputs contextual cues represented in signal form. The neural network can take the form of a fully-connected (FC) architecture that converts the integer/floating vector into a one-hot encoding of the N, e.g., 10,000, most popular apps on an application store. The number of popular apps, N, can be a variable hyperparameter of the neural network. The FC can be roughly, e.g., 10 layers deep and 100 nodes wide, although the exact parameters of the FC may depend on how extensive the input contextual cues are.

The list of predictive dictionary apps can contain the top M apps with the largest SoftMax scores from the above prediction result. Note that M is less than or equal to N, as apps that are not predicted cannot be surfaced. Typically, M can be set at or near 5. The exact value varies with the device and the available memory or free storage space on the device.

Key confirmation does not require an actual transfer of a large dataset describing the app itself and can therefore happen rapidly. After key confirmation, one app from the list of predictive dictionary apps on the target device (the smartwatch in the above example) is identified as the app of interest. Since that app is likely already pre-downloaded on the basis of the recommendations of the predictive dictionary, the app can be readily installed on the target device.

In the event that the key match fails, e.g., if the predictive dictionary does not include the app intended by the user, multi-device app installation reverts to the legacy protocol of transferring the app from a host device. The user experience in this case is identical to the current one, such that there is no net negative compared to the previous experience.

In this manner, the described techniques use a predictive dictionary to reduce the time needed for app installation on multiple devices. Effectively, some tasks related to app installation

to target devices are executed prior to the user request for app installation which reduces the time for installation.

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 the collection of user information (e.g., information about a user's devices, apps installed on devices, application store purchases, social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications 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

This disclosure describes techniques that, with user permission, use predictive signals to generate a dictionary of apps for a user's devices. After the user completes app installation on a host device and the device is paired with a target device, a simple, host-target key-match is used to identify apps in the dictionary that are to be installed on the target device. This approach can reduce installation time and provide an immediate and seamless user experience for app installation.