Bartłomiej GAJEWSKI, Tomasz MARTYN
WARSAW UNIVERSITY OF TECHNOLOGY, INSTITUTE OF COMPUTER SCIENCE
15/19 Nowowiejska St., 00-665 Warsaw, Poland

Smart mobile P2P communication optimization for close range by an automatic interface switch

Abstract

The efficiency of peer-to-peer data connection for mobile applications is still an issue, and an essential problem for their users. The newest 4G LTE networks, though offering high throughput, still do not allow a direct peer-to-peer communication. Moreover, the users of cellular networks are still charged for each megabyte of data transfer. Much better connection parameters and smaller latencies are ordered by Wi-Fi networks. However, setting up hotspot Wi-Fi networks still requires some effort, as it must be done manually in most cases. In this paper, we propose a solution based on a technique of automatic switching between a Wi-Fi hotspot and a cellular network. Our method allows one to minimize latency and maximize throughput. It also lowers the time and effort needed to establish a Wi-Fi connection.

Keywords: Peer-to-Peer Wi-Fi, minimize latency, mobile communication, off-range connectivity.

1. Introduction

The following common problem of mobile smartphones users is observed. Whenever two or more users meet in one place and they wish to quickly exchange large files (like photos, movies, music or others), they need to establish a connection by Wi-Fi or pair the devices using Bluetooth, and use appropriate software. This usually takes time and effort, and is often regarded as difficult by users with low technical knowledge.

Alternatively, the users can use centralized exchange servers. Such an approach requires an internet connection with an acceptable transfer speed. Moreover all the users will be charged for data transfer, and the exchanged data packages are very often large and, thus, expensive in terms of 1 MB price. One should note that a typical phone-recorded movie of 5 minutes can be up to 100 MB large, while the average monthly transfer limit is at 500 MB. About 60% of mobile phone subscribers still have no transfer included in their subscription fee, and have to pay up to 0.60$ for each MB of data [7].

Another problematic situation occurs when a few users would like to play a multiplayer game using their mobile devices. First of all, the users face a discovery problem (which game should they join, who to play with) even though they might occupy the same room or vehicle.

One more problem arises when the users are in no-range zone of a cellular network or they are located in a moving vehicle - connection to the internet based service might be unsatisfactory or even impossible. Moreover, even if the connection through a cellular network is possible, it will often provide high connection latency unacceptable for real-time applications (e.g. multiplayer games).

In this paper, we propose a solution to the mentioned problem of mobile devices connectivity. Our method combines the advantages of cellular and Wi-Fi networks using smart automatic switching between one another when needed, and, thus, trying to minimize latency and to maximize throughput.

2. Existing solutions and related works

There are various methods of communication interfaces available on each mobile device, including different generations of cellular networks, Wi-Fi networking and hotspotting, Bluetooth communication, and NFC interface. Establishing Wi-Fi communication networks and improving mobile communication throughput and minimizing latency has been studied (and still is) extensively. There are a few prominent techniques which are summarized below.

- **Mobile p2p communication by hole punching** makes it possible to establish a direct connection with the omission of a server in the middle for two devices behind NAT of the users' internet providers, which results in significant performance improvement. However, as shown in [1], Hole Punching cannot be easily obtained, and for some internet providers it might be blocked.
- **Wi-Fi Direct [8]** is a popular standard for interconnecting devices with a Wi-Fi interface. It is widely used for establishing connections between mobile devices and stationary devices, like printers, scanners, routers, etc. Different methods of devices pairing are shipped that may require the proximity of a near field communication, a Bluetooth signal, or a button press on one or all the devices. The main purpose of Wi-Fi Direct is to simplify the connection establishment for users with no technical background. Thanks to that, there is no need for the users to choose network IDs nor credentials. Wi-Fi Direct is supported by many software and hardware vendors, regardless of the platform. Unfortunately, Wi-Fi Direct does not support interface switching and long-distance communication.
- **Wi-Fi ad-hoc networks** utilize a concept of creating larger Wi-Fi networks. The purpose is to either offload cellular network [2, 3] or work as emergency networks when the cellular network is not available [5, 6]. Those solutions often must be supported by a mobile operator's infrastructure or even require an interference in the mobile device hardware. Since the maximal communication distance depends on a number of peers, Wi-Fi Ad-hoc solutions do not support interface switching and long-distance communication.

