Faculdade de Tecnologia de Americana – Ministro Ralph Biasi
Curso Superior de Tecnologia em Segurança da informação

Micael Souza Borges dos Santos

Murilo Vasconcellos

Testes de conformidade IPV6 de acordo com RFC 2460 em redes 3G e 4G

Americana, SP

2019

Faculdade de Tecnologia de Americana – Ministro Ralph Biasi

Curso Superior de Tecnologia em Segurança da informação

Micael Souza Borges dos Santos

Murilo Vasconcellos

Testes de conformidade IPV6 de acordo com RFC 2460 em redes 3G e 4G

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob a orientação do Prof.(a) Esp. Marcus Vinícius Lahr Giraldi.

Área de concentração: Segurança da Informação.

Americana, SP.

2019

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

S236t SANTOS, Micael Souza Borges dos

Testes de conformidade IPv6 de acordo com RFC 2460 em redes 3G e 4G. / Micael Souza Borges dos Santos, Murilo Vasconcellos. – Americana, 2019.

54f.

Monografia (Curso Superior de Tecnologia em Segurança da Informação) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Esp. Marcus Vinícius Lahr Giraldi

1 Internet – rede de computadores 2. Unix – sistema operacional I. VASCONCELLOS, Murilo II. GIRALDI, Marcus Vinícius Lahr III. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.519

681.3.066

Micael Souza Borges dos Santos
Murilo Vasconcellos

Testes de conformidade IPV6 de acordo com RFC 2460 em redes 3G e 4G

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Segurança da Informação, pelo CEETEPS/Faculdade de Tecnologia – Fatec/ Americana.
Área de concentração: Segurança da Informação.

Americana, 09 de dezembro de 2019.

Banca Examinadora:

Marcus Vinícius Lahr Giraldi (Presidente)
Especialista
Fatec Americana

Maria Cristina Aranda (Membro)
Doutora
Fatec Americana

Adriano Cilhos Doimo (Membro)
Especialista
Fatec Americana

AGRADECIMENTOS

Agradecemos ao nosso orientador pelas orientações recebidas e ao Grupo IBRACE por ter sido um grande incentivador para realizar esse trabalho e buscar conhecimentos sobre IPV6 e sua interoperabilidade.

RESUMO

A tecnologia mudou a maneira de como as pessoas fazem as coisas no seu dia, seja para realizar uma simples transferência financeira ou para pedir comida pelo dispositivo móvel. O fato é que o protocolo de internet está presente em todos os processos, diretamente ou indiretamente, aumentando vertiginosamente o número de dispositivos conectados à rede todos os anos, criando a necessidade de maiores endereçamentos no protocolo. Para atender esta demanda instaurou-se o padrão IPv6, e para garantir seu pleno funcionamento foram desenvolvidos testes automatizados, mas todos em ambientes controlados, pela sua facilidade de desenvolvimento e maior simplicidade de execução, uma vez que a organização não precisará se preocupar com configurações externas. Entretanto, este modelo de ambiente exige maior conhecimento técnico do desenvolvedor, pois ele precisará se certificar e entender se os testes foram realizados de forma adequada e seus possíveis erros encontrados. A proposta nesta monografia, é apresentar aos desenvolvedores menores uma solução já modificada de acordo com a regras que a Anatel exige para a certificação de dispositivos no Brasil. Com pouco conhecimento em Unix o desenvolvedor poderá testar seus dispositivos conectado em IPv6 diretamente de um servidor, obtendo no final um relatório apresentando os problemas encontrados, concentrando seus esforços no aprimoramento do dispositivo.

Palavras-chave: ANATEL; IPv6; Unix;

ABSTRACT

Technology has changed the way people do things in their day, whether it's to make a simple financial transfer or to order food from their mobile device. The fact is that the internet protocol is present in all processes, directly or indirectly, dramatically increasing the number of devices connected to the network every year, creating the need for greater protocol addresses. To meet this demand, the IPv6 standard has create up and automated testing has been developed to ensure its full operation, but all in controlled environments, due to their ease of development and greater simplicity of execution, since the organization does not have to worry about configurations external. However, this environment model requires more technical knowledge from the developer as they will need to make sure and understand if the tests were performed properly and their possible errors found. The proposal in this monograph is to present to smaller developers a solution already modified according to the rules that Anatel requires for device certification in Brazil. With little knowledge of Unix, the developer will be able to test their IPv6-connected devices directly from a server, ultimately getting a report outlining the issues they have encountered, focusing their efforts on device enhancement.

Keywords: ANATEL; IPv6; Unix;

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ANATEL	Agência nacional de telecomunicações
DAD	Duplicate address detection
DHCPv6	Internet control message protocol versão 6
DVD	Disco digital versátil
ERB	Estação rádio base
ETSI	European telecommunications standards institute
HTML	Hypertext markup language
ICMPv6	Internet control message protocol versão 6
IETF	Internet engineering task force
INMETRO	Instituto nacional de metrologia, qualidade e tecnologia
IP	Protocolo de internet
IPv4	Protocolo de Internet de quarta geração
IPv6	Protocolo de Internet de sexta geração
M	More
MAC	Media access control
MTU	Maximum transmission unit
NA	Neighbor advertisement
NDP	Protocolo de descoberta de vizinhos
NS	Neighbor solicitation
NUT	Node under test
RA	Router advertisement
RFC	Request for comments
RS	Router solicitation
TN	Tester Node

SUMÁRIO

1 INTRODUÇÃO

Para a realização da certificação de aparelhos com IPV6 no Brasil, a fabricante do dispositivo, necessita submeter o mesmo à certificadores credenciados junta à ANATEL, Agência Nacional de Telecomunicações, as quais não possuem uma ampla concorrência, ou seja, o número de empresas que prestam este tipo de serviço é muito limitado, até a data da publicação, cerca de meia dúzia.

Em decorrência da falta de concorrência, os valores cobrados pelas organizações certificadoras tornam-se elevados e são cobrados todas as vezes em que o dispositivo retorna para os testes (manutenção).

Com o avanço exponencial tecnológico nos últimos anos e com um número maior de dispositivos eletrônicos que assim fazem parte da era da internet das coisas, houve a necessidade de criação de um novo protocolo de endereçamento, que suportasse um número maior de IPs em sua estrutura e que suprisse algumas necessidades específicas e com a preocupação da segurança da informação. O protocolo de endereçamento na sexta versão (IPV6) é a solução para esses problemas, dentre eles: escassez de endereços; novo formato de cabeçalho; melhora no roteamento; configuração de endereço *stateless* ou *stateful* e além de um novo protocolo de descoberta de vizinhança (NDP).

Com o novo protocolo, instaurou-se um consenso, um padrão de desenvolvimento entre os desenvolvedores, para que a interoperabilidade aconteça deve existir um documento normatizando o protocolo, não com a intenção de obrigatoriedade, destacando-se a organização IETF, Força Tarefa da Engenharia da Internet, em inglês, Internet Engineering Task Force. Dentre os inúmeros consensos publicados a respeito do comportamento e ideais em relação ao IPv6, a principal é a “Protocolo Internet, Versão 6 (IPv6) Especificações”, conhecida apenas como, RFC 2460[1], que mais tarde, quando foi atualizada, tornou-se a RFC 8200[2]. No Brasil, até a publicação desta monografia, a RFC 2460 é utilizada como documento normativo para certificação junto ao órgão ANATEL. Entretanto, essa obrigatoriedade imposta pelo Agência Nacional de Telecomunicações tem como documentos normativos

[1] "RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification." Disponível em https://tools.ietf.org/html/rfc2460. Acessado em 26 nov.. 2019.
[2] "RFC 8200 - Internet Protocol, Version 6 (IPv6 ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc8200. Acessado em 26 nov.. 2019.

complementares os procedimentos de teste do comitê IPv6 *Ready,* que colabora ativamente com ferramentas de código aberto e na RFC 2460, que se tornou obsoleta sendo importante tais exigência no quesito transição do protocolo IPv4 para IPv6 e nos quesitos segurança e conformidade, com determina as especificações do IETF.

O objetivo deste trabalho é apresentar a modificação na ferramenta de testes de IPv6 de acordo com os critérios da ANATEL, no escopo global, uma vez que a ferramenta disponibilizada pelo IPv6 Ready funciona apenas de forma local.

Embora a ferramenta esteja disponível de forma *open source* pelo comitê IPv6 *Ready*, permitindo inclusive modificações conforme a necessidade do usuário, há necessidade de um terminal Unix para ser executada, podendo haver algumas dificuldades caso seja executada em determinadas topologias, pois existe a necessidade da modificação dos trinta e três testes conforme a topologia e até conhecimentos de programação Perl e principalmente da distribuição FreeBSD.

O tema surgiu devido aos preços impostos pelos laboratórios certificadores e seus altos preços em manutenção de certificados, é interessante relembrar que existem ferramentas de testes de conformidade e interoperabilidade IPv6 gratuitas na Internet, de fácil *download* e até mesmo de entendimento. Entretanto, tais ferramentas muitas vezes estão em inglês ou em idiomas de países asiáticos, o que impossibilita a disseminação delas e consequentemente a cobrança de altas quantias para a realização dos testes.

2 EVOLUÇÃO DO PROTOCOLO DE INTERNET

A tecnologia evolui consideravelmente todos os anos, conectando cada vez mais pessoas e facilitando processos dentro de organizações, não se faz muito tempo onde para compartilhar fotos de família precisava armazená-las em um *pen drive* ou em DVD, para então entregar em mãos para nosso ente querido. Se pensar em uma organização de médio irá perceber o quanto isso era ineficiente, suponhamos que o financeiro estaria recebendo informações de todos os setores, como folha de pagamento de funcionários e fornecedores, gastos pontuais, investimentos e afins.

Para solucionar este tipo de ineficiência a tecnologia caminhou para o desenvolvimento de redes de computadores utilizando o protocolo de endereçamento *internet*, o IP, onde um dispositivo conectado na rede recebe uma identificação de 32 bits, conseguindo assim enxergar outro e realizar a troca de informação, como arquivos e dados. Esta ideia de conectividade utilizando o Protocolo de Internet, o IP, foi crucial para o avanço tecnológico, conectando o mundo como um todo, a internet. Com ela por exemplo, os estudantes conseguem acessar o acervo acadêmico da sua faculdade em casa e a organização consegue centralizar os dados em um único ambiente, simplificando o acesso aos seus colaboradores.

Como já mencionado, a tecnológica tem o seu avanço quase que exponencial, necessitando de padronização e disponibilidade. Isso resultou com o amplo apoio da quarta versão do IP, o IPv4, que é utilizado até os dias de hoje com quatro octetos (bits) representados no formato decimal como, por exemplo, "192.168.1.2". Neste tipo de formatação é possível conectar cerca de 4,3 bilhões de endereços únicos.

A primeira vista, é comum ter a impressão de que a disponibilidade do IPv4 é o suficiente para atender a demanda atual, mas com a popularização da tecnologia e o crescente número populacional, ele tornou-se o suficiente para atender apenas metade da população mundial, isso se for levado em consideração que todas as pessoas possuem um único dispositivo conectado.

Para suprir este cenário alarmante foi instaurado e padronizado o sucessor do IPv4, o IPv6.

O IPv6 é o sucessor do IPv4, embora sua implementação esteja em passos largos, muitos desenvolvedores e fornecedores de componentes ainda precisam atualizar suas soluções já em funcionamento. Como destacado na base de

conhecimento da IBM Brasil[3], a sexta geração do IP traz benefícios não apenas na carência de endereços, mas em diversos outros segmentos, como:

Em configurações, por exemplo, diferente do IPv4 que necessita ser atribuído o endereçamento e sua tabela de encaminhamentos de IP antes do funcionamento, na sexta versão a autoconfiguração pode não conter registro e ir adicionando os encaminhamentos conforme a descoberta. No cabeçalho, houve também simplificações, transformando de um comprimento variável de 20 a 60 bytes para um tamanho fixo de 40 bytes.

Com número de endereçamento do IPv6 é o principal motivo para o seu desenvolvimento, o IPV6.BR destaca, que com o endereçamento em 128 Bits, é possível obter aproximadamente 340.282.366.920.938.463.463.374.607.431.768.211.456 endereços únicos ou 56 octilhões de endereços por ser humano, quando considerado uma população de 6 bilhões de habitantes. (IPV6.BR,[2012])

[3] "Comparação entre IPv4 e IPv6 - IBM." Disponível em https://www.ibm.com/support/knowledgecenter/pt/ssw_ibm_i_73/rzai2/rzai2compipv4ipv6.htm. Acessado em 17 nov.. 2019.

3 TIPOS DE MENSAGENS NO IPV6

Para formar a rede de computadores, onde um dispositivo conectado nela consegue enxergar outro e realizar a troca de informação, como arquivos e dados, está estruturada em cima dos protocolos que consistem na troca de determinadas mensagens entre os dispositivos. E não seria diferente no protocolo de endereçamento na sexta versão, encontra-se protocolos de controle de mensagens, o ICMPv6[4], protocolo de descoberta de vizinhos, NDP e o protocolo de configuração dinâmica, o DHCPv6[5].

Todos os protocolos e mensagens mencionados a seguir são padronizados segundo a referência "*Internet Protocol Version* 6", a RFC 3513[6], escrita pela Cisco, publicado em 2003.

3.1 ICMPv6

O ICMPv6 é usado pelos dispositivos que suportam IPv6 para relatar erros encontrados no processamento de pacotes e para executar funções da camada da Internet, como diagnóstico (ICMPv6 "ping"). O ICMPv6 é parte integrante do IPv6, o protocolo base. Todas as mensagens seguem o formato demonstrado na figura 1:

Figura 1 - Formato Geral dos Pacotes ICMPv6.

Fonte: IETF (2006)[7].

O campo Tipo, ou *Type*, indica o tipo da mensagem, como: Tempo excedido e

[4] "RFC 4443 - Internet Control Message Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4443. Acessado em 17 nov.. 2019.
[5] "RFC 8415 - Dynamic Host Configuration Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc8415. Acessado em 17 nov.. 2019.
[6] "RFC 3513 - Internet Protocol Version 6 (IPv6 ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc3513. Acessado em 17 nov.. 2019.
[7] "RFC 4443 - Internet Control Message Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4443. Acessado em 22 nov.. 2019.

