UNISINOS	ATIVIDADE ACADÊMICA: Redes de Computadores: Internetworking, Roteamento e Transmissão TRABALHO: Atividade 3 ALUNOS: Rafael Hansen Klauck

Para realizar essa atividade, antes de criar as máquinas virtuais na AWS, eu verifiquei se seria possível utilizar o IPV6 através da minha máquina. Para isso rodei o comando *curl -6 ifconfig.co.* Esse comando faz uma requisição HTTP para o serviço *ifconfig.co* e ele retorna o endereço IP. Como a flag *-6* está sendo utilizada, isso irá forçar que seja utilizado o IPV6. Para minha surpresa, obtive o seguinte resultado:

```
❯ ~ curl -6 ifconfig.co

2804:2a4c:140f:5000:c40f:6937:2466:3cd1
```

Com isso, é possível visualizar qual meu endereço IPV6, fazendo com que não fosse necessário a criação das máquinas virtuais na AWS.

Após isso, enviei cinco pacotes ICMP com pingv6 para o google. Para isso, utilizei o comando *ping6 -c 5 google.com*. Esse comando envia cinco pacotes ICMPv6 e mostra as respostas. Porém, antes de executar o *ping6*, eu rodei em outro terminal o *tcpdump* para capturar os pacotes. Para isso rodei o comando *sudo tcpdump -i en0 -w pingv6_capture.pcap ip6 and icmp6.* Esse comando irá rodar o *tcpdump* para capturar pacotes na interface en0, salvar no arquivo *pingv6_capture.pcap* e irá filtrar por pacotes IPv6 e pacotes ICMPv6.

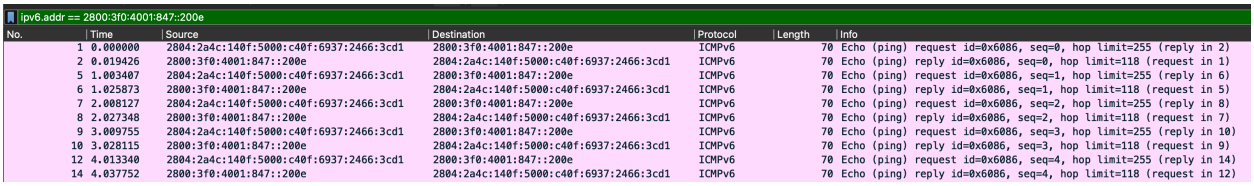
```
❯ ~ ping6 -c 5 google.com

PING6(56=40+8+8 bytes) 2804:2a4c:140f:5000:c40f:6937:2466:3cd1 --> 2800:3f0:4001:847::200e
16 bytes from 2800:3f0:4001:847::200e, icmp_seq=0 hlim=118 time=19.579 ms
16 bytes from 2800:3f0:4001:847::200e, icmp_seq=1 hlim=118 time=22.669 ms
16 bytes from 2800:3f0:4001:847::200e, icmp_seq=2 hlim=118 time=19.453 ms
16 bytes from 2800:3f0:4001:847::200e, icmp_seq=3 hlim=118 time=18.593 ms
16 bytes from 2800:3f0:4001:847::200e, icmp_seq=4 hlim=118 time=24.551 ms
```

```
❯ ~ sudo tcpdump -i en0 -w pingv6_capture.pcap ip6 and icmp6

tcpdump: listening on en0, link-type EN10MB (Ethernet), snapshot length 524288 bytes
^C24 packets captured
319 packets received by filter
0 packets dropped by kernel
```

Após isso, abri o arquivo gerado no Wireshark para analisar os pacotes. Como estava utilizando a interface do meu computador, havia muitos pacotes no arquivo, então para conseguir apenas ver os do ping realizado utilizei o filtro *ipv6.addr == 2800:3f0:4001:847::200e*. Com esse filtro, foi possível de visualizar apenas os pacotes que havia o endereço do Google no pacote ipv6.



