FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA" CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM CURSO DE CIÊNCIA DA COMPUTAÇÃO

GUSTAVO MENEZES VENCIGUERA

COMPARAÇÃO DAS FORMAS DE TRANSIÇÃO DE IPV4 PARA IPV6: TEMPO DE RESPOSTA, CONSUMO TOTAL DA BANDA E TEMPO DE ENVIO DE ARQUIVO

Marília

2016

GUSTAVO MENEZES VENCIGUERA

COMPARAÇÃO DAS FORMAS DE TRANSIÇÃO DE IPV4 PARA IPV6: TEMPO DE RESPOSTA, CONSUMO TOTAL DA BANDA E TEMPO DE ENVIO DE ARQUIVO

Trabalho de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília - UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profº Paulo Rogério de Mello Cardoso

Marília

2016

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
MANTIDO PELA FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Gustavo Menezes Venciguera

Comparação das formas de transição de IPv4 para IPv6.

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: B (Oito)

Orientador:Paulo Rogerio de Mello Cardoso

1°.Examinador:Rodolfo Barros Chiaramonte

2°.Examinador:Fábio Dacêncio Pereira

Marília, 07 de dezembro de 2016.

Venciguera, Gustavo Menezes

Comparação das formas de transição de IPv4 para IPv6: TEMPO DE RESPOSTA, CONSUMO TOTAL DA BANDA E TEMPO DE ENVIO DE ARQUIVO / Gustavo Menezes Venciguera; orientador: Prof. Paulo Rogério de Mello Cardoso, SP, 2016.

68 folhas

Monografia (Bacharelado em Ciência da Computação): Centro Universitário Eurípedes de Marília.

Dedico este trabalho a minha família, pelo apoio durante o curso, e em toda a minha vida.

AGRADECIMENTOS

Agradeço primeiramente meus pais, no qual me ajudaram desde a minha escolha, até os últimos passos dessa longa caminha. Agradeço as pessoas que me ajudaram diretamente, professores e amigos, que nas horas que mais precisei, estavam dispostos a me ajudar.

Agradeço também a toda minha família, que também me apoiou na escolha do curso, me incentivando e sempre me ajudando a não desistir, apesar das dificuldades.

Após todos esses anos, devo agradecer também a Deus, que me proporcionou todas as oportunidades que tive de melhorias e engrandecimento pessoal.

“Obstáculo é aquilo que você enxerga quando tira os olhos do seu objetivo.”

Justin Herald - Atitude

RESUMO

O trabalho é a comparação das formas de transição de IPv4 para IPv6, apresentando os motivos que levaram ao término do IPv4, o porque da implantação do IPv6, as melhorias que o IPv6 em relação ao IPv4.

A implementação dessas formas de transição, foi feita em uma rede simulada, na qual foram feitas todas as configurações necessárias para que cada transição seja aplicada da maneira correta e mantenha seu melhor funcionamento.

Teste de latência, de consumo total da rede, tempo de resposta no envio de um arquivo único, trace route no qual será verificado o caminho que o pacote faz, são os testes utilizados no projeto para a escolha da melhor forma de transição para o cenário de uma pequena/média empresa.

Palavras chave: IPv6, Transição, IPv4, Transporte, Pacotes.

ABSTRACT

The work is the comparison of the forms of transition from IPv4 to IPv6, presenting the reasons that led to the termination of IPv4, because of the implementation of IPv6, the improvements that IPv6 in relation to IPv4.

The implementation of these transition forms was done in a simulated network, in which all the necessary configurations were made so that each transition is applied correctly and maintains its better functioning.

Latency test, total network consumption, response time to send a single file, trace route in which to verify the path that the packet makes, are the tests used in the project to choose the best way to transition to the scenario Of a small / medium company.

Keywords: IPv6, Transition, IPv4, Transportation, Packets

LISTA DE IMAGENS

LISTA DE ABREVIATURAS E SIGLAS

IBGE	INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA
IP	INTERNET PROTOCOL
IPV4	INTERNET PROTOCOL VERSION 4
IPV6	INTERNET PROTOCOL VERSION 6
DHCP	DYNAMIC HOST CONFIGURATION PROTOCOL
DOD	DEPARTAMENT OF DEFENSE
ARPA	ADVANCED RESEARCH PROJECTS AGENCY
RFC	REQUEST FOR COMMENTS
IETF	INTERNET ENGINEERING TASK FORCE
ROAD	ROUTING AND ADDRESSING
CIDR	CLASSLESS INTER-DOMAIN ROUTING
IANA	INTERNET ASSIGNED NUMBERS AUTHORITY
DNS	DOMAIN NAME SYSTEM
NAT	NETWORK ADDRESS TRANSLATION
VOIP	VOICE OVER INTERNET PROTOCOL
VPN	VIRTUAL PRIVATE NETWORK
P2P	PEER-TO-PEER
NAPT	NETWORK ADDRESS AND PORT TRANSLATION
TCP	TRANSMISSION CONTROL PROTOCOL
UDP	USER DATAGRAM PROTOCOL
ICMP	INTERNET CONTROL MESSAGE PROTOCOL
ARP	ADDRESS RESOLUTION PROTOCOL
IGMP	INTERNET GROUP MANAGEMENT PROTOCOL
QOS	QUALITY OF SERVICE
TTL	TIME TO LIVE
MTU	MAXIMUM TRANSMISSION UNIT
AH	AUTHENTICATION
ICV	INTEGRITY CHECK VALUE
SPI	SERIAL PERIPHERAL INTERFACE
SAS	SOURCE ADDRESS STATION
ESP	ENCAPSULATING SECURITY PAYLOAD
NTP	NETWORK TIME PROTOCOL

SIP	SESSION INITIATION PROTOCOL
MAC	MEDIA ACCESS CONTROL
RSVP	RESOURCE RESERVATION PROTOCOL
ULA	UNIQUE LOCAL ADRESS
HTTP	HYPERTEXT TRANSFER PROTOCOL
CPU	CENTRAL PROCESSING UNIT
OSPF	OPEN SHORTEST PATH FIRST
ISP	INTERNET SERVICE PROVIDER

SUMÁRIO

INTRODUÇÃO

A internet vem tendo um crescimento rápido, no Brasil, segundo o IBGE (Instituto Brasileiro de Geografia e Estatística), em 2014 os dados mostram que 36,8 milhões de casas estavam conectadas a internet, 54,9% do total. Em 2013 esse índice era de 48%.

O IBGE indiciou que são 95,4 milhões de brasileiros com acesso à internet, entre computadores, smartphones, tablets, TVs e outros dispositivos.

Em uma escala mundial, esse número mais que quadruplicaria, levando também em consideração que esses números podem aumentar consideravelmente, pois hoje a internet vem ficando cada vez mais acessível às classes mais baixas. Acarretando diversas dificuldades na manutenção dos endereços IPv4 já existentes.

Esse crescimento acabou gerando a escassez dos endereços IPv4 e indicando a necessidade de criação de um novo protocolo para que pudessem ser gerados mais endereços e com um tempo de vida útil maior do que o endereço IPv4 teve, assim foi criado o protocolo IPv6.

O IPv6 trouxe com ele melhorias nas formas de atribuição de IP, trouxe também melhoria na forma de roteamento, priorizando o DHCP Stateless, onde o próprio host irá gerar seu endereço, baseando-se no seu endereço MAC e no prefixo enviado a ele.

Motivação e Justificativa

Devido a falta de IPv4 disponível, a utilização do IPv6 é o próximo passo a ser dado, com isso gera uma diversidade de problemas. A transição entre os protocolos, que tem estruturas diferentes, geram diferentes formas de implantação, podendo ter vantagens e desvantagens. O estudo dessas formas de transição é de grande importância, para que ao ser iniciado o processo de transição, com os dados apresentados, seja definido para o ambiente, qual a forma que será mais bem adotada, gerando assim um número menor de incidentes, e com a maior eficiência possível.

Objetivos Gerais

Tendo como objetivo que o estudo feito aqui, possa ser de utilidade, para empresas que implementarão o protocolo IPv6 a sua estrutura, a companhias de fornecimento de internet, para que possam compreender e escolher qual a melhor forma de transição pode ser utilizada. Para cada tipo de cliente, como cliente físico (residencial), cliente corporativo (pequenas / médias / grandes empresas), clientes com necessidades especiais e específicas (clientes dedicados, onde dependem única e exclusivamente de internet para desempenhar suas funções) e para conhecimento sobre esse novo protocolo que vem sendo implementado.

Organização deste documento

Esse projeto foi dividido em 6 partes: primeiramente será feita a apresentação e introdução ao IPv4, apresentando os motivos do esgotamento e as formas paliativas tomadas para aumentar o tempo de vido do protocolo. Segunda parte será feita a apresentação do protocolo IPv6, as melhorias que ele trás. Terceira parte fala sobre como as formas de transição funcionam. Quarta parte fala sobre os softwares utilizados para que a implementação do projeto e geração dos resultados. Quinta parte é feita a apresentação como as formas de transição foram implementadas e os resultados. Sexta parte trás a comparação das formas de transição e qual é a melhor, para o cenário proposto.

1 Contextualização

A teve inicio em um projeto, no qual fosse feita a interligação entre os computadores das bases militares e os computadores das bases de pesquisa. Esse projeto foi feito pela Agencia de Pesquisas e de Projetos Avançados (ARPA - Advanced Research Projects Agency), em um projeto iniciado pelo Departamento de Defesa dos Estados Unidos, no ano de 1966. (ipv6.br, 2012).

O projeto tinha como objetivo principal formar uma arquitetura sólida capaz de não perder a comunicação mesmo com a queda de alguma estação, mantendo a comunicação com as demais estações. (ipv6.br, 2012).