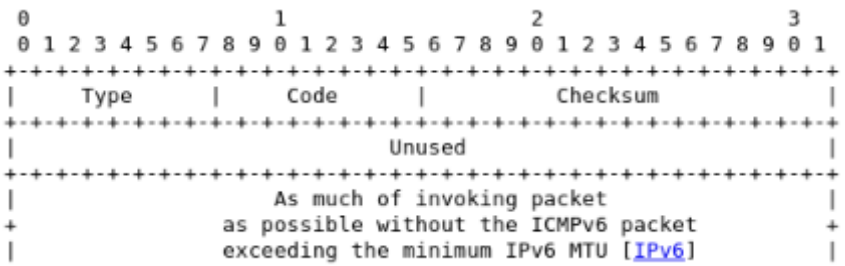
Destino Inacessível. O campo *Code* contém o identificador do tipo de mensagem, como: Falha na rota de origem e máquina de destino desconhecido. O campo de *Checksum*[8] é usado para detectar corrupção de dados na mensagem ICMPv6 e partes do cabeçalho IPv6. Já no campo *Message Body* estão as mensagens ICMPv6 que estão agrupadas em duas classes: mensagens de erro e mensagens informativas. Tais mensagens serão mais bem descritas nos tópicos seguintes.

3.1.1 Tipos de Mensagens

Os tipos de mensagens estão divididos em dois grupos como dito acima, sendo mensagens de erro e mensagens informativas. As mensagens de erro são as mensagens destino inalcançável, em inglês *Destination Unreachable,* pacote muito grande*, Packet Too Big,* tempo excedido*, Time Exceeded e* parâmetro com problema*, Parameter Problem*. Quanto às mensagens informativas, possui apenas duas, o *Echo Request e Echo Reply.*

Uma mensagem de Destino Inacessível, conforme a figura 2, é gerada por um roteador ou pelos dispositivos de origem que suportam o IPv6, em resposta a um pacote que não pode ser entregue ao seu endereço de destino por outros motivos que não seja o congestionamento.

Figura 2 - Formato do Pacote ICMPv6 Destination Unreachable.



Fonte: IETF (2006)[9].

Campos ICMPv6: *Destination Unreachable.*

Type 1

Code 0 - *No route to destination*

[8] "RFC 1071 - Computing the Internet checksum - IETF Tools." Disponível em https://tools.ietf.org/html/rfc1071. Acessado em 17 nov.. 2019.
[9] "RFC 4443 - Internet Control Message Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4443. Acessado em 22 nov.. 2019.

1 - *Communication with destination administratively prohibited*

2 - *Beyond scope of source address*

3 - *Address unreachable*

4 - *Port unreachable*

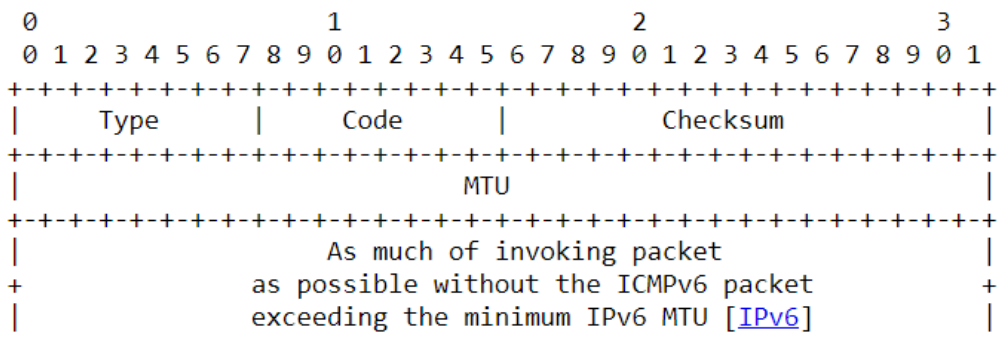
5 - *Source address failed ingress/egress policy*

6 - *Reject route to destination*

O campo *Unused,* não é utilizado para todos os valores de código. Ele deve ser inicializado em zero pelo emissor e ignorado pelo receptor.

Um pacote muito grande, conforme a figura 3, é enviado por um roteador em resposta a um pacote que ele não pode encaminhar porque o pacote é maior que o MTU, sigla em inglês para Maximum Transmission Unit, em português Unidade Máxima de Transmissão, do *link* de saída. O campo MTU, é o responsável por informar ao dispositivo qual o tamanho máximo que o pacote pode trafegar dentro da rede.

Figura 3 - Formato do Pacote ICMPv6 Packet Too Big Message.



Fonte: IETF (2006)[10].

Campos ICMPv6: *Packet Too Big Message.*

Type 2

Code Defina como 0 (zero) pelo emissor e ignorado pelo receptor.

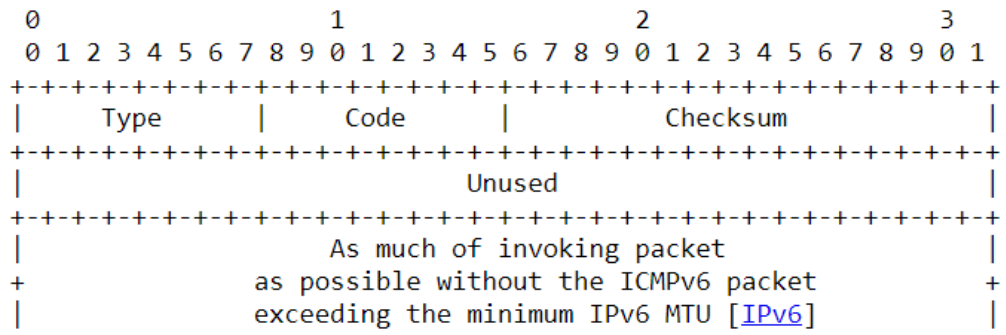
MTU A unidade máxima de transmissão do link do próximo salto.

Se um roteador recebe um pacote com um limite de salto zero, ou se um roteador diminui o limite de zero para um pacote, o mesmo descarta o pacote e origina uma mensagem ICMPv6 de tempo excedido com o código 0 para a origem do pacote,

[10] "RFC 4443 - Internet Control Message Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4443. Acessado em 22 nov.. 2019.

conforme a figura 4. Isso indica um *loop* de roteamento ou um valor inicial de limite de salto muito pequeno.

Figura 4 - Formato do Pacote ICMPv6 Time Exceeded Message.



Fonte: IETF (2006)[11].

Campos ICMPv6: *Time Exceeded Message.*

Type 3

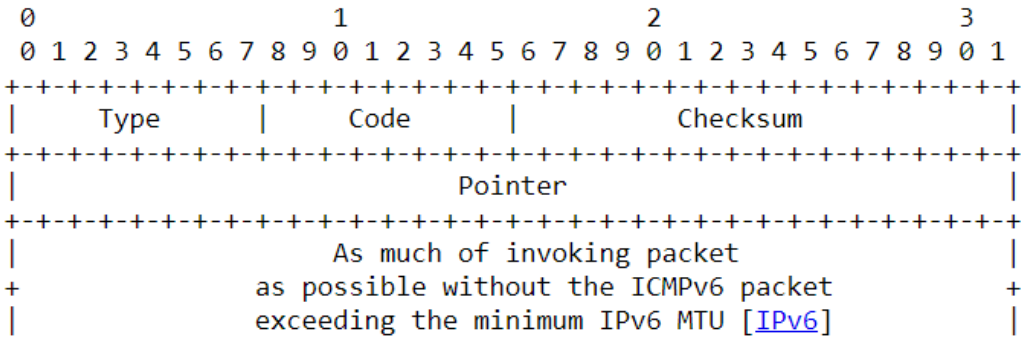
Code 0 - *Hop limit exceeded in transit*

1 - *Fragment reassembly time exceeded*

Caso um nó IPv6 esteja processando um pacote e encontra um problema com um campo no cabeçalho IPv6 ou nos cabeçalhos de extensão, de modo que não possa concluir o processamento do pacote, o mesmo descarta o pacote e origina uma mensagem de Problema de Parâmetro ICMPv6 na origem do pacote, indicando o tipo e localização do problema, conforme a figura 5.

[11] "RFC 4443 - Internet Control Message Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4443. Acessado em 22 nov.. 2019.

Figura 5 - Formato do Pacote ICMPv6 Parameter Problem Message.



Fonte:IETF (2006)[12].

Campos ICMPv6: *Parameter Problem Message.*

Type 4

Code 0 - *Erroneous header field encountered*

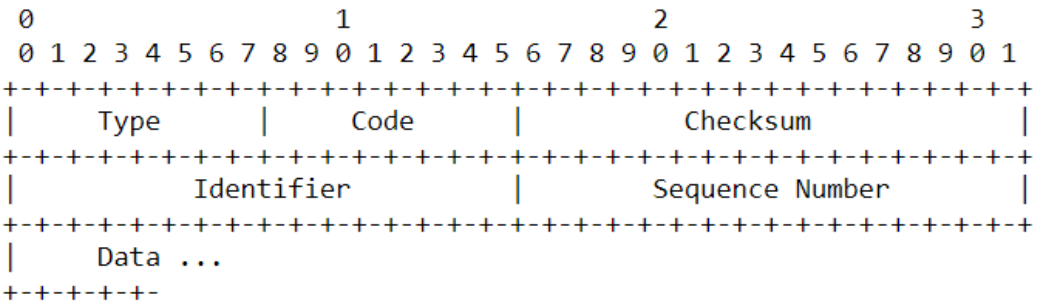
1 - *Unrecognized Next Header type encountered*

2 - *Unrecognized IPv6 option encountered*

O campo *Pointer* aponta além do final do pacote ICMPv6, se o campo com erro estiver além do que pode caber no tamanho máximo de uma mensagem de erro ICMPv6.

Mensagem de diagnóstico, ou seja, mensagem enviada para requisição/solicitação de um nó.

Figura 6 - Formato do Pacote ICMPv6 Echo Request Message.



Fonte: IETF (2006)[13].

Campos ICMPv6: *Echo Request Message.*

[12] "RFC 4443 - Internet Control Message Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4443. Acessado em 22 nov.. 2019.
[13] "RFC 4443 - Internet Control Message Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4443. Acessado em 22 nov.. 2019.

Type 128

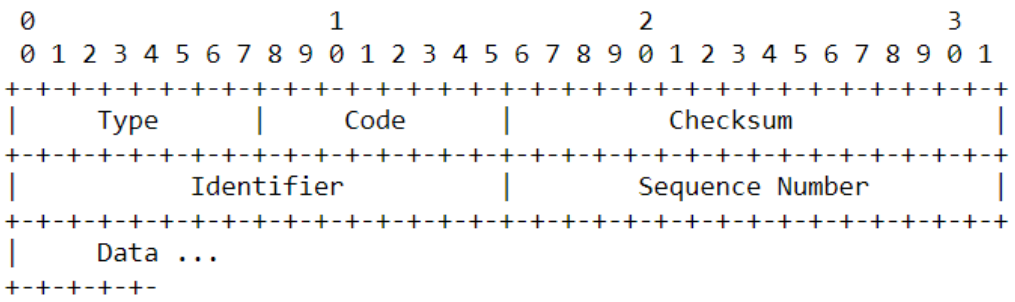
Code 0

O campo identificador ajuda na correspondência de respostas de *Echo* a uma solicitação de *Echo*.

O campo número de sequência conforme a figura 6 ajuda na correspondência de respostas de *Echo* a uma solicitação de *Echo*.

Cada nó implementa uma função de resposta de *Echo* ICMPv6 que recebe solicitações de *Echo* e origina respostas de *Echo* correspondentes conforme a imagem 7. Uma resposta de *Echo* é enviada em resposta a uma mensagem de solicitação de *Echo* enviada para um endereço IPv6 *multicast* ou *anycast*.

Figura 7 - Formato do Pacote ICMPv6 Echo Reply Message.



Fonte: IETF (2006)[14].

Campos ICMPv6: *Echo Reply Message.*

Type 129

Code 0

O campo identificador é usado para identificar a mensagem de solicitação de *Echo* de nó. Já o campo número de sequência é usado para identificar a sequência da mensagem de solicitação de *Echo* de nó. Quanto ao campo *date* contém os dados da mensagem de solicitação de *Echo* de nó.

3.1.2 O PROTOCOLO DE DESCOBERTA DE VIZINHANÇA

O protocolo descoberta de vizinhança (*Neighbor Discovery Protocol*) é usado pelos dispositivos para determinar e reconhecer diversas características na rede,

[14] "RFC 4443 - Internet Control Message Protocol ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4443. Acessado em 22 nov.. 2019.

como o reconhecimento dos endereços da camada de enlace (endereços MAC[15]) dos vizinhos conhecidos por residirem na mesma rede e limpar rapidamente valores em *cache* que ao se tornarem inválidos. Os *dispositivos* também usam o NDP para encontrar roteadores vizinhos que desejam encaminhar pacotes em seu nome. E por último o protocolo é utilizado para rastrear ativamente os vizinhos acessíveis e suas alterações. Quando um roteador ou o caminho para um roteador falha, o dispositivo procura ativamente alternativas utilizando-se das mensagens NDP.

O NDP além de reconhecimentos, é utilizado para estabelecer configurações de pacote e rede, como o conjunto de prefixos de endereço que definem os destinos que estão na rede, parâmetros da rede, como o MTU e parâmetros da Internet, como o valor limite de salto para colocar nos pacotes de saída, realizar a configuração automática de endereço, detecção de endereço duplicado (DAD) e como um roteador irá informar a um *host* de um nó de salto melhor para alcançar um destino específico.

O processo de descoberta de vizinhos define cinco tipos de pacotes ICMPv6 diferentes: um par de mensagens de solicitação de roteador (RS), anúncio de roteador (RA), mensagens de solicitação de vizinho (NS), anúncios de vizinho (NA) e uma mensagem de redirecionamento. Os objetivos das mensagens serão mais bem detalhado nos tópicos seguintes.

Quando uma *interface* é ativada, os *hosts* podem enviar solicitações de roteador, *Router Solicitation*, que solicitam aos roteadores para gerar anúncios de roteador imediatamente e não no próximo horário agendado.

Os roteadores anunciam sua presença juntamente com vários parâmetros de *link* e da Internet periodicamente ou em resposta a uma mensagem de solicitação de roteador. Os anúncios de roteador (*Router Advertisement*) contêm prefixos usados para determinar se outro endereço compartilha o mesmo *link* (determinação no *link*) e / ou configuração de endereço, um valor limite de salto sugerido etc.

O *Neighbor Solicitation* é enviado por um nó para determinar o endereço da camada de *link* de um vizinho ou para verificar se um vizinho ainda está acessível por meio de um endereço da camada de *link* em *cache*. Solicitações de vizinhos também são usadas para detecção de endereço duplicado.

