International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:7, No:12, 2013

Design of Middleware for Mobile Group Control in Physical Proximity

Moon-Tak Oh, Kyung-Min Park, Tae-Eun Yoon, Hoon Choi, Chil-Woo Lee

***Abstract*—**This paper is about middleware which enables group-user applications on mobile devices in physical proximity to interact with other devices without intervention of a central server. Requirements of the middleware are identified from service usage scenarios, and the functional architecture of the middleware is specified. These requirements include Group Management, Synchronization, and Resource Management. Group Management needs to provide various capabilities to such applications with respect to managing multiple users (e.g., creation of groups, discovery of group or individual users, member join/leave, election of a group manager and service-group association) using D2D communication technology. We designed the middleware for the above requirements on the Android platform.

***Keywords*—**Group user, middleware, mobile service, physical proximity.

I. Introduction

RECENTLY, the number of smart mobile devices is increasing rapidly and application services that interact with multiple, mobile users have become popular. Some examples of services in which groups of users participate are Facebook and Twitter. Most of these applications are provided and managed by a central server which may be located in a remote area; connected through the Internet. Therefore, mobile devices using such services need to be able to access the Internet. Such server-based service architecture has many weaknesses in traffic and security. As the number of users increase, the service quality degrades due to the heavy data traffic to the server [1], [2].

If users of an application service are close together, the service may consist of a small ad-hoc network of devices without a central server, and user applications on each of the user devices can control the service themselves in Fig. 1.

For nearby users to transfer data directly without a server, D2D (Device-to-Device) communication technology [3] may be used. By D2D communication, a device can send data directly to other devices without passing through a base-station infrastructure or wireless access point (AP). Wireless communication technologies for D2D include Bluetooth [4], Wi-Fi Direct [5], and Flash LinQ [6].

Communication middleware for managing data communication or for sharing computing resources or for controlling group members of different computing nodes, have been a classical subject in the field of desktop and server computers. However, middleware for controlling P2P-based services on mobile devices has not been studied a lot. Middleware for managing sensor nodes on a wireless sensor network [7], and middleware for member management on Ad-hoc networks [8] are some examples of the latter. Recently developed middleware called Chord is similar to the middleware proposed in this study, but Chord does not provide for management of information synchronization and sharing of general computing resources [9].

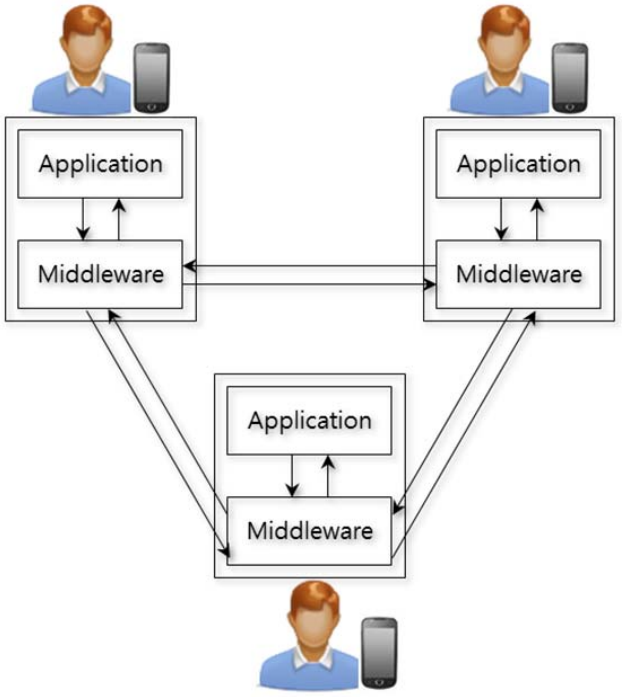

Fig. 1 Form of new mobile service

This paper is about new middleware which enables group-user applications on mobile devices in physical proximity to interact without servers. The requirements and functional architecture of the middleware are described in this paper. Some key requirements of the middleware are described in Section II. The functional architecture of the middleware, Part of the procedure for joining the group and synchronization of the proposed middleware are given in Section III. Concluding remarks are presented in Section IV.

Moon-Tak Oh, Kyung-Min Park, Tae-Eun Yoon, and Hoon Choi are with the Chungnam National university, 99 Daehak-ro, Yuseong-gu, Daejeon, Korea (e-mail: moontak0214@cnu.ac.kr, km-park@cnu.ac.kr, yteun@cnu.ac.kr, hc@cnu.ac.kr).

Chil-Woo Lee is with the Chonnam National university, 77 Yongbong-ro, Buk-gu, Gwangju, Korea (e-mail: leecw@chonnam.ac.kr).

II. Requirements

Requirements of the middleware were identified by the following steps. First, a usage scenario of the application service was described. Second, key functions needed to implement these scenarios were identified. Some of these functions may be provided by application logic, other functions may be provided by separate middleware. Therefore, the next step was to identify the key functions which may be common to many applications, and include them in a separate middleware.

