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

Rafael Souza Oliveira

Confiabilidade de conexão entre dispositivos aplicado ao 5G

João Monlevade
2017

Rafael Souza Oliveira

Confiabilidade de conexão entre dispositivos aplicado ao 5G

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".

Orientador: Theo Silva Lins

João Monlevade

2017

O482c Oliveira, Rafael Souza.
Confiabilidade de conexão entre dispositivos aplicado ao 5G [manuscrito] / Rafael Souza Oliveira. - 2017.

49f.: il.: color; tabs.

Orientador: Prof. Me. Theo Silva Lins.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Redes de computadores. 2. Confiabilidade . 3. Roteadores. 4. Internet. I. Lins, Theo Silva. II. Universidade Federal de Ouro Preto. III. Titulo.

CDU: 004.7

Catalogação: ficha@sisbin.ufop.br

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DE SISTEMAS DE INFORMÇÃO

ATA DE DEFESA

Aos 31 dias do mês de março de 2017, às 17:00 horas, na sala C204 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo aluno **Rafael Souza Oliveira,** sendo a Comissão Examinadora constituída pelos professores: Prof. MSc. Theo Silva Lins, Prof. MSc. Marlon Paolo Lima e Prof. Dr. Vinicius Fernandes Soares Mota.
O candidato apresentou a monografia intitulada: *"Confiabilidade de conexão entre dispositivos aplicado ao 5G"*. A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 9 (NOVE), 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 graduando.

João Monlevade, 31 de março de 2017.

Prof. MSc. Theo Silva Lins
Professor Orientador/Presidente

Prof. MSc. Marlon Paolo Lima
Professor Convidado

Prof. Dr. Vinicius Fernandes Soares Mota
Professor Convidado

Rafael Souza Oliveira
Graduando

Agradecimentos

Devo agradecer a Deus, pela força nas horas em que pensei em desistir e pensei que não conseguiria desenvolver este trabalho. Agradeço a Deus por ter colocado pessoas em meu caminho que me impulsionaram a chegar no fim deste capitulo da vida. Agradeço a Deus pelos meus pais, irmãos,namorada, amigos que estiveram do meu lado durante todo este tempo.

Depois de agradecer a Deus, devo colocar no alto desta lista, meus pais. Sem eles, eu não seria nada do que sou hoje. Não teria chegado aqui se minha mãe não tivesse cobrado tudo que cobrou de mim ou meu pai não tivesse apoiado e dado o exemplo de pessoa estudiosa e dedicada que é. Não teria alcançado metade das minhas conquistas profissionais e pessoais, se não fosse o esforço da minha mãe e do meu pai em sempre querer o melhor pra mim e meus irmãos. Saibam que sempre serão os mais importantes e que a opinião e o reconhecimento de vocês é o que eu mais considero para a minha vida. Obrigado Deus, pelos meus pais, e obrigado pais, pelo que sou.

Agradeço a meus irmãos, por me ensinarem a estudar programação, revisarem meus trabalhos, lerem este documento de TCC várias vezes e darem opiniões. Obrigado pelo incentivo.

Agradeço a minha namorada, que durante toda a minha graduação, em momentos de estresse, de decepções ou de pensar em desistir esteve do meu lado e me impulsionou a continuar. Obrigado por se manter do meu lado nas escolhas, sendo elas erradas ou corretas.

Obrigado amigos que contribuíram de alguma forma para que eu chegasse a este momento da minha vida, seja com apoio emocional, carona ou qualquer outro tipo de coisa que amigos verdadeiros fazem por amigos.

Por fim, agradeço a meu orientador, sem você este trabalho provavelmente teria ficado esquecido em alguma prateleira. Se eu não estivesse escutado a frase "Trabalho Teórico não dá!!"talvez eu nunca teria concluído este estudo. Obrigado pelo incentivo, pelo tempo disponibilizado e pelas sugestões. Enfim, obrigado pela orientação.

Espero não ter me esquecido de nenhum contribuinte direto deste trabalho. Por aqui finalizo meus agradecimentos para a honra e gloria do Senhor Deus, certo de que Ele me ajudou até aqui e continuará me ajudando se eu a Ele continuar fiel.

Resumo

Os recursos tecnológicos para transmissão de dados entre dispositivos móveis existentes se tornarão insuficientes se considerarmos a velocidade com que a tecnologia móvel evolui em função disso o surgimento do conceito de redes 5G se fez necessário. As redes 5G apresentam uma estrutura diferente das redes conhecidas até hoje bem como novos problemas relacionados a confiabilidade da conexão que podem causar transtornos durante a comunicação entre os dispositivos. Desenvolver uma solução eficaz para o tratamento destes problemas se torna necessário e em função disso um algoritmo nomeado *DSniffer* que classifica cada nó da rede com uma nota em função de algumas variáveis foi idealizado. Este algoritmo tem a função de percorrer os nós de uma rede em busca de melhores caminhos para transmitir a informação solicitada entre remetentes e destinatários. A seleção dos melhores caminhos é dada através da avaliação de cada dispositivo em função de algumas variáveis. Essas variáveis são fornecidas pelo próprio dispositivo e avaliadas em função dos valores retornados verificando fatores relacionados a velocidade de transmissão, consumo de energia e segurança. Com esta solução espera-se obter melhor confiabilidade na conexão entre os dispositivos durante uma comunicação numa rede, e que os dados enviados trafeguem através dos nós da rede sem risco de perda até o destino.

Palavras-chaves: *DSniffer*, Algoritmo, Roteamento, Confiabilidade, Conexão, 5G, D2D, Redes

Abstract

The technological resources for data transmission between existing mobile devices will become insufficient considering the speed which mobile technology evolves as a result that emergence the concept of 5G networks has become necessary. The 5G networks present a different structure of networks known and it presents new problems related to the reliability of the connection that can cause inconvenience during the communication between the devices. To develop an effective solution for the treatment of these problems becomes necessary and an algorithm named textit DSniffer that classifies each node of the network with a note in function of some variables was idealized to classify each node of the network with a note in function of some variables. This algorithm has the function of go through the network nodes searching for better ways to transmit the requested information between senders and recipients. The selection of the best paths is given through the evaluation of each device in function of some variables that are provided by the device itself and evaluated according to the returned values, verifying values related to transmission speed, energy consumption and safety. This solution is expected to achieve better reliability in the connection between the devices during a network communication where the data sent through the nodes arrives to the destination secured.

Keywords: *DSniffer*, Algorithms, Routing, Reliability, Connection, 5G, D2D, Networks

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

D2D - *Device-to-Device*

MBPS - *Megabits* Por Segundo

GBPS - *Gigabits* Por Segundo

GSM - *Global System for Mobile Communication*

1G - Primeira Geração de padrões de tecnologia para dispositivos moveis

2G - Segunda Geração de padrões de tecnologia para dispositivos moveis

2.5G - Geração intermediaria entre 2G e 3G de padrões de tecnologia para dispositivos moveis

3G - Terceira Geração de padrões de tecnologia para dispositivos moveis

4G - Quarta Geração de padrões de tecnologia para dispositivos moveis

5G - Quinta Geração de padrões de tecnologia para dispositivos moveis

BS - *Base Station*

QoS - *Quality of Service*

IoT - *Internet Of Things*

MMS - *Mult Media Messages*

SDN - *Software Defined Networks*

IEEE - *Institute of Electrical and Electronic Engineers*

WEP - *Wired Equivalent Privacy*

WPA - *Wi-Fi Protected Access*

WPA2 - *Wi-Fi Protected Access II*

Sumário

1 Introdução

A história da Internet se inicia em 1962, antes que a palavra Internet fosse inventada. Os aproximadamente 10000 computadores eram primitivos apesar de custarem centenas de dólares. Eles tinham apenas alguns milhares de palavras num núcleo magnético de memória e programa-los era de longe algo fácil. A agencia de projetos para pesquisas avançadas (ARPA) do departamento de defesa dos Estados Unidos iniciou uma pesquisa a longo prazo de alto risco e alto ganho que se tornou a *ARPANET* e mais tarde a Internet. (COMPUTERHISTORY.ORG, 2015)

Com o passar dos anos, a Internet cresceu consideravelmente, e hoje conta com bilhões de *hosts* e usuários que podem ou não serem confiáveis para o tráfego do volume astronômico de informações que estes *hosts* enviam e recebem todos os dias. Estes *hosts* deixaram de ser simplesmente computadores físicos e se tornaram dispositivos móveis, que ficam ao alcance das mãos das pessoas a qualquer hora fornecendo praticidade e velocidade na comunicação, transmitindo dados pessoais e muitas vezes sigilosos que necessitam de trafegar por caminhos onde não corram risco de se perderem ou serem desviados a destinos incorretos, caindo em mãos erradas.

A confiabilidade de conexão entre dispositivos móveis no cenário das redes de comunicação é indispensável quando informações de cunho pessoal ou sigiloso devem ser transmitidas através dos nós dessas redes. Considerando este fato, devem ser desenvolvidas tecnologias que assegurem o tráfego de dados entre os dispositivos em uma rede de computadores ou dispositivos móveis.

Algoritmos de roteamento combinados com verificações específicas diretamente feitas em cada um dos nós da rede devem ser implementados com a intenção de resolver um problema que será recorrente no tráfego de dados em redes do tipo 5G. O problema citado se deve ao fato de dispositivos de pessoas comuns serem utilizados para auxiliar na transmissão dos dados nessa nova tecnologia. Para isso esses dispositivos devem ter um nível de confiabilidade alto. Esse nível de confiabilidade deve assegurar que os dados de um usuário sejam entregues em seu destino sem sofrer interferências. Os dispositivos devem ter requisitos mínimos que variam entre utilização de encriptação, latência de conexão, intensidade de sinal dentre outros que serão tratados nos próximos capítulos

O tipo de pesquisa utilizado para implementação deste documento teve sua parcela bibliográfica, porém, as conclusões só puderam ser finalizadas com testes práticos.