1.1 Protocolo IP

Conforme a RFC 791 o protocolo IP foi projetado com as seguintes intenções:

> *"O protocolo de Internet é projetado para uso em sistemas interligados de redes de comunicação de computadores. O protocolo de internet prevê transmissão de blocos de dados chamado datagramas... O protocolo de internet também prevê fragmentação e remontagem, para o envio de pacotes maiores que o limite do tráfego estabelecido."*

A versão do protocolo utilizada até então é a versão 4, conhecida como protocolo IPv4.

O protocolo apresenta uma fácil implementação e interoperabilidade, porém seu projeto original não previa alguns problemas pelo qual o protocolo poderia passar, esgotamento dos endereços IP; o aumento da tabela de roteamento; problemas com a segurança do dados transmitidos; necessidade de priorizar um determinado tipo de pacote. (ipv6.br, 2012).

1.2 Endereços IPv4

Classe	Formato	Redes	Hosts
A	7 bits Rede, 24 bits Host	128	16.77.216
B	14 bits Rede, 16 bits Host	16.384	66.536
C	21 bits Rede, 8 bits Host	2.097.152	256

Imagem 1: Tabela de classes. Fonte: ipv6.br, 2012.

O IPv4 reserva 32 bits para endereçamento, o que possibilita mais de 4 bilhões de endereços distintos. Conforme a imagem 2, esses endereços foram divididos da seguinte maneira:

- Classe A: define o bit mais significativo como 0, utiliza os 7 bits restantes do primeiro octeto para identificar a rede, e os 24 bits restantes para identificar o host. Esses endereços utilizam a faixa de 1.0.0.0 até 126.0.0.0 (RFC 1918, 1996);

- Classe B: define os 2 bits mais significativos como 10, utiliza os 14 bits seguintes para identificar a rede, e os 16 bits restantes para identificar o host. Esses endereços utilizam a faixa de 128.1.0.0 até 191.254.0.0 (RFC 1918, 1996).;

- Classe C: define os 3 bits mais significativos como 110, utiliza os 21 bits seguintes para identificar a rede, e os 8 bits restantes para identificar o host. Esses endereços utilizavam a faixa de 192.0.1.0 até 223.255.254.0 (RFC 1918, 1996);

O intuito desse tipo de divisão, definida pela IETF (Internet Engineering Task Force) em 3 de agosto de 1990, com auxílio das RFC 1287 e RFC 1296, foi criar uma flexibilidade na distribuição dos endereços, englobando redes de diferentes tipos de tamanho, porém ocorre um problema com esse tipo de divisão. A classe A, atende um número muito pequeno de redes e ocupa metade de todos os endereços disponíveis, enquanto a classe C permite criar diversas redes, só que com poucos endereços disponíveis.

Enquanto uma classe gera desperdício a outra classe gera falta de endereços para atender a demanda. Utilizando um exemplo simples, se é necessário disponibilizar 300 endereços para dispositivos, seria necessário utilizar um bloco de endereços de classe B, porém seriam desperdiçados em torno de 65 mil endereços.

Devido ao ritmo de crescimento da internet e da política de distribuição de endereços, que apresentava desperdícios, com disponibilização de faixas classe A, para grandes empresas, algumas faixas também separadas para multicast, loopback e uso no futuro, 38% das faixas de endereços classe A, 43% da classe B e 2% da classe C, já estavam alocados, é possível verificar a distribuição feita na imagem 2. Nesta época, a rede já possuía 1.136.000 hosts conectados. (Pablo Luis Fazzanaro – Protocolo IPv6 Uma Abordagem Geral, 2013).

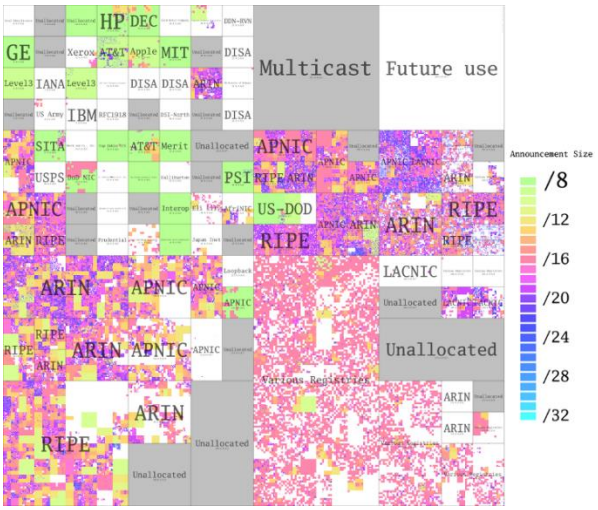

Imagem 2: Tabela de distribuição de endereços IP no ano de 2008. Fonte: maps.measurement-factory.com/, 2009.

1.3 Soluções Paliativas

Diante de problemas recorrentes ao alto crescimento da internet, a IETF (Internet Engeenering Task Force) passou a discutir e implantar novas técnicas para solucionar o problema com esgotamento de endereços IP e o problema do aumento da tabela de roteamento. A partir daí o IETF cria o ROAD (Routing and Addressing), o grupo de trabalho responsável por apresentar soluções para estes problemas.

1.3.1 CIDR

O ROAD tem como ideia básica o fim do uso de classes de endereços. “No sentido mais simples, a mudança de classe A / B / C de números de rede classes prefixos é tornar explícito que os bits em um endereço IPv4 32-bit são interpretados como o número da rede (ou prefixo) associados com um site e que são usados para o número final sistemas individuais dentro do site. Na notação CIDR, um prefixo é apresentado como quatro octetos, apenas como uma tradicional rede IPv4 ou número, seguido pelo "/" (barra) de caracteres, seguido por um valor decimal entre 0 e 32 que descreve o número de bits significativos. (RFC 4632, 2006).

Com a utilização do CIDR os blocos são referenciados como prefixo de rede. No endereço 192.168.0.65/X, o “X” indica os bits mais significativos, isso se aplica para qualquer outra estrutura de endereço, mais conhecido como máscara de sub-rede.

Usando classes de prefixos com comprimentos, permite uma melhor flexibilidade com a utilização dos blocos correspondentes de real necessidade. Anteriormente eram utilizados apenas 3 tamanhos de redes, sendo que após a implementação do CIDR, a rede pode ser segmentada de acordo com o tamanho necessário.

Todas as faixas de endereços são atribuídas pela IANA às entidades regionais. Onde são divididas em faixas de endereços menores e são atribuídos às operadores responsáveis pelos links, empresas de hospedagem, provedores de acesso e outras instituições. Cada empresa que recebe essas faixas de endereços os quebra em faixas menores para atribuição aos consumidores finais. (RFC 4632, 2006).

1.3.2 Protocolo DHCP

O protocolo DHCP foi outra solução apresentada pelo ROAD. Através do protocolo, um host é capaz de obter um endereço IP automaticamente, adquirindo máscara de sub-rede, endereço IP e endereço do servidor DNS do roteador padrão. (ipv6.br, 2012)

O DHCP oferece endereços IP temporários, não necessitando gerar um endereço para cada host conectado.

O Dynamic Host Configuration Protocol (DHCP) oferece configuração de parâmetros para hosts da Internet. DHCP consiste em dois componentes: um Protocolo para a entrega de

parâmetros de configuração específicos do hospedeiro a partir de um servidor DHCP para um host e um mecanismo para a atribuição de rede aos hosts. DHCP é construído sobre um modelo cliente-servidor, onde o servidor DHCP designado aloca endereços de rede e fornecer parâmetros de configuração para configuração dinâmica dos hosts. (Ralph Droms, RFC 2131).

O DHCP suporta três mecanismos de atribuição de endereços IP. Em "Atribuição automática", o DHCP atribui um endereço IP permanente para o cliente. Em "alocação dinâmica", DHCP atribui um endereço IP a um cliente por um período limitado de tempo (ou até que o cliente explicitamente abandone o endereço). Em "alocação manual", IP de um cliente é atribuído pelo administrador da rede, e o DHCP é usado simplesmente para transmitir o endereço atribuído para o cliente. Uma rede particular irá utilizar um ou mais desses mecanismos, dependendo da política do administrador de rede. (Ralph Droms, RFC 2131).

1.3.3 NAT

NAT foi mais uma técnica desenvolvida para economia dos endereços IP. A ideia do NAT gira basicamente em um único IP, ou um pequeno número de endereços IP, em que diferentes hosts possam trafegar na internet, tendo endereços para comunicação interna e quando fosse necessária a comunicação externa, utiliza o endereço global.

Dentro de uma rede, cada host recebe um IP interno (privado) único, que é utilizado para o roteamento internet, porém ao enviar um pacote para fora da rede, há a tradução do endereço interno para um endereço público (global), no qual será reconhecido na rede externa. (Pyda Srisuresh & Kjeld Borch Egevang, RFC 3022).

O NAT apresentou melhorias na economia de endereços IP, facilitação na numeração interna das redes, o gerenciamento de rede ficou mais organizado, encaminhando somente pacotes que foram solicitados por algum host interno. (Pyda Srisuresh & Kjeld Borch Egevang, RFC 3022).

Todavia apresenta também desvantagens, como dois hosts perderem o acesso direto entre eles, o que dificulta alguns serviços como VoIP, VPN, P2P. O número de conexões simultâneas é limitado, além de exigir um grande processamento. (Pyda Srisuresh & Kjeld Borch Egevang, RFC 3022).

