Evaluation of a Car2Car communication system on different platforms using Wifi Direct as communication protocol

Roland Leher, Florian Grill und Mario Murrent

SEMINARARBEIT

Software Architecture and Design

in Wiener Neustadt

Contents

Abstract

Car2Car (C2C) also known as Vehicle2Vehicle (V2V) Communication is a communication technology that allows vehicles, in the future road signs and other traffic related things to exchange information. This technology was first developed and successfully demonstrated by General Motors in 2005[1].
Since C2C communication is a fairly new and unexplored topic, only a few internet sources currently offer relevant information about this topic. A standard does not exists yet.
The aim of this work is to provide a basic understanding of the principles of C2C communication and to implement three prototypes for Windows Phone and Android based mobile phones and a raspberry pi. Furthermore this paper should demonstrate a cheap way for upgrading an ordinary car with a C2C communication system and illustrate the possibilities which customer products are offered for realizing such a system.

[1]http://www.worldcarfans.com/10510278356/general-motors-develops-vehicle-to-vehicle-communication

Abbreviations

API	Application Programming Interface
C2C	Car2Car
GPS	Global Positioning System
P2P	Peer-to-Peer
SDK	Software Development Kit
V2V	Vehicle to Vehicle

Chapter 1

Car2Car Communication

C2C describes the communication between vehicles and other infrastructure. The goal is to improve the safety on the streets and to inform road users about upcoming problems on the road immediately including different car manufacturers and roadside units. Furthermore the C2C Communication technology should be a basis for decentralized active safety applications and therefore reduce accidents and their severity. Besides active safety functions, it includes active traffic management applications and helps to improve traffic flow.

1.1 Actors

A C2C System consists of three important actors: the driver, the road operator and hotspot and internetproviders. The driver receives road information and warning messages or route recommendations.
The road operator receives road information from cars or other infrastructure and therefore will improve the control of the traffic in a more efficient way.
Hotspot and internet providers can install their communication systems for example at gas stations.[1]

1.2 Car 2 Car Communication Safety Scenarios

In this section the safety aspect in public traffic will be explained. C2C Systems help to increase the safety on the streets and therefore concepts like Cooperative forward collision warning, Pre-crash Sensing/Warning or Hazardous Location Notification will be applied.

Cooperative forward collision warning:
This scenario should avoid rear-end collisions, for example if a following vehicle suddenly brakes. The vehicles share information about speed,

position and heading. To avoid collisions, the system has to use the own vehicle information and the information of vehicles nearby. If the system detects a critical proximity, it will warn the driver.[1]

Pre-crash Sensing/Warning:
If a crash is unavoidable, information will be provided about vehicle size and exact position. Crash involved vehicles will exchange data about predicted impact zones, therefore airbags or bumper systems will be informed, where the impact takes place.[1]

Hazardous Location Notification:
The vehicle will inform about hazardous road conditions. If, for example, the ESP (Electronic Stability Program) is activated, the location and road condition will be transmitted to nearby vehicles. This information could be used for optimizing the chassis of the vehicle if it reaches the hazardous location. Such information is not limited to vehicles. Road signs could provide information as so called road side units, which will be served by external service providers.[1]

1.3 Car 2 Car Communication Traffic Efficiency

In the future C2C Systems should improve traffic efficiency. If all cars are connected together, traffic jams or the chance of accidents will be minimized. The cars communicate with each other and are able to intervene actively in traffic situations. Concepts like Enhanced route guidance and navigation, Green light optimal speed advisory or Merging Assistance are introduced to accomplish this goal.

Enhanced route guidance and navigation:
Every car should have internet access, which will be used for enhanced route guidance and navigation. For example, if no vehicle or roadside unit is ahead, road information can be provided by the internet connection. Because of the navigation system the car knows exactly where it will go. With this information data about the route can be downloaded an displayed to the driver.[1]

Green light optimal speed advisory:
This Scenario should help the driver to make their driving smoother and avoid stopping. The information will be provided by signal intersections. The timing (when turns the light green) and exact location of the intersection will be transmitted. With this information, the vehicle calculates an optimal vehicle speed using the distance from the vehicle to the intersection and the time when the signal is green. The vehicle notifies the driver of the optimal speed. It is the goal to increase traffic flow and to increase fuel economy.[1]

Merging Assistance:
If the vehicle wants to merge into traffic on a roadway, nearby vehicles will be informed about the approaching vehicle. The vehicle itself receives information about the current behavior of nearby vehicles. The assistance will guarantee that the vehicle can enter the traffic flow without major disruptions to the flow.[1]

1.4 Car 2 Car Communication Infotainment and Other Services

In this section the infotainment aspects and other services will be described for C2C Systems. This services will provide information over the internet. Services like remote diagnostics can be useful for mechanics, or the driver can receive information about point of interests nearby his location.