1.1 Objetivo Geral

Este trabalho tem como objetivo geral idealizar um algoritmo que poderá selecionar as melhores rotas para o tráfego de informações entre dispositivos nas redes 5G baseado em métricas que serão posteriormente apresentadas. Inicialmente deseja-se apresentar soluções para o problema do tráfego de dados entre dispositivos nas redes 5G que trabalharão com modelo D2D(*Device-to-Device*) sem controle de operadora onde a confiabilidade de conexão entre cada dispositivo deve ser fortemente considerada.

1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

1. Apresentar e implementar a topologia de trabalho de uma rede 5G.
2. Apresentar problemas do tráfego de informações entre dispositivos.
3. Implementar um algoritmo que consiga resolver o problema de confiabilidade de conexão entre dispositivos numa rede.
4. Definir métricas para que o algoritmo a ser implementado assegure que os dispositivos que servirão de enlaces para o trafego da informação sejam confiáveis.
5. Tornar o algoritmo implementado funcional em uma rede.

1.3 Descrição do Problema

O problema a ser tratado neste trabalho está diretamente ligado ao tráfego confiável das informações entre dispositivos móveis considerando que, na tecnologia que será apresentada, as informações dos usuários trafegarão entre dispositivos de pessoas comuns que serão usados como enlaces. Estes enlaces devem ter o mínimo de confiança para trafegar informações muitas vezes sigilosas e/ou pessoais.

Chamamos aqui de tráfego confiável, a situação de que a informação estará sendo transmitida entre aparelhos que tem nível de segurança alto relacionado a conexão. Um dispositivo que não é considerado confiável não atende os requisitos mínimos de segurança que serão mostrados no decorrer deste documento.

O principal problema a ser tratado aqui são os dispositivos que não são considerados confiáveis dentre um conjunto de dispositivos que estarão disponíveis.

Acima foram apresentados o objetivo geral e os objetivos específicos e o problema a ser resolvido neste trabalho. No capítulo 2, será abordada a revisão bibliográfica contendo

uma breve demonstração das tecnologias que serão discutidas no decorrer deste documento obtidas de fontes especializadas no assunto. No capitulo 3 a metodologia utilizada neste trabalho será descrita juntamente com as configurações do ambiente de desenvolvimento e algumas ferramentas. Em seguida, no capitulo 4 o desenvolvimento de um algoritmo para o tratamento da confiabilidade entre dispositivos em uma rede será explicado contendo a descrição deste algoritmo função por função. No capitulo 5 são apresentados os testes em um ambiente controlado e no capitulo 6 as considerações finais. No capitulo 7, são sugeridos trabalhos futuros para complementar este estudo.

2 Revisão Bibliográfica

A transmissão de dados em alta velocidade entre dispositivos móveis vem sendo cada vez mais necessária. Com o crescimento deste tipo de transmissão e a necessidade da evolução das tecnologias surge o conceito das gerações. As chamadas 1G, 2G, 3G e 4G vieram com o objetivo de tentar aprimorar os meios de transmitir dados entre dispositivos móveis. Entretanto a tecnologia 5G não será uma evolução das anteriores. Essa tecnologia revolucionará todo o modo conhecido de comunicações entre dispositivos e será focada no conceito de *Internet of Things*(IoT). A separação de usuários com largura de banda baixa será possível e, para isso, serão necessários dispositivos com conexão confiável e que sejam capazes de operar com outros dispositivos e redes da internet das coisas. O núcleo da rede deverá atingir altos níveis de flexibilidade, inteligência e eficiência, os custos energéticos terão de ser considerados com mais cuidado e velocidades de 1GBPS a 10 GBPS deverão ser alcançadas. Esta revolução tecnológica lança muitos desafios em várias áreas de desenvolvimento (ANDREWS et al., 2014).

2.1 Gerações da tecnologia de transmissão de dados móvel

2.1.1 Primeira Geração : Redes 1G

Chamada de 1G, a primeira geração da tecnologia de comunicação entre dispositivos móveis consiste num conjunto de padrões desenvolvidos nos anos 80 e que veio para substituir a geração anterior chamada de 0G que basicamente era formada por telefones moveis a radio e aparelhos *push to talk* segundo o site (ITPORTAL.IN, 2014) . O artigo (THIRUPATHI; ANGELINROSY, 2012) cita que as redes 1G utilizavam sinais analógicos de rádio e uma ligação de voz comum era modulada a uma frequência alta de aproximadamente 150MHz e transmitida por torres de radio. Isso era feito utilizando uma técnica chamada *(FDMA-Frequency-Division Multiple Access)*. Segundo o artigo (MENDES, 2014), 1G foi lançada primeiramente em 1983 na Austrália, e em seguida no norte e sul da china.

2.1.2 Segunda Geração: Redes 2G

Segundo o (THIRUPATHI; ANGELINROSY, 2012) a segunda geração foi baseada no GSM ou abreviação de *global system for mobile communication*. Foi lançada primeiramente na Finlândia em 1991.

A tecnologia 2G já permitia serviços como mensagens de texto criptografadas digitalmente, mensagens com fotografias e MMS's(*mult media messages*). Era considerada muito mais eficiente que por oferecer segurança para os 2 lados(receptor e emissor).

Os principais benefícios da tecnologia 2G segundo o artigo era a clareza das ligações de voz e a interferência reduzida nas linhas. O sinal era digital ocasionando a redução de ruídos e melhorando a qualidade das ligações.

Segundo o (MENDES, 2014), a segunda geração deu o passo inicial para a transformação da rede de comunicação móvel para o momento presente. Esta tecnologia conseguia suportar em uma mesma área de cobertura 1G um numero maior de usuários com uma qualidade considerável.

Outro fator a se considerar em relação ao 2G segundo o artigo (MENDES, 2014) é a economia de energia fornecida aos aparelhos. A economia de energia fez com que a bateria celular pudesse ter maior durabilidade.

2.1.3 Intermediária entre Redes 2G e Redes 3G - Redes 2.5G

Segundo o artigo (THIRUPATHI; ANGELINROSY, 2012) existe uma tecnologia intermediária às 2G e 3G que pode ser chamada de 2.5G. Essa tecnologia é conhecida como GPRS ou *general packet radio service*. Esta tecnologia teria a capacidade de diminuir algumas limitações da tecnologia 2G(GSM). Segundo o artigo, 2.5G foi importante pois introduziu o conceito de pacotes direcionados preparando usuários, operadores e outros interessados para a tecnologia 3G.

2.1.4 Terceira Geração : Redes 3G

Seguindo o raciocínio do artigo (THIRUPATHI; ANGELINROSY, 2012), a tecnologia 3G adicionam serviços como transmissão de TV, GPS e vídeo conferência além de todos os outros serviços oferecidos anteriormente pelas tecnologias mais antigas além de velocidades mais altas de transferência de dados. A transferência de dados foi fixada em esperados 2Mbps para usuários estáticos e 348Kbps para usuários em movimento.

Existem varias tecnologias 3G, dentre elas estão W-CDMA, GSM, EDGE, UMTS, WiMax dentre outras. Segundo o artigo (MENDES, 2014), a tecnologia 3g foi implementada pela primeira vez em 2001 no Japão, no mesmo ano na Coreia do Sul, em 2003 foi aos Estados Unidos e na Índia em 2008. Em 2010 quase todo o mundo ja utilizava a cobertura 3G.

A usabilidade relativa a dispositivos móveis e a sincronia com a Internet são características marcantes desta tecnologia. A largura de banda varia de 5 a 20Mbps e frequência entre 16-25GHz.

2.1.5 Quarta Geração : Redes 4G

A tecnologia 4G é considerada uma extensão da tecnologia 3G com mais largura de banda e com a oferta de mais serviços. A expectativa em relação ao 4G é basicamente que esta tecnologia ofereça maior qualidade de *streaming* de áudio e vídeo utilizando o protocolo IP. A tecnologia 4G oferece altas taxas de transferência de dados, mais segurança de conexão e possibilitará às operadoras de rede a utilização de varias tecnologias novas no mercado. Segundo o artigo, (MENDES, 2014), o processo evolutivo das tecnologias móveis levou a quarta geração ser 10 vezes melhor do que a terceira geração, o que proporcionou um desempenho elevado no uso da tecnologia digital fornecendo serviços ligados a TV digital móvel, vídeo conferencias e altas velocidade de acesso a internet. Outra característica marcante da quarta geração segundo o artigo, é o aproveitamento eficiente das bandas acima de 5MHz além da latência abaixo de 5 milisegundos . A largura de banda é escalável ate 20Mbits com a taxa máxima de *download* de 100Mbits/s e *updload* de 50Mbits/s considerando esta largura de banda de 20Mbits.

2.2 Quinta Geração: Redes 5G

O artigo (MENDES, 2014), nos dá a ideia de que as redes 5G serão uma mistura de níveis de rede de diferentes densidades , com poderes de transmissão de conexão inteligente e heterogênea acessadas por um grande numero de dispositivos sem fio.

Segundo (TEHRANI; UYSAL; YANIKOMEROGLU, 2014) o termo dispositivo nesta tecnologia se refere a um celular ou qualquer outro dispositivo portátil com conexão *wireless* pertencente a algum usuário. A implementação dos dispositivos enlace se torna necessária para que a tecnologia 5G atinja seu potencial máximo. Ainda segundo (TEHRANI; UYSAL; YANIKOMEROGLU, 2014), os dispositivos enlace podem funcionar como enlaces de transmissão de dados para outros dispositivos e criar uma enorme rede *adhoc*. Isso só é possível com comunicação *device-to-device*(D2D) que permite que dois dispositivos próximos se comuniquem entre si numa rede licenciada de celular sem uma estacão base(BS) e por isso podemos dizer que esta tecnologia é revolucionaria e sai totalmente dos padrões convencionais da arquitetura de redes celulares. Em (TEHRANI; UYSAL; YANIKOMEROGLU, 2014) é citado que tecnologias como *wifi* e *bluetooth* fornecem um tipo de comunicação D2D porém trabalham em frequências não licenciadas e a interferência é incontrolável, além disso, não conseguem fornecer nenhuma segurança e QoS como as redes de celular conseguem.