O *Neighbor Advertisement* é uma resposta a uma mensagem de solicitação de vizinho. Um nó também pode enviar anúncios de vizinhos não solicitados para

[15] "RFC 7769 - Media Access Control (MAC ... - IETF Tools." Disponível em https://tools.ietf.org/html/rfc7769. Acessado em 17 nov.. 2019.

anunciar uma alteração de endereço da camada de *link*. (HINDEN *et al*, 2003).

4 DOCUMENTOS NORMATIVOS PARA O TESTE DE IPV6

O que compõem documentos normativos são os regulamentos, normas, procedimentos, especificações técnicas e normas editadas pelo Ministério das Comunicações que devem ser seguidas pelos laboratórios certificados junto a ANATEL. Para checar os escopos que um laboratório possui liberação para realizar testes de acordo com determinado documento normativo acesse o site do INMETRO e coloque a CRL[16] do laboratório em questão.

Dentro os documentos normativos para ensaios de IPV6 estão a RFC 2460 e o documento IPv6 *Core Protocols Test Specification* [17]do comitê IPV6 Forum. Ambos os documentos são usados para especificar os testes de conformidade, o *Test Specification* com os procedimentos de testes e a RFC 2460 com descrição/enumeração minuciosa das características do Internet *Protocol, Version* 6 (IPv6). É importante ressaltar que a RFC 2460[18] tinha o status de rascunho, com o simples objetivo de gerar discussão e sugestões de melhorias, consequentemente obter o estado de padronização, e foi o que aconteceu em 2017 com a RFC 8200, 19 anos após a publicação da RFC 2460.

4.1 IPv6 Ready

O documento em questão foi desenvolvido pelo comitê IPv6 Forum que tem um papel importante na reunião de fabricantes, no desenvolvimento e na implantação da nova geração do protocolo de endereçamento. Está especificação dos testes está organizada em grupos com base na metodologia ou objetivos relacionados. Com isso, cada grupo começa com um breve conjunto de comentários referentes a todos testes deste grupo e consequentemente por uma série de blocos, a seguir está descrito cada bloco de acordo com a norma:

Test Label: Etiqueta do teste é composto pelo nome curto do conjunto de testes, pelo número do grupo e pelo número de teste dentro do grupo, separados por pontos.

Purpose: Objetivo é uma breve declaração que descreve o que o teste tenta

[16] "Laboratórios de Ensaio - RBLE - Inmetro." Disponível em http://www.inmetro.gov.br/laboratorios/rble/. Acessado em 17 nov.. 2019.
[17] "IPv6 Core Protocols - IPv6 Ready Logo." Disponível em https://www.ipv6ready.org/?page=documents&tag=ipv6-core-protocols. Acessado em 17 nov.. 2019.
[18] "RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification." Disponível em https://tools.ietf.org/html/rfc2460. Acessado em 17 nov.. 2019.

alcançar.

References: A seção Referências lista referências cruzadas para as especificações e documentação que podem ser úteis para entender e avaliar o teste e os resultados.

Resource Requirements: A seção Requisitos específica o *software*, *hardware* e equipamento de teste que serão necessários para executar o teste.

Discussion: É uma discussão geral do teste e a seção relevante da especificação, incluindo quaisquer suposições feitas no *design* ou implementação do teste, bem como limitações conhecidas.

Test Setup: A seção Configuração do teste descreve a configuração de todos os dispositivos antes do início do teste. Diferentes partes do procedimento podem envolver etapas de configuração que diferem do que é fornecido na instalação de teste.

Procedure: Esta seção da descrição do teste contém as instruções passo a passo para a realização do teste. Essas etapas incluem descrições, como ativar *interfaces*, desconectar dispositivos da rede ou enviar pacotes de uma estação de teste. O procedimento de teste também leva o testador a fazer observações, que são interpretadas de acordo com os resultados observáveis fornecidos para essa parte do teste.

Observable Results: Esta seção lista os resultados observáveis que podem ser examinados pelo testador para verificar se o equipamento está funcionando corretamente. Quando vários resultados observáveis são possíveis, esta seção fornece uma breve discussão sobre como interpretá-los. A determinação de aprovação ou reprovação de cada teste geralmente se baseia em como o comportamento do equipamento se compara aos resultados descritos nesta seção.

Possible Problems: Esta seção contém uma descrição dos problemas conhecidos do procedimento de teste, que podem afetar os resultados do teste em determinadas situações.

Por fim, foque na seção 1 (RFC 2460) e seus grupos de testes, que está dividido em três grupos, sendo IPv6 *Header, Extension Headers and Options* e *Fragmentation*.

4.1.1 Objetivo da Seção 1 do Documento IPv6 Ready Test Specification Core Protocols

A seção abrange a especificação básica do Protocolo da Internet versão 6, que se encontra na RFC 2460, a seção também discute problemas de tamanho de pacote, a semântica de rótulos de fluxo e classes de tráfego e os efeitos do IPv6 em protocolos de camada superior. Ou seja, os testes descritos na seção foram projetados para verificar a prontidão de uma implementação do IPv6 em relação à especificação básica do IPv6.

O grupo 1 abrange os campos no cabeçalho IPv6 básico, verificando se um nó processa e gera adequadamente os campos Versão, Classe de tráfego, Rótulo de fluxo, Comprimento da carga útil, próximo cabeçalho e Limite de saltos no cabeçalho IPv6. Esses testes também verificam se um nó transmite as mensagens de Problema no Parâmetro ICMPv6 apropriadas em resposta a campos inválidos ou desconhecidos. Já o grupo 2 abrange o processamento de opções e cabeçalhos de extensão, particularmente as opções de salto por salto, opções de destino e cabeçalhos de roteamento, ou seja, os testes neste grupo verificam se um nó processa e gera adequadamente o campo comprimento do cabeçalho de extensão nos cabeçalhos de extensão e os campos Tipo de opção e Comprimento dos dados da opção nas opções IPv6. Esses testes também verificam se um nó processa corretamente as opções do cabeçalho em ordem, pacotes com um cabeçalho de roteamento destinado ao nó e muitos cabeçalhos ou opções de extensão em um único pacote. Além disso, estes testes garantem que um nó gere a mensagem ICMPv6 adequada em resposta a um campo inválido ou desconhecido. E para finalizar o grupo 3 que cobre a fragmentação no IPv6, os testes neste grupo verificam se um nó expira adequadamente a remontagem de fragmentos, abandona a remontagem em pacotes que excedam um tamanho máximo, processa fragmentos de *stub* e remontagem de fragmentos sobrepostos. Esses testes também verificam se um nó gera a mensagem ICMPv6 apropriada.

4.2 Internet Protocol, Version 6 (IPv6) Specification (RFC 2460)

O documento que especifica a versão 6 do Internet *Protocol* (IPv6), também conhecido como IP *Next Generation* ou Ipng funcionava como um rascunho, ele definia padrões da Internet para a comunidade da Internet e incitava a discussão e

sugestões de melhorias. O documento é um produto da Internet *Engineering Task Force* (IETF), porém, tornou-se obsoleto pela RFC 8200, que é o padrão de *Internet* oficial (Internet *Standard*). Para se tornar um protocolo padrão da internet como aconteceu com IPV6 em 2017, deve mostrar-se ampla utilização e seu amadurecimento nas implementações, como aconteceu com a nova geração do protocolo de endereçamento. Com o crescimento das tecnologias e com a nova geração da Internet, no caso a internet das coisas o IPv6 acabou se tornando o protocolo de endereçamento amplamente utilizado pelos fabricantes de produtos eletrônicos. Já para o amadurecimento é possível identificar inúmeras melhorias e sofisticações no protocolo, para evidenciar tal situação é só acessar a RFC 8200 na *internet* e ir até o apêndice B do documento padronizador, lá será enumerada as diferenças e modificações desde o *draft* RFC 2460.

Em 2014 a ANATEL começou a publicar requisitos técnicos para os produtos com *interface* aérea destinada aos serviços móveis que utilizam da nova geração do protocolo de endereçamento, com o objetivo de garantir os benefícios desse novo protocolo. Entretanto, última atualização foi em 2016, onde aplicaram a uso da RFC 2460, com objetivo de verificar a estrutura dos pacotes transmitidos em ICMPv6, e consequentemente se tornando um documento normativo.

Os testes de IPv6 tem como requisitos para atender três normas, sendo a RFC 2460 com análise das estruturas dos pacotes, ETSI 102 514[19] com a análise do endereçamento e a TS 36.523-1[20] do 3GPP que tem como intenção verificar se algumas das operações essenciais dos serviços móveis são executadas corretamente em conjunto com o protocolo.

[19] "TS 102 514 - V2.1.1 - Methods for Testing and ... - ETSI." Disponível em https://www.etsi.org/deliver/etsi_ts/102500_102599/102514/02.01.01_60/ts_102514v020101p.pdf. Acessado em 17 nov.. 2019.
[20] "Specification # 36.523-1 - 3GPP." Disponível em https://www.3gpp.org/DynaReport/36523-1.htm. Acessado em 17 nov.. 2019.

5 MODIFICAÇÕES NO V6EVAL PARA TESTES NO MODELO SERVIDOR/CLIENTE

Neste item será apresentado a ferramenta de teste, denominada V6eval e a nossa contribuição para que o projeto da TAHI funcionasse no escopo global e na *release* mais atual do FreeBSD. A ferramenta de teste é do mesmo comitê de um dos documentos normativos.

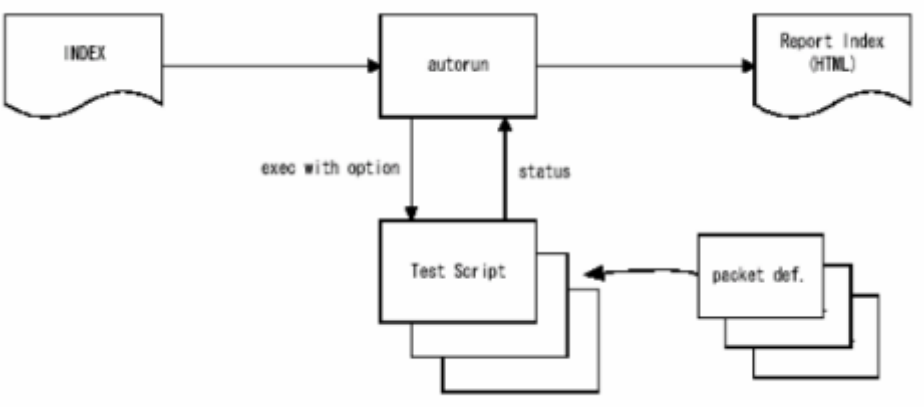
A ferramenta está dividida em duas estruturas, no primeiro momento só possui capacidade para ser compilada e instalada em um ambiente Unix, em específico em uma distribuição FreeBSD[21] 32 bits, porém, com algumas mudanças no código fonte, a ferramenta pode ser compilada independente do sistema de arquivo. Para que os testes de conformidade fossem feita por um servidor, foi realizada a instalação da ferramenta V6eval e do *Script Self_Test*[22] (*version* 5.0.1). É importante salientar que foram realizadas modificações nas duas partes/estruturas da ferramenta.

A principal ferramenta denomina-se V6eval, tem como objetivo executar todos os testes e reportar os resultados. Sua estrutura obedece a seguinte composição: um *Index,* que contém os testes que devem ser realizados, o programa, nomeado de *autorun,* utilizado para executar os testes em *host* e que examina o arquivo de configuração de destino (nut.def) e o arquivo de configuração do testador (tn.def), decidindo assim se deve realizar o teste.

Com as informações inseridas, basta executar os *scripts* de testes, obtendo um arquivo HTML com todos os *status* realizados. Este comportamento está evidenciado na figura 8, que foi retirado no manual da ferramenta em questão.

[21] "FreeBSD 12.0-RELEASE Announcement." 11 dez.. 2018, Disponível em https://www.freebsd.org/releases/12.0R/announce.html. Acessado em 21 nov.. 2019.
[22] "IPv6 Ready Logo Phase-2 Test." 21 nov.. 2018, Disponível em https://www.ipv6ready.org.cn/home/views/default/resource/logo/phase2-core/index.htm. Acessado em 17 nov.. 2019.

Figura 8 - Estrutura Comportamental da Ferramenta.



Fonte:Ipv6ready.([s.d])[23]

Para compilar a ferramenta de forma efetiva, dependências devem ser instaladas, e dentre elas estão: OpenSSL *Library*, Perl 5.10.1 ou superior, algumas *libs* perl sendo *Expert*, IO-Stty,IO-Tty, *Digest*-MD5 e p5-YAML. Para instalar todas essas dependências podem ser usadas o *ports*[24] ou qualquer gerenciador de pacotes da distribuição FreeBSD, atentando-se às versões a serem usadas.

Para que a ferramenta funcione no servidor de forma efetiva, ou seja, para que o servidor capture somente os pacotes ICMPv6 enviado pelo equipamento sob teste, a captura será e é feita mediante a duas condições, primeiro o MAC *Address* e depois pelo endereço de escopo global que o equipamento adquiriu ao se conectar a uma ERB[25].

No *script Self_Test* (*version* 5.0.1) existe o procedimento de testes e os arquivos com estrutura e campos dos pacotes ICMPv6 a serem enviados, assim como deve ser os pacotes de resposta. No diretório do *script* estão os arquivos de extensão .def e .seq que são os arquivos que são usados para fazer os testes, para ser mais exato nos arquivos .def contém as estruturas dos pacotes a serem enviados e os pacotes de respostas, já nos arquivos .seq estão os procedimentos de testes com as condições de análise, quais pacotes devem ser enviados e quais devem ser as respostas esperadas, cada teste tem um arquivo .seq e um .def. Com tudo isso fica fácil realizar a modificações no *script* de teste de cada item de ensaio, como os testes

[23] "v6eval reference manual." Disponível em https://www.ipv6ready.org.cn/home/views/default/resource/logo/release/v6eval-e.pdf. Acessado em 25 nov.. 2019.
[24] "About FreeBSD Ports." Disponível em https://www.freebsd.org/ports/. Acessado em 17 nov.. 2019.
[25] "ERB - teleco.com.br." Disponível em https://www.teleco.com.br/tutoriais/tutorialerb/pagina_1.asp. Acessado em 17 nov.. 2019.

vão ser realizados em produção, mudanças são exigidas no *script* de teste, principalmente no tipo de endereçamento e a na questão dos endereços MACs.

