基于Wi-Fi直连的文件传输应用程序

常瑞恒
515021910459
crh19970307@sjtu.edu.cn

洪韫妍
515021910452
hongyunyan@sjtu.edu.cn

洪逸宁
515021910453
evelinehong@sjtu.edu.cn

毛威超
515021910559
maoweichao@sjtu.edu.cn

ABSTRACT

我们基于Wi-Fi直连技术实现了可以在安卓端、PC端之间发送文件的应用程序，以方便在没有无线网络连接的环境中进行文件共享。我们调研了现有的实现相同功能的应用程序，并分析了它们各自的优点及局限性。在对Wi-Fi直连技术进行深入了解之后，我们独立开发出一款实用性、可靠性更强，功能更完善的应用程序。此外，我们对文件群发和保存分组等功能进行探讨，并在我们的应用程序中进行了实现。最后，我们通过实验对此应用程序的文件传输速度、有效传输距离等进行了评估。

1 背景介绍

随着无线网络技术的发展和无线接入点数量的增多，我们在生活中可以很方便地实现设备间文件的互传与分享。现有的很多应用程序针对这个问题实现了方便而高效的文件跨平台传输功能 [5, 6, 9]，人们也常采用在多个平台登录社交软件的方式进行文件传输[11]。

常用的文件传输方式固然方便，但它们的局限性在于，它们只能在具有网络连接的情况下才能正常使用，而在没有无线网络的环境中，人们可以选择的文件传输方式将大大受限。利用移动数据流量配合共享热点的方式可能产生一定的费用开销，基于蓝牙的方式传输速度难以令人满意，而基于USB数据线或U盘等便携存储设备则需要携带额外的硬件。

随着越来越多的电子设备开始支持Wi-Fi直连技术，我们认为基于Wi-Fi直连进行设备间文件传输已成为可能。Wi-Fi直连技术克服了现有方法的缺点，它可以在没有无线路由的环境中实现设备间的点对点连接，无需额外费用开销和硬件设备，且传输速率远高于蓝牙。现有电子设备虽在硬件上支持Wi-Fi直连技术，却往往并不提供便捷可用的文件传输软件，或仅支持在同品牌的设备间进行传输[4]。考虑到以上因素，我们基于Wi-Fi直连技术实现了可以在安卓端和PC端进行文件传输的第三方应用程序，以实现跨平台、跨品牌的文件便捷传输。PC端的实现依赖于微软UWP平台 (Universal Windows Platform)的WiFiDirect 应用程序接口，安卓端的实现依赖于API等级14的Wi-Fi P2P应用程序接口，因此我们的应用程序在Windows 10和Android 4.0及以上的系统中都可以稳定运行。

本文将包含以下部分：我们将在第2章简单介绍Wi-Fi直连的原理，第3章介绍相关工作，本项目安卓端和PC端的实现方式分别在第4章和第5章，第4章同时介绍了群发和分组功能的原理和实现，第6章展示了测试结果。

2 WI-FI直连相关原理

Wi-Fi直连，也被称作Wi-Fi P2P，是一种允许两台设备在没有无线接入点的情况下相互连接的Wi-Fi标准。在普通Wi-Fi通信中，两台设备并不是直接通信，而是将无线接入点作为中心枢纽，所有的信息都会在无线接入点上进行转发。而在Wi-Fi直连中，设备间

的通信并不需要专用的无线接入点，而是在两台设备进行第一次连接时进行协商，根据设备的电量情况和已建立连接数等因素，将其中一台设备作为无线接入点使用，另一台设备只需直接连接这个无线接入点即可进行通信。Wi-Fi直连支持WPA2的加密机制，最大传输速度为250Mbps，使用2.4GHz与5GHz两种频段频段，并它支持一对一，以及一对多模式[13]。

需要注意的是，Wi-Fi直连与无线Ad-hoc网络[1]并不相同。两者最大的区别在于，无线Ad-hoc网络支持多跳通信，而Wi-Fi直连只适用于单跳通信。Wi-Fi直连技术是Ad-hoc模式的延续，它要比Ad-hoc模式更快，同时可以更容易的检测周围设备并进行连接。