Segundo o site (W3II.COM, 2014) a tecnologia 5G se diferencia das outras tecnologias não somente pelo aumento da taxa de bits para transferência de dados, mas também, ela aumentaria o volume de dados por unidade de área que poderão ser transferidos, permitiria maior conectividade a dispositivos, promoveria menor consumo de bateria assim

como forneceria maior confiabilidade na comunicação entre dispositivos.

Ainda segundo (W3II.COM, 2014), a arquitetura 5G é altamente avançada. A capacidade de escalabilidade dos sistemas é baseada na tecnologia de radio cognitivo que inclui varias características importantes tais como a capacidade de dispositivos de identificar a sua localização geográfica, tempo, temperatura dentre outros.

Segundo (WANG et al., 2014), a implantação do sistema 5G trará muitos desafios e para que os objetivos sejam alcançados será necessária uma drástica mudança no *design* da arquitetura das redes de celular. Ainda em (WANG et al., 2014), a otimização da medição de performance das redes e o gerenciamento de interface ainda serão desafios a serem enfrentados no futuro.

Generation	Speed	Technology	Time period	Features
1G	14.4 Kbps	AMPS,NMT, TACS	1970 – 1980	During 1G Wireless phones are used for voice only.
2G	9.6/ 14.4 Kbps	TDMA,CDMA	1990 to 2000	2G capabilities are achieved by allowing multiple users on a single channel via multiplexing. During 2G Cellular phones are used for data also along with voice.
2.5G	171.2 Kbps 20-40 Kbps	GPRS	2001-2004	2.5G the internet becomes popular and data becomes more relevant.2.5G Multimedia services and streaming starts to show growth. Phones start supporting web browsing though limited and very few phones have that.
3G	3.1 Mbps 500-700 Kbps	CDMA 200 (1xRTT, EVDO) UMTS, EDGE	2004-2005	3G has Multimedia services support along with streaming are more popular. In 3G, Universal access and portability across different device types are made possible. (Telephones, PDA's, etc.)
3.5G	14.4 Mbps 1-3 Mbps	HSPA	2006 – 2010	3.5G supports higher throughput and speeds to support higher data needs of the consumers
4G	100-300 Mbps. 3-5 Mbps 100 Mbps (Wi-Fi)	WiMax LTE Wi-Fi	Now (Read more on Transitioning to 4G)	Speeds for 4G are further increased to keep up with data access demand used by various services. High definition streaming is now supported in 4G. New phones with HD capabilities surface. It gets pretty cool. In 4G, Portability is increased further. World-wide roaming is not a distant dream.
5G	Probably gigabits	Not Yet	Soon (probably 2020)	Currently there is no 5G technology deployed. When this becomes available it will provide very high speeds to the consumers. It would also provide efficient use of available bandwidth

Tabela 1 – Gerações da tecnologia Móvel(MIR; KUMAR, 2015)

2.2.1 Tipos de Conexão D2D que poderão ser utilizados no 5G

Neste capítulo serão apresentadas 4 topologias de comunicação D2D que podem ser utilizadas na comunicação entre dispositivos numa rede 5G segundo (TEHRANI; UYSAL; YANIKOMEROGLU, 2014).

Como já foi citado anteriormente, as redes 5G trabalharão com comunicação direta entre dispositivos e por isso são necessárias formulações de topologias de rede para suportar essa comunicação. Quatro desses modelos são:

DE-CO - Dispositivo enlace com estabilização de link controlado pela operadora de celular.

CD-CO - Comunicação direta D2D com estabilização de link controlado pela operadora.

DE-CD - Dispositivo enlace com estabilização de link controlado no proprio dispositivo.

CD-CD - Comunicação direta D2D com estabilização de link controlado no próprio dispositivo.

No modelo DE-CO um dispositivo na borda da cobertura da antena ou numa área com baixa intensidade de sinal pode se comunicar com o BS através da comunicação com dispositivos enlaces, isso permitiria ao dispositivo maior QoS e maior duração de bateria. A operadora se comunica com os dispositivos enlaces para estabilizar parcialmente ou por completo a conexão. Este modo de operação é ilustrado na Figura 1.

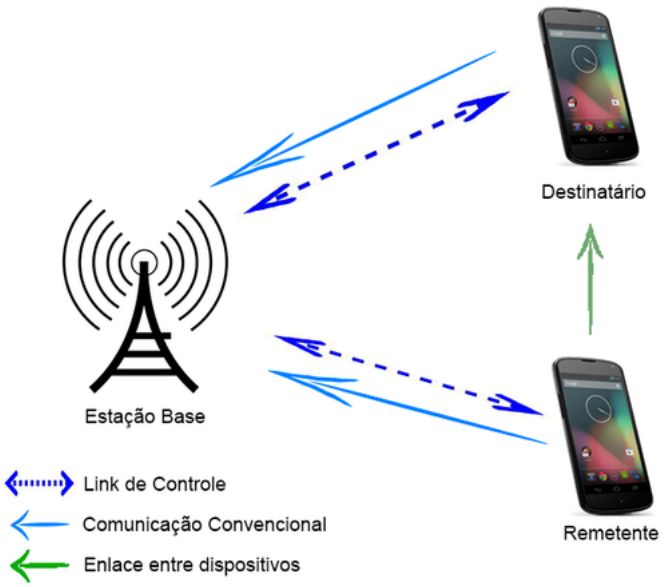

Figura 1 – Modelo DE-CO, reproduzido de (TEHRANI; UYSAL; YANIKOMEROGLU, 2014)

No modelo CD-CO o dispositivo de destino e o dispositivo remetente da informação se comunicam diretamente sem a necessidade do BS, porém ainda estarão ligados a operadora para estabilização do link de comunicação. Este modo de operação é ilustrado na Figura 2.

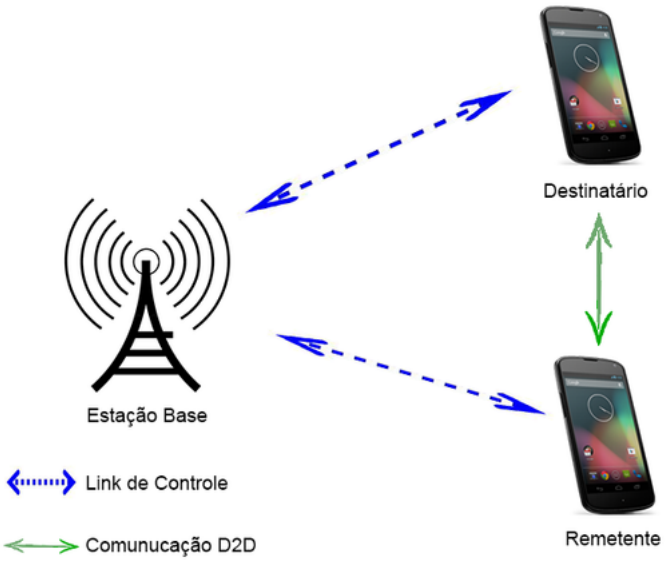

Figura 2 – Modelo CD-CO, reproduzido de (TEHRANI; UYSAL; YANIKOMEROGLU, 2014)

No modelo DE-CD a operadora não esta envolvida na estabilização da conexão e os dispositivos remetente e destinatário estão totalmente responsáveis por coordenar a comunicação utilizando dispositivos enlace entre eles. Este modo de operação é ilustrado na Figura 3.

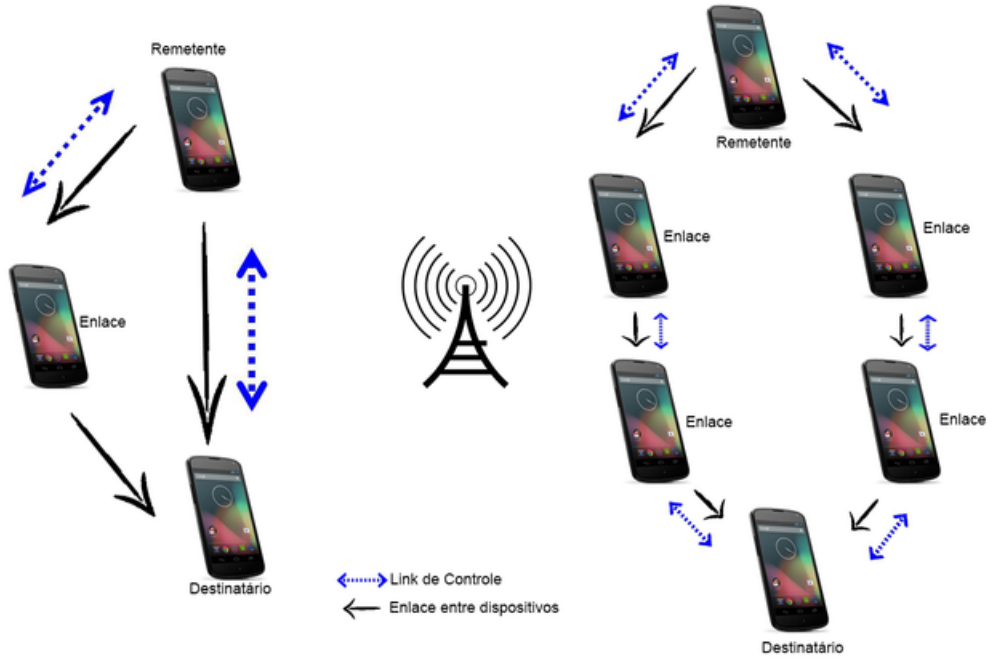

Figura 3 – Modelo DE-CD, reproduzido de (TEHRANI; UYSAL; YANIKOMEROGLU, 2014)

No modelo CD-CD o remetente e o destinatário se comunicam diretamente sem nenhum controle da operadora. Com isso, o remetente e o destinatário devem assegurar que exista interferência limitada entre sua comunicação utilizando recursos próprios. Este modo de operação é ilustrado na Figura 4.