3. The proposed solution

The solution proposed in this paper is based on combining cellular and Wi-Fi networks. A cellular network is used for long-range communication, and it is assumed that its availability is equivalent to the ability to connect to the Internet. For close range communication or in the case of the lack of the Internet connection, Wi-Fi interface is used. The choice of an interface as well as a method of switching between interfaces can be made by the clients applications. We propose a rendezvous server to enhance the decision process and to allow long range communication.

To propose a solid solution, we focused on the following aspects:

- the users might lack of or can lose connection to the Internet;
- at least one of the devices supports a Wi-Fi hotspot setup;
- the users might be in a close range, preferably not exceeding 100 meters;
- the users must declare a wish to communicate between each other;
- the users require the smallest possible communication latency and maximal throughput.

3.1. Communication via cellular networks

Mobile network operators do not allow direct communication of peers yet (until IPv6 is introduced to mobile). Neither workarounds like Hole Punching do not guarantee successful direct connection [1]. Therefore a server with public IP that is available worldwide is necessary In this paper, we refer to the server as the Rendezvous Server. By default, the users try to

connect to the Rendezvous Server and can communicate through it. Communication via server forwarding is possible only if clients have connection to the Internet. The efficiency of the server is not in the scope of this paper but is discussed for example in [4].

3.2. Communication via Wi-Fi networks

Alternatively, clients can connect using more efficient Wi-Fi communication, as far as this is possible due to the limited distance imposed on Wi-Fi users. Two ways of establishing Wi-Fi communication are possible in our solution: the usage of an available Wi-Fi router or automatic Wi-Fi network creation (hotspotting).

3.3. Decision elements

The communication method is chosen by means of two complementary algorithms:

3.3.1. Rendezvous server

While users communicate via server, they also provide information about their location and visible Wi-Fi networks. The server calculates the geo-graphical distance between clients using the spatial hashing cell algorithm for optimization [1].

If the distance is in an acceptable range, the server decides that the clients A and B should start to communicate directly. It chooses which client should start the hotspot and which should join it when it is available. During the process a communication through Rendezvous Server is still possible.

The server can also decide whether the clients are in the range of the same Wi-Fi network and can access the network , according to the data they provide. If so, the clients receive an instruction saying that they should use a specific Wi-Fi network. It is even possible that a client could share a hashed password to a secured network with another client, however the password hashing method is platform-dependent and is not shared in public. Therefore, in such a case, the success of connection to a secured network is not guaranteed.

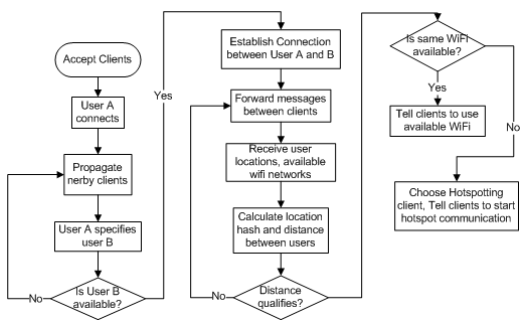


Fig. 1. Rendezvous Server decision process

3.3.2. Client application

In the first step, the client application should try to access the Rendezvous Server. If it is available, the client provides its geographical location together with visible Wi-Fi networks. In return, the server sends a list of proposed nearby clients. The user should then specify the client he wishes to communicate with (either from currently available users or not). When a present user is chosen, a server based connection is established between the clients to allow the fastest possible connection start. Then, the Rendezvous Server can assist in switching to a more efficient Wi-Fi communication.

If one or more of the users have neither internet connection nor they are connected to a Wi-Fi network already, the clients have to decide by themselves to set up the network.

When one of the users makes decision he wish to start a connection, and no networks are visible, the application will automatically start the hotspot with a predefined Wi-Fi name and a password. When another client in a close range also decides to start a connection, and a network is visible, the application will automatically connect to the network and communication can be established.

A problem occurs when two users start a hotspot simultaneously, since many devices do not allow hotspotting and checking for available networks at the same time. In this case the clients would not be able to determine if they are within the range or not. If the client device enables checking the available networks while hotspotting, the network started with a lower ID will be chosen. If the device does not allow this, a manual refresh has to be done by one of the clients to resolve the problem – this means that the hotspot has to be turned off and the networks have to be checked.

