



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

MARCOS LAELBER JACÓ OLIVEIRA

**DESENVOLVIMENTO DE UM APLICATIVO DE TROCA DE MENSAGENS ENTRE
DISPOSITIVOS CELULARES DE MANEIRA TOTALMENTE *OFFLINE***

RUSSAS

2021

MARCOS LAELBER JACÓ OLIVEIRA

DESENVOLVIMENTO DE UM APLICATIVO DE TROCA DE MENSAGENS ENTRE
DISPOSITIVOS CELULARES DE MANEIRA TOTALMENTE *OFFLINE*

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus de Russas da Universidade Federal
do Ceará, como requisito à obtenção do grau de
bacharel em Engenharia de Software.

Orientadora: Profa. Dra. Jacilane de Ho-
landa Rabelo

RUSSAS

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- O48d Oliveira, Marcos Laelber Jacó.
Desenvolvimento de um aplicativo de troca de mensagens entre dispositivos celulares de maneira totalmente offline / Marcos Laelber Jacó Oliveira. – 2021.
34 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2021.
Orientação: Prof. Dr. Jacilane de Holanda Rabelo.
1. Nearby. 2. Connections. 3. Comunicação. 4. Off-line. 5. Ad hoc On-demand Distance Vector (AODV). I. Título.

CDD 005.1

MARCOS LAELBER JACÓ OLIVEIRA

DESENVOLVIMENTO DE UM APLICATIVO DE TROCA DE MENSAGENS ENTRE
DISPOSITIVOS CELULARES DE MANEIRA TOTALMENTE *OFFLINE*

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus de Russas da Universidade Federal
do Ceará, como requisito à obtenção do grau de
bacharel em Engenharia de Software.

Aprovada em:

BANCA EXAMINADORA

Profa. Dra. Jacilane de Holanda
Rabelo (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Pablo Luiz Braga Soares
Universidade Federal do Ceará (UFC)

Prof. Dr. Rafael Fernandes Ivo
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

À Deus e a todos que estiveram comigo nessa caminhada.

RESUMO

A comunicação entre pessoas através de dispositivos móveis é muito comum atualmente. Entretanto, ainda existem alguns lugares ou situações onde a comunicação não é possível pela falta de internet. No entanto, podem ser formadas redes de comunicação offline através de estruturas ad hoc, MANET e Nearby Connections. Contudo, essas estruturas apresentam alguns desafios, como a necessidade de equipamentos para auxiliar o envio de mensagens ou que o receptor esteja dentro do raio de transmissão do emissor. Tendo em vista o baixo raio de alcance nessas formas de comunicação, criou-se um aplicativo de troca de mensagens off-line de forma que o receptor não necessita obrigatoriamente estar dentro do raio de transmissão do emissor. Para isso foram realizadas pesquisas de caráter exploratório a fim de encontrar protocolos, estrutura de redes, APIs e algoritmos de roteamento que auxiliassem na resolução do problema. Além disso, uma comparação entre os protocolos identificados foi realizada. Os resultados mostraram que a API Nearby Connections fornece serviços de conexão e troca de dados off-line entre dispositivos. Desenvolveu-se o aplicativo com o auxílio do Nearby Connections e testou-se em ambiente real. Obteve-se um aplicativo que conseguiu trocar mensagens de maneira off-line que permite saltos entre os dispositivos. Conclui-se que o aplicativo é uma boa alternativa para ambientes onde necessitam de comunicação off-line. Contudo, podem ser realizados estudos a fim de garantir uma maior segurança.

Palavras-chave: Nearby. Connections. Comunicação. Off-line. Mensagens.

ABSTRACT

Communication between people via mobile devices is very common nowadays. However, there are still some places or situations where communication is not possible due to lack of Internet. Nonetheless, offline communication networks can be formed through ad hoc structures, MANET, and Nearby Connections. Although, these structures present some challenges, such as the need for equipment to help send messages or for the receiver to be within the transmission radius of the sender. Considering the low range of these forms of communication, an application was created to exchange offline messages in a way that the receiver does not necessarily need to be within the transmission radius of the sender. For this, exploratory research was carried out in order to find protocols, network structure, APIs and routing algorithms that could help solve the problem. In addition, a comparison between the identified protocols was performed. The results showed that APINearby Connections provides services for connecting and exchanging data offline between devices. The application was developed with the help of Nearby Connections and tested in a real environment. An application was obtained that was able to exchange messages in an offline manner that allows hopping between devices. We conclude that the application is a good alternative for environments that require offline communication. However, studies can be conducted in order to ensure greater security.

Keywords: Nearby. Connections. Communication. Off-line. Messaging.

LISTA DE FIGURAS

Figura 1 – Comunicação em redes <i>ad hoc</i>	13
Figura 2 – Etapas da pesquisa	15
Figura 3 – Enviando mensagens com <i>Nearby connections</i>	18
Figura 4 – Criação de rota usando o AODV	20
Figura 5 – Divulgação da informação de roteamento no DYMO.	21
Figura 6 – Etapas para estabelecer conexão	26
Figura 7 – Distância entre os dispositivos	28
Figura 8 – Tempo médio em diferentes cenários	29
Figura 9 – Saltos entre os dispositivos	30
Figura 10 – Envio de mensagens de A para B realizando saltos	31
Figura 11 – Todos os dispositivos enviando mensagens simultâneas	32

LISTA DE QUADROS

Quadro 1 – Comparação dos Protocolos	25
Quadro 2 – Tempo médio em diferentes cenários	30

LISTA DE ALGORITMOS

Algoritmo 1 – Conexão Iniciada	27
Algoritmo 2 – Calcular tempo médio para as mensagens	29

LISTA DE ABREVIATURAS E SIGLAS

AODV	<i>Ad hoc On-demand Distance Vector</i>
API	<i>Application Programming Interface</i>
CEDAR	<i>Core Extraction Distributed Ad Hoc Routing</i>
CGSR	<i>Cluster Gateway Switch Routing</i>
DSDV	<i>Destination-Sequenced Distance Vector routing</i>
DSR	<i>Dynamic Source Routing</i>
DYMO	<i>Dynamic MANET On-demand</i>
MANET	<i>Mobile Ad hoc Network</i>
OLSR	<i>Optimized Link State Routing</i>
P2P	<i>Peer-to-peer</i>
RREP	<i>Route Reply</i>
RREQ	<i>Route Requests</i>
VANET	<i>Vehicular Ad Hoc Networks</i>

SUMÁRIO

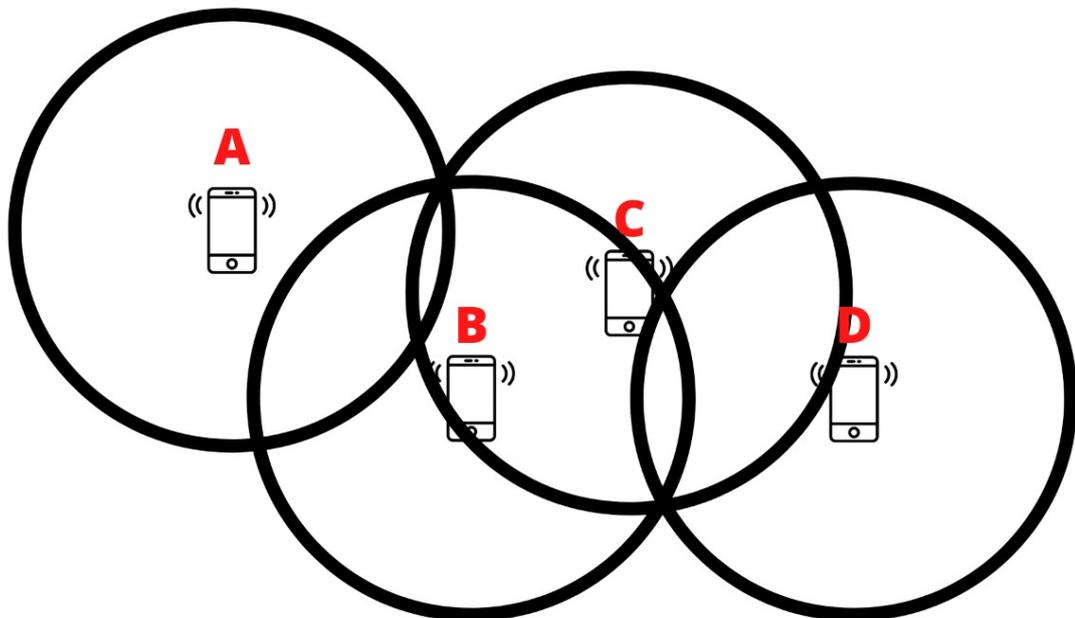
1	INTRODUÇÃO	13
1.1	Objetivos	15
<i>1.1.1</i>	<i>Objetivo geral</i>	<i>15</i>
<i>1.1.2</i>	<i>Objetivos específicos</i>	<i>15</i>
1.2	Metodologia	15
<i>1.2.1</i>	<i>Realizar revisão da literatura</i>	<i>16</i>
<i>1.2.2</i>	<i>Pesquisar protocolos similares ao problema</i>	<i>16</i>
<i>1.2.3</i>	<i>Comparar protocolos identificados</i>	<i>16</i>
<i>1.2.4</i>	<i>Compreender o Nearby Connections</i>	<i>16</i>
<i>1.2.5</i>	<i>Adaptar e implementar o protocolo junto ao Nearby connections para o desenvolvimento do aplicativo</i>	<i>16</i>
<i>1.2.6</i>	<i>Testar o uso do aplicativo</i>	<i>17</i>
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Nearby connections	18
2.2	Redes ad hoc móvel	19
2.3	Protocolos de roteamento	19
<i>2.3.1</i>	<i>Protocolo Ad hoc On-demand Distance Vector (AODV)</i>	<i>19</i>
<i>2.3.2</i>	<i>Protocolo Dynamic MANET On-demand (DYMO)</i>	<i>20</i>
3	TRABALHOS RELACIONADOS	22
3.1	Redes ad hoc	22
3.2	Redes VANET	23
3.3	Protocolos de roteamento	23
4	TESTES E RESULTADOS	24
4.1	Comparação dos protocolos	24
4.2	Adaptação e implementação do protocolo junto ao Nearby connections para o desenvolvimento do aplicativo	25
4.3	Realização de testes	27
<i>4.3.1</i>	<i>Teste de distância</i>	<i>27</i>
<i>4.3.2</i>	<i>Teste de tempo</i>	<i>28</i>
<i>4.3.3</i>	<i>Teste do Número de saltos</i>	<i>30</i>

