



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

**Comunicação Dispositivo a
Dispositivo (D2D) em Redes Móveis:
Visão Geral e Experimentação**

Daniel Martins Reis

**TRABALHO DE
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:

Theo Silva Lins

COORIENTAÇÃO:

Vinícius Fernandes Soares Mota

**Fevereiro, 2018
João Monlevade–MG**

Daniel Martins Reis

**Comunicação Dispositivo a Dispositivo (D2D)
em Redes Móveis: Visão Geral e
Experimentação**

Orientador: Theo Silva Lins

Coorientador: Vinícius Fernandes Soares Mota

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

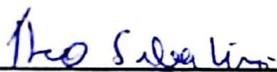
Fevereiro de 2018

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

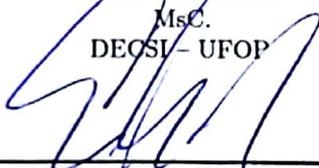
Comunicação Dispositivo a Dispositivo (D2D) em Redes Móveis: Visão Geral e Experimentação

Daniel Martins Reis

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:



Theo Silva Lins
MsC.
DECSI - UFOP



Erik de Britto e Silva
MsC.
Examinador
DECSI - UFOP

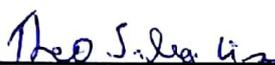


Marlon Paolo Lima
MsC.
Examinador
DECSI - UFOP

João Monlevade, 16 de fevereiro de 2018

ATA DE DEFESA

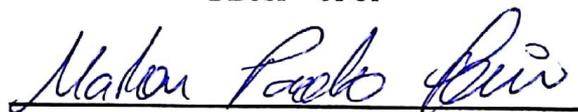
No dia 16 do mês de Fevereiro de 2018, às 17:00 horas, na sala C304 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo(a) aluno(a) **Daniel Martins Reis**, sendo a Comissão Examinadora constituída pelos professores: Theo Silva Lins, Vinicius Fernandes Soares Mota, Erik de Britto e Silva, Marlon Paolo Lima. O(a) candidato(a) apresentou a monografia intitulada: “**Comunicação Dispositivo a Dispositivo (D2D) em Redes Móveis: Visão Geral e Experimentação**”. A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 9,5 (NOVE E MEIO), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo(a) graduando(a).



Theo Silva Lins
MsC.
DECSI - UFOP



Erik de Britto e Silva
MsC.
Examinador(a)
DECSI - UFOP



Marlon Paolo Lima
MsC.
Examinador(a)
DECSI - UFOP



Daniel Martins Reis

TERMO DE RESPONSABILIDADE

Eu, **Daniel Martins Reis** declaro que o texto do trabalho de conclusão de curso intitulado “*Comunicação Dispositivo a Dispositivo (D2D) em Redes Móveis: Visão Geral e Experimentação*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 16 de fevereiro de 2018


Daniel Martins Reis

Dedico este trabalho a Deus, por ser essencial em minha vida e autor do meu destino. Ao meu pai Agnaldo, minha mãe Shirlene, aos meus irmãos, avós, tios, sobrinha e toda minha família, que esteve sempre presente com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa em minha vida. Aos professores, amigos e às pessoas que convivi na UFOP durante a jornada deste curso.

Agradecimentos

Agradeço a Deus que iluminou o meu caminho. À minha família por sua capacidade de acreditar e investir em mim. Mãe, seu cuidado e dedicação deram a esperança para seguir em frente. Pai, sua presença significou segurança e certeza de que não estou sozinho nesta caminhada. Irmãos, sobrinha, avós e tios que me incentivaram. Professores que durante muito tempo me ensinaram e mostraram o quão bom é estudar - em especial ao Vinícius Mota, Erik de Britto e Theo Lins com quem partilhei a essência deste trabalho, pela paciência, convívio, compreensão, apoio, incentivo, amizade e ensinamentos - nossas conversas e análises foram fundamentais. Aos amigos e colegas pelas alegrias, tristezas, incentivos e apoio constantes. A todos aqueles que de alguma forma estiveram e estão próximos de mim.

“Science is more than a body of knowledge; it is a way of thinking.”

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

A comunicação *Dispositivo a Dispositivo (D2D)* vêm se destacando pela capacidade de reduzir o tráfego de dados nas operadoras de celular. Diversos modelos e algoritmos para comunicação *D2D* eficiente têm sido propostos recentemente. Entretanto, apesar da disponibilidade de tecnologias para comunicação *D2D* na maioria dos dispositivos atuais, tais como *Wi-Fi Direct*, existem poucas iniciativas de aplicações reais que utilizem este modelo de comunicação. Este trabalho apresenta uma avaliação experimental e discute as limitações e desafios da implementação da comunicação *D2D* em dispositivos de prateleira. Neste sentido, propomos o *ShareFile*, um aplicativo para compartilhamento de conteúdo via *Wi-Fi Direct* em dispositivos *Android*. Por meio de uma análise sistemática, foi realizada uma verificação de desempenho comparando-o a soluções centralizadas. Os resultados mostram que a distância influencia fortemente no desempenho da comunicação *D2D*. O tempo para encontrar dispositivos próximos varia entre 400ms a 1.5s, enquanto o tempo de conexão pode chegar à 6s e vazão atinge até 52Mbps em dispositivos próximos.

Palavras-chaves: Comunicação *D2D*, *WiFi Direct*, Redes Móveis.

Abstract

Device-to-Device (D2D) communication has been notable for its ability to reduce data traffic on mobile operators. Several models and algorithms for efficient *D2D* communication have been proposed recently. However, despite the availability of technologies for *D2D* communication on most current devices, such as *Wi-Fi Direct*, there are few actual application initiatives using this communication model. This paper presents an experimental evaluation and discusses the limitations and challenges of implementing *D2D* communication on shelf devices. In the sense, we propose the *ShareFile*, an application for sharing content via *Wi-Fi Direct* on *Android* devices. Through a systematic analysis, a performance check was performed comparing it to centralized solutions. The results show that the distance influences the communication performance *D2D* strongly. The time to find nearby devices varies from 400ms to 1.5s, while the connection time can reach 6s and flow rate reaches up to 52Mbps on nearby devices.

Key-words: *D2D* Communication, *WiFi Direct*, Mobile Networks.

Lista de ilustrações

Figura 1 – Transmissão de conteúdo utilizando rede infraestruturada.	15
Figura 2 – Envio de conteúdo utilizando <i>Wi-Fi Direct</i>	15
Figura 3 – Fluxo de dados para Comunicação <i>D2D</i>	22
Figura 4 – <i>ShareFile</i> : Diagrama de pacotes.	24
Figura 5 – <i>ShareFile</i> : Menu principal e modelos de comunicação.	27
Figura 6 – <i>ShareFile</i> : Configurações.	27
Figura 7 – <i>ShareFile</i> : Diagrama de atividades <i>D2D GO</i>	28
Figura 8 – <i>ShareFile</i> : Diagrama de atividades <i>D2D Client</i>	29
Figura 9 – <i>ShareFile</i> : Criando o <i>GO</i>	30
Figura 10 – <i>ShareFile</i> : Fase de busca por grupos <i>Wi-Fi Direct</i>	30
Figura 11 – <i>ShareFile</i> : <i>Client</i> se conectando no <i>GO</i>	31
Figura 12 – <i>ShareFile</i> : <i>Client</i> enviando arquivo para o <i>GO</i>	32
Figura 13 – <i>ShareFile</i> : Diagrama de atividades do servidor local.	33
Figura 14 – <i>ShareFile</i> : Diagrama de atividades do servidor externo.	33
Figura 15 – Transmissão de conteúdo via comunicação <i>D2D</i>	36
Figura 16 – Transmissão de conteúdo via servidor externo.	36
Figura 17 – Transmissão de conteúdo via servidor local.	36
Figura 18 – Tempo para encontrar <i>GO</i>	39
Figura 19 – Tempo estabelecimento de conexão <i>D2D</i>	39
Figura 20 – Tempo conexão <i>GO</i>	40
Figura 21 – Tempo de transmissão de conteúdo <i>Client</i>	41
Figura 22 – Tempo de transmissão de conteúdo <i>GO</i>	41
Figura 23 – Largura de banda média <i>D2D</i>	41
Figura 24 – Largura de banda em relação a distância <i>D2D</i>	41
Figura 25 – Largura de banda servidor local.	42
Figura 26 – Largura de banda servidor na nuvem.	42

Lista de tabelas

Tabela 1 – Arquivos transmitidos nos experimentos em <i>testbed</i>	37
Tabela 2 – Tempo médio e desvio padrão de conexão entre um par de dispositivos.	40

Lista de abreviaturas e siglas

3GPP	3rd Generation Partnership Project (Projeto de Parceria de Terceira Geração)
AES	Advanced Encryption Standard (Padrão de Criptografia Avançada)
BS	Base Station (Estação Base)
COSI	Colegiado do Curso de Sistemas de Informação
CSV	Comma Separated Values (Valores separados por vírgula)
D2D	Device-to-Device (Dispositivo-a-Dispositivo)
D2D ProSe	Device-to-Device Proximity Service (Serviço de proximidade Dispositivo-a-Dispositivo)
DECSI	Departamento de Computação e Sistemas
GB	Gigabyte
GO	Group-Owner (Proprietário do grupo)
ICEA	Instituto de Ciências Exatas e Aplicadas
ism	Industrial Scientific and Medical (Faixas de frequência)
JM	João Monlevade
KB	Kilobyte
Kbps	Kilobite/segundo (Taxa de transferência)
LTE-A	Long Term Evolution-Advanced (Evolução a longo prazo - Avançada)
m	Metros
MANETs	Mobile Ad-hoc Networks (Redes Móveis Ad-Hoc)
MB	Megabyte
Mbps	Megabite/segundo (Taxa de transferência)
ms	Milissegundos
RTT	Round-Trip-Time (Tempo de ida e volta)

s	Segundos
SI	Sistemas de Informação
TCP	Transmission Control Protocol (Protocolo de Controle de Transmissões)
UDP	User Datagram Protocol
UE	User Equipament (Equipamento de usuário)
UFOP	Universidade Federal de Ouro Preto
WLAN	Wireless Local Area Network (Rede Local Sem Fios)

Sumário

1	INTRODUÇÃO	15
1.1	Problema	16
1.2	Objetivos	16
1.3	Metodologia	17
1.4	Organização do trabalho	17
2	REVISÃO BIBLIOGRÁFICA	18
3	COMPARTILHAMENTO DE CONTEÚDO	20
3.1	Tráfego de dados nas redes móveis	20
3.2	Comunicação <i>ad-hoc</i> entre dispositivos	20
3.3	Tecnologias para comunicação	21
4	IMPLEMENTAÇÃO	23
4.1	Menu principal	27
4.2	<i>D2D</i>	28
4.3	Servidor local e externo	32
4.4	Registro de eventos	34
5	ANÁLISE DE DESEMPENHO	35
5.1	Cenários	35
5.2	Experimentos	36
5.3	Processamento dos registros de eventos	37
6	RESULTADOS	39
7	DESAFIOS E LIMITAÇÕES	43
8	CONCLUSÃO	44
	REFERÊNCIAS	45
	APÊNDICES	47
	APÊNDICE A – MATERIAIS ELABORADOS PELO AUTOR	48
A.1	Política de segurança	48

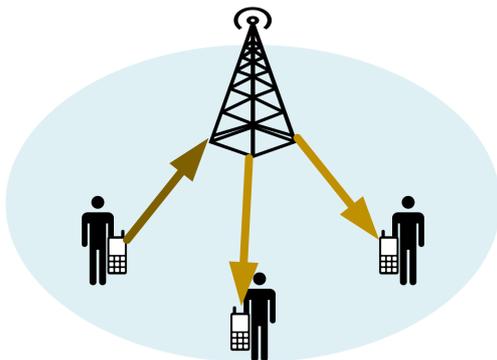
1 Introdução

A comunicação é algo essencial para as pessoas, promovendo a construção do conhecimento através da informação. Com o crescente desenvolvimento tecnológico, essa troca de informações vêm se intensificando cada vez mais, se tornando um imenso desafio aos provedores de serviços de *Internet* para dispositivos móveis.

De fato, espera-se um crescimento no tráfego global de dados móveis para 49 exabytes por mês até 2021 - sete vezes maior em relação a 2016 (CISCO, 2017). Tal tráfego nas redes sem fio cresce exponencialmente devido a demanda por transmissões cada vez mais elevadas, causada por vários fatores, dentre eles a produção de conteúdo e o aumento do número de dispositivos móveis.

