Universidade Federal do ABC
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Engenharia de Informação

Felipe Roberto Martins de Andrade

Testes de desempenho de redes 4G baseadas em IPv6

Trabalho de Graduação III

Santo André – SP

Testes de desempenho de redes 4G baseadas em IPv6

Felipe Roberto Martins de Andrade

Relatório submetido como requisito parcial para obtenção do grau de bacharel em Engenharia de Informação

Orientado por Prof. Dr. Claudio José Bordin Júnior

Santo André – SP

Agosto de 2023

Resumo

Conforme o aumento de aplicações que necessitam de uma comunicação com outros computadores através da internet foi observado, seja pelo aumento do número de usuários ou pelo surgimento de novas tecnologias, como, por exemplo, o IOT, houve a demanda de criar um novo protocolo da camada de rede para comportar a carência de novos endereços, assim ocorreu a criação do protocolo IPv6 que está em processo de substituição do seu antecessor, o protocolo IPv4. Um dos grandes precursores desta mudança foi a tecnologia de comunicação móvel 4G, pois essa prevê um grande aumento da intensidade de tráfego de sinal, com novos usuários, com a possibilidade de um humano se conectar à internet ou uma máquina abastecer um banco de dados de informações obtidas de diferentes vias. Assim, o presente trabalho pretende explorar o protocolo IPv6 e os principais desdobramentos que a tecnologia 4G pode ocasionar no protocolo, seja por meio estatístico ou por instrumento de avaliação de dispositivos que a utilizam.

Sumário

1. Introdução

Testes de conformidade devem ser realizados antes do lançamento de um dispositivo, pois o produto deve estar dentro de uma série de padrões estabelecidos por agências reguladoras. Porém, quando se imaginam testes de conformidade em dispositivos móveis, a primeira coisa que é possível pensar são testes envolvendo parâmetros de rádio frequência, isto é, testes que vão analisar se o dispositivo está emitindo uma potência aceitável que não irá afetar o funcionamento de outros dispositivos. Caso isto não ocorra, pode-se ocasionar danos a outros dispositivos (KITCHEN, 2001, p. 327), ou até mesmo o nível de radiação emitida pode causar alguma lesão ao usuário. Esses testes são conhecidos como testes de SAR (*Specific Absorption Rate*), nos quais se avalia se a exposição excessiva pode causar hipertermia, às vezes chamada de exaustão pelo calor, uma condição aguda e tratável que, se negligenciada, poderia ter resultados sérios (KITCHEN, 2001, p. 58).

No entanto, testes de conformidade que verificam se o dispositivo é capaz de processar corretamente o cabeçalho IPv6, são de suma importância também, conforme é explorado nesta monografia.

1.1 Descrição do problema

No contexto do protocolo IPv6, os dispositivos que desejam ser lançados no mercado precisam ser submetidos a um teste de conformidade especificado pela ANATEL (Agência Nacional de Telecomunicações) que são baseados na RFC 2460, disponibilizado pela IETF *(Internet Engineering Task Force)*, em que são testados vários tipos de pacotes IPv6. Em vista disso é interessante notar que esse tipo de teste é importante para verificar se o dispositivo em teste é capaz de trocar pacotes com a rede mesmo se os pacotes enviado pela rede sejam construídos de maneira inadequada, visto que, neste tipo de projeto, é

necessário ratificar todos os casos possíveis, até mesmo aqueles em que a rede envia pacotes construídos de forma inapropriada.

Esses testes citados examinam diversos parâmetros presentes na construção do pacote, isto é, os componentes do seu cabeçalho, para verificar se o dispositivo é capaz de processá-los corretamente, sendo que em determinados casos é necessários uma mensagem respondendo ao pacote inadequado ou não enviar nenhuma resposta, sendo que neste caso é necessário um tempo de espera prolongado, pois alguns dispositivos são de baixo custo e não utilizam bons processadores e chipsets que são capazes de enviar a resposta rapidamente.

Portanto, é importante que os dispositivos passem por esse tipo de teste antes de irem para o mercado, pois, caso ocorra um erro durante a fase de teste, é possível efetuar correções antes de ocorrer uma falha em campo. Uma falha como essa pode ser catastrófica, pois, em determinadas situações, é possível que ocorra um descasamento de pacotes, de forma que o usuário pode até mesmo perder a conexão com a internet. Em uma falha como essa pode ser difícil de encontrar o responsável, em razão de existirem vários componentes da comunicação entre rede e o dispositivo que podem apresentar falhas. Um meticuloso trabalho de depuração de todo o sistema pode ser iniciado, de forma que a depuração pode ser até escalonada para as operadoras.

1.2 Motivação

Nesta monografia descreve-se a implementação de um software que envia os pacotes descritos pela norma supracitada, utilizando, no entanto, a porta RF dos dispositivos que possuem a tecnologia 4G implementada. É utilizado uma plataforma de testes wireless UXM da Keysight para funcionar como um simulador da estação rádio base assim como de agregador de todos os componentes presentes no cerne da estrutura do 4G.

Como se trata de um teste da camada IP, não há diferença se os testes forem irradiados ou conduzidos, como é na maioria dos testes que analisam parâmetros de radiofrequência, pois não há avaliação das medidas de potência apesar de estar utilizando a porta RF irradiada. No entanto, deve se atentar para a diferença de potência selecionada em ambos os casos, porque a atenuação de caminho é muito menor quando a amostra está conduzida, podendo até danificar algum componente mais sensível do dispositivo, como, por exemplo o amplificador de baixo ruído (LNA) que é o componente mais vulnerável da parte frontal dos componentes RF (BAEK; CHO; KO, 2018).

A principal contribuição deste trabalho foi implementar um programa para enviar e analisar pacotes IPv6. O programa foi desenvolvido na linguagem C# pois ela possui uma fácil integração com o instrumento e também por possuir a biblioteca Pcap.Net que é capaz de enviar e ler pacotes transferidos entre a rede, plataforma de testes wireless UXM e a rede com muita facilidade. Em suma, pretende-se realizar uma análise das mensagens trocadas entre a rede e o dispositivo.

1.3 Trabalhos relacionados

Santos (2019) implementa os mesmo testes vistos nesta monografia, no entanto utiliza um servidor Unix para simular o servidor que irá enviar e analisar os pacotes trocados entre a rede e o dispositivo sob teste. é interessante

notar que algumas empresas realizam esses testes no mercado a fim de realizar a homologação dos dispositivos.

Silva (2022) desenvolve um software capaz de estabelecer um servidor IMS (*IP Multimedia Subsystem*) que utiliza elementos de baixo custo, como um SDR (*Software Defined Radio*), além de conseguir realizar uma ligação utilizando a tecnologia 4G configurando uma ligação em VoLTE (*Voice over LTE*). Este não é viés do presente trabalho, pois foi utilizado um instrumento já validado e aperfeiçoado pela indústria.

Machado (2015) realiza uma análise do processo de transição do protocolo IP que já estava em processo de substituição no qual é apresentada a implementação dos métodos Pilha Dupla, tunelamento, *IPv6-over-IPv4* e tradução NAT64/DNS64, que são processos já existentes. No entanto, são utilizadas máquinas virtuais a fim de realizar a simulação desses procedimentos.

2. Fundamentação teórica

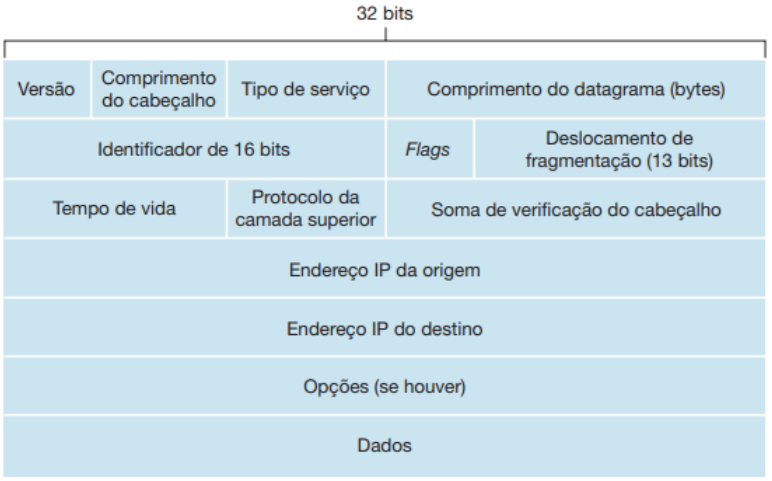
2.1 Internet Protocol version 4 (IPv4)

O protocolo IPv4 foi projetado em 1980 sendo descrito através da norma RFC 791 da IEFT em 1981, tendo sido implementado pela primeira vez em 1983. Este protocolo continha, aproximadamente, 4,3 bilhões de endereços, o que era uma ótima estimativa dado que o mundo nesta época continha um população mundial de, aproximadamente, 4.434.682 pessoas, pode-se considerar que todas as pessoas do mundo poderiam se conectar simultaneamente a internet, dado que os cientistas da época tinham em mente que apenas pessoas iriam se conectar a internet, desconsiderando tecnologias posteriores.

A estrutura do protocolo é expressão de quatro blocos de oito bits resultando em 32 bits, isto é, os blocos podem assumir valores de 0 até 255, sendo alguns dos primeiros blocos definem endereços que são reservados para conexões de dispositivos em rede local.

Abaixo pode-se ver uma imagem que ilustra o cabeçalho do protocolo com todas as suas particularidades.

Figura 1: Formato do datagrama IPv4



Fonte: Kurose, 2017, p. 246

Abaixo segue a definição de cada um dos blocos dentro do cabeçalho.

- **Versão**: Esses quatro bits especificam a versão do protocolo IP do datagrama (KUROSE, 2017, p. 268)
- **Comprimento do cabeçalho**: Como um datagrama IPv4 pode conter um número variável de opções (incluídas no cabeçalho do datagrama IPv4), esses quatro bits são necessários para determinar onde, no datagrama IP, os dados começam de fato. (KUROSE, 2017, p. 268)
- **Tipo de serviço:** Os bits de tipo de serviço *(type of service — TOS)* foram incluídos no cabeçalho do IPv4 para poder diferenciar os diferentes tipos de datagramas IP (por exemplo, que requerem, particularmente, baixo atraso, alta vazão ou confiabilidade). (KUROSE, 2017, p. 268)
- **Identificador, flags, deslocamento de fragmentação:** Esses três campos estão relacionados com a fragmentação do pacote IP.
- **Comprimento do datagrama:** É o comprimento total do datagrama IP (cabeçalho mais dados) medido em bytes. (KUROSE, 2017, p. 269)
- **Comprimento do datagrama:** É o comprimento total do datagrama IP (cabeçalho mais dados) medido em bytes. (KUROSE, 2017, p. 268)
- **Tempo de vida:** O campo de tempo de vida (*time-to-live — TTL*) é incluído para garantir que os datagramas não fiquem circulando para sempre na rede. (KUROSE, 2017, p. 269)
- **Protocolo**: O valor do campo indica o protocolo de camada de transporte específico ao qual a porção de dados desse datagrama IP deverá ser passada. Em suma, identifica o tipo de protocolo utilizado, por exemplo, UDP, assumindo o valor 17, e TCP, assumindo o valor 6. (KUROSE, 2017, p. 269)
- **Soma de verificação do cabeçalho:** A soma de verificação do cabeçalho auxilia um roteador na detecção de erros de bits em um datagrama IP recebido. (KUROSE, 2017, p. 269)

- **Endereços IP de origem e de destino:** endereço de origem e endereço do destino final do pacote IP. (KUROSE, 2017, p. 270)
- **Opções e Padding:** Usado em situações de teste e verificação de erros na rede, sendo opcional. (KUROSE, 2017, p. 270)
- **Dados (carga útil)**: Contém o segmento da camada de transporte (TCP ou UDP) a ser entregue ao destino. (KUROSE, 2017, p. 270)

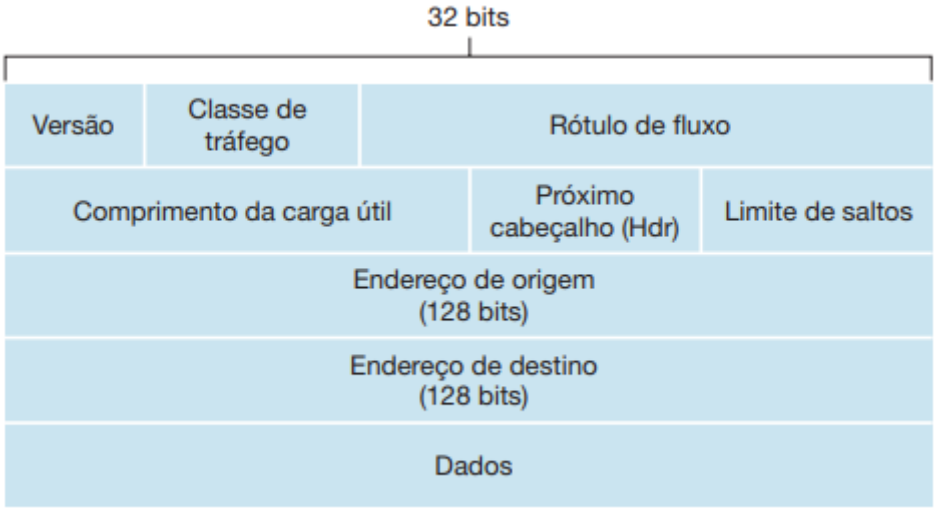
2.2 Internet Protocol version 6 (IPv6)

O protocolo IPv6 foi descrito na RFC 2460 da IEFT de 1998, mas foi oficializado somente em 2012. O IPv6 é capaz de fornecer até undecilhão (10^{36}) de endereços, um número exorbitantemente maior quando comparado ao IPv4, sendo ideal para comportar toda a demanda requisitada pelos novos usuários pelas novas tecnologias.

A estrutura do protocolo é expressão de quatro blocos de dezesseis bits, em que cada unidade dos blocos pode assumir valores de 0 até F, sendo que os zeros à esquerda podem ser eliminados.

Uma das vantagens do IPv6 sobre o IPv4 é que o cabeçalho do modelo mais novo pode ser customizado, resultando numa maior eficiência. Indica-se que o cabeçalho deve seguir o seguinte modelo.

Figura 2: Formato do datagrama IPv6



Fonte: Tanenbaum, 2021, p. 359

Abaixo segue o detalhamento de cada um dos blocos menores dentro do cabeçalho do IPv6.