5	CONCLUSÃO	33
	REFERÊNCIAS	34

1 INTRODUÇÃO

Uma *Mobile Ad-Hoc Network* (*Mobile Ad hoc Network* (MANET)) é um grupo interconectado de dispositivos portáteis tais como notebooks, telefones celulares entre outros que trocam informações sem uma infraestrutura fixa e definida. Ao contrário do que ocorre em redes convencionais, tal como uma rede de telefonia celular em que a comunicação obrigatoriamente passa pela central, ainda que os dispositivos móveis encontrem-se a uma distância que possibilitaria a comunicação direta (OMARI; DAHOU, 2016). Os dispositivos individuais também chamados de nós, ganham capacidade para se comunicar diretamente com outros nós na rede. Portanto, pode-se criar uma rede MANET usando tecnologias sem fio, como *Bluetooth*, *Wi-fi* dentre outras. Em redes *ad hoc* de múltiplos saltos, dois nós que não podem ser alcançados diretamente através do seu raio de transmissão, conseguem ainda se comunicar, desde que tenha uma cadeia de outros dispositivos que possam ser alcançados por ambos (REY *et al.*, 2014). Na Figura 1, os círculos representam a sua área de transmissão e recepção de dados. Logo, em um exemplo de mensagem enviada de A para D, ela deverá passar pelos nós B e C para que possa chegar até o nó D.

Figura 1 – Comunicação em redes *ad hoc*



Em Redes MANETs as mobilidades dos nós e uma topologia dinâmica de rede contribuem para deixar a construção de protocolos de roteamento mais complexa. Omari e Dahou (2016) enfatizam que a presença da mobilidade dificulta a projeção e implementação de um MANET na vida real. É um grande desafio projetar controle de topologia, roteamento, descoberta de serviços, operações e gerenciamento de rede e serviços de segurança para MANET, visto que os métodos para as redes tradicionais não são aplicáveis a esse problema (LEME *et al.*, 2016).

Devido o surgimento de novas tecnologias, as redes MANET estão se tornando parte integrante das redes da próxima geração. Esse fato deve-se a capacidade de configuração automática, falta de infraestrutura, flexibilidade, facilidade de manutenção e economia. Contudo, pode-se notar na literatura diversas aplicações com MANET, tais como, áreas militares, comunicações veiculares, ajuda em desastres e redes tolerantes a atrasos (PAPASTERGIOU *et al.*, 2009) e (SHARMA *et al.*, 2019).

Dito isto, uma rede de comunicação de maneira *offline* para usuários, pode ser de importância, quando necessita-se trocar informações com pessoas próximas e não encontra-se com *Wi-fi* e comunicação via rede telefônica. Por exemplo, quando os funcionários estão em uma empresa e falta energia elétrica, ou até mesmo quando estão localizados em áreas onde a infraestrutura da internet é impossibilitada. No estado da arte pode-se notar trabalho que propõe solução para a troca de mensagens quando a infraestrutura da internet é impossibilitada, como por exemplo o trabalho proposto por Zhang (2011) que usa redes *ad hoc*.

Diante deste contexto, este trabalho visou a criação de um aplicativo de troca de mensagens de maneira totalmente *offline*. O desenvolvimento deste aplicativo ocorreu através do entendimento da *Application Programming Interface (API) Nearby Connections*, que foi desenvolvida pela *Google* com o intuito de haver troca de mensagens entre dispositivos próximos, e da seleção e adaptação de um protocolo de roteamento para a construção da comunicação do aplicativo de maneira eficaz. O aplicativo foi testado e verificou-se a sua eficiência, avaliou-se os parâmetros de tempo de envio e recebimento de dados, distância entre os dispositivos e a quantidade de dispositivos na rede.

Um usuário do aplicativo, conseguiu enviar e receber mensagens sem que o celular de ambos estivesse conectado à internet. Desde que o emissor e o receptor estivessem geograficamente próximos para que ocorra a transmissão de múltiplos saltos, ou seja, foram necessários outros celulares com este aplicativo para que eles pudessem servir de pontes e a mensagem

passasse por eles até chegar ao destino.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver um aplicativo de comunicação entre celulares que troque mensagens de maneira totalmente *offline*.

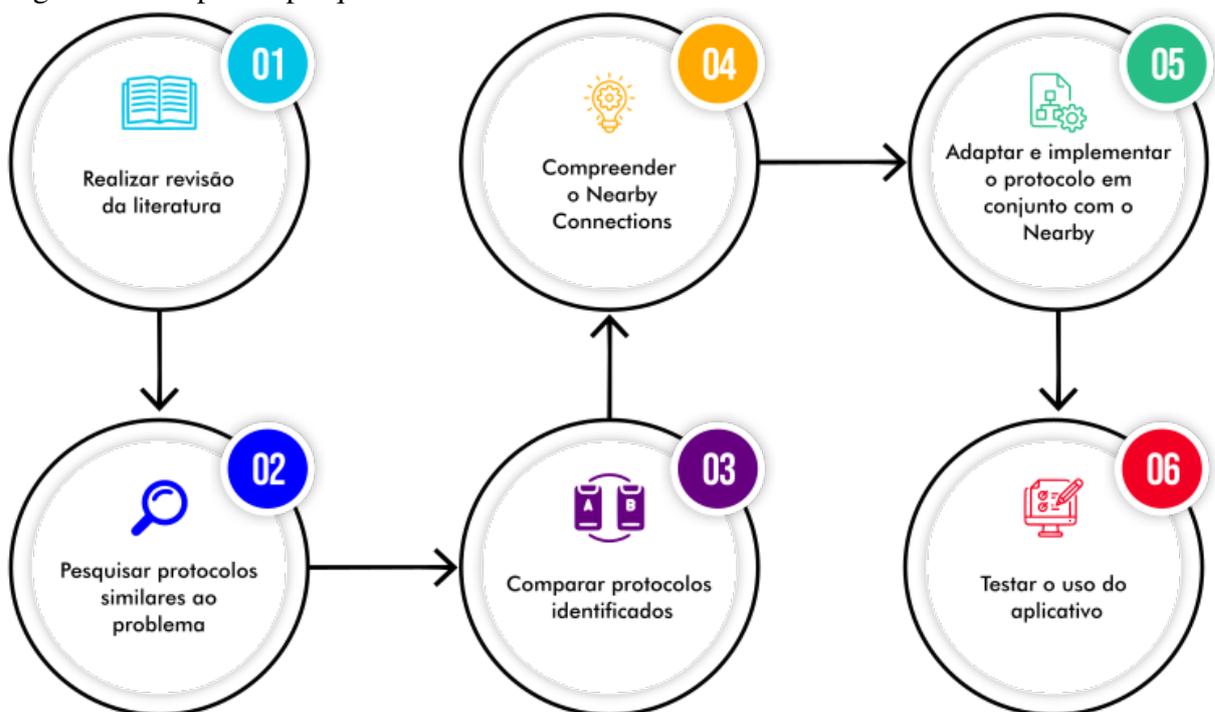
1.1.2 Objetivos específicos

- Comparar o desempenho dos protocolos de roteamento
- Adaptar protocolo de roteamento em conjunto com o *Nearby Connection*
- Testar o alcance da troca de mensagem *Peer-to-peer* (P2P) do aplicativo.