在Wi-Fi直连的工作流程中，一台设备首先根据Wi-Fi直连协议的内容扫描周围的设备(Device Discovery)，并识别它们各自支持的功能(Service Discovery)，当需要实现打印功能时，这台设备会首先扫描到周围所有支持Wi-Fi直连功能的设备，然后筛选出可以提供打印功能的设备列表。当两台设备进行连接时，它们会自动采用Wi-Fi保护设置(WPS)进行连接。设备间的通信过程采用WPA2进行加密，因此通信过程的安全性基本可以得到保护[2]。

现阶段已有部分产品采用了Wi-Fi直连技术进行通信。Roku 3[7]的遥控器采用Wi-Fi直连对电视进行控制，而非采用传统的红外或蓝牙控制。部分惠普打印机[3]同样支持Wi-Fi直连功能，在这种情况下，电脑无需连接局域网即可传输文件给打印机打印，这种方式可以实现比蓝牙通信更快的传输速率和更远的通信范围。三星在Android 2.3时代就在Galaxy S2[8]中添加了Wi-Fi直连功能，但只支持在三星的产品间进行通信。但需要注意的是，Wi-Fi直连至今只是一个由Wi-Fi联盟[12]定义的理论上的通信标准，现阶段想要实现像蓝牙一样便捷的通信绝非易事，目前所有基于Wi-Fi直连的产品都需要在软件层进行特别地设计。

[1]Ad-hoc模式是最早期的直连方式，它是一种特殊的无线移动网络。网络中所有结点的地位平等，无需设置任何的中心控制结点。

[2]值得一提的是，很多设备的生产厂商仍在采用WPS的PIN码模式进行连接，这会给基于Wi-Fi直连的通信带来一定的安全隐患。

3 相关工作

在本章中，我们将列举现有的基于Wi-Fi直连实现的文件传输应用程序，并分析其各自的优缺点。

3.1 WiFi Shoot[2]

WiFi Shoot是Google Play上第一个基于Wi-Fi直连进行安卓设备间文件传输的应用程序，该软件运行截图如图1(a)所示。该软件操作简单快捷，在不同设备上兼容性较好，作者声称已在Galaxy Nexus、Nexus 7、Galaxy S2等多种设备上进行了兼容性测试。

这款应用的缺点在于，它的免费版对传输文件类型、单个文件大小、每次传输的文件数量和传输速度都进行了限制，导致免费版实际使用并不方便。另外该软件把Google Play作为唯一正式发布渠道，对于国内用户来说软件升级和更新并不方便。

3.2 WiFi Direct File Transfer[1]

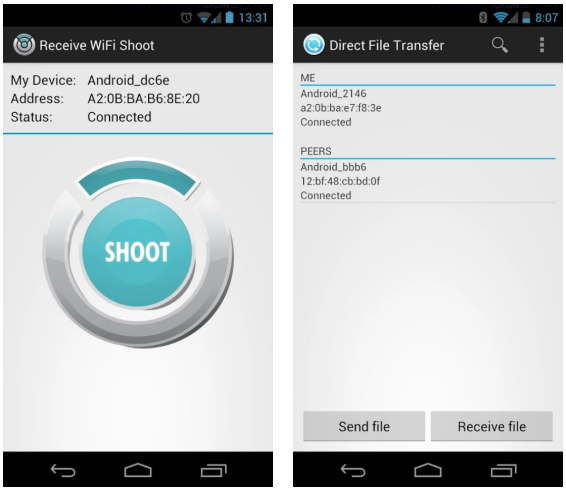
WiFi Direct File Transfer同样是一款基于Wi-Fi直连进行安卓设备间文件传输的应用程序，它的运行时截图如图1(b)所示。该软件的优点在于并没有对用户使用作出任何限制，且代码已在Github开源。缺点在于该软件只能实现安卓设备间的文件传输，并没有提供PC端的配套版本，且由于作者并没有对软件提供维护，软件最近一次更新是在2012年，由于API版本的变化，在如今的设备上已经无法正常使用。

3.3 SuperBeam[10]

SuperBeam是一款功能更加完善的基于Wi-Fi直连的跨平台文件传输软件，它的运行时截图如图1(c)和图1(d)所示。该软件同时提供了安卓端和PC端版本，支持任意格式的文件发送，同时提供基于二维码和共享密钥的主动接收功能，且作者长期提供软件维护和升级。

但该软件仍有一些使用时不尽如人意的地方，它只有安卓端应用是免费的，PC端软件需要额外付费；它的免费版本存在大量广告；同样，由于作者将Google

Play作为唯一正式发布渠道，国内用户的更新和升级同样并不方便。



(a) WiFi Shoot (b) WiFi Direct File Transfer

(c) SuperBeam(1) (d) SuperBeam(2)