A. Group Management

There are many applications which require participation of multiple users (i.e., groups of users). Middleware needs to provide various capabilities to such applications with respect to managing multiple users, including creation of groups, discovery of group or individual users, member join/leave, election of a group manager and service-group association.

The way to connect multiple users in the underlying communication interface may be any of a number of effective communication interfaces (Wi-Fi Peer-to-peer, Mobile hotspot or Wireless AP). Regardless of which communication interface is used, middleware needs to provide a logical view of group structure and a consistent interface to manage the groups. The logical group structure is shown in Fig. 2. A top-level group consists of devices that use the same physical channel of the communication interface. Groups may have multiple subgroups which are called the service-groups. The service-group consists of members of the group, and is formed for each application service. The members of a service-group use the same application-service associated with that service-group. The middleware needs to provide two levels of groups because users that are close together may use several services in parallel, or may switch from one service to another. It is inefficient to change the physical channels of devices for each subgroup whenever a user switches application services. Changing the service-group of a member is implemented by authenticating the member and verifying his service profile, which is much easier than changing the physical communication channel.

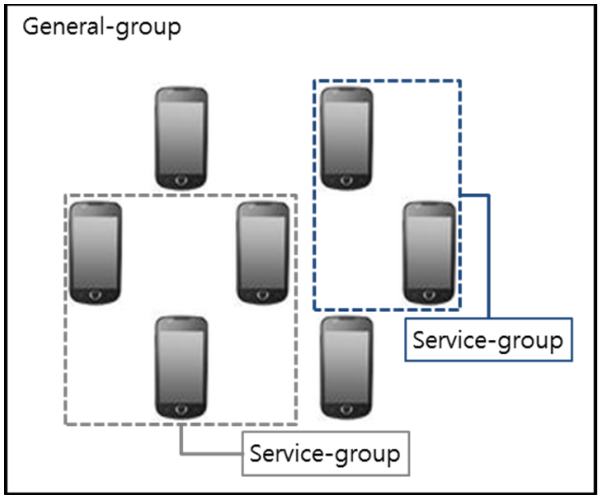

Fig. 2 Group structure

B. Synchronization

A variety of information is exchanged between in the group. Most of this information is governed by the application, but some of the exchange needs to be done by the underlying middleware. For example, if an application of a member shares picture images or files with another member application, then the middleware needs to synchronize multiple access requests to storage. Maintaining a common view of group-member lists on different mobile devices is another example. Synchronization provides a common view of the information distributed among mobile devices within the groups. In additional, the middleware needs to synchronize information about events that occur during service.

- Object resources: clock, memory(photo album, contact information, file), peripherals(speaker, mike, camera)
- Synchronization operations:

- Real-time control for running multimedia synchronization
- Reliable or unreliable transmission of event occurrences and results(1:1, 1:N, request-reply)
- Synchronization of resource profiles
- Synchronization of group member views

C. Resource Management

Mobile devices in the group may share hardware resources. By using the resources of other devices, group applications can be enhanced to provide higher-level service. For example, a single camera can take a picture from one direction only, but augmentation using the camera modules of three other devices, would allow three dimensional and directional pictures to be taken at the same time. If group applications share memory and files, file transfer can be performed much faster; like the torrent service does in distributed desktop computing systems.

For all these purposes, the middleware needs to coordinate the operating systems of each mobile device to manage hardware resources efficiently. When a user joins a service-group, the middleware must authenticate the user and authorize access to hardware resources in order to block access by non-member users.

- Object resources: processor, memory(photo album, contact information, file), peripherals(speaker, mike, view, camera)
- Management operations:

- Authentication of members, control access to shared resources
- Management of profile information for resources
- Judgment of availability and interoperability of shared resources required by an application using resource profiles
- Control of conflicts on a shared resource
- Information transmission function (RPC, FTP)

III. System Architecture

We designed middleware which satisfies the above requirements on the Android platform in Fig. 3. Android

version 4.0 or later is required because these versions support Wi-Fi Peer-to-peer (P2P) [10], and Wi-Fi P2P is a dominant network interface for communication with group members. Our middleware consists of six modules and is located at the same level as the Android application framework.

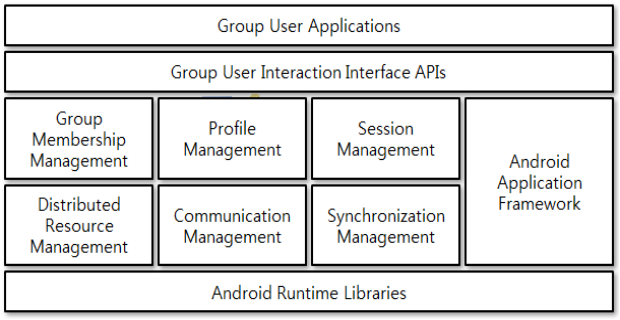

Fig. 3 Group-user middleware architecture

Communication Management provides common APIs to the upper layer by interacting with various network interfaces of Android Libraries. This is because the middleware needs to handle group communications the same way even though the network interface for each group is different.