A necessidade de tradução de endereço IP surge quando uma rede de endereços de IP interno não pode ser utilizada fora da rede, quer por razões de privacidade ou porque são inválidas para uso fora da rede. Topologia de rede fora de um domínio local pode mudar de muitas maneiras. Os clientes podem mudar de fornecedor, backbones da empresa podem ser reorganizados, ou provedores podem fundir ou dividir. Sempre que com o tempo as mudanças de topologia externa ocorrer, a atribuição de endereços para nós dentro do domínio local também deve mudar para refletir as mudanças externas. Mudanças deste tipo podem ser

escondidas de usuários dentro do domínio através da centralização de alterações em um único roteador de tradução de endereços. (Pyda Srisuresh & Kjeld Borch Egevang, RFC 3022).

O tradicional NAT permite que os hosts dentro de uma rede privada, acessem de maneira transparente hosts da rede externa, na maioria dos casos. Em um NAT tradicional, sessões são unidirecionais, de saída da rede privada. Sessões em direção oposta podem ser autorizadas a título excepcional usando endereço estático mapeado para hosts pré-selecionados. NAT básico e NAPT são duas variações do tradicional NAT, em que a tradução em NAT Básico é limitada a endereços IP sozinho, enquanto que a tradução em NAPT é estendida para incluir o endereço IP e identificador de Transporte (tais como TCP / UDP ou ICMP ID query) (Pyda Srisuresh & Kjeld Borch Egevang, RFC 3022).

2 Ipv6

O Ipv6 diferentemente do IPv4, tem 128 bits de endereçamento, enquanto IPv4 possuí 32 bits. Uma alteração significativa nos números de IPs gerados, permitindo níveis mais específicos de agregação de endereços, podendo identificar uma quantidade maior de dispositivos na rede, implementando um mecanismo de autoconfiguração. A escalabilidade do endereçamento multicast, também foi melhorada, com a adição do campo Escopo. E um novo tipo de endereço, o endereço anycast, foi definido. (Curso Ipv6 Básico, 2010)

IPv4	**IPv6**
32 Bits	128 Bits
ICMPv4 reporta erros	ICMPv6 responsável pelas funções dos protocolos ARP
Cabeçalho com campos fixos	Cabeçalhos de extensão, não processados por roteador
DHCP feito por roteadores	DHCP feito sem a utilização de roteadores
Em caso de troca de provedor, endereços da rede devem ser renumerados	Em caso de troca de provedor, a renumeração é feita automaticamente através do servidor
MTU pode se fragmentar durante o caminho	MTU é fragmentado somente no envio

Alguns campos que eram utilizados no IPv4 foram removidos e tornaram-se opcionais.

Suporte a cabeçalho de extensão, com as opções não fazendo mais parte do cabeçalho de base, o roteamento se torna mais eficaz, com limites menos rigorosos em relação ao tamanho e a quantidade de opções e uma maior flexibilidade para a introdução de novas opções no futuro.

Um novo recurso, permitindo identificar pacotes que pertençam a determinados tráfegos de fluxos, e que precisem de algum tipo de tratamento especial, foram adicionados.

Foram especificados cabeçalhos de extensão, capazes de fornecer mecanismos de autenticação e garantir a integridade e a confiabilidade dos dados transmitidos. (Curso Ipv6 Básico, 2010) e (RFC 2460, 1998).

2.1 ICMPv6

Definido na RFC 4443 para ser utilizado com o Ipv6, o ICMPv6 é uma versão atualizada do ICMP, utilizado no Ipv4.

Embora apresente as mesmas funções que o ICMPv4, reportando erros no processamento de pacotes e enviando mensagens sobre o status e as características da rede, ele não é compatível com o antecessor, com um número maior de mensagens e funcionalidades que o antigo. Agora o ICMPv6 é responsável pelas funções dos protocolos ARP, mapeando os endereços de camada dois para IPs, e vice-versa no Ipv4, e é responsável pelo IGMP,

gerenciando os membros dos grupos multicast no IPv4. (RFC 4443, 2006).

2.1.1 Cabeçalho

O cabeçalho, utilizado no IPv4, é composto por 12 campos fixos, de tamanho variável entre 20 e 60 bytes.

Esses campos são responsáveis pela transmissão de informações sobre o protocolo que esta sendo transportado, ou seja, informando sobre o tamanho dos dados, se os pacotes estão fragmentados, o tipo de dado que esta sendo transportado, qual o protocolo da camada seguinte, a origem e destino e a integridade dos dados. (ipv6.br, 2012).

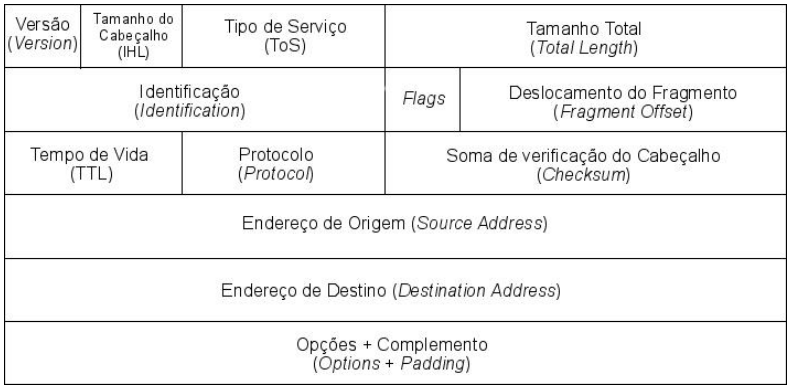

Imagem 3: Cabeçalho IPv4. Fonte: ipv6.br.

Já no IPv6 o cabeçalho foi modificado, do padrão assumido anteriormente no IPv4.

Foram realizadas alterações de modo a torná-lo mais simples. O número de campos foi reduzido para apenas oito e o tamanho fixado de 40 bytes. Ficou mais flexível e eficiente com a adição de cabeçalhos de extensão, que não precisam ser processados por roteadores intermediários, essas alterações fazem com que mesmo o espaço de endereçamento quatro vezes maior que o IPv4, o tamanho total do cabeçalho IPv6 seja apenas duas vezes maior. (ipv6.br, 2012).

Ocorreram remoções em alguns campos do cabeçalho. O campo “Tamanho do Cabeçalho”, encontrado no IPv4, no IPv6 tornou-se desnecessário, uma vez que o valor foi fixado. Os campos “Identificação”, “Flags”, ”Deslocamento do Fragmento” e “Opções e Complementos” passaram a ter suas informações indicadas em cabeçalhos de extensão apropriados. O campo “Soma de Verificação” foi descartado com o objetivo de deixar o protocolo mais eficiente já que outras validações são realizadas pelos protocolos das camadas superiores da rede. (ipv6.br, 2012).

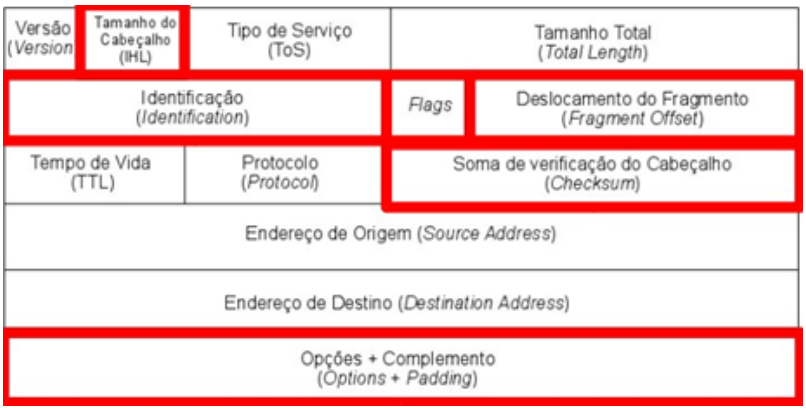

Imagem 4: Campos que perderam o valor no cabeçalho IPv6. Fonte: ipv6.br

A imagem 5 é referente ao cabeçalho IPv6, após todos as alterações dos campos que perderam valor.

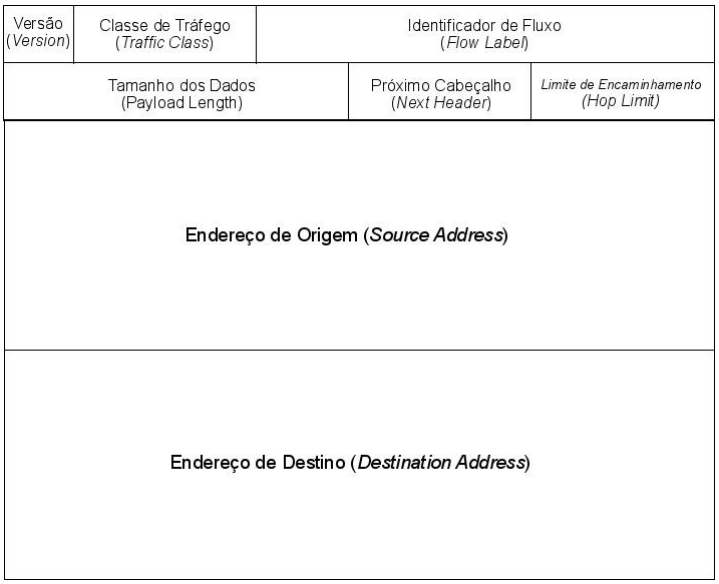

Imagem 5: Cabeçalho IPv6. Fonte: ipv6.br.

O campo “Identificador de Fluxo” foi adicionado, para possibilitar o funcionamento de um mecanismo extra de suporte a QoS. (ipv6.br, 2012).