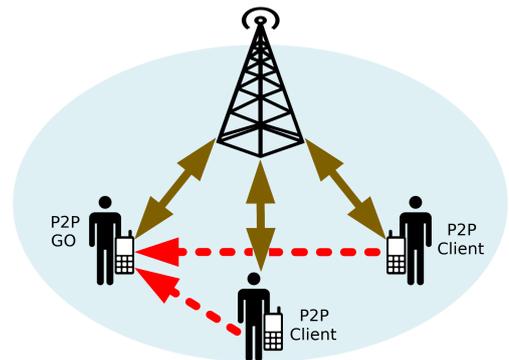
A troca de informações entre dispositivos é uma tarefa rotineira. Tradicionalmente, a transmissão de conteúdo ocorre utilizando a rede infraestruturada, Figura 1. Aplicativos de comunicação móvel, como o popular *Whatsapp*, enviam dados para a nuvem e os reenviam para os *Clients*. Isto acarreta em consumo do plano de dados para todos os envolvidos na transmissão e redundância de transmissão de conteúdo na infraestrutura da operadora.

Figura 1 – Transmissão de conteúdo utilizando rede infraestruturada.



Fonte: Elaborada pelos autores.

Figura 2 – Envio de conteúdo utilizando *Wi-Fi Direct*.



Fonte: Elaborada pelos autores.

A comunicação entre dispositivos próximos pode acontecer diretamente entre eles, Fig. 2. Neste caso, a comunicação direta entre dois dispositivos móveis, chamada comunicação Dispositivo a Dispositivo (*Device-to-device* - D2D), pode ocorrer sem ou com mínima intervenção da Estação Base (*Base Station* - BS) e utilizando espectro licenciado da rede celular ou o não licenciado (ISM) (Asadi; Wang; Mancuso, 2014).

Tecnologias como *Bluetooth* e *Wi-Fi* podem prover a comunicação direta entre aplicações D2D utilizando o espectro não licenciado. Em outubro de 2010, a *Wi-Fi Alliance*

lançou o *Wi-Fi Direct* e o definiu como uma certificação para dispositivos que suportem comunicação *ad-hoc* via *Wi-Fi*.

No *Wi-Fi Direct*, um dispositivo assume o papel de *Group Owner (P2P GO)* e os demais participantes do grupo assumem o papel de *P2P Clients*. Ao mesmo tempo, a conexão com a operadora continua ativa.

1.1 Problema

Poucos trabalhos apresentam uma experimentação que busca mensurar o funcionamento da comunicação *D2D* por meio do *Wi-Fi Direct* em Redes Móveis. Diante desse cenário, fica a dúvida quanto a diversas questões relacionadas ao seu uso em principais aspectos, como: latência da troca de informações e tempo gasto para que um dispositivo encontre o outro e estabeleça uma comunicação. Esses são alguns dos fatores que permitem definir se é mais eficaz utilizar a comunicação *D2D* ou a rede infraestruturada. Dessa maneira, o problema consiste em explorar os limites da comunicação *D2D* utilizando o *Wi-Fi Direct*. Até onde conhecemos, não existe uma ferramenta personalizável para analisar este tipo de comunicação.

1.2 Objetivos

O presente trabalho propõe o *ShareFile*, uma arquitetura para compartilhamento de conteúdo via comunicação *D2D*. O principal objetivo deste trabalho é avaliar o desempenho do *Wi-Fi Direct* em dispositivos já disponíveis no mercado.

Este trabalho possui os seguintes objetivos específicos:

- Propor uma arquitetura para compartilhamento de conteúdo via *Wi-Fi Direct* em dispositivos *Android* nativos;
- Avaliar o desempenho da comunicação *D2D* em um cenário controlado (*testbed*) e no uso real da aplicação por meio da captura e processamento de dados referente as ações do usuário:

Alcance para transferência de dados (Distância possível de se comunicar);

Taxa de transferência (Velocidade da transmissão);

Tempo em que os dispositivos levam para se encontrarem (Tempo de descoberta).

- Discutir as atuais limitações e desafios na implementação da comunicação *D2D* em dispositivos de prateleira.

1.3 Metodologia

O *ShareFile* foi implementado utilizando *Wi-Fi Direct* em dispositivos Android¹, sem a necessidade de modificação no sistema operacional embarcado. As análises foram realizadas em um *testbed* composto de dispositivos móveis, *access points* e servidores da nuvem.

A proposta inicial é permitir a criação de grupos e compartilhamento de conteúdo entre dispositivos próximos. Adicionalmente, o *ShareFile* registra todas as ações executadas: procurar por dispositivos; conectar; enviar/receber arquivos. A partir destes registros podemos analisar o desempenho do *Wi-Fi Direct* na utilização de dispositivos disponíveis no mercado. Com o intuito de realizar comparações com soluções centralizadas, o *ShareFile* permite enviar/receber arquivos de um servidor local e um servidor na nuvem.

Foram analisados os atrasos para formação de grupos, alcance e taxa de transferência em relação ao tamanho dos arquivos e à distância entre os dispositivos.

Os passos para execução deste trabalho são assim definidos:

- Revisão da literatura;
- Implementação do *ShareFile*;
- Captura de dados por meio do *ShareFile* utilizando o *Wi-Fi Direct* e a rede infraestruturada;
- Processamento dos dados;
- Análise e discussão dos resultados.

1.4 Organização do trabalho

O restante deste trabalho é organizado como se segue. O Capítulo 2 discute os trabalhos relacionados. O Capítulo 3 apresenta o tráfego de dados nas redes móveis, a comunicação *ad-hoc* entre dispositivos e as tecnologias para a comunicação. A arquitetura do *ShareFile* para compartilhamento de conteúdo, menu principal, *D2D*, servidor local/externo e os dados registrados são expostos no Capítulo 4. A análise de desempenho é discutida no Capítulo 5, que apresenta os cenários, experimentos e o processamento dos registros gerados. O Capítulo 6 descreve os resultados obtidos. Os desafios e limitações são descritos no Capítulo 7. Por fim, as considerações finais são apresentadas no Capítulo 8.

¹ O aplicativo pode ser instalado em qualquer dispositivo Android versão ≥ 5.0 . Está disponível para *download* na *Google PlayStore* em [<goo.gl/gwEC3d>](http://goo.gl/gwEC3d)

2 Revisão bibliográfica

Este capítulo apresenta uma revisão da literatura, bem como trabalhos correlatos. Embora comunicação *ad-hoc* seja um tema explorado há bastante tempo na indústria e academia, este modelo de comunicação ganhou destaque desde que a *3GPP* anunciou o *Device-to-Device Proximity Service (D2D ProSe)* como parte integrante das futuras redes 5G (*3GPP*, 2013). O *D2D ProSe* utiliza o *Long Term Evolution-Advanced (LTE-A)* como tecnologia de rádio para comunicação *ad-hoc* (comunicação *inband*). No entanto, o *D2D ProSe* também pode explorar o espectro não licenciado, como *Bluetooth*, *Wi-Fi* em modo *ad-hoc* ou *Wi-Fi Direct*, para a comunicação *ad-hoc* entre dispositivos (comunicação *outband*) (Asadi; Wang; Mancuso, 2014).

A avaliação das propostas de algoritmos para seleção de *relay*, formação de grupo e aplicações que utilizem comunicação *D2D* são comumente realizadas por análise teórica ou simulações (LIU et al., 2015). De fato, a viabilidade do uso de comunicação *D2D* para atenuar o tráfego das redes móveis foi estudada por simulação em (Mota et al., 2016). No entanto, ainda existem poucas iniciativas de desenvolvimento e avaliação de protótipos, destacando-se (Whitbeck et al., 2012), (Camps-Mur; Garcia-Saavedra; Serrano, 2013), (Yao; Zhang; Song, 2015), (Amaral et al., 2016) e (Asadi; Mancuso, 2017).

Um arcabouço com estratégias para selecionar nós retransmissores é apresentado em (Whitbeck et al., 2012), aproveitando oportunidades de comunicação *ad-hoc* para minimizar a carga na infra-estrutura sem fio, coletando o reconhecimento de usuários para determinar se novas cópias de informações precisam ser reinjetadas na rede, decidindo quando e para quem um conteúdo deve ser enviado, batizado de *Push-and-Track*. Com base em um conjunto de dados de 10 mil veículos da cidade de Bolonha identificaram que a *Push-and-Track* reduz o uso da rede *3G* em cerca de 90%.

Em (Camps-Mur; Garcia-Saavedra; Serrano, 2013) os autores realizaram uma experimentação da comunicação *D2D* sob o *Wi-Fi Direct* com 2 laptops executando o sistema operacional *Linux*, no qual mensuraram que o atraso para detecção e formação dos grupos é inferior a 5 segundos e notam que a fase de descoberta de vizinhos tem um grande impacto no consumo de energia. Segundo eles a medida que a largura de banda diminui, aumenta o consumo de energia.

Um protocolo de encaminhamento de dados utilizando comunicação *D2D* multi-saltos é proposto em (Yao; Zhang; Song, 2015) utilizando 6 dispositivos móveis executando *Android*. O protocolo forma os grupos baseados na topologia e tolerância a atraso dos dispositivos, que funciona com uma tabela de roteamento, direcionando a mensagem direto ao destinatário caso ele esteja na tabela ou ao dispositivo com maior potencial de movimento

- chance de sair do grupo. Os autores apresentam uma prova de conceito implementando a proposta em dispositivos *Android*, porém, não é apresentada uma avaliação de desempenho.

Uma ferramenta para transmissão de dados oportunística baseada em *Wi-Fi Direct* em dispositivos *Android* batizada *Oi!* é apresentado em (Amaral et al., 2016). Tal ferramenta se baseia na interação social entre as pessoas - mensagens trocadas diretamente por proximidade. Um módulo gerenciador define o sucessor da mensagem com base nos pesos sociais. Os autores focam em apresentar um demo e prova de conceito, sem apresentar avaliações.

A viabilidade do descarregamento de dados móveis por cenários *D2D* é discutida em (Mota et al., 2016) por simulações: *Opportunistic Relaying (OpR)* em que nós oportunistas transmitem mensagens a retransmissores em vez de enviá-las diretamente a infraestrutura; *Cache-and-Forward (CaF)* no qual nós retransmissores encaminham conteúdos recebidos a todos oportunistas encontrados num prazo e *Relay-as-Cache (RaC)* onde nós oportunistas procuram conteúdo no cache dos retransmissores. Utilizaram o *Random* e o *OppLite* (ferramenta que se baseia nas características da rede) para escolha dos nós retransmissores. Em cenários com alta mobilidade e alta tolerância a atrasos os retransmissores podem transmitir até 70% do tráfego de dados no *OpR*, 94% na *CaF* e até 35% na *RaC*.

Uma arquitetura híbrida, no qual a infraestrutura da rede é responsável pela formação de grupos de dispositivos e os dispositivos internos de cada grupo comunicam-se via *Wi-Fi Direct*, é proposta em (Asadi; Mancuso, 2017). Por meio de simulação e protótipo, os autores mostram que grupos pequenos atingem maiores valores de vazão e menores atrasos.

Os trabalhos mencionados apontam para a lacuna ainda existente quanto à necessidade de avaliar comunicação *D2D* em redes móveis, principalmente, em cenários reais. Em comum, os trabalhos supracitados apresentam experimentação em *testbeds* que não podem ser utilizados pelo público em geral.

Este trabalho apresenta uma ferramenta que utiliza comunicação *D2D* em dispositivos de prateleira e pode ser utilizada por qualquer um. Desta forma, foram avaliados cenários controlados da comunicação *D2D* para compartilhamento de conteúdo.

3 Compartilhamento de conteúdo

Nesta seção, apresentamos uma breve descrição do tráfego de dados nas redes móveis e o *ShareFile* - ferramenta que permite o compartilhamento de conteúdo por meio de rede infraestruturada ou via comunicação *D2D*. De fato, a troca de informações utilizando a infraestrutura de rede no *ShareFile* somente foi implementada apenas para fins de comparação, sendo o foco principal da proposta a comunicação *ad-hoc* entre os dispositivos.

3.1 Tráfego de dados nas redes móveis

Segundo o estudo da CISCO (CISCO, 2017), até 2021, a população global terá mais telefones celulares (5,5 bilhões) do que contas bancárias (5,4 bilhões), água canalizada (5,3 bilhões) e telefones fixos (2,9 bilhões). Quase 75% destes dispositivos classificados como inteligentes - aquele que tem conexões móveis que possuem capacidades avançadas de multimídia/computação com no mínimo conectividade 3G - correspondendo a quase 98% do tráfego.

A necessidade de atender essa gama de usuários com rapidez se torna uma imensa exigência de recursos as operadoras de telefonia móvel. A CISCO (CISCO, 2017) estima que a velocidade de conexão da rede móvel aumentará três vezes em 2021, e menciona que em 2021, um *smartphone* irá gerar em média 6,8GB de tráfego por mês, o que segundo eles corresponde a 4,25 vezes o tráfego de dados em um mês no ano de 2016. Ressaltam que a transmissão de vídeo em 2021 tende a ser 9 vezes a de 2016 - 78% do tráfego total de dados móveis entre 2016 e 2021. Isto é explicado pelo tamanho dos arquivos neste formato. Paralelo a isso, o crescimento do número de tablets em 26%, e 8% para os computadores.