- **Versão:** Identifica a versão de protocolo utilizado, sendo que, no caso do datagrama do IPv6, possui valor igual a 6. (KUROSE, 2017, p. 264)
- **Classe de tráfego**: Identifica os pacotes por classes de serviços ou prioridade, sendo que possui funcionalidade similar ao TOS do IPv4. (KUROSE, 2017, p. 264)
- **Rótulo de fluxo:** É utilizado para identificar um fluxo de datagrama. (KUROSE, 2017, p. 264)
- **Tamanho dos dados:** Aponta o tamanho, em Bytes, dos dados enviados junto ao cabeçalho.
- **Próximo cabeçalho:** Esse campo identifica o protocolo ao qual o conteúdo (campo de dados) desse datagrama será entregue (por exemplo, TCP ou UDP). (KUROSE, 2017, p. 287)
- **Limite de saltos**: Esse campo é decrementado a cada salto de roteamento e indica o número máximo de roteadores pelos quais o pacote pode passar antes de ser descartado. Se a contagem do limite de saltos chegar a zero, o datagrama será descartado. (KUROSE, 2017, p. 287)
- **Endereços de origem e de destino**: Os vários formatos do endereço de 128 bits do IPv6 são descritos no RFC 4291. (KUROSE, 2017, p. 287)
- **Dados**: Esta é a parte da carga útil do datagrama IPv6. (KUROSE, 2017, p. 287)

2.3 *Internet Control Message Protocol 6* (ICMPv6)

O ICMPv6 é um protocolo vital para o funcionamento do IPv6, pois através desse recurso é possível realizar diversas parametrizações da rede, como, por exemplo, relatar erros no processamento do pacote, realizar diagnósticos da rede e relatar características da rede, etc. É interessante notar que o protocolo IPv4 também possui um cabeçalho ICMPv4. No entanto para este caso, o cabeçalho é menos importante para o protocolo do que para a versão sucessora, pois assume várias funcionalidades que o próprio IPv4 realizava, tal como, o multicast e a descoberta de vizinhos. Dessa forma, muitas mensagens do próprio protocolo foram mantidas, porém com códigos diferentes, com destacado abaixo:

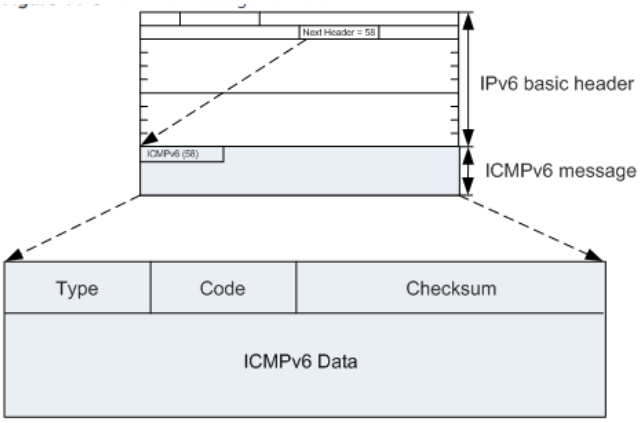
- *Destination Unreachable*: É usada quando a sub-rede ou um roteador não consegue localizar o destino (TANENBAUM, 2021, p. 346), ficou conhecida como mensagem do tipo 1.
- *Packet too big*: Ocorre quando o pacote possui uma quantidade de bytes acima do permitido pelo caminho, ficou conhecida como mensagem do tipo 2.
- *Time exceeded*: É enviada quando um pacote é descartado porque seu contador chegou a zero (TANENBAUM, 2021, p. 346). ficou conhecida como mensagem do tipo 3
- *Parameter problem*: Indica que um valor inválido foi detectado em um campo de cabeçalho (TANENBAUM, 2021, p. 346), ficou conhecida como mensagem do tipo 4.

Porém, como foi supracitado, novas funcionalidades foram adicionadas a esse protocolo, podendo-se ressaltar o descobrimento de redes, visto que, para realizar a comunicação através destes protocolos, os roteadores e dispositivos na rede necessitam conhecer os endereços da camada de enlace (endereços MAC),

e o IPv6 utiliza o protocolo de descoberta da vizinhança NPD (*Neighbor Discovery Protocol*) para reconhecimento dos endereços da camada de enlace dos vizinhos que estão resididos na mesma rede.

O ICMPv6 é reconhecido através do campo próximo cabeçalho, que será preenchido pelo valor 58, na figura abaixo pode-se ter mais detalhes sobre a localização do protocolo citado, assim como um detalhamento do pacote.

Figura 3: Estrutura do pacote ICMPv6



Fonte: HUAWEI. Captura de tela da página "ICMPv6". 2023. Disponível em: https://support.huawei.com/enterprise/en/doc/EDOC1000178170/d005b7c7/icmpv6. Acesso em: 14 de maio de 2022.

Como se pode ver, o ICMPv6 possui uma estrutura simples, dispondo de apenas quatro campos:

- Tipo: O campo tipo indica o tipo da mensagem. Seu valor determina o formato dos dados restantes. (M Gupta, 2006, p.3)
- Código: O campo de código depende do tipo de mensagem. Ele é usado para criar um nível adicional de grandeza da mensagem. (M Gupta, 2006, p.3)
- Soma de verificação: É usado para detectar corrupção de dados no ICMPv6 e partes do cabeçalho IPv6. (M Gupta, 2006, p.3)

- Dados ICMPv6: As mensagens ICMPv6 são agrupadas em duas classes: mensagens de erro e mensagens informativas, de modo que as com valores de 0 a 127 no campo tipo são mensagens de erro, as com valores de 128 a 255 no campo tipo são mensagens informativas. (M Gupta, 2006, p.3)

2.4 Frequência FR1

Em redes móveis de última geração, LTE e 5G, foi delimitada a utilização de bandas específicas para as aplicações com essas tecnologias, sendo conhecidas como FR1, caracterizada pelo intervalo de frequência de 410 MHz a 7125 MHz, que também é conhecida como sub 6-GHz, e FR2 que é utilizada exclusivamente pela tecnologia 5G, caracterizada pelo intervalo de frequência de 24.35GHz a 52.6GHz, que também é conhecida como ondas milimétricas. No primeiro caso, ocorre um compartilhamento de bandas entre a tecnologia 4G e 5G, sendo utilizadas técnicas para a distribuição das bandas para as tecnologias, como, por exemplo, o DSS (*dynamic sharing spectrum*).

Na tabela abaixo, listam-se todas as bandas e suas respectivas frequências, assim como o tipo de duplexação utilizada por cada banda.

Tabela 1: Frequência de operação FR1

Banda de operação da E-UTRA	Intervalo de frequência de *uplink* (UL)	Intervalo de frequência de *downlink* (DL)	Tipo de duplexação
	F_{UL_low} até F_{UL_high}	F_{DL_low} até F_{DL_high}	
1	1920 MHz a 1980 MHz	2110 MHz a 2170 MHz	FDD
2	1850 MHz a 1910 MHz	1930 MHz a 1990 MHz	FDD
3	1710 MHz a 1785 MHz	1805 MHz a 1880 MHz	FDD
4	1710 MHz a 1755 MHz	2110 MHz a 2155 MHz	FDD
5	824 MHz a 849 MHz	869 MHz a 885 MHz	FDD
6	830 MHz a 840 MHz	875 MHz a 885 MHz	FDD
7	2500 MHz a 2570 MHz	2620 MHz a 2690 MHz	FDD
8	880 MHz a 915 MHz	925 MHz a 960 MHz	FDD
9	1749.9 MHz a 1784.9 MHz	1844.9MHz a 1879.9MHz	FDD
10	1710 MHz a 1770 MHz	2110 MHz a 2170 MHz	FDD

11	1427.9 MHz a 1447.9 MHz	1475.9 MHz a 1495.9 MHz	FDD
12	699 MHz a 716 MHz	728 MHz a 746 MHz	FDD
13	777 MHz a 787 MHz	746 MHz a 756 MHz	FDD
14	788 MHz a 798 MHz	758 MHz a 768 MHz	FDD
15	Reserved	Reserved	FDD
16	Reserved	Reserved	FDD
17	704 MHz a 716 MHz	758 MHz a 768 MHz	FDD
18	815 MHz a 830 MHz	860 MHz a 875 MHz	FDD
19	930 MHz a 945 MHz	875 MHz a 890 MHz	FDD
20	832 MHz a 862 MHz	761 MHz a 821 MHz	FDD
21	1447.9 MHz a 1462.9 MHz	1495.9 MHz a 1510.9 MHz	FDD
22	3410 MHz a 3490 MHz	3510 MHz a 3590 MHz	FDD
23	2000 MHz a 2020 MHz	2180 MHz a 2200 MHz	FDD
24	1626.5 MHz a 1660.5 MHz	1525 MHz a 0226 MHz	FDD
25	1850 MHz a 1915 MHz	1930 MHz a 1995 MHz	FDD
33	1900 MHz a 1920 MHz	1900 MHz a 1920 MHz	TDD
34	2010 MHz a 2025 MHz	2010 MHz a 2025 MHz	TDD
35	1850 MHz a 1910 MHz	1950 MHz a 1910 MHz	TDD
36	1930 MHz a 1990 MHz	1930 MHz a 1990 MHz	TDD
37	1910 MHz a 1990 MHz	1910 MHz a 1930 MHz	TDD
38	2570 MHz a 2620 MHz	2570 MHz a 2620 MHz	TDD
39	1880 MHz a 1920 MHz	1880 MHz a 1920 MHz	TDD
40	2300 MHz a 2400 MHz	2300 MHz a 2400 MHz	TDD
41	2496 MHz a 2690 MHz	2496 MHz a 2690 MHz	TDD

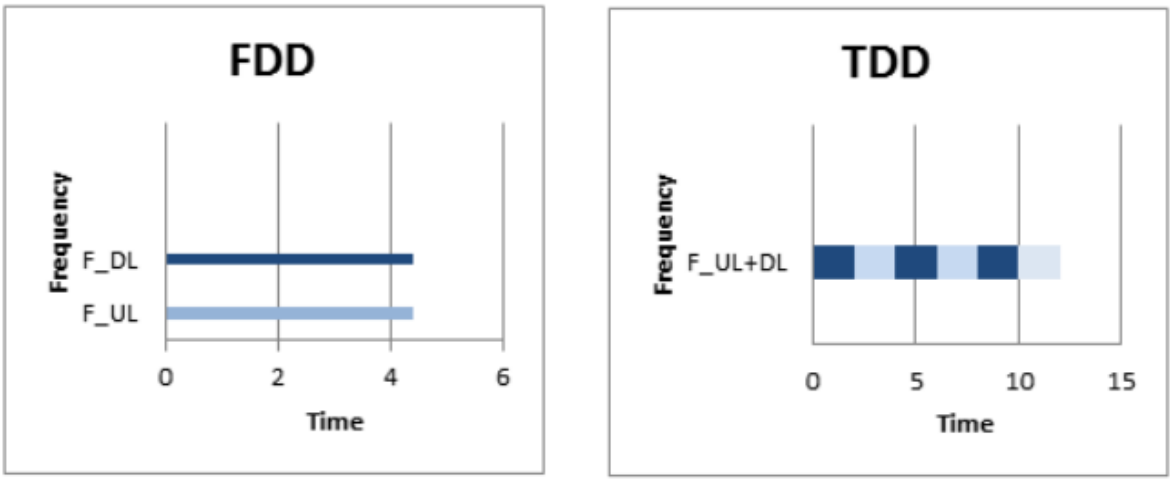
42	3400 MHz a 3600 MHz	3400 MHz a 3600 MHz	TDD
43	3600 MHz a 3800 MHz	3600 MHz a 3800 MHz	TDD

Fonte: Christina Gebner. Long Term Evolution. EUA, 2011 . p 46

No entanto, os dispositivos fabricados não necessariamente implementam todas as bandas supracitadas. Isso ocorre devido a série de fatores: o primeiro fato é relacionado ao projeto do dispositivo, isto é, que poupe o máximo de bateria; tendo em vista isso, os fabricantes de celulares escolhem um conjunto de bandas que melhor condizem com o projeto do seu dispositivo. O segundo ponto, e mais importante, é que a utilização dessas bandas é definida pelo país onde o dispositivo irá operar, isto é, cada país tem seu próprio grupo de bandas que são utilizadas para comunicação. Sendo assim, não faz sentido um dispositivo ser capaz de se comunicar em bandas das quais ele não irá utilizar para comunicação, além do mais, isso acarretaria em uma perda excessiva de bateria e de tempo, pois o dispositivo procuraria uma rede nessas bandas para realizar a comunicação.

Pode-se ressaltar que existem dois tipos de duplexação citadas na tabela acima, a primeira é a FDD (*frequency duplex division*), que utiliza diferentes frequências para realizar a comunicação entre *uplink* e *downlink*, isto é, a estação rádio base enviará dados através de uma frequência específica, visto que o dispositivo em comunicação utilizará outra frequência para enviar dados para a estação rádio base, todavia o TDD (*time duplex division*), utiliza a mesma frequência para realizar a comunicação, porém utiliza o tempo para intercalar a comunicação de *downlink* e *uplink*. Na imagem abaixo é ilustrado como ocorre a duplexação em ambos os casos.

Figura 4: Operação do TDD e FDD



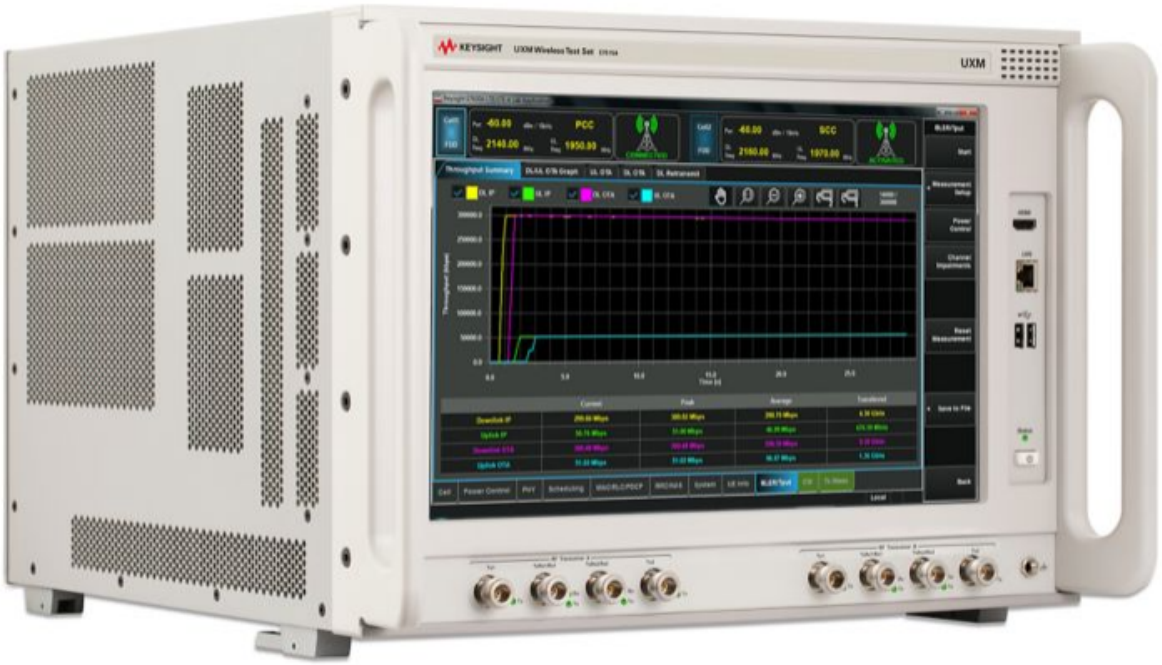
Fonte: Elshaer, Hisham. (2012)

3. Materiais e métodos

3.1 Plataforma de teste Wireless UXM 4G

O UXM 4G, também conhecido pelo *Part Number* E7515A, é uma plataforma de testes de celulares que é capaz de realizar diversas análises em dispositivos, como, por exemplo, celulares com as tecnologias mais recentes.