Também ouve a alteração no nome dos campos, com intuito de agilizar o processamento, e reposicionamento de quatro campos:

IPv4	IPv6
Tipo de Serviço	Classe de Serviço
Tamanho Total	Tamanho dos Dados
Tempo de Vida (TTL)	Limite de encaminhamento
Protocolo	Próximo Cabeçalho

Imagem 6: Renomeação de campos do cabeçalho. Fonte: ipv6.br.

O Cabeçalho IPv6, conforme imagem 5, está dividido da seguinte maneira:

Versão (4 bits) - Identifica a versão do protocolo utilizado. No caso, o valor desse campo é 6.

Classe de Tráfego (8 bits) - Identifica os pacotes por classes de serviços ou prioridade. Ele provê as mesmas funcionalidades e definições do campo "Tipo de Serviço do IPv4".

Identificador de Fluxo (20 bits) - Identifica pacotes do mesmo fluxo de comunicação. Idealmente esse campo é configurado pelo endereço de destino para separar os fluxos de cada uma das aplicações e os nós intermediários de rede podem utiliza-lo de forma agregada com os endereços de origem e destino para realização de tratamento específico dos pacotes.

Tamanho do Dados (16 bits) - Indica o tamanho, em Bytes, apenas dos dados enviados junto ao cabeçalho IPv6. Substituiu o campo Tamanho Total do IPv4, que indicava o tamanho do cabeçalho mais o tamanho dos dados transmitidos. Contudo, o tamanho dos cabeçalhos de extensão também é somado nesse novo campo.

Próximo Cabeçalho (8 bits) - Identifica o cabeçalho de extensão que segue o atual. Ele foi renomeado (no IPv4 chamava-se Protocolo) para refletir a nova organização dos pacotes IPv6, uma vez que ele deixou de conter os valores referentes a outros protocolos, para indicar os tipos dos cabeçalhos de extensão.

Limite de Encaminhamento (8 bits) - 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. Ele padronizou o modo como o campo Tempo de Vida (TTL) do IPv4 vinha sendo utilizado, o qual diferia significativamente da descrição original que o definia como o tempo, em segundos, para o pacote ser descartado caso não chegasse à seu destino.

Endereço de origem (128 bits) - Indica o endereço de origem do pacote.

Endereço de Destino (128 bits) - Indica o endereço de destino do pacote. (Stephen E. Deering & Robert M. Hinden, RFC 2460**).**

2.1.2 Cabeçalho de Extensão

Diferentemente do IPv4, que no cabeçalho base trás todas as informações opcionais, o IPv6 trata essas informações através de um cabeçalho de extensão. (Curso IPv6 Básico, 2010).

O cabeçalho de extensão se localiza entre o cabeçalho base o cabeçalho da camada, e não possuem quantidade ou tamanho fixo. No caso de existirem muitos cabeçalhos, eles serão adicionais sequencialmente formando uma “cadeia de cabeçalhos”. (Curso IPv6 Básico, 2010).

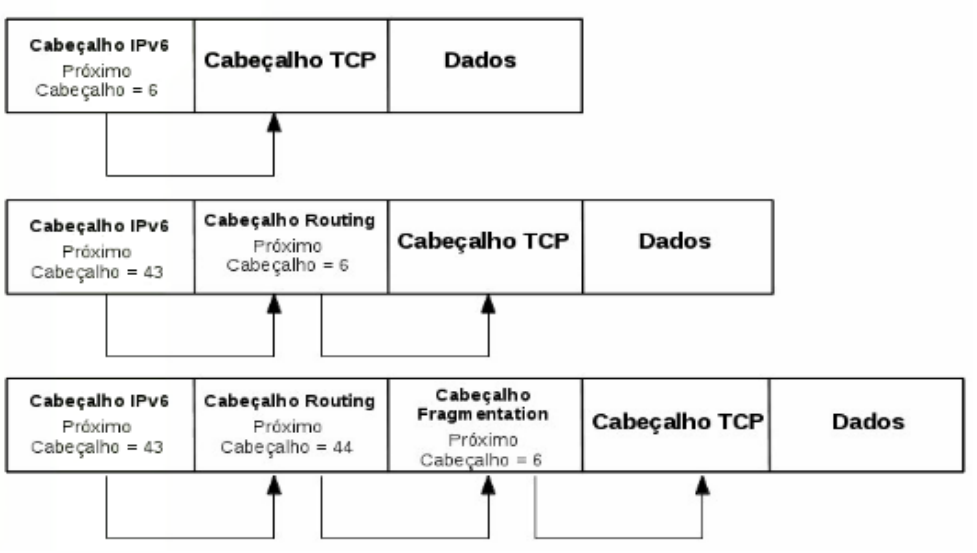

Imagem 7: Sequência de cabeçalhos. Fonte: ipv6.br.

2.1.2.1 HOP-BY-HOP

A opção de cabeçalho Hop-by-Hop é usado para transportar informações opcionais que devem ser examinadas por cada nó ao longo do trajeto de entrega de um pacote. O cabeçalho Hop-by-Hop é identificado por um valor do próximo cabeçalho de 0 no cabeçalho IPv6, e tem o seguinte formato:

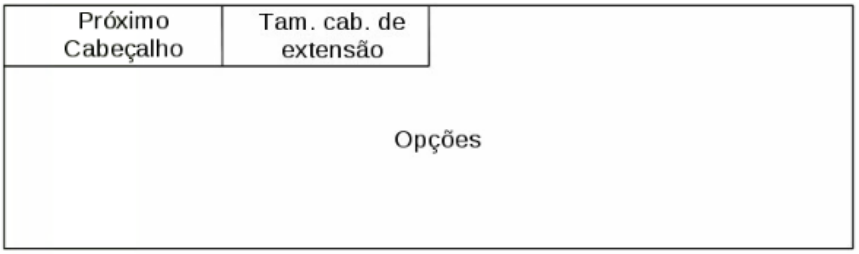

Imagem 8: Cabeçalho Hop-by-Hop. Fonte: ipv6.br.

Próximo Cabeçalho (8 bits) - Identifica o tipo de cabeçalho imediatamente após o cabeçalho Hop-by-Hop. Usa os mesmos valores como o campo de protocolo IPv4.

Tamanho Cabeçalho de Extensão (8 bits) - Comprimento do cabeçalho Hop-by-Hop em unidades de 8 octetos, não incluindo os primeiros 8 octetos.

Opções - Campo de comprimento variável , de tal modo que o comprimento completa cabeçalho Hop-by-Hop é um inteiro múltiplo de 8 octetos de comprimento. O valor dos dois primeiros bits especificam qual das seguintes ações devem ser tomadas:

00 - pular esta opção e continuar o processamento do cabeçalho;

01 - descartar o pacote;

10 – descartar o pacote e enviar uma mensagem ICMP Parameter Problema para o endereço de origem do pacote;

11 – descartar o pacote e enviar uma mensagem ICMP Parameter Problem para o endereço de origem do pacote, somente se o destino não for um endereço multicast.

O terceiro bit de ordem mais alta indica se a informação opcional pode mudar ou não de rota (valor 1 ou 0), respectivamente. (Stephen E. Deering & Robert M. Hinden, RFC 2460).

2.1.2.2 Destination Options

O cabeçalho Destination Options é usado para transportar informações opcionais que precisam ser analisadas apenas pelo nó de destino de um pacote(s). O cabeçalho Destination Options é identificado pelo valor 60 no campo Próximo Cabeçalho, e tem o formato exatamente igual ao do Hop-by-Hop. Ele é utilizado no suporte ao mecanismo de mobilidade do IPv6 através da opção Home Address, que contém o Endereço de Origem do Nó Móvel quando este está em transito. (ipv6.br, 2012).

2.1.2.3 Routing

O cabeçalho Routing é utilizado por uma fonte IPv6 para listar um ou mais nós intermediários a serem "visitados" a caminho de um pacote de destino. Esta função é muito semelhante ao de origens do IPv4 e opção Record Route. O cabeçalho Routing é identificado pelo campo Próximo Cabeçalho de valor 43, e tem o seguinte formato: (Stephen E. Deering & Robert M. Hinden, RFC 2460).

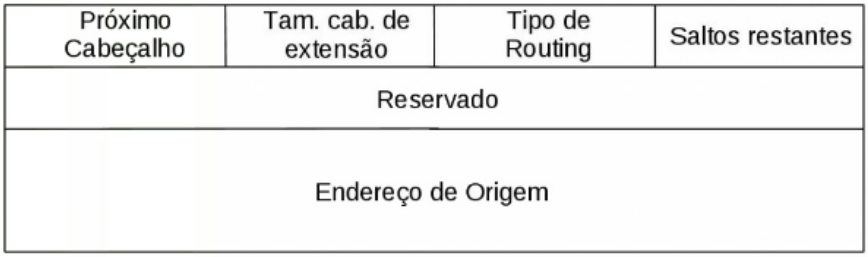

Imagem 9: Cabeçalho Routing. Fonte: ipv6.br.

O cabeçalho de extensão Routing foi desenvolvido inicialmente para listar um ou mais nós intermediários que deveriam ser visitados até o pacote chegar ao destino, de forma semelhante às opções Loose Source e Record Route do IPv4. No entanto, esta função tornou-se obsoleta pela RFC 5095 devido a problemas de segurança. Um novo cabeçalho Routing, tipo 2, foi definido para ser utilizado como parte do mecanismo de suporte a mobilidade do IPv6. (ipv6.br, 2012).

Segundo essa nova definição, ele deve carregar o Endereço de Origem do Nó Móvel em pacotes enviados pelo Nó Correspondente. As definições de cada campo desse cabeçalho são as seguintes:

Próximo Cabeçalho (8 Bits): Identifica o tipo de cabeçalho que segue ao cabeçalho

Routing.