A CISCO (CISCO, 2017) afirma que o tráfego de dados móveis cresceu 18 vezes nos últimos 5 anos. Tal crescimento, induz a pensar em soluções alternativas para aliviar o tráfego de dados sem fazer uso ou com a mínima utilização da rede infraestruturada.

3.2 Comunicação *ad-hoc* entre dispositivos

As redes móveis *ad-hoc*, ou *Mobile Ad-hoc Networks (MANETs)*, são paradigmas para a comunicação móvel em que os nós são móveis, sendo assim dinamicamente e arbitrariamente localizados, de tal forma que a comunicação entre os nós não depende de qualquer infraestrutura estática subjacente (FERNANDES, 2012).

A falta de uma infraestrutura faz com que redes *ad-hoc* sejam auto-organizadas

e com alcance de transmissão limitado a capacidade dos nós. Uma estratégia bastante utilizada são os múltiplos saltos (*multi-hop*) (FERNANDES, 2012), em que os nós transmitem a informação a outro(s) nó(s) até que ela chegue ao destino final. Portanto, caso seja possível transmitir a informação diretamente de um nó ao outro, realiza-se um único salto, encaminhando os pacotes do remetente ao destinatário. Isto é possível devido a mobilidade, que permite aos nós se associarem e disassociarem de forma fácil e rápida.

3.3 Tecnologias para comunicação

A comunicação direta entre aplicações *D2D* pode ser realizada por meio de algumas tecnologias, como *Bluetooth* e *Wi-Fi*, que fazem uso do espectro não licenciado.

Dada à ampla adoção do *Wi-Fi* em diversos tipos de dispositivos, uma maneira natural para a tecnologia progredir é a conectividade de dispositivo a dispositivo, isto é, sem exigir a presença de *Access Points (AP)*. O *Wi-Fi Direct*, padrão definido pela *Wi-Fi Alliance*, visa prover uma arquitetura transparente para permitir comunicações diretas entre dispositivos utilizando o *Wi-Fi* do próprio.

No *Wi-Fi Direct*, um dispositivo precisa assumir o papel de líder, chamado *P2P Group Owner (GO)*. Os demais participantes são chamados de *P2P Client*. Para participar da rede e se tornarem *Clients*, os demais dispositivos precisam procurar e encontrar um *Group Owner*. Após isto, os dispositivos estabelecem uma conexão e podem trocar mensagens. A Figura 3 apresenta um diagrama do processo de comunicação do *Wi-Fi Direct*. Em resumo, esta troca funciona da seguinte maneira¹:

- Dispositivo A solicita criação do grupo e recebe notificação informando que o *Wi-Fi* será desativado e ligado em modo *Direct*;
- Dispositivo A concorda com a notificação e é classificado como *GO*;
- Dispositivo B realiza uma busca e encontra um ou mais dispositivos *GO*;
- Dispositivo B seleciona na lista o dispositivo A e requisita conexão, no qual recebe notificação informando que o *Wi-Fi* será desativado e ligado em modo *Direct*;
- Dispositivo B concorda com a notificação, o que faz com que o Dispositivo A receba um convite de B (Deseja se conectar comigo?);
- Dispositivo A aceita o convite, o que faz com que a comunicação seja estabelecida;
- A qualquer momento, um dos dispositivos pode solicitar o cancelamento do grupo. Caso o *GO* saia do grupo, este deixa de existir.

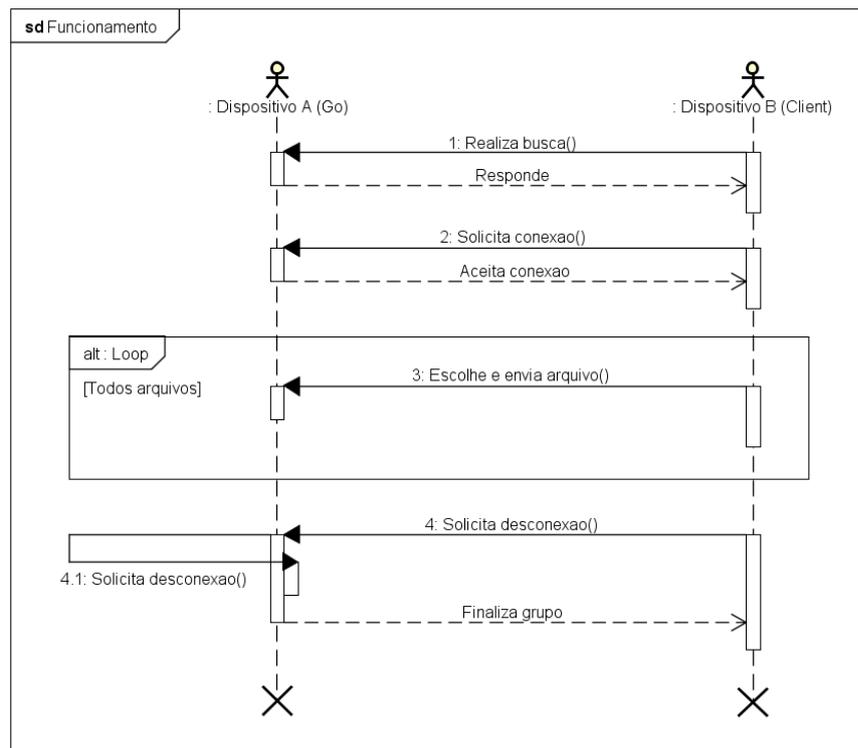


Figura 3 – Fluxo de dados para Comunicação *D2D*.

Fonte: Elaborada pelos autores.

A pesquisa realizada por PPLWARE em (PPLWARE, 2010) compara o *Wi-Fi Direct* e o *Bluetooth*, descrevendo algumas características principais:

- Velocidade: o *Wi-Fi Direct* suporta taxas de transferência de aproximadamente 250Mbps, enquanto o *Bluetooth* cerca de 25 Mbps;
- Consumo energético: Segundo a *Wi-Fi Alliance*, o *Wi-Fi Direct* faz uma melhor gestão da energia - pode resultar em ganho de 15 a 40% de bateria;
- Segurança: o *Wi-Fi Direct* suporta *WPA2* e criptografia *AES* (*Advanced Encryption Standard* - Padrão de Criptografia Avançada) de 256bits, enquanto o *Bluetooth* na sua versão 4.0, suporta criptografia *AES* de 128bits.

Nota-se que o *Wi-Fi Direct* é até 10 vezes mais rápido, com taxas elevadas de transferência (*throughput*), maior alcance e menor consumo de energia que o *Bluetooth*, além de suportar uma criptografia *AES* utilizando chaves com o dobro do tamanho, representando maior segurança nas trocas de informações. Tal superioridade justifica o foco no *Wi-Fi Direct* neste trabalho.

¹ Leitores interessados em detalhes do funcionamento do *Wi-Fi Direct* podem consultar em (Camps-Mur; Garcia-Saavedra; Serrano, 2013).

4 Implementação

Para avaliar o compartilhamento de conteúdo, o *ShareFile* foi implementado para dispositivos móveis com Android (versão ≥ 5.0). Com o objetivo de permitir comparações, além do modo de comunicação *D2D* a ferramenta desenvolvida permite utilizar um servidor central ou compartilhar conteúdo na nuvem.

Para a comunicação *D2D*, o *ShareFile* utiliza *API P2P* do *Android* - implementa o *WifiP2pManager.ChannelListener* e *D2DDeviceActionListener* para lidar com o controle do *D2D* sob o *Wi-Fi Direct*. Por meio dos filtros de intenções detecta alterações no status e estado da conectividade *Wi-Fi P2P* além da lista de pares disponíveis. A *API P2P* com o *WifiP2pManager* gerencia o canal e por meio das funções (*discoverPeers*, *createGroup*, *removeGroup*, *connect*, *cancelConnect*) permite localizar os dispositivos disponíveis, criar ou remover um grupo, conectar ou cancelar uma conexão - no *D2D* cria-se um grupo que é controlado por um dispositivo *GO* e os demais dispositivos (*Clients*) podem se conectar a esse grupo. Todas as funções mencionadas, exigem o tratamento dos casos de sucesso e falha por meio do *onSuccess* e *onFailure*. Para simplificar as análises, esta implementação *D2D* do *ShareFile* é unilateral - apenas dispositivos *Clients* enviam arquivos para o *GO*.

As permissões exigidas pelo *Android* são necessárias para que se possa acessar e modificar o estado do *Wi-Fi*, acessar o servidor externo por meio da internet, salvar os arquivos na memória do dispositivo, ler o estado do dispositivo para detectar uma possível conexão ou tentativa de conexão e fazer *login* por meio do *Google*. São estas:

- ACCESS_WIFI_STATE;
- CHANGE_WIFI_STATE;
- CHANGE_NETWORK_STATE;
- INTERNET;
- ACCESS_NETWORK_STATE;
- READ_PHONE_STATE;
- WRITE_EXTERNAL_STORAGE;
- READ_EXTERNAL_STORAGE;
- MANAGE_DOCUMENTS;
- GET_ACCOUNTS.

A Figura 4 mostra o diagrama de pacotes do *ShareFile*.



Figura 4 – *ShareFile*: Diagrama de pacotes.

Fonte: Elaborada pelos autores.

1. *Activity*: Componente de aplicativo que fornece uma tela ao usuário permitindo fazer algo (API..., 2017).
 - *SplashActivity*: Exibe uma imagem padrão da comunicação *D2D* sob o *Wi-Fi Direct* e simula um carregamento, que ao seu fim redireciona para o menu do *ShareFile*;
 - *MenuActivity*: Controla todo o fluxo da aplicação, verificando se as permissões necessárias para que o *ShareFile* execute estão habilitadas e direcionando para o *D2D*, servidor local/externo, configurações e/ou exibindo aonde são salvos os dados da aplicação;
 - *SearchConnectDisconnectD2DActivity*: Permite encontrar os dispositivos próximos, criar um grupo ou se conectar a um grupo existente, desconectar e/ou mostrar os detalhes do dispositivo;
 - *SendFileD2DandServerLocalActivity*: Solicita encontrar o servidor local (não realizado no *D2D*), possibilita ao usuário escolher uma ou mais imagens, algum vídeo, documento ou até mesmo enviar todos arquivos de um diretório e por fim inicia o serviço de transferência de arquivo;
 - *SendFileServerExternalActivity*: Conecta ao servidor externo, possibilita ao usuário escolher uma ou mais imagens, algum vídeo, documento ou até mesmo enviar todos arquivos de um diretório e por fim inicia o serviço de transferência de arquivo;

- *SettingsActivity*: Possibilita ao usuário alterar as preferências - configurações do *ShareFile*. Este, pode optar por mostrar os arquivos após recebê-los ou apenas indicar que recebeu algo. Também, definir se deseja ou não gerar arquivos de *LOGs*.
2. *Adapter*: Permite personalizar a forma como este componente é apresentado para o usuário (API..., 2017).
 - *PeerListAdapter*: Compõe um lista contendo o nome, endereço e status de cada dispositivo encontrado durante a busca.
 3. *ControlType*: Direciona o fluxo da aplicação.
 - *ControlD2DAndLocal*: Solicita o serviço de transferência para a comunicação *D2D* e/ou servidor local;
 - *ControlType*: Solicita o serviço de transferência de acordo com o tipo da comunicação e tipo de arquivo escolhido. Caso seja *D2D* ou servidor local, repassa a tarefa para o *ControlD2DAndLocal*.
 4. *Entities*: Representa as entidades utilizadas pelo *ShareFile* para armazenar os dados capturados. Tal armazenamento é feito utilizando arquivos de texto.
 - *DataConnectDisconnect*: Dados de conexão e desconexão;
 - *DataSearch*: Dados de busca por dispositivos;
 - *DataSendReceived*: Dados de envio/recebimento de arquivos.
 5. *DAO*: Lida com os arquivos de entrada e saída do *ShareFile* - copiar, ordenar e/ou escrever.
 - *FileCopy*: Copia o arquivo do *InputStream* para o *OutputStream* e vice-versa. Portanto, começa a marcar o tempo, copia o conjunto de registros, para de marcar o tempo e insere os dados no arquivo de *LOGs*;
 - *FileOrder*: Ordena um conjunto de arquivos por nome ou data de modificação utilizando o método de ordenação *QuickSort*;
 - *WriteLOG*: Cria o arquivo de *LOGs* e insere os dados capturados em modo *append* nele - Anexa ao fim do arquivo.
 6. *Fragment*: representa o comportamento ou uma parte da interface do usuário em um *Activity* (API..., 2017).
 - *DeviceFragment*: Fragmento que mostra os dados do dispositivo na conexão e permite conectar/desconectar, e caso esteja conectado, direciona para o envio de arquivo;

- *DeviceFragmentList*: Exibe a lista de fragmentos - pontos encontrados durante a realização da busca.