Internet access in vehicle:
This scenario should avoid rear-end collisions, for example if a following vehicle suddenly brakes. The vehicles share information about speed, position and heading. To avoid collisions, the system has to use the own vehicle information and the information of vehicles nearby. If the system detects a critical proximity, it will warn the driver.[1]

Point of interest information:
The Point of Interest Notification allows local businesses, tourist attractions, or other points of interest to advertise their availability to nearby vehicles. In this case a roadside unit broadcasts information about opening hours or prices. The information will only be shown to the driver in appropriate situations. For example, if the fuel is running low, the vehicle presents the driver information about nearby gas stations.[1]

Remote Diagnostics:
Remote diagnostic allows service stations to assess the state of the vehicle without a making physical connection. This would allow software updates directly to the car, without the need to drive to a service stations. When a vehicle enters the area of a service garage, the service garage can query the vehicle for its diagnostic information to support the diagnosis of the problem reported by the customer. Furthermore the vehicles' past history and the customers' information can be loaded from a database to support the technician. With remote diagnostics, the time in service garages will be reduced and it will also result in lower cost for repair.[1]

Chapter 2

Wifi Direct

This chapter gives a short overview over Wifi Direct. Everything not mentioned or details about Wifi Direct can be found in the article "Wi-Fi CERTIFIED Wi-Fi Direct" from the Wi-Fi Alliance[2].

2.1 Overview

Wi-Fi Direct, or sometimes simply Wi-Fi P2P, is a standard which allows devices to connect directly to each other without requiring a wireless access point. With this technology users can connect to other devices in a way that makes it simpler and convenient for them. Because of the ability to connect directly to other Wi-Fi Direct devices, smartphones, printers, PCs and gaming devices can share their services without accessing a traditional network. Instead of connecting first to an existing infrastructure network and then connecting to another device, users can so directly connect to the device which offers the services they need. Wi-Fi Direct devices are allowed to create a one-to-one connection, or they could form a group with several devices[1].

Wi-Fi Direct devices support also the possibility to establish a connection with existing legacy Wi-Fi devices. This offers the possibility to create a direct connection with the hundreds of millions legacy Wi-Fi certified devices (802.11 a/g/n). According to documentation of the Wi-Fi Alliance paper the usage of Wi-Fi Direct brings some benefits for their users, among these:

Mobility and Portability:
Wi-Fi Direct devices can connect anytime and everywhere, because a Wi-Fi router or an access point is not required[2].

[1] http://www.isb.co.jp/product/mobile_solution/e/wifi_direct.html

Immediate Utility:

Once the user buys his first Wi-Fi Direct device, he is immediately able to create a direct connection between devices. Even if it is his first Wi-Fi Direct device at home, he could establish a direct connection with his existing legacy Wi-Fi devices[2].

Ease of Use:

The ability of Wi-Fi Direct discovery and the Service discovery allow users to find and identify available devices and services before establishing a connection[2].

Simple Secure Connection:

Devices with Wi-Fi direct use Wi-Fi Protected Setup (WPS) which allows to simple create a secure connection. To establish a secure connection the user has to press a button on both devices, or type in a Pin. The procedure depends on the device type[2].

At the present time[2] there are four Wi-Fi Direct standard generations. Devices that are equipped with the latest 802.11n Wi-Fi Direct standard, must be backwards compatible and must be able to communicate with devices which adapts older standards. The table below gives an overview of the current standards:

Standard	Bandwidth	Range
Wi-Fi Direct 802.11.a	54 Mbps	10 meters
Wi-Fi Direct 802.11.b	11 Mbps	20-50 meters inside buildings. Up to 500 meters outside
Wi-Fi Direct 802.11.g	54 Mbps	90 meters inside buildings. Up to 400 meters outside
Wi-Fi Direct 802.11.n	450 Mbps	150 meters inside building. Up to 500 meters outside

Table 2.1: Overview of the Wi-Fi Direct[3]

The data in the table were taken from the official Wi-Fi Direct standards there is no guarantee that all Wi-Fi Direct devices have to comply with these[2].

[2]June 2014.

[3]http://wifi-direct.net/wi-fi-direct-was-ist-das-eigentlich/

2.2 Technology Basics

Wi-Fi direct devices are capable of establish a peer-to-peer connection. They can form groups in a one-to-one or one-to-many topology. One Wi-Fi direct device is responsible for the group and acts as group owner. For legacy clients the group owner will appear as an Access Point on which they could connect. All Wi-Fi direct devices must be able to be in charge of a group and act as group owner. Furthermore all devices must be able to negotiate which device adopts the group owner role when they forming a new group with other Wi-Fi Direct devices. A group can contain Wi-Fi Direct devices and legacy devices, with the limitation that legacy devices can only act as clients within a group. The picture below shows a typical Wi-Fi Direct P2P group[2].

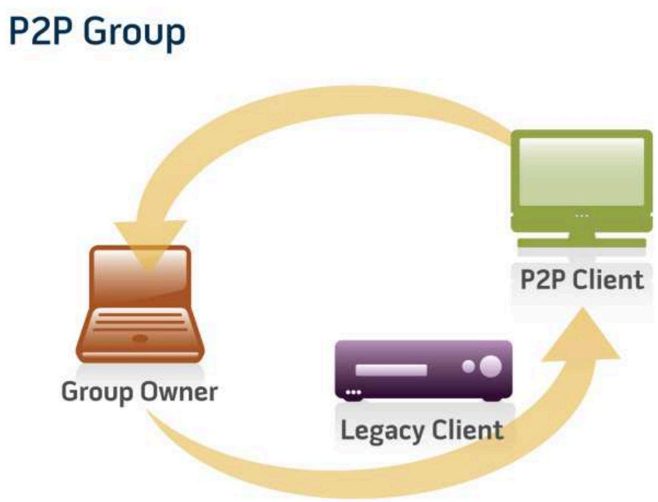

Figure 2.1: The figure shows a P2P group with a Wifi Direct client and a legacy device[2]

Chapter 3

Prototypes

This chapter will describe the implementation of a C2C application on three different devices. These devices are an Android Device, a Windows Phone and a rasberry pi.