Tamanho do Cabeçalho de Extensão (8 Bits): Indica o tamanho seu tamanho (em unidades de 8 Bytes) excluídos os oito primeiros bits.

Tipo de Routing (8 Bits): Identifica o tipo de cabeçalho Routing. Atualmente apenas o Type 2 está especificado.

Saltos restantes: Definido para ser utilizado com o Routing Type 0, indica o número de saltos a serem visitados antes de o pacote atingir seu destino final.

Endereço de Origem: Carrega o Endereço de Origem de um Nó Móvel. (RFC 3775, 2004).

2.1.2.4 Fragmentation

O cabeçalho Fragmentation é utilizado por uma fonte IPv6 para enviar um pacote maior que se encaixam no caminho MTU para o seu destino. (Nota: ao contrário do IPv4, a fragmentação em IPv6 é realizada apenas por nós de origem, e não por roteadores ao longo caminho de entrega de um pacote). O cabeçalho Fragmentation é identificado pelo valor 44 no campo Próximo Cabeçalho, e tem o seguinte formato: (IPv6 Básico, 2012).

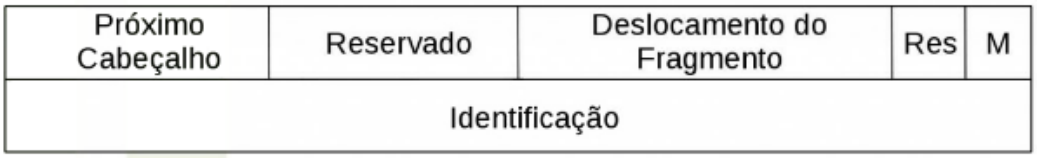

Imagem 10: Cabeçalho Fragmentation. Fonte: ipv6.br.

Próximo Cabeçalho (8 Bits): Identifica o tipo de cabeçalho que segue ao cabeçalho Fragmentation.

Reservado (8 Bits): Inicializado em zero para transmissão; ignorado na recepção.

Deslocamento do Fragmento (13 Bits): Indica, em unidades de oito Bytes, a posição dos dados transportados pelo fragmento atual em relação ao início do pacote original.

Res (2 Bits): Inicializado em zero para transmissão; ignorado na recepção.

Flag M (1 Bit): Se marcado com o valor 1, indica que há mais fragmentos. Se marcado com o valor 0, indica que é o fragmento final.

Identificação (32 Bits): Valor único gerado pelo nó de origem, para identificar o pacote original. É utilizado para detectar os fragmentos de um mesmo pacote. (Stephen E. Deering & Robert M. Hinden, RFC 2460).

2.1.2.5 Authentication (AH)

O protocolo AH, provêm a proteção de repetição, autenticação da origem dos dados e a integridade dos dados, porém não proporcionando a confiabilidade de que os dados que estão

sendo enviados, são seguros. (Stephen Kent, RFC 4302).

Próximo Cabeçalho	Tamanho Payload	Reservado
Índice de Parâmetros de Segurança (SPI)		
Número Sequencial		
Dados de Autenticação (ICV)		

Imagem 11: Cabeçalho Authentication.

Próximo Cabeçalho (8 Bits): O próximo cabeçalho é um campo que identifica o tipo da próxima payload após o cabeçalho de autenticação. O valor deste campo é escolhido a partir do conjunto de números de protocolo IP definido na página web da Internet Assigned Numbers Authority (IANA). Por exemplo, um valor de 4 indica o IPv4 , um valor de 41 indica o IPv6, e um valor de 6 indica TCP.

Tamanho Payload (8 Bits): Este campo especifica o comprimento do AH em palavras de 32 bits (4 bytes unidades), menos "2". Assim, por exemplo, se um algoritmo de integridade produz um valor de autenticação 96 - bit, este campo de comprimento será "4" (3 campos de texto fixo de 32 bits além de 3 palavras de 32 bits para o ICV, menos 2). Para o IPv6, o comprimento total do cabeçalho deverá ser um múltiplo de unidades de 8 octetos. (Note que, embora IPv6 [DH98] caracteriza AH como uma extensão de cabeçalho, o seu comprimento é medido em palavras de 32 bits, não por palavras de 64 bits usadas por outros cabeçalhos de extensão IPv6).

Reservado (16 Bits): Este campo está reservado para uso futuro. Ele deve ser ajustado para "Zero" pelo remetente e deve ser ignorado pelo destinatário. (Note-se que o valor é incluído no cálculo ICV, mas caso contrário, é ignorado pelo destinatário).

Índice de Parâmetros de Segurança (SPI) (32 Bits): O SPI é utilizado por um receptor para identificar a SA que um pacote de entrada está ligado. Para uma SA unicast, o SPI pode ser usada por si só para especificar uma SA, ou pode ser utilizada em conjunto com o tipo de protocolo IPsec (neste caso, AH). Uma vez que para o SAS unicast o valor SPI é gerado pelo receptor, se o valor for suficiente para identificar um SA, por si só ou se o mesmo deve ser utilizado em conjunto com o valor do protocolo IPsec é uma questão local. O campo SPI é obrigatório, e este mecanismo para mapear o tráfego de entrada para SAs unicast acima descritos devem ser suportado por todas as implementações AH.

Número Sequencial (32 Bits): Este campo contém um valor de contagem que aumenta um para cada pacote enviado, isto é, por um número de sequência de pacote - SA. Para um SA unicast ou de um remetente único multicast SA, o remetente incrementa este campo para cada pacote transmitido. Compartilhamento de uma SA entre vários remetentes é permitida, embora

geralmente não recomendado.

Dados de Autenticação (ICV) (32 Bits): Este é um campo de comprimento variável que contém o valor de verificação de integridade (ICV) para este pacote. O campo deve ser um múltiplo inteiro de 32 bits (IPv4 ou IPv6) de comprimento. Este campo pode incluir o preenchimento explícito, se necessário para assegurar que o comprimento do cabeçalho AH é um múltiplo inteiro de 32 bits (IPv4) ou 64 bits (IPv6). (Stephen Kent, RFC 4302).

2.1.2.6 Encapsulating Security Payload

O cabeçalho ESP, provê a segurança dos serviços fornecidos entre as comunicações dos hosts, em combinação com o AH.

A comunicação entre hosts, gateways e outros equipamentos é feita através desse cabeçalho, sendo assim o cabeçalho AH faz a autenticação dos dados e o ESP faz a segurança, para que seja enviado sem nenhum problema. (Stephen Kent, RFC 4303).

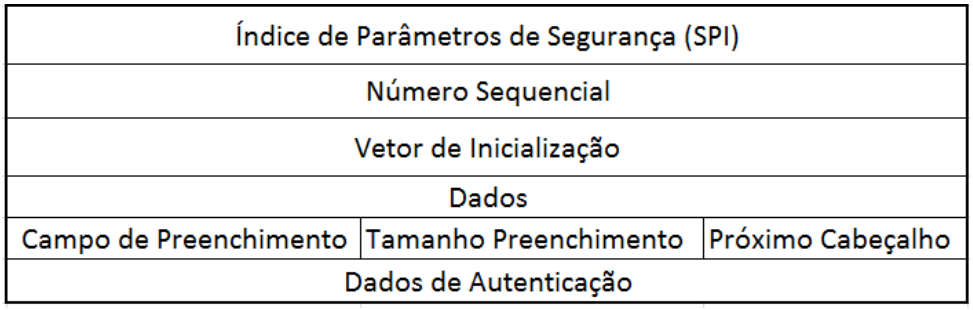

Imagem 12: Cabeçalho Encapsulating Security Payload.

Índice de Parâmetros de Segurança (SPI) (32 Bits): O SPI é utilizado por um receptor para identificar a SA que um pacote de entrada está ligado. Para uma SA unicast, o SPI pode ser usada por si só para especificar uma SA, ou pode ser utilizada em conjunto com o tipo de protocolo IPsec (neste caso, AH). Uma vez que para o SAS unicast o valor SPI é gerado pelo receptor, se o valor for suficiente para identificar um SA, por si só ou se o mesmo deve ser utilizado em conjunto com o valor do protocolo IPsec é uma questão local. O campo SPI é obrigatório, e este mecanismo para mapear o tráfego de entrada para SAs unicast acima descritos devem ser suportado por todas as implementações AH.

Número Sequencial (32 Bits): Este campo contém um valor de contagem que aumenta um para cada pacote enviado, isto é, por um número de sequência de pacote - SA. Para um SA unicast ou de um remetente único multicast SA, o remetente incrementa este campo para cada pacote transmitido. Compartilhamento de uma SA entre vários remetentes é permitido, embora geralmente não recomendado.

Vetor de Inicialização / Dados Payload: Dados Payload é um campo de comprimento variável que contém os dados (do pacote IP original) descritos pelo campo próximo cabeçalho.

O campo de dados Payload é obrigatório e é um integrante do número de bytes de comprimento. Se o algoritmo utilizado para encriptar o Payload requer criptografia de dados de sincronização, por exemplo, um vetor de inicialização (IV), então este dados são levados explicitamente em campo de carga útil, mas isso não é posto como um campo separado em ESP, isto é, a transmissão de um IV explícito é invisível ao ESP.

Campo de Preenchimento: Campo de preenchimento para o algoritmo de alinhamento ou razões citados acima poderia ser usado para ocultar o comprimento real do payload, em apoio ao TFC. No entanto, o campo de preenchimento descrito é muito limitado para ser eficaz para o TFC e, portanto, não deve ser utilizado para esse fim.

Tamanho do Preenchimento: O campo Tamanho do Preenchimento indica o número de bytes de preenchimento imediato precedendo-o no campo de preenchimento. A gama de valores válidos é 0 a 255 , onde um valor de zero indica que nenhum byte de preenchimento se encontram presente.