7. *Connect*: Trata de todo e qualquer tipo de conexão realizada pelo *ShareFile*.

- *D2DDeviceActionListener*: *Interface* da lista de ações do dispositivo - mostrar detalhes, criar grupo, encontrar dispositivos, conectar ou desconectar;
- *D2DAndServerLocalClientServiceTransfer*: Serviço que processa cada solicitação de transferência de arquivo. *Intent* abre um *socket*, conecta, copia o arquivo e por fim fecha o *socket*;
- *D2DGOFileCopyTaskAsync*: Tarefa assíncrona do servidor de arquivos (*GO*). Em *background*, abre um *socket* no *GO* numa porta conhecida pelos *Clients* e fica aguardando conexões. Logo após, cria o arquivo, copia os dados e fecha o *socket*;
- *D2DReceiver*: Permite que o receptor da transmissão envie atualizações para a atividade e acesse o *hardware* do *Wi-Fi*. Portanto, executa a ação de acordo com a intenção recebida:

WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION: Verifica estado do *Wi-Fi* e ativa *P2P*;

WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION: Requisita a lista de pontos ao canal;

WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION: Requisita as informações de conexão ao canal;

WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION: Responde a mudança de estado do *Wi-Fi* -> Indica que conectou.

- *PingServerLocalAndExternal*: Realiza uma série de *PINGs* consecutivos durante um intervalo de 30s e armazena estes dados em um arquivo de *LOGs*;
- *ServerLocalUdpClientThread*: Cria um *socket UDP* na rede local e por meio de uma sinalização em *broadcast* busca o servidor local, que irá responder com uma sinalização conhecida por este *Client*.

8. *Util*: Funções úteis utilizadas por diversas classes.

- *Constants*: Representa todas as constantes utilizadas pelo *ShareFile*;
- *Dialog*: Trata de identificar a linguagem do dispositivo e traduzir o *ShareFile* para inglês ou português;
- *Get*: Retorna o *mimeType*, tamanho, extensão e/ou tipo de um arquivo, status e ou endereço de um dispositivo, verifica se o *Wi-Fi* está disponível e realiza *Pings* na rede.

4.1 Menu principal

O menu principal do aplicativo é mostrado pela Figura 5. Esse, controla todo fluxo da aplicação verificando se as permissões necessárias para que o *ShareFile* funcione estão habilitadas e direcionando para os três modos de comunicação mencionados, além de permitir alterar as configurações e mostrar onde são guardados os arquivos recebidos. A Figura 6 mostra as configurações da aplicação. O usuário pode optar por não gerar os *LOGs* e não mostrar os arquivos após receber (apenas uma mensagem).

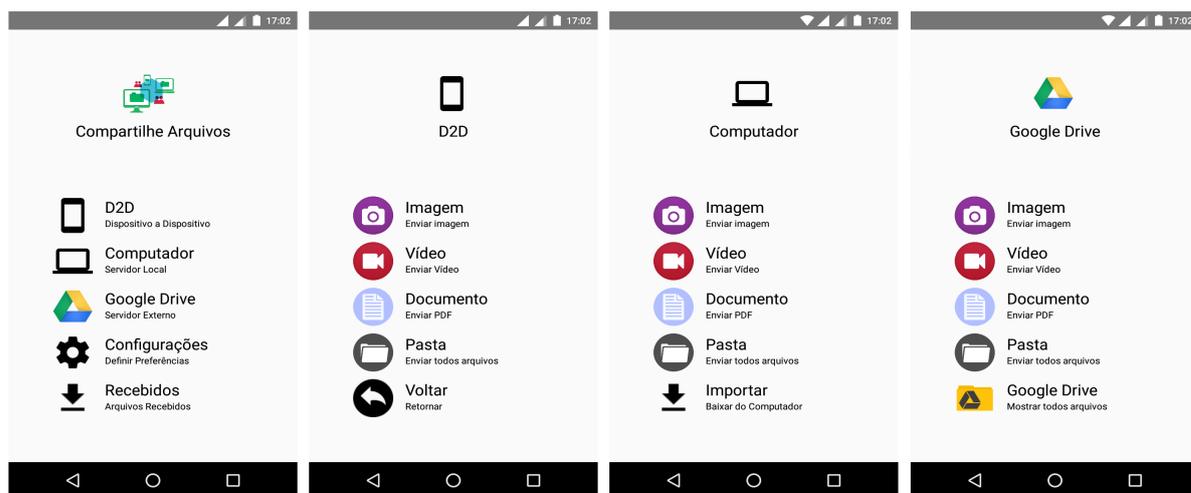


Figura 5 – *ShareFile*: Menu principal e modelos de comunicação.

Fonte: Elaborada pelos autores.



Figura 6 – *ShareFile*: Configurações.

Fonte: Elaborada pelos autores.

4.2 D2D

No *Wi-Fi Direct* um dispositivo precisa assumir o papel de líder, chamado *P2P Group Owner (GO)*. A Figura 7 mostra o diagrama de atividades para a comunicação *D2D* do ponto de vista do *GO* - O usuário deve abrir o aplicativo, selecionar a opção *D2D*, clicar no ícone de busca (lupa) para então criar um grupo de comunicação. Após o grupo ser criado, *Clients* podem se conectar e enviar arquivos para o *GO*. A qualquer momento, o *GO* pode se desconectar, o que faz com que o grupo deixe de existir.

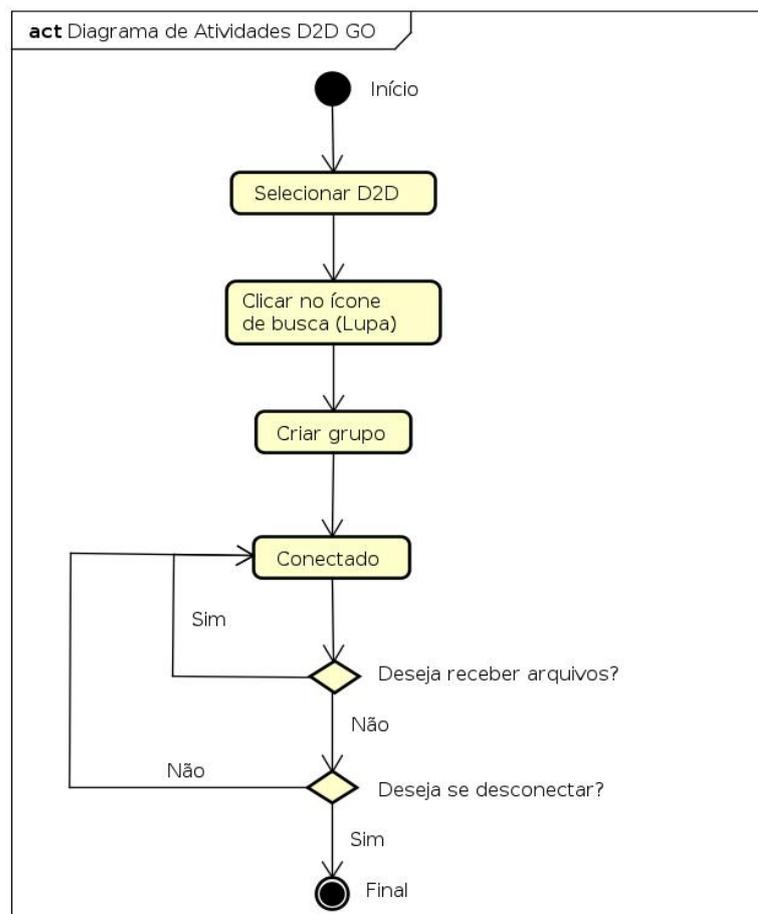


Figura 7 – *ShareFile*: Diagrama de atividades *D2D GO*.

Fonte: Elaborada pelos autores.

A Figura 8 mostra o diagrama de atividades para a comunicação *D2D* do ponto de vista do *Client* - O usuário deve abrir o aplicativo, selecionar a opção *D2D*, clicar no ícone de busca (lupa) para então procurar por um grupo de comunicação. Após encontrar um grupo, ou seja, um dispositivo classificado como *GO*, pode solicitar a conexão a este grupo. Caso o *GO* aceite o pedido de conexão, é possível enviar vários arquivos para o *GO*. A qualquer momento, o *GO* pode se desconectar, o que faz com que o grupo deixe de existir. Também, este dispositivo, ou seja, o *Client* pode se desconectar do grupo.

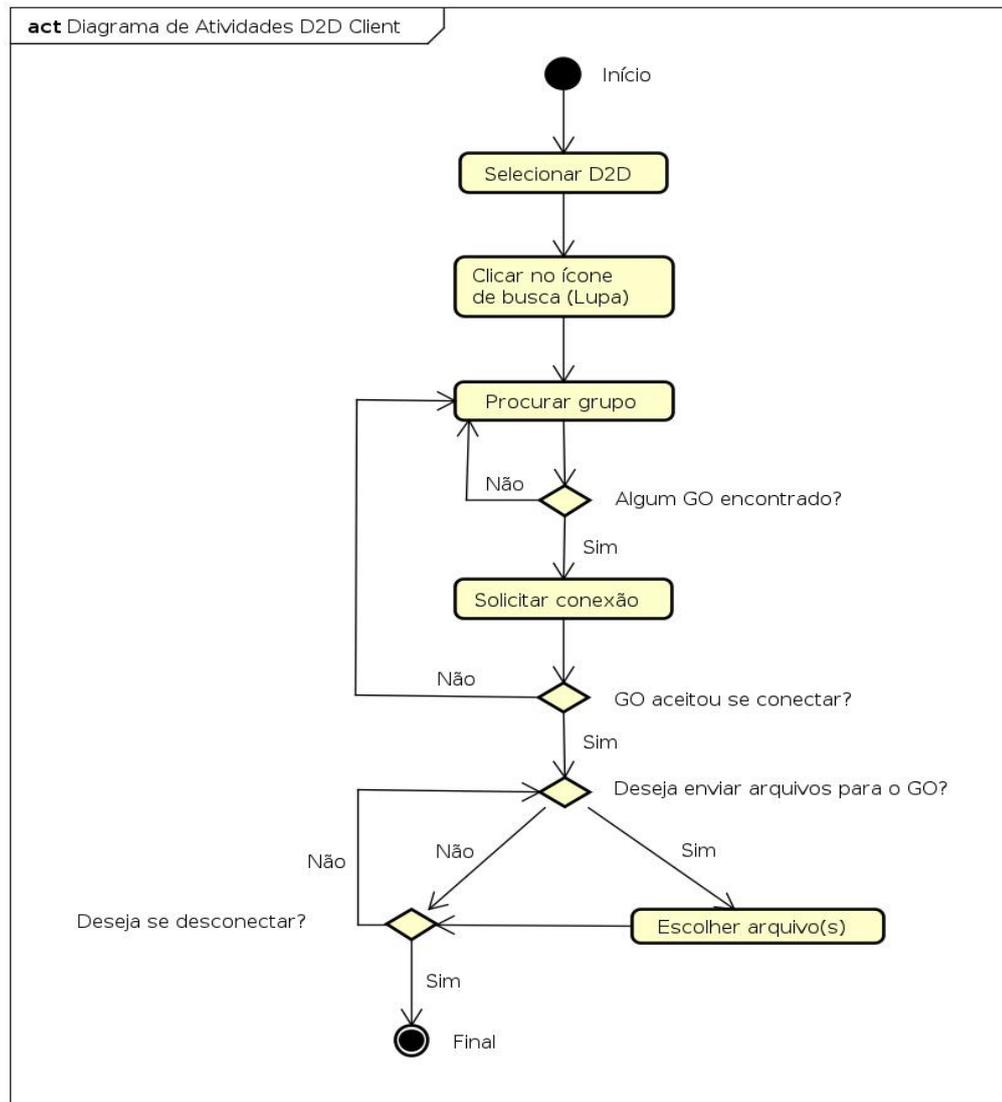


Figura 8 – *ShareFile*: Diagrama de atividades *D2D Client*.

Fonte: Elaborada pelos autores.

A Figura 9 mostra o processo de criação do *GO* no *ShareFile*.