1.2 Metodologia

Nesta seção são apresentados os procedimentos metodológicos que foram seguidos neste trabalho. A pesquisa foi conduzida em 6 etapas, conforme a Figura 2.

Figura 2 – Etapas da pesquisa



1.2.1 Realizar revisão da literatura

Adotou-se a metodologia de pesquisa exploratória para levantamento dos conceitos de redes ao qual a proposta deste trabalho está inserida, como: redes Ad hoc, MANET e Protocolos de roteamento. A pesquisa por esses termos foi realizada em livros e em acervos virtuais da comunidade acadêmica, como: Google Acadêmico, IEEE Explorer, ACM Digital Library entre outros.

1.2.2 Pesquisar protocolos similares ao problema

Foram realizadas buscas por protocolos semelhantes ao problema, dando ênfase a protocolos usados em redes Ad hoc, pois a idealização do projeto visou um aplicativo cuja sua estrutura de rede é baseada na topologia Ad hoc.

1.2.3 Comparar protocolos identificados

Esta etapa consistiu em comparar os protocolos identificados buscando encontrar o protocolo que melhor fosse utilizado em conjunto com a API Nearby Connections, visando os critérios de comparação de latência, reativo e tolerância à mobilidade da rede.

1.2.4 Compreender o Nearby Connections

Nesta fase realizou-se estudo detalhado almejando compreender as fases do funcionamento do *Nearby connections*, bem como sua estrutura, recursos utilizados, desempenho, o processo de anúncio e descoberta dos dispositivos, visando o seu uso na utilização do desenvolvimento do aplicativo.

1.2.5 Adaptar e implementar o protocolo junto ao Nearby connections para o desenvolvimento do aplicativo

Esta etapa consistiu, mediante os protocolos selecionados examinou-se a melhor forma de adaptá-los, visto que o modo como o Nearby connections comportou-se, criando sempre um novo código a cada nova conexão que foi usado como identificador do dispositivo. Contudo, mesmo com essa peculiaridade, na literatura não existem trabalhos que tenham protocolos específicos para atuar em conjunto com o Nearby connections. A implementação do aplicativo

foi na linguagem Java usando ambiente de desenvolvimento integrado (IDE) Android Studio, uma vez que esta IDE facilita a integração e manipulação do Nearby connections, além de terem o Google como desenvolvedor em comum.

1.2.6 Testar o uso do aplicativo

Foram realizados testes de forma real com o objetivo de verificar e validar o funcionamento do aplicativo, como o tempo que a mensagem leva para sair do emissor e chegar ao receptor, a distância entre eles e os saltos que a mensagem realiza.

O restante desse trabalho contém 4 seções e está organizado da seguinte forma. A seção 2 apresenta a Fundamentação Teórica. Na seção 3 serão apresentados os Trabalhos Relacionados. Já na seção 4 serão apresentados os Testes e Resultados. E por fim, na seção 5 será apresentado a Conclusão e os Trabalhos Futuros.

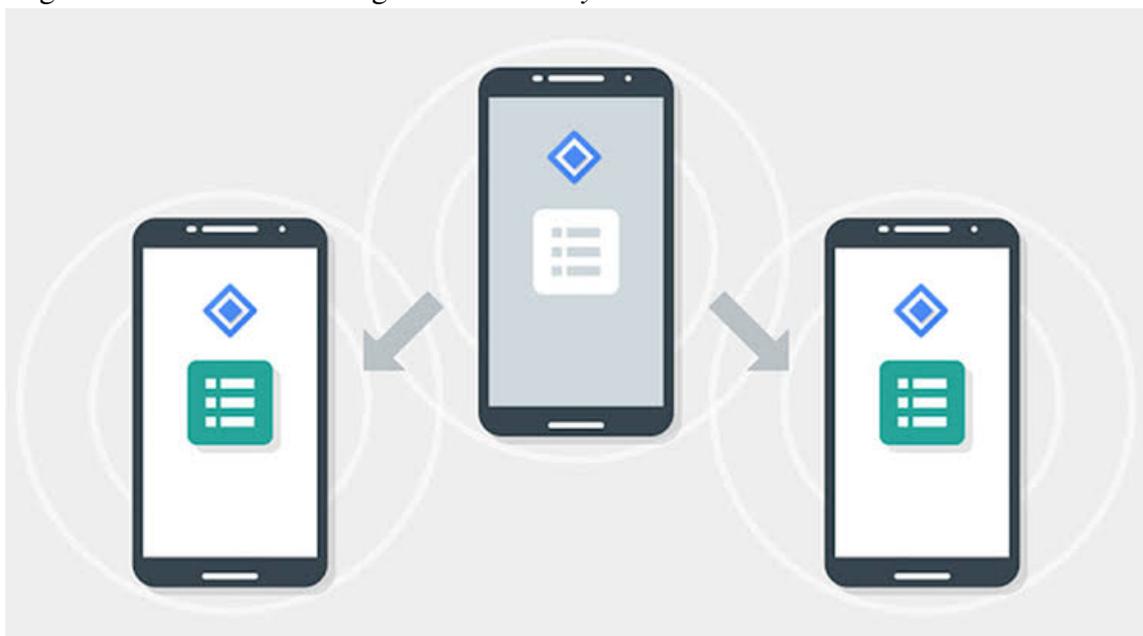
2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os conceitos fundamentais deste trabalho. A primeira (subseção 2.1) mostra a principal ferramenta usada para o desenvolvimento do aplicativo. Em seguida, a subseção 2.2 apresenta a área onde este trabalho se insere. Nas subseções restantes (2.3 e 2.4), os protocolos de roteamento mais utilizados na topologia de rede deste trabalho.

2.1 *Nearby connections*

Nearby connections é uma aplicação desenvolvida pela *Google* que usa o *wi-fi* e o *Bluetooth* para anunciar, descobrir e estabelecer conexões próximas com dispositivos próximos. Essa aplicação abstrai toda a complexidade existente na transferência de dados via rádio, conseguindo assim transferir dados criptografados de baixa latência e alta largura de banda entre esses dispositivos de uma maneira P2P totalmente *offline*. P2P é uma arquitetura de redes de computadores onde cada um dos pontos ou nós da rede funciona tanto como cliente quanto como servidor (MONTRESOR; JELASITY, 2009). A forma como o *Nearby Connections* funciona é semelhante a uma conexão do Unix que pode ser usada para transferir bytes, arquivos ou fluxos de dados. Na Figura 3 que é mostrada logo em seguida, os círculos brancos representam a área de transferência dos dados. Logo, o celular que está no centro, pode transferir mensagens para os outros dois celulares (MEFTAH *et al.*, 2019).

Figura 3 – Enviando mensagens com *Nearby connections*.



Fonte: (TECHTUDO, 2006)

2.2 Redes *ad hoc* móvel

Segundo Geetha *et al.* (2006), uma MANET é normalmente criada de maneira espontânea. Consiste em nós móveis que se comunicam através da mídia sem fio, sem nenhuma infraestrutura fixa. Em uma MANET não existe um nó central para o qual convergem todas as informações vindas dos outros nós, eliminando, assim, a necessidade de um dispositivo roteador central que faça a comunicação da rede com outros destinos, logo, o seu funcionamento consiste em vários nós móveis autônomos que se comunicam através de links sem fio. Entretanto, há um limite no alcance da transmissão sem fio, cada *host* precisa da ajuda de *hosts* próximos para encaminhar o pacote, assim cada *host* atua como um roteador para si mesmo.

2.3 Protocolos de roteamento

Protocolos de roteamento são usados para determinar como é realizado o encaminhamento de pacotes dentro de um sistema autônomo. Protocolos de roteamento também são denominados como protocolos de roteadores internos (KUROSE; ROSS, 2006).