Próximo Cabeçalho: O próximo cabeçalho é um campo que identifica o tipo da próxima payload após o cabeçalho de autenticação. O valor deste campo é escolhido a partir do conjunto de números de protocolo IP definido na página web da Internet Assigned Numbers Authority (IANA). Por exemplo, um valor de 4 indica o IPv4 , um valor de 41 indica o IPv6, e um valor de 6 indica TCP.

Dados de Autenticação: O Integrity Check Value é um campo de comprimento variável calculado sobre o cabeçalho ESP, Payload, e ESP Trailer Field. Implica ESP Trailer Fields (preenchimento de integridade e de alta ordem ESN bits, se aplicável) são incluídos no cálculo ICV. O campo ICV é opcional. (Stephen Kent, RFC 4303).

2.1.3 Mensagens

O protocolo de mensagens de controle da internet para IPv6 é um padrão IPv6 necessário. Definido na RFC 2463, os hosts e roteadores que usam a comunicação IPv6 reportam erros e enviam mensagens de caráter simples, geralmente são enviadas automaticamente, quando um pacote não chega em seu destino.

As mensagens são identificadas no cabeçalho ICMPv6, porém não são confiáveis, já que transportadas em pacotes IPv6.

2.1.4 MLD

Descoberta de escuta de difusão seletiva (MLD), é definido como TCP/IP na RFC 1112. A RFC define o endereço e as extensões de host para o modo como os hosts IP oferecem suporte à difusão seletiva. Os mesmos conceitos originalmente desenvolvidos para a versão atual do IP, conhecido como IP versão 4 (IPv4), também se aplicam ao IPv6.

O tráfego de difusão seletiva é enviado a um único endereço, porém processados por vários outros hosts. Apenas o computador host, que pertencem ao grupo de difusão, recebem e processam o tráfego enviado ao endereço reservado do grupo. Os hosts que escutam em um endereço de difusão seletiva específico é chamado de grupo de difusão seletiva.

Alguns outros aspectos importantes do MLD são os seguintes, participar ou sair do grupo é uma escolha dinâmica, pois o host é permitido ingressar ou sair do grupo a qualquer momento. O ingresso do host no grupo, é feito através de um envio de mensagem de participação.

Cada grupo não são limitados por tamanho e os membros podem se espalhar por outros segmentos de rede, em caso de os roteadores oferecem suporte ao encaminhamento de tráfego de difusão seletiva.

Um host pode enviar tráfego ao endereço de um grupo, sem que faça parte do grupo.

2.1.5 Neighbor Discovery

O protocolo de descoberta de vizinhança, adicionou métodos não existentes na versão anterior do protocolo IP, se tornou mais dinâmico em comparação com outros processos de configurações de rede IPv4. (ipv6.br, 2012).

O Neighbor Discovery é utilizado por hosts e roteadores, para determinar o endereço MAC dos nós da rede, encontrar roteadores vizinhos, determinar informações de configurações de rede, endereços duplicados, acessibilidade dos roteadores, redirecionamento de pacotes, autoconfiguração de endereços. (RFC 4861, 2007).

2.2 DHCP

O DHCP faz a distribuição de endereços IP dinamicamente em uma rede, mantendo um maior controle na atribuição de endereços aos hosts.

O DHCPv6 pode ser utilizado quando não há roteadores na rede, fornecendo endereços IPv6 e diversos parâmetros de rede, como endereços de servidores DNS, NTP, SIP. Para que isso seja feito a troca de mensagem entre cliente e servidor, é feita através do protocolo UDP. (RFC 3315, 2003).

Os clientes utilizam um endereço local, transmitindo ou recebendo DHCP, e os servidores utilizam um endereço multicast reservado, recebendo mensagens dos clientes. Em casos que o cliente tenha a necessidade de enviar mensagem a um servidor fora de sua sub-rede, é utilizado um Relay DHCP. (RFC 3315, 2003).

2.2.1 Stateless

Essa modalidade consiste em o servidor não manter um registro de dos endereços

atribuídos aos hosts, porque cada host (máquina) irá formar automaticamente seu endereço, a partir de seu endereço físico (MAC) e prefixo dos roteadores. O Servidor DHCPv6 informa apenas os endereços como, DNS e/ou Opções. (Servidores DHCPv6 em Redes IPv6, 2013).

Essa modalidade gera uma base de processo de autoconfiguração, exemplificando e consumindo menos recursos do servidor. (Servidores DHCPv6 em Redes IPv6, 2013).

2.2.2 Stateful

Apesar de pouco utilizada, ainda existe a versão Stateful do DHCPv6, para aqueles que precisam manter os registros de endereço dinamicamente atribuídos e que querem determinar o escopo explicitamente. Normalmente utilizada em servidores Linux e Windows Server, no qual, dependendo da funcionalidade, necessita fazer o controle dos endereços registrados. Também utilizado para caso o sistema não encontre nenhum roteador, fornecendo os parâmetros de rede necessários, como endereços de servidores DNS, NTP, SIP, etc. (RFC 3315, 2003).

2.3 Router Renumbering

Em um caso de mudança de provedor, necessariamente os endereços da rede devem ser renumerados. No IPv6 esse processo de alteração de endereços dos hosts é feito através do mecanismo do protocolo de Neighbor Discovery, recebendo um novo prefixo e anunciando para os hosts. No caso da utilização do DHCPv6, o servidor receberia o prefixo e trataria as configurações, fazendo a reconfiguração dos prefixos nos roteadores e nos hosts. Conhecido como Router Renumbering. (RFC 2894, 2000).

2.4 MTU

O MTU é o limite máximo, no qual o pacote pode trafegar na rede. Caso ocorre de um pacote ser maior que o MTU permitido, esse pacote é fragmentado em pacotes menores, para que se adeque ao MTU máximo.

No protocolo IPv4, essa fragmentação pode ser feita inúmeras vezes, dependendo do MTU do próximo enlace. Porém no IPv6 isso não é permitido. O protocolo Path MTU descobre de forma dinâmica qual o tamanho máximo permitido na rede, fazendo a verificação prévia dos enlaces no caminha até o destino do pacote, e no início do processo, o pacote já é fragmentado.

Esse processo tem o intuito de diminuir os problemas com overhead do calculo dos cabeçalhos, pois são alterados nos roteadores intermediários. (ipv6.br, 2012).

2.5 DNS

O servidor DNS utilizado no IPv6, tem como aspecto ser capaz de armazenar registros quad-A. Independentemente da versão que o servidor opera, o servidor DNS armazena tanto

registros IPv6 quanto registros IPv4. (ipv6.br, 2012).

Com isso o servidor que utiliza apenas IPv4 pode responder a consultas quad-A quanto A, porém as informações obtidas devem ser iguais na consulta IPv6 e IPv4. (ipv6.br, 2012).

2.6 QoS

Inicialmente o protocolo IP trata todos os pacotes da mesma forma, não dando prioridade no momento de envia-los. Porém isso pode gerar diversos problemas, como desempenho, entrega fora de ordem, variação de sinal. Esses problemas podem ocorrer devido a forma de entrega desses pacotes, devido a manipulação sofrida dentro da rede por diferentes interfaces, os roteadores processam os pacotes na ordem em que são recebidos. (ipv6.br, 2012).

Para controlar esse problema, existe o conceito QoS (Quality of Service / Qualidade de Serviço). Que é responsável por prover a transmissão de determinados tráfegos de dados, com prioridade e com garantia de qualidade. (ipv6.br, 2012).

2.6.1 DiffServ

O DiffServ, trabalhar por meio de classes, onde agrega e prioriza pacotes com requisitos QoS similares. Esses pacotes, são identificados pelos oito bits dos campos Tipos de Serviço (IPv4) e Classe de Tráfego (IPv6), identificando e distinguindo as diferentes classes ou prioridades de pacotes. (ipv6.br, 2012).

2.6.2 IntServ

O IntServ, tem como função de trabalho a reserva de recursos por fluxo. Sua utilização normalmente se associa ao protocolo RSVP, no qual reserva recursos ao longo do caminho da fonte até o destino de um fluxo, que requer QoS. (ipv6.br, 2012).

2.7 Mobilidade

A utilização do IPv6 gera uma grande mobilidade para um nó que sai de sua rede de origem. Ao ingressar em uma rede remota, o nó móvel recebe um ou mais endereços remotos, através da autoconfiguração, com o prefixo válido da rede em que se encontra. Para garantir o recebimento de pacotes IPv6, enviados a sua rede de origem, o nó faz uma associação entre o endereço de origem e o endereço remoto, onde se encontra, onde faz o registra de seu novo endereço no agente de origem. Essa associação também pode ser feita diretamente com o nó, otimizando assim a comunicação. (ipv6.br, 2012).

2.8 Endereçamento

O IPv6 tem como um de seus principais características e justificativas o aumento do

espaço de endereçamento. O IPv4, assim como o IPv6 tem um número limitado de endereços, porém por ter 32 bits, somente, o IPv4 se limita a 4.294.967.296 endereços distintos. Enquanto o IPv6, com seus 128 bits, pode obter até 340.282.366.920.938.463.463.374.607.431.768.211.456 endereços. Esse valor, apresenta aproximadamente 79 octilhões de vezes a quantidade de endereços IPv4 e mais de 56 octilhões de endereços por ser humano na Terra, considerando a população estimada em 6 bilhões de habitantes. (ipv6.br, 2012.)