A special case is resolved when clients already share the same Wi-Fi network (router or mobile hotspot). Therefore the client applications can also decide by themselves to use it, without Server mediation.

In this case, a message is sent through a Wi-Fi network, using broadcasting IP address. If only the network supports broadcasting, the clients can discover themselves and direct connection can be established.

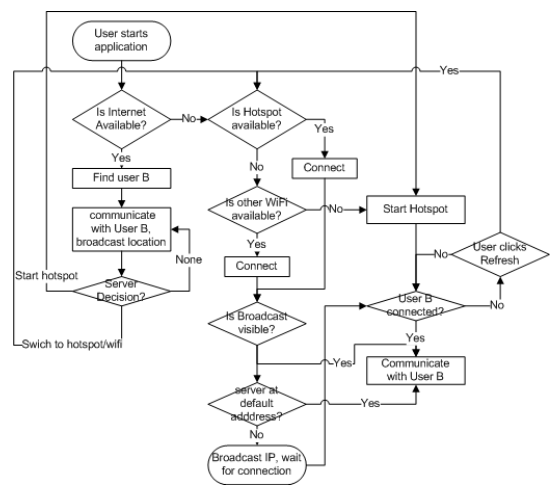


Fig. 2. Client application algorithm

4. Implementation

A test software was implemented for Android devices using generic framework. Nevertheless, the solution is suitable for devices with other operating system. Currently there are no frameworks available that would allow creating multiplatform communication middleware for Android, iOS and Windows Mobile.

One of the main implementation problems is that the Wi-Fi operations are not available in the Android framework before API v14 (Android 4.0), that is, when Wi-Fi management service was introduced [9]. However, before that, methods can be extracted that makes it possible to operate the built-in Wi-Fi interface.

The test application allows one to establish a connection between two clients using the proposed solution. The test data was sent and the average message delivery times were calculated, together with other statistics, like the time needed to establish connection.

5. Performance testing

In order to determine the performance of the solution, a few tests were performed. The tests were made using the following Android devices:

- LG L5 (Android v 4.4)
- HTC Desire Z (Android v 2.3.3)
- Samsung Galaxy S3 (Android v 4.0)

The test application facilitates connections by Wi-Fi, 3G and 4G networks. The internet service was provided by Plus (Polkomtel Sp. z o.o.). During performance tests, 100 000 packets containing 1Kb of random data with a timestamp were sent between clients both ways. The results are provided in Table 1. The tests were conducted with the following scenarios:

- **Connection via Rendezvous Server** - In this scenario the both users were able to use 3G or 4G network and access the Rendezvous Server.
- **Direct connection via Wi-Fi Hotspot or router** - The server determined that users should switch connection to Wi-Fi.
- **Connection via Wi-Fi route** - In this scenario two users, both connected to same Wi-Fi network, simultaneously started the test application.
- **Independent connection establishment** - Two users in a close distance (10-20 meters) started the test application simultaneously or with delay of 5 seconds.
- **Switching distance** - Different acceptable switching distances were provided to the Rendezvous Server algorithm so as to determine the satisfactory success rate of interface switching. The tests were made in open, suburban space, 10 times for each distance. The results are presented in Table 2.

Tab. 1. Performance tests results

Via:	Effective speed	Latency	Time to connect
3G (HSDPA)	233 Kb/s	317 ms	1-5 s
4G (LTE)	562 Kb/s	228 ms	1-3 s
Wi-Fi router	832 Kb/s	30 ms	1-2 s
Wi-Fi P2P	795 Kb/s	28 ms	3-5 s (with server) 10-15 s (simultaneous start) 3-10s (one-by-one) 15-300s (handmade setup)

Tab. 2. Switching distance tests results

Distance:	20 m	50 m	100 m	150 m	200 m
Success rate:	100%	100%	90%	70%	10%

6. Limitations

The proposed solution has few limitations:

