Computer Communications 37 (2014) 25–39

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

# A context-rich and extensible framework for spontaneous smartphone networking

Elmurod Talipov, Jianxiong Yin, Yohan Chon, Hojung Cha *

*Department of Computer Science, Yonsei University, Seodaemun-gu, Shinchon-dong 134, Seoul 120-749, Republic of Korea*

ARTICLE INFO

*Article history:*
Received 24 December 2012
Received in revised form 30 September 2013
Accepted 13 October 2013
Available online 24 October 2013

*Keywords:*
Spontaneous networking
Context learning
Delay-tolerant applications
Content sharing

ABSTRACT

This paper presents the design principles, implementation, and evaluation of SPONET, a framework that has been specifically developed for spontaneous networking among smartphone users. SPONET has four distinct objectives, providing (1) a rich context for location-aware networking, (2) robust cognitive networking, (3) extensibility with various routing protocols, and (4) a convenient programming interface for delay-tolerant applications. The key technical challenges are, therefore, unsupervised place learning, network construction without user intervention, and a networking policy with low complexity. We have designed a place-learning algorithm using the properties of scanned Wi-Fi access points to identify meaningful places. SPONET provides dynamic neighbor discovery and data exchange mechanisms for autonomous networking. We have implemented SPONET on Android-based, off-the-shelf smartphones without any adaptation of their networking architecture. Experimental results show that the proposed system is indeed acceptable as a framework for various delay-tolerant applications in smartphones.



## 1. Introduction

With the current proliferation of mobile devices, spontaneous interaction between colocated devices that do not acknowledge each other *a priori* will become commonplace. Among mobile devices, smartphones are rapidly growing in number, and the administration effort of handling the traffic generated by smartphones is also increasing.

At the same time, cellular network operators are deploying more and more Wi-Fi access points (APs) to offload traffic from cellular networks to wired networks. However, the operators are well aware that this is not a permanent solution, and alternative solutions are being sought. One approach lies in using spontaneous opportunistic networking. Unlike traditional ad hoc networks, spontaneous networking is an application-oriented approach with the following advantages: (1) It overcomes the limitations of communication services built upon the widely used concept of end-to-end connectivity, and (2) it works in a plug-and-play manner, minimizing the effort required to integrate devices and services into network environments. In the literature [1–3], spontaneous networks have been referred to as pocket-switched networks (PSNs), delay- or disruption-tolerant networks (DTNs), and opportunistic networks (ONs), to name a few. In this paper, we refer to these networks as spontaneous smartphone networks (SSNs) in order to elaborate target networks. SSNs specifically assume the existence of unique constraints, such as the availability of localization techniques, motion sensors, various radio interfaces, and regularities in the movements of smartphone users.

The opportunistic network research community believes that many applications would benefit from the implementation of opportunistic network frameworks. Lindgren et al. [4] have presented a number of such applications, such as telemedicine for developing regions that would allow doctors to remotely give correct diagnoses and prescribe treatment [5], unrestricted communication in the presence of oppressive governments [6], file sharing and bulk data transfer, and automatically synchronizing mobile and static devices. In particular, bulk data sharing and automatic content synchronizing benefit from the implementation of SSNs as smartphone users generate massive amounts of content each a day and intend to share and synchronize it with others or with other personal digital devices.

Efforts in the field of designing and implementing SSN-like networks are generally classified into two groups: special purpose implementations [7–11] and generic implementations [12–15]. While generic implementations can be used by various applications on many different devices, special purpose implementations are targeted to specific applications or devices. Although some of these implementations [7,9,11,12] have shown good performance in terms of data delivery, the systems are neither efficient in networking nor instantly usable on smartphones. This is because these systems do not satisfy at least one of the requirements [4] listed below.

* Corresponding author. Tel.: +82 2 2123 5711; fax: +82 2 365 2579.
*E-mail address:* hjcha@yonsei.ac.kr (H. Cha).

0140-3664/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2013.10.004

First, the system should be context aware, meaning that human dynamics should be considered and networking tasks should be conducted intelligently. In other words, the places where users often visit should be identified, and user mobility needs to be predicted to reduce communication overhead. Second, implementation must overcome technological constraints and should be instantly usable on smartphones. The current level of technology poses a number of constraints in terms of ad hoc networking. Such constraints are the connection-oriented nature of Bluetooth radio, the Wi-Fi ad hoc mode not being supported, the limited battery capacity of mobile devices, and the diversity of mobile devices. For instance, devices that use Bluetooth radio should be pre-paired. Although the Wi-Fi ad hoc mode is supported by many smartphones, the functionality is normally not enabled due to the power issue. But despite the existence of these limitations, previous implementations have required Wi-Fi ad hoc mode support or used Bluetooth communication. Third, extensibility and multiple application support must be available. The system should support various routing mechanisms and message management methods, and it should provide rich APIs to enable diverse applications. Finally, users are always concerned about their battery life and are often concerned about privacy issues regarding their locations and social contexts. Hence, the system ought to be energy and resource efficient and must protect the privacy of users.

