An Android Framework for Opportunistic Wireless Mesh Networking

Andre Ippisch
Technology of Social Networks, University of Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf, Germany
Email: andre.ippisch@uni-duesseldorf.de

Kalman Graffi
Technology of Social Networks, University of Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf, Germany
Email: graffi@cs.uni-duesseldorf.de

***Abstract*—Smartphones are more and more widespread in the society. They are powerful devices with huge storage space and high bandwidth communication capabilities. However, several use cases cannot be supported nowadays, especially when it comes to the exchange of large files between geographically close participants. Currently, smartphones only communicate through services in the Internet or through Bluetooth, both are limited in their bandwidth and scale. The exchange of large files of hundreds of MB between dozens of users is very time consuming and thus nearly impossible. We present an Android Framework and Application Toolkit for wireless, opportunistic mesh networking in combination with a peer-to-peer-based data management functionality. The smartphones create the wireless opportunistic mesh network without user interaction and negotiate the files to exchange, the nodes to contact and the transmission schedule. Thus, huge files can be exchanged through (multi-hop) wireless local communication between dozens of smartphones. This we demonstrate with several devices and also invite the participants to join the demonstration with their Android devices.**

I. Introduction

Localized communication including the exchange of files and messages is a casual application for mobile devices. Having powerful smartphones, with sufficient bandwidth, storage space and computational power, also the exchange of larger files, such as movies, music or working documents, is a desired task. A functionality should support the exchange of large files between dozens of geographically close users.

Currently, however, work groups and collocated friends either have the option to use cellular communication to upload their files to email servers or the cloud and to download them subsequently, to use 1-to-1 communication means such as Bluetooth and USB sticks or to manually enable a virtual hotspot and open a one-to-many connection and an FTP server. The exchange of large files of hundreds of MB between dozens of users is very time consuming and thus nearly impossible. Additionally, when using cellular communication, the Internet connection can fail due to bad connectivity, such as in massively built buildings or rural areas, as well as on purpose, as it was the case at the Hong Kong protests in 2014. A system that only uses local, ad-hoc Wi-Fi connections can avoid mentioned problems and work under extreme conditions. Users that are geographically close can easily exchange large files like scripts and foils in the university, working documents in a meeting or movie clips and pictures at a party.

Our contribution to that problem is a framework for wireless opportunistic mesh networks for the Android operating system that allows one-to-many (1-to-n) and multi-hop connections over Wi-Fi without user interaction as well as a peer-to-peer (P2P) data plane on top, which allows to signal availability and interest in specific files and that initiates, schedules and organizes the data transfer between the connected nodes.

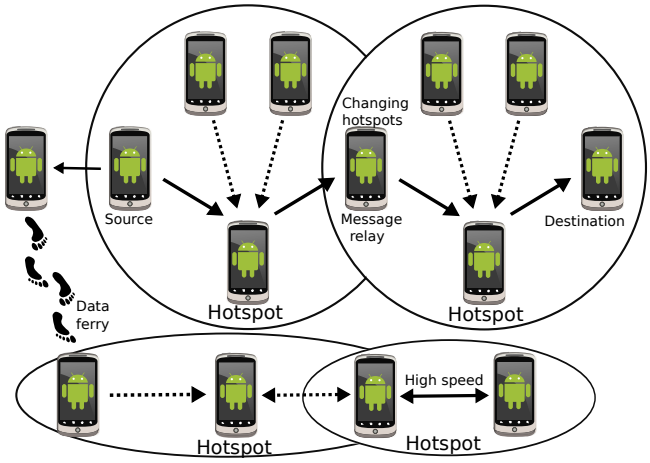

Fig. 1. Characteristics: 1-to-n wireless hotspots with message relaying, mobile data ferries, opportunistic networking and high speed direct transmissions.

In Section II, we present the software architecture for our Android-based framework for P2P-data exchange in wireless mesh networks. In Section III, we sketch how our software tool will be presented at the conference using various devices.

II. Android Framework for Wireless P2P Networks

The desired functionality of wireless connectivity without user interaction and P2P-based applications such as file synchronization and multi-hop chatting is implemented in a two-layered software architecture. We highlight the following properties of our solution: 1.) *Android* as the most common smartphone operating system is supported. 2.) *Connectivity* for an unlimited number of smartphones is supported. Please note, that while our protocols scale well, the physical layer does not support to many network participants. 3.) *No user interaction needed* for creating the wireless mesh network. 4.) *Multi application support* allowing to run various different applications, such as file synchronization or chat, aside.