5.1 Ambiente Servidor

Para que o ambiente instalado no servidor fosse o mais seguro e mais atualizado possível, foi utilizado o sistema operacional FreeBSD assim como a ferramenta exige na documentação, entretanto, a última atualização da ferramenta foi em 2017, problemas com a *release* 12 do FreeBSD surgiram conforme a tentativa de compilação. Entre os problemas, estão a questão da compilação com versão mais recente da linguagem perl, e com isso foi optado por usar uma versão mais antiga, a versão 5.10 do perl5 e para isso acrescido a ferramenta perlbrew, consequentemente fazendo o *switch* para usar a versão do perlbrew ao invés da versão nativa do sistema operacional em questão, outro problema que surgiu é a versão 1.1.1a do Openssl, sendo necessário a utilização da versão 1.0.2s, entretanto, a finalizar a compilação da ferramenta v6eval, foi realizado atualização para a versão mais recente. Outros problemas surgiram ao compilar a ferramenta, porém, durante o *debug* esses problemas foram sanados. É pertinente recapitular, ao usar o perlbrew o diretório do /bin é modificado e consequentemente essa modificação foi refletida no código fonte da ferramenta.

Segue o quadro 1 com as características do ambiente a ser montado para que a ferramenta trabalhe de forma efetiva e correta:

Quadro 1 - Características e Dependências Servidor.

Ferramentas	Versão / *Release*
FreeBSD	12.0-*RELEASE*-p10 *GENERIC* i386
OpenSSL	1.1.1a-freebsd 20 Nov 2018
Perl5 (Perlbrew)	v5.10.1
p5-YAML	0.71
p5-*Expect*	1.35
p5-*Digest*-MD5	2.55
IO-Stty	0.03
IO-Tty	1.11

Fonte: Autoria própria.

5.2 Modificações Realizadas no Servidor

Para que o projeto funcionasse no escopo global e na *release* mais recente do FreeBSD, foram realizadas modificações no ambiente de teste e no código fonte. No tópico anterior essas modificações e adaptações foram descritas de forma sucinta, já neste capítulo são listado todas as etapas para montagem do servidor de teste e as modificações na ferramenta v6eval. No capítulo seguinte será descrito as modificações no *script* de teste.

Primeiro, foi realizado o *download* da ISO *release* 12 do FreeBSD, montando uma máquina virtual e fazendo suas respectivas modificações nesse ambiente controlado, para que todas as modificações posteriormente fossem passadas para servidor. Com o FreeBSD instalado, foram acrescentadas as dependências listadas no quadro 1, entretanto, problemas com o openssl 1.1.1a e algumas *structs* surgiram, por fim, foi utilizado a versão 1.0.2s, pois esses problemas de *structs* foram sanados. Outros problemas surgiram ao compilar o código fonte com a versão nativa do perl, o perl 5.26, mas para superar os problemas, foi necessário instalar perlbrew[26], que nos permite fazer o *download* da versão perl desejada e até mesmo nativa no sistema operacional , com isso, foi realizado o *download* da versão do 5.10.1 do perl. A ferramenta perlbrew faz todo o *download* e instalação no diretório *root*, assim como o bin, ao finalizar o *download* basta alterar o *switch* para que a versão 5.10.1 tornando-se nativa no sistema operacional.

Para que a ferramenta v6eval em específico fosse compilada utilizando a versão 5.10.1, algumas pequenas modificações foram feitas, alterando todas as linhas que fazem referência ao bin perl instalado no “/usr” (recursos de sistema Unix) e mudando para o executável do perlbrew instalado, que se encontra no diretório “/root[27]”.

Antes: #!/usr/bin/perl.

Depois: #!/root/perl5/perlbrew/perls/perl-5.10.1/bin/perl.

Uma outra modificação ainda no código fonte da ferramenta v6eval é alteração na linha 105 do arquivo LxLexer.cc, para obter um retorno do valor de 256.

A única dependência que deve ser instalada depois de todas modificações é o

[26] "Perlbrew." Disponível em https://perlbrew.pl/. Acessado em 20 nov.. 2019.
[27] "Introdução ao Sistema Operacional UNIX: Comandos - Unicamp." Disponível em https://www.cenapad.unicamp.br/servicos/treinamentos/tutorial_unix/unix_tutor-10.html. Acessado em 17 nov.. 2019.

módulo YAML[28] , na versão 0.71. Dado que sua utilização é dada para serialização de dados legíveis por humanos na hora de reportar o resultado, e tem relação com a versão do Perl, a versão que foi compilada para que *reports* fosse gerado da forma correta.

Para finalizar, todas as dependências instaladas e suas modificações foram realizadas com base no estudo da ferramenta e nos retornos que ela dava nas tentativas de compilação.

[28] "YAML - metacpan.org." Disponível em https://metacpan.org/pod/YAML. Acessado em 20 nov.. 2019.

6 DESCRIÇÃO DOS TESTES REALIZADOS

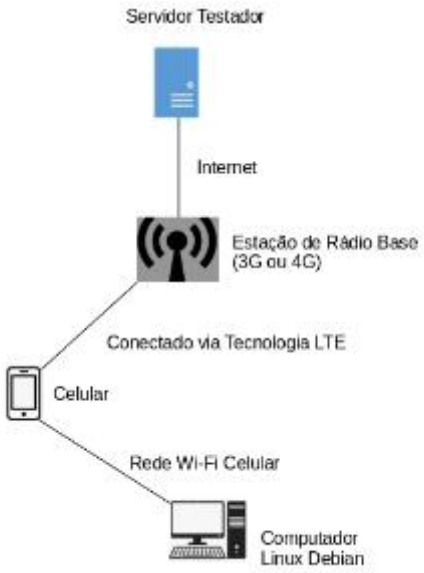
Dentre os documentos normativos apresentados, os documentos publicados pelo IPv6 Forum, são os que possuem maior embasamento prático, sendo divididos em três partes.

6.1 Topologia dos Testes

Os testes serão realizados com um servidor FreeBSD[29] 12.0-*RELEASE*-p10 FreeBSD 12.0-*RELEASE*-p10 *GENERIC* i386 instanciado na Vultr[30] ,uma empresa de nuvem e indústria de hospedagem gerenciada. Como os testes são análises dos pacotes de respostas ICMPv6, precisava de uma provedora de instâncias que faria a entrega de endereços IPv6 de escopo global, para que consequentemente pudéssemos fazer uso deste endereço 2001:19f0:b001:c3e:5400:2ff:fe55:8474, para enviar as requisições *Echo*.

A figura 9 está baseada em uma topologia de um dos testes realizados para evidenciar se os pacotes ICMPv6 estavam sendo enviado para o *host* sob teste independente do continente.

Figura 9 - Topologia dos Testes.



Fonte: Autoria própria.

[29] "The FreeBSD Project." Disponível em https://www.freebsd.org/. Acessado em 17 nov.. 2019.
[30] "Vultr.com." Disponível em https://www.vultr.com/. Acessado em 21 nov.. 2019.

O pacote ICMPv6 sai do testador com o objetivo de chegar ao *host,*que na demonstração é um computador com o sistema operacional Debian com com o Wireshark instalado para identificar se os pacotes estão chegando e consequentemente as respostas estão sendo enviadas por parte do equipamento sob teste.

6.2 Testes Realizados pelo Servidor

No quadro 2 estão todos os testes exigidos pela ANATEL de acordo com RFC 2460 e os testes que nosso servidor testador consegue realizar após as modificações.

Quadro 2 - Testes Exigidos pela ANATEL, segundo o CRL 0347*, Brasil,2019.

Testes Exigido pela ANATEL (20 Itens de Teste)	Servidor Testador
RFC 2460-*Group* 1: IPv6 *Header*	-
V6LC.1.1.2	OK
V6LC.1.1.4	OK
V6LC.1.1.5	OK
V6LC.1.1.6	OK
RFC 2460 -*Group* 2: *Extension Headers and Options*	-
V6LC.1.2.1	NOK
V6LC.1.2.2	OK
V6LC.1.2.3	OK
V6LC.1.2.4	NOK
V6LC.1.2.5	OK
V6LC.1.2.8: Exceto subitens G e H	-
RFC 2460 -*Group* 3*: Fragmentation*	OK
V6LC.1.3.1: Exceto partes C, D, E e F	OK
V6LC.1.3.2: Exceto parte D	OK
V6LC.1.3.3	OK

*** Fonte: http://www.inmetro.gov.br/laboratorios/RBLE/docs/CRL0347.pdf**

6.3 Modificações no Script de Teste

Com análise do manual foi constatado que a ferramenta quando compilada e instalada sem as modificações, faz testes com uso do endereço de link-local (fe80::), entretanto, como a inserção do MAC *address* é feita em um dos arquivos de configuração por nome NUT.def, o endereço de destino e baseado no método EUI-

64[31], mas sistemas operacionais como Windows que não utilizam desse método, os testes se tornam difíceis de serem realizados sem modificações. Por fim, constatou-se duas funções, sendo a _SRC e a _DST, que aceitam como parâmetro endereços de escopo global, não precisando usar as funções tnv6 ou nutv6, que tem como objetivo montar o endereço de escopo de *link* local com base nos endereços MACs inseridos nos arquivos de configuração NUT.def e TN.def, com isso, bastava criar dois *defines*, um com endereço global de origem e outro de destino, endereço global que deseja testar. Como todos os arquivos .def tem um include do arquivo *CommonHost*.def, foram criados os dois defines nesse arquivo, quando alterar o endereço destino, basta editar somente o arquivo por nome *CommonHost*.def e o define respectivo ao endereço de destino. É importante ressaltar que no arquivo TN.def contém o endereço MAC do sistema operacional FreeBSD que a ferramenta está instalada e no NUT.def contém o endereço MAC do roteador de borda de onde o servidor está localizado.

Com essas simples modificações foi possível realizar os testes. Nos próximos capítulos serão descritos os objetivos de cada teste e em que arquivos foram realizadas as modificações para funcionar no escopo global.

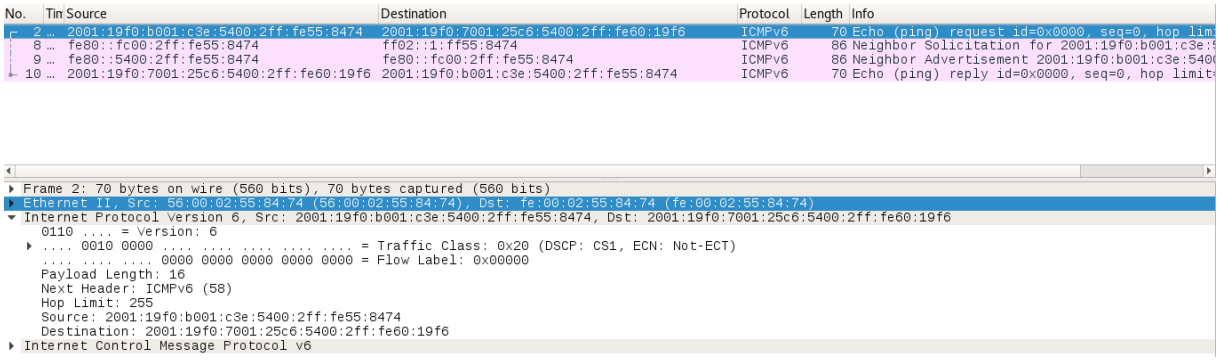
6.3.1 Testes e suas Modificações

A proposta do item *Test* v6LC.1.1.2: *Traffic Class Non-Zero – End Node* é verificar se um nó processa adequadamente o campo Classe de Tráfego dos pacotes recebidos e gera um valor válido nos pacotes transmitidos. As modificações foram realizadas no arquivo IP_TC_NonZeroEN.seq, das linhas 97 a 103 foram comentadas, pois tem relação com o *cleanup* e não precisamos que ela seja realizada, já o arquivo IP_TC_NonZeroEN.def, em todas as chamadas das funções tnv6 e nutv6 foram trocadas pelos *defines* de origem e destino.

Conforme a figura 10, o teste em questão o servidor transmite uma solicitação de *Echo* com um campo de classe de tráfego 32, que é diferente de zero. E o mesmo espera como retorno um *Echo reply.*

[31] "RFC 4291 - IP Version 6 Addressing Architecture - IETF Tools." Disponível em https://tools.ietf.org/html/rfc4291. Acessado em 17 nov.. 2019.

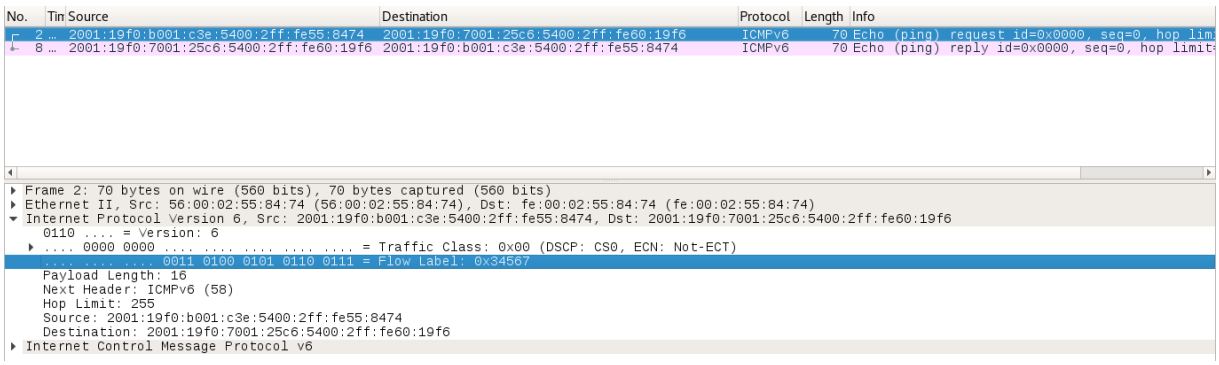
Figura 10 - Evidência do Test v6LC.1.1.2 no servidor.



Fonte: Autoria própria.

Já o item de teste *Test* v6LC.1.1.4: *Flow Label Non-Zero* tem como objetivo verificar se um nó processa adequadamente o campo *Flow Label* dos pacotes recebidos e gera um valor válido nos pacotes transmitidos, ou seja, conforme a figura 11, o servidor envia uma solicitação de Echo com um rótulo de fluxo de 0x34567 para a nó sob teste. As modificações foram as mesmas do teste anterior, porém, os arquivos alterados foram os IP_TC_NonZeroEN.seq e o IP_TC_NonZeroEN.def, as modificações têm relação com o *cleanup* e com as chamadas das funções tnv6 e nutv6.