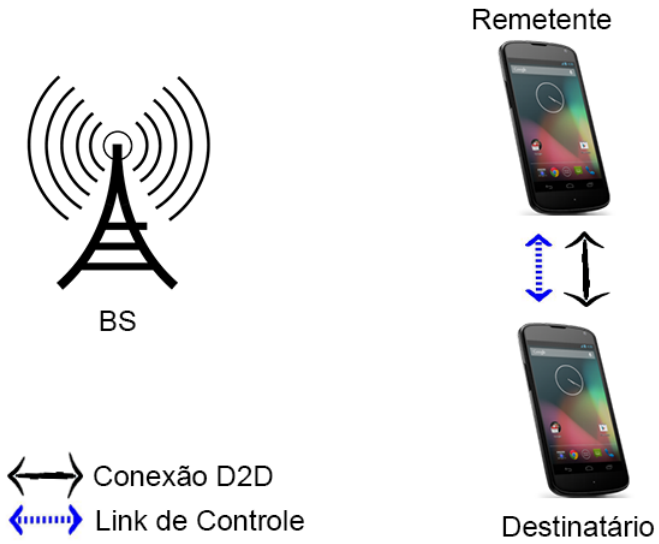

Figura 4 – Modelo CD-CD, reproduzido de (TEHRANI; UYSAL; YANIKOMEROGLU, 2014)

Estes modelos sugerem formas de comunicação entre os dispositivos de maneira eficaz. Porém, eles não asseguram a confiabilidade de conexão entre os dispositivos, somente apresentam uma forma para eles se conectarem, e opções para essa conexão.

3 Metodologia

A metodologia utilizada para a criação deste documento foi inteiramente bibliográfica. Foram necessárias pesquisas em artigos acadêmicos voltados para o assunto redes *wireless*, 5G, geração de tecnologia móvel, dentre outros. Pesquisas tiveram de ser realizadas também para definir qual o melhor ambiente para desenvolvimento e qual a melhor linguagem de programação a ser utilizada para a criação da solução.

A seguir, serão apresentados o ambiente de desenvolvimento e as ferramentas utilizadas e selecionadas após as pesquisas para a construção deste trabalho.

3.1 Ambiente de Desenvolvimento

3.1.1 Configurações do Ambiente de Desenvolvimento

O Algoritmo DSniffer foi desenvolvido em uma estação de trabalho com a seguinte configuração:

- Processador : Intel Core I3,
- Memoria RAM: 4GB,
- Placa de Video: Intel HD Graphics,
- Armazenamento: Disco Rigido 320GB,
- Sistema Operacional : Ubuntu Linux 16 LTS,
- Editor de Código: Visual Studio Code,
- Linguagem : Python 2.7,
- Ambiente de Testes: Inicialmente Mininet WIFI , Posteriormente VM Virtual Box.

3.1.2 Preparação do Ambiente

3.1.2.1 Mininet-Wifi

O Mininet-Wifi, (FONTES et al., 2015), é uma extensão do conhecido simulador de Redes SDN que tem como objetivo possibilitar ao desenvolvedor simular redes sem fio, recurso que o Mininet original não permite. Este software permite configurar *hosts* em redes wifi de forma personalizada, setando variáveis principais como tipo de encriptação,

velocidade de transmissão, tipo de rede(Se wifi utilizando roteador, adhoc, mesh dentre outras) dentre vários outros recursos.

Este software é citado em (FONTES et al., 2015), cujo autor deste artigo é também o idealizador do projeto.

Mininet Wifi ainda é um software em fase de aprimoramento, porém os recursos existentes o tornam um poderoso simulador de redes definidas por software.

No decorrer do estudo o uso do *Mininet Wifi* foi importante, porém este uso foi desconsiderado em função de este software não conseguir, num primeiro momento, fornecer informações vitais para o funcionamento do algoritmo a ser desenvolvido. Com isso, foram necessários buscar outros recursos para a criação da rede para testes.

3.1.2.2 VM VirtualBox

A solução encontrada para a criação da rede de testes foi a utilização do software *VM VirtualBox*. O *VirtualBox* é um software livre multi-plataforma que permite efetuar virtualização de um ou mais sistemas operacionais em um computador com Sistema Operacional (SO) Windows (XP ou superior), Linux, Macintosh ou Solaris,(PRADO, 2012).

Neste software serão criados os *hosts* para executar os testes do algoritmo que foi implementado e apresentado no capitulo seguinte.

3.1.2.3 Configuração dos hosts

Cada *host* teve de ser configurado para o funcionamento correto do algoritmo que utiliza bibliotecas baseadas em algumas ferramentas.

Inicialmente a ferramenta NMAP teve de ser instalada em todos os *hosts* para o funcionamento do algoritmo.

NMAP é um software muito utilizado para avaliar a segurança dos computadores, com ele é possível descobrir serviços disponíveis em um *host* através de uma rede de computadores. Neste estudo, este software será usado para descobrir os *hosts* dentro de um segmento de rede.

Para a comunicação entre os *hosts* e transmissão de informação, utiliza-se a biblioteca socket da linguagem python. Socket é uma biblioteca que possibilita o trabalho e a comunicação utilizando redes em python.

3.1.2.4 Parâmetros de Avaliação

Alguns parâmetros em cada *host* deverão ser avaliados. Dentre eles estão os parâmetros de segurança:

- Chave de Encriptação

  Segundo (CRUZ, 2009), criptografia é uma técnica ou conjunto de técnicas que possibilita, com o emprego de chave cifradora/decifradora, tornar ininteligíveis textos em claro e, inversamente, tornar inteligíveis textos cifrados, de forma a proteger a informação contra acesso não autorizado ao seu conteúdo. Uma chave cifradora ainda segundo (CRUZ, 2009), é um método que permite decifrar ou decodificar uma informação de forma que somente o remetente e o destinatário poderão acessar esse dado.

- Tipo de Autenticação

  No Artigo (CARRIÓN, 2005), existem 3 tipos mais utilizados de autenticação, WEP, WPA e WPA2.

  - WEP - Segundo (CARRIÓN, 2005) este protocolo não é recomendado por ser sucessível a vários ataques em função de sua fraca capacidade de autenticação que considera a validação de máquinas e não de usuários. É um protocolo de segurança para o padrão 802.11 da IEEE.
  - WPA - Ainda segundo (CARRIÓN, 2005), o protocolo WPA também utilizado para o padrão 802.11 da IEEE, foi uma resposta ao insucesso do protocolo WEP. O objetivo do WPA é prover mecanismos de autenticação, confidencialidade e integridade utilizando algoritmos mais robustos e uma arquitetura mais adequada.
  - WPA2 - Em (SILVA, 2010), é citado que WPA2 tem como principal objetivo suportar as características adicionais de segurança do padrão 802.11i que não estão incluídas nos produtos que suportam WPA. Assim como o WPA, o WPA2 provê autenticação e criptografia, propondo a garantia de confidencialidade, autenticidade e integridade em redes Wi-Fi. Sendo assim, WPA2, é um melhoramento de WPA que agrega melhor desempenho e segurança a autenticação em redes sem fio.

- WPS:

  A função WPS trás facilidade na conexão entre dispositivos evitando que por exemplo um senha tenha de ser digitada toda vez em que um novo dispositivo acessa um enlace. O uso desta tecnologia pode trazer inúmeros problemas de segurança para uma rede.

- Nível de Bateria:

  O nível de bateria de um dispositivo é uma variável indispensável a ser analisado. Um dispositivo que apresenta nível de bateria muito baixo, deve ser desqualificado para a transmissão de dados considerando que esta transmissão pode ser interrompida

em função de falta de carga do dispositivo e com isso a informação não chegará ao seu destino.

3.1.2.5 Linguagem

A linguagem escolhida para a implementação da solução foi o Python por ser uma linguagem de programação de aprendizado rápido e de entendimento e logica simples.

4 Desenvolvimento

Inicialmente, foram idealizados cenários que simulam as possibilidades para a informação trafegar entre os dispositivos com a intenção de melhor compreender o funcionamento e as possibilidades de conexão entre os nós em uma rede com as características do 5G. Estes cenários são:

Cenário 1

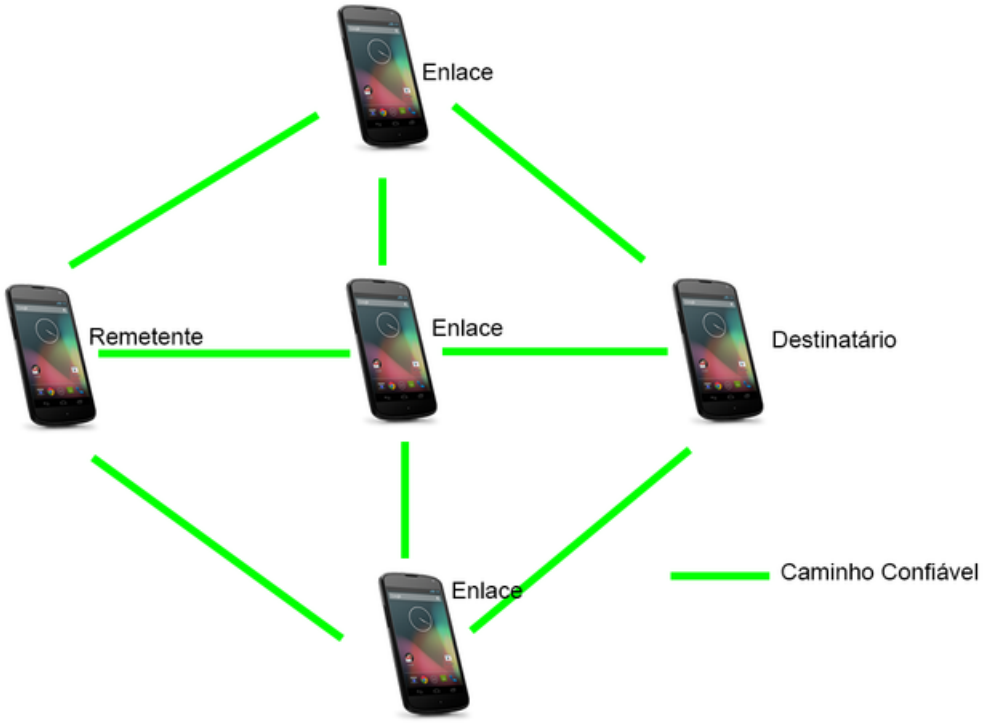

Figura 5 – Cenário 1

O primeiro cenário, nos mostra uma rede de dispositivos interligados entre si onde todos os nós da rede tem como confiáveis os seus vizinhos. Este caso pode ser tratado como o melhor caso, onde a informação pode ser transmitida através de qualquer nó, até qualquer nó com um nível de confiança alto.

