ColPhone: A Smartphone is just a piece of the puzzle

Ahmed Salem
Computer Science Department
Old Dominion University
4700 Elkhorn Ave, Suite 3300
Norfolk, VA 23529 USA
asalem@cs.odu.edu

Tamer Nadeem
Computer Science Department
Old Dominion University
4700 Elkhorn Ave, Suite 3300
Norfolk, VA 23529 USA
asalem@cs.odu.edu

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
Ubicomp '14 Adjunct, September 13-17 2014, Seattle, WA, USA
ACM 978-1-4503-3047-3/14/09.
http://dx.doi.org/10.1145/2638728.2638743

Abstract

Multiple smartphone coexistence has been a fact of life. However, each smartphone is though of as a single unit. We believe that cooperation between coexisting smartphones can provide users with a cheap hardware upgrade (e.g processing). In addition to sharing of sensing information that can be collected by only one phone saving others energy of redundant tasks (e.g. GPS readings). In this work, we propose COLlaboration smartPHONE (colPhone) a framework that manages the collaboration between smarphones. ColPhone aims to achieve mutual benefit for collaborators by utilizing idle resources on smartphone in the proximity.

Author Keywords

Smartphone; Collaboration, Android; Bluetooth; WiFi-P2P; Pure Data.

ACM Classification Keywords

H.5.3 [Group and Organization Interfaces]: Collaborative Computing.

Introduction

Mobile devices yet smartphones have made the break through to human lives. It is very likely to have plenty of smartphones in the same proximity due to their owners motion, or availability of old unused ones. Each phone

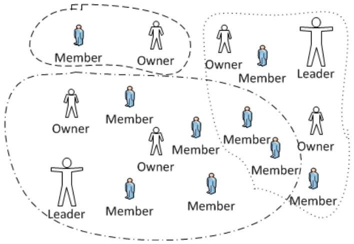


Figure 1: Groups

holds plenty of processing, sensing, and storage capabilities. In this work we propose a framework to utilize this frequently occurring scenario.

Smartphone collaboration have been addressed in infrastructure based networks from some applications. Migrating Peer-to-Peer applications to smartphones [4]. Sharing sensing information was addressed either analytically [3] or both analytical and empirical as in CoMon [5]. Controlling the I/O resources of one phone (e.g. Camera, Speaker) through another [2]. In this work we propose a framework to utilize the proximity coexistence to build an infrastructureless system.

ColPhone is a framework that facilitates a co-ordinated cooperation between smart devices available in the proximity. It provides smartphone users access to a new set of resources without changing their devices. Smart devices in the proximity can share a portion of their idle resources to be used by others. In return they will receive the same feature upon need, in addition to the results of previous operations (e.g. sensing information) to avoid duplicated processing and power misuse. Further, this new approach opens the horizon for application that was not possible on one device. 3D story teller is an example where a story can be read from multiple device (e.g. one character per device). The devices' location and character distribution among devices can emulate the story as a movie [6].

Framework

ColPhone is a role oriented architecture where each smartphone can play one or more roles throughout the execution life-cycle. In this section we will explain the roles used, the features every role has as shown in Fig 2.

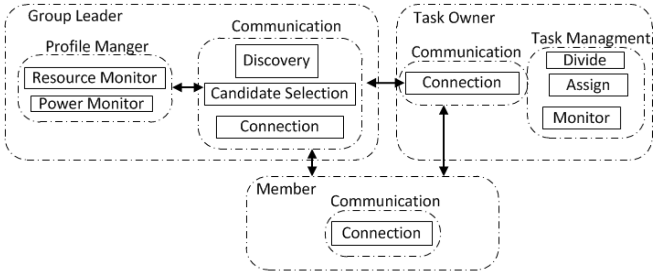


Figure 2: Framework.

Roles and Players
Roles are functional behaviors that can be played by every smartphone. We define three roles for our scheme as shown in Fig. 1 :

Group Leader: the role responsible for the management and monitoring of task execution. Further, this role is responsible for coordinating between the task owner role players.