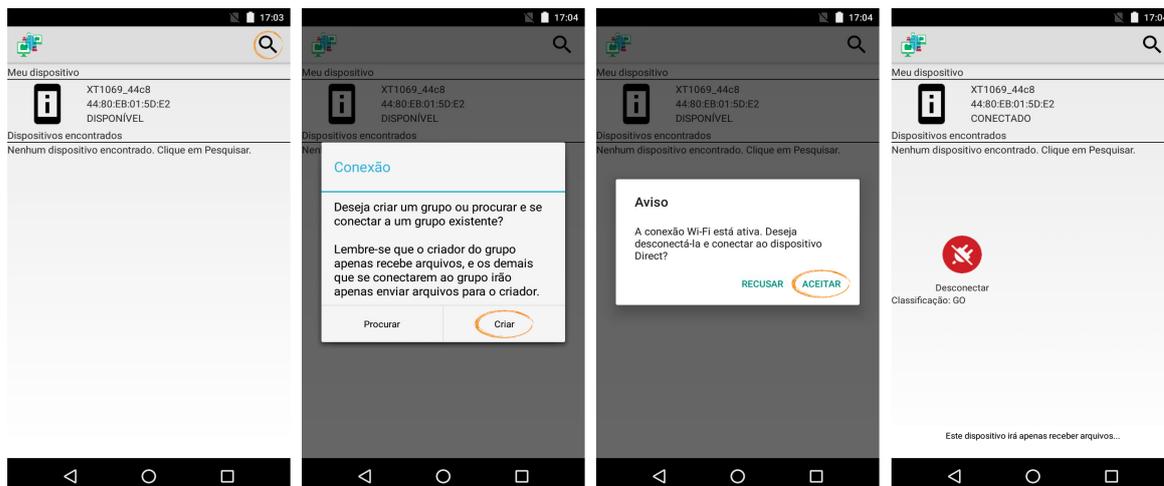


Figura 9 – *ShareFile*: Criando o *GO*.

Fonte: Elaborada pelos autores.

A Figura 10 apresenta a fase de descoberta, em que os dispositivos se encontram. Logo após, um aviso informa que o *Wi-Fi* tradicional será desligado para permitir a formação de grupos via *Wi-Fi Direct* - alguns smartphones não mostram tal informação por apresentarem mais de uma interface de rede, o que faz com que o *Wi-Fi* tradicional não seja desligado e religado em modo *Direct*.

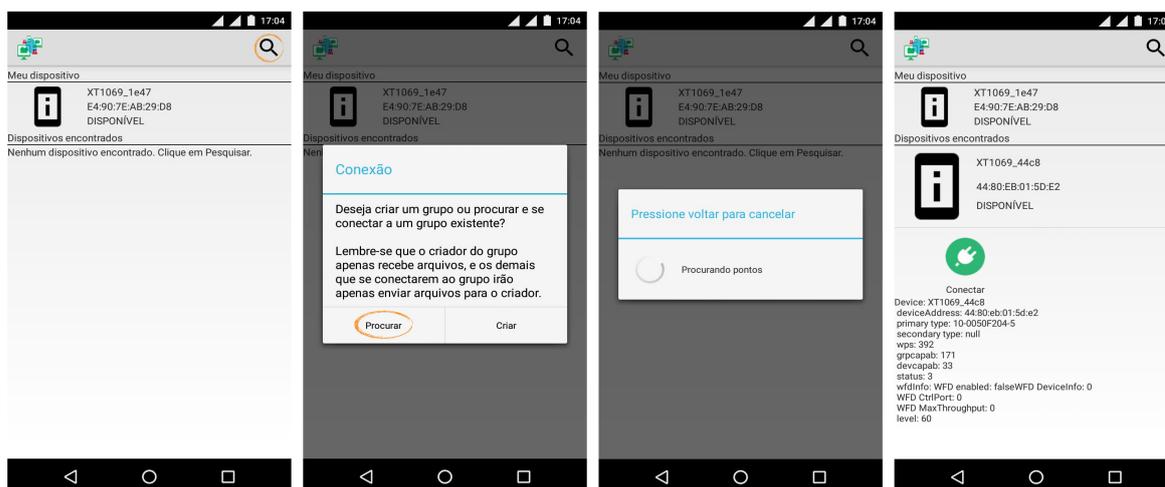


Figura 10 – *ShareFile*: Fase de busca por grupos *Wi-Fi Direct*.

Fonte: Elaborada pelos autores.

Após detectar algum dispositivo *GO*, é preciso se conectar a ele - Figura 11. Neste processo o *GO* recebe uma notificação informando que tem um potencial *Client* solicitando se conectar nele, de modo que o *Client* fica na espera por alguns segundos da recusa/aceite do convite. Caso o convite seja aceito, ambos estão conectados, possibilitando então que o *Client* possa enviar arquivos para o *GO*. A partir deste momento, tem-se estabelecido um meio de comunicação entre o *Client* e o *GO*.

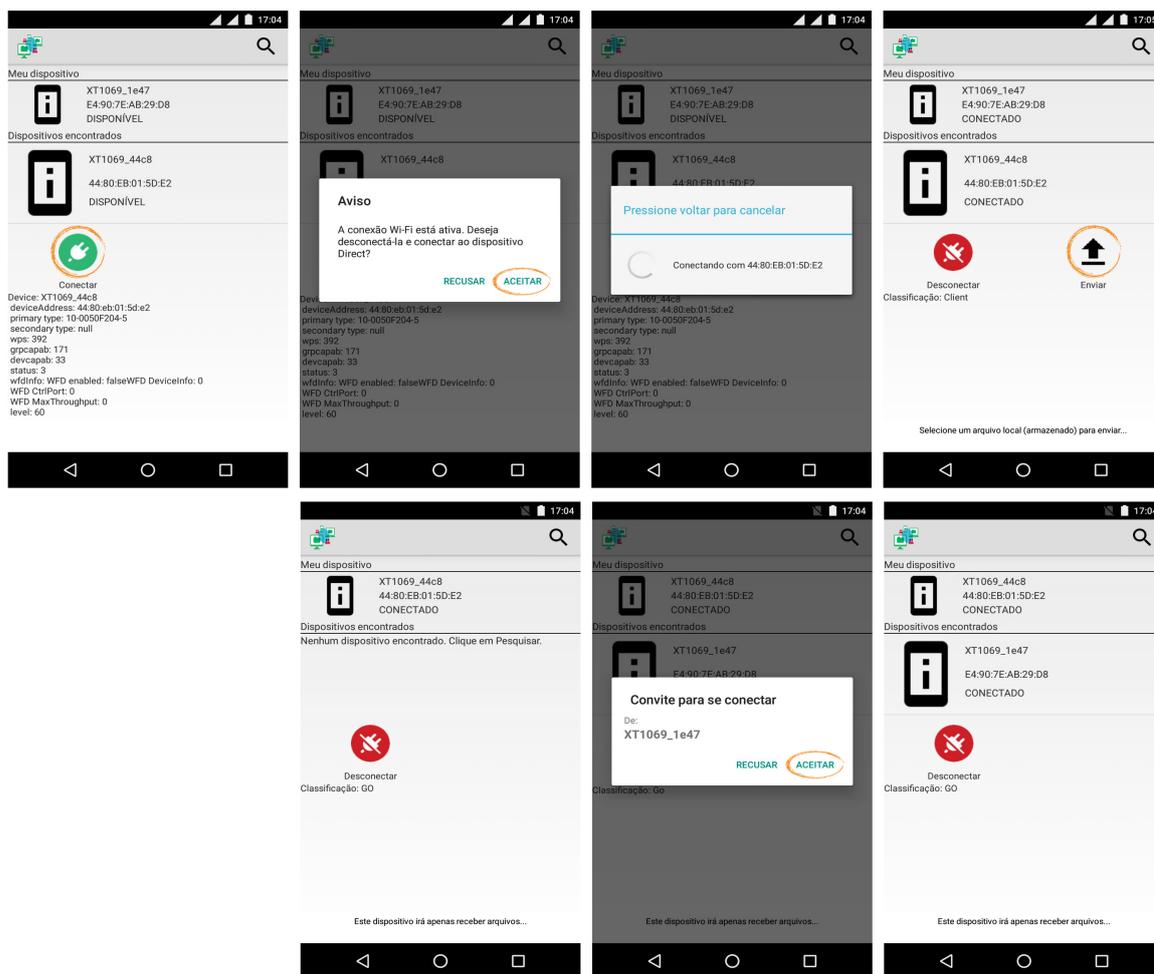


Figura 11 – *ShareFile: Client* se conectando no *GO*.

Fonte: Elaborada pelos autores.

Assim que a conexão é estabelecida os participantes do grupo podem escolher entre compartilhar apenas um arquivo, um conjunto de arquivos ou todo conteúdo de um diretório - Figura 12. Tal compartilhamento funciona da seguinte maneira:

- *Client* seleciona arquivo(s) a ser(em) enviado(s);
- *Client* cria um *socket* com o *GO* - mecanismo de comunicação entre processos dos dois dispositivos;
- *Client* copia os dados do arquivo no *stream* da rede;

- *GO* cria um arquivo e copia os dados do *stream* da rede;
- *GO* fecha o *socket*.

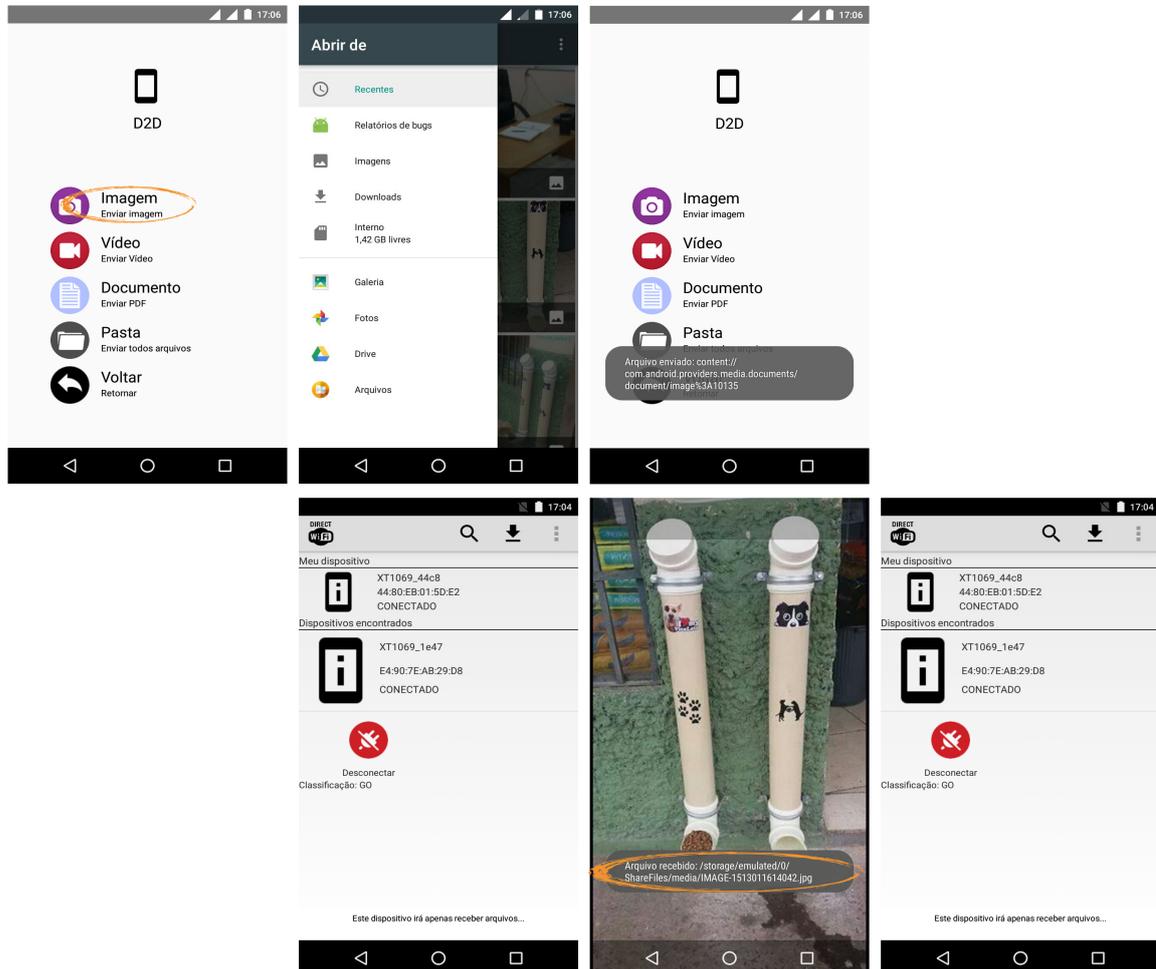


Figura 12 – *ShareFile: Client* enviando arquivo para o *GO*.

Fonte: Elaborada pelos autores.

4.3 Servidor local e externo

Foi implementada uma aplicação que executa no computador/*notebook* para trocar informações com o dispositivo (celular). Tal aplicação, foi desenvolvida em *Java*, e executa um servidor *UDP* que fica responsável por receber o *beacon* enviado em *broadcast* na rede pelo dispositivo (celular) e responder informando o seu endereço *IP* na rede local. Também, executa um servidor *TCP* que fica a cargo de trocar as informações entre o dispositivo (celular) e o computador/*notebook*. Portanto é um servidor *multithread* - suporta conexão de múltiplos *Clients*. Dessa maneira, cria um *socket* numa porta conhecida pelo *ShareFile* e fica em modo *accept*, ou seja, aguardando conexões. Com isso, o *Client* se conecta e envia ou solicita algum arquivo, obtendo uma resposta deste servidor.