Cenário 2

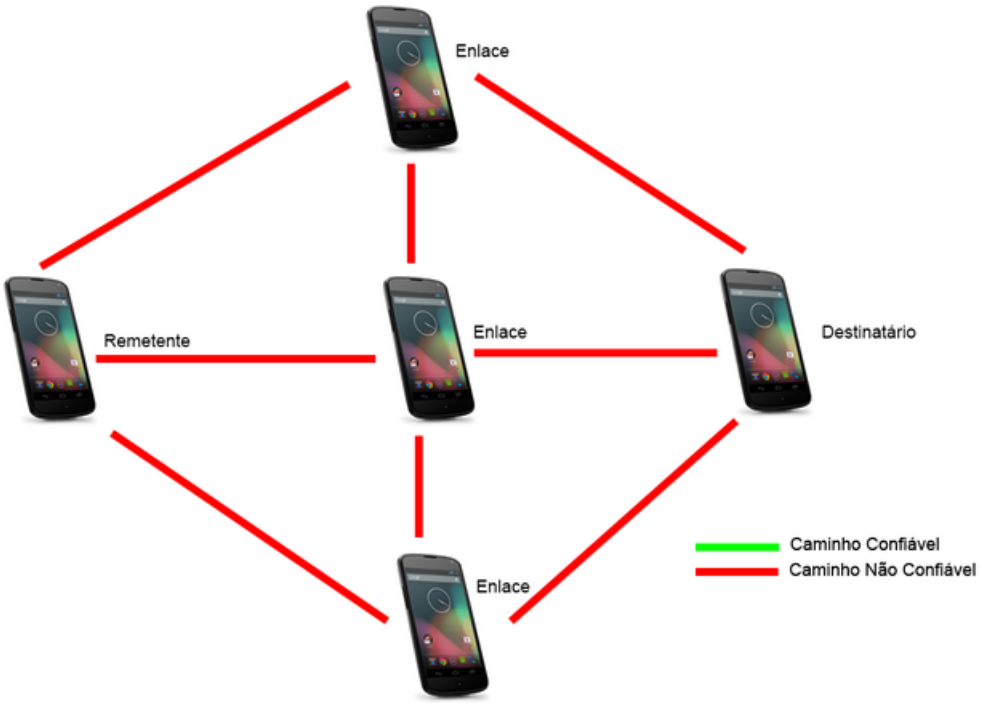

Figura 6 – Cenário 2

No cenário 2, consideramos como a pior situação, pior caso, que um nó na rede pode se encontrar. Nenhum dos seus vizinhos é considerado confiável. Com isso, a conexão fica inviável e a informação, inicialmente, não pode ser transmitida.

Cenário 3

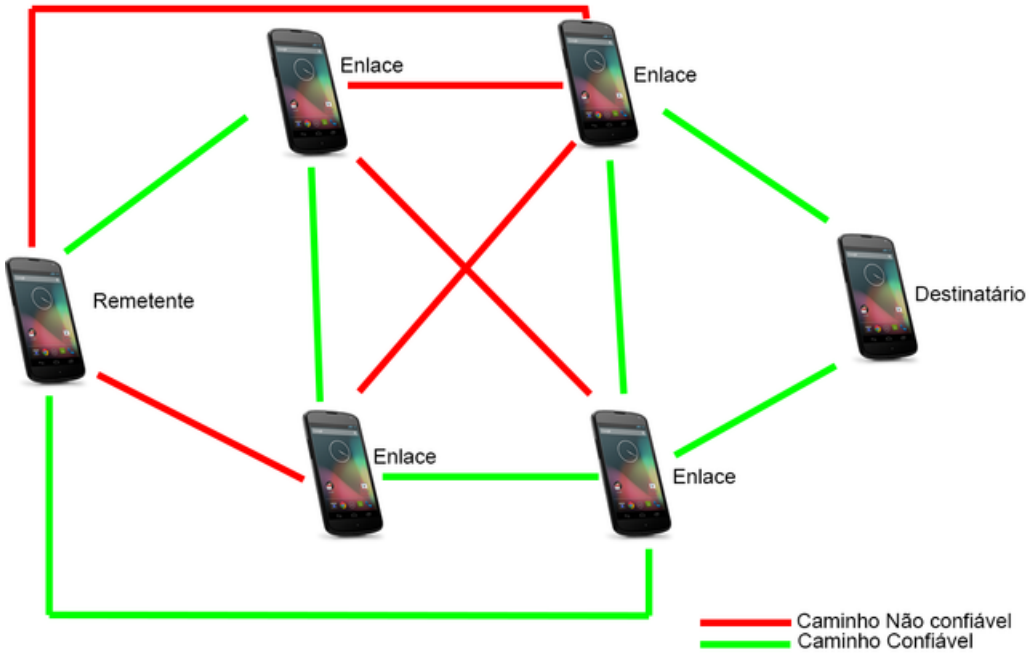

Figura 7 – Cenário 3

A abordagem do cenário 3 pode ser considerada como um caso médio onde existem vários caminhos por onde a informação pode trafegar. Com isso, o nó com maior nível de confiabilidade seria o transmissor da informação até seu destino, ou para o nó subsequente que continuará com o processo de transmissão da informação.

Cenário 4

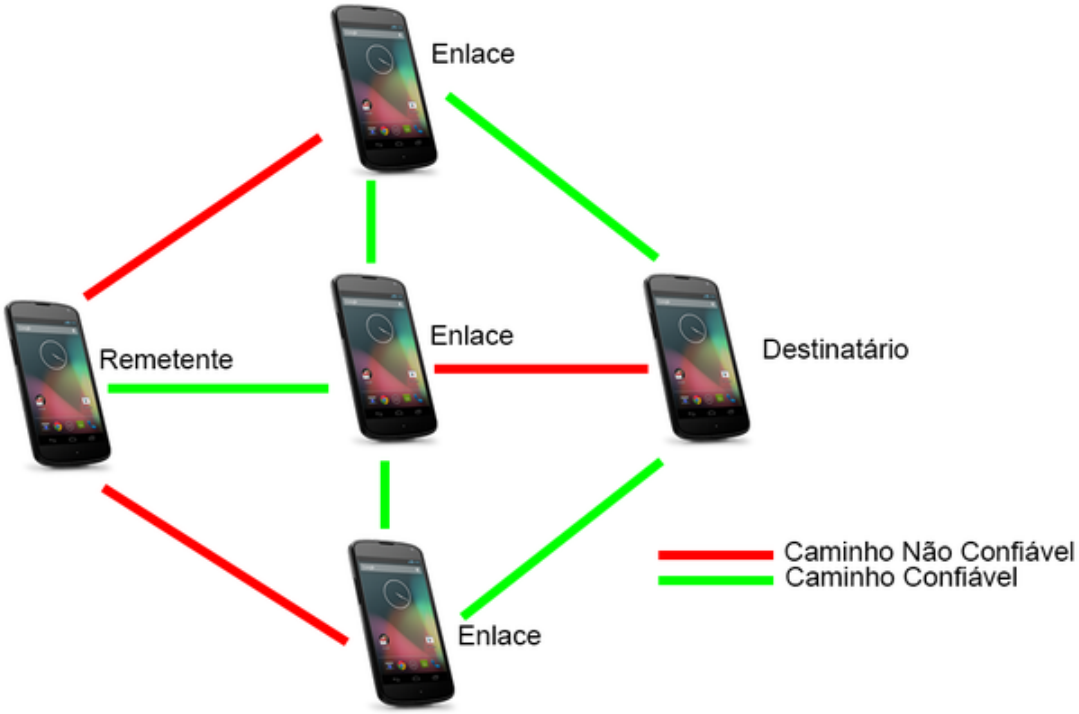

Figura 8 – Cenário 4

Por ultimo consideramos um caso especial, mas não incomum, onde o nó poderá ter um enlace confiável a sua frente, porém, os enlaces subsequentes não são confiáveis para este primeiro nó, porém, são confiáveis para seu vizinho. Este caso, que pode ocorrer facilmente dentro dos outros cenários e será o alvo principal de tratamento deste trabalho.