Figura 11 - Evidência do Test v6LC.1.1.4 no servidor.

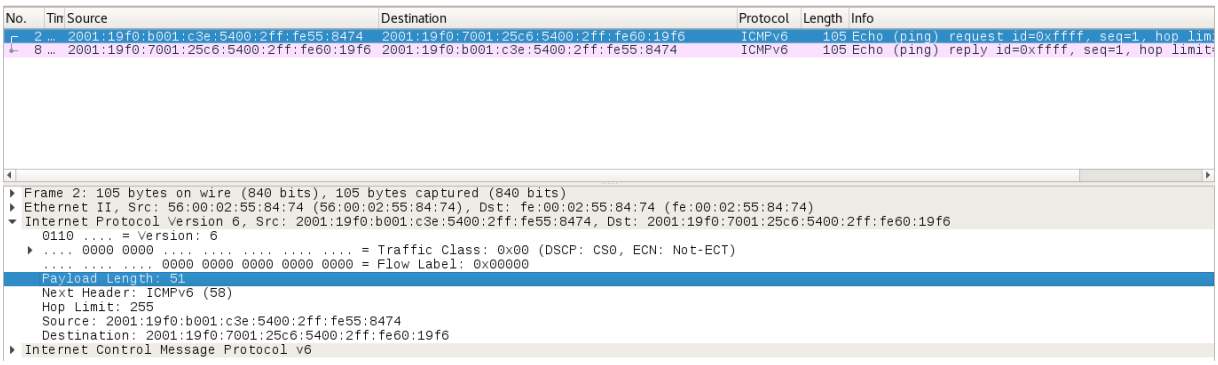


Fonte: Autoria própria.

O item *Test* v6LC.1.1.5: *Payload Length* faz testes com o objetivo de verificar se um nó processa adequadamente o campo Comprimento da carga útil dos pacotes recebidos. Os arquivos alterados foram os IP_FL_NonZeroEN.seq, IP_TC_NonZeroEN.def, IP_TC_NonZeroEN.seq e IP_FL_NonZeroEN.def e suas alterações foram no *cleanup* e nas chamadas das funções tnv6 e nutv6.

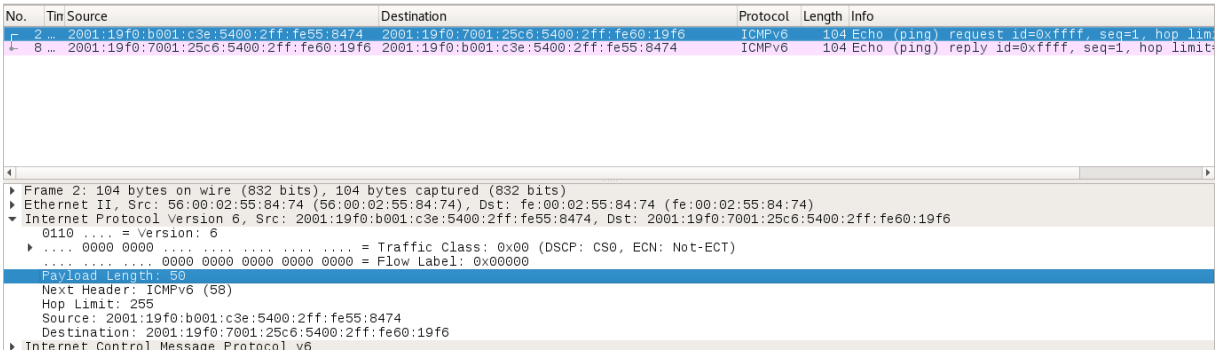
O teste está dividido em duas partes, onde no primeiro momento o servidor envia uma solicitação de *Echo* com um cabeçalho IPv6 com um comprimento de carga útil de 0x33 (51), evidência do teste está na figura 12 e no segundo momento uma solicitação de *Echo* que possui um cabeçalho IPv6 com um comprimento de carga útil de 0x32 (50), evidência na figura 13.

Figura 12 - Evidência do Test v6LC.1.1.5 Part A no servidor.



Fonte: Autoria própria.

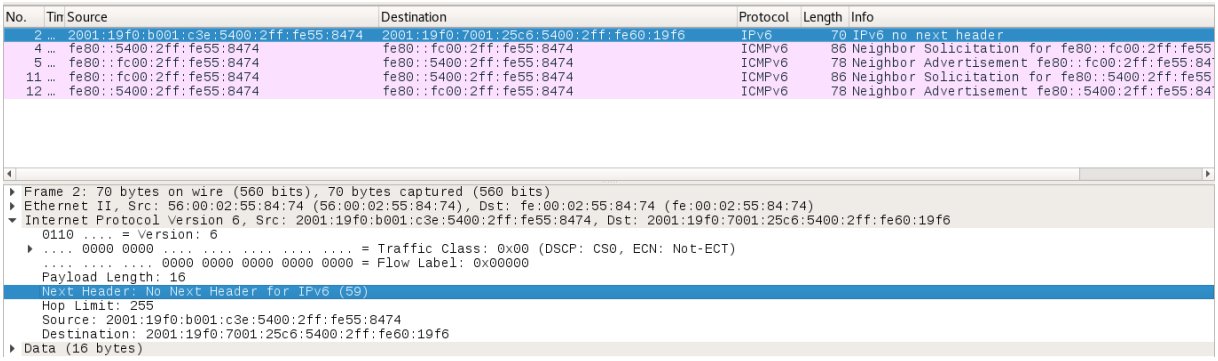
Figura 13 - Evidência do Test v6LC.1.1.5 Part B no servidor.



Fonte: Autoria própria.

Os testes deste item *Test* v6LC.1.1.6: *No Next Header after* IPv6 *Header* tem a características de verificar o comportamento adequado de um nó quando encontrar um valor de próximo cabeçalho de 59 (sem próximo cabeçalho), nesse momento o *server tester* envia *Echo Request* que contém um cabeçalho IPv6 com um Próximo cabeçalho de 59, tais campos podem ser visualizados na figura 14. As modificações foram nos arquivos IP_NH_NoneEN.def e IP_NH_NoneEN.seq e tem relação com o *cleanup* e chamadas das funções tnv6 e nutv6.

Figura 14 - Evidência do Test v6LC.1.1.6 no servidor.

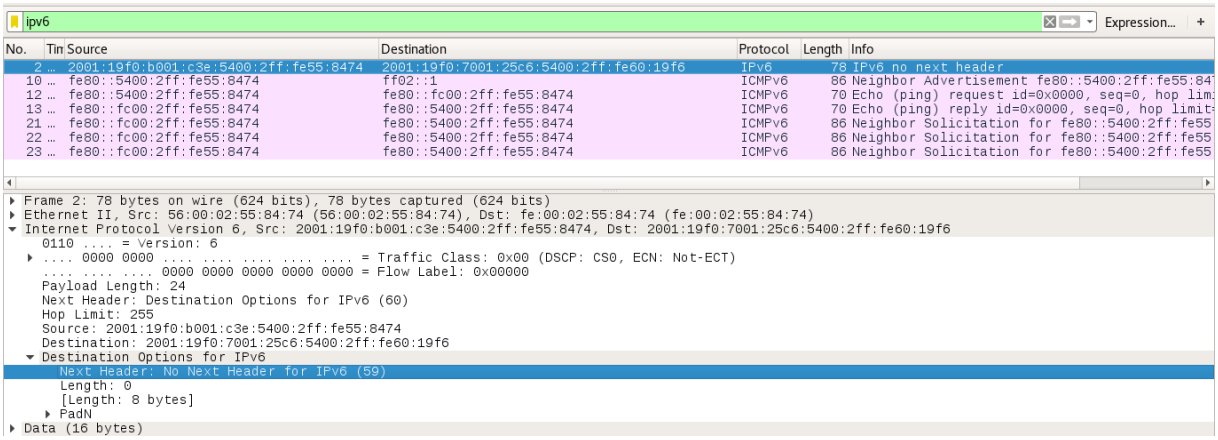


Fonte: Autoria própria.

O teste *Test* v6LC.1.2.2: *No Next Header after Extension Header* verifica o comportamento adequado de um nó quando encontrar um valor de próximo cabeçalho de 59 (sem próximo cabeçalho) no campo *Destination Options header*. Os arquivos modificados foram os EX_NH_NoneEN.def e EX_NH_NoneEN.seq e as alterações são idênticas as outras já listadas.

Conforme a figura 15, o servidor transmite um pacote de solicitação de *Echo* para a nó sob teste, que contém um cabeçalho de Opções de destino com um próximo cabeçalho de 59. Após o cabeçalho de Opções de destino, há um cabeçalho de solicitação de *Echo* ICMPv6.

Figura 15 - Evidência do Test v6LC.1.2.2 no servidor.



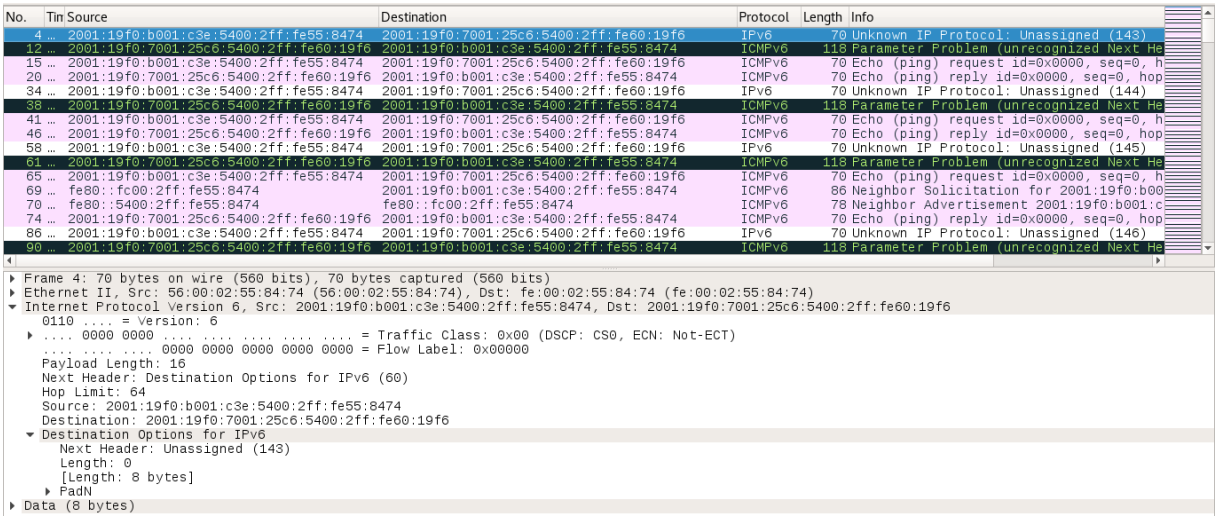
Fonte: Autoria própria.

O tópico *Test* v6LC.1.2.3: *Unrecognized Next Header in Extension Header – End Node* verifica se um nó descarta um pacote com um próximo cabeçalho não reconhecido ou inesperado em um cabeçalho de extensão e transmite uma

mensagem ICMPv6 *Parameter Problem* para a origem do pacote. Os arquivos alterados foram os EX_NH_UnrecognizedEN.def, EX_NH_UnexpectedEN.def, EX_NH_UnexpectedEN.seq e EX_NH_UnrecognizedEN.seq e são modificações no *cleanup* e nas funções de chamada para geração dos endereços de link local com base nos arquivos de configurações.

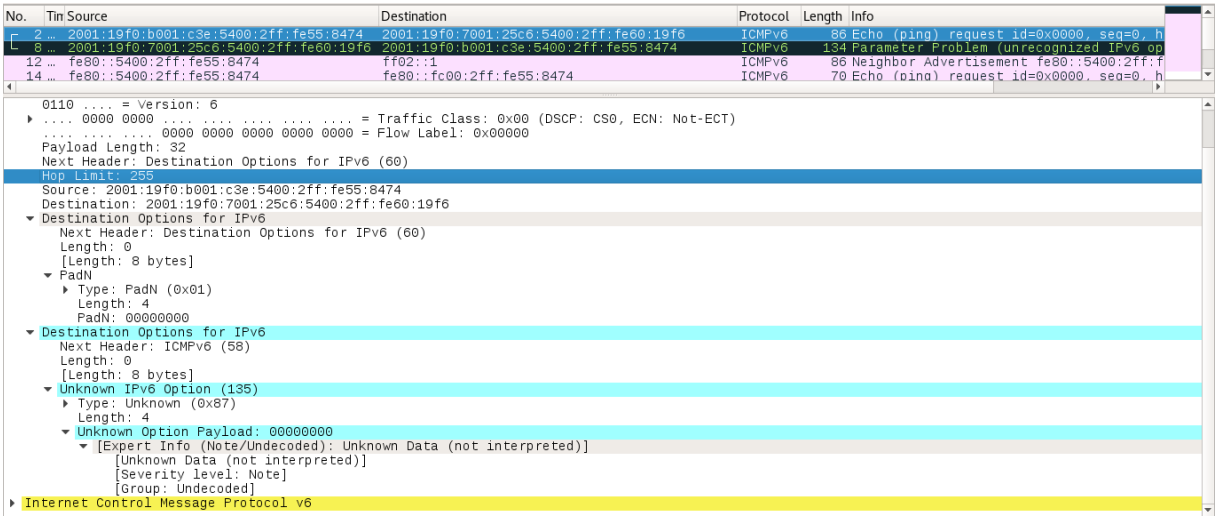
O teste em questão também está dividido em dois procedimentos, um o primeiro momento o servidor transmitindo um pacote de solicitação de *Echo*, que possui um cabeçalho Opções de destino com um campo Próximo cabeçalho de 143, o teste repete esse passo inúmeras vezes alterando somente o valor no Próximo cabeçalho (teste na figura 16). Já no segundo procedimento evidenciado na figura 17, o servidor transmite a solicitação de *Echo*, que possui um cabeçalho de Opções de destino com um campo Próximo cabeçalho de 60. O cabeçalho de extensão real a seguir é um cabeçalho de fragmento. O deslocamento do fragmento é 0x10E0 (para que os primeiros 8 bits desse campo de 13 bits sejam 135). O segundo campo reservado é 0x2 e o bit mais está limpo (Se processado como um cabeçalho Opções de destino, isso seria processado como Comprimento dos dados da opção igual a 4).