A Figura 13 mostra os passos que o usuário deve realizar para trocar informações com este servidor. Para isso, deve abrir o aplicativo, selecionar a opção de servidor local e selecionar quais arquivos deseja enviar para o servidor, ou se deseja receber arquivos do servidor. O mesmo ocorre para o servidor externo, mostrado pela Figura 14.

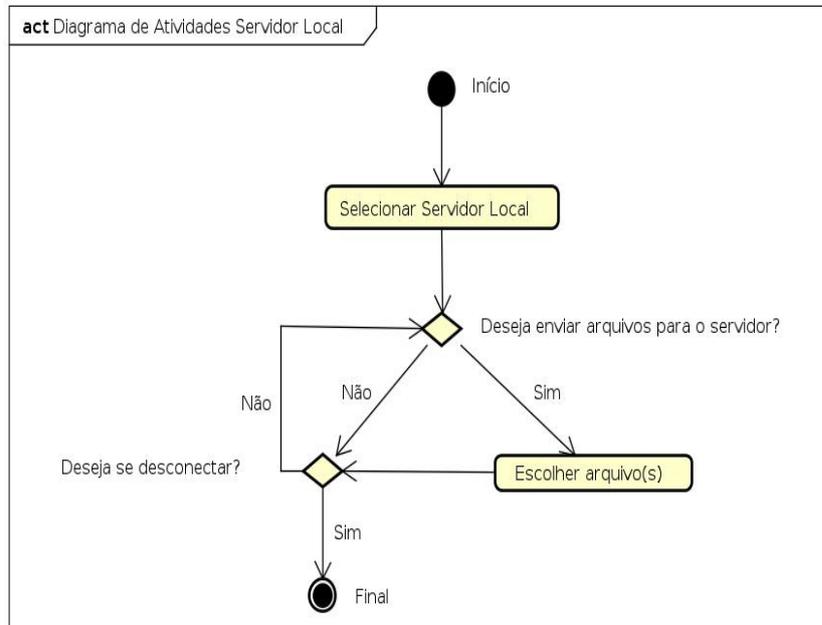


Figura 13 – *ShareFile*: Diagrama de atividades do servidor local.

Fonte: Elaborada pelos autores.

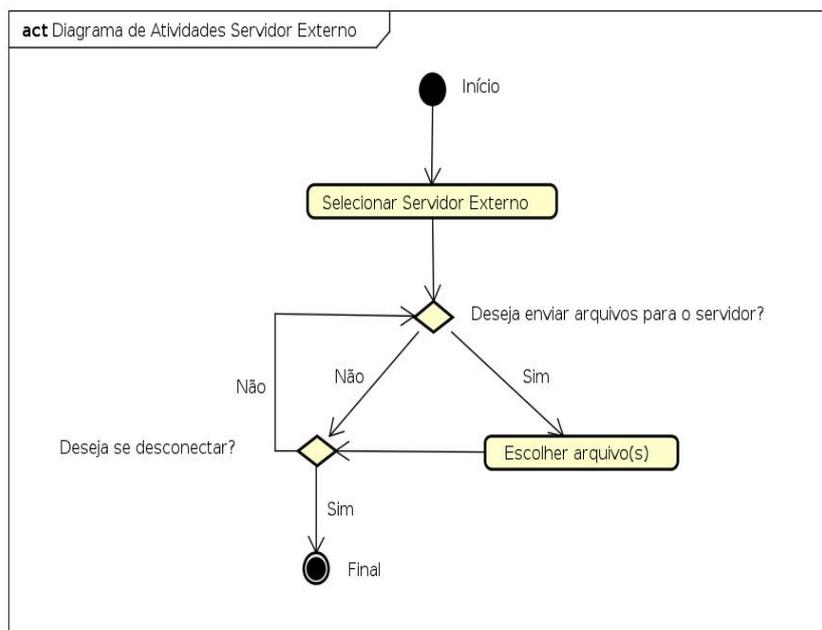


Figura 14 – *ShareFile*: Diagrama de atividades do servidor externo.

Fonte: Elaborada pelos autores.

4.4 Registro de eventos

Todas as operações de busca de dispositivo, conexão, desconexão, envio e recebimento de dados são registradas em arquivos utilizando o formato *Comma Separated Values* (CSV). Os arquivos são transmitidos posteriormente para um servidor para serem analisados. As principais informações salvas em cada ação são:

- Ação - (REQUEST_CONNECT, CONNECT, DISCONNECT, SEARCH, SEND/RECEIVED);
- Tipo do dispositivo - (GO, CLIENT);
- Timestamp inicial do evento;
- Timestamp final do evento;
- Endereço do dispositivo;
- Nome do dispositivo;
- Quantidade de dispositivos encontrados na busca;
- Tamanho do arquivo transmitido.

Portanto, são gerados arquivos de *LOG* que contém o registro das informações principais para análise do funcionamento de tal comunicação. Em essência, é de suma importância identificar quem está realizando a ação (*GO/Client*), qual o momento em que a ação está sendo realizada, ou seja, quando ela começou e terminou, e o que foi feito em tal ação - dispositivo pode estar realizando uma busca, conexão, desconexão ou até mesmo enviando ou recebendo algum arquivo.

5 Análise de desempenho

Um *testbed* foi criado para fazer a análise de desempenho da comunicação *D2D* em dispositivos de prateleira, comparando-a com um servidor local e um servidor na nuvem. Para isto, avaliamos as seguintes métricas:

- Tempo de busca de dispositivos próximos;
- Tempo de estabelecimento de conexão;
- Largura de banda (*upload/download*).

5.1 Cenários

Para permitir a comparação entre a solução distribuída e soluções centralizadas, foram desenvolvidos três modos de transmissão de conteúdo, ilustrado pelas Figuras 15, 16 e 17: Transmissão via comunicação *D2D*, nuvem e rede local respectivamente.

Comunicação *D2D* : No *D2D*, Figura 15, o dispositivo *GO* representa o líder do grupo e vários *Clients* podem conectar-se a ele. Para testar a influência da distância, foram realizados testes entre um par de dispositivos a: [0, 1, 3, 5, 10, 15] metros.

Servidor Local : No servidor local, Figura 16, a aplicação realiza um *broadcast* na rede e encontra o servidor que oferece o serviço de compartilhamento. O dispositivo conecta-se ao servidor para enviar/receber arquivos utilizando a rede local, ou seja, não exige nenhuma conexão com a Internet. Para encontrar o servidor, o dispositivo envia um *beacon* em *broadcast* na rede. Ao recebê-lo, o servidor responde ao *Client*. O *Client* então solicita ao servidor a lista de arquivos disponíveis e realiza o *download* de cada um destes.

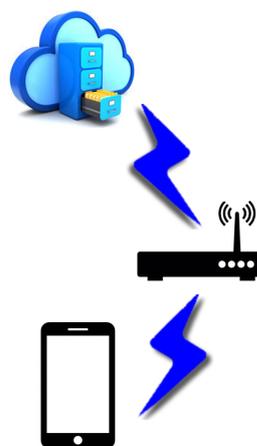
Nuvem : Foi utilizado um servidor externo, Figura 17, que permite enviar/receber arquivos da nuvem. O funcionamento é semelhante ao servidor local. No entanto, este servidor possui IP fixo e está hospedado na Locaweb¹.

¹ Locaweb é um serviço privado de hospedagem na web - <http://www.locaweb.com.br>

Figura 15 – Transmissão de conteúdo via comunicação *D2D*.

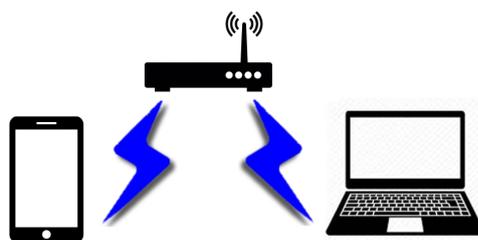
Fonte: Elaborada pelos autores.

Figura 16 – Transmissão de conteúdo via servidor externo.



Fonte: Elaborada pelos autores.

Figura 17 – Transmissão de conteúdo via servidor local.



Fonte: Elaborada pelos autores.

5.2 Experimentos

Os experimentos foram feitos numa cobertura de 15,6m x 6,5m localizada na cidade de João Monlevade. Esses, realizados em um ambiente sem barreiras, em que os dispositivos foram colocados em cima de um banco e espaçados conforme mencionado posteriormente. Nos testes com o servidor local e/ou externo, o *Modem Wi-Fi* ficou a uma distância de 1,5m do *Notebook* e *Smartphone*. Segue a descrição dos *hardwares*:

- 02 *Smartphones Motorola XT1069* 16GB;
- 01 *Notebook Lenovo G400S* Core I3 de Quarta geração com 8GB de *RAM DDR3* (Servidor Local) - *Wireless Atheros Ar5b125*;
- *Modem Wi-Fi ZTE F660*.

Foi utilizado um acesso a *Internet* com velocidade de 15Mbps por meio de enlaces de fibra ótica ofertado pela *Valenet*. Ressalta-se que no momento de realização dos testes,

nenhum outro dispositivo estava conectado a rede local.

Em cada um dos três cenários foram transmitidos um conjunto de 37 arquivos de tamanhos variados (total = 942,3MB), sendo que cada experimento foi repetido 10 vezes. Portanto, foram executados 2200 envios em *D2D* (total = 55,21GB), ou seja, 370 envios para [0, 1, 3, 5, 10, 15] metros. Utilizando o servidor local foram 370 envios (total = 9,2GB) e 370 envios no servidor externo (total = 9,2GB). Totalizando, temos 73,61GB de informações trocadas. A Tabela 1 sumariza os tipos e tamanhos dos arquivos transmitidos.

Arquivos	
Tipo (Extensão)	Tamanhos
Imagem (png)	42.6KB; 59.2KB; 60.8KB; 60.9KB; 68KB; 76.5KB; 79.8KB; 84.5KB; 121.2KB; 828.9KB
Música (mp3)	3.2MB; 4.6MB; 4.7MB; 4.9MB; 5.7MB; 6.6MB; 7.2MB; 7.3MB; 7.6MB; 9.5MB
Documento (pdf)	104.5KB; 314.2KB; 396.2KB; 452.8KB; 560.4KB; 985.4KB; 1MB; 2.3MB; 3.3MB; 6.6MB
Vídeo (mp4)	467.7KB; 589.4KB; 2.1MB; 7.6MB; 7.9MB
Disco (iso)	227MB; 617.8MB

Tabela 1 – Arquivos transmitidos nos experimentos em *testbed*.

Fonte: Elaborada pelos autores.

A implementação do envio de um diretório padrão se deu com fins de automatizar a seleção dos arquivos. Portanto, os 37 arquivos mencionados foram inseridos neste diretório. Assim, para realizar dos testes, após se conectar se exigiu apenas um clique no desenho da pasta, que solicita enviar todos arquivos deste *folder*.

Sempre que um dispositivo tenta se conectar ao servidor local ou externo, é feito automaticamente um teste na rede - envio de *PINGs* para medir a Latência no momento que precede os testes. Neste caso, é registrado o *Round-Trip-Time (RTT)* médio em milissegundos.

5.3 Processamento dos registros de eventos

Nos servidores local e externo, mensura-se apenas a troca de conteúdo. Já em *D2D*, além disso captura-se o tempo em que um dispositivo leva para encontrar o outro e se conectar. Localizar outro dispositivo é o processo desde a solicitação da lista de pontos disponíveis a *API P2P* até o recebimento dessa lista. Conectar com outro dispositivo é o tempo que se gasta para solicitar a conexão até o momento em que a *API P2P* notifica a alteração do status do dispositivo (Disponível para conectado). Foram realizadas 60 conexões, sendo 10 para cada distância.

Cada arquivo de LOG gerado foi submetido a várias etapas de processamento. Para os arquivos importados/exportados dos servidores local/externo, o processamento se deu da mesma forma descrita abaixo, a diferença é que neles o *GO* é substituído pelo servidor e no *Client* também se armazena o *RTT* da rede antes de enviar o arquivo (dado inserido no processamento intermediário e final).