Figura 5: UXM 4G Plataforma de testes wireless



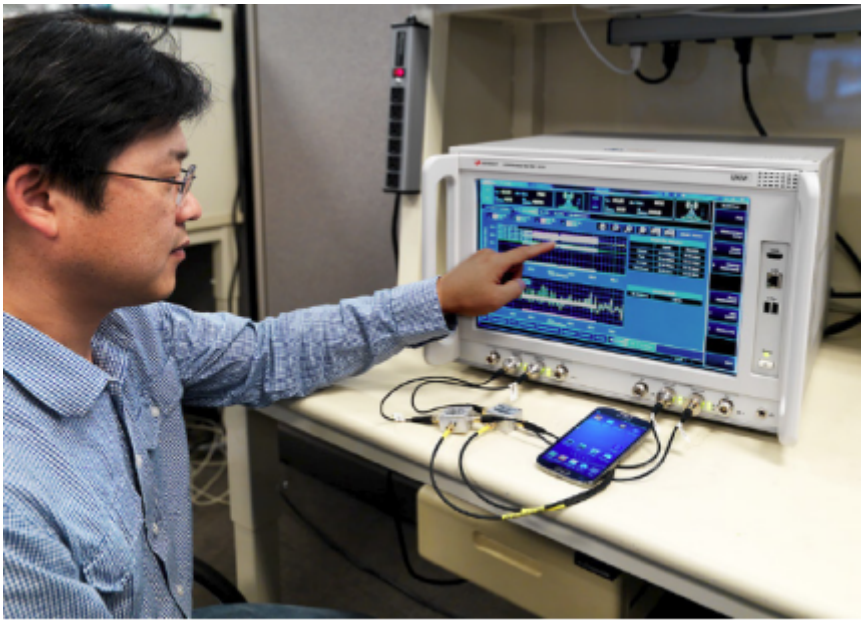
Fonte: Keysight technologies.visão geral técnica E7530A. EUA, 2019. p 3

Vale notar que apesar de essas tecnologias não estarem no mercado ou estarem presentes de forma unânime, existe a necessidade de os dispositivos que utilizam essa tecnologia passarem por diversos teste de conformidade, sendo que, para cada país, existem diferentes tipos de teste e variáveis a serem contabilizadas. Nesse ponto é que entra a plataforma de testes de celulares, tanto na bancada dos engenheiros que estão analisando os processos que foram configurados anteriormente quanto para o órgão regulador, isto é, o dispositivo deve atender às expectativas do fabricante e também estar dentro de todas as normas existentes.

Focando um pouco mais sobre a configuração do instrumento, o UXM é munido de oito portas, sendo que elas podem ser utilizadas com a configuração de saída de RF e entradas de RF. Isto é, há a possibilidade de configurar algumas portas para atuar como um analisador vetorial de sinais ou como gerador vetorial de sinais. Assim, ele é capaz de implementar uma série de técnicas utilizadas na comunicação sem fio, como, por exemplo, o MIMO (*Multiple Input Multiple Output*), em que há múltiplas antenas se comunicando simultaneamente ou completando uma a outra.

Na figura abaixo, pode-se ver um usuário operando o instrumento. É interessante ressaltar que, neste caso, é utilizando a tecnologia MIMO, pois como pode se ver há mais de uma saída e entrada do instrumento sendo utilizado. Pode-se ressaltar que esse dispositivo possui algumas aberturas na parte posterior onde são encaixados conectores e cabos que são ligados diretamente na porta do dispositivo. No entanto, essa não é a única configuração que pode ser utilizada, pois podem-se empregar antenas no instrumento, e assim fazer com que a comunicação seja totalmente sem fio.

Figura 6: Exemplo de uso do instrumento



Fonte: Keysight technologies.visão geral técnica E7530A. EUA, 2019. p 2

O equipamento é operado através do sistema operacional, Windows. Isso dá ao instrumento e ao usuário uma possibilidade muito maior de funcionalidades, pois é possível utilizar aplicativos de diferentes naturezas, como, por exemplo, um VSA (*Vector Signal Analyzer*), Signal Studio, etc. Porém, o próprio instrumento possui sua aplicação onde podem ser realizadas as principais configurações de maneira mais facilitada do que utilizar os softwares citados para cada porta. Vale a pena evidenciar que o instrumento e as aplicações que rodam nele podem ser controladas por um computador externo, através de comandos SCPI (*Standard Commands for Programmable Instruments*); dessa forma o instrumento funcionará como um interlocutor das mensagens trocadas pelos dispositivos que estão envolvidos na comunicação, porém não apenas como uma estação rádio base, mas sim como um agregador de todos os componentes necessários para simular uma rede 4G. Isto é, com todas as suas funções, como, por exemplo a *Mobility Management Entity* (MME), que faz o gerenciamento do acesso e mobilidade do dispositivo, e também com todas as interfaces envolvidas entre as funções.

O instrumento envia e recebe pacotes IPv6 para um dispositivo que está passando por um teste de conformidade utilizando a conexão direta, seja ela com fio ou sem fio, entre os dois dispositivos. Pode-se ressaltar que o próprio aplicativo principal do instrumento permite o envio de pacotes IPv6 através do canal existente, sendo que o operador do instrumento, após uma pequena configuração, não tem a necessidade de se preocupar com o mais baixo nível da comunicação, focando principalmente na transferência de pacotes. Ainda sim, no entanto, é possível analisar os parâmetros de baixo nível (potência, frequência…).

Portanto, neste trabalho pretende-se inspecionar como é o comportamento de vários dispositivos que utilizam o protocolo IPv6, seguindo os testes previstos pela norma RFC 2460, pois o instrumento é capaz de identificar cada mensagem que o dispositivo sob teste envia para a rede, tanto através da comunicação através de ondas eletromagnéticas, por meio de antenas, quanto por meio de conexão direta.

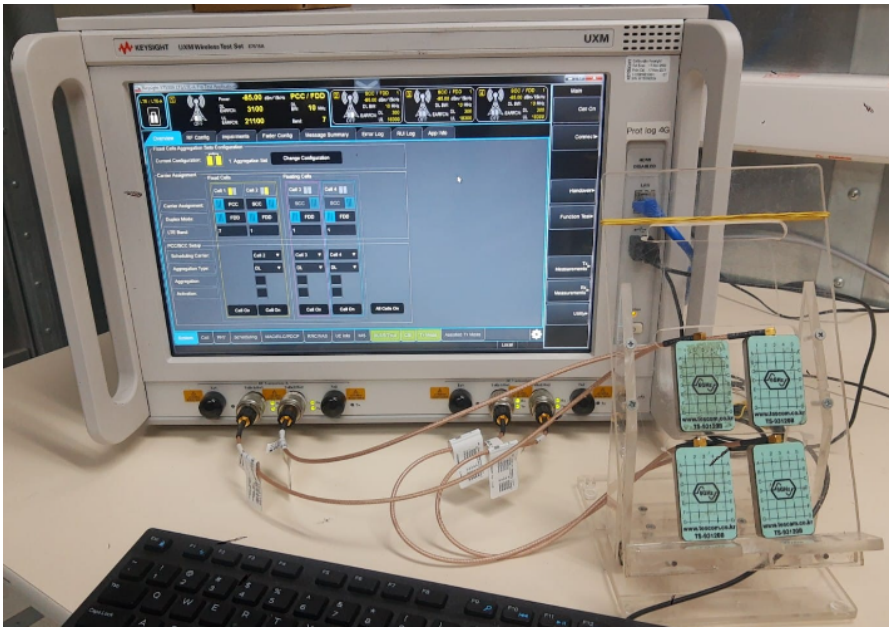
3.2 Realizando a conexão entre o dispositivo e o instrumento

3.2.1 Teste irradiado ou conduzido

O primeiro passo a ser tomado é definir como se pode conectar o instrumento ao dispositivo sob teste existem duas possibilidades: a primeira é conectar os cabos saem das portas da plataforma de teste diretamente no conector de RF do dispositivo, neste caso ocorre uma delimitação dos erros ocasionado pelos efeitos de transmissão, pois o dispositivo sob teste está diretamente conectado a porta do instrumento. No entanto, para partir desse método deve-se saber previamente quais bandas são usadas, pois cada porta RF do celular possui um conjunto de bandas possíveis, ou seja, é necessário possuir a mapa de porta do dispositivo. Porém, muitas vezes não é possível obter facilmente este documento, visto que os fabricantes de celulares o mantém em

sigilo uma vez que o mesmo pode revelar informações sigilosas sobre o funcionamento do aparelho. Em contrapartida, o método mais comumente empregado é utilizar antenas nas portas do instrumento, conforme é ilustrado pela imagem abaixo.

Figura 7: Imagem do setup configurado para testes irradiados



Fonte: Autoria do Autor

Desse modo, a conexão entre o dispositivo e o instrumento será realizada de maneira irradiada, isto é, o dispositivo será posicionado sobre as das antenas e, devido à congruência de configurações do instrumento com o dispositivo, a conexão será realizada e, assim, o teste poderá ser realizado. Tais configurações serão aprofundadas no item a seguir.

3.2.2 Configuração do instrumento

Os primeiros parâmetros a serem configurados são o tipo de duplexação e a banda, apesar de os dois parâmetros estarem diretamente ligados, conforme explicado no item 1.4.

Figura 8: Tela principal do instrumento

Fonte: Autoria do autor

Conforme dito no item 1.4, não há como saber previamente as bandas que o dispositivo suporta. Porém, alguns dados podem ser reunidos com o intuito de realizar a conexão, como, por exemplo, o local onde o dispositivo sob teste irá operar e as bandas que são utilizadas de maneira mais frequente.

O segundo fator a ser configurado são os parâmetros de segurança da comunicação. Para ocorrer o processo de comunicação entre o dispositivo sob teste e o instrumento é necessário que o instrumento saiba qual é o tipo de *Simcard* que está sendo utilizado no smartphone. Deve-se levar em conta que para a realização dos testes foi utilizado um *Simcard* da 3GPP que utiliza o algoritmo de autenticação Milenage (Rijndael) e o operador OPc, dessa forma precisa-se navegar até a aba de configuração e ajustar os parâmetros citados, conforme imagem abaixo.

Figura 9: Tela de configuração de parâmetros de segurança

Fonte: Autoria do autor

Por último, é necessário configurar os parâmetros de identificação do *Simcard*. Esse parâmetro se resume em dois parâmetros conhecidos como MCC (*Mobile Country Code*), que está relacionado ao país onde o dispositivo vai operar, e o parâmetro MNC (*Mobile Network Code*), que está relacionado à operadora que o celular irá utilizar.

No entanto, nesse caso não é necessário especificar um país ou uma operadora em específico, pois não é realizada nenhuma análise de métricas específicas relacionadas a algum país em específico ou a uma operadora.
Dessa forma pode-se utilizar o MCC como “001” e o MNC como “01”, que são parâmetros padrões, e que atendem à demanda para os testes realizados, pois, no caso, o dispositivo deve operar de forma independente do país ou operadora, já que os testes realizados são de padrões internacionais.

Figura 10: Tela de configuração de parâmetros IMSI

Fonte: Autoria do autor

Em seguida, deve-se ligar a célula e esperar que o dispositivo realize a conexão, isto é, passe da condição “*On*” para “*Connected*”.

Figura 11: Instrumento aguardando pelo dispositivo realizar a conexão

Fonte: Autoria do autor

Dependendo do fabricante do dispositivo em teste é possível realizar alguns passos para que a conexão se realize mais facilmente, como, por exemplo, “fechar em banda”. Esse procedimento se baseia em utilizar código que é utilizado no discador do smartphone, e, dessa forma, entrar em um menu em que é possível habilitar ou desabilitar bandas de todas as tecnologias suportadas pelo dispositivo. Posto isso, pode-se selecionar a tecnologia LTE e escolher a banda configurada no instrumento. Assim, o dispositivo procura redes apenas na banda que foi selecionada, o que, no final, auxilia no processo de conexão do dispositivo com o instrumento.

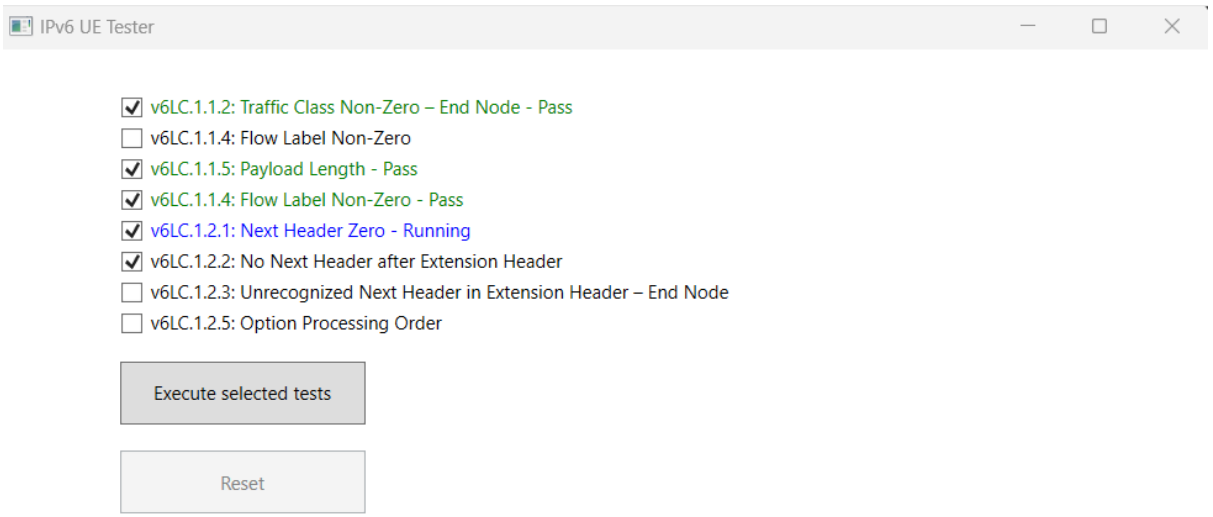
Figura 12: Instrumento e dispositivo conectados

Fonte: Autoria do autor

3.3 Implementação do software

O software proposto foi implementado em C# e utilizando a interface gráfica WPF, através da qual é possível escolher os testes que serão executados através da seleção de *checkbox*, conforme imagem abaixo.

Figura 13: Software implementado



Fonte: Autoria do autor

A escolha dessa linguagem de programação deve-se principalmente à facilidade de integração do instrumento com o programa e pela biblioteca conhecida como Pcap.Net, através da qual é possível manipular, criar, ler e enviar os pacotes diretamente para a porta RF do dispositivo sob teste.