2.3.1 Protocolo AODV

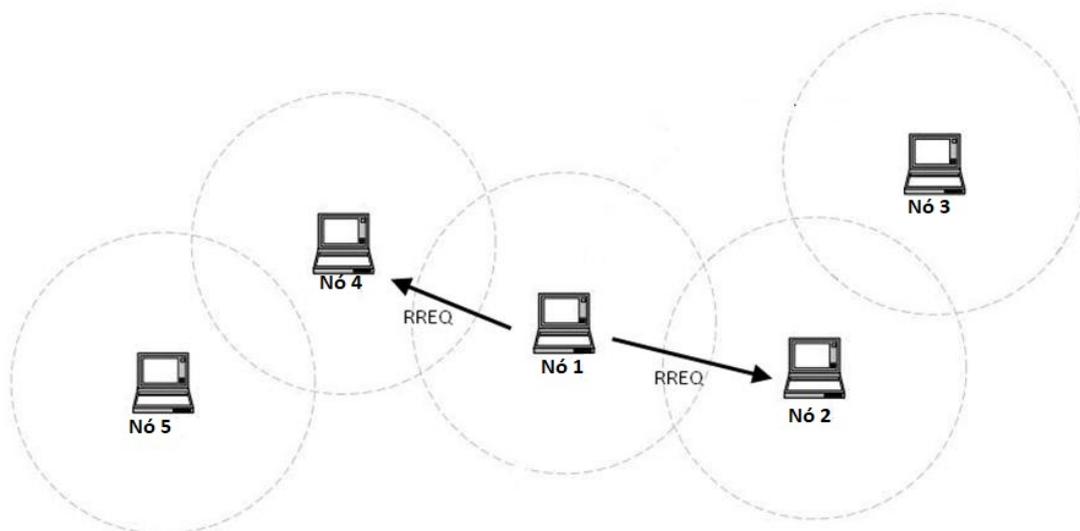
O protocolo de roteamento *Ad-hoc On Demand Distance Vector* (AODV) foi projetado para uso em MANET. O mesmo pode realizar tanto roteamento *unicast*, ou seja, um endereçamento para um pacote feito a um único destino, quanto roteamento *multicast* que é a transmissão de informação para múltiplos destinatários simultaneamente. Por sua característica de agir por demanda, é considerado um protocolo reativo (PEREIRA; MORAES, 2010).

Quando um nó de origem solicita uma rota, este protocolo cria rotas dinamicamente e as mantém, desde que a origem necessite delas. De acordo com Kumar *et al.* (2014) o processo de descoberta de rota do AODV ocorre da seguinte forma, o nó de origem transmite a mensagem *Route Requests* (RREQ) para todos os nós vizinhos para que possa encontrar a rota até o destino. Sempre que um nó intermediário recebe a mensagem RREQ, ele cria uma rota reversa em direção ao nó de origem que envia a mensagem RREQ para esse nó. Ele verifica a tabela de entrada da rota para pesquisar a rota até o destino. Se a entrada for encontrada, ela enviará a mensagem *Route Reply* (RREP) para o nó de origem usando a rota reversa; caso contrário, encaminhará a mensagem RREQ para os próximos nós vizinhos, um por um. Se o nó intermediário for o próprio nó de destino, ele enviará uma mensagem RREP para o nó de origem ao longo da rota

reversa. Após o recebimento da mensagem RREP, todos os nós atualizam a entrada da rota em direção ao destino em suas tabelas de roteamento e formarão uma rota para o destino. Quando a mensagem RREP é acessada no nó de origem, ela identifica o RREP que contém a menor contagem de saltos, seleciona como rota principal. O processo de descoberta é encerrado e as novas informações de rota são atualizadas na tabela de roteamento pela qual a transmissão pode ser feita.

A Figura 4 mostrada logo abaixo, ilustra uma rede *ad hoc* usando o AODV, o nó 1 pretende enviar mensagem ao nó 3. O círculo em volta de cada dispositivo significa o alcance máximo de sua comunicação. Os nós 2 e 4 são os vizinhos do nó 1, portanto o nó 1 envia uma RREQ para eles solicitando um caminho de roteamento até o nó 3. Quando a mensagem de RREQ chega aos nós 2 e 4, eles analisam o pedido de roteamento e verificam se existe uma rota para o destino solicitado pelo nó 1. Neste exemplo, como o nó 3 é vizinho do nó 2, este envia uma mensagem RREP para o nó 1 informando que o destino é o seu vizinho.

Figura 4 – Criação de rota usando o AODV



Fonte: Santos (2006).

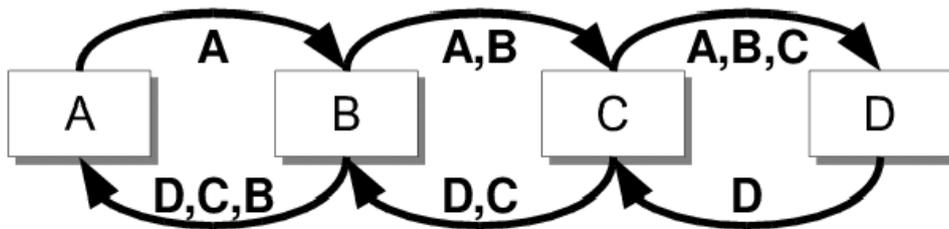
2.3.2 Protocolo DYMO

De acordo com Miao e Xu (2010), o *Dynamic MANET On-demand* (DYMO) é um protocolo de roteamento desenvolvido para ser usado em redes ad hoc sem fio, é um protocolo reativo, pois age sob demanda. Oferece uma adaptação na mudança da topologia e determina

rotas *unicast* entre os nós de origem e os nós de destino. Considerado o sucessor do AODV, possui operações básicas como a descoberta e o gerenciamento de rotas. Durante a descoberta de rota, o nó de origem inicia a propagação de uma RREQ por toda a rede para encontrar o nó de destino. Durante esse processo de propagação, cada nó intermediário registra uma rota para o nó de origem. Quando o nó de destino recebe o RREQ, ele responde com um *unicast* de resposta de rota RREP em direção ao nó de origem. Cada nó que recebe o RREP registra uma rota para o nó de destino e, em seguida, o RREP é *unicast* em direção ao nó de origem. Quando o nó de origem recebe o RREP, isso significa que as rotas foram estabelecidas entre o nó de origem e o nó de destino nas duas direções. Um dos principais recursos do DYMO é não armazenar explicitamente a topologia da rede. Somente quando necessário é que os nós calculam uma rota *unicast* para o destino. Com isso, poucas informações de roteamento são trocadas, evitando a sobrecarga do tráfego e economizando na largura de banda, além de facilitar o uso em dispositivos com restrição de memória. (ARAVIND *et al.*, 2015).

Na Figura 5 pode-se notar que ao usar o DYMO o nó A conhece as rotas para B, C e D após a solicitação de rota ser atendida para o nó D. Isso ocorre pelo fato do DYMO salvar as informações de rotas dos nós intermediários.

Figura 5 – Divulgação da informação de roteamento no DYMO.



Fonte: (DIETRICH *et al.*, 2007)

3 TRABALHOS RELACIONADOS

Esta seção apresenta os principais trabalhos, nos quais esta pesquisa se relaciona. Na subseção 3.1 é apresentado trabalhos referente a redes ad hoc. Na subseção 3.2 é mostrado sobre redes MANET, já na subseção 3.3 é apresentado trabalhos referente a protocolos de roteamento e seus desempenhos.

3.1 Redes *ad hoc*

O trabalho proposto por Ren *et al.* (2016) buscou projetar um sistema de rede ad hoc de alta velocidade com antena direcional. Nesse projeto ele discutiu a composição do sistema de comunicação que pode satisfazer a rede ad hoc cujo nó nesta rede possui alta velocidade (mais de 100m / s) e longa distância (10km-40km) entre si. Os resultados da simulação mostraram que o esquema pode ser satisfeito com os cenários específicos de redes ad hoc entre os veículos aéreos de alta velocidade com antena direcional. Este trabalho é similar ao proposto nesta monografia, pois ele também usa redes Ad hoc e o protocolo AODV na camada de roteamento. Entretanto as diferenças são bem específicas pelo fato dele usar antenas fixas para a comunicação entre os dispositivos.

O trabalho proposto por Velloso (2003) analisa a capacidade de transmissão de voz em redes *ad hoc*, revelando aspectos específicos do comportamento deste tipo de rede. Os seus resultados mostram que mesmo em cenários bastante adversos onde a rede está carregada e os nós se movem constantemente é viável a comunicação de voz sem perda de interatividade. Mas também comprovou que o aumento, tanto da mobilidade quanto da carga da rede, implica a degradação da capacidade de transmissão de voz. Este trabalho é similar ao proposto nesta monografia, pois ele também usa redes ad hoc. Todavia, as diferenças são diversas, desde dos dados que são trocados que no caso dele são dados de voz, como também a mobilidade dos dispositivos. Pois a mobilidade de seus dispositivos são previsíveis e limitada ao seu campo de locomoção, facilitando assim a conectividade dos nós, pois eles não poderão se afastar a uma distância tão grande. Entretanto, o proposto nesta monografia difere nas possibilidades de mobilidade dos nós, que podem se movimentar para qualquer lugar, sem limite de locomoção, dificultando assim a conectividade destes nós para cada salto na rede.