3.1 Android

Since Android 4.0, devices with appropriate hardware are allowed to connect directly to each other over WI-FI P2P without an access point between them. According to the official Android Wi-Fi Peer-to-Peer documentation, Android's P2P framework complies with the WI-FI Alliances' WI-FI Direct certification program. With the usage of this API you are able to discover and connect to other devices when they support WI-FI P2P. According to documentations the advantage of WI-FI P2P besides Bluetooth or similar connection types is a fast connection across distances much longer than others, see table 2.1. This allows applications a fast exchange of data between multiple users, which could be useful for applications such as multiplayer games, photosharing applications and in general, all applications which are relying on a fast connection between a long distance.[1]

Prototype Implementation

In regard to the Car2Car project an Android application which tests the reliability and the functions of the WI-FI P2P APIs was developed. In light of the idea behind the Car2Car project and the ability of modern Android phones, to track the location of a user, this subchapter will show the results of the simple WI-FI P2P and GPS prototype. The simple prototype discover available peers, after a successful connection it sends the GPS location of the user to all connected peers. All peers mark the position of the other devices

[1] http://developer.android.com/guide/topics/connectivity/wifip2p.html

on the included google maps map with a marker. The picture below shows the design of the prototype application and describes the different sections.

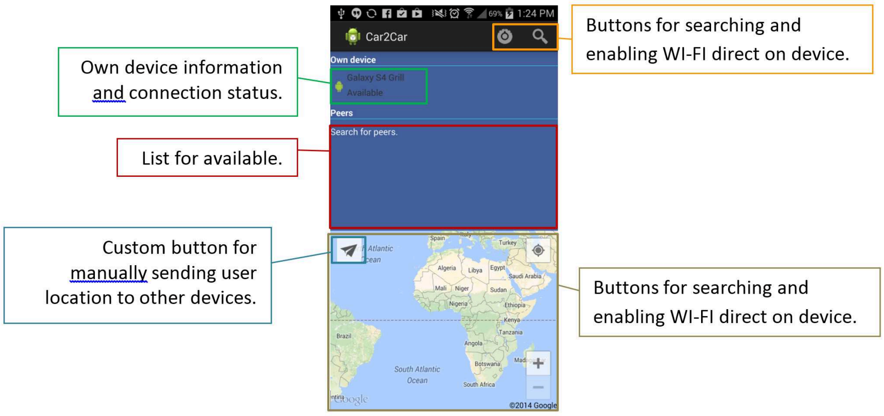

Figure 3.1: The figure shows the Android prototype application

When the application is started it automatically begins to search for available peers. Additionally, the user can search manually by pressing the search button in top of the application. The application filters available peers per instance name, which means only other C2C peers are shown. If peers are available they appear immediately in the list view. A user can connect to other devices by clicking on the specific item in the list view. If the connection attempt was successful the invited device will get connection invitation which the user has to accept. If the other user has accepted the invitation the two application shares their GPS location information. This means, every time the location of the device will change the current GPS information (Longitude and Latitude) will send to all connected peers. In some regions with weak GPS signals it could take several minutes until the application shares their GPS information with the other connected peers. The result of a successful connection is shown in the image below. The position of the other device is marked with a red google marker icon.

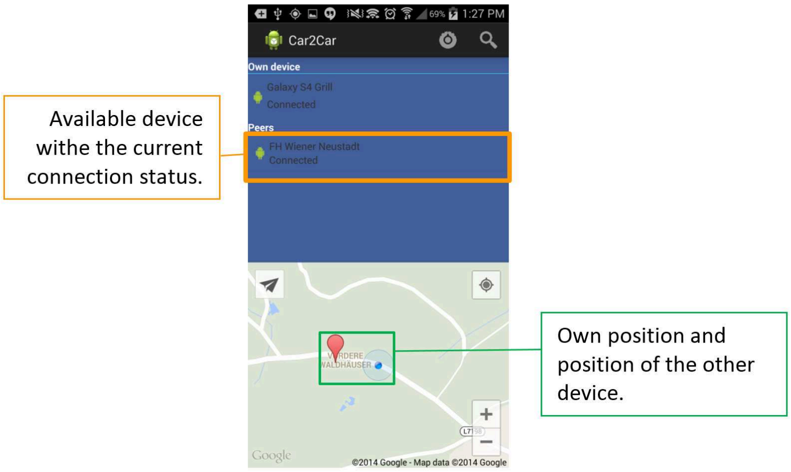

Figure 3: Application with connected device

Figure 3.2: The figure shows the Android prototype application with a connected device

Limitations and Problems

One of the biggest problems with Androids WI-FI P2P APIs, from the perspective of the C2C project, is the fact that every time a device wants to connect to your smartphone, it requires a confirmation from the user. Of course this had some security backgrounds and it make sense in other scenarios, but it is a major problem for using Android devices and this type of communication in the C2C project. For obtaining detail information, like the current location or other important details, it is necessary that all devices are connected automatically to each other when they are in the same area. The image below shows an invitation message which shows up on every device which receives a connection request[2].