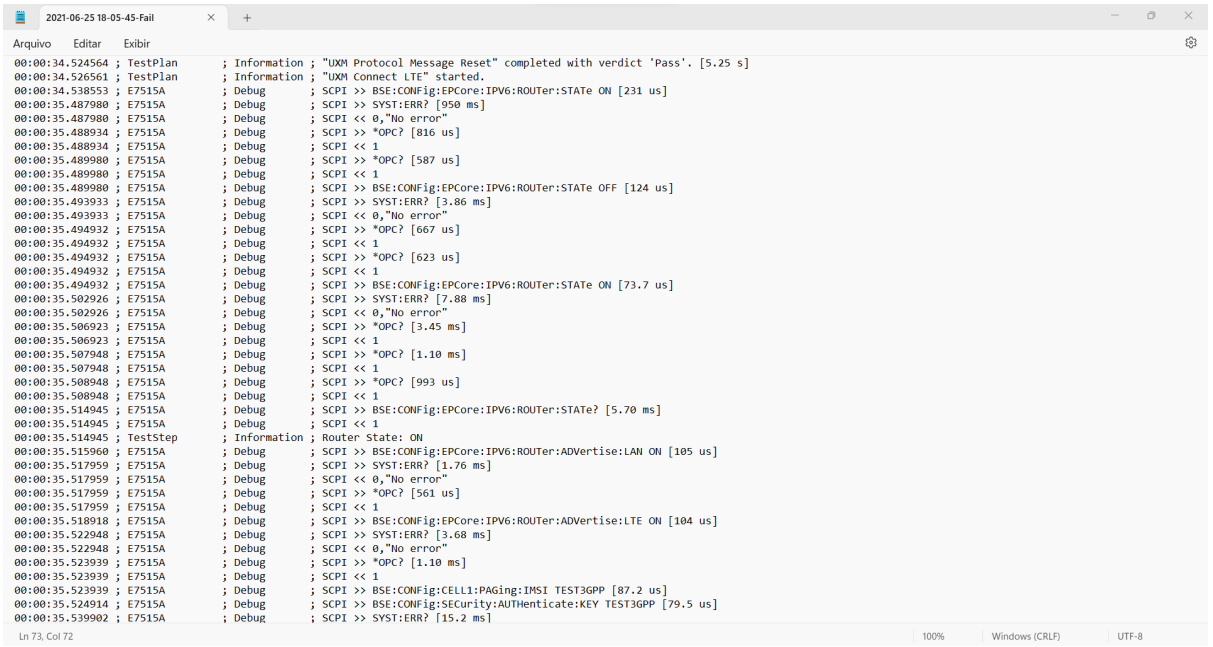
Deve-se levar em consideração que, quando utilizamos o wireshark, há uma limitação em relação às capturas, pois o wireshark na sua configuração padrão não consegue capturar os pacotes que saem pela porta RF do instrumento, pois é apenas possível vigiar a porta ethernet do dispositivo ou computador em que o software está instalado.

Dessa forma, se torna mais evidente a necessidade de se utilizar o Pcap.Net, dado que ele é capaz de capturar toda a interação da camada TCP/IP

que ocorre entre o dispositivo em teste e o instrumento. Essa funcionalidade se deve principalmente ao fato de que o Pcap.Net não foi implementado utilizando o Wireshark, mas sim utilizando o WinPcap. Deve-se levar em consideração que WinPcap é uma das melhores bibliotecas para captura de pacotes e envio de pacotes em camada TCP/IP, sendo utilizada em diversas aplicações que necessitam de manipulação dos pacotes IPv4 ou IPv6.

Também é gerado um arquivo teste que é utilizado para fazer a depuração de código em que é informado se o instrumento está em determinada condição ou se a mensagem foi devidamente criada e enviada para o dispositivo sob teste. Na imagem abaixo, pode-se ver uma imagem do arquivo gerado pelo código.

Figura 14: Software implementado



Fonte: Autoria do autor

3.4 Testes realizados

Os testes desenvolvidos foram baseados com base na lista de casos necessários e tendo em conta as especificações ETSI e TAHI, isto é, o UXM encaminha pacotes com a devida configuração indicada pelas especificações e o DUT (*Device Under Test*) deveria mandar uma resposta dentro critério relatados pelos documentos.

Abaixo está uma explicação, em formato de tópicos, de todos os testes realizados.

2.4.1 Teste v6LC.1.1.2: *Traffic Class Non-Zero – End Node*

O propósito deste teste é verificar se o DUT consegue processar corretamente o campo classe de tráfego dos pacotes recebidos e gera um valor válido em pacotes transmitidos. O pacote que deve ser enviado tem o formato abaixo:

Figura 15: Descrição do pacote do teste 1.1.2

Pacote A

IPv6 Header
Traffic Class: 32
Next Header: 58
ICMPv6 Echo Request

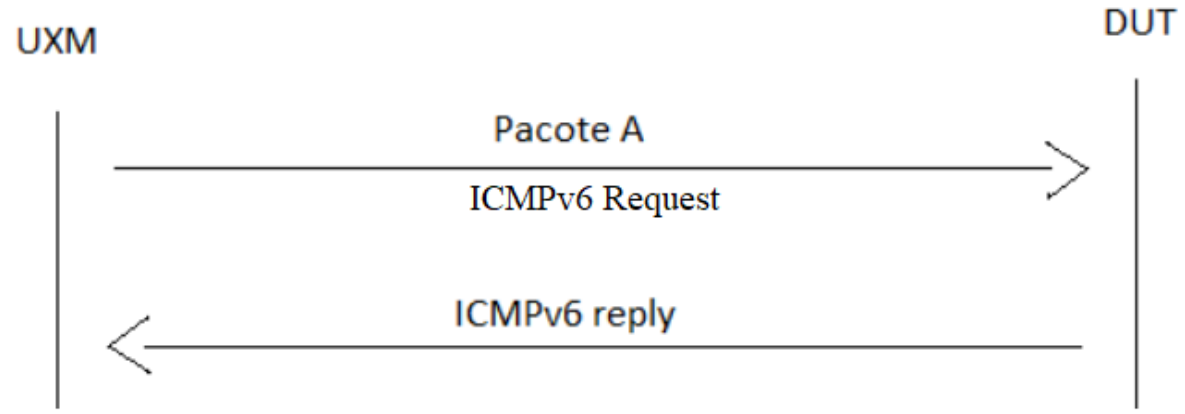
Fonte: Autoria do autor

O procedimento do teste deve ser o seguinte:

1. O UXM deve transmitir o pacote A para o DUT, uma solicitação de eco com um campo de classe de tráfego de 32, que é diferente de zero.
2. O DUT deve gerar uma resposta de eco. Se o DUT suportar um uso específico do campo tráfego de classe, com o valor classe de tráfego diferente de zero. Caso contrário, o campo tráfego de classe deve ser zero.

Na imagem abaixo vemos um diagrama que ilustra o teste realizado.

Figura 16: Ilustração dos pacotes enviados do teste 1.1.2



Fonte: Autoria do autor

Em suma, o UXM envia um pacote com um valor específico no campo tráfego de classe e espera o dispositivo responder a solicitação de eco que é enviada junto com o pacote.

2.4.2 Teste v6LC.1.1.4: *Flow Label Non-Zero*

O propósito deste teste é verificar se o DUT consegue processar corretamente o campo identificador de fluxo dos pacotes recebidos e gera um valor válido em pacotes transmitidos. O pacote que deve ser enviado tem o formato abaixo:

Figura 17: Descrição do pacote do teste 1.1.4

Pacote A

IPv6 Header
Flow Label: 214375
Next Header: 58
ICMPv6 Echo Request

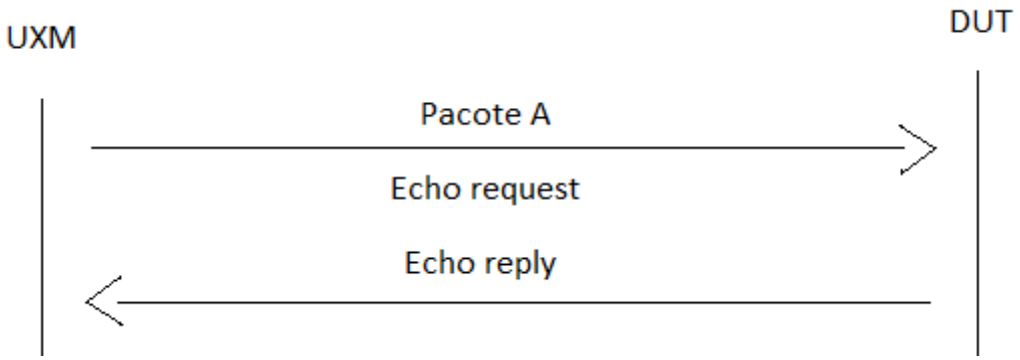
Fonte: Autoria do autor

O procedimento do teste deve ser o seguinte:

1. O UXM transmite o pacote A, uma solicitação de eco com o campo identificador de fluxo de 0x34567 para o DUT.
2. O DUT deve gerar uma resposta de eco. Se o DUT suportar o uso do campo identificador de fluxo, o identificador de fluxo na resposta de eco pode ser diferente de zero. Caso contrário, o campo identificador de fluxo deve ser zero.

Na imagem abaixo vemos um diagrama que ilustra o teste realizado.

Figura 18: Ilustração dos pacotes enviados do teste



Fonte: Autoria do autor

Pode-se dizer que esse teste é bastante parecido com o anterior, pois através de um valor em um campo específico, nesse caso o campo identificador de fluxo, espera-se uma resposta a solicitação de eco que é enviada com o pacote A descrito acima.

2.4.3 Teste v6LC.1.1.5: *Payload Length*

O propósito deste teste é averiguar se o DUT consegue verificar se o DUT processa corretamente o campo tamanho dos dados dos pacotes recebidos. O pacote que deve ser enviado tem o formato abaixo:

Figura 19: Descrição do pacote do teste 1.1.5

Pacote A

IPv6 Header Payload Length: Depende do tipo de teste (mais detalhes abaixo) Next Header: 58
ICMPv6 Echo Request

Fonte: Autoria do autor

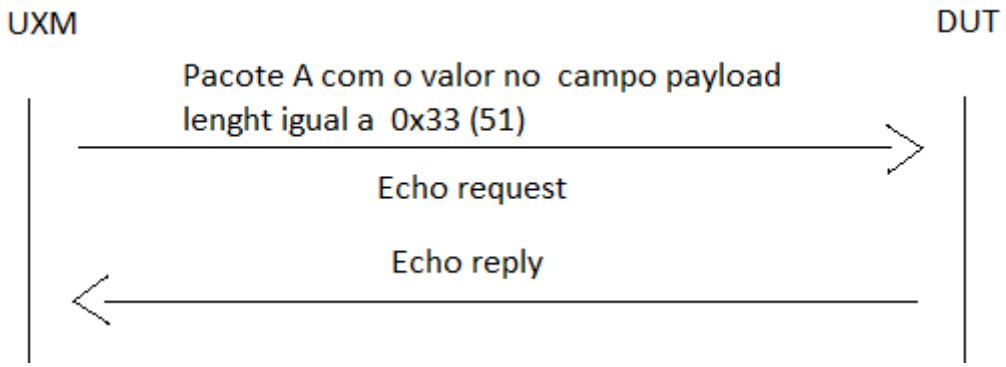
O procedimento do teste deve ser o seguinte:

Parte A:

1. O UXM transmite o pacote A para o DUT, uma solicitação de eco que tem um cabeçalho IPv6 com o campo tamanho dos dados de 0x33 (51).
2. O DUT deve gerar uma resposta de eco, indicando o processamento bem-sucedido do pacote.

   Na imagem abaixo vemos um diagrama que ilustra o teste realizado

Figura 20: Ilustração dos pacotes enviados do teste



Fonte: Autoria do autor

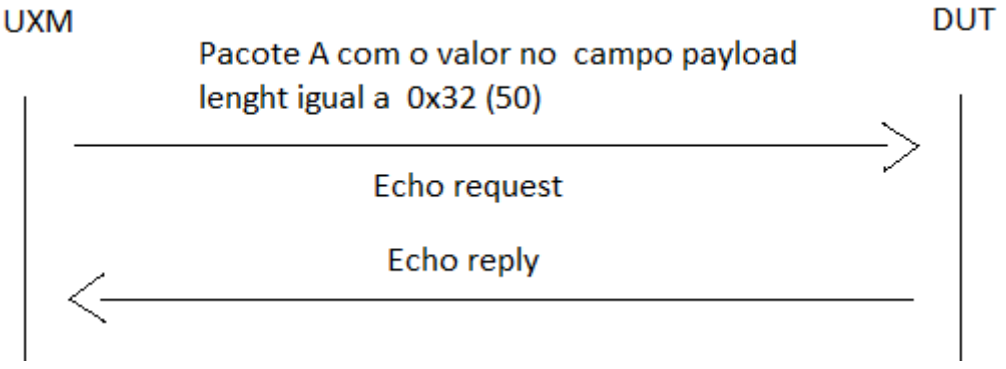
Parte B: Teste apenas para roteadores, não foi implementado.

Parte C: Tamanho dos dados par

1. O UXM transmite o pacote A para o DUT, uma solicitação de eco que tem um cabeçalho IPv6 com o campo tamanho dos dados de 0x32 (50).
2. O DUT deve gerar uma resposta de eco, indicando o processamento bem-sucedido do pacote

Na imagem abaixo vemos um diagrama que ilustra o teste realizado.

Figura 21: Ilustração dos pacotes enviados do teste



Fonte: Autoria do autor

Neste teste temos a variação do parâmetro tamanho dos dados e em ambos os casos o dispositivo sob teste deve enviar uma resposta à solicitação de

eco. É interessante notar que em alguns casos a amostra em teste pode não responder à solicitação de eco, indicando que o pacote não foi corretamente processado.

Resumidamente, o software irá enviar o pacote A e espera a resposta de eco da amostra durante um intervalo de tempo, sendo que se esse retorno não ocorrer durante esse intervalo de tempo o teste é avaliado com o veredito de falha.

2.4.4 Teste v6LC.1.1.6: *No Next Header after IPv6 Header*

O propósito deste teste é verificar se o comportamento do DUT é adequado quando ele recebe um valor de próximo cabeçalho de 59 (sem próximo cabeçalho). O pacote que deve ser enviado tem o formato abaixo:

Figura 22: Descrição do pacote do teste 1.1.6

Pacote A

IPv6 Header
Next Header: 59
ICMPv6 Echo Request

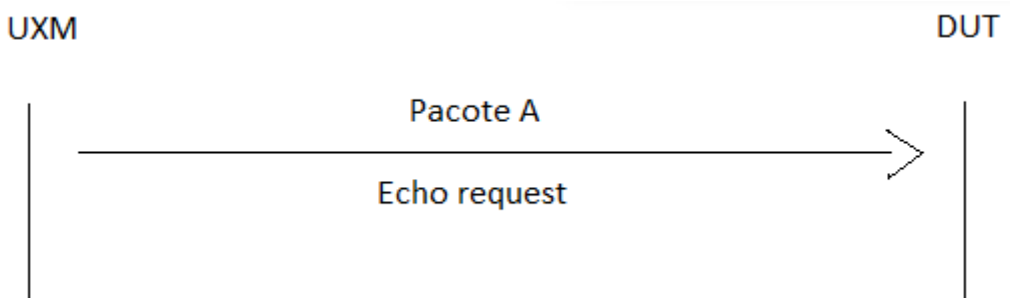
Fonte: Autoria do autor

Parte A: DUT um pacote com o campo No Next header

1. O DUT transmite o pacote A para o DUT, que contém um cabeçalho IPv6 com o campo próximo cabeçalho de 59. Após o cabeçalho IPv6, há um cabeçalho de solicitação de eco ICMPv6.
2. O DUT não deve enviar nenhuma resposta.

Na imagem abaixo vemos um diagrama que ilustra o teste realizado.

Figura 23: Ilustração dos pacotes enviados do teste



Fonte: Autoria do autor

Esse é o primeiro teste em que o dispositivo não deve responder a solicitação de eco do instrumento, pois a estrutura da solicitação é construída de

forma inadequada, de forma que testa o comportamento da amostra em teste caso ele receba um pacote inapropriado. Em geral, testes nos quais o smartphone não deve responder são mais demorados que os testes que deve ocorrer a resposta do dispositivo, pois deve-se esperar uma maior quantidade de tempo em razão da resposta da solicitação de eco do dispositivo, sendo que em alguns casos como esse deve-se esperar mais de dez minutos.