3.2 Redes VANET

O trabalho proposto por Yassine e Salah (2019) apresenta uma visão geral da *Vehicle Ad Hoc Networks* (VANET) e seus protocolos de roteamento. Buscando enfatizar o uso do roteamento de camada cruzada que visam otimizar a função de roteamento na camada de rede. A técnica de camada cruzada é aplicada entre a camada de rede e as camadas inferiores (a camada MAC, a camada física ou ambas). A VANET inclui veículos ativados por rádio que atuam como roteadores para garantir a comunicação. Embora o trabalho de Yassine e Salah (2019) use VANET que é bem semelhante as MANET, ele se difere do proposto nesta monografia pelos seguintes motivos, nas redes VANET dispõe de energia e armazenamento suficientes, forma geográfica de comunicação, mobilidade previsível e restrições de atraso.

3.3 Protocolos de roteamento

Fontoura *et al.* (2009) propõe em seu trabalho avaliar resultados de desempenho dos protocolos de roteamento MANET em uma topologia de malha. As redes de malha foram criadas com o objetivo de dinamizar a inserção rápida e segura do usuário. Entretanto, o dinamismo encontrado nessa topologia de rede, cria muitos desafios, principalmente na escolha de algoritmos de roteamento que atendam todos os requisitos das diferentes topologias. Este trabalho mostrou resultados significativos dos desempenhos de alguns protocolos existentes na literatura. Mas percebeu-se que não existe um protocolo mais eficiente em todos os parâmetros requeridos nas redes MANET. As principais diferenças dos trabalhos para o proposto nesta monografia é que nesta monografia além de buscar os protocolos mais eficiente, o protocolo será adaptado e implementado em conjunto com o *Nearby Connections*.

O trabalho proposto por Piechowiak *et al.* (2016) estudou a eficiência dos protocolos de roteamento de rede para redes mesh sem fio com OMNeT ++. Buscou uma análise comparativa de protocolos de roteamento representativos. Os resultados indicaram uma menor latência na entrega de pacotes e menores valores de fator de perda de pacotes usando protocolos proativos. Este trabalho é similar ao proposto nesta monografia, pois usa comparação de protocolos de roteamentos, todavia, ele difere no ambiente de simulação como também na topologia de rede que no caso dele é redes em malha.

4 TESTES E RESULTADOS

Neste capítulo serão apresentados os resultados obtidos a partir da aplicação da metodologia.

4.1 Comparação dos protocolos

Foi realizado um levantamento dos protocolos mais utilizados em redes de topologia Ad hoc. Desta forma foram selecionados os seguintes protocolos: *Dynamic MANET On-demand* (DYMO), *Optimized Link State Routing* (OLSR), *Ad hoc On-demand Distance Vector* (AODV), *Cluster Gateway Switch Routing* (CGSR), *Destination-Sequenced Distance Vector routing* (DSDV), *Core Extraction Distributed Ad Hoc Routing* (CEDAR), *Dynamic Source Routing* (DSR). Na literatura não encontrou-se uma padronização para a seleção dos critérios de avaliação de desempenho de uma rede, desta forma estabeleceu-se três critérios baseando-se no trabalho feito por Kurose e Ross (2006), sendo eles: reativo, latência e alta mobilidade da rede.

Conforme os critérios estabelecidos por Kurose e Ross (2006) e as definições dos algoritmos estabelecidas pelos autores que os criaram. Definiu-se reativo, pois necessariamente ele deveria agir sob demanda, pelo fato de não precisar ficar enviando periodicamente uma mensagem de controle dos nós, pois isso iria sobrecarregar a rede. A latência é o tempo de ida e volta de um pacote na rede, essa característica é importante pelo fato de calcular o tempo de atraso de uma mensagem. Quanto à mobilidade, o protocolo teria que suportar uma rede com alta mobilidade pelo fato de sempre haver movimentação dos nós, tornando assim a rede dinâmica. O AODV e o DYMO foram os que atenderam os critérios definidos. Conforme apresentado no Quadro 1.

Quadro 1 – Comparação dos Protocolos

Nome do Protocolo	Autor	Reativo	Latência	Tolerância a mobilidade
DYMO	Chakeres & Perkins, 2010	SIM	ALTA	ALTA
OLSR	Clausen & Jacquet, 2003	NÃO	BAIXA	BAIXA
AODV	Perkins et al., 2003a	SIM	ALTA	ALTA
CGSR	Chiang, 1997	NÃO	BAIXA	BAIXA
DSDV	Perkins & Bhagwat, 1994	NÃO	BAIXA	BAIXA
CEDAR	Sivakumar, Sinha & Bharghavan, 1999	SIM	MÉDIA	ALTA
DSR	Johnson et al., 2002	SIM	ALTA	MÉDIA

Fonte: O autor

4.2 Adaptação e implementação do protocolo junto ao Nearby connections para o desenvolvimento do aplicativo

O primeiro passo para o desenvolvimento do algoritmo foi o estudo da API Nearby Connections, com o objetivo de entender o funcionamento de seus métodos e das suas três estratégias, P2P, P2P Star, P2P Cluster. P2P é uma estratégia ponto a ponto que permite estabelecer conexão de 1 para 1. Ou seja, ela aceita conectar dispositivos dentro do alcance do rádio de aproximadamente 100 metros, entretanto, não permite mais de uma conexão por vez.

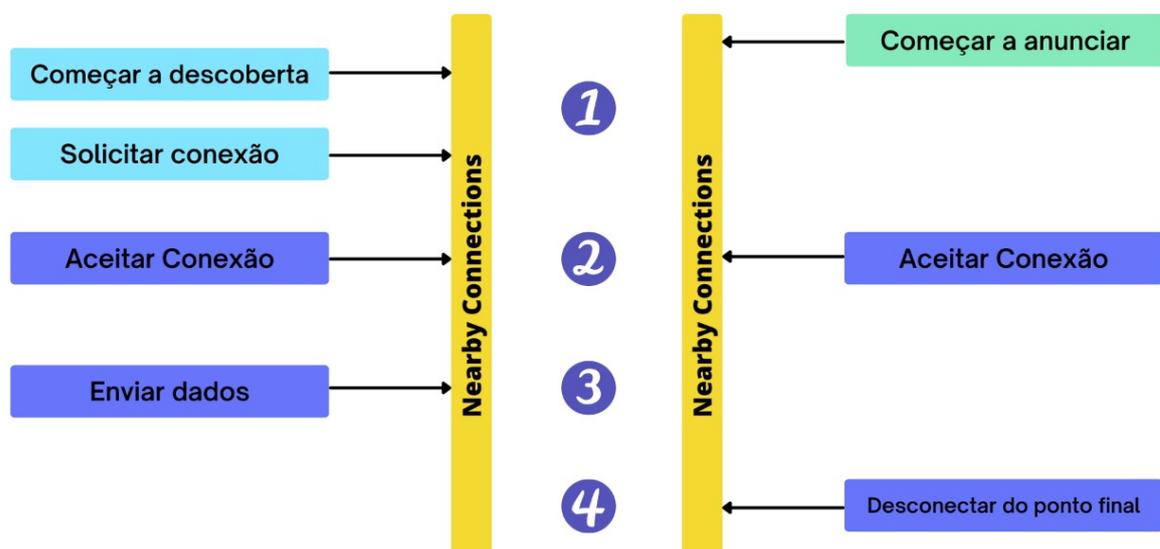
A P2P Star é uma estratégia que permite conexão de 1 para N ou em forma de estrela. Desta forma os dispositivos podem, a qualquer momento, aceitar conexões de entrada de outros N dispositivos ou podem iniciar uma conexão de saída para um único dispositivo. A última das estratégias é a P2P Cluster, que suporta conexão de M para N. Assim cada dispositivo pode iniciar M conexões de saída para outros dispositivos e aceitar conexões de entrada de N outros dispositivos.

Os métodos disponibilizados pela API Nearby Connections, estão relacionados aos processos de anúncio, descobrimento, gerenciamento da conexão e a troca de dados entre os dispositivos. O processo de anúncio consiste em o dispositivo indicar para a rede que ele está disponível para estabelecer conexões. A etapa de descobrimento consiste em os dispositivos

procurar por outros que estejam no processo de anúncio. Os métodos de gerenciamento de conexão são chamados quando um dispositivo anunciante é descoberto, podendo assim aceitar ou negar as conexões. Depois de estabelecidas as conexões os dispositivos podem trocar dados nos tipos de byte, arquivos e stream.