[2]http://developer.android.com/training/connect-devices-wirelessly/wifi-direct.html

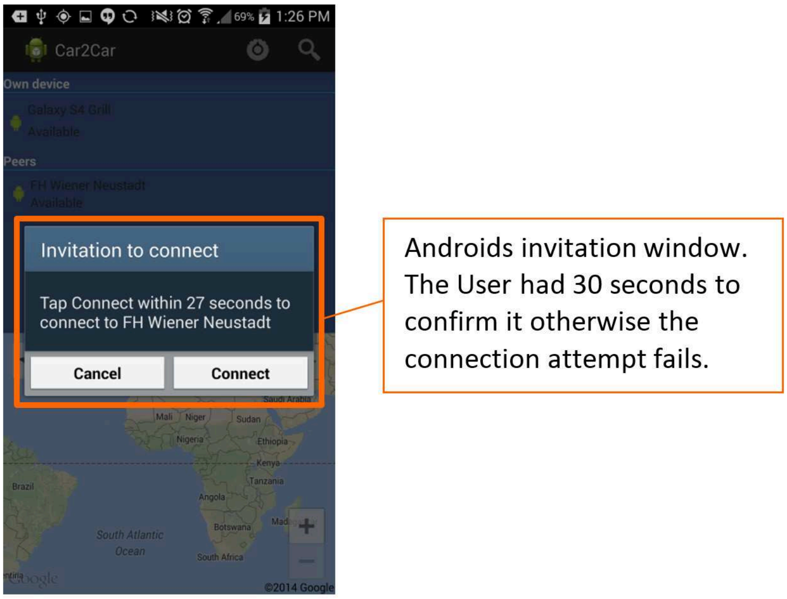

Figure 3.3: The figure shows the Android prototype application with a Wifi P2P connection request

According to the official WI-FI P2P documentation the range of the WI-FI P2P signal could be up to 500 meters. For the Android Car2Car prototype the test devices Samsung Galaxy S4 and Samsung Galaxy S2 Plus was used. It was not possible to confirm the range of 500 meters with the two devices. To test the maximal range of the signal, a few tests on a straight level road were carried out. It was determined that the signal at about 100-120 meters is lost. The tests were performed on foot and by car without any major differences. For a successful reconnect the distance between the devices was about 50-70 meters. The pictures below show the performed tests and describe their results.