2.4.5 Teste v6LC.1.2.1: *Next Header Zero*

O propósito deste teste é verificar se o DUT consegue verificar se ele é capaz de descartar um pacote que tem o campo próximo cabeçalho igual a zero em um cabeçalho diferente de um cabeçalho IPv6 e gera uma mensagem de problema de parâmetro ICMPv6 para a origem do pacote. O pacote que deve ser enviado tem o formato abaixo:

Figura 24: Descrição do pacote do teste 1.2.1

Pacote A

IPv6 Header Next Header: 0
Hop-by-Hop Options Header Next Header: 0 Header Ext. Length: 0 Option: PadN Opt Data Len: 4
Hop-by-Hop Options Header Next Header: 58 Header Ext. Length: 0 Option: PadN Opt Data Len: 4
ICMPv6 Echo Request

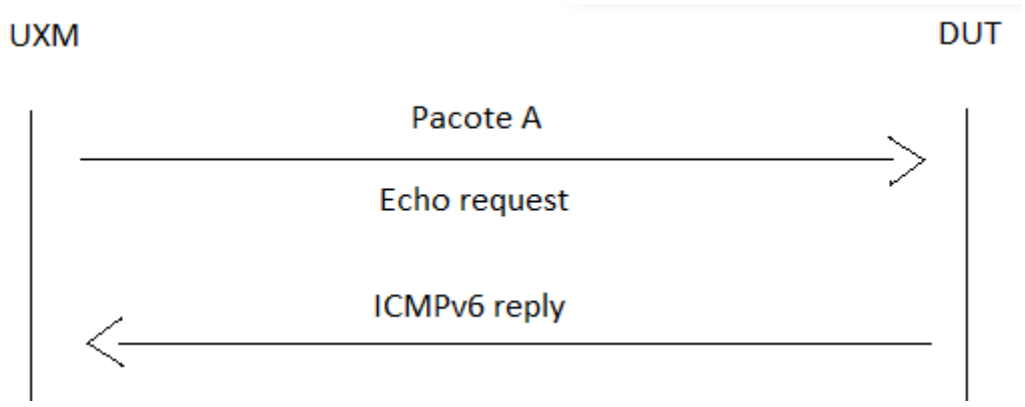
Fonte: Autoria do autor

O procedimento do teste deve ser o seguinte:

1. O UXM transmite o pacote A para o DUT, que possui um cabeçalho opções *Hop-by-Hop* com o campo próximo cabeçalho igual a zero.
2. O DUT deve enviar uma mensagem ICMPv6 com problema de parâmetro para o UXM. O campo código ICMPv6 deve ser 1 (encontrado algum tipo de próximo cabeçalho não reconhecido). O campo ponteiro ICMPv6 deve ser 0x28 (deslocamento do campo do próximo cabeçalho do cabeçalho opções *Hop-by-Hop*). O DUT deve descartar a solicitação de eco e não enviar uma resposta de eco ao UXM.

   Na imagem abaixo vemos um diagrama que ilustra o teste realizado.

Figura 25: Ilustração dos pacotes enviados do teste



Fonte: Autoria do autor

Neste caso também tem-se um caso diferente aos supracitados, pois no momento que o dispositivo recebe o pacote, ele deve retornar um pacote ICMPv6 detalhando o erro que ele observou. Assim, o pacote de controle emite um parâmetro de código indicando que ele não reconheceu os dados que compõem o Pacote A. Além disso, ele deve ignorar a solicitação de eco do instrumento.

2.4.6 Teste v6LC.1.2.2: *No Next Header after Extension Header*

O propósito deste teste é verificar se o comportamento do DUT é adequado quando ele encontra um valor de próximo cabeçalho de 59 (sem próximo cabeçalho). O pacote que deve ser enviado tem o formato abaixo:

Figura 26: Descrição do pacote do teste 1.2.2

Pacote A

IPv6 Header
Traffic Class: 32
Next Header: 58
ICMPv6 Echo Request

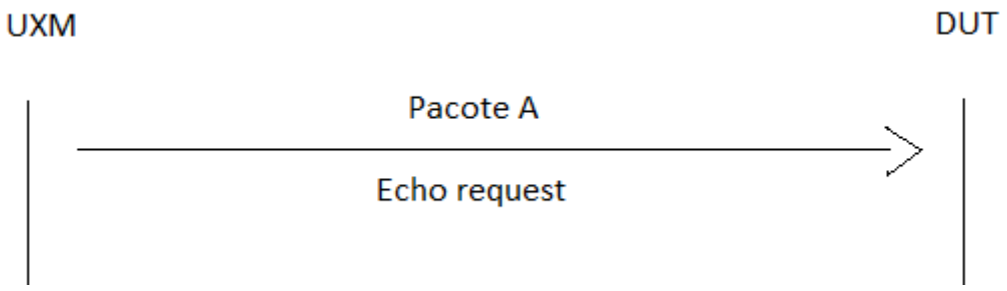
Fonte: Autoria do autor

O procedimento do teste deve ser o seguinte:

1. O UXM transmite o pacote A ao DUT, que contém um cabeçalho de opções de destino com um cabeçalho de 59. A seguir ao cabeçalho opções de destino está um cabeçalho ICMPv6 com solicitação de eco.
2. O DUT não deve enviar nenhum pacote em resposta ao pacote A.

Na imagem abaixo vemos um diagrama que ilustra o teste realizado.

Figura 27: Ilustração dos pacotes enviados do teste



Fonte: Autoria do autor

Em suma, neste caso o dispositivo sob teste não deve enviar nenhuma resposta a solicitação de eco enviada junto com o pacote A devido ao campo próximo cabeçalho estar inadequado.

2.4.7 Teste v6LC.1.2.3: *Unrecognized Next Header in Extension Header – End Node*

O propósito deste teste é verificar se um DUT descarta um pacote com um próximo cabeçalho não reconhecido ou inesperado e transmite uma mensagem de problema do parâmetro ICMPv6 para a fonte do pacote. Os pacotes que devem ser enviados têm o formato abaixo:

Figura 28: Descrição do pacote do teste 1.2.3

Pacote A

IPv6 Header Next Header: 60
Destination Options Header Next Header: Depende do tipo de teste, mais detalhes abaixo Header Ext. Length: 0 Option: PadN Opt Data Len: 4

Fonte: Autoria do autor

O procedimento do teste deve ser o seguinte:

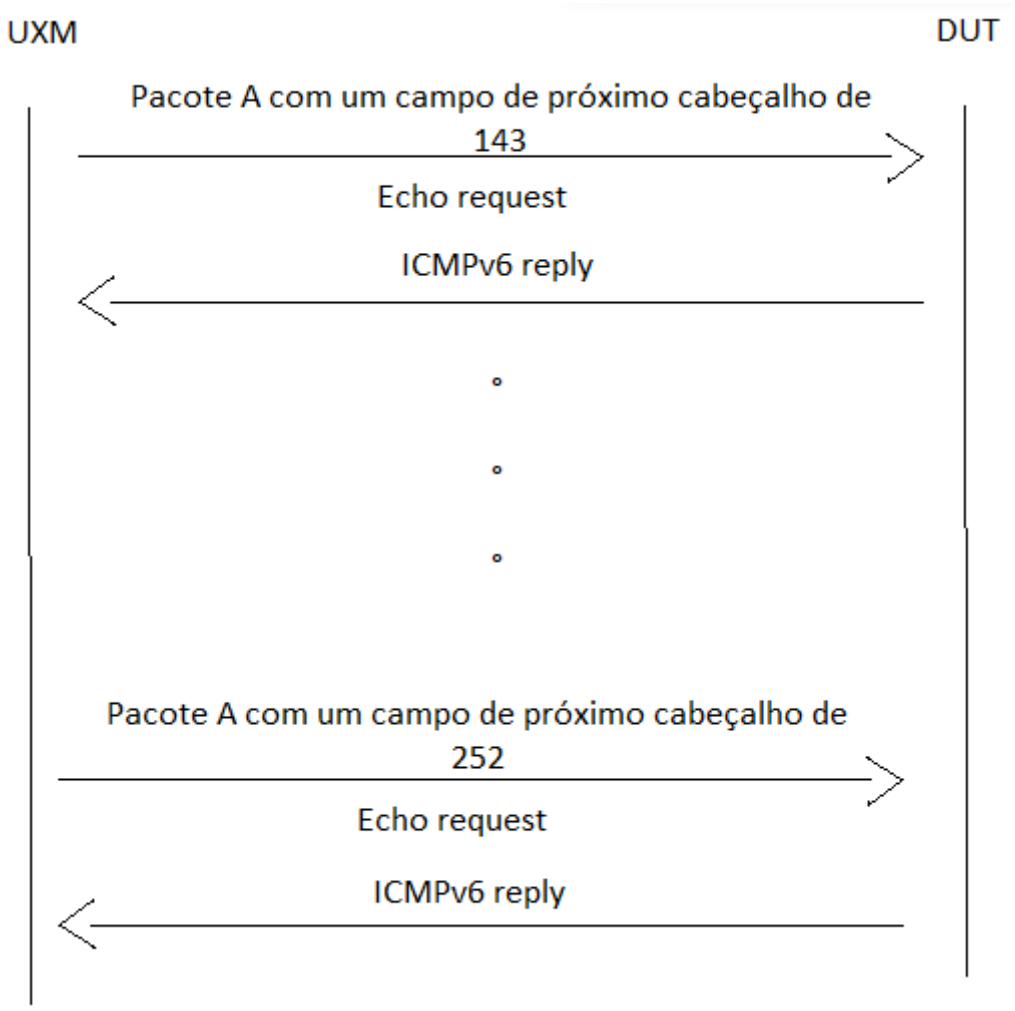
Próximo cabeçalho não reconhecido em cabeçalho de extensão (valores múltiplos)

1. O UXM transmite o pacote A, que tem um cabeçalho de opções de destino com um campo de próximo cabeçalho de 143.
2. O UXM transmite um Pedido de Eco válido para o DUT.
3. Deve ser repetido os passos 1 e 2 com todos os valores não reconhecidos do próximo cabeçalho entre 144 e 252 no passo 1.

4. O DUT deve enviar uma mensagem de problema de parâmetros ICMPv6 para o UXM. O campo de código ICMPv6 deve ser 1 (próximo cabeçalho encontrado não reconhecido). O campo do ponteiro do ICMPv6 deve ser 0x28 (deslocamento do campo próximo cabeçalho). O DUT deve enviar uma resposta de eco em resposta ao pedido de eco enviado por UXM no Passo 2.

Na imagem abaixo vemos um diagrama que ilustra o teste realizado.

Figura 29: Ilustração dos pacotes enviados do teste



Fonte: Autoria do autor

Pode-se ver que nesse teste diversos pacotes são enviados para o DUT, sendo que ele deve sempre responder com um pacote ICMPv6 indicando que foi encontrado um erro na estrutura do pacote.

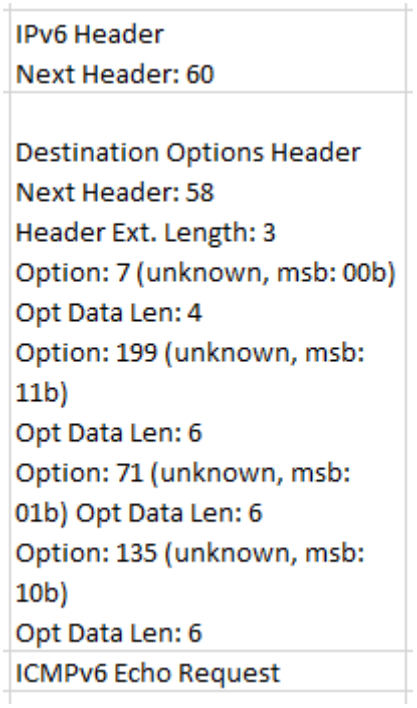
2.4.8 Teste v6LC.1.2.5: *Option Processing Order*

O propósito deste teste é verificar se um DUT processa corretamente opções num único cabeçalho, pela ordem de ocorrência. Os pacotes que devem ser enviados têm o formato abaixo:

Figura 30: Descrição dos pacotes do teste 1.2.5

Pacote A	Pacote B
IPv6 Header Next Header: 60	IPv6 Header Next Header: 60
Destination Options Header Next Header: 58 Header Ext. Length: 3 Option: 7 (unknown, msb: 00b) Opt Data Len: 4 Option: 71 (unknown, msb: 01b) Opt Data Len: 6 Option: 135 (unknown, msb: 10b) Opt Data Len: 6 Option: 199 (unknown, msb: 11b) Opt Data Len: 6	Destination Options Header Next Header: 58 Header Ext. Length: 3 Option: 7 (unknown, msb: 00b) Opt Data Len:: 4 Option: 135 (unknown, msb: 10b) Opt Data Len: 6 Option: 199 (unknown, msb: 11b) Opt Data Len: 6 Option: 71 (unknown, msb: 01b) Opt Data Len: 6
ICMPv6 Echo Request	ICMPv6 Echo Request

Pacote C



Fonte: Autoria do autor

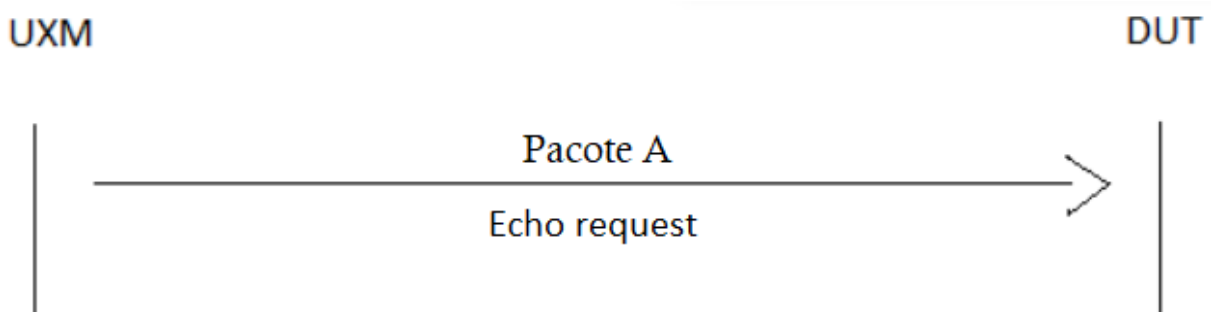
O procedimento do teste deve ser o seguinte:

Parte A: Primeira Opção tem Bits Mais Significativos 00b, seguinte tem Bits Mais Significativos 01b

1. O UXM transmite o Pacote A para o DUT, um pedido de eco que tem um cabeçalho de Opções de Destino com quatro opções desconhecidas. Os tipos de opções são 7, 71, 135, e 199.
2. O DUT deve descartar silenciosamente o pedido de eco ICMPv6 e não enviar quaisquer pacotes para o UXM.

Na imagem abaixo vemos um diagrama que ilustra o teste realizado.

Figura 31: Ilustração dos pacotes enviados do teste



Fonte: Autoria do autor

Pode-se ver neste caso que o pacote enviado para o dispositivo possui o campo cabeçalho de opções de destino que não seguem a norma, isto é, possuem valores que não são esperados pelo dispositivo, neste caso o dispositivo sob teste não deve enviar nenhum pacote em resposta ao pacote enviado.