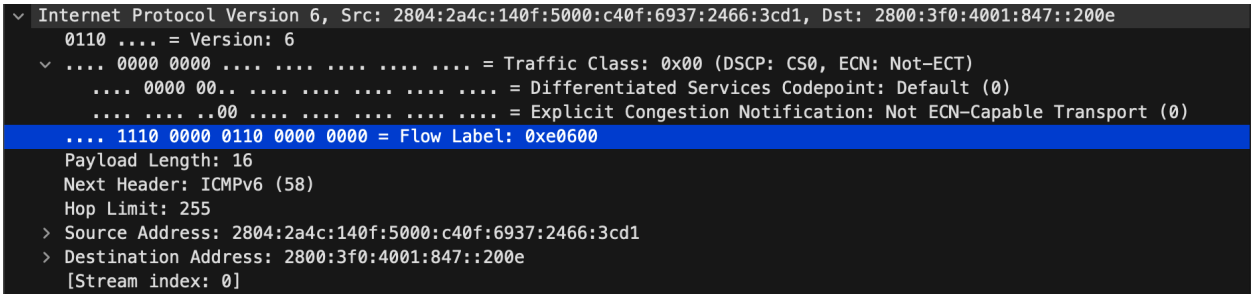
ipv6.addr == 2800:3f0:4001:847::200e

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	2800:3f0:4001:847::200e	ICMPv6	70	Echo (ping) request id=0x6086, seq=0, hop limit=255 (reply in 2)
2	0.019426	2800:3f0:4001:847::200e	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	ICMPv6	70	Echo (ping) reply id=0x6086, seq=0, hop limit=118 (request in 1)
5	1.003407	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	2800:3f0:4001:847::200e	ICMPv6	70	Echo (ping) request id=0x6086, seq=1, hop limit=255 (reply in 6)
6	1.025873	2800:3f0:4001:847::200e	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	ICMPv6	70	Echo (ping) reply id=0x6086, seq=1, hop limit=118 (request in 5)
7	2.008127	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	2800:3f0:4001:847::200e	ICMPv6	70	Echo (ping) request id=0x6086, seq=2, hop limit=255 (reply in 8)
8	2.027348	2800:3f0:4001:847::200e	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	ICMPv6	70	Echo (ping) reply id=0x6086, seq=2, hop limit=118 (request in 7)
9	3.009755	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	2800:3f0:4001:847::200e	ICMPv6	70	Echo (ping) request id=0x6086, seq=3, hop limit=255 (reply in 10)
10	3.028115	2800:3f0:4001:847::200e	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	ICMPv6	70	Echo (ping) reply id=0x6086, seq=3, hop limit=118 (request in 9)
12	4.013340	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	2800:3f0:4001:847::200e	ICMPv6	70	Echo (ping) request id=0x6086, seq=4, hop limit=255 (reply in 14)
14	4.037752	2800:3f0:4001:847::200e	2804:2a4c:140f:5000:c40f:6937:2466:3cd1	ICMPv6	70	Echo (ping) reply id=0x6086, seq=4, hop limit=118 (request in 12)

Na imagem, podemos ver um total de 10 pacotes, isso porque o ping tem a ida e a volta. Na coluna *Source* e *Destination*, é possível de visualizar que fica alternando entre meu IP e o IP do Google, evidenciando que há a ida e a volta no ping.

Quais forma os valores do campo “Flow Label” para o replay e request?

O Flow Label foi o mesmo para o replay e o request, sendo ele 0xe0600.