4.1 Algortimo *DSniffer*

Objetivando resolver os problemas apresentados anteriormente, foi idealizado um algoritmo batizado de *DSniffer*, abreviação de *Device Sniffer*, que tenta encontrar o melhor caminho para o tráfego das informações entre dispositivo remetente e dispositivo destinatário. *DSniffer* se utiliza de métricas para assegurar que a informação trafegará por dispositivos confiáveis. As métricas avaliadas pelo algoritmo são:

- Latência de conexão: Este parâmetro trata a velocidade com a qual o *host* destino responde um requisição vinda do *host* remetente.
- Nível de Sinal: Este parâmetro verifica se o nível de sinal do *host* destino é satisfatório. Varia entre -30dBm e -90dBm onde, o valor -30 é ótimo e -90 é descartável.
- Nível de Bateria: Neste parâmetro será avaliado o nível da bateria do *host* que esta a ser avaliado. Quanto maior o valor da carga melhor para a avaliação do *host*.
- Se existem portas importantes abertas no *host*(porta 22 como exemplo)
- Se o *host* está ativo na rede: Caso o *host* não retorne um valor *up*, este não é considerado para as iterações.
- Se todas as portas do *host* estão fechadas.
- Qualidade do Sinal : Neste parâmetro é avaliada a qualidade do sinal do *host*. É um parâmentro equivalente ao de nível de sinal, porém dado em porcentagem. Quanto mais próximo de 100% melhor.
- Se este *host* utiliza Chave de Encriptação.
- Qual tipo de autenticação utilizada: Este parâmetro avalia se o *host* utiliza autenticação dos tipos WEP, WPA ou WPA2.
- Se o *host* tem WPS habilitado ou não.

Estes parâmetros deve representar valores ideais de trabalho assegurando assim um nível de confiabilidade alto dentre os nós e criando *clusters* de nós confiáveis. Vale ressaltar que este algoritmo considera um quadro estático apesar de, no mundo real, os dispositivos estarem em constante movimentação.

O termo confiabilidade continuamente será citado neste trabalho. Este é o alvo principal do algoritmo *DSniffer*.

Considerando o tráfego de informações, um sistema ou dispositivo é considerado confiável se ele tem a capacidade de repassar informações da origem ao destino sem interferências de fontes externas. Em redes de dispositivos pode-se acrescentar ainda

variáveis como alta velocidade de transmissão e tempo de resposta como variáveis de confiabilidade. Um dispositivo não confiável, neste contexto, é um dispositivo que não tem os requisito mínimos para transmitir informações de um *host* até outro. Visando este conceito, foi desenvolvido o *DSniffer*.

4.2 Desenvolvimento do Algoritmo

O desenvolvimento do algoritmo *DSniffer* inicialmente tentou traçar melhores formas de mensurar a confiabilidade da conexão entre os dispositivos. A ideia do desenvolvimento de métricas foi crucial para o sucesso da abordagem.

```
1  Receba um endereco de IP.
2  Escaneie a sua volta todos os dispositivos dentro desta mesma faixa de IPs
       dado um determinado intervalo de tempo.
3    Se forem encontrados dispositivos:
4      Invoque uma funcao que mensure os parametros especificos de cada
       dispositivo encontrado.
5      Se algum dispositivo se enquadrar dentro das condicoes consideradas
       aceitaveis:
6        Se este dispositivo for seu destino:
7          Conecte-se a ele e transmita a informacao
8        Se este dispositivo nao for seu destino:
9          Conecte-se a Ele , transmita a informacao e reinicie o processo
       DSniffer nele.
10     Repita isso em todos os nos ate que cheguemos ao destino.
11     Se nenhum dispositivo se enquadrar dentro das condicoes consideradas
       aceitaveis:
12       Reinicie a Busca.
13   Se nao forem encontrados dispositivos:
14     Reinicie a Busca.
```

O trecho de código descrito acima, mostra superficialmente o funcionamento do algoritmo *DSniffer*.

4.3 Detalhamento do Algoritmo

Nesta seção serão descritas as funções do algoritmo *DSniffer* por completo. As métricas utilizadas também serão apresentadas.

O código fonte deste algoritmo será disponibilizado em um link da plataforma *GitHub* em seção anexa a este trabalho.(Anexo I).

DSniffer foi dividido em 2 códigos diferentes que devem trabalhar paralelamente nos dispositivos.

- *DSnifferClient*,
- *DSnifferServer*.

DSniffer trata todos os dispositivos como clientes/servidores que transmitem e recebem informações entre si. Serão discutidas as duas instâncias do algoritmo citadas acima nas subseções seguintes.

4.3.1 DSnifferClient

Esta instância cliente do algoritmo *DSniffer* tem como objetivo buscar os melhores caminhos para a informação ser transmitida. Este funciona como o remetente da informação e escaneia os nós vizinhos em busca de melhores valores de confiabilidade. O dispositivo executando esta instância será o emissor da informação.

Como em toda aplicação, *DSniffer* necessita de bibliotecas que fornecem funções pré-definidas para o funcionamento do algoritmo.

Como este trabalho é focado em redes, precisou-se utilizar bibliotecas voltadas diretamente para este assunto como *NMAP* e *Socket*.

Toda aplicação para ser instanciada precisa de uma função principal, que invoca os métodos e *threads* operárias que executam os procedimentos para o funcionamento do algoritmo. Com *DSniffer* não é diferente, por isso, também é necessária a função *main*. A função *main* invoca a função *sniffcalc()*. A função *sniffcalc()* tem como objetivo calcular o valor do *SniffNumber*.

- *SniffNumber* é o nome dado a variável que foi definida para receber o valor final de confiabilidade do dispositivo que está sendo analisado, uma espécie de nota. O valor desta variável é baseado nas métricas e define se o dispositivo em questão é ideal ou não para a transmissão da informação. Com isso, ela se torna o item mais importante do algoritmo *DSniffer*.

Foram definidos um numero limitado de hosts para testes, nove em seu total. A função *sniffcalc()* utiliza-se de um recurso do sistema operacional Ubuntu Linux chamada *NMAP* para escanear toda a rede dentro de uma faixa de IPs especifica em busca de *hosts* ativos. Quando é passado o comando *NMAP.PortScanner()* com uma faixa de IPs especificada, neste caso, 192.168.56.0/24, todos os *hosts* ativos nessa fatia de rede são retornados em forma de uma lista, aqui chamada de *hosts_list*. A faixa de IPs citada acima foi considerada para fins de testes, porém num cenário real, a abrangência dos endereços deverá ser de uma rede de Internet.

Em seguida, é executado um *Loop* do tipo *FOR* em *hosts_list* com a intenção de percorrer todos os nós executando testes.

Ao encontrar o primeiro nó presente em *hosts_list* iniciam-se as medições de desempenho. Inicialmente é executado um *ping* no nó encontrado e seu retorno contém dentre outros valores, o tempo de resposta de uma requisição e seu retorno entre o nó remetente e o nó destino, este valor é armazenado em *sniffnum*. Na sequencia, são verificados quantos *hosts* estão ativos na rede no momento. Este valor é armazenado em *online_hosts*.

Após o armazenamento das variáveis supracitadas em tempo de execução,é chamada a função *CalculateSniffNumber(sniffnum,host,online_hosts)*. A função *CalculateSniffNumber(sniffnum,host,online_hosts)* será detalhada posteriormente. Esta função tem o objetivo de calcular o valor de *SniffNumber* e fornecer a nota final do *host* a uma segunda função que foi chamada de *Path(SniffCalculated,host)* e também será detalhada posteriormente. *Path(SniffCalculated,host)* recebe o valor final de nota dada ao *host* baseado em métricas que serão apresentadas no decorrer deste detalhamento. Esta função tem como objetivo, com base na nota recebida, dizer se o *host* em questão pode ser utilizado para trafegar a informação ou não.

Estas verificações feitas em *sniffcalc()* bem como as chamadas de todas as funções previamente descritas são feitas para todos os *hosts* que foram encontrados ativos na rede até que um deles seja apto a transmitir a informação. Vale lembrar que um *host* considerado ativo, é aquele *host* que retorna o valor *up* juntamente com seu endereço de IP quando é disparada a função *NMAP.PortScanner()*.

Em *CalculateSniffNumber(sniffnum,hostname,online_hosts)*, inicialmente recebemos a variável *sniffnum*, um *hostname*, e a lista de *hosts* disponiveis chamada de *online_hosts*.

Inicialmente é obtida o valor da latência, armazenado na variável *latency* e imediatamente verificado se esse valor é 0. Após essa verificação, é calculada a primeira parte do valor final da nota do *host*, chamada aqui de *SniffNumber*, utilizando um valor resultante da divisão da variável *latency*, pelo valor da variável *online_hosts*(tamanho da lista de

hosts), desse modo, o valor da fase 1 do calculo da nota do *host* é concluída

Em seguida, é iniciada a fase 2. Nesta fase, são invocadas duas funções, *GetParams* e *NmapChecking* onde o valor resultante das verificações feitas na primeira é armazenado na variável *sniffParams*, e o valor resultante das verificações da segunda é armazenado em *sniffPoints*. As duas funções serão detalhadas posteriormente. Imediatamente após as duas funções citadas anteriormente retornarem seus resultados, são feitas verificações baseadas nas variáveis que armazenam esses retornos para verificar não retorna valores vazios. Se vierem vazias, é atribuído o valor zero. A variável *SniffNumberPart2* armazena a soma dos retornos das funções *GetParams* e *NmapChecking*, que são valores inteiros, e com isso é concluída a fase dois de verificações.

O valor final , soma de *SniffNumberPart1* com *SniffNumberPart2* nos dá a nota, *SniffNumber*, do *host* em questão. Esta nota é avaliada na função *Path()* que será apresentada posteriormente.

As funções *NmapChecking, GetParams,IwChecking e IwListGenerica* são as funções de maior importância deste algoritmo. Essas funções são responsáveis por calcular a nota, *SniffNumber*, baseada em métricas definidas.

Essas métricas são notas que podem variar entre os valores três e menos três em relação a cada parâmetro e seu estado atual. É importante ressaltar que essas notas podem ser alteradas a qualquer momento fazendo com que o algoritmo seja personalizável. Abaixo serão citadas as funções que avaliam parâmetros e como estes parâmetros são avaliados.

Função *NMapChecking*:

- Se as portas de conexão do *host* estão fechadas.
  - Recebe +1 se *True*, Se *false* recebe 0.
- Se a porta SSH(22) está aberta,
  - Recebe -1 se *True*, Se *false* Recebe 0.
- Se o *Host* Está Online
  - Recebe +1 se *True*.