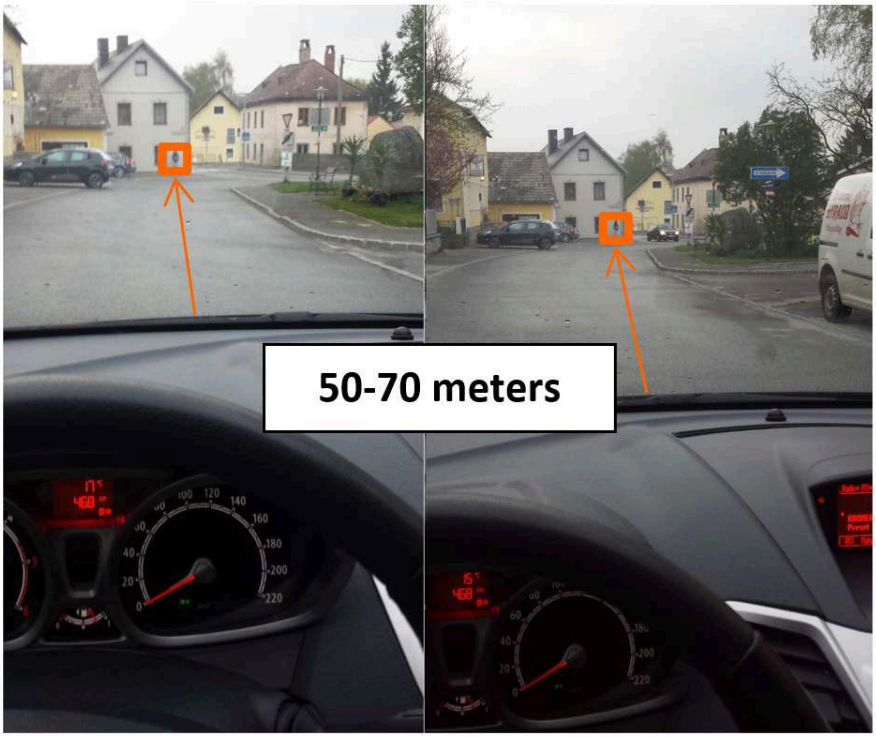

Figure 3.5: The figure shows the maximum distance for a successful reconnect

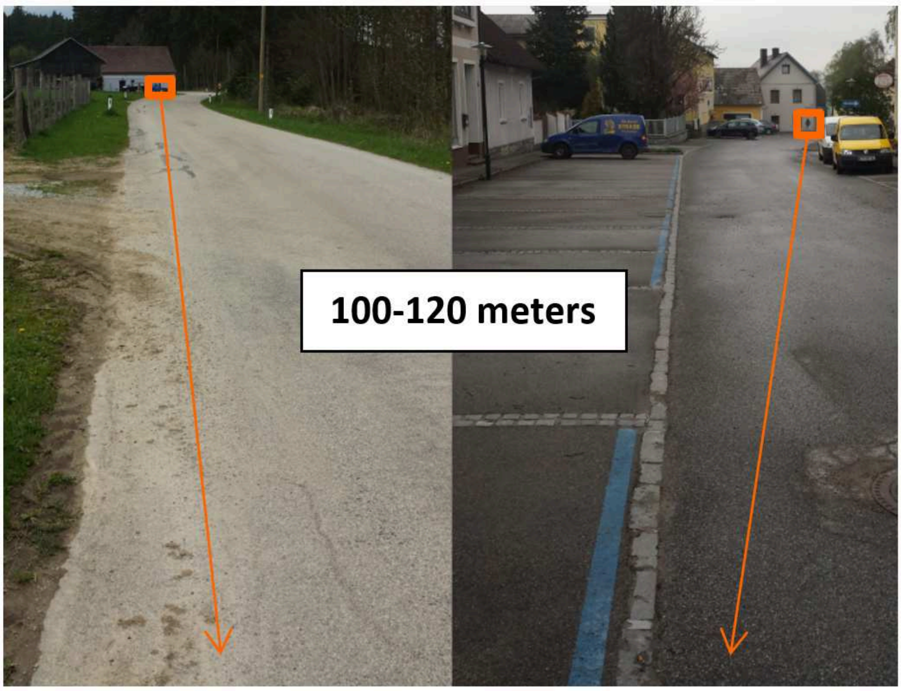

Figure 3.4: The figure shows the maximum range of the Wifi P2P signal

3.2 Raspberry Pi

With the right wifi dongle and driver, it is possible to use wifi direct on the Raspberry Pi too. After doing some research on the topic wifi direct on the big mobile plattforms the outcome was not satisfying. The problem was, that espacially iOS and Windows Phone had own proprietary implementations for wifi direct. Therefore the platforms weren't compatible with each other. To guarantee compatibility we had to find an alternative. The Raspberry Pi fits exactly our needs. It would be the wifi direct receiver and sender. Information will be transmitted from the Raspberry Pi to the smartphone, which presents the data to the user. We created a protoype which is able to connect to another wifi direct device. In further implementation it should connect to various devices and broadcast information. Forthermore the problem with accepting wifi direct connections, mentioned in 3.1, would be eliminated. In the current prototype it is not implemented, however the automatically connection confirmation could be handled in the connection script.

Prototype Implementation

A simple prototype was created for the Raspberry Pi, showing the possibility for establishing Wifi direct connections with other Wifi direct enabled devices in case of Car2Car communication. After establishing the connection with the device, a simple UDP-Server is started on the Raspberry Pi and a message will be transmitted over Wifi direct. The Realtek driver includes a console application named P2P UI, which enables Wifi direct and can be used for establishing connections.
After analysing the P2P UI Application, a shell script, which automates the pairing process was created. The user only needs to specify the name of his device and the MAC adress from the device he wants to start the pairing process with.
The progress of the connection will be displayed as debug messages, which show the status codes from the wifi dongle driver. Status code "9" means, that the negotiation process is running. The negotiation process will be repeated after 20 circles, if no status code "10" message for successful negotiation appears. The negotiation process has to be performed, otherwise the devices could not define which device should be the soft access point[3].
If the devices were paired successfully, a Python script with the server implementation will be started. The whole pairing process was tested with an Android device (Nexus 4) and the Rapsberry Pi. With an Android application called QPython, the client Python script could have been started.

[3]http://dishingtech.blogspot.co.at/2012/01/realtek-wi-fi-direct-programming-guide.html

If the pairing was successfull, the server should receive a message from the client.

Limitations and Problems

For using Wifi direct on a Rasberry Pi a special chipset in the Wifi dongle should be used. This prototype uses a Realtek chipset and the Wifi dongle TP-Link TL-WN725N V2. The corresponding driver allows the user to enable Wifi direct compatibility on the chipset. However finding the right driver for the Wifi dongle was not easy. The Raspberry Pi uses an arm chipset, where most of the driver are not precompiled. Either you find a driver or you have to compile the driver for your needs. After some trail and error, the right driver was found to support the Raspberry Pi. Another problem was the TP-Link nano wifi dongle. There is a tutorial for the Raspberry Pi to enable the wifi direct feature, however it was not possible to use