图 1: 已有软件运行截图

4 安卓端实现方式

在安卓的手机端，我们的工作分为了三部分，第一部分根据WIFI Direct的协议实现两个安卓手机之间的连接以及他们之间信息与文件等传输。第二我们考虑了如何将WiFi direct的协议用于多个即大于两个安卓手机的传输，提出一个组连的结构想法。最后我们简单地实现了一下基本的组连架构和思想。

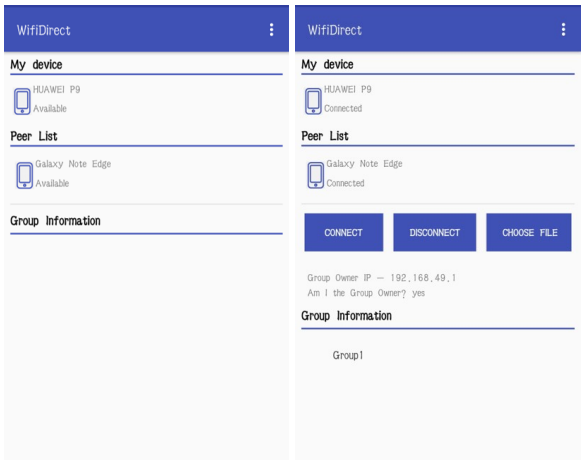


图 2: 软件UI设计

4.1 Wi-Fi直连协议双机的实现

在整个P2P连接的过程中，我们通过WiFi直连的底层机制来进行最初的设备查找和相互连接，再连接后通过socket通信的方式进行双向的文件发送和接受。

首先根据 WiFi Direct协议的底层需求，我们先进行相应权限的申明和设置，在 manifest中申明了包括 ACCESS_COARSE_LOCATION, CHANGE_WIFI_STATE，ACCESS_WIFI_STATE 和 INTERNET 等权限内容。

其次，考虑到 WiFi direct中我们需要一个监听相关状态的变化，并在改变时进行广播该项 Intent，因此我们实例化一个 IntentFilter，设置它用于状态改变的监听，监听的状态包括了 WiFi开关状态的改变 (WIFI-_P2P_STATE_CHANGED_ACTION),P2P 列表的改变 (WIFI-_P2P_PEERS_CHANGED_ACTION),P2P连接状态的改变 (WIFI_P2P_CONNECTION_CHANGED_ACTION), 以及相应设备配置信息的改变 (WIFI_P2P_THIS_DEVICE-_CHANGED_ACTION)。并且我们建立一个 Broadcast - Receiver 类，来监听这些事情的改变，并做出相应的操作

同时我们先获得一个WifiP2pManager的实例，并对其进行进行初始化，返回得到WifiP2pManager.channel的对象。

在完成基本的初始化设置和对象的获得后，我们通过调用discoverPeers()函数来开始寻找周围的设备。在成功搜索到设备后，我们会被监听到 peer 列表发生了改变，从而根据代码进行后续的操作。整个搜索的过程会持续保持活动状态，实时更新设备列表，直至建立连接或者形成P2P的组。

随后我们通过实现了WifiP2pManager.PeerList - Listener接口，获取到相应搜索到的设备的各项信息，包括设备的名字，对应的MAC地址等。之后，我们可以进行 P2P的连接。我们创建了 WifiP2pConfig的对象，并根据上面列表中获得的设备信息，进行调用connect()函数。整个的连接过程是通过mac地址信息进行的连接。由于我们这边仅考虑了支持wifi direct的设备（即大于Android4的设备）所以我们默认在连接时不需要明确相应的密码。若如果需要邀请不支持的设备入组，我们可以通过 requestGroupInfo中的信息来获取密码以及名称等各项P2P组内信息。

在进行相应的连接的过程中，我们的两个设备中的其中之一会被指定为Group Owner，这个的指定可以通过系统随机，也可以在程序中申明优先级来进行强制选择。在P2p连接建立后，参与连接的设备都可以直接获得本次Owner的IP地址，则相应的非GroupOwner设备对该IP地址进行Socket连接（这边我们这轮的连接固定了双方的端口选用8888），而Owner的设备则时刻保持对对应端口的监听，实现第一轮socket的建立，并据此获得队友设备的IP地址，并进行了相应的保存。