- Processamento inicial: Junção dos arquivos *GO* e *Client* correspondentes, em que captura-se o tempo gasto pelo *Client* para fazer *upload* do arquivo e o *Go* para fazer *download* deste arquivo (10 arquivos do *Client* para cada distância + 10 arquivos do *GO* para cada distância -> geram 10 arquivos para cada distância - mesclam *Client* e *GO*) - Tempo de *upload/download* corresponde ao instante inicial do evento subtraído do instante final.
- Processamento intermediário: Junção dos 10 arquivos processados inicialmente, ou seja, o agrupamento dos tempos gastos pelo *GO* e *Client* nas 10 execuções. Neste, calcula-se uma média de tempo gasto por cada um dos envolvidos. (10 arquivos que mesclam *Client* e *GO* pra cada distância geram apenas 1 arquivo para cada distância).
- Processamento final: Junção dos 6 arquivos gerados no processamento anterior [0, 1, 3, 5, 10, 15] metros em dois arquivos separados em *GO* e *Client* contendo o tempo médio gasto de *upload/download* de cada arquivo.

É de suma importância ressaltar que o tempo utilizado para as análises é sempre o do *Client*, visto que é contabilizado no *Client* desde o momento da criação do *socket*, envio do arquivo até o fechamento do *socket*, enquanto no *GO* apenas se captura o tempo que ele gastou para copiar o arquivo do *stream* da rede.

6 Resultados

No cenário de comunicação *D2D*, primeiramente analisamos o tempo para encontrar dispositivos utilizando o *Wi-Fi Direct* e, após encontrado, o tempo para estabelecer a conexão entre o par de dispositivos. As Figuras 18, 19 e 20 apresentam a *Cumulative Distribution Function* para estas métricas considerando várias distâncias.

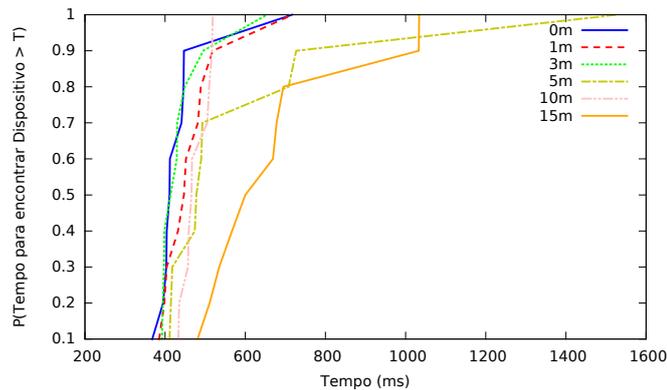


Figura 18 – Tempo para encontrar *GO*.

Fonte: Elaborada pelos autores.

Como mostrado na Figura 18, em 90% dos testes com os dispositivos encostados um ao outro (distância = 0), gastou-se menos de 450ms para o dispositivo *Client* encontrar o *GO*. No outro extremo, com distância de 15m, o *Client* demorou no mínimo 500ms para encontrar o *GO*, sendo que em 90% dos testes este tempo foi inferior a 1050ms. Ao serem separados com uma distância de 10m, o tempo para o *Client* encontrar o *GO* ficou no intervalo de [433,519]ms, média de 477ms.

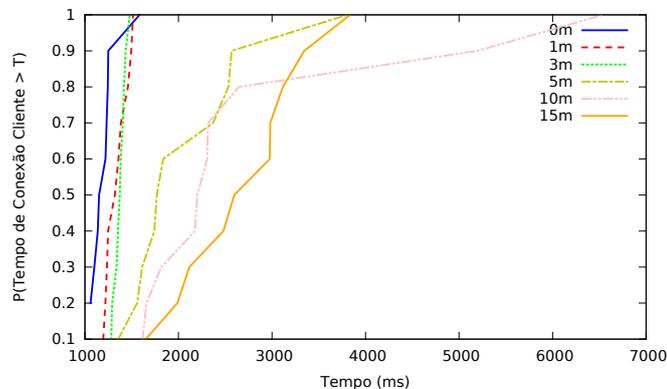


Figura 19 – Tempo estabelecimento de conexão *D2D*.

Fonte: Elaborada pelos autores.

Uma vez que os dispositivos se encontram, podem estabelecer uma conexão. Neste caso, medimos o tempo de conexão do dispositivo *Client* requisitando a conexão para o dispositivo *GO*, Figura 19. Neste teste, o comportamento do aumento do tempo em relação a distância entre os dispositivos se repete. Em distâncias menores que 3m, em todos os testes a conexão demorou menos de 1.5s. O tempo de conexão foi inferior a 2s para 70% dos testes com dispositivos a 5 e 10m de distância. Ao posicionar os dispositivos a 15m um do outro, apenas 20% das conexões demoraram menos de 2s. A média e o desvio padrão para todas as distâncias analisadas são sumarizadas na Tabela 2.

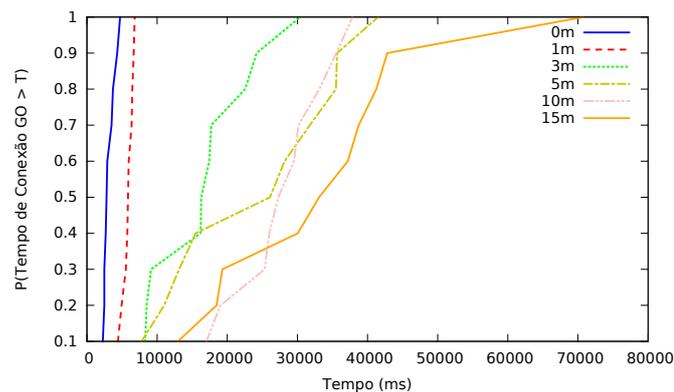


Figura 20 – Tempo conexão *GO*.

Fonte: Elaborada pelos autores.

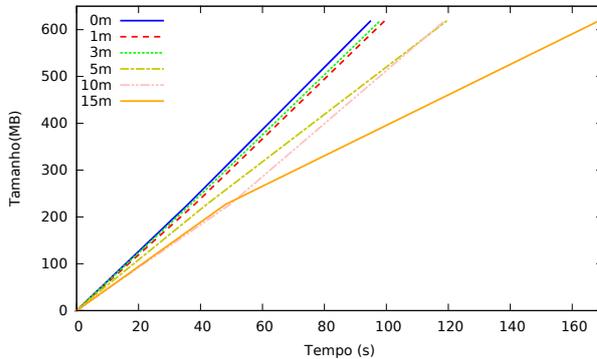
Distância (m)	Média (ms)	Desvio padrão (ms)
0	1203.5	48.1
1	1342.7	37.5
3	1375.3	19.7
5	1375.3	19.7
10	2847	523
15	2708	212
Agregada	1932	131

Tabela 2 – Tempo médio e desvio padrão de conexão entre um par de dispositivos.

Fonte: Elaborada pelos autores.

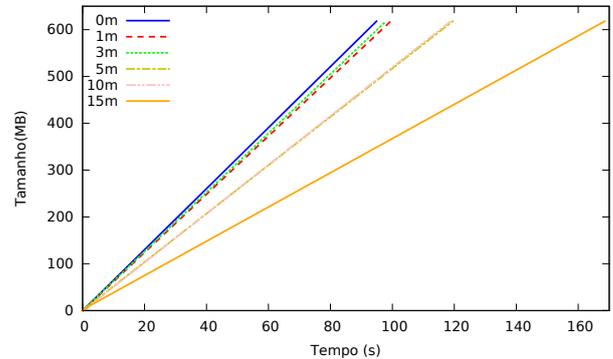
As Figuras 21 e 22 apresentam o tempo de transmissão do conjunto de arquivos com distâncias variadas. Embora a influência da distância seja pequena para arquivos menores que 200MB, para arquivos maiores a diferença pode chegar a mais de 100s, como ocorreu nos arquivos com tamanho superior a 600MB.

Figura 21 – Tempo de transmissão de conteúdo *Client*.



Fonte: Elaborada pelos autores.

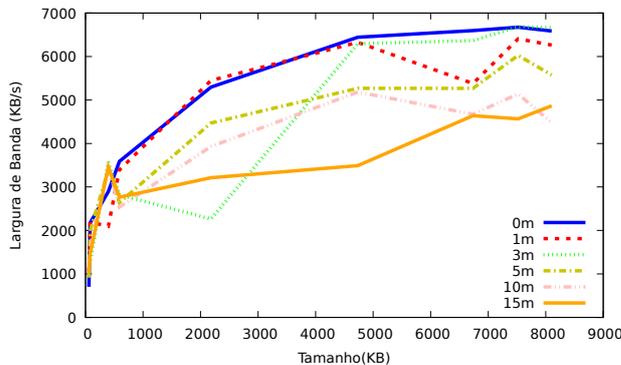
Figura 22 – Tempo de transmissão de conteúdo *GO*.



Fonte: Elaborada pelos autores.

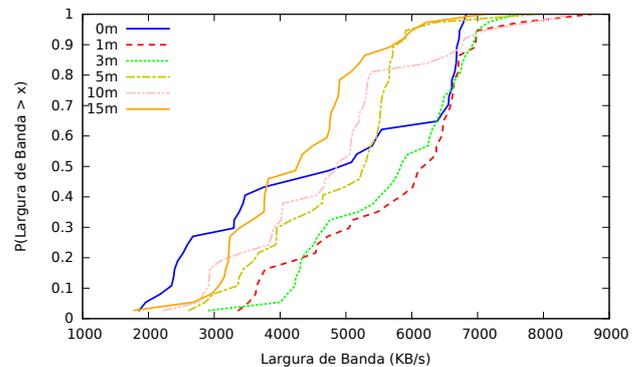
A vazão média entre dispositivos próximos foi de 45600Kbps, com picos de até 69680Kbps. Por outro lado, com distância de 15m, a vazão média foi de 34128Kbps. A Figura 23 apresenta a largura de banda média para cada distância em relação ao tamanho do arquivo. De fato, o tempo de processamento gera um impacto na medição do tempo de transferência em arquivos menores. De modo geral, a taxa de transmissão entre dispositivos foi em média, 40064Kbps, com um desvio padrão de 11584Kbps.

Figura 23 – Largura de banda média *D2D*.



Fonte: Elaborada pelos autores.

Figura 24 – Largura de banda em relação a distância *D2D*.



Fonte: Elaborada pelos autores.

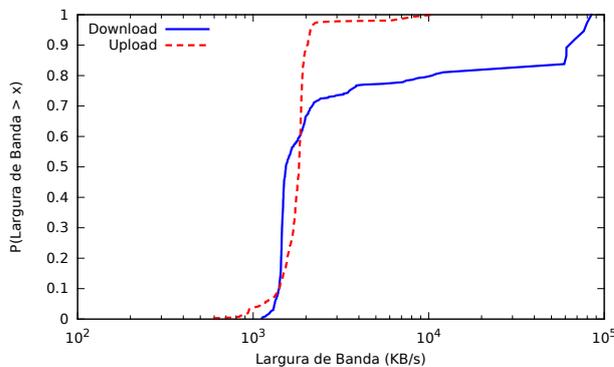
A Figura 24 apresenta a função de distribuição acumulada para a vazão em cada uma das distâncias analisadas. Em 75% dos experimentos, terceiro quartil (Q3), transferências de conteúdo com distâncias até 3m obtiveram até 52728Kbps de taxa de transmissão. Enquanto para distâncias maiores que 5m, o terceiro quartil foi inferior a 47200Kbps, ou seja, 10,4% menor.

Este resultado é fundamental para soluções em que a infra estrutura apenas inicializa a rede *D2D* e a comunicação via *Wi-Fi* em modo *ad-hoc*, como o *Wi-Fi Direct*. Estas

soluções consideram que dispositivos conectados na mesma estação base estão próximos o suficiente para comunicarem-se de modo *ad-hoc*. No entanto, nos dispositivos disponíveis no mercado, a distância entre eles é determinante para a experiência do usuário, como mostram os resultados acima.

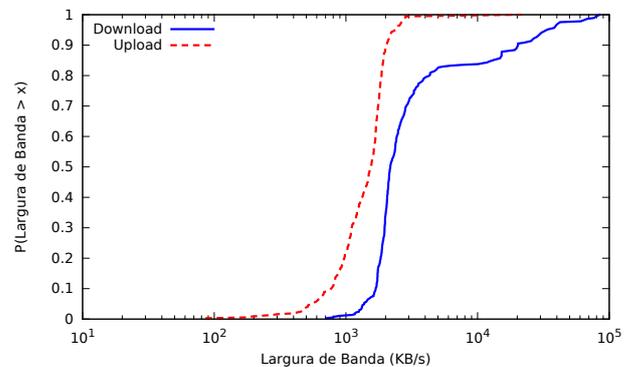
As CDFs da largura de banda nas execuções no servidor local e na nuvem são apresentados nas Figuras 23 e 24. Utilizando um servidor local, Figura 25, a largura de banda foi em média 119200Kbps contra 63200Kbps utilizando o servidor na nuvem, Figura 26. Importante ressaltar que em até 70% dos testes, a vazão de *download* é menor que 20800Kbps em ambos os cenários, o que pode ser explicado pela qualidade de sinal da rede local em que foram realizados os testes.

Figura 25 – Largura de banda servidor local.



Fonte: Elaborada pelos autores.