O IPv4 se limita, devido as 3 classes na qual é divido, A, B e C. A classe A, tem um número de hosts muito grande (16.777.216), porém limita o número de redes (128) que podem ser utilizadas. A classe B tem um número menor de hosts (65.536), porém seu número de redes aumenta consideravelmente (16.384), enquanto a classe C, tem um número de hosts muito pequeno (256), porém podem ser geradas diversas redes (2.097.152). Os 32 bits de endereços IPv4, são divididos em 4 grupos de 8 bits cada. Por exemplo, 192.168.0.12. (ipv6.br, 2012.)

Diferentemente do IPv4, o IPv6 apresenta 128 bits, divididos em 8 grupos de 16 bits cada, como são bits hexadecimais os dígitos vão de 0-F, como por exemplo: 2001:0DB8:AD1F:25E2:CADE:CAFE:F0CA:84C1. (ipv6.br, 2012).

Diferentemente do IPv4, onde existe o endereço broadcast, que é responsável por direcionar um pacote para todos os nós de um mesmo domínio, o IPv6 é atribuído tipos específicos de endereços multicast.

2.8.1 Unicast

O endereçamento unicast, foi criado com a intenção de ser utilizado de permitir agregações com prefixos de tamanho flexível. É utilizado para comunicação entre dois nós, por exemplo computadores de uma rede privada. (ipv6.br, 2012).

Existem alguns tipos de endereços unicast, como Global Unicast, Unique Local Adress, Link-Local e alguns tipos de endereços para usos específicos, como endereços IPv4 mapeados em IPv6, endereços de loopback e endereços não-especificados. (ipv6.br, 2012).

O endereço Global Unicast, é o endereço globalmente roteável e acessível na rede IPv6. Possuindo 3 partes: prefixo, utilizado para indicar o tamanho do bloco atribuído a uma rede, a identificação da sub-rede, utilizada para identificar um enlace na rede, e a identificação da interface, que identifica de forma única uma interface no enlace. Como a estrutura é projetada para utilizar 64 bits na identificação da rede e os outros 64 bits para identificação da interface, exceto em casos específicos, todas as sub-redes IPv6 tem o mesmo tamanho de prefixo. (ipv6.br, 2012).

O endereço Link Local é utilizado somente em um enlace especifico onde a interface esta conectada. Sendo o endereço link local atribuído automaticamente utilizando o prefixo

FE80::/64. É utilizado o formato IEEE EUI-64 para identificação da interface e sua configuração. Os roteadores não devem encaminhar pacotes que possuam como origem ou destino o link-local, para outros enlaces. (ipv6.br, 2012).

O endereço Unique Local Adress (ULA) é utilizado apenas para comunicações locais, tendo grande probabilidade de ser utilizado com globalmente único, geralmente dentro de um mesmo enlace ou de um conjunto de enlaces. Um endereço ULA criado utilizando um ID global e alocado pseudo-randomicamente, é composto por prefixo FC00::/7; a flag local (L), tendo o valor 1 (FD) o prefixo é atribuído localmente, tendo valor 0 (FC) o prefixo será atribuído por uma organização central; identificador global, utilizado para criar o prefixo globalmente único indicando 40 bits; identificador de interface, indicador de interface de 64 bits. A estrutura do endereço ULA é FDUU:UUUU:UUUU:: onde U são os bits do identificador único, gerado aleatoriamente por um algoritmo específico. Permitindo que qualquer enlace possua um prefixo /48 privado e único globalmente. Deste modo, caso duas redes distintas, de duas empresas, sejam interligadas elas não entraram em conflito, não necessitando a renumeração de rede. Além disso, o endereço ULA é independente de provedor, podendo ser utilizada na comunicação dentro de um enlace mesmo sem conexão com a internet. O prefixo é facilmente bloqueado, caso um endereço ULA seja anunciado acidentalmente fora do enlace, através de um roteador ou via DNS, não haverá conflito com outros endereços. (ipv6.br, 2012).

O endereço não especificado é representado pelo endereço 0:0:0:0:0:0:0:0 ou ::0. Nunca devendo ser atribuído a nenhum nó, apenas indicando a ausência de um endereço. Pode ser utilizado no campo Endereço de Origem de um pacote IPv6 enviado por um host, no processo de inicialização, antes de ter seu endereço exclusivamente determinado. Também nunca deve ser utilizado como endereço destino. (ipv6.br, 2012).

O endereço loopback é representado pelo endereço unicast 0:0:0:0:0:0:0:1 ou ::1. Utilizado para referenciar a própria máquina, sendo muito utilizado para testes internos. Não deve ser utilizado como endereço de origem em pacotes IPv6, nem atribuído em nenhuma interface física. Um pacote IPv6 com um endereço loopback não pode ser enviado por um roteador e é descartado caso tenha como destino um endereço loopback.

O endereço IPv4-mapeado é representado por 0:0:0:0:0:FFFF:wxyz ou ::FFFF:wxyz. Usado para mapear um endereço IPv4 em um endereço IPv6, onde o wxyz são os 32 bits do endereço IPv4. Algumas faixas de endereços são reservadas para uso específico:

2002::/16 – utilizado na forma de transição 6to4;

2001:0000::/32 – utilizado na forma de transição TEREDO;

2001:db8::/32 – utilizado na representação de endereços IPv6 em formas de textos e em documentações. (ipv6.br, 2012).

2.8.2 Anycast

O endereço IPv6 anycast é utilizado para identificar um grupo de interfaces, porém com a propriedade de que um pacote enviado a um endereço anycast é encaminhado única e exclusivamente a interface do grupo mais próxima da origem do pacote. (RFC 2373, 1998).

Não há diferença sintática entre a faixa de endereços anycast e unicast. Um endereço unicast atribuído a mais de uma interface se transforma em um endereço anycast, se configurando explicitamente os nós para que saibam que lhes foram atribuídos um endereço anycast. Sendo também configurados nos roteadores como uma entrada separada. Utilizando esse esquema de endereçamento, podem ser utilizados para descoberta de serviços de rede, servidores DNS e proxies HTTP, para redundância desses serviços. Pode ser utilizado para balanceamento de carga onde múltiplos hosts ou roteadores provem o mesmo serviço, para localização de roteadores que forneçam acesso a uma determinada sub-rede ou para localização dos Agentes de Origem em redes com suporte a mobilidade IPv6. (ipv6.br, 2012).

2.8.3 Multicast

Multicast é utilizado para identificar grupos de interface, sendo que cada interface pode pertencer a mais de um grupo. Os pacotes são entregues a todas as interfaces que compõe o grupo. (ipv6.br, 2012)**.**

O multicast é uma extensão do protocolo IPv6, que no IPv4 era tido como opcional. Em todos os nós IPv6, o multicast é uma obrigatoriedade, visto que muitas funcionalidades da nova versão do protocolo IP utilizam esse tipo de endereço. Sua funcionalidade lembra a funcionalidade do broadcast, porém ao invés de enviar o pacote para todos os hosts da rede, ele é enviado apenas para um grupo de hosts. (ipv6.br, 2012)**.**

Os endereços multicast não devem ser utilizados como endereço de origem de um pacote. Os endereços se derivam do bloco FF00::/8, onde o FF, que indica o endereço multicast, é precedido por 4 bits, que representam 4 flags, e um valor de 4 bits que define o escopo do grupo. Os 112 bits restantes são utilizados para identificar o grupo multicast.

Flag R: Se o valor for 1, indica que o endereço “carrega” o endereço de um Ponto de Encontro. Se o valor for 0, não há endereço de Ponto de Encontro embutido;

Flag P: Se o valor for 1, indico que o multicast é baseado em um prefixo de rede. Se o valor for 1, indica que não é baseado em um prefixo de rede;

Flag T: Se o valor for 0, indica que o multicast é permanente, ou seja, é atribuído pela IANA. Se o valor for 1, indo que o multicast não é permanente, ou seja, é atribuído dinamicamente. (ipv6.br, 2012).

Os quatro bits que representam o escopo do endereço multicast, são utilizados para delimitar a área de abrangência de um grupo multicast. Os valores atribuídos a esse campo são o seguinte:

1 - abrange apenas a interface local;

2 - abrange os nós de um enlace;

3 - abrange os nós de uma sub-rede;

4 - abrange a menor área que pode ser configurada manualmente;

5 - abrange os nós de um site;

8 - abrange vários sites de uma mesma organização;

E - abrange toda a Internet;

0, F - reservados;

6, 7, 9, A, B, C, D - não estão alocados.

Desta maneira, um roteador ligado a um backbone da internet não encaminhará pacotes com escopo menos que 14 (em hexadecimal). No IPv4, o escopo de um grupo multicast é especificado através do campo TTL do cabeçalho. (ipv6.br, 2012)**.**

Endereço	Escopo	Descrição
FF01::1	Interface	Todas as interfaces (*all-nodes*)
FF01::2	Interface	Todos os roteadores (*all-routers*)
FF02::1	Enlace	Todos os nós (*all-nodes*)
FF02::2	Enlace	Todos os roteadores (*all-routers*)
FF02::5	Enlace	Roteadores OSFP
FF02::6	Enlace	Roteadores OSPF designados
FF02::9	Enlace	Roteadores RIP
FF02::D	Enlace	Roteadores PIM
FF02::1:2	Enlace	Agentes DHCP
FF02::1:FFXX:XXXX	Enlace	*Solicited-node*
FF05::2	Site	Todos os roteadores (*all-routers*)
FF05::1:3	Site	Servidores DHCP em um site
FF05::1:4	Site	Agentes DHCP em um site
FF0X::101	Variado	NTP (*Network Time Protocol*)

Imagem 13: Endereços Multicast Permanente. Fonte: ipv6.br.