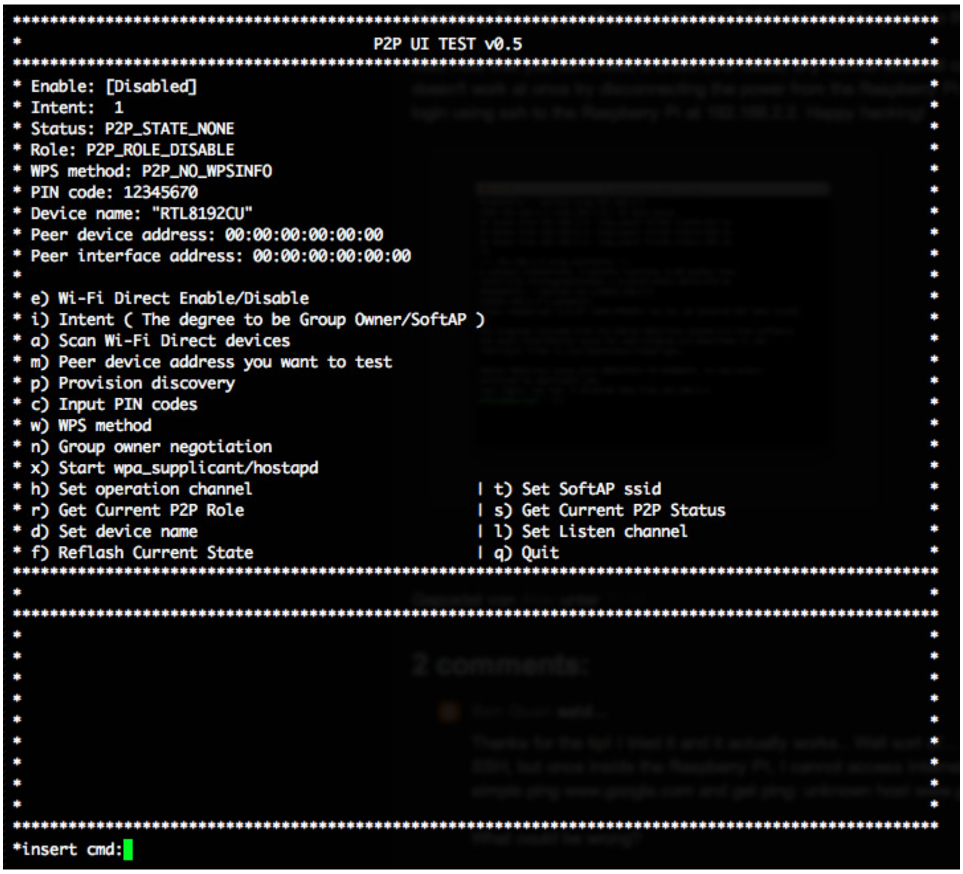

Figure 3.6: The figure shows the P2P UI application on the Raspberry Pi

the dongle, it was not recognized by the Raspberry Pi[4] [5]. After some more research we found out, that the wifi dongle we used was the second version[6]. The name of the dongle was identically with the one in the tutorial, only on

[4]http://www.youtube.com/watch?v=6GPv8TfZqe4

[5]http://www.youtube.com/watch?v=-HCrfkKeIM0

[6]http://www.mendrugox.net/2013/08/tp-link-tl-wn725n-v2-working-on-raspberry-raspbian/

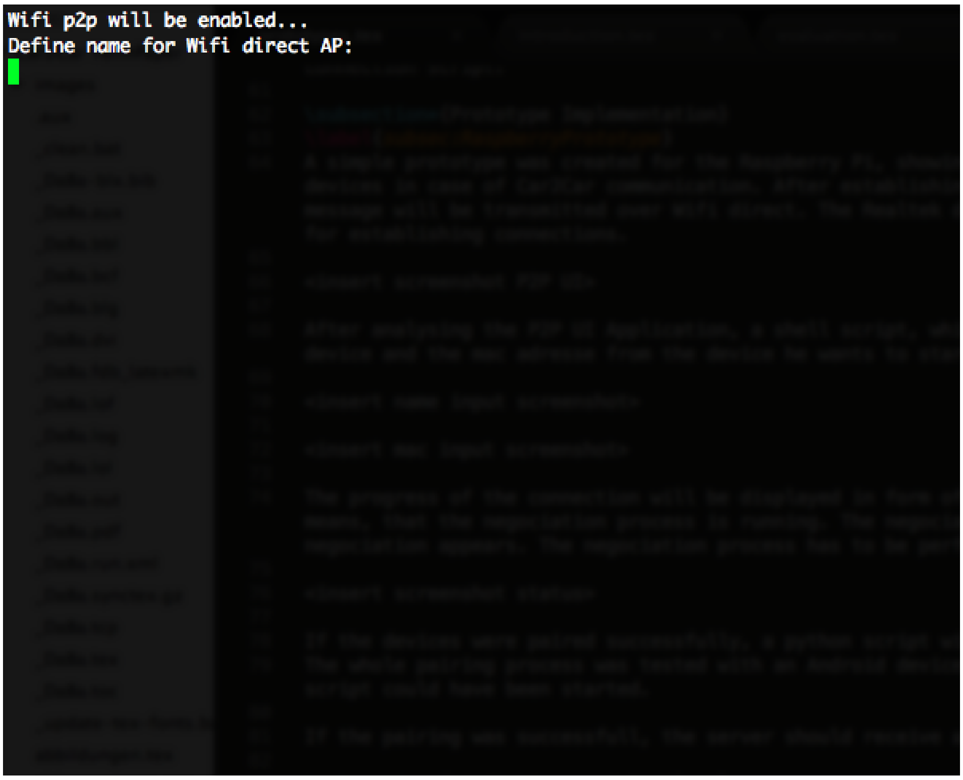

Figure 3.7: The figure shows the name input in the connect script

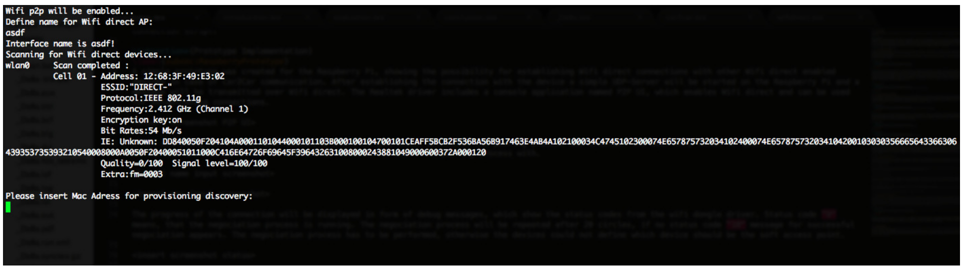

Figure 3.8: The figure shows the MAC adress input from the device you wish to connect to

the barcode of the packaging was a reference of the model.
The driver which was used in the tutorial was not compatible with the second version of this dongle. Furthermore the whole implementation is not reliable at the moment. Unfortunately it is not possible to garuantee a correct soft access point negotiation on every connect. An investiagtion to fix