Figura 16 - Evidência do Test v6LC.1.2.3 Part A no servidor.



Fonte: Autoria própria.

Figura 17 - Evidência do Test v6LC.1.2.3 Part B no servidor.

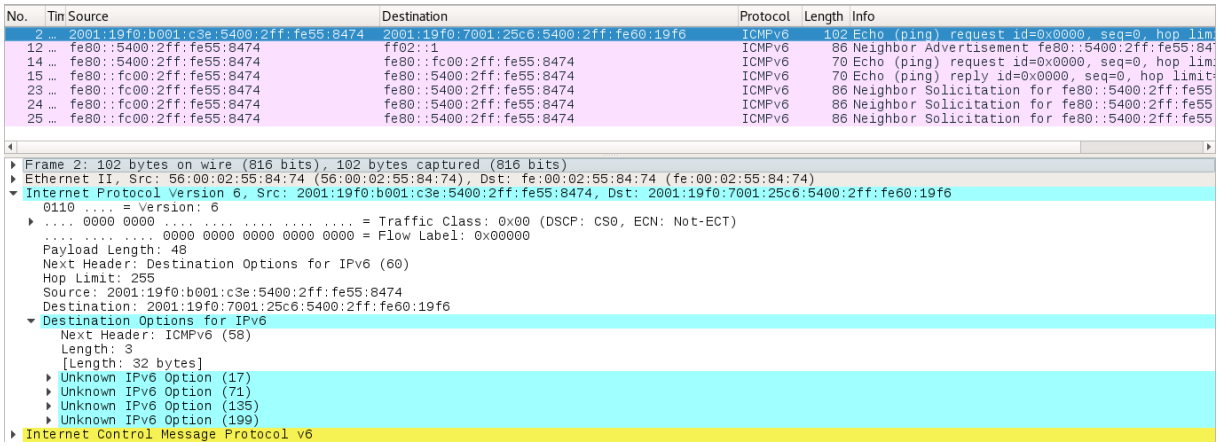


Fonte: Autoria própria.

O item *Test* v6LC.1.2.5: *Option Processing Order* verifica se um nó processa corretamente as opções em um único cabeçalho na ordem da ocorrência. Modificações nos arquivos EX_Opt*ProcessingOrder*_01.def, EX_Opt*ProcessingOrder*_01.seq, EX_Opt*ProcessingOrder*_10.def, EX_Opt*ProcessingOrder*_10.seq, EX_Opt*ProcessingOrder*_11.def e EX_Opt*ProcessingOrder*_11.seq e são modificações no *cleanup* e nas funções de chamada para geração dos endereços de *link* local com base nos arquivos tn.def e nut.def.

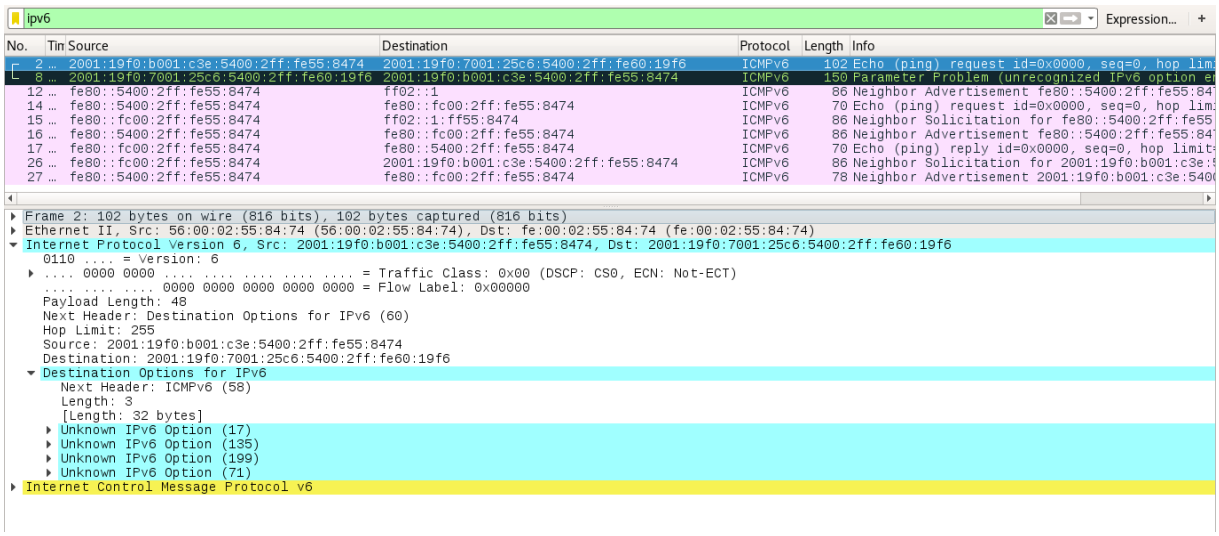
Este item de ensaio possui três procedimentos, no primeiro o servidor transmite uma solicitação de *Echo* que possui um cabeçalho Opções de destino com quatro opções desconhecidas, os tipos de opção são 7, 71, 135 e 1. Já no segundo procedimento os tipos de opção ainda são desconhecidos, entretanto, os valores são outros, no caso os tipos de opção são 7, 135, 199 e 71. E no último ensaio o pacote de Solicitação de *Echo* com um cabeçalho Opções de destino com quatro opções desconhecidas, os tipos de opção são 7, 199, 71 e 135. As figuras 18, 19 e 20 são as evidências dos testes realizados pelo servidor na Vultr.

Figura 18 - Evidência do Test v6LC.1.2.5 Part A no servidor.



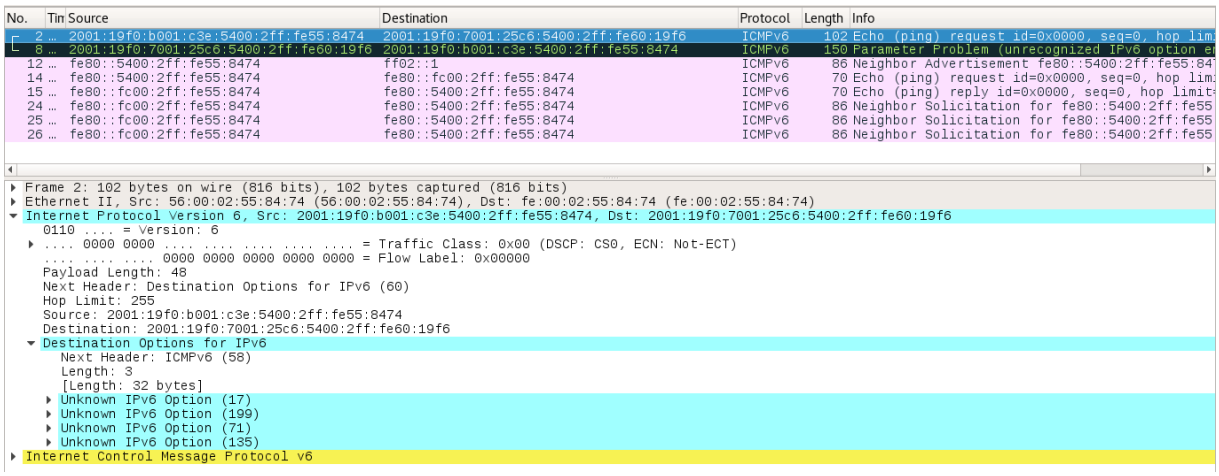
Fonte: Autoria própria.

Figura 19 - Evidência do Test v6LC.1.2.5 Part B no servidor.



Fonte: Autoria própria.

Figura 20 - Evidência do Test v6LC.1.2.5 Part C no servidor.

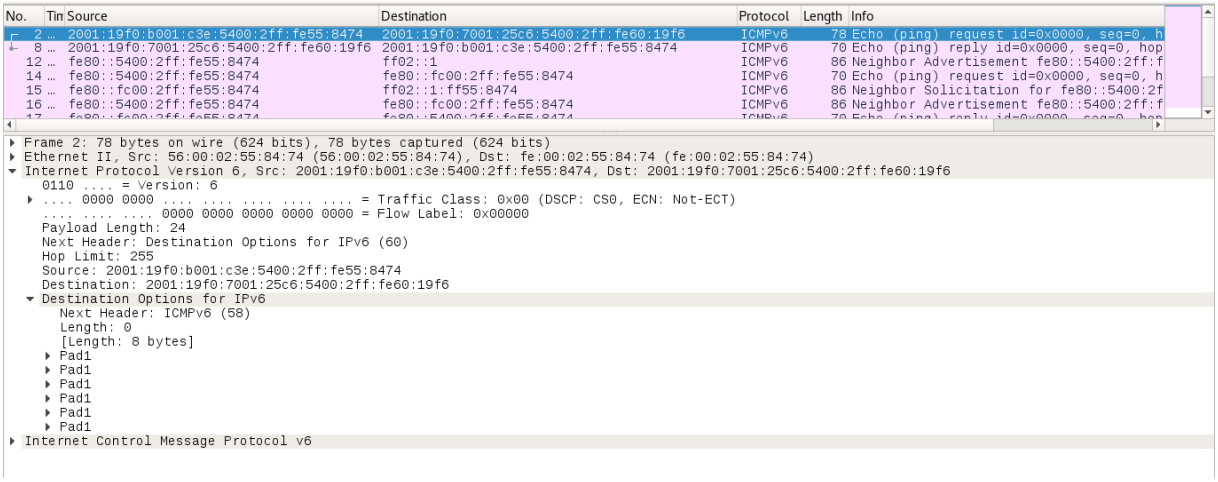


Fonte: Autoria própria.

O item *Test* v6LC.1.2.8: *Option Processing, Destination Options Header* verifica se um nó processa adequadamente as opções conhecidas e desconhecidas e age de acordo com a ordem mais alta de dois bits da opção. Os arquivos modificados tem relação os testes 30 (figura 21), 31 (figura 22), 32 (figura 23), 33 (figura 24), 34 (figura 25) e 35 (figura 26), as modificações foram somente no *cleanup* e na inserção dos *defines* no lugar das funções tnv6 e nutv6.

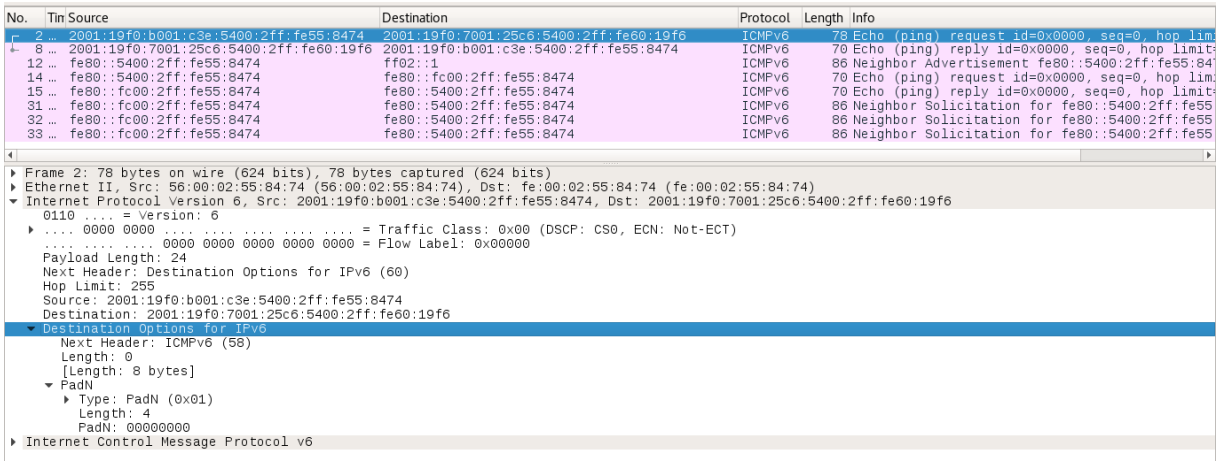
Os procedimentos de testes para este item de ensaio são transmissão de um solicitação de *Echo* com um cabeçalho Opções de destino com seis opções Pad1, uma solicitação de *Echo* com um cabeçalho Opções de destino com uma opção PadN com 4 bytes de dados da opção, uma solicitação de *Echo* com um cabeçalho Opções de destino com um tipo de opção desconhecido 7, uma solicitação de *Echo* com um cabeçalho Opções de destino com um tipo de opção desconhecido 71, uma solicitação de *Echo* com um cabeçalho Opções de destino com um tipo de opção desconhecido 135 e por último a transmissão de uma solicitação de *Echo* com um cabeçalho Opções de destino com um tipo de opção desconhecido 199.

Figura 21 - Evidência do Test v6LC.1.2.8 Part A no servidor.



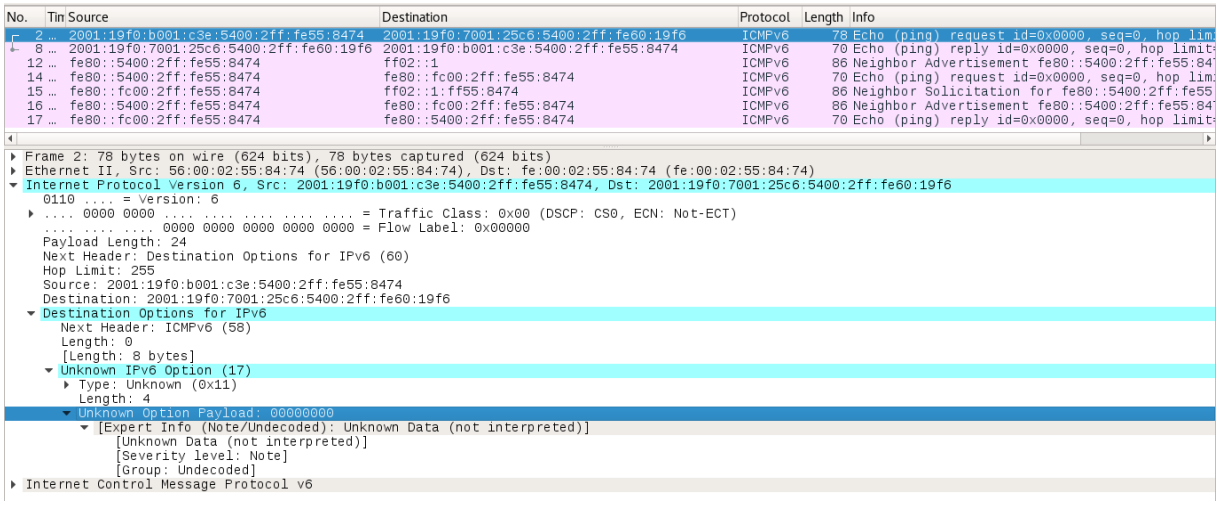
Fonte: Autoria própria.