Our software solution is split in two parts: the wireless opportunistic mesh networking framework, which establishes

and coordinates the connection to other nodes, and the P2P-based application toolkit, which allows to build various applications on top with direct or multi-hop-based device-to-device interaction. To this point the Android operating system has no possibility of using peer-to-peer connections between Android devices without user interaction. The functionality called Wi-Fi Direct is available in the settings of Android. Various devices can be connected using this technology, however, it does not reliably scale to more than a few devices. Also, it requires to actively invite every new user to the Wi-Fi Direct network, which is unacceptable for large networks. Our framework uses Android's API for Wi-Fi directly in order to establish and manage connections to other Android devices without user interaction. It offers an interface for user applications which can use the framework to exchange data. The user applications can specify preferences for following connections so that the framework chooses the best device for connections with respect to the preferences and other criteria like signal strength and motion profiles. The framework offers a basis for data exchange in wireless opportunistic mesh networks on which user applications can build on. Next, we present the framework and consecutively the applications.

A. Framework for Wireless Mesh Networking

The framework is written as an application for the Android operating system. It runs as a background service and therefor does not disrupt the users normal use of this phone. The main part is the network service that manages all connections between devices. For that three submodules are used: the *hotspot manager* establishing short-time individual 1-to-n communication hotspots, the *session manager* which maintains the information between the short-time hotspot connections, as well as the *messaging manager* which provides end-to-end connectivity between applications on different nodes.

1) Hotspot Manager: The framework establishes a IEEE 802.11 Wi-Fi hotspot (or joins existing ones) to connect to other nodes. The SSID of this hotspot is deterministically chosen and thus pre-known. Other wireless technologies like Bluetooth and Wi-Fi Direct require the interaction with the user and thus do not comply with our requirements. The framework application scans its surroundings for Wi-Fi hotspots which are created by other devices. If there are no devices (or hotspots) nearby the framework itself creates a hotspot for other devices to connect to if they come in range. The service provides necessary information like IP and MAC addresses or reception strength that allows the user applications to react to the given connection appropriately. Since users should be able to be connected to the Internet nonetheless the framework offers phases without hotspot activity to allow all applications on the user's smartphone to work without limitations.

2) Session Manager: In addition to the single hotspot usage, nodes apply an election mechanism that promotes the creation of new hotspots by nodes that are not interested in communicating within the existing hotspots network. By this, collocated multi-hop relaying of messages is enabled. Please note the 1-to-n communication in hotspots, all traffic involves the hotspot host. Thus only signaling traffic and traffic directed from/to the hotspot host is efficient. In order to coordinate the information about the nodes, corresponding hotspots, the hotspot timings and general information about which messages are to be delivered, the *session manager* is used.

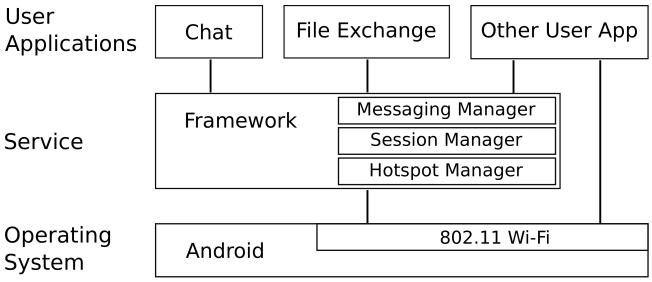

Fig. 2. Software Architecture

3) Messaging Manager: During the membership in various hotspots, nodes learn about other nodes and their interest to send messages between each other. Similar to routing in opportunistic networks, messages can be relayed from one application on a device to the destination node and its application. In summary, the framework module offers reliable network establishment with the option to decide to which nodes to connect, as well as a message passing interface.

B. P2P-based Applications: File Synchronization and Chat

User applications can use the framework's existing links for data exchange or decide which (known) nodes to connect to. We implemented two user applications.

The file exchange application uses IHAVE and IWANT lists to tell other devices which files are available and desired on a device. The application calculates exchanges not only for the current connection but also for connections that could be established with other devices and gives preferences to the framework for those future connections. The files are chunked, transferred with high bandwidth within the same hotspot and validated based on hashes of the chunk and complete file.

The chat application can multicast messages in the local area with the option to use publish/subscribe-based chat rooms. Thus, without Internet connectivity and without being able to be traced, local communication is enabled.

In both cases the user decides if he wants to be informed immediately about new files or chat messages by the application or not. With applications like these, file exchange in a office meeting or information exchange at a fair is easier and more robust than over cellular networks. The transmission speed for local file transfer reaches 14.4 Mbit/s, which outperforms cellular communication. The framework supports Android version 4.1 (Jelly Bean) or higher which makes a total of 78% of all Android devices (as of September 2014). As the user applications are independent of the framework, new apps using this functionality can be created easily.

III. Demonstration Setup

We demonstrate the chat program and the file exchange application on several Android smartphones and tablets with pre-installed framework and user applications. All visitors can participate in the demonstration with their Android devices by installing the framework and additional apps. The demonstration shows how data is shared among the devices based on data offerings and preference settings and that a location-based chat can be done within a multi-hop neighborhood.