Task Owner: the role of the node issuing the task to be performed. This role has four main jobs: First, task breakdown into services that can run on different devices in parallel; Second, request a list of recruited members from the group leader and assign services to them; Third, delegate the execution monitoring to the group leader; Finally, Collect the resulting feedback from members.

Group Member: the role responsible for following the instructions received from the group leader to achieve a collaboration architecture.

Task owner role is introduced to enhance *ColPhone* performance. It is likely for wireless networking protocols to have a group leader that is responsible for managing and controlling members. The infrastructure-less nature

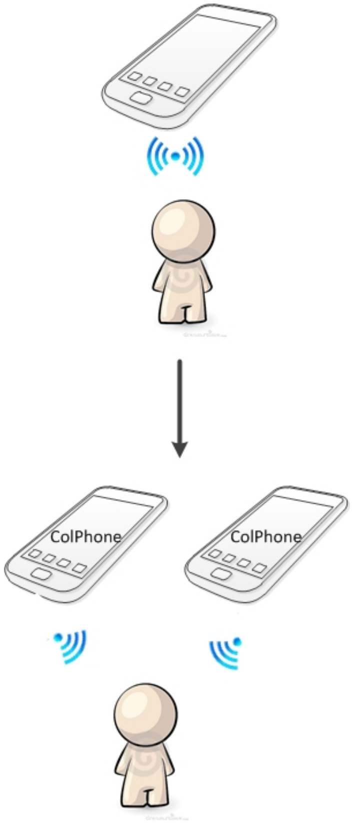


Figure 3: 3D Story Teller

of *ColPhone* in addition to the expected multiple service dissemination is likely to decrease the network throughput and consume the smartphone power inefficiently if done through the group leader.

Functional Modules
ColPhone framework in Fig. 2 has three main functional modules Communication, Profile Management, and Task Management.

The communication module is responsible for finding available smartphones in the proximity through device discovery and candidate selection modules. The *Device Discovery* module senses the medium for available devices. The list of available smartphones are then passed to the *Candidate Selection* module, which sorts them based on their readiness for collaboration. This readiness is categorized in terms of available resources, power levels, and collaboration enrollment. First, the devices that can not provide the requested resources for collaboration are excluded from the list. The remaining smartphones that are currently charging have the highest priority followed by others according to their current power levels. Finally, devices that are currently sharing or have collaboration services to execute are relocated to the end of the list.

The Profile Management module is responsible for providing the *Candidate Selection* with the required in information gathered through it's two sub-modules *Power Monitor*, and *Resource Monitor*. These information is used for sorting of the available smartphones.

The Task Management Module is responsible for splitting the available task into multiple services based on the number of the available smartphones, assigning the services to different smartphone, and monitoring their execution. In case of an execution problem on a smartphone (e.g. it moved out of the communication range), the task manager should assign that service to another smartphone to continue its execution.

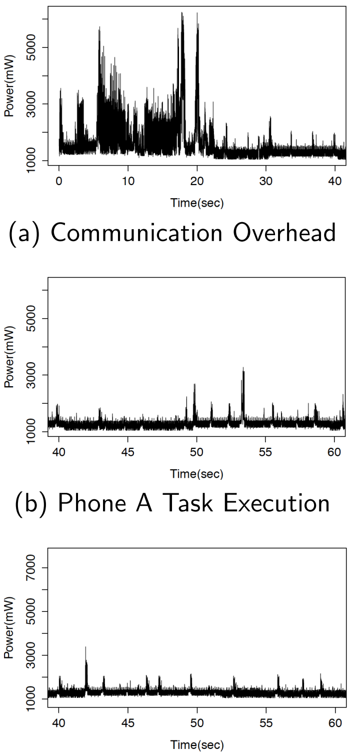
3D Story Teller Prototype

Media applications is a popular use of smartphones. 3D Story Teller besides being a normal audio application, it provides the possibility of playing the audio track on multiple devices as shown in Fig. 3. The application's idea is based on the fact that people enjoy movies in cinemas more than TVs, because of the difference in sound effects. 3D Story Teller using the features provides by *ColPhone* , splits audio tracks by character to be played on multiple devices. For example, It is now very likely that kids listen to stories through smartphones. 3D Story Teller will enable the sound of each character in the story to come from a different device. We believe that this can provide kids with a more engaging story time.