Figura 22 - Evidência do Test v6LC.1.2.8 Part B no servidor.



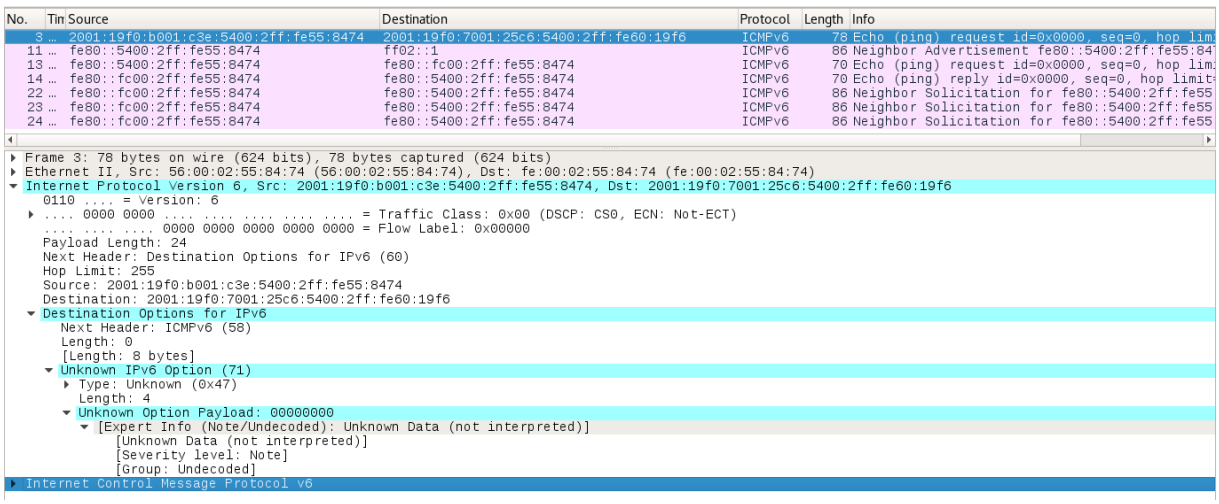
Fonte: Autoria própria.

Figura 23 - Evidência do Test v6LC.1.2.8 Part C no servidor.



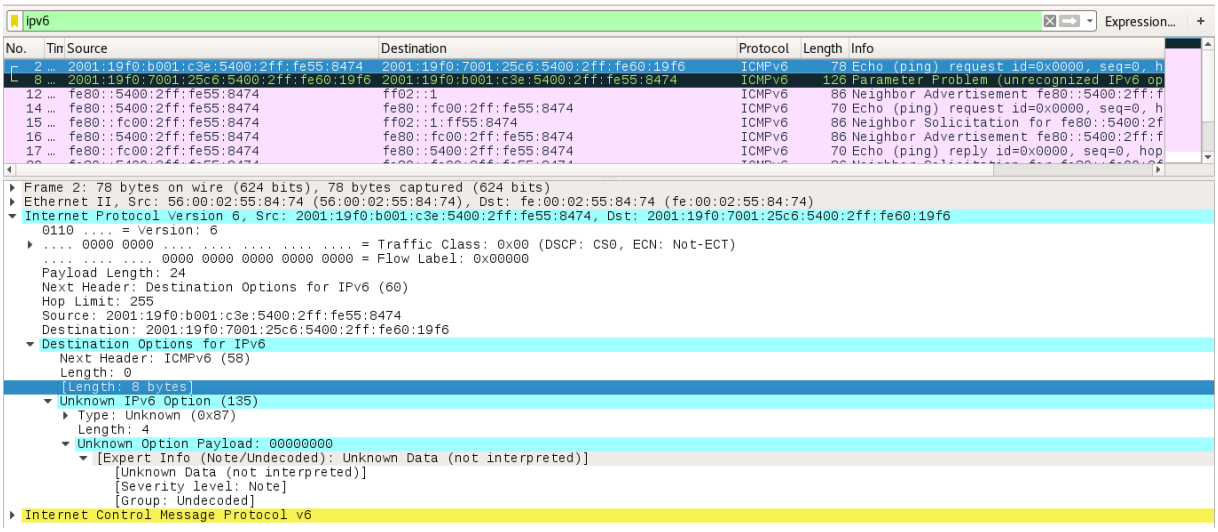
Fonte: Autoria própria.

Figura 24 - Evidência do Test v6LC.1.2.8 Part D no servidor.



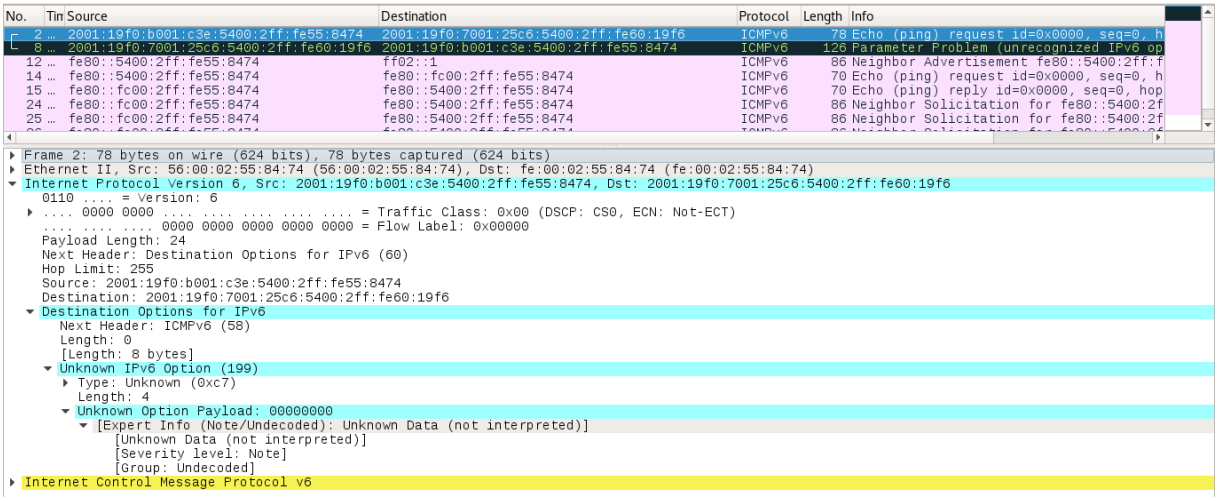
Fonte: Autoria própria.

Figura 25 - Evidência do Test v6LC.1.2.8 Part E no servidor.



Fonte: Autoria própria.

Figura 26 - Evidência do Test v6LC.1.2.8 Part F no servidor.

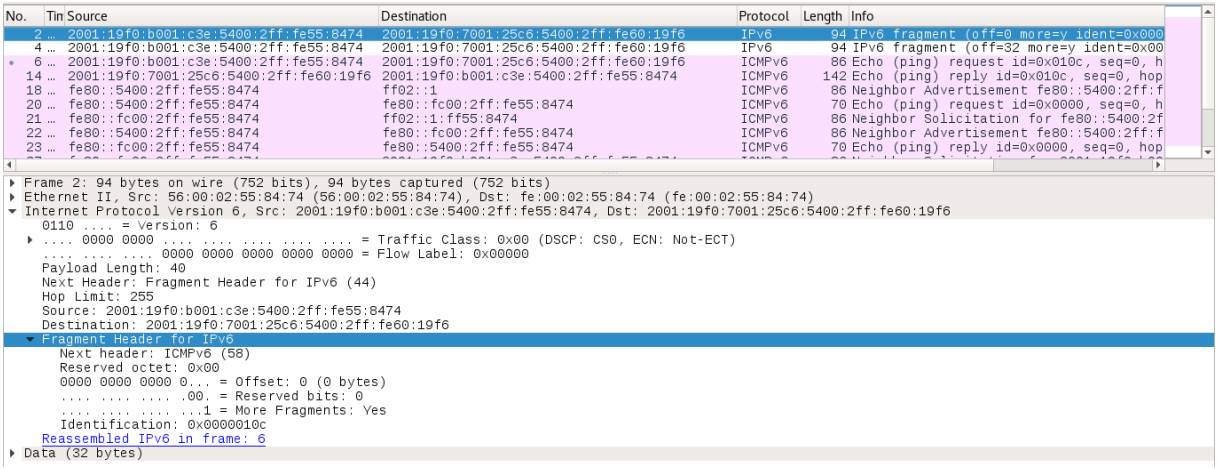


Fonte: Autoria própria.

O elemento *Test* v6LC.1.3.1: *Fragment Reassembly* verifica se um nó remonta corretamente pacotes fragmentados e distingue entre fragmentos de pacote usando o Endereço de origem, Endereço de destino e ID do fragmento. Foram alterados os arquivos F_*Reassembly_Valid*.def, F_*Reassembly_Valid*.seq, F_*Reassembly_reverse*.def e F_*Reassembly_reverse*.seq onde foram comentadas as linhas do *cleanup* que se encontravam nos arquivos com extensão .seq. No teste do item v6LC.1.3.1 os seguintes procedimentos são realizados. O servidor transmite três

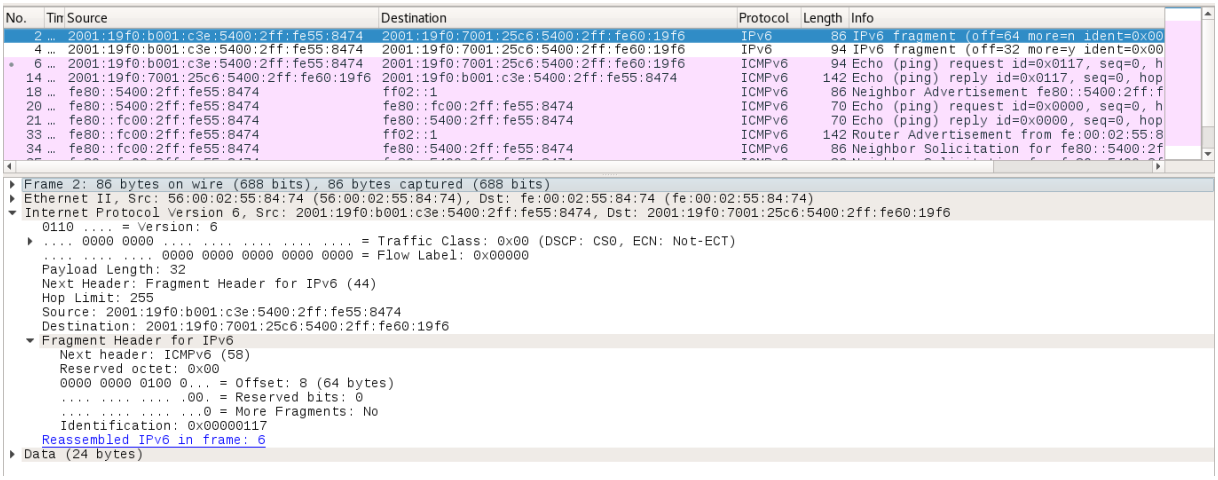
fragmentos em ordem. Todos os fragmentos têm o mesmo endereço de origem, endereço de destino e ID do fragmento, conforme a figura 27. Já no segundo procedimento do item, o servidor transmite três fragmentos, em ordem decrescente, conforme a figura 28. Todos os fragmentos têm o mesmo endereço de origem, endereço de destino e ID do fragmento.

Figura 27 - Evidência do Test v6LC.1.3.1 Part A no servidor.



Fonte: Autoria própria.

Figura 28 - Evidência do Test v6LC.1.3.1 Part B no servidor.



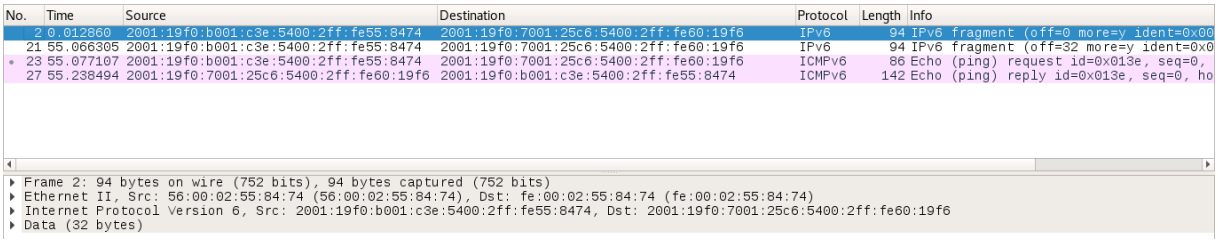
Fonte: Autoria própria.

A proposta do item *Test* v6LC.1.3.2: *Reassembly Time Exceeded* é verificar se um nó executa as ações apropriadas quando o tempo de remontagem for excedido para um pacote. As modificações foram no arquivos .def e .seq dos testes 48, 49 e 50 e as modificações são mesma do item v6LC.1.3.1, ou seja, comentar as linhas do

cleanup e do envio do RA e adicionar os *defines* nas funções _SRC e _DST.

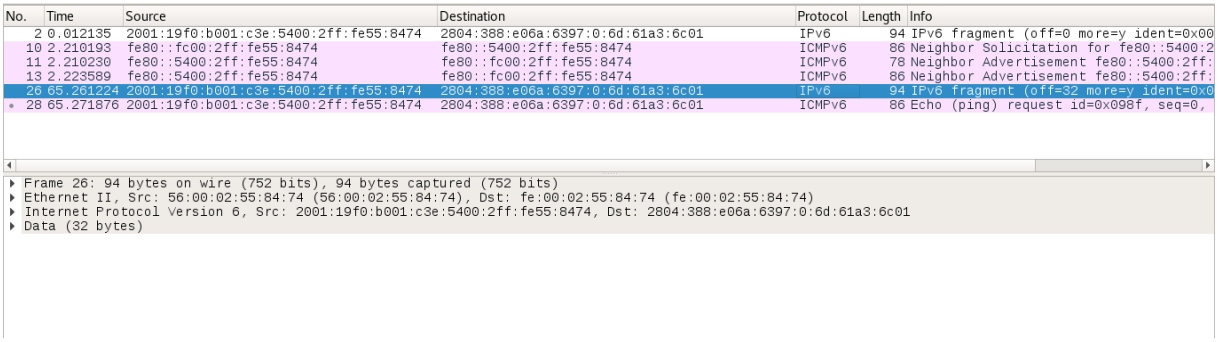
Os ensaios realizados neste item de ensaio são. Na parte A (figura 29) o servidor transmite três fragmentos em ordem. Há um atraso de 55 segundos entre a transmissão do primeiro fragmento e os fragmentos dois e três. Já o teste B (figura 30) o servidor transmite três fragmentos em ordem. Entretanto, o atraso agora é de 65 segundos entre a transmissão do primeiro fragmento e os fragmentos dois e três. E nos dois últimos testes (figura 31 e figura 32) um único fragmento é enviado com objetivo de que o nó sob teste retorne com uma mensagem ICMPv6 *Time Exceeded.*

Figura 29 - Evidência do Test v6LC.1.3.2 Part A no servidor.