In this paper, we present the design principles and implementation of a context-rich, flexible spontaneous networking framework called SPONET. The design challenges of SPONET have been unsupervised place learning and an autonomous networking policy with low complexity. SPONET overcomes the aforementioned challenges and provides the following contributions:

- SPONET autonomously identifies and remembers users' mobility behavior and social relationships for context-aware networking.
- A spontaneous networking mechanism analogous to Wi-Fi Direct [16] has been designed to quickly discover neighboring devices and exchange bulk data. The key advantage is that the scheme does not require modification to the software architecture of mobile devices. Also, user intervention is avoided in device pairing.
- SPONET is extensible with various routing mechanisms and supports diverse DTN applications.

The remainder of the paper is organized as follows. Section 3 presents a brief overview of the system. Section 4 describes the context-generation method, and Section 5 covers the details of spontaneous networking and integration into the proposed system. In Section 6, we discuss implementation challenges, programming interfaces, and case study application. System evaluation in terms of complexity, context-generation accuracy, and networking efficiency is presented in Section 7. We discuss relevant work and limitations of our system in Section 8, and conclude the paper in Section 8

## 2. The SPONET architecture

SPONET is a context-aware, spontaneous networking framework. Here, *context-awareness* means the system can generate various types of user information and devices states, and *spontaneous networking* means an autonomous peer-to-peer communication and data delivery. Hence, SPONET can largely be divided into two functionalities: context generation and spontaneous networking. The system is designed to provide a context-rich, efficient networking environment for delay-tolerant, peer-to-peer applications. We have implemented SPONET on Android phones using the available programming interfaces and built-in sensors such as an accelerometer, electronic compass, GPS, Wi-Fi, and GSM. Fig. 1 shows the overview of the SPONET system, which consists of eight key components: the activity manager, landmark manager, context manager, link-state manager (LSM), neighbor manager (NM), message manager (MM), routing protocol adaptor (RPA), and context provider.

We have assumed that bulk data are usually exchanged in places where smartphone users stay for a significant amount of time. Consequently, the system tracks a user's movements and identifies places where the user stays for a significant duration. The activity manager monitors user activity and schedules other system tasks. The component continuously checks the variance of the accelerometer signal and the phone state to identify user activity, which is classified as either "moving" or "stationary.".

The landmark manager discovers and learns landmark information, which is comprised of both physical and logical location information. The landmark in short is a place where user stays longer than a certain duration. The component obtains the physical location information from the GPS provider and the network provider. Each provider generates location information along with an error bound that reflects the accuracy of the location estimation. The providers are readily implemented into the Android protocol stack and can be invoked using special APIs. The logical provider determines the logical identification of a landmark and decides if the current location is a landmark that the user has previously visited.

In parallel to context generation, the LSM discovers neighboring smartphones and creates a spontaneous network. The NM collects neighbor information and neighbor encounter information and hands it over to the context manager. The NM also monitors active neighbors and assists routing protocols for data exchange.

The MM and RPA are the core components for spontaneous networking. The MM receives application data and schedules it for delivery to the destination. Additionally, the messages processed by the RPA are handed to the MM and delivered to applications. The RPA contains a number of prebuilt DTN routing protocols, and based on current routing protocols, this component makes the decisions regarding message forwarding. Additionally, the RPA provides an interface for adding new routing protocols.

Finally, the context provider is responsible for managing storage, such as aggregating raw data into refined data and removing expired message information. This component also manages the database to provide data to the internal user interface and other applications. When a stationary activity is detected, the system runs two tasks. First, the context manager generates the user context from the landmark manager and activity manager. Then, the LSM initiates neighbor discovery and creates a spontaneous network. The NM collects neighbor information and posts it to the context manager as additional context. The system immediately deactivates each sensor after obtaining the necessary context information. The relationship between a user's activity and a set of components is predefined to minimize energy consumption. Fig. 2 illustrates the activity-based decision flow used to dynamically provide context-aware networking. In the following section, we will describe how each component is used to construct a complete spontaneous networking system.

## 3. Context generation

Context generation is the process of collecting and updating the user's context. We formally define context as processed data on a landmark, user activity at the landmark, neighbors, and neighbor encounters associated with the landmark. Generated context is stored in the database via the context provider. We are only interested in collecting context when a user is stationary. Thus, we