Parte B: A Primeira Opção tem Bits Mais Significativos 00b, A Seguinte tem Bits Mais Significativos 10b

1. O UXM transmite o pacote B para o DUT, um pedido de eco que tem um cabeçalho de Opções de Destino com quatro opções desconhecidas. Os tipos de opções são 7, 135, 199, e 71.
2. O DUT deve enviar uma mensagem ICMPv6 de problemas de parâmetros para o UXM. O campo de código deve ser 2 (Opção IPv6 não reconhecida encontrada). O campo ponteiro tem de ser 0x30 (deslocamento do campo tipo de opção da segunda opção). O DUT deve descartar o pedido de eco enviado pelo UXM e não deve enviar uma resposta.

Parte C: A Primeira Opção tem Bits Mais Significativos 00b, A Seguinte tem Bits Mais Significativos 11b

1. O UXM transmite o pacote C para o endereço de ligação-local do DUT, um pedido de eco que tem opções de destino de cabeçalho com quatro opções desconhecidas. Os tipos de opções são 7, 199, 71, e 135.
2. O DUT deve enviar uma mensagem ICMPv6 de problemas de parâmetros para o UXM. O campo de código deve ser 2 (Opção IPv6 não reconhecida encontrada). O campo de ponteiro tem de ser 0x30 (deslocamento do campo tipo de opção da segunda opção). O DUT deve descartar o pedido de eco enviado pelo UXM e não deve enviar uma resposta.

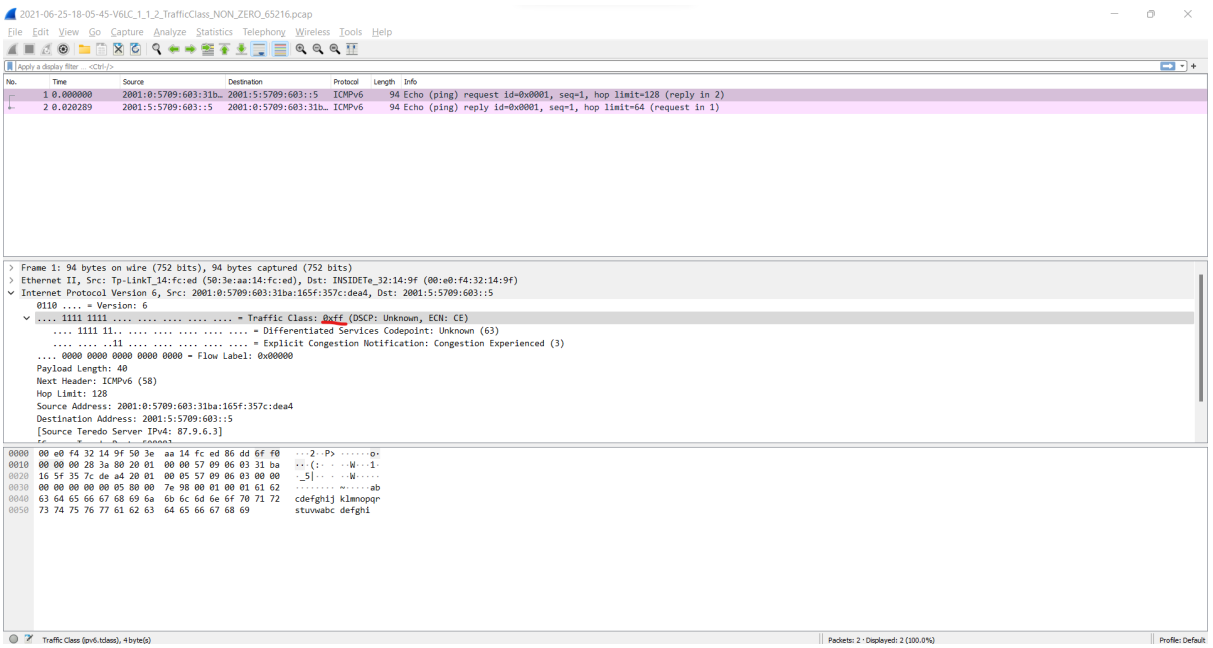
4. Resultados obtidos

Nesta seção serão descritos os testes realizados em um dispositivo que possui a tecnologia LTE. O software escolhido para apresentar os resultados foi o Wireshark, pois, apesar de ter sido utilizado o Pcap.Net para a construção e envio de pacotes, este não possui uma visualização amigável e facilitada para o usuário. Porém, o software mencionado é um dos principais softwares para análise de tráfego de rede tanto no meio acadêmico quanto na indústria, visto que possui diversas funcionalidades que permitem ao usuário uma rápida análise minuciosa do tráfego de pacotes na rede.

3.1 Teste v6LC.1.1.2: *Traffic Class Non-Zero – End Node*

O teste consiste no envio de um pacote IPv6 que possui o campo classe de tráfego diferente de zero, no qual o dispositivo em teste, caso suporte essa funcionalidade, deve enviar uma resposta ao pedido de eco com o valor de classe de tráfego diferente de zero. Na imagem abaixo pode-se ver os dois pacotes envolvidos no teste.

Figura 32: Imagem do arquivo pcap gerado pelo teste v6LC.1.1.2: *Traffic Class Non-Zero – End Node* ressaltando a mensagem enviada pela rede.

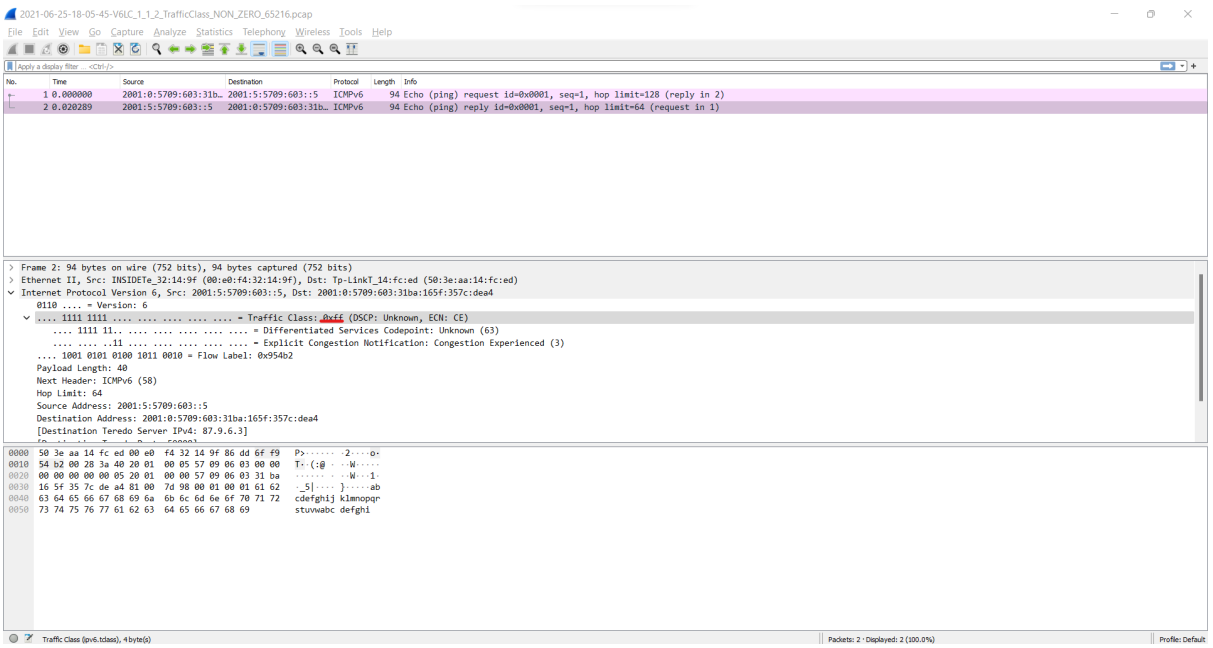


Fonte: Autoria do autor

Em vermelho está destacado o valor da classe de tráfego que é enviado para o DUT, neste caso ele assume um valor de 255, “0xff” em hexadecimal, tendo em mente que, para esse teste, basta que o valor de classe de tráfego seja diferente de zero.

Na imagem subsequente vê-se a resposta enviada do dispositivo sob teste ao instrumento, sendo ressaltado em vermelho o valor de classe de tráfego, o qual configura um caso positivo, pois foi retornado um valor diferente de zero, isto é, um valor de 255, “0xff” em hexadecimal.

Figura 33: Imagem do arquivo pcap gerado pelo teste v6LC.1.1.2: *Traffic Class Non-Zero – End Node* ressaltando a mensagem enviada pelo dispositivo.



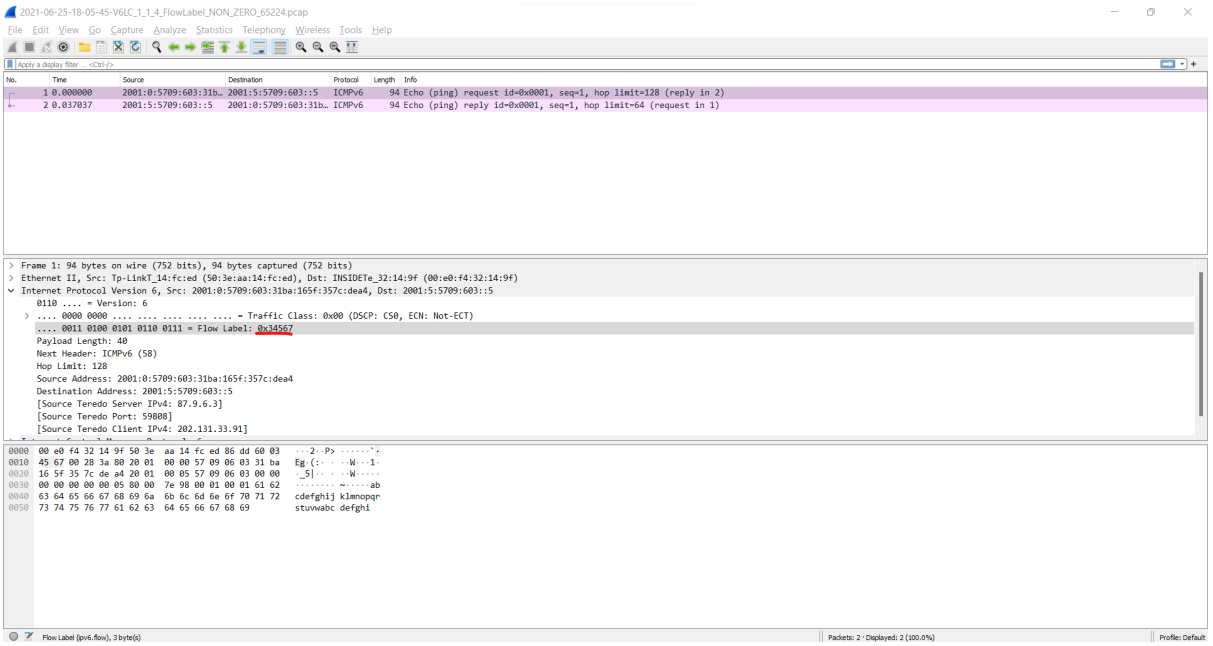
Fonte: Autoria do autor

3.2 Teste v6LC.1.1.4: *Flow Label Non-Zero*

Neste teste a rede deve enviar um pacote com um valor de rótulo de fluxo com um valor de 0x34567, sendo que o dispositivo em teste deve responder, caso suporte o uso do rótulo de fluxo, o pedido de eco com o campo identificador de fluxo, diferente de zero.

Na figura abaixo pode-se ver o pacote que é enviado do instrumento para o aparelho que possui a tecnologia LTE, ressaltando-se em vermelho o campo rótulo de fluxo que possui um valor especificado acima.

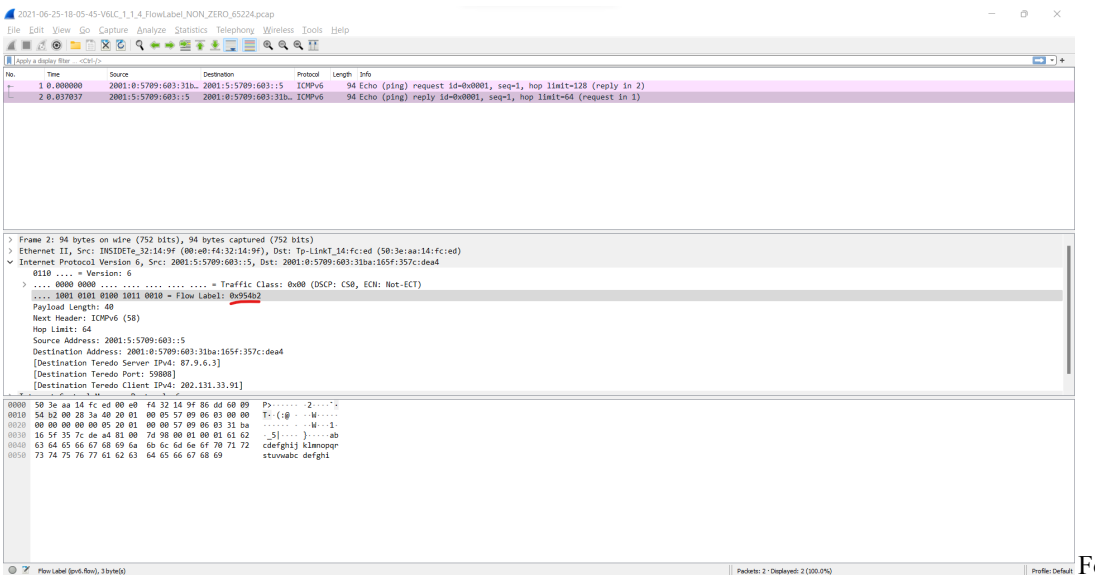
Figura 34: Imagem do arquivo pcap gerado pelo teste v6LC.1.1.4: *Flow Label Non-Zero* ressaltando a mensagem enviada pela rede.



Fonte: Autoria do autor

Na figura a seguir pode-se ver a resposta do DUT ao pacote enviado pela rede, em que se destaca o campo rótulo de fluxo que possui um valor diferente de zero.

Figura 35: Imagem do arquivo pcap gerado pelo teste v6LC.1.1.4: *Flow Label Non-Zero* ressaltando a mensagem enviada pela dispositivo.