Na figura 6 é exemplificado o processo para estabelecimento da conexão. Sendo que a etapa 1 refere-se aos processos de anúncio e descobrimento. A etapa 2 mostra o processo de estabelecimento da conexão. Já a etapa 3 exemplifica como é realizada a troca de dados. E por fim a etapa 4 demonstra como é feito o encerramento da conexão.

Figura 6 – Etapas para estabelecer conexão



Fonte: (ANDROID, 2020)

Após observado o comportamento das estratégias disponibilizadas pelo Nearby Connections percebeu-se que a P2P Cluster era a que se adequava melhor ao problema, devido ao fato de ser necessário que os dispositivos atuassem como anunciantes e descobridores simultaneamente. Contudo, os dispositivos ainda precisavam estar dentro do mesmo raio de alcance, assim para permitir que dispositivos fora do seu raio de alcance se comunicassem foi necessária uma adaptação dos métodos do Nearby Connections. Com isso, foi desenvolvido no ambiente Android Studio na linguagem Java um aplicativo Android que implementou e adaptou os métodos do Nearby Connections para tornar possível saltos entre os dispositivos.

O aplicativo foi implementado de forma que para cada dispositivo descoberto um

pedido de conexão era enviado automaticamente, e para cada pedido de conexão recebido, ele era automaticamente aprovado. Cada conexão estabelecida tem seu Identificador (ID) que é gerado automaticamente pela API e seu Nome que é informado pelo usuário no momento em que o aplicativo é iniciado, salvos em uma lista de conexões. O método descrito pode ser visualizado no Algoritmo 1.

Algoritmo 1: Conexão Iniciada

```

Function conexaoIniciada(nomeEndpoint):
    |   aceitarConexao(nomeEndpoint)
    |   conexas.adicionar(nomeEndpoint)
End Function

```

Depois que os dispositivos estão visíveis na rede, a troca de mensagens pode ser realizada. Baseando-se nos algoritmos DYMO e AODV, o algoritmo de envio de mensagens para a realização de saltos foi implementado de forma que cada mensagem tem um Identificador de destino, assim o destinatário de cada mensagem enviada é verificado a sua existência na lista de conexões. Se ele estiver contido na lista a mensagem é encaminhada diretamente para ele, caso não esteja a mensagem é enviada para todos os dispositivos daquela lista.

Quando uma mensagem é recebida, o dispositivo verifica o real destinatário, caso não seja ele, o algoritmo de envio de mensagem é aplicado. Caso seja ele próprio o destinatário este exibirá a mensagem no chat da conversa e uma mensagem de confirmação é enviada para o remetente. Desse modo a mensagem consegue ser propagada, pois assim ela irá saltando entre os dispositivos até conseguir chegar ao destinatário real.

4.3 Realização de testes

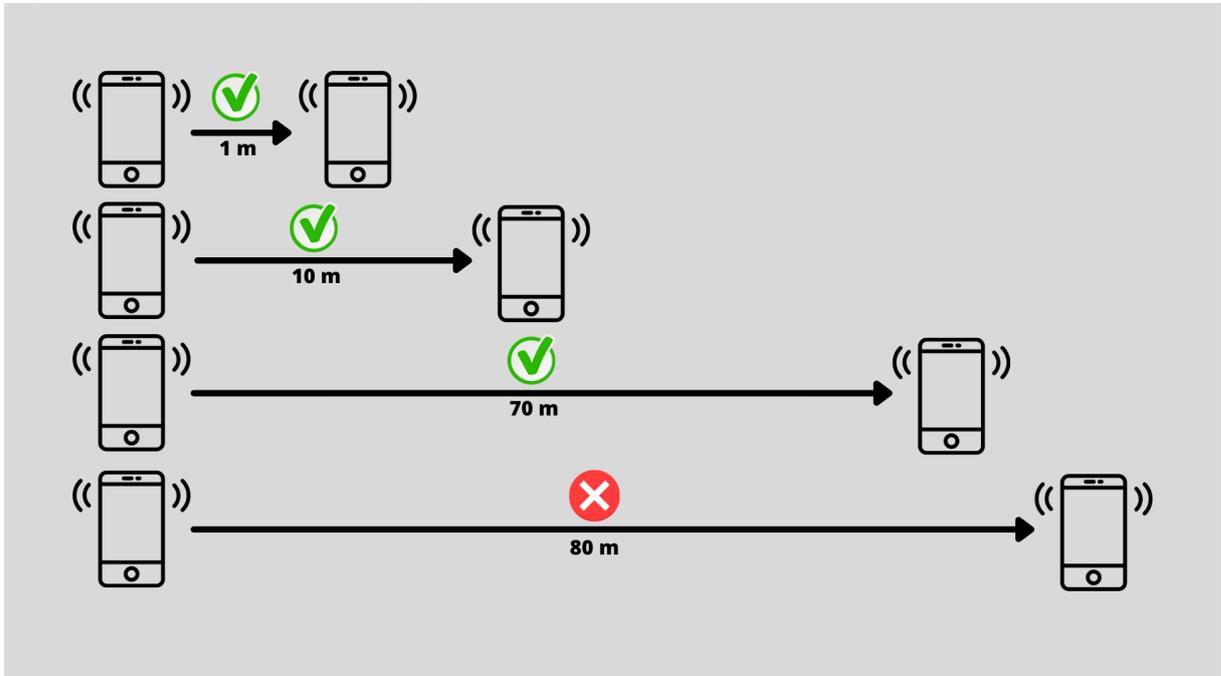
4.3.1 Teste de distância

O teste de distância foi realizado de maneira incremental, iniciou-se com a distância de 1 metro entre os dispositivos, logo em seguida a distância foi sendo aumentada de 10 em 10 metros até a conexão ser perdida. Para a verificação do número de saltos, os dispositivos foram conectados de maneira direcionada, para evitar que mesmo os dispositivos com possibilidade de conexão direta, não se conectassem.

Os testes tiveram início com o cenário de transferência de mensagens de um dispositivo A para o dispositivo B com a distância de 1 metro entre eles. Após a verificação do recebimento da mensagem, aumentou-se a distância para 10m, até chegar aos 100m. Percebeu-se

que a distância máxima onde um dispositivo conseguiu se comunicar com o outro sem realizar saltos foi de 70 metros. Conforme ilustrado na Figura 7

Figura 7 – Distância entre os dispositivos



Fonte: O autor

4.3.2 Teste de tempo

Em cada mensagem enviada é atribuído um identificador único, a data e hora do envio e um campo para a confirmação do recebimento. Para cada mensagem recebida uma confirmação do recebimento é enviada para o remetente. Quando a confirmação chega ao remetente ele calcula a diferença de tempo entre o envio e chegada da confirmação. Desta forma o teste de tempo de envio e recebimento da mensagem foi feito através de uma função que calcula o tempo de ida e volta de uma mensagem. Conforme mostrado no Algoritmo 2.

Algoritmo 2: Calcular tempo médio para as mensagens

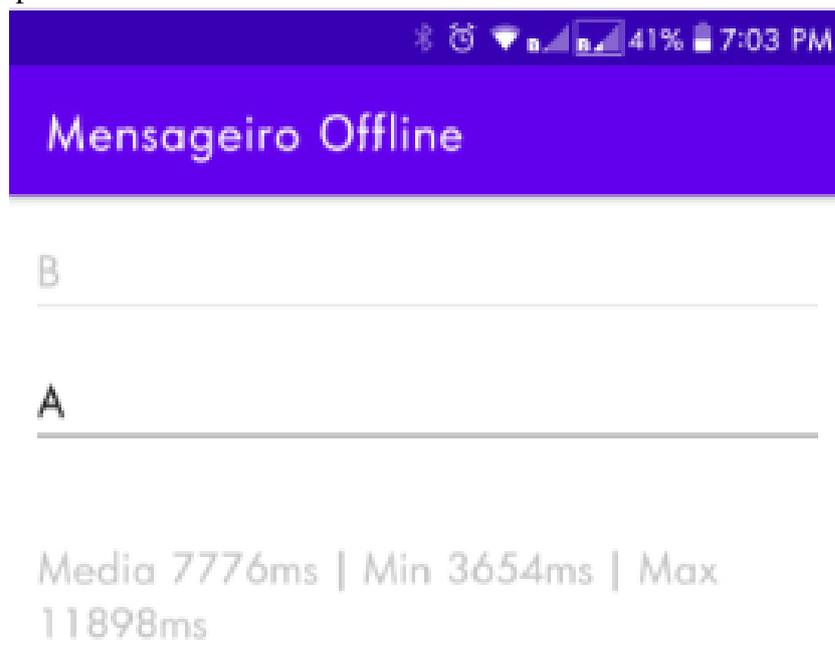
```

tempoMedio ← 0
for mensagem ← Mensagens do
  if mensagem.origem = id usuario then
    diferenca ← diferencaEnvioConfirmacao(mensagem)
    tempoMedio ← tempoMedio + diferenca
    if diferenca < tempoMínimo then
      | tempoMinimo ← diferenca
    end
    if diferenca > tempoMáximo then
      | tempoMaximo ← diferenca
    end
  end
tempoMedioEnvioConfirmacao ← tempoMedio / mensagens.tamanho
end

```

Foram realizados testes em diferentes cenários, onde calculou-se por meio da função ilustrada no Algoritmo 2, o tempo médio, máximo e mínimo para os envios. Esses dados foram exibidos na interface do aplicativo. Conforme apresentado na Figura 8.