通过 socket的首轮连接我们互相之间都能获得了对方的IP地址，这样我们在接下来的传输中，可以通过对应的 socket连接来收发各类文件，包括 jpg，gif，word，pdf，mp3等等格式。

以上就是一个 P2P建立连接并且收发文件的整个过程。我们可以发现整个过程的通信原理还是socket的通信。而 socket通信必需需要对应的 IP地址，而 Wifi Direct 是支持在不连接具体的 WIFI仅开启 WIFI开关并且不开启3G/4G等通信的方式下（即没有真实的IP地址分配的情况下）完成传输的。我们通过具体的实验发现，在这种情况下Owner的地址固定为192.168.49.1，即是通过了一种近似热点网络的方式建立了一个几个手机间的内网，来实现接下来的socket连接。

4.2 基于Wi-Fi直连群发的组连架构

在实现基于 WiFi Direct的 P2P通信后，我们思考如何对其功能实现进行扩展。在 Wifi Direct的底层设计中，提供了 createGroup的函数可以满足邀请不具有 Wifi Direct功能的设备加入 group中来接受文件传输。而我们这边主要考虑的是多个支持的设备之间的一对多的进行发送和接受文件，满足一个 group 的每一个 device 都可以像组内的每一个设备发送文件（当然也可以单一的进行文件的发送）。因此我们设计了以下结构：

(1) 由一个设备通过选择列表中对应的设备，进行组建group，进行命名并且保存这个group的所有设备信息。
(2) 在首轮组建中，主设备对每一个选中的设备进行 connect，从而获得对应设备的IP地址，并向其传输 group的名字以及设备信息，做到group信息的广播。在首轮连接后run 不进行文件的传输，则设备之间断开相互的连接，处于 unpaired状态。
(3) 当群组中有设备需要向组内成员传输文件，他根据设备的信息，对每一个设备同时进行连接，并同时发送文件。该设备向群组成员发送完文件后，若在设定时间内无继续操作，则断开设备见的连接。
(4) 当群组中的某一设备想要离开群组时，通过和组内的各个设备连接，告知列表的更新变化，并且在通告完毕后在自己的所有 group列表中移除该 group。
(5) 当整个组中仅剩下一个设备时，该设备也自动移除了自己 group列表这个 group的所有信息。