Fonte: Autoria do autor

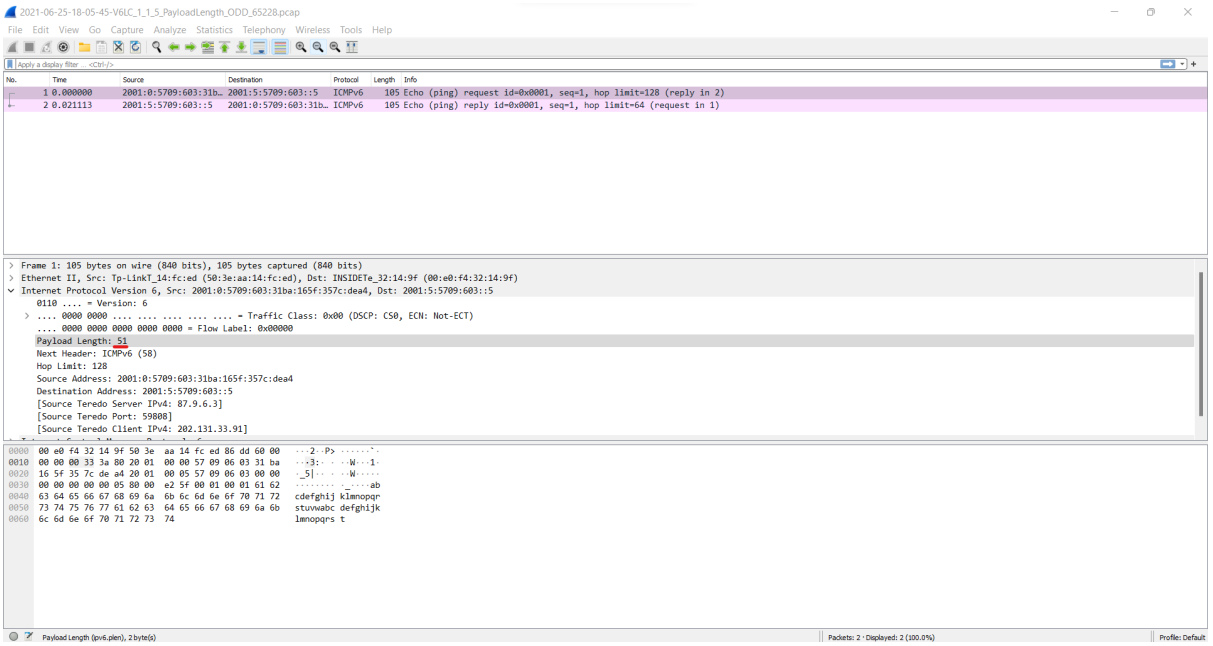
Diante da resposta com o valor de rótulo de fluxo diferente de zero, é retratado um caso positivo do teste, sendo que se a resposta fosse igual a zero ou se nenhuma resposta fosse enviada, representaria um caso negativo para o teste.

3.3 Teste v6LC.1.1.5: *Payload Length*

No presente teste temos um caso similar aos anteriores, pois será enviado um pacote com um campo que possui um valor específico, onde o campo mencionado é o campo *Payload length*. No entanto, esse teste possui mais de um valor dividindo-se em duas partes.

Na primeira parte será enviado campo tamanho dos dados igual a 51, “0x33” em hexadecimal, isto é, ele possui um valor ímpar. O dispositivo em teste deve enviar uma resposta de eco indicando que o pacote foi corretamente processado. Na figura abaixo vê-se o pacote enviado ao DUT e logo abaixo esse pacote está a resposta de eco, destacando-se em vermelho o campo supracitado.

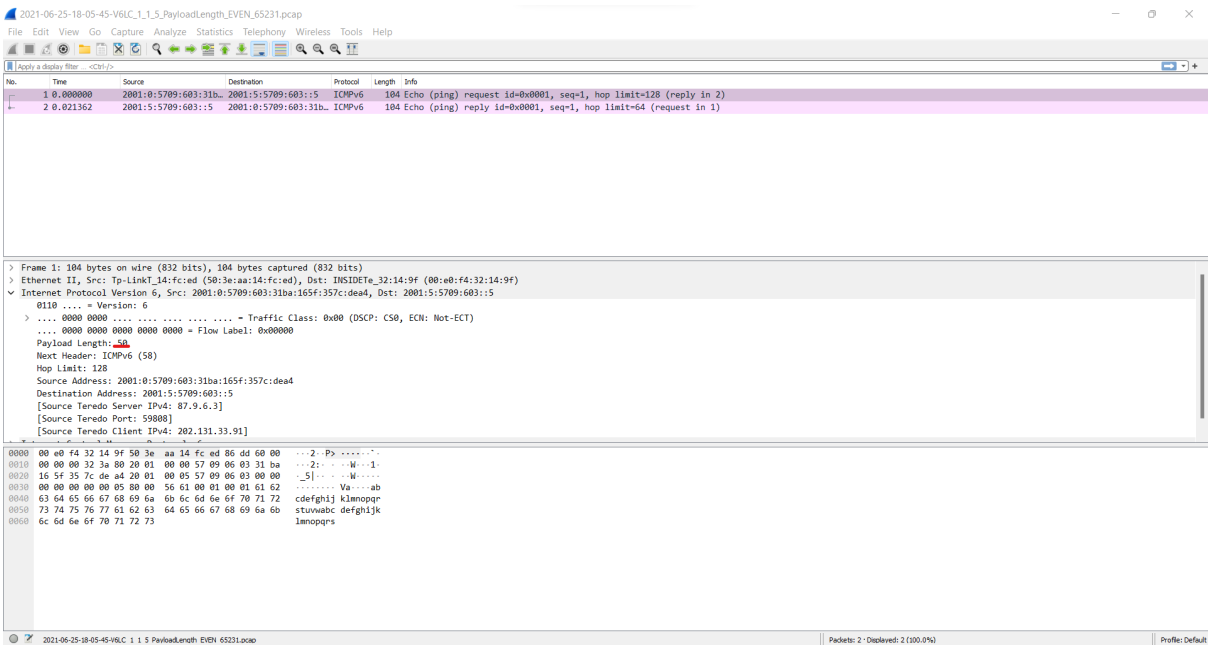
Figura 36: Imagem do arquivo pcap gerado pelo teste Teste v6LC.1.1.5: *Payload Length* ressaltando a mensagem enviada pela rede.



Fonte: Autoria do autor

Em seguida temos o segundo caso, em que o pacote tem o campo *Payload length* com o valor de 50, 0x32 em hexadecimal, isto é, um valor par. Pode-se ressaltar que o DUT não deve responder de maneira diferente da primeira parte, ou seja, deve ser enviado uma resposta de eco ao pacote enviado. Na imagem abaixo pode-se ver o pacote enviado ao DUT, em vermelho está o campo citado com o valor referenciado, logo em seguida está a resposta de eco solicitada.

Figura 37: Imagem do arquivo pcap gerado pelo teste Teste v6LC.1.1.5: *Payload Length* ressaltando a mensagem enviada pelo dispositivo.



Fonte: Autoria do autor

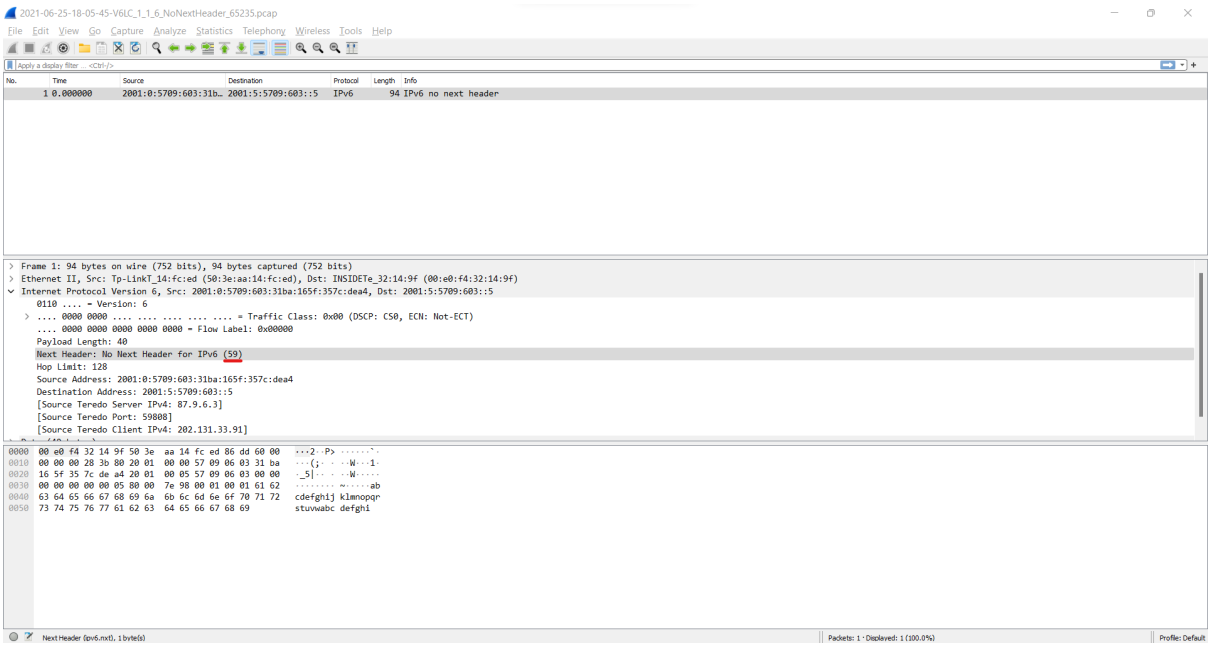
Destaca-se que, em ambos os casos, o DUT respondeu à solicitação de resposta de eco solicitada pela rede, caracterizando um caso positivo para o teste, caso o dispositivo não respondesse a solicitação, seria determinado um caso de falha.

3.4 Teste v6LC.1.1.6: *No Next Header after IPv6 Header*

A rede envia um pacote com o valor do campo próximo cabeçalho igual a 59, o que caracteriza um caso no qual não há um próximo cabeçalho. Dessa

forma o dispositivo sob teste não deve enviar nenhuma resposta à rede. Na imagem abaixo pode-se ver o pacote que é enviado para da rede para o DUT, em vermelho está ressaltado o valor do campo próximo cabeçalho.

Figura 38: Imagem do arquivo pcap gerado pelo teste *No Next Header after IPv6 Header* ressaltando a mensagem enviada pela rede.



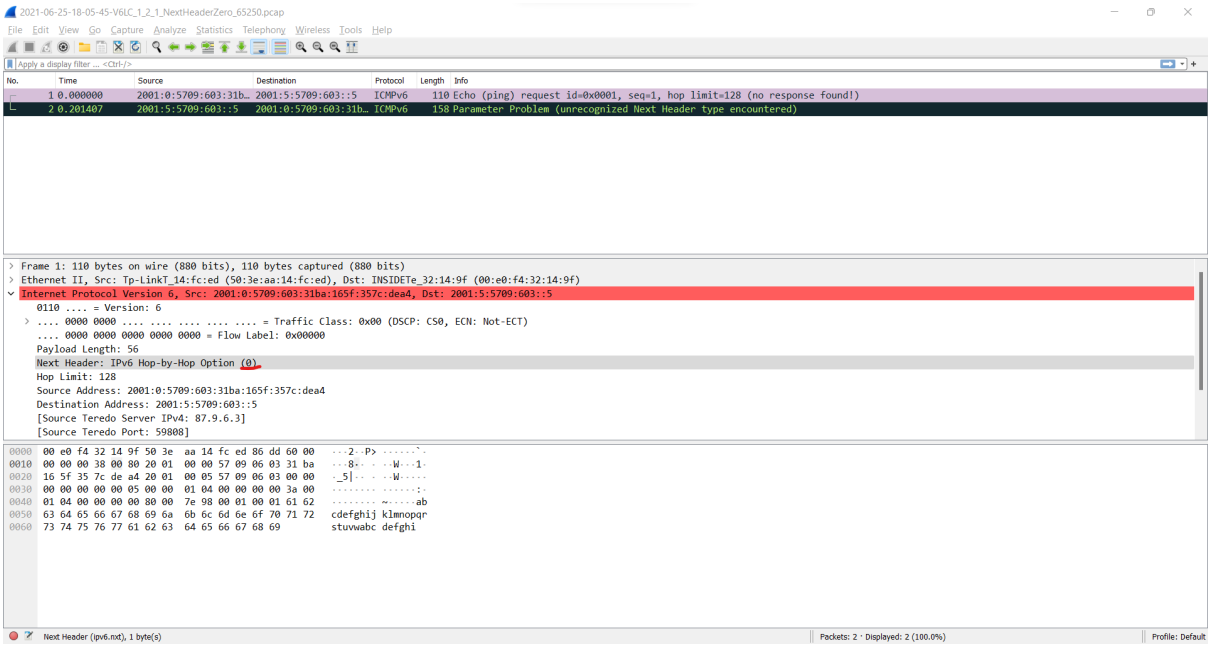
Fonte: Autoria do autor

Como pode-se ver, não há nenhum pacote além do pacote que é enviado pela rede, o que indica uma caso de sucesso no teste, no entanto caso o DUT enviasse qualquer pacote a rede, caracterizaria um caso de falha.

3.5 Teste v6LC.1.2.1: *Next Header Zero*

Neste caso, tem-se o primeiro teste em que o DUT irá enviar um pacote ICPMv6 informando que há um erro no pacote enviado pela rede, este erro é definido pelo campo próximo cabeçalho igual a zero, sendo que o dispositivo não deve ser capaz de processar esse tipo de pacote. Na imagem abaixo, vê-se o pacote enviado pela rede. Está destacado em vermelho o campo citado previamente com o valor igual a zero.

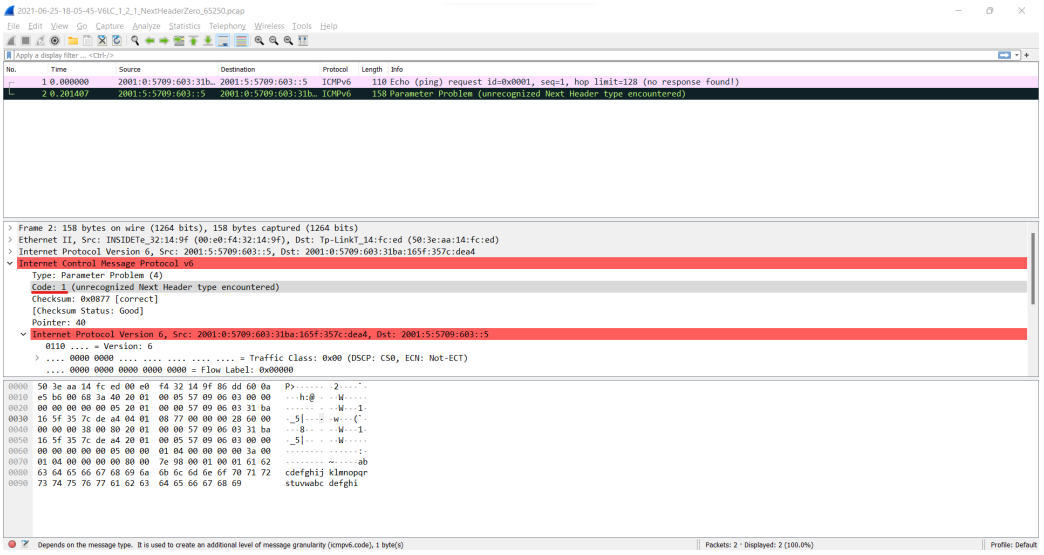
Figura 39: Imagem do arquivo pcap gerado pelo teste v6LC.1.2.1: *Next Header Zero* ressaltando a mensagem enviada pela rede.



Fonte: Autoria do autor

Na imagem a seguir, pode-se ver a resposta do pacote enviado, no qual destaca-se o código 1 do pacote ICPMv6, que indica que o campo de próximo cabeçalho não é reconhecido.

Figura 40: Imagem do arquivo pcap gerado pelo teste Teste v6LC.1.2.1: *Next Header Zero* ressaltando a mensagem enviada pelo dispositivo.