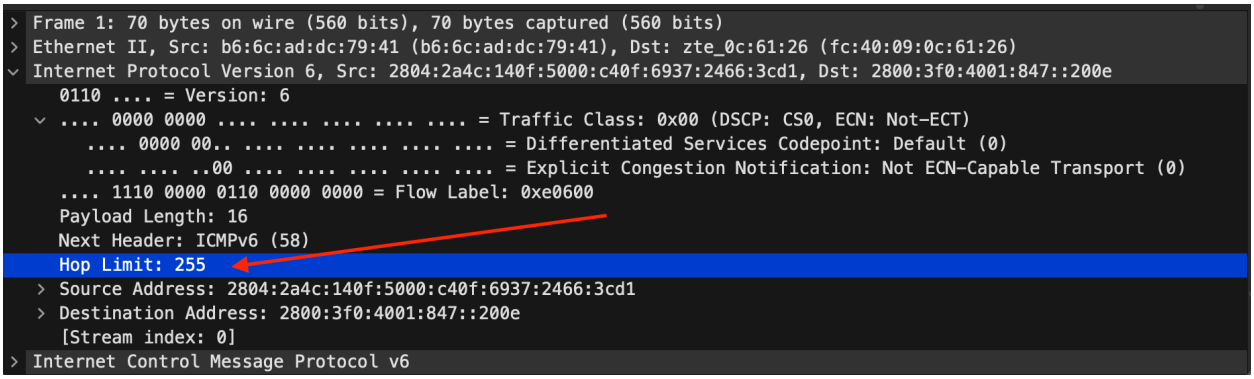


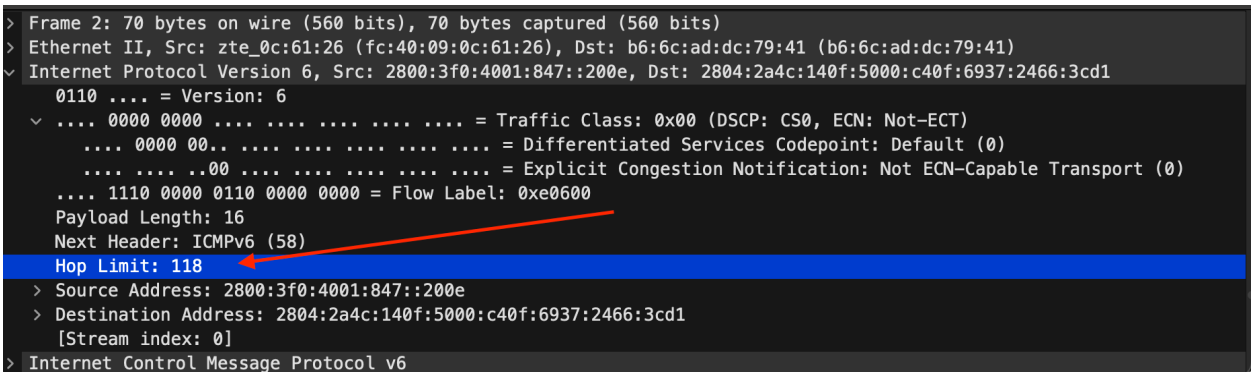
```
Internet Protocol Version 6, Src: 2804:2a4c:140f:5000:c40f:6937:2466:3cd1, Dst: 2800:3f0:4001:847::200e
    0110 .... = Version: 6
  ▾ .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
      .... 0000 00.. .... .... .... .... .... = Differentiated Services Codepoint: Default (0)
      .... .... ..00 .... .... .... .... .... = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    .... 1110 0000 0110 0000 0000 = Flow Label: 0xe0600
    Payload Length: 16
    Next Header: ICMPv6 (58)
    Hop Limit: 255
  › Source Address: 2804:2a4c:140f:5000:c40f:6937:2466:3cd1
  › Destination Address: 2800:3f0:4001:847::200e
    [Stream index: 0]
```

Esse valor foi o mesmo porque o Flow Label já que ele é um identificador de 20 bits que agrupa pacotes pertencentes ao mesmo fluxo de comunicação, permitindo que roteadores os tratem de forma mais eficiente sem precisar analisar protocolos como TCP ou UDP.

Qual os valores para o “Hop Limit”?

O valor do Hop Limit foi diferente entre os pacotes. Os pacotes com o *source* minha máquina, tem o Hop Limit 255, já os que tem como *source* o Google, tem o Hop Limit 118.





A primeira imagem é referente ao pacote que tem o *source* minha máquina e o segundo que tem o *source* o Google. O Hop Limit é o número máximo de roteadores que um pacote pode passar antes de chegar ao destino. Os valores estão diferentes porque o primeiro pacote é o que saiu da minha máquina, ou seja, ele ainda não havia passado por um roteador para decrementar o valor. Já o que recebi do Google, está decrementado, porque o pacote passou por roteadores que decrementaram o valor do Hop Limit. Possivelmente o valor inicial do Hop Limit do pacote do Google era 128(um dos valores padrões), fazendo com que o pacote passasse por 10 roteadores até chegar em minha máquina. Porém, infelizmente com o Wireshark não há como ter certeza do valor inicial do pacote.