Figura 8 – Tempo médio em diferentes cenários



Criou-se também uma tabela onde mostra o tempo em alguns cenários, especificando a quantidade de dispositivos conectados na rede, a distância entre os dispositivos em metros, número de saltos e o Tempo médio em milissegundos. Conforme o Quadro 2.

Quadro 2 – Tempo médio em diferentes cenários

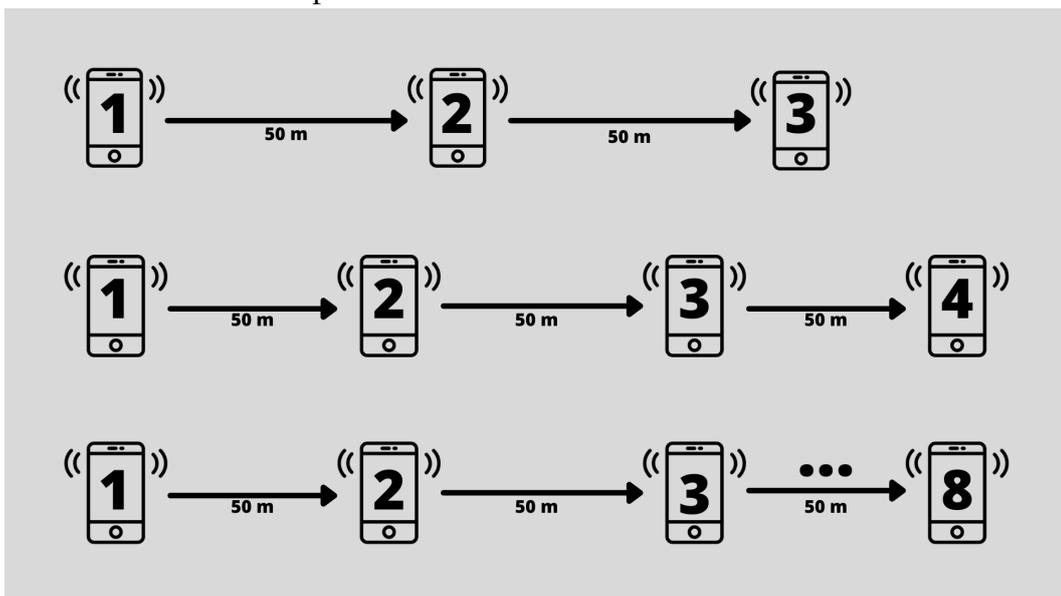
Número de Dispositivos	Distância entre os dispositivos (m)	Número de saltos	Tempo médio (ms)
3	1	1	145
5	10	3	343
7	15	5	815
9	10	7	1.091
10	5	8	2.006
15	5	13	3.624

Fonte: O autor

4.3.3 Teste do Número de saltos

A verificação do número de saltos iniciou-se com o cenário de três dispositivos com 50m a distância entre cada dispositivo, após a verificação se a mensagem havia chegado corretamente, aumentou-se a quantidade de dispositivos gradativamente para a realização de saltos. Onde foi possível testar a quantidade de até 6 saltos, totalizando a distância de 350m do dispositivo de número 1 até o dispositivo de número 8. Conforme apresentado na figura 9.

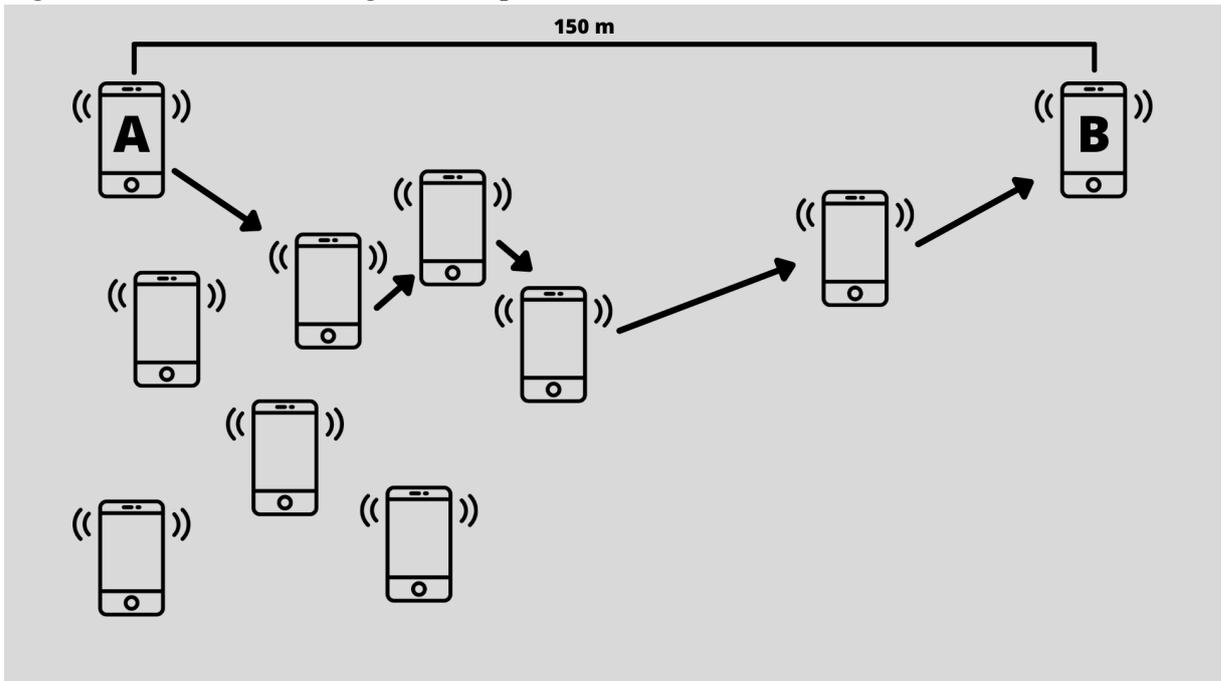
Figura 9 – Saltos entre os dispositivos



Fonte: O autor

Realizou-se teste onde o cenário tinha 10 dispositivos conectados, entretanto, o dispositivo A deveria enviar mensagem para o dispositivo B a uma distância entre eles de 150 metros, onde ele conseguiu realizar a troca de mensagem fazendo salto entre os dispositivos que estavam ao seu alcance. Conforme ilustrado na Figura 10.

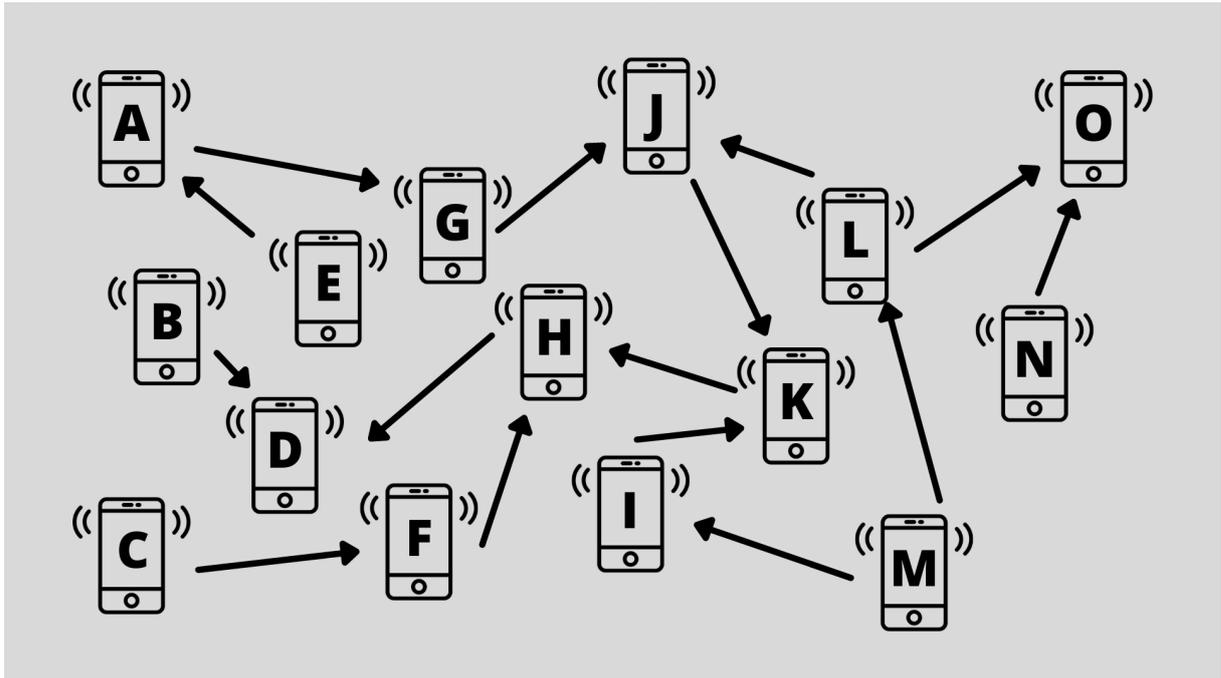
Figura 10 – Envio de mensagens de A para B realizando saltos