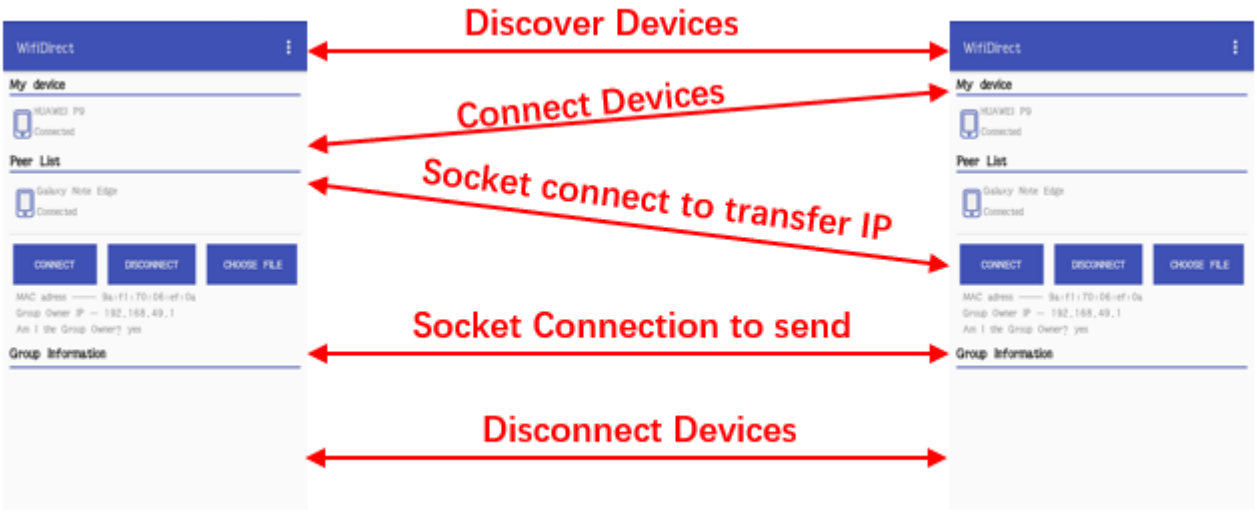


图 3: P2P连接过程

4.3 基于Wi-Fi直连群发的组连实现

在安卓端的代码实现中，我们目前主要实现了单一设备进行一对多发送的功能，但没有实现设备信息的组内共享。在设备的一对多发送中，我们通过进行伪并行方式，逐一与组内设备进行了 P2P的连接，获取设备的IP地址并进行保存。然后再选定发送的文件内容后逐一的根据IP地址建立socket连接，对相应的设备进行文件的传输。可以看到我门这边所谓的同时一对多发送，其实是一个伪并行的连接和发送的方式。由于连接设备需要一定的时间，因此在连接过程中逐个设备其实会有一定的延迟。但是在发送过程中，虽然也是一个伪并行的方式，但是由于文件传输速率较快，所以在文件大小适宜的情况下各个设备之间的接收近似为同时。

这个实现部分，体现了我们对群发传输的一个思想的初步实现，也很大程度的缓解了单个设备对多设备发送同一文件时候的复杂操作。

5 PC端实现方式

由于Windows平台并没有自带Wifi Direct的功能，在PC端，我们基于Universal Windows Platform（UWP）平台，使用C#实现了设备声明、设备握手、文件传输的功能，由于UWP平台的特性，我们的Wifi Direct软件可以在一切装有Windows系统的电脑、平板以及Windows phone上面运行，再结合前面的安卓端实现，具有很强的通用性。

5.1 Wi-Fi握手协议

PC之间通过Wifi Direct进行直连的第一步是设备之间进行握手配对，需要有一台设备作为Advertiser，对外“公布”自己的信息，这样另一台设备才会发现它，因为PC端没有控制wifi direct的软件，因此这个过程我们通过调用了Windows.Devices.WiFiDirect命名空间的 API，WiFiDirectAd - vertisement、WiFiDirect-AdvertisementPublisher直接控制硬件，使得advertiser作为一个WirelessAccessPoint（AP）被发现。

当AP建立之后，另一个设备首先进行搜索，此处使用WiFiDirectDevice的api，搜索到设备之后通过WiFi - DirectConnectionRequest 与这个AP进行连接，进行连接之后，connector与advertiser互相得到对方ip，开始建立socket连接，建立之后，发送一个字符串作为session ID，对方回应后可以进行会话。

PC端的wifi握手协议实现如下（流程图见图5）：

(1) Advertiser设备调用底层硬件，建立AP。


CS339 Project
Wi-Fi Direct
1) Advertiser
2) Connector
Advertisement
Listen State:
Normal
Enable Listener
Group Owner
Informational Element:
Add
Start Advertisement
Stop Advertisement
Connected devices:
Close Device
Enter a message
Send
Choose File
Send File
Choose Path
Save File
Connection
Group Owner Intent:
Default
Add configuration methods in order of preference, most preferred first.
Push Button
Add
Preferred Pairing Procedure:
Group Owner Negotiation
© 2017 Computer Network Group. All rights reserved
CS339 Project
Wi-Fi Direct
1) Advertiser
2) Connector
Discovering
Device Selector:
Association Endpoint
Start Watcher
Show IEs
Discovered Devices:
Connection
Group Owner Intent:
Default
Add configuration methods in order of preference, most preferred first.
Push Button
Add
Preferred Pairing Procedure:
Group Owner Negotiation
Connect
Unpair
Enter a message
Send
Close Device
Choose File
Send File
Choose Path
Save File
Connected Devices:
© 2017 Computer Network Group. All rights reserved