Profile Management handles information about user and hardware resources used by the middleware, as explained in Section II *C*, and Group Membership Management performs the functions explained in Section II *A*.

Session Management maintains connections by exchanging messages between members periodically when one member connects to multiple other members. Also, it releases a session when it judges a connection is disconnected, even though the event of leaving a group has not explicitly occurred. In this way, the middleware can reduce consumption of resources.

Distributed Resource Management obtains resource profiles from the Profile Management module then uses them to determine the interoperability and availability of hardware resources, when an application service needs the hardware resources of another device. If the hardware resource is available to use, the middleware grants group members access rights to this resource. If not, the middleware notifies Group Membership Management by creating an event which notifies it that group members cannot use these resources.

Synchronization Management provides synchronization capability for Distributed Resource Management and Group Membership Management.

The operation procedure of each module of the middleware is described in sequence diagrams. For example, Fig. 4 shows the procedure for a user to join a specific group. This sequence diagram is composed of the following steps.

- Step 1: A user discovers available groups
- Step 2: A user selects a group and joins after authentication
- Step3: After joining the group, the middleware synchronizes changed information with other group members

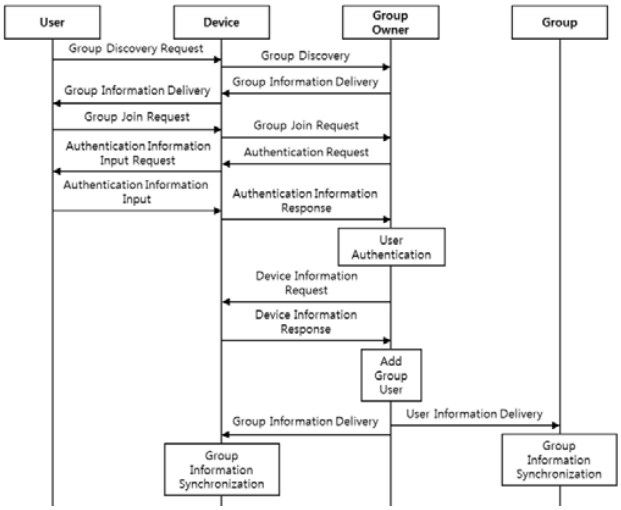

Fig. 4 Sequence diagram for joining the group

IV. Conclusion

The middleware proposed in this paper is designed to provide P2P group communication and resource sharing of mobile devices for applications. One of the key benefits of the middleware is that it avoids generating network traffic to the backbone network and server. This middleware has only core features for group services because it is designed to run on light-weight mobile devices. Other enhanced features for group services may be offered by the application itself.

Acknowledgment

Chil-Woo Lee was supported by the MSIP, Korea, under the ITRC support program (NIPA-2013-H0301-13-3005) supervised by the NIPA (National It Industry Promotion Agency).

Oh, Park, Yoon, Choi were financially supported by the ministry of education (MOE) and NRF of Korea through the human resource training project for regional innovation (No. 2013H1B8A2032180).

References

[1] Wilson, Christo, et al. "Better never than late: Meeting deadlines in datacenter networks," *ACM SIGCOMM Computer Communication Review*, Vol. 41, No. 4, pp.50-61, 2011.
[2] Gill, Phillipa, Navendu Jain, and Nachiappan Nagappan. "Understanding network failures in data centers: measurement, analysis, and implications," ACM SIGCOMM Computer Communication Review, Vol. 41, No. 4, pp.350-361, 2011.
[3] Doppler, Klaus, et al. "Device-to-device communication as an underlay to LTE-advanced networks," IEEE Communications Magazine, Vol. 47, No. 4, pp.42-49, 2009.
[4] Bluetooth, https://www.bluetooth.org.
[5] Wi-Fi Peer-to-peer, www.wi-fi.org/discover-and-learn/wi-fi-direct.
[6] FlashLinQ,http://www.qualcomm.com/media/releases/2011/02/08/qualcomm-demonstrate-new-peer-peer-technology-mobile-world-congress.
[7] Vieira, Mardoqueu and Rosa, Nelson, "A reconfigurable group management middleware service for wireless sensor networks," The 3rd ACM International Workshop on Middleware for Pervasive and Ad-hoc Computing, pp.1-8, 2005.
[8] Bottzzi, Dario and at al., "A context-aware group management middleware to support resource sharing in MANET environments," The

6th ACM International Conference on Mobile Data Management, pp. 147-151, 2005.
[9] Samsung Inc., Samsung Chord SDK Developers Guide, http://developer.samsung.com/chord, 2013.
[10] Wi-Fi Peer-to-peer support according to Android version, http://developer.android.com/guide/topics/connectivity/wifip2p.html.