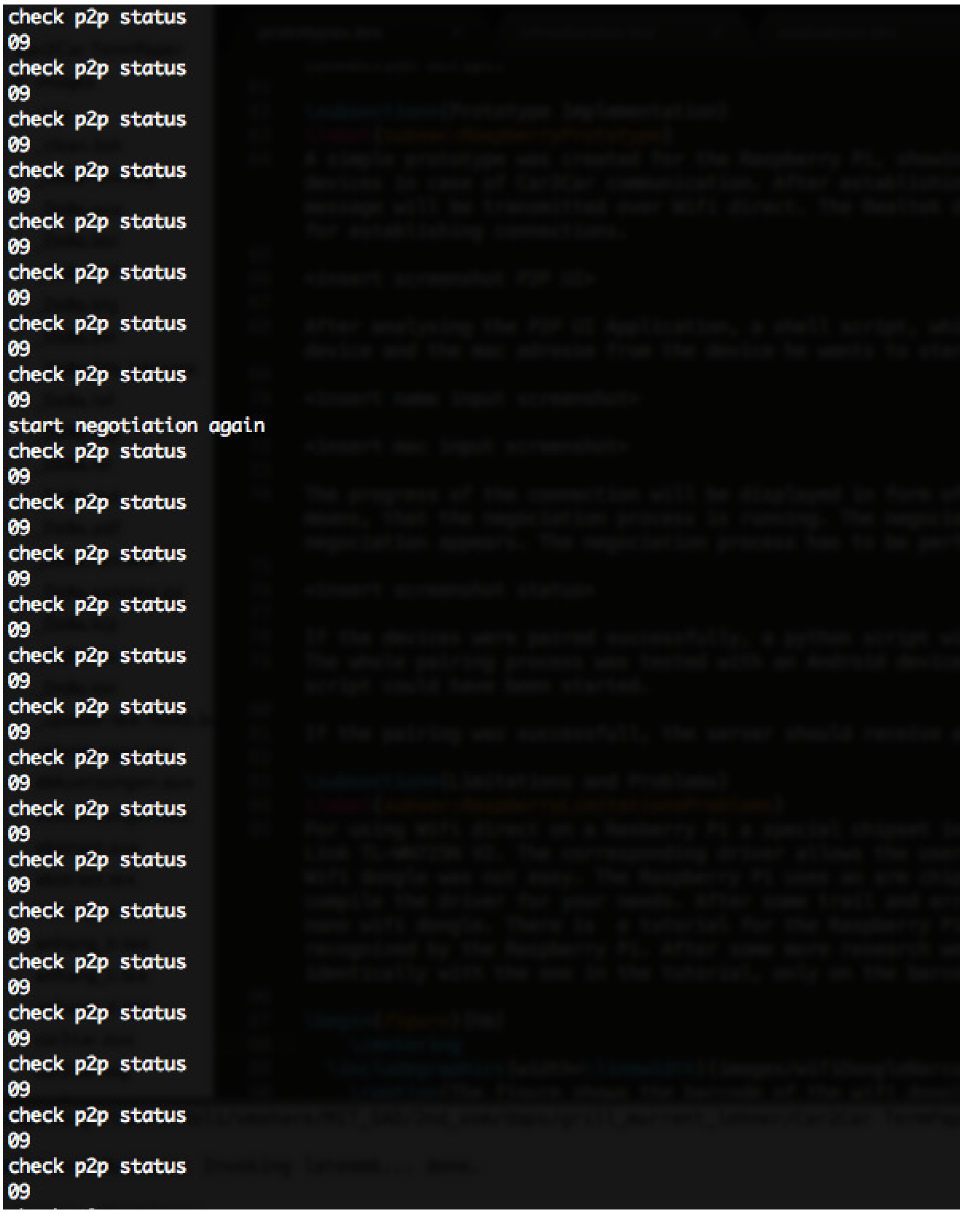

Figure 3.9: The figure shows the status output while the negotiation process is in progress

the problem was initiated, however this goal could not be reached for this paper. Theoratically the Raspberry Pi or Android device, depending which will be the soft access point, should share the IP address from the other device which is in the soft access point mode.

3.3 Windows Phone

Microsoft included Wifi Direct in his new Windows Phone 8.1 SDK, but actually there is no suitable documentation or sample which describes the usage of Wifi Direct within a Windows Phone app.
The other option would be to use there Windows.Networking.Proximity namespace which connects two phones directly to each other, but this requires a Bluetooth connection, a WIFI connection and the same app on both devices. Since this is not compatible with any other devices than a Windows Phones, this is, in our case, not the best solution. In the sample app description it is mentioned that an app which communicates via the proximity driver is used if no NFC is supported by the windows phone[7]. The biggest disadvantages are that the distance is limited to the Bluetooth distance and that it is not compatible with other phone manufacturer.