Figura 26 – Largura de banda servidor na nuvem.



Fonte: Elaborada pelos autores.

Antes da execução de cada teste, o *RTT* é registrado, via *PINGS* consecutivos, por 30s. O *RTT* médio (de todos os testes) na rede local foi de 4.59ms (desvio padrão 16.76) contra 29.28ms do servidor na nuvem (desvio padrão 15.10).

Em média, a taxa de transmissão da rede local e da comunicação em nuvem é 3x e 1,57x a taxa de transmissão via *Wi-Fi Direct*. De fato, a rede local mostra-se mais vantajosa para o compartilhamento de conteúdo. Por outro lado, embora tenha uma taxa de transmissão 57% superior, o compartilhamento de conteúdo utilizando nuvem implica em custos para todos os usuários envolvidos no processo. A economia de recursos e a dispensa de recursos como roteadores, servidores e links de acesso a *Internet* pode compensar uma menor qualidade de experiência causada pelo *Wi-Fi Direct*.

7 Desafios e Limitações

O principal desafio para implementar a comunicação *D2D* em dispositivos de prateleira está relacionado à criação do grupo e ao estabelecimento de conexão. Devido à questões de segurança, os grupos devem ser estabelecidos de maneira manual no *Android*. A criação de grupos de maneira autônoma, base para diversos algoritmos em redes *ad-hoc*, necessita de acesso ao sistema operacional do dispositivo que usuários comuns não possuem.

Trocar informações entre dispositivos (celulares) exige que no dispositivo que irá receber arquivos, neste caso o *GO*, fique executando em *background* uma tarefa assíncrona que cria um *socket* numa porta conhecida pela *Client* e fica aguardando o envio de um ou mais arquivos. Para permitir que se possa receber quaisquer tipo de arquivo, a abordagem utilizada foi o envio do tipo e extensão do arquivo antes mesmo de enviar o seu conjunto de registros.

Além disso, como discutido na seção anterior, o tempo total para estabelecimento da comunicação (busca + conexão) pode ser considerado alto, em especial quando comparado à soluções centralizadas. Entretanto, o tempo médio de conexão do *Wi-Fi Direct* é inferior ao *Bluetooth*, que gasta cerca de 12 segundos na média (Palma; Mancuso; Marsan, 2017). Neste caso, usuários devem suportar o atraso causado por este *overhead* de comunicação.

A vazão obtida pelo *Wi-Fi Direct* também é inferior às soluções centralizadas. Neste caso, a economia na utilização de planos de dados pode compensar o atraso introduzido pela comunicação *D2D*. Interferência e tráfego de *WLANs* vizinhas são a razão para os resultados experimentais (Camps-Mur; Garcia-Saavedra; Serrano, 2013).

Devido as permissões utilizadas no dispositivo *Android*, a *PlayStore* exige o desenvolvimento de uma política de segurança para o aplicativo, visto que são capturados dados, a aplicação tem autoridade sob a rede do dispositivo e pode executar serviços em *background*. Quanto ao uso do *WiFi Direct*, as informações são trocadas diretamente de um dispositivo ao outro e no servidor local, a troca de informações entre o celular e o computador ou *notebook* ocorre usando apenas o *access point*. Portanto, nestes dois tipos, não é possível clonar arquivos e submetê-los a um servidor externo, visto que apenas no servidor externo é utiliza a rede infraestruturada (conexão com a operadora ou *WiFi*) - ato questionado por algumas pessoas preocupadas com a sua segurança.

8 Conclusão

Com o surgimento de novos paradigmas de redes sem fio, a comunicação Dispositivo-a-Dispositivo figura como uma abordagem promissora para permitir o descarregamento de dados da rede celular. Em especial, o compartilhamento de conteúdo entre usuários próximos pode ocorrer diretamente por meio da comunicação *D2D*, sem fazer uso da rede infraestruturada.

Embora existam diversos trabalhos propondo algoritmos, protocolos e alguns *test-beds*, neste trabalho apresentamos uma avaliação do paradigma *D2D* em uma aplicação que pode ser utilizada pelo público geral. Propomos o *ShareFile*, um sistema para compartilhamento de conteúdo que utiliza o *Wi-Fi Direct*, em dispositivos de prateleira e sem modificações no sistema operacional para comunicação *D2D*.

O *ShareFile* pode ser utilizado com um propósito geral de compartilhamento de arquivo via comunicação *D2D* e também, como uma ferramenta para mensurar qualidade deste modo de comunicação. A ferramenta desenvolvida já encontra-se disponível para uso geral (dispositivos com *Android*) e vem sendo preparada para ser uma ferramenta de código aberto.

Por meio de uma análise sistemática, observamos que o tempo para localizar os dispositivos na vizinhança e o tempo de estabelecimento de conexão crescem com a distância, enquanto a vazão diminui. Para o sucesso do paradigma *D2D*, a qualidade de experiência dos usuários está diretamente relacionada com essas métricas. O atraso causado pela busca e o estabelecimento de conexão entre dispositivos pode impactar na qualidade de experiência dos usuários.

Como trabalhos futuros, pretendemos analisar a utilização massiva do *ShareFile*. Deste modo, poderemos avaliar a eficiência da comunicação *D2D* em um conjunto de fabricantes e *hardwares* diversos.

Referências

- 3GPP. *Study on Architecture Enhancements to Support Proximity Services (ProSe) (Release 12)*. [S.l.], 2013. Citado na página 18.
- Amaral, L. et al. Oi! - opportunistic data transmission based on wi-fi direct. In: *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. [S.l.: s.n.], 2016. p. 578–579. Citado 2 vezes nas páginas 18 e 19.
- API Guia do desenvolvedor Android: Conectividade. <<https://developer.android.com/guide/topics/connectivity/wifip2p.html>>, 2017. Citado 2 vezes nas páginas 24 e 25.
- Asadi, A.; Mancuso, V. Network-assisted outband d2d-clustering in 5g cellular networks: Theory and practice. *IEEE Transactions on Mobile Computing*, v. 16, n. 8, p. 2246–2259, Aug 2017. ISSN 1536-1233. Citado 2 vezes nas páginas 18 e 19.
- Asadi, A.; Wang, Q.; Mancuso, V. A survey on device-to-device communication in cellular networks. *IEEE Communications Surveys Tutorials*, v. 16, n. 4, p. 1801–1819, Fourthquarter 2014. ISSN 1553-877X. Citado 2 vezes nas páginas 15 e 18.
- Camps-Mur, D.; Garcia-Saavedra, A.; Serrano, P. Device-to-device communications with wi-fi direct: overview and experimentation. *IEEE Wireless Communications*, IEEE Wireless Communications, v. 20, n. 3, p. 96–104, 2013. Citado 3 vezes nas páginas 18, 22 e 43.
- CISCO. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021*. 2017. <<http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>>. Citado 2 vezes nas páginas 15 e 20.
- FERNANDES, A. S. L. *Comunicação Ad Hoc em Equipas de Robôs Móveis Utilizando a Tecnologia ZigBee*. Dissertação (Mestrado) — Faculdade de Ciências e Tecnologia Universidade de Coimbra, Portugal, 2012. Citado 2 vezes nas páginas 20 e 21.
- LIU, J. et al. Device-to-device communication in lte-advanced networks: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 4, p. 1923–1940, 2015. Citado na página 18.
- Mota, V. F. S. et al. Analisando a capacidade de descarregamento de redes móveis por meio de redes oportunísticas. In: *XXXIV Simpósio Brasileiro de Redes de Computadores (SBRC 2016)*. Salvador: Universidade Federal da Bahia, 2016: [s.n.], 2016. p. 1039–1052. Citado 2 vezes nas páginas 18 e 19.
- Palma, N.; Mancuso, V.; Marsan, M. On the persistence of wireless advertising without infrastructure support. *9th Annual IMDEA Networks Workshop*, Madrid, Spain, June 2017. Citado na página 43.
- PPLWARE. *Bluetooth 4.0 vs Wi-Fi Direct – Quem vencerá?* 2010. Disponível em: <<https://pplware.sapo.pt/tutoriais/networking/bluetooth-4-0-vs-wi-fi-direct-%E2%80%93-93-quem-vencera/>>. Citado na página 22.

Whitbeck, J. et al. Push-and-Track: Saving Infrastructure Bandwidth Through Opportunistic Forwarding. *ArXiv e-prints*, jan. 2012. Citado na página 18.

Yao, C.; Zhang, H.; Song, L. Demo: Wifi multihop: Implementing device-to-device local area networks by android smartphones. In: *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. New York, NY, USA: ACM, 2015. (MobiHoc '15), p. 405–406. ISBN 978-1-4503-3489-1. Disponível em: <<http://doi.acm.org/10.1145/2746285.2764931>>. Citado na página 18.

Apêndices

APÊNDICE A – Materiais elaborados pelo autor

A.1 Política de segurança

Política de segurança exigida pela *PlayStore* no ato da publicação do *ShareFile*. Devido as permissões utilizadas no dispositivo *Android*, a *PlayStore* exige o desenvolvimento de uma política de segurança para o aplicativo, visto que são capturados dados e a aplicação tem autoridade sob a rede do dispositivo.

Política de Privacidade

O aplicativo ShareFiles criado pelo desenvolvedor Daniel Reis é totalmente gratuito. Este serviço é fornecido sem nenhum custo e destina-se a todos usuários na linguagem português e inglês. Tal aplicativo permite o compartilhamento de conteúdo - Envie arquivos de um dispositivo pro outro pelo WiFi Direct, ou para o seu computador pela rede local. Ainda mais, você pode enviar seus arquivos diretamente para o seu Google Drive, mas lembre-se que para enviar arquivos para nuvem exige-se um WiFi ou conexão com operadora.

Este documento destina-se a informar os nossos usuários e potenciais usuários com relação às políticas de coleta, uso e divulgação de informações pessoais de que utiliza os serviços oferecidos por este aplicativo. Portanto, se você optar por usá-lo, você concorda com a coleta e uso de informações em relação a esta política. Tais informações pessoais só são capturadas com o intuito de mensurar o funcionamento deste tipo de comunicação, de modo que apenas grava em arquivo txt no seu dispositivo. É importante ressaltar que a qualquer momento estes podem parar de serem gravados, basta uma alteração nas configurações deste aplicativo.

Coleta e uso de informações

Para mensurar o funcionamento do nosso serviço, é realizada a captura de algumas informações de acordo com a ação do usuário por meio de LOGs. É de suma importância ressaltar que nada é copiado dos arquivos enviados por este aplicativo, mas sim dados a respeito dele. Tais ações, são basicamente a de conexão/desconexão, busca por outros dispositivos, envio/recebimento de algum arquivo e o RTT (Round Trip Time), que nada mais é um teste de latência da rede. Dessa maneira, os dados capturados são o momento em que ocorreu (timestamp), endereço de rede, modelo do dispositivo e qual o tipo de conexão fornecido pelo aplicativo foi utilizado. Com relação aos arquivos, armazena-se apenas o momento em que ele começou e terminou de ser copiado (timestamp), além do seu tamanho e extensão. Quanto a integração com a nuvem, é utilizado serviços de terceiros

(Google Drive) que pode coletar informações usadas para identificá-lo, disponível em <<https://support.google.com/drive/answer/2450387?hl=pt-BR&vid=0-461521047174-1513045213229>>.

Serviços do Google Play

Quero informá-lo de que, sempre que usar o meu Serviço, em caso de erro no aplicativo, é coletado dados e informações (através de produtos de terceiros) em seu telefone chamado Log Data. Esses dados do registro podem incluir informações como o endereço do dispositivo IP (dispositivo IP), o nome do dispositivo, a versão do sistema operacional, a configuração do aplicativo quando utilizar o meu Serviço, a hora e a data do uso do serviço e outras estatísticas .

Segurança

Eu valorizo sua confiança em nos fornecer suas informações pessoais, portanto estamos nos esforçando para usar meios comercialmente aceitáveis para protegê-la. Mas lembre-se de que nenhum método de transmissão pela internet, ou método de armazenamento eletrônico é 100% seguro e confiável, e não posso garantir sua segurança absoluta.

Links para outros sites

Este serviço pode conter links para outros sites. Observe que esses sites externos não são operados por mim. Portanto, eu recomendo vivamente que você analise a Política de Privacidade desses sites. Não tenho controle e não posso assumir nenhuma responsabilidade pelo conteúdo, políticas de privacidade ou práticas de sites ou serviços de terceiros.

Alterações nesta Política de Privacidade

Posso atualizar a Política de Privacidade de tempos em tempos. Assim, te aconselho revisar esta página periodicamente para qualquer alteração. Vou avisá-lo de quaisquer alterações publicando a nova Política de Privacidade aqui. Essas alterações são efetivas imediatamente após serem publicadas.

Contate-nos

Se você tiver dúvidas ou sugestões sobre minha Política de Privacidade, não hesite em contactar-me.