Typical Scenario
Upon enabling the collaboration feature, the user plays his audio track as normal from his device. The user's device is now playing the *task owner* role. It then communicates with the *task manager* role to request a list of available smartphones for collaboration specifying the set of requested resources, in this case speakers.

Task manager role being in action uses the features explained before from the *Communication* and *Profile Management* modules to sense the medium and provide the user with a list of devices ready to collaborate.

Once the *task owner* receives the list, it splits the current audio track to N sub-tracks, where N is minimum of characters in the audio track ,and the number available smartphones willing to collaborate. The N sub-tracks are then forwarded to the smartphones, and starts to play. Collaborating smartphones are supposed to keep contact with the *task owner* throughout the execution period. In



(a) Communication Overhead

(b) Phone A Task Execution

(c) Phone B Task Execution

Figure 4: Power Consumption.

case of a smartphone quitting the collaboration, the *task owner* is responsible for the track re-assignment.

Prototype implementation
We implemented a prototype for the 3D story teller application using Android Samsung Galaxy note N7000 phones to play the leader/owner, and member roles. Further, we generated an audio track of 2 characters, and used Pure Data (PD) to control the tracks through Android APIs [1].

The scenario starts with a smartphone playing the full audio track. Upon the arrival of a collaborator, a version of the file is disseminated to it with the character to play and timing information. The audio then plays from both phone till one of the phones quit or the audio track is over.

Performance Measures
The execution of the implemented prototype was monitored used Monsoon power monitor for both phones. Fig. 4a shows the effect on the communication module on the smartphone power consumption from 0 to 40 seconds. Figures 4b, c shows the execution power consumptions for phones A, and B respectively. The audio file started playing at second 40 and is terminated at 50 seconds, the figures monitors the power consumption 10 seconds after the end of execution. It is clear from Fig. 4 that *ColPhone* has a mild effect on the smartphone power consumption.

Conclusion and Future Work

In this work, we proposed an initial attempt to build an infrastructureless smartphone collaboration system that we called *ColPhone* . We presented the system design roles and modules. We explained how both roles and modules interact together to achieve a smooth collaboration process. Finally, we implemented a prototype for the 3D story teller application using *ColPhone* , and showed its energy consumption.

Our Future work will include expanding the framework design and implementation to host other applications. In this Demonstration we show an I/O bound application (3D story teller), we further plan to measure *ColPhone* 's performance while hosting other application categories. Further we will be using a hybrid of Bluetooth, and WiFi-P2P for power efficient connection phase, and higher data rate in service dissemination respectively. This work assumes a secure and privacy preserving communication between phones, our future work will consider relaxing this assumption.

References

[1] http://puredata.info/.

[2] Ardalan Amiri Sani, Kevin Boos, M. H. Y., and Zhong, L. Rio: A system solution for sharing i/o between mobile systems. In *Proc. MobiSys*, ACM (2014).

[3] Duan, L., Kubo, T., Sugiyama, K., Huang, J., Hasegawa, T., and Walrand, J. Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing. In *INFOCOM, 2012 Proceedings IEEE*, IEEE (2012), 1701–1709.

[4] Fisher, E. R., Badam, S. K., and Elmqvist, N. Designing peer-to-peer distributed user interfaces: Case studies on building distributed applications. *International Journal of Human-Computer Studies 72*, 1 (2014), 100–110.

[5] Lee, Y., Ju, Y., Min, C., Kang, S., Hwang, I., and Song, J. Comon: cooperative ambience monitoring platform with continuity and benefit awareness. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ACM (2012), 43–56.

[6] Salem, A., and Nadeem, T. Colphone: A smartphone is just a piece of the puzzle. In *Proceedings of the 20th annual international conference on Mobile computing & networking*, ACM (2014).