[7] http://code.msdn.microsoft.com/wpapps/Proximity-Sample-88129731

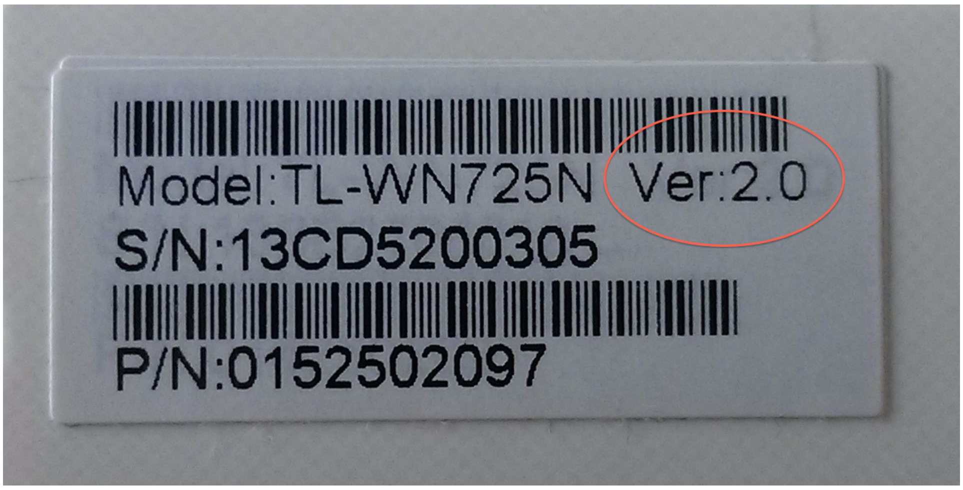

Figure 3.10: The figure shows the barcode of the wifi dongle packaging

Chapter 4

Evaluation

This chapter will point out some difficulties that have been experienced while developing the prototypes for the car2car project.

4.1 Communication between mobile platforms

One big problem is the compatibility between mobile platforms. The goal was to cover as many platforms as possible. For this project Windows Phone and Android was chosen. iOS was considered at the beginning too, however it uses a proprietary protocol, which is not compatible with other platforms and therefore can only be used for iOS devices. Problems occured with Windows Phone too. According to the Windows Phone documentation, it should be possible to connect with other platforms. However there are no good sample implementation for accomplishing this goal. In general the wifi direct technology is not mature on Windows Phone. Only Android could satisfy the needs. An implementation of a wifi direct prototype could be accomplished and a connection with a Raspberry pi could be generated.

4.2 Wifi direct

The field study of wifi direct is very new. In case of the Raspberry Pi prototype, there were not much sources to gather information on how to setup a wifi direct connection successfully. Even the Realtek P2P UI console application did not work reliably. For Android wifi direct is supported since Android 4.0. However the hardware manufacturer have the final say to enable this feature.

Chapter 5

Conclusion and future work

This chapter will recap on the goals of this paper and will present future work and changes in the field of C2C communication.

5.1 Conclusion

As a conclusion it can be said, that wifi direct is at the moment not the ideal solution to bring C2C communication to the public. The problem is, that most manufacturers have their own protocol for this technology. iOS has the MultiPeerFramework, which can be used only for iOS Devices. Windows Phone introduced Wifi direct within the Windows Phone 8.1 SDK release, which is at the moment only available for developers, however no suitable documentation was found. Android has introduced Wifi direct since Android 4.0 and obviously uses the standard protocol, because a connection could be established between the Raspberry Pi and an Android Device. Speaking of the Rapsberry Pi the current implementation for supporting wifi direct is not reliable enough at the moment to be used in such field as C2C communication. In some tests the connection was aborted all of a sudden. This technology needs to be more mature and consistent protocols or implementations among the smartphone manufacturers are needed.

5.2 Future work

If the implementation of the prototype will be continued, the next step would be to concentrate on the Raspberry Pi platform. Further tests should be processed and a deeper investigation of the wifi direct/P2P protocol would be necessary. If wifi direct will not become more reliable in the future, it would be possible to use a different technology, like communication over light. Breaking lights could be used, however it depends on free sight to the lights of the car in front and at the back. At the moment the prototype implementation is able to communicate with one device, later it should be possible to broadcast messages and information to more than one device in the near surroundings. Recent news show that C2C communication will be more and more important for the information technology. INTEL for example, recently launched the automotive platform "Kendrick Peak", which should be used in in-vehicle-infotainment systems, based on an embedded atom chipset[1]. This platform could be used in the field of C2C communication too. At the moment INTEL works with car manufacturers like BMW, Jaguar, Landrover, Kia and Hyundai on this platform. A cheap and interesting solution for C2C communication could be the "Vehicle Pi"[2]. It is a shield specially designed for the Raspberry Pi, which offers sensors and interfaces that can be connected to the car and retrieve information from it.

[1] http://www.golem.de/news/intel-kendrick-peak-automotive-plattform-mit-atom-antrieb-im-kofferraum-1405-106821.html

[2] http://vehiclepi.com/

Listings

List of Figures

References

[1] Car 2 Car Communication Consortium. “CAR 2 CAR Communication Consortium Manifesto”. In: (2007), p. 94.

[2] Wi-Fi Alliance. “Wi-Fi CERTIFIED Wi-Fi Direct”. In: (2010), p. 14.