Os parâmetros para avaliação na função *NMapChecking* são obtidos através de um comando *NMAP* + (Endereço do *Host* em Questão), e a resposta desse comando é filtrada para avaliarmos somente o parâmetros citados acima.

Função *IwChecking*:

- Qual o nível de bateria do *host*,
  - Se Acima de 50% recebe +3.
  - se abaixo de 50% e acima de 20% recebe +1.
  - se abaixo de 20% recebe -3.
- Qual a qualidade de sinal do *host*,
  - Se tem a qualidade de sinal acima de 75% e abaixo de 100% recebe +2.
  - Se tem a qualidade de sinal acima de 45% e abaixo de 75% recebe +1.
  - Se tem a qualidade de sinal entre 20% e 45% recebe -1.
  - Se tem a qualidade de sinal entre 0% e 20% o *host* é descartado.
- Qual o nível do sinal do *host*,
  - Se o nível de sinal apresentar um valor de 0db a 100db , o dispositivo é descartado pois é provável que haja algum erro.
  - Se o nível de sinal variar de -30dBm a -67dBm, recebe +3.
  - Se o nível de sinal variar de -68dBm e -80dBm, recebe +1.
  - Se o nível de sinal variar de -81dBm a -90dBm, recebe -3.
  - Se o nível de sinal for menor que -90dBm o dispositivo é descartado.
- Se o *host* usa chave de encriptação,
  - Se o valor de *Encryption key* é *on* recebe +1.
  - Se o valor de *Encryption key* é *off* recebe -1.
- Qual o tipo de autenticação utilizada pelo host(WEP, WPA, WPA2),
  - Se o valor de *Auth Type* é "wpa"recebe +1.
  - Se o valor de *Auth Type* é "wep"recebe -3.
  - Se o valor de *Auth Type* é "wpa2"recebe +3.
- Se o host Utiliza WPS
  - Se o valor de *"Has Wps"* é *on* recebe -1.
  - Se o valor de *"Has Wps"* é *off* recebe +1.

Da mesma forma, os parâmetros para avaliação da função *IwChecing*, são obtidos do comando *iw list + host* em questão + *scanning*. No caso deste algoritmo que ainda é um protótipo, foi criada uma função *Iw List* Genérica que gera esses dados aleatoriamente para o ambiente de testes. Em um cenário real, somente será necessária a substituição a chamada da função *IwListGenerica* pela *Iw List* original.

As funções apresentadas anteriormente (*NmapChecking e IwChecking*) retornam um valor inteiro que é uma parte da nota ,*SniffNumber*. Essas partes de notas são somadas na função *CalculateSniffNumber* que já foi apresentada anteriormente.

Ainda existem mais duas funções de fundamental importância As funções *Path (SniffCalculated, Host)* e *Connection(host,destino)*, essas funções serão apresentadas abaixo:

Função *Path*

Esta função recebe o valor da nota do *host*, *SniffNumber*, e o endereço de IP do *host*, para efetuar verificações. Essas verificações são os passos que direcionarão o destino da informação, e se o *host* que esta a ser avaliado está apto para transmitir essa informação.

Nessa função verificamos se o valor da nota, *SniffNumber*, é ideal considerando um valor ideal de *SniffNumber* pré definido e também se o *host* em questão é o *host* destino. Caso este *host* tenha valor de *SniffNumber* ideal, e seja o *host* destino, a informação é diretamente entregue. Se o valor de *sniffnumber* for ideal, mas este não for o *host* destino, a informação será transmitida a este *host*, considerando-o confiável em função de sua nota, e neste *host* subsequente, será iniciado outro processo de verificação de *hosts* disponiveis e cálculo de *sniffnumber* utilizando o algoritmo *DSniffer* com o intuito de encontrar novas conexões até que o destino seja atingido. Se o valor de *sniffnumber* não for ideal, este *host* é automaticamente descartado e a avaliação é reiniciada.Se a lista de *hosts* terminar e nenhum *host* for encontrado com nota ideal, a avaliação é reiniciada.

Podemos sempre reiniciar a avaliação considerando que valores como, latência, nível de sinal e qualidade de sinal podem mudar e consequentemente melhorar a nota dos *hosts*.

Função *Connection()*

Esta função é uma função de conexão simples que utiliza *sockets* para se conectar a um *host* definido. Ela recebe um endereço de *host* e o destino. Este endereço de *host*, é o *host* que foi avaliado e certamente tem o *sniffnumber* ideal. A variável destino é o endereço do *host* destino, sendo o *host* subsequente, o destino final ou não. Esta condição é avaliada no *host* a ser conectado utilizando a instância de *DSnifferServer*.

4.3.2 DSnifferServer

DSnifferServer é a instancia do algoritmo *DSniffer* em modo servidor que aguarda conexões de *hosts* confiáveis. Se esta instância for chamada e a conexão foi bem sucedida,

quer dizer que este *host* foi considerado confiável pelo algoritmo e pode transmitir a informação adiante. Caso este seja o *host* destino, a informação é entregue nele. Se não, este *host* reinicia o processo, chamando uma instancia do algoritmo *DSnifferClient*, ja embutida em *DSnifferServer*.

Com isso finalizamos a apresentação das funções do algoritmo *DSniffer* e suas instâncias de servidor e cliente. Todo o algoritmo é funcional para um ambiente de testes porém ainda deve ser aprimorado se existir probabilidade de alocação de suas funções em um cenário real. Os testes poderão ser realizados em redes de internet e não somente em redes locais. No próximo capitulo serão apresentados testes feitos em um ambiente controlado.

5 Experimentos

Após a implementação do algoritmo que foi apresentado acima, testes de funcionamento e desempenho tiveram de ser realizados com a intenção de definir qual seria um valor médio ótimo para *SniffNumber*.

Para os testes foram utilizadas maquinas virtuais criadas no software aberto *VM VirtualBox*. Os *hosts* tinham a função de simular um dispositivo móvel e os parâmetros de rede sem fio são gerados pela função *IwListGenerica* do algoritmo *DSniffer* foram criados com a seguinte configuração:

- Sistema operacional Ubuntu Linux 14.04,
- 256Mb de memória principal,
- 2Gb de memória para armazenamento,
- Duas placas de rede, uma em modo NAT para manter a conexão com a internet ativa com o computador principal, e uma em modo hospedeiro)*host-only* criando uma rede de *hosts* isolada da rede da maquina principal.

Os testes podem ser feitos com a quantidade infinita de *hosts*. Para este cenário foram considerados nove *hosts* para facilitar o tratamento dos resultados e estes *hosts* representam dispositivos móveis em uma rede. Os testes foram feitos na ordem que segue:

1. Nenhum *host* disponível na rede além do remetente,
2. Somente dois *hosts* conectados na rede,
3. Somente quatro *hosts* conectados a rede,
4. Todos os nove *hosts* conectados a rede.

5.1 Testes

Seguindo a sequência citada acima, os testes efetuados serão apresentados a seguir. Os procedimentos para a execução dos testes serão apresentados em cada teste. Em cada *host* que não está inicialmente enviando uma informação é instanciado o algoritmo *DSnifferServer* pois estes estarão aguardando conexões.

5.1.1 Parâmetros Pré Definidos para os testes

- *Host* Remetente: "192.168.56.101",
- *Host* Destino: "192.168.56.103",
- *SniffNumber* Ideal

Parâmetro	Pior Caso	Melhor Caso	Caso Médio
Porta 22	Aberta	fechada	fechada
Demais Portas	Aberta	Fechada	fechada
Bateria	< 20%	> 50%	20%>X<60
Qualidade Sinal	20%<X>45%	75%>X<100%	45%<X>75%
Nível Sinal	-81dBm>X<-90dBm	-30db>X<-67dBm	-68dBm>X<-80dBm
Encriptação	Não	Sim	Sim
Autenticação	WEP	WPA2	WPA2
WPS	on	off	off
SniffNumber	-14+latência	15+latência	10+latência

Tabela 2 – Definição do SniffNumber Ideal

O sniffNumber ideal foi definido em função do caso médio que assim como o pior e o melhor caso é a soma das avaliações com a latência.

1. Nenhum *host* disponível na rede além do remetente:

   Um dos *hosts* é inicializado,neste caso,aleatóriamente,o *host* 192.168.56.101, e uma instancia de *DSnifferClient.py* é executada neste *host*. Ao iniciar, *DSniffer* verifica a rede e busca por *hosts* ativos encontrando somente o próprio host e o gateway da rede, que neste caso pode ser descartado considerando que estamos simulando uma rede móvel D2D em *hosts* no virtualBox.

   DSniffer retorna uma mensagem de "Fim das Tentativas"sinalizando que a lista de *hosts* está vazia e não foram encontrados dispositivos que pudessem transmitir a informação. Esse comportamento já era esperado, considerando que não temos nenhum dispositivo conectado a não ser o próprio remetente.

Figura 9 – Teste 1 - Nenhum host conectado além do remetente

2. Somente dois *hosts* conectados na rede:

   Além do *host* 192.168.56.101, foi iniciado mais um *host*, agora o host 192.168.56.104. No dispositivo com *IP* de final 101, é novamente instanciado o algoritmo *DSniffer-Cliente.py*, e a resposta obtida é diferente e já mostra o funcionamento do algoritmo em relação ao tratamento de *hosts*.