Fonte: O autor

Realizou-se também testes com 15 dispositivos conectados trocando mensagens entre si, onde o objetivo foi testar o comportamento do aplicativo quando várias mensagens de vários dispositivos eram enviadas simultaneamente. Percebeu-se que algumas mensagens chegaram com um pouco mais de atraso se comparado com as que eram enviadas diretamente. Tendo um atraso médio máximo de 7.776 milisegundos. Conforme ilustrado na Figura 11.

Figura 11 – Todos os dispositivos enviando mensagens simultâneas



Fonte: O autor

5 CONCLUSÃO

Apesar das dificuldades encontradas para realização dos testes propostos, principalmente pela quantidade de dispositivos necessários para simular situações em que acredita-se que seja o contexto de uso deste aplicativo. Foi possível atingir o objetivo geral por meio do desenvolvimento de um aplicativo de comunicação entre celulares que troca mensagens de maneira totalmente *offline*. Como também os objetivos específicos: Comparar o desempenho dos protocolos de roteamento, adaptar protocolo de roteamento em conjunto com o *Nearby Connection* e testar o alcance da troca de mensagem P2P do aplicativo. Conforme demonstrado no capítulo de resultados. Diante dessas informações concluiu-se que este aplicativo pode ser usado como uma ferramenta de troca de mensagens no contexto em que ocorra a falta de estrutura da internet ou quando os usuários julgarem necessário.

Algumas melhorias podem ser realizadas em trabalhos futuros para garantir mais qualidade, segurança e melhor desempenho do aplicativo. A primeira delas diz respeito à experiência do usuário, que é uma refatoração na interface do aplicativo, buscando atender os critérios de IHC. A segunda melhoria refere-se a criptografia das mensagens, visto que a criptografia entre conexões já é realizada pelo *Nearby Connections*. A terceira melhoria remete-se a tabela de conexões, visando diminuir a sobrecarga na rede e aumentar o desempenho do aplicativo, por meio de um melhor gerenciamento das conexões.

REFERÊNCIAS

- ANDROID. **android NearbyConnections**. 2020. Disponível em: <<https://androidkt.com/nearby-connections-api-2-0/>>. Acesso em: 10 dez. 2020.
- ARAVIND, M. C.; SANGEETHA, C.; SURIYAKALA, C. Enhanced dynamic manet on-demand (en-dymo) routing protocol for mobile adhoc networks. In: IEEE. **2015 Global Conference on Communication Technologies (GCCT)**. [S.l.], 2015. p. 544–549.
- DIETRICH, I.; SOMMER, C.; DRESSLER, F. Simulating dymo in omnet++. **University of Erlangen, Dept. of Computer Science**, v. 7, 2007.
- FONTOURA, A. C. da; SCHMIDT, R. de O.; TRENTIN, M. A. S. Simulation of ad-hoc routing protocols in an interurban university mesh network. **IEEE Latin America Transactions**, IEEE, v. 7, n. 1, p. 122–125, 2009.
- GEETHA, V.; AITHAL, S.; CHANDRASEKARAN, K. Effect of mobility over performance of the ad hoc networks. In: IEEE. **2006 International Symposium on Ad Hoc and Ubiquitous Computing**. [S.l.], 2006. p. 138–141.
- KUMAR, R.; ARYA, K.; SHEKHAR, S.; AGRAWAL, R. An on demand routing protocol aodv with end to end reliability and backward route information. In: IEEE. **2014 9th International Conference on Industrial and Information Systems (ICIIS)**. [S.l.], 2014. p. 1–6.
- KUROSE, J. F.; ROSS, K. W. Redes de computadores e a internet. **Uma nova**, 2006.
- LEME, E. *et al.* Análise de falhas em comunicação multicast-gossip no ambiente manet. [sn], 2016.
- MEFTAH, L.; ROUVOY, R.; CHRISMENT, I. Testing nearby peer-to-peer mobile apps at large. In: IEEE PRESS. **Proceedings of the 6th International Conference on Mobile Software Engineering and Systems**. [S.l.], 2019. p. 1–11.
- MIAO, Q.-x.; XU, L. Dymo routing protocol research and simulation based on ns2. In: IEEE. **2010 International Conference on Computer Application and System Modeling (ICCSM 2010)**. [S.l.], 2010. v. 14, p. V14–41.
- MONTRESOR, A.; JELASITY, M. Peersim: A scalable p2p simulator. In: IEEE. **2009 IEEE Ninth International Conference on Peer-to-Peer Computing**. [S.l.], 2009. p. 99–100.
- OMARI, M.; DAHOU, A. Simulation comparison and analysis of dsr and dymo protocols in manets. In: IEEE. **2016 International Conference on Industrial Informatics and Computer Systems (CIICS)**. [S.l.], 2016. p. 1–6.
- PAPASTERGIOU, G.; PSARAS, I.; TSAOUSSIDIS, V. Deep-space transport protocol: a novel transport scheme for space dtms. **Computer Communications**, Elsevier, v. 32, n. 16, p. 1757–1767, 2009.
- PEREIRA, N. C. V. N.; MORAES, R. M. de. Comparative analysis of aodv route recovery mechanisms in wireless ad hoc networks. **IEEE Latin America Transactions**, IEEE, v. 8, n. 4, p. 385–393, 2010.

- PIECHOWIAK, M.; ZWIERZYKOWSKI, P.; OWCZAREK, P.; WASŁOWICZ, M. Comparative analysis of routing protocols for wireless mesh networks. In: IEEE. **2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)**. [S.l.], 2016. p. 1–5.
- PROJETODEREDES. **projeto redes**. 2019. Disponível em: <https://www.projetoderedes.com.br/artigos/artigo_redes_moveis_ad_hoc.php>. Acesso em: 29 out. 2019.
- REN, B.; ZHANG, X.; GOU, X. System design of high speed ad hoc networking with directional antenna. In: IEEE. **2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)**. [S.l.], 2016. p. 429–433.
- REY, L. C.; QUIÑONES, T. O. L.; GARCÍA, W. B. Protocolos de enrutamiento aplicables a redes manet. **Revista Telemática**, v. 13, n. 3, p. 59–74, 2014.
- SANTOS, H. N. dos. **Rede Mesh**. 2006. Disponível em: <<https://sites.google.com/site/redemesh/Home/roteamento-dinamico/aodv>>. Acesso em: 30 jul. 2019.
- SHARMA, A. K.; NOIDA, D. C.; MISHRA, A. A study of energy optimization for manet. In: IEEE. **2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)**. [S.l.], 2019. p. 264–267.
- TECHTUDO. **techtudoNearbyConnections**. 2006. Disponível em: <<https://www.techtudo.com.br/noticias/2017/08/nearby-connections-como-funciona-o-recurso-que-conecta-dispositivos-offline.ghtml>>. Acesso em: 17 jul. 2019.
- VELLOSO, P. B. **Transmissão de voz em redes ad hoc**. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2003.
- YASSINE, H.; SALAH, M. Vanet cross-layer routing. In: IEEE. **2019 International Conference of Computer Science and Renewable Energies (ICCSRE)**. [S.l.], 2019. p. 1–2.
- ZHANG, H. Cluster-to-cluster overlay network for video systems over wireless ad hoc networks. In: IEEE. **2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks**. [S.l.], 2011. p. 356–357.