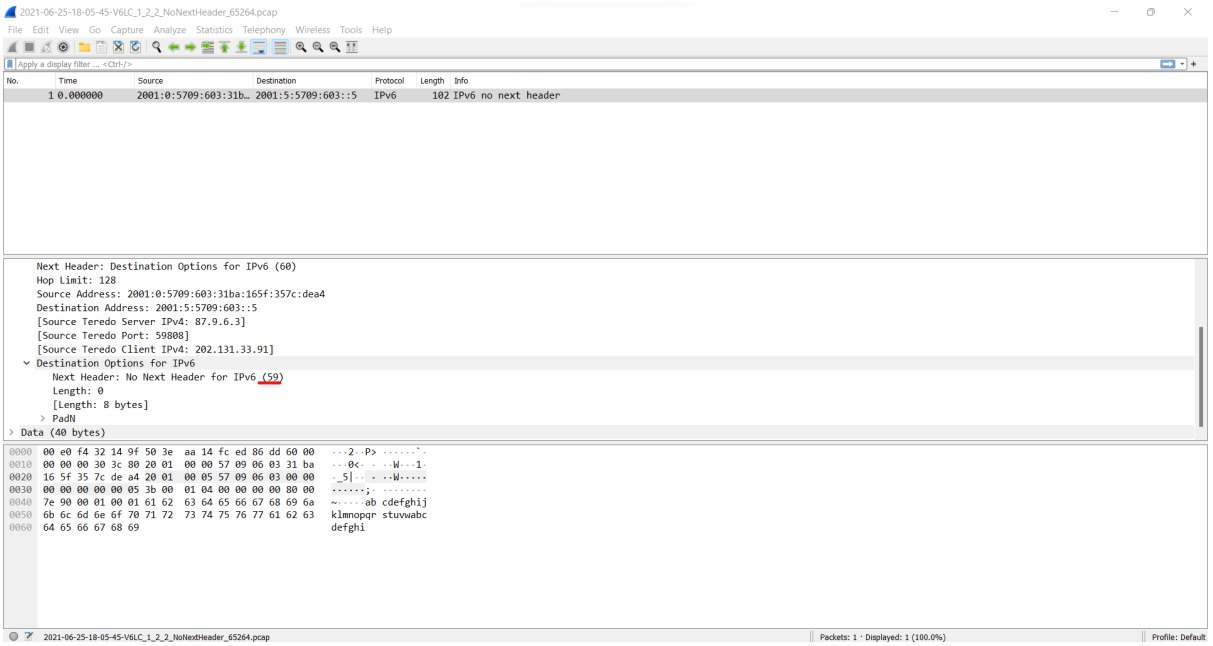
Fonte: Autoria do autor

3.5 Teste v6LC.1.2.2: *No Next Header after Extension Header*

Esse teste é definido pelo envio de um único pacote da rede para o dispositivo sob teste, no qual a rede enviará um pacote IPv6 com o campo próximo cabeçalho igual a 59, dessa forma o DUT deve processar o pacote de maneira que ele não irá esperar por um próximo pacote, sendo que o aparelho não deve enviar nenhum pacote em resposta para a rede.

Na imagem abaixo, pode-se ver o pacote enviado pela rede com o valor de 59 que está destacado em vermelho, pode-se ressaltar que não há outro pacote enviado neste teste.

Figura 41: Imagem do arquivo pcap gerado pelo teste v6LC.1.2.2: *No Next Header after Extension Header* ressaltando a mensagem enviada pela rede.



Fonte: Autoria do autor

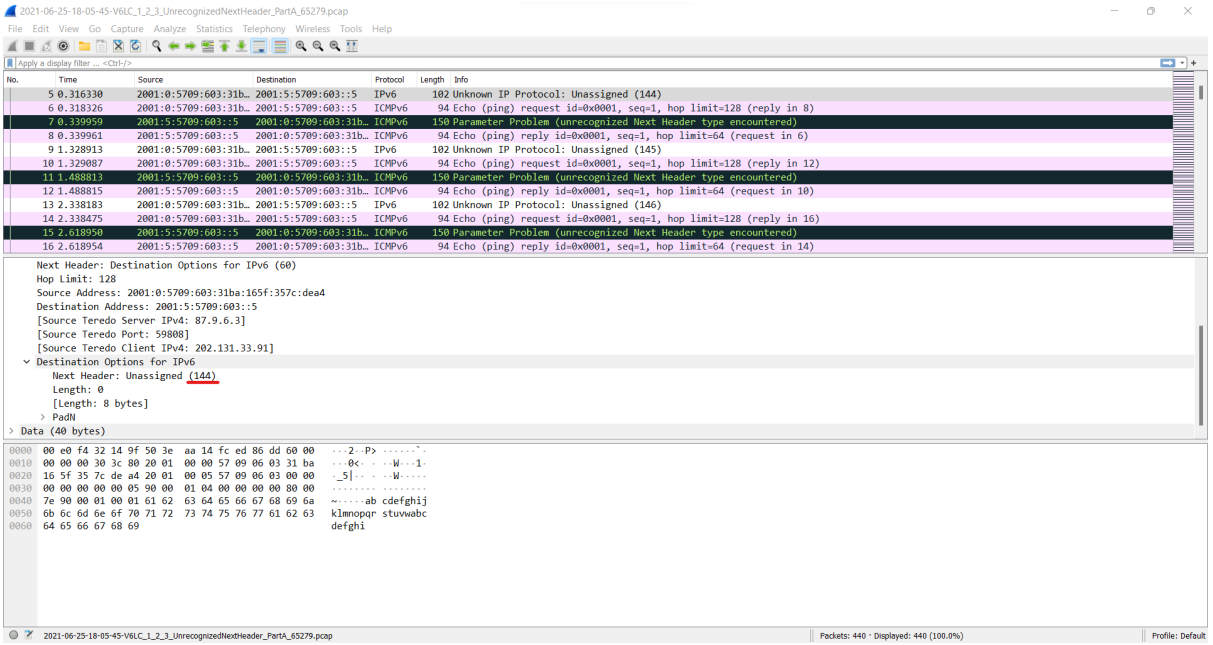
3.6 Teste v6LC.1.2.3: *Unrecognized Next Header in Extension Header – End Node*

O instrumento enviará diversos pacotes com valores entre 144 e 252 junto com uma solicitação de eco, e o dispositivo deve a resposta ao eco e enviar um

pacote ICPMv6 informando que o pacote não foi processado corretamente, pois os valores enviados do campo próximo cabeçalho não são reconhecidos.

Nas imagens abaixo pode-se ver a sequência citada anteriormente, em que é destacado em vermelho o campo de próximo cabeçalho com um valor que não é reconhecido pelo DUT.

Figura 42: Imagem do arquivo pcap gerado pelo Teste v6LC.1.2.3: *Unrecognized Next Header in Extension Header – End Node* ressaltando a mensagem enviada pela rede.



Fonte: Autoria do autor

É interessante notar que o padrão se repete para todas as possibilidades de próximo cabeçalho “não reconhecido”.

4. Conclusão

O presente trabalho aborda um tema de relevância no campo da engenharia de informação: a testagem do envio de pacotes IPv6 por dispositivos que suportam a tecnologia 4G. Especificamente, o estudo é baseado nas diretrizes e especificações descritas na RFC 2460, que é um documento fundamental para o protocolo IPv6.

Na introdução, foi realizada uma explanação do problema central que é o enviar pacotes IPv6 para dispositivos que operam em redes 4G, no qual o dispositivo sob teste precisa responder ao pacote enviado pela rede de maneira apropriada, independentemente de se o pacote estiver estruturado de forma correta ou não, tendo em mente que é necessário se comportar de maneira ilustrada na RFC 2460. A motivação para realizar esse trabalho está relacionada às crescentes demandas por serviços de internet e à necessidade de suporte para uma maior quantidade de dispositivos conectados. Além disso, são revisados trabalhos relacionados, como artigos científicos, teses e projetos, que abordam temas semelhantes e contribuem para a fundamentação teórica do estudo.

Na seção de fundamentação teórica, foram explorados conceitos essenciais para a compreensão do trabalho, incluindo o *Internet Protocol version 4* (IPv4), que é protocolo antecessor que foi explorado no decorrer do trabalho, o *Internet Protocol version 6* (IPv6), que foi o foco do presente trabalho, *Internet Control Message Protocol* (ICMPv6), que é um protocolo de controle e relatório de erros utilizado no IPv6, e também uma explicação sobre o FR1, que é uma gama de frequências que a tecnologia LTE pode operar,

Em seguida, foi descrita a plataforma de teste Wireless UXM 4G utilizada para realizar os experimentos. Essa plataforma é reconhecida pela sua capacidade de simular cenários de rede realistas e permite a análise do desempenho dos dispositivos que suportam 4G ao receberem pacotes IPv6. A

conexão entre o dispositivo de teste e o instrumento de teste é estabelecida seguindo procedimentos específicos, que são explicados no texto. São abordados aspectos como teste irradiado ou conduzido, em que são exploradas diferentes abordagens de envio de pacotes para o dispositivo em teste. Além disso, são fornecidas informações sobre a configuração do instrumento, que é crucial para garantir que os pacotes IPv6 sejam enviados corretamente e que os parâmetros relevantes sejam devidamente configurados.

Foi ilustrada a implementação do software utilizado no projeto, destacando as principais funcionalidades e a interface com o usuário. A linguagem C# foi utilizada pela fácil integração com o instrumento e a facilidade de enviar, de analisar um pacote IPv6 com o uso da biblioteca Pcap.Net, que a tarefa de explorar o protocolo com a tecnologia LTE muito mais inteligível.

Os testes realizados foram apresentados em seções específicas, em que são discutidos os diferentes aspectos avaliados. Cada teste é explicado em termos de sua finalidade, parâmetros analisados e metodologia utilizada.

Os resultados obtidos foram analisados e discutidos nas seções correspondentes. Foram apresentados os arquivos pcap gerados pelo pacote Pcap.Net utilizando a ferramenta Wireshark para uma melhor compreensão dos resultados dos testes. Foram destacadas tanto as observações positivas quanto as limitações encontradas durante a execução dos testes.

Em conclusão, o programa desenvolvido foi capaz de realizar uma série de testes abrangentes para certificar dispositivos móveis em relação ao protocolo IPv6. Estes testes permitiram verificar e garantir que os dispositivos estejam livres de falhas quando utilizados em campo. Ao oferecer uma aplicação robusta para a realização desses testes, a implementação desempenha um papel importante na garantia da interoperabilidade e desempenho dos

dispositivos móveis compatíveis com o IPv6. Desse modo, os fabricantes e fornecedores podem assegurar aos usuários finais que seus dispositivos estão adequadamente preparados para o futuro da Internet e que serão capazes de lidar com o crescente número de dispositivos conectados. Com base nos resultados obtidos através do programa, os fabricantes podem identificar e corrigir quaisquer problemas ou falhas antes do lançamento dos dispositivos no mercado, aumentando a satisfação do cliente e a reputação da marca.

No geral, o programa desenvolvido demonstrou ser uma ferramenta para a certificação de dispositivos móveis no que diz respeito ao protocolo IPv6, promovendo a excelência na qualidade dos produtos e garantindo uma experiência positiva aos usuários finais.

5. Referência bibliográfica

[1] MURUGESAN, R. K.; RAMADASS, S. IPv6 address distribution: An alternative approach. In: 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT), 2010.

[2] NAQVI, I. F.; SIDDIQUI, A. K.; FAROOQ, A. IPv6 adoption rate and performance in the 5G wireless internets. In: 2016 IEEE Region 10 Conference (TENCON), 2016.

[3] GUPTA, S. C.; GUPTA, G.; SARAN, H. New Vision for 5G Backbone Network Architecture. In: 2020 IEEE 3rd 5G World Forum (5GWF), 2020.

[4] FRANKEL, S.; GREEN, D. Internet Protocol Version 6. In: IEEE Security & Privacy, vol. 6, no. 3, pp. 83-86, Maio-Junho de 2008.

[5] INTERNET ENGINEERING TASK FORCE. RFC 2460: Protocolo de Internet Versão 6. Disponível em: https://www.rfc-editor.org/rfc/rfc2460.txt. Acesso em: 23 de novembro de 2022.

[6] INTERNET ENGINEERING TASK FORCE. RFC 791: Protocolo de Internet. Disponível em: https://www.rfc-editor.org/rfc/rfc791.txt. Acesso em: 23 de novembro de 2022.

[7] SHIRANZAEI, A.; KHAN, R. Z. Internet protocol versions: A review. In: 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015.

[8] PITIMON, I.; NINTANAVONGSA, P. An IPv6 network congestion measurement based on network time protocol. In: TENCON 2014 - 2014 IEEE Region 10 Conference, 2014.

[9] LEE, D. C.; LOUGH, D. L. The Internet Protocol version 6. In: IEEE Potenciais, vol. 17, no. 2, pp. 11-12, Abril-Maio de 1998.

[10] DEBBARMA, S.; DEBNATH, P. Internet protocol version 6 (IPv6) Extension Headers: Issues, challenges and mitigation. In: 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015.

[11] GEßNER, Christina. Long Term Evolution: A Concise Introduction to LTE and Its Measurement Requirements. Rohde & Schwarz, 2011. 216 p. ISBN 9783939837114.

[12] KSHATRIYA, S. N. S. et al. On interference management based on subframe blanking in Heterogeneous LTE networks. In: 5º International Conference on Communication Systems and Networks (COMSNETS), Bangalore, Índia, 2013, p. 1-7. DOI: 10.1109/COMSNETS.2013.6465557.

[13] HUAWEI. Captura de tela da página "ICMPv6". 2023. Disponível em: https://support.huawei.com/enterprise/en/doc/EDOC1000178170/d005b7c7/icmpv6. Acesso em: 14 de maio de 2022.

[14] KEYSIGHT TECHNOLOGIES. E7530A and E7630A LTE/LTE-A Test and Lab Applications.Disponível em:https://www.keysight.com/br/pt/assets/7018-04896/technical-overviews/5992-0920.pdf. Acesso em: 27 de setembro de 2022.

[15]TANENBAUM, Andrew S. Redes de Computadores. 5. ed. Rio de Janeiro: Elsevier, 2011. ISBN 9788535251811.

[16] IETF. "RFC 4443: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification." Disponível em: https://datatracker.ietf.org/doc/rfc4443/. Acesso em: 17 de dezembro de 2022.

[17] KITCHEN, Ronald. RF and Microwave Radiation Safety Handbook. 2nd ed. Oxford: Butterworth-Heinemann, 2001. ISBN 0-7506-43552.

[18] BAEK, Ji-Eun; CHO, Young-Maan; KO, Kwang-Cheol. Analysis of Design Parameters Reducing the Damage Rate of Low-Noise Amplifiers Affected by High-Power Electromagnetic Pulses. IEEE Transactions on Plasma Science, v. 46, n. 3, p. [páginas], mar. 2018.

[19] SILVA, Leandro Almeida da. Plataforma de testes para simulação de rede 4G e chamada VoLTE utilizando Rádio Definido por Software. Projeto Final de Curso - Universidade Federal do Amazonas, Faculdade de Tecnologia, Departamento de Engenharia Elétrica e Computação, Manaus, 2022.

[20] MACHADO, Letícia da Silva. Análise dos Métodos de Transição para o Protocolo IPv6. Trabalho de Conclusão de Curso - Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Curso Superior de Tecnologia em Redes de Computadores, Santa Maria, RS, Brasil, 2015

[20] SANTOS, Micael Souza Borges dos; VASCONCELLOS, Murilo. Testes de conformidade IPV6 de acordo com RFC 2460 em redes 3G e 4G.