图 4: 软件UI设计

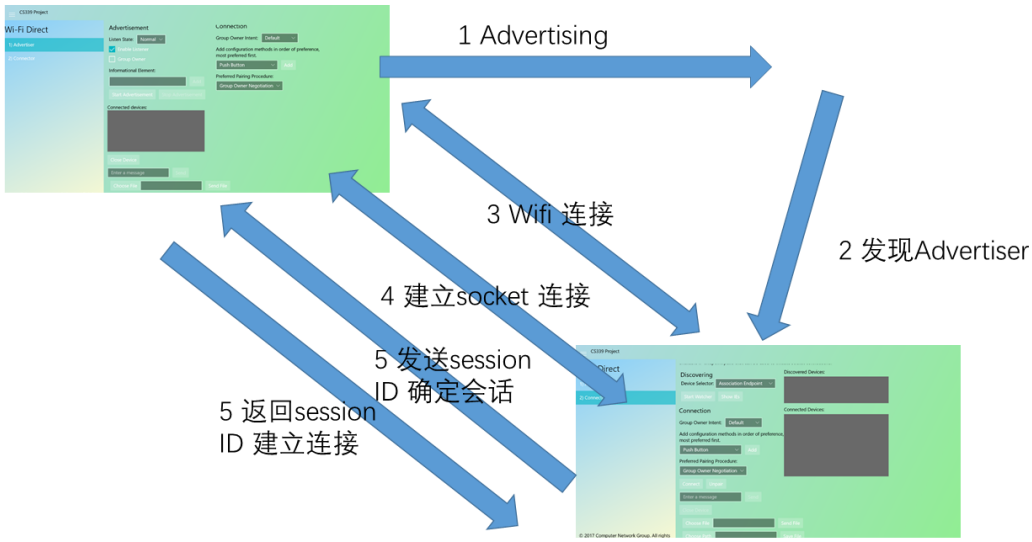


图 5: 握手流程图

(2) Connector设备查找到该设备，连接该设备。
(3) Connector与Advertiser之间进行socket连接。
(4) Connector向Advertiser发送字符串，指定session ID，Advertiser返回字符串，连接建立。

5.2 Wi-Fi文件传输

当设备之间建立起socket连接之后，即可开始文件的传输，我们采用StreamSocket类进行传输，将文件转化为比特流，通过socket进行传输，无论是adversiser还是connector都需要对本地固定端口（我们使用的是8888）进行侦听，如果收到比特流则进行文件传输。

6 安卓端测试情况

6.1 传输速率测试

首先，我们用596MB和249MB的视频测试文件传输速率，两个视频文件的传输速率分别为4.07MB/s和3.43MB/s。然而，我们考虑到视频文件在传输过程中有被压缩的可能，因此通过文件传输前大小与传输时间作除法所得到的数值不一定能准确反映实际的发送速率。考虑到这种情况，我们利用线性同余随机数生成法生成了满足随机性检验的01比特串进行测试，测试文件的大小分别为500MB和100MB。其中，500MB的随机比特文件的传输速率为3.33MB/s，而100MB的文件的传输速率为3.24MB/s。上述所有数据都是重复10次实验后的平均值。

6.2 传输距离测试

我们在室外进行直线距离传输。我们测试的最远距离为280m（大致为从电院一号楼到叔同路的距离）。而在有墙壁等物理障碍的真实环境下，我们发现最大传输楼层为2层（电院一号楼2楼和4楼同一位置测试）。

REFERENCES

[1] 2012. WiFi Direct File Transfer. (2012). https://play.google.com/store/apps/details?id=ca.nickadams.wifi.direct.file.transfer [Online; accessed 19-December-2017].
[2] 2015. WiFi Shoot. (2015). https://play.google.com/store/apps/details?id=com.budius.WiFiShoot [Online; accessed 19-December-2017].
[3] 2017. HP Printer. (2017). http://store.hp.com/us/en/cv/printers [Online; accessed 19-December-2017].

[4] 2017. Huawei P8 User Manual. (2017). https://www.manualslib.com/manual/992914/Huawei-P8.html?page=66 [Online; accessed 19-December-2017].

[5] 2017. Portal. (2017). http://portal.pushbullet.com/ [Online; accessed 19-December-2017].

[6] 2017. Resilio. (2017). https://www.resilio.com/ [Online; accessed 19-December-2017].

[7] 2017. Roku. (2017). https://www.roku.com/index [Online; accessed 19-December-2017].

[8] 2017. Samsung Galaxy S2. (2017). http://www.samsung.com/uk/smartphones/galaxy-s2/GT-I9100LKAXEU/ [Online; accessed 19-December-2017].

[9] 2017. Send Anywhere. (2017). https://send-anywhere.com/ [Online; accessed 19-December-2017].

[10] 2017. SuperBeam. (2017). https://superbe.am/ [Online; accessed 19-December-2017].

[11] 2017. Telegram. (2017). https://telegram.org/ [Online; accessed 19-December-2017].

[12] 2017. Wi-Fi Alliance. (2017). https://www.wi-fi.org/discover-wi-fi/wi-fi-direct [Online; accessed 19-December-2017].

[13] W Alliance. 2010. Wi-Fi Peer-to-Peer (P2P) Specification v1.1. 1 (01 2010), 1–159.