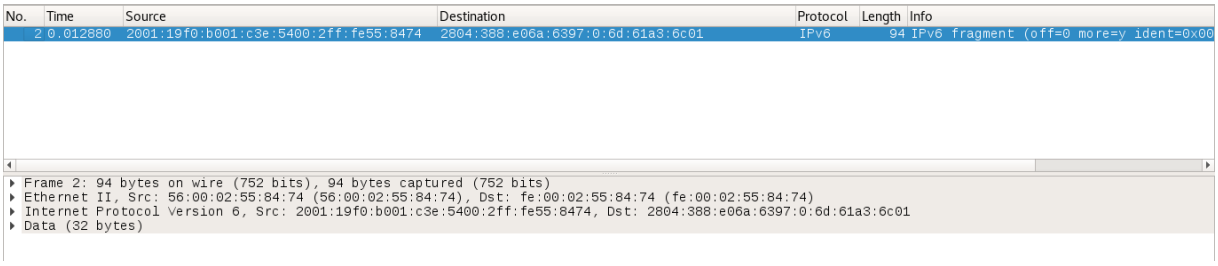
Fonte: Autoria própria.

Figura 30 - Evidência do Test v6LC.1.3.2 Part B no servidor.



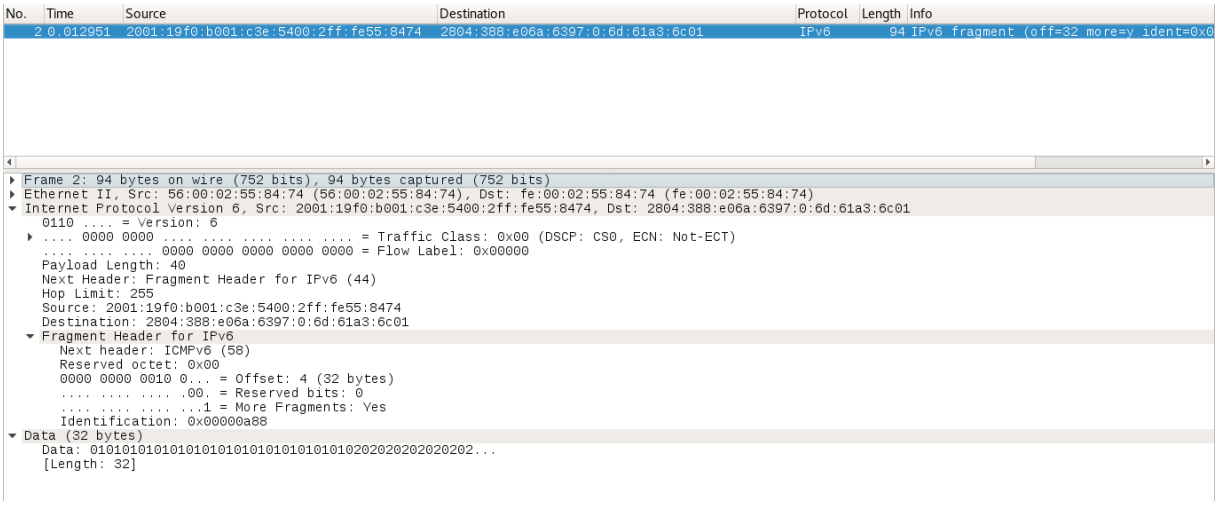
Fonte: Autoria própria.

Figura 31 - Evidência do Test v6LC.1.3.2 Part C no servidor.



Fonte: Autoria própria.

Figura 32 - Evidência do Test v6LC.1.3.2 Part D no servidor.

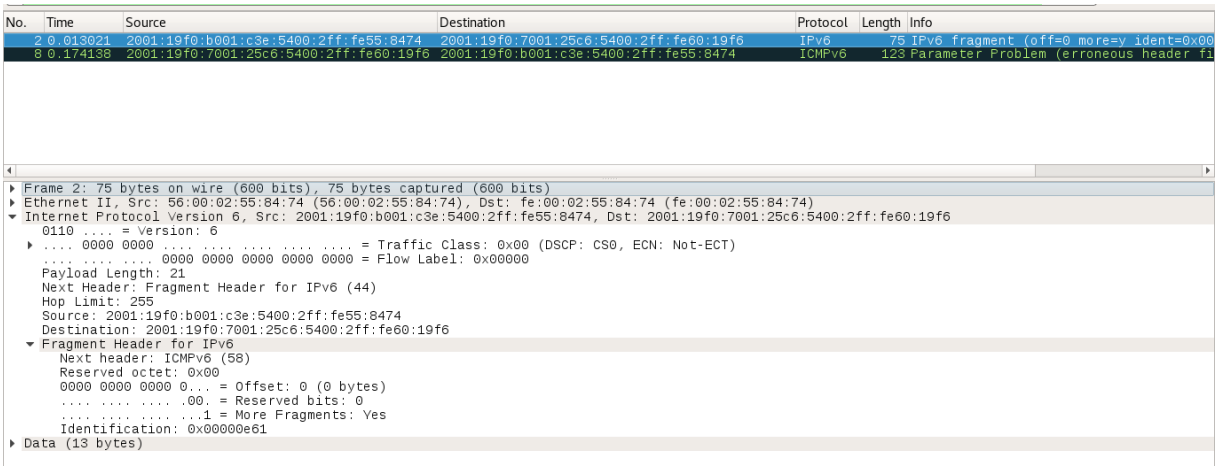


Fonte: Autoria própria.

O procedimento *Test* v6LC.1.3.3: Fragment *Header* M-Bit *Set*, *Payload Length Invalid* verifica se um nó executa as ações apropriadas ao receber um fragmento com o conjunto de bits M (mais fragmentos), mas que possui um comprimento de carga útil que não é um múltiplo de 8 bytes. Os arquivos alterados foram os F_*PLinvalid*.def e F_*PLinvalid*.seq, foram comentadas as linhas do *cleanup* que se encontravam nos arquivos com extensão .seq além dos procedimentos que o testador faz ao fornecer ao nó sob teste um roteador padrão e um endereço global. Os testes no escopo global não à necessidade enviar RA para o nó sob teste com prefixo de uma rede. E a outra modificação foram no arquivo .def adicionando os defines nas funções _SRC e _DST.

O procedimento de teste é descrito onde o servidor transmite uma solicitação de *Echo* que possui um cabeçalho de fragmento com o conjunto de bits M, tal pacote está evidenciado na figura 33. O comprimento da carga útil é 21, que não é um múltiplo de 8 octetos.

Figura 33 - Evidência do Test v6LC.1.3.3 no servidor.



Fonte: Autoria própria.

7 CONSIDERAÇÕES FINAIS

O trabalho em questão apresentou os principais protocolos, pacotes e mensagens que fazem parte do novo protocolo da internet. A princípio o objetivo é esse, pois com isso possibilita que o leitor dê continuidade na leitura do trabalho e identifique e compreenda o seu objetivo, independente do conhecimento que o mesmo tem sobre a tecnologia.

Por fim, com a identificação das particularidades da tecnologia, os requisitos técnicos foram apresentados, seus procedimentos de testes e a ferramenta de testes com a devidas modificações, já com os resultados apresentados neste trabalho, pudemos observar que com o IPv6, testes de interoperabilidade ou da *stack* IPv6 em dispositivos eletrônicos finais podem ser realizados com os produtos já em uso, ou seja, conectado na internet e possuindo um endereço de escopo global. E para isso foi montado um servidor e algumas modificações no código fonte da ferramenta do projeto TAHI foi realizado, como informado anteriormente.

É importante destacar as dificuldades encontradas na busca por matérias sobre os assuntos, foram encontrados vários livros falando sobre as melhorias e particularidades do IPV6, entretanto, informações de quais itens de testes são aplicados, quais ferramentas são utilizadas pelos laboratórios e quais são as topologias usadas para estes testes não foram encontradas tão facilmente e algumas delas nem existem na internet.

Podemos concluir que testes do *Core* IPv6 podem ser realizados no âmbito global e já em produção. Entretanto, modificações com objetivo de melhoria já existe na RFC 8200 e para isso, a ANATEL deve alterar alguns dos seus requisitos técnicos (documentos normativos). Consequentemente as ferramentas de testes irão surgir com o novo *Core,* e basta o desenvolver aplicar as mesmas modificações apresentadas neste trabalho para as novas versões das ferramentas.

REFERÊNCIAS BIBLIOGRÁFICAS

3GPP. **Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Packet Core (EPC); User Equipment (UE) conformance specification; Part 1: Protocol conformance specification.** Disponível em: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2472> Acesso em: 22 nov. 2019

BRITO, S. H. B. **IPv6. O Novo Protocolo da Internet**. 1. ed. São Paulo: Novatec Editora, 2013. 213p.

CARNIER *et al.* **Laboratório de IPv6 Aprenda na Prática usando um Emulador de Redes**. São Paulo: Novatec Editora, 2015. 417p. Disponível em: <http://ipv6.br/media/arquivo/ipv6/file/64/livro-lab-ipv6-nicbr.pdf> Acesso em: 16 nov. 2019

CENAPAD UNICAMP. **10. Comandos**. Disponível em: <https://www.cenapad.unicamp.br/servicos/treinamentos/tutorial_unix/unix_tutor-10.html> Acesso em: 22 nov. 2019

ETSI. **Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT): IPv6 Core Protocol; Requirements Catalogue.** Disponível em: <https://www.etsi.org/deliver/etsi_ts/102500_102599/102514/02.01.01_60/ts_102514v020101p.pdf> Acesso em: 22 nov. 2019

FREEBSD. **About FreeBSD Ports**. Disponível em: <https://www.freebsd.org/ports/> Acesso em: 22 nov. 2019

FREEBSD. **FreeBSD 12.0-RELEASE Announcement**. Disponível em: <https://www.freebsd.org/releases/12.0R/announce.html> Acesso em: 22 nov. 2019
FREEBSD. **The FreeBSD Project.** Disponível em: <https://www.freebsd.org/> Acesso em: 22 nov. 2019

GUPTA *et al.* **ICMPv6 (ICMP for IPv6)**. Disponível em: <https://tools.ietf.org/html/rfc4443> Acesso em: 15 nov. 2019

HINDEN *et al.* **Internet Protocol Version 6 (IPv6) Addressing Architecture**. Disponível em: <https://tools.ietf.org/html/rfc3513> Acesso em: 22 nov. 2019

HINDEN *et al.* **IP Version 6 Addressing Architecture**. Disponível em: <https://tools.ietf.org/html/rfc4291> Acesso em: 22 nov. 2019

HINDEN *et al.* **IPv6 Specification.** Disponível em: <https://tools.ietf.org/html/rfc8200> Acesso em: 17 nov. 2019

HINDEN *et al.* **IPv6 Specification**. Disponível em: <https://tools.ietf.org/html/rfc2460> Acesso em: 17 nov. 2019

IBM**. Comparação entre IPv4 e IPv6**. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt/ssw_ibm_i_73/rzai2/rzai2compipv

4ipv6.htm> Acesso em: 17 nov. 2019

IETF. **IETF Main Page**. Disponível em: <https://ietf.org/> Acesso em: 22 nov. 2019

INMETRO. **Laboratórios.** Disponível em: <http://www.inmetro.gov.br/laboratorios/rble/> Acesso em: 22 nov. 2019

IPV6 READY**. Estrutura Comportamental da Ferramenta**. Disponível em: <https://www.ipv6ready.org.cn/home/views/default/resource/logo/release/v6eval-e.pdf> Acesso em: 17 nov. 2019

IPV6 READY. **IPv6 Ready Logo Phase-2**. Disponível em: <https://www.ipv6ready.org.cn/home/views/default/resource/logo/phase2-core/index.htm> Acesso em: 22 nov. 2019

IPV6 READY. **ipv6-core-protocols.** Disponível em: <https://www.ipv6ready.org/?page=documents&tag=ipv6-core-protocols> Acesso em: 22 nov. 2019

IPV6.BR. **Endereçamento**. Disponível em: <http://ipv6.br/post/enderecamento/> Acesso em: 17 nov. 2019

MRUGALSKI et al. **Dynamic Host Configuration Protocol for IPv6 (DHCPv6)**. Disponível em: <https://tools.ietf.org/html/rfc8415> Acesso em: 22 nov. 2019

MULLER. **YAML**. Disponível em: <https://metacpan.org/pod/YAML> Acesso em: 22 nov. 2019

PARTRIDGE *et al*. **Computing the Internet Checksum**. Disponível em: <https://tools.ietf.org/html/rfc1071> Acesso em: 22 nov. 2019

PERLBREW. **Perlbrew**. Disponível em: <https://perlbrew.pl/> Acesso em: 22 nov. 2019

TAHI PROJECT **Manual do V6eval**. Rev.2.3. ed. Japão: TAHI Project, 2018. 93p. Disponível em: <https://www.ipv6ready.org.cn/home/views/default/resource/logo/release/v6eval-e.pdf> Acesso em: 17 nov. 2019

TELECO. **ERB: O que é?.** Disponível em: <https://www.teleco.com.br/tutoriais/tutorialerb/pagina_1.asp> Acesso em: 22 nov. 2019

UNIVERSITY OF NEW HAMPSHIRE- INTEROPERABILITY LAB. **Test Specification Core Protocols**. Revision 4.0.8. ed. Japão: IPv6 Forum, 2018. 295p. Disponível em: <https://www.ipv6ready.org/docs/IPv6_Ready_Test_Specification_Core_Protocols_v4.0.8.pdf> Acesso em: 14 nov. 2019

VULTR. **The Infrastructure Cloud™.** Disponível em: <https://www.vultr.com/>

Acesso em: 22 nov. 2019