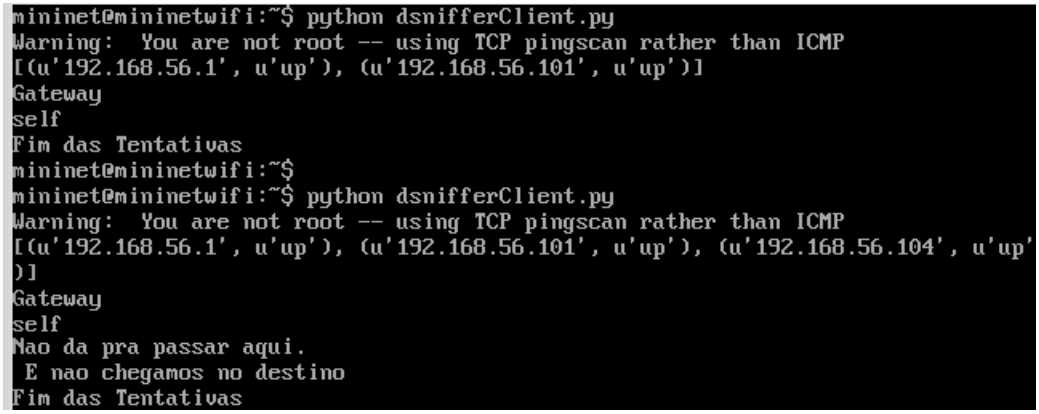

Figura 10 – Teste 2 - dois hosts conectados

   Neste segundo teste pode-se notar que a rede foi novamente percorrida, e um *host* adicional foi encontrado, o *host* 192.168.56.104, porém, este não é o *host* destino e o valor do *SniffNumber* calculado pelo algoritmo não foi satisfatório, esta afirmação é confirmada pela resposta do algoritmo que é mostrada na figura acima. Lembrando que as mensagens exibidas são somente para fins de testes, estas serão removidas para funcionamento em ambientes de produção.

3. Somente quatro *hosts* conectados a rede:

   São iniciados nesse momento mais 2 *hosts* na rede, com o intuito de testar o funcionamento de *DSniffer*. Os hosts de final 105 e 107 são iniciados.

   A demonstração de funcionamento do algoritmo, mais uma vez foi executada com sucesso. Neste caso, foi encontrado um *host* com o valor do *sniffnumber* ideal, porém, este era o dispositivo de final 104. Como foi convencionado que o dispositivo destino seria o *host* de final 103, a informação foi direcionada para o dispositivo de final 104 e o algoritmo *DSniffer* será reiniciado neste *host* buscando por novas conexões confiáveis.

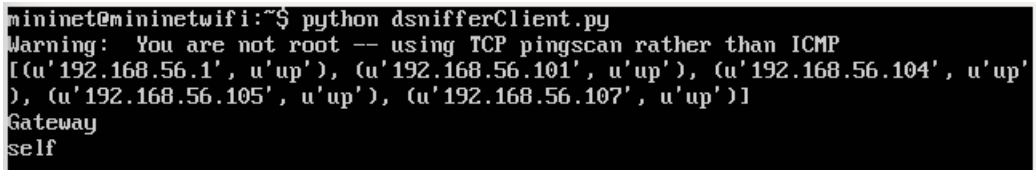

Figura 11 – Teste 3 - quatro hosts conectados

4. Todos os nove *hosts* conectados a rede: Para este teste, foram instanciados os nove *hosts* disponíveis para os testes.

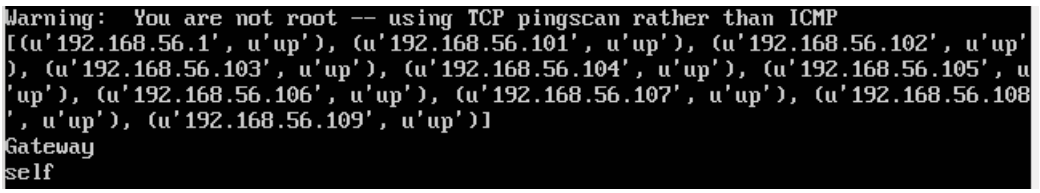

Figura 12 – Teste 4 - nove*Hosts* conectados.

Tendo como *host* remetente o dispositivo de final 101 e *host* destino o dispositivo de final 103, foi instanciado *DsnifferClient.py* em 101 e nos dispositivos restantes *DSnifferServer.py*. O resultado está na figura a seguir:

Figura 13 – *Host* destino de final 103 aguardando conexão com *DsnifferServer* instanciado.

No dispositivo de final 102, recebemos esta mensagem, pois, o algoritmo *DSniffer* encontrou um *SniffNumber* ideal para se conectar a este *host* porém verificou que o host de final 102 não é o seu destino.

Figura 14 – Mensagem mostrada em 102 por não ser o destino final da informação.

Pelo motivo de este não ser o motivo final, *DSniffer* é reiniciado em 102. E quando o destino final é alcançado as seguintes mensagens são exibidas em remetente e destinatário:

Figura 15 – Mensagem mostrada no remetente da informação se a conexão for um sucesso.

Figura 16 – Mensagem mostrada no destinatário da informação se a conexão for um sucesso.

6 Considerações Finais

Compreender a transmissão de dados em uma rede de dispositivos foi essencial para a implementação do algoritmo *DSniffer*. Seu funcionamento é completamente baseado na avaliação de *hosts* e troca de informações que utilizam métodos tradicionais de comunicação numa rede obtendo informações de dispositivos que estão em toda parte. Avaliar a confiabilidade dos enlaces por onde a informação deve passar será um diferencial importante na nova era de tecnologia de transmissão entre dispositivos móveis levando em conta que a necessidade de segurança que ja existe se tornará muito mais marcante com a implantação da nova abordagem de comunicação proposta pelo 5G.

O algoritmo *DSniffer* foi implementado com o intuito de conseguir se conectar e transmitir informações de um *host* a outro com maior confiabilidade na conexão entre estes dispositivos. Os testes apresentados no capitulo anterior, mostram que o funcionamento do algoritmo é satisfatório considerando o ambiente controlado no qual as instancias foram executadas. Para ambientes reais o algoritmo *DSniffer* ainda deve ser otimizado para suportar testes em ambientes diferentes do apresentado com exceção da modificação de redes locais para redes de internet.

Este algoritmo pode ser um passo para a otimização da confiabilidade entre os enlaces numa rede de tecnologia 5G. Mais estudos são necessários para aprimorar o funcionamento e melhorar a abrangência desta solução levando em conta métricas aprimoradas de avaliação de *hosts* e com isso melhorar a precisão das medições e maximizar a confiabilidade dos caminhos formados.

7 Trabalhos Futuros

Como trabalho futuro, deve ser executado o aprimoramento deste algoritmo para suportar ambientes dinâmicos, ou seja, ambientes que mudam de condições tanto de localização quanto de condições físicas das redes. Futuramente devem ser aprimorados os tratamentos para promoção ou não de um *host* como confiável. Esse aprimoramento pode ser feito adicionando mais variáveis relacionadas, por exemplo a seleção de qual canal será melhor para trabalho baseado na interferência ou congestionamento dos canais. Outra variável possível de se adicionar é a seleção de frequência para trabalho dentro dos canais disponíveis. Um outro fator importante a ser estudado mais a fundo é a definição do *SniffNumber* ideal para promoção do *host*.

Referências

ANDREWS, J. G. et al. What will 5g be? *IEEE Journal on selected areas in communications*, IEEE, v. 32, n. 6, p. 1065–1082, 2014. Citado na página 15.

CARRIÓN, D. d. S. D. *Avaliação de protocolos de autenticação em redes sem fio*. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2005. Citado na página 25.

COMPUTERHISTORY.ORG. *Computer History*. 2015. Www.computerhistory.org/internethistory/ - Acessado em 02/03/2017. Citado na página 12.

CRUZ, E. F. D. A criptografia e seu papel na segurança da informção e das comunicações (sic)–retrospectiva, atualidade e perspectiva. 2009. Citado na página 25.

FONTES, R. R. et al. Mininet-wifi: Emulating software-defined wireless networks. In: IEEE. *Network and Service Management (CNSM), 2015 11th International Conference on*. [S.l.], 2015. p. 384–389. Citado 2 vezes nas páginas 23 e 24.

ITPORTAL.IN. *It Portal*. 2014. Www.itportal.in, Acessado em 05/12/2016. Www.itportal.in, Acessado em 05/12/2016. Citado na página 15.

MENDES, J. R. R. 5g: a quinta geração. Universidade Tecnológica Federal do Paraná, 2014. Citado 3 vezes nas páginas 15, 16 e 17.

MIR, M.; KUMAR, S. Evolution of mobile wireless technology from og to 5g. *International Journal of Computer Science and Information Technologies*, v. 6, n. 3, p. 2545–2551, 2015. Citado 2 vezes nas páginas 8 e 18.

PRADO, F. *Instalando o Oracle VM Virtual Box*. 2012. Disponível em: <http://www.fabioprado.net/2012/01/instalando-o-oracle-vm-virtual-box.html>. Citado na página 24.

SILVA, L. R. Segurança em redes sem fio (wireless). *Pontifícia Universidade Católica do Paraná*, 2010. Citado na página 25.

TEHRANI, M. N.; UYSAL, M.; YANIKOMEROGLU, H. Device-to-device communication in 5g cellular networks: challenges, solutions, and future directions. *IEEE Communications Magazine*, IEEE, v. 52, n. 5, p. 86–92, 2014. Citado 6 vezes nas páginas 7, 17, 19, 20, 21 e 22.

THIRUPATHI, R.; ANGELINROSY, M. Generation of mobile wireless 1g to 5g technology. 2012. Citado 2 vezes nas páginas 15 e 16.

W3II.COM. *5G Quick Guide*. 2014. Http://www.w3ii.com/pt/5g/5g_quick_guide.html, Acessado em 26/02/2017. Citado 2 vezes nas páginas 17 e 18.

WANG, C.-X. et al. Cellular architecture and key technologies for 5g wireless communication networks. *IEEE Communications Magazine*, IEEE, v. 52, n. 2, p. 122–130, 2014. Citado na página 18.

Anexos

Anexo A

Link para o algoritmo *DSniffer* em código aberto no GitHub.

- https://github.com/rafaeloliveira29/Dsniffer - GitHuB Rafael Souza Oliveira - Algoritmo *DSniffer*