- **Compatibility** – The tests revealed that some devices can have issues with Wi-Fi network broadcasting (observed on HTC Desire Z, Android version 2.3.3). Due to the limited number of devices available for tests, it is yet unknown how the solution works with a wider range of devices.
- **Range** – the solution provides an improvement in communication parameters only if both devices are in the range of Wi-Fi communication. This can vary from 50 up to 300 meters, depending on the environment and strength of device interfaces.
- **Switching time** – It was determined that the time of switching connection can vary from 5 to 20 seconds, depending on the device. The time of switching is mostly limited by the time of starting up the Wi-Fi onboard interface, setting up the hotspot and/or accepting the connection. The connection time to the server as well as the server load can also be an issue. It was tested that the computing times have little effect on the switching time.
- **Security** - If the algorithm of hashing the Wi-Fi SSID would be revealed, the users could be easily exposed to connections from unauthorized users, who could use their internet connection or intercept value data. Therefore the hashing algorithm is not described in this paper.

7. Conclusions and future work

The tests results show that there is a great potential in automatic interface switching and automatic Wi-Fi setting up. The throughput is almost doubled and the latency lowered almost 10 times, as long as the users are in a distance not exceeding 100 meters (e.g., in the same room, preferably). The automatic Wi-Fi setup puts aside –manual setup, since the latter can take a few minutes in comparison to 15 seconds of the automatic one.

Although security might still be an issue, it is up to the users and developers if they decide to use the solution. A suitable middleware is planned to be released both for servers and smartphones to speed up the application development process. The Hole Punching method can be integrated to minimize latency when the users are in larger distances. Finally, integration with Wi-Fi ad-hoc network creation remains an open problem.

8. References

[1] Satish Narayana Srirama, Mohan Liyanage: TCP Hole Punching Approach to Address Devices in Mobile Networks. Future Internet of Things and Cloud (FiCloud), 90-97. Barcelona, 2014.
[2] LU Xiaofeng, HUI Pan, Pietro Lio: Offloading Mobile Data from Cellular Networks Through Peer-to-Peer WiFi Communication: A Subscribe-and-Send Architecture. In: China Communications, p. 36-46, 2013.
[3] Savio Dimatteo, Pan Hui, Bo Han, Victor O.K. Li: Cellular Traffic Offloading through WiFi Networks. The IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS 2011). In Proceedings of the 8th MASS, 2011, p. 192--201, Valencia, 2011.
[4] Greg E. Blonder: Systems and methods for providing increased server performance, in a communications network. Patent US5946299, 1999.
[5] Sacha Trifunovic, Bernhard Distl, Dominik Schatzmann, Franck Legendre: WiFi-Opp: ad-hoc-less opportunistic networking CHANTS '11 Proceedings of the 6th ACM workshop on Challenged networks, 37-42, New York, 2011.
[6] Timothy X Brown, Brian Argrow, Cory Dixon, Sheetalkumar Doshi, Roshan-George Thekkekunnel, Daniel Henkel: Ad Hoc UAV Ground Network, "Unmanned Ulimited", Chicago, 2004.
[7] Rhett Allain: How much does your data cost. Wired.com, 2011, http://www.wired.com/2011/06/how-much-does-your-data-cost
[8] Wi-Fi Direct homepage, http://www.Wi-Fi.org/discover-Wi-Fi/Wi-Fi-direct
[9] Android Developers API Guides, http://developer.android.com/guide/topics/connectivity/wifip2p.html

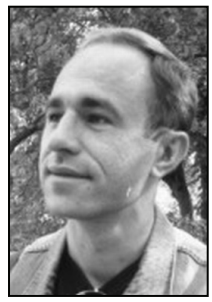
Received: 21.04.2015 *Paper reviewed* *Accepted: 02.06.2015*

Bartłomiej GAJEWSKI, MSc

Ph.D. student at Warsaw University of Technology. Specializes in mobile communication technologies, geolocalization and analysis of geographical data. Generation of location-based data and content is his main area of research.

e-mail: b.gajewski@ii.pw.edu.pl

Tomasz MARTYN, PhD, DSc



Assistant Professor at Institute of Computer Science, Warsaw University of Technology. Author and coauthor of 4 books as well as many research papers and articles published in various peer-reviewed international journals and conferences. His research interests include fractal geometry, scientific visualization and real-time rendering techniques.

e-mail: martyn@ii.pw.edu.pl