O endereço multicast solicited-node identifica um grupo multicast onde todos os nós fazem parte, assim que um endereço unicast ou anycast lhe é atribuído. O endereço solicited-node é formado do prefixo FF02::1::FF00:0000/104 agregando os 24 bits mais a direita do identificador da interface. Existindo um endereço multicast solicited-node correspondente para cada endereço unicast ou anycast. (ipv6.br, 2012)**.**

Para resolver o mac da interface, o endereço solicited-node utiliza o protocolo de Descoberta de Vizinhança, no qual envia uma mensagem Neighbor Solicitation para o endereço solicited-node. Com isso, apenas as interfaces registradas no grupo examinam o pacote. Enquanto em uma rede IPv4 se é enviado uma mensagem ARP Request para o endereço broadcast da camada de enlace, onde todas as interfaces do enlace examinam a mensagem. (ipv6.br, 2012)**.**

3 Formas de Transição

O IPv6 não é uma extensão do IPv4, e sim um substituto. Porém existem maneiras para que o IPv6 e o IPv4, funcionem em paralelo, esse cenário é chamado de pilha dupla, ou dual stack. (H. Soliman, RFC 4977).

Enquanto o IPv6 é implantado, é necessário que seja utilizada técnicas auxiliares de transição, para interconexão do IPv6 com uma internet majoritariamente IPv4, e mesmo ai fim dessa transição dos protocolos, ainda seria necessário a utilização dessas técnicas, devido a alguns poucos que ainda utilizem IPv4.

Porém a implantação do IPv6 não vem sendo tão simples de ser executada. Atualmente o IPv6 não esta sendo amplamente utilizado na internet e o esgotamento do IPv4 já se tornou uma realidade. Com a internet sempre em uma crescente os usuários ainda necessitam de IPv4, mas sem endereços IPv4 para atende-los. Com isso novas técnicas foram e vem sendo desenvolvidas.

As formas hoje encontradas para essa transição são:

Pilha Dupla: consiste na convivência do IPv4 e IPv6 nos mesmos equipamentos, simultaneamente.

Túneis: Permitem que redes IPv4 se comuniquem através de uma rede IPv6, ou o contrario.

Tradução: É feita a conversão dos pacotes, para que equipamentos que utilizam IPv6, se comuniquem com outros que usam IPv4.

Essas técnicas também podem ser subdivididas em stateful e stateless. Onde a técnica stateful utiliza a tabela de estado com informações dos endereços ou pacotes e tem um custo mais alto, pois gastam mais CPU e memória, para que sejam processados, enquanto a stateless não necessita, cada pacote é tratado de forma independente. (ipv6.br, 2012).

3.1 Pilha Dupla

Diante da pouca utilização do IPv6, não é aconselhável ter suporte apenas para essa versão do protocolo IP, tendo em vista que ainda hoje a maior parte da internet utiliza a versão antiga do protocolo IP. Esta técnica é conhecida como pilha dupla (Dual Stack). Esse método permite que dispositivos e roteadores sejam equipados com pilhas de ambos os protocolos, se capacitando para receber e enviar pacotes de ambas as versões.

Apesar de ser a principal forma, e a mais recomendada, até o fim da transição dos protocolos, a pilha dupla não é possível de ser feita em todas as ocasiões. Em caso de necessidade de um provedor atender seus novos usuários com IPv6 e IPv4 e não há mais IPv4 disponíveis. Quando se existem situações onde não há suporte de IPv6 no equipamento, e o equipamento não é de fácil substituição, é onde entram outras técnicas de transição.

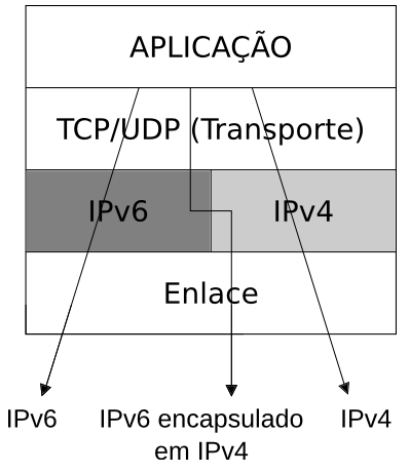

Imagem 14: Pilha Dupla (Dual Stack). Fonte: ipv6.br.

Se deve ser considerado alguns aspectos referentes a infraestrutura para se implementar a pilha dupla. O serviço DNS, por exemplo, deve fazer a configuração dos endereços IPv6, já que no IPv6 se utilizada os registros AAAA (quad-A), onde são armazenas seus endereços. O servidor DNS faz a resposta aos endereços IPv6, mesmo que trabalhe somente com IPv4, que é decido pela aplicação qual protocolo usar, sendo que a preferencia sempre será do IPv6, caso a resposta seja negativa a aplicação faz a consulta ao IPv4.

O happy eyeballs é utilizado para corrigir o problema encontrado na decisão sobre qual conexão priorizar, em caso das duas pontos possuírem IPv4 e IPv6. Seu funcionamento consiste em se conectar as duas conexões ao mesmo tempo e utilizar a que for estabelecida com mais rapidez, dando sempre uma leve preferencia para a conexão IPv6. (Wing & Yourtchenko, RFC 6555).

Em uma rede de pilha dupla, a configuração de roteamento IPv6 é feita independente da de IPv4, devido a antes da implementação do IPv6 somente era utilizado o protocolo de roteamento interno OSPFv2, tendo suporte apenas ao IPv4. Nesse caso é necessário a migração para um protocolo de roteamento que tenha suporte para ambos os protocolos, ou forçar a execução do OSPFv3, suporte para IPv6, para funcionarem paralelamente.

A filtragem de pacotes é diretamente dependente de qual plataforma esta sendo utilizada. Em um caso de utilização Linux, a filtragem de pacotes é feita totalmente independente uma das outras, onde o ip4tables faz a filtragem dos pacotes IPv4 e o ip6tables a filtragem dos pacotes IPv6. (ipv6.br, 2012).

3.2 Túneis 6in4

O túnel 6in4 consiste em encapsular o pacote IPv6 dentro de um pacote IPv4, ajustar os endereços de origem e destino e colocar no cabeçalho o tipo 41. Esse encapsulamento é conhecido como “protocolo 41”.

Assim que o destino recebe o tipo 41, ele retira o cabeçalho IPv4 e trata o pacote como

IPv6. Pode também ser feito o inverso, encapsular pacotes IPv4 para o envio como IPv6, porém é melhor aplicado em outros técnicas que serão apresentadas.

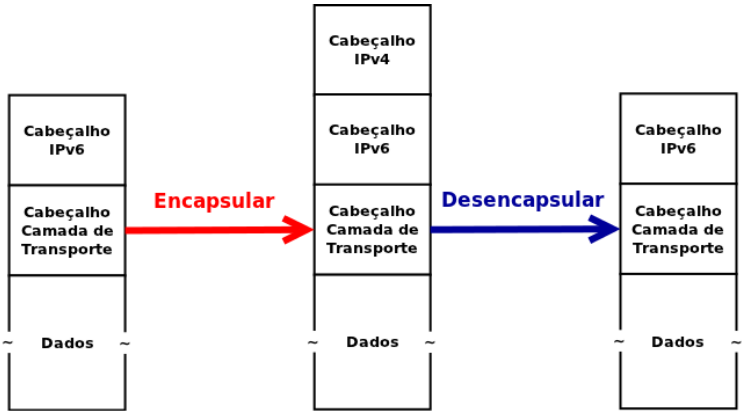

Imagem 15: Funcionamento Túnel 6in4. Fonte: ipv6.br.

Diversos cenários podem fazer esse tipo de transição, um deles é o Router-to-router. O cenário router-to-router, são interligados roteadores IPv6/IPv4 por uma infraestrutura IPv4 onde podem fazer o tunelamento IPv6 entre eles. Nesse cenários o túnel traça um caminho de extremidade a extremidade, que é a trajetória que o pacote IPv6 faz. (Erik Nordmark & Robert E. Gilligan, RFC 4213).

Host-to-router, onde hosts IPv6/IPv4 fazem o tunelamento IPv6 até um roteador intermediário, que é acessível através de uma infraestrutura IPv4. Esse tipo de túnel abrange o primeiro segmento de caminho de pacote end-to-end. (Erik Nordmark & Robert E. Gilligan, RFC 4213).

Host-to-host, onde dois hosts IPv6/IPv4 estão interligados por uma infraestrutura IPv4. O túnel se estende por todo o caminho que o pacote leva. (Erik Nordmark & Robert E. Gilligan, RFC 4213).

Router-to-host, onde um roteador IPv6/IPv4 esta interligado com o host final IPv6/IPv4.

Apesar de alguns tipos de cenários, o mais comum é o router-to-router, onde é feita a conexão entre 2 equipamentos de distribuição. (Erik Nordmark & Robert E. Gilligan, RFC 4213).

3.3 Túnel GRE

O túnel GRE foi desenvolvido pela equipe de engenharia da Cisco, com a finalidade de encapsular diversos tipos de protocolos.

Um túnel GRE é uma interface lógica onde fornece uma maneira de encapsular pacotes de passageiro dentro de um protocolo de transporte. Proporciona um serviço a fim de executar o encapsulamento Point-to-Point. (cisco.com, 2013)

Os pacotes que serão transmitidos através do túnel sofrem além do encapsulamento comum, feito pelo cabeçalho referente aos protocolos comuns da rede, é encapsulado também

por um cabeçalho GRE, em seguida é encaminhado pelo túnel até o endereço do túnel, onde o cabeçalho GRE é retirado e o pacote segue o caminho até seu destino, designado pelo IP original do pacote. Existem implementações onde é permitido túneis de um ponto com diferentes pontos de destino, porém é mais comum o seu uso no ponto a ponto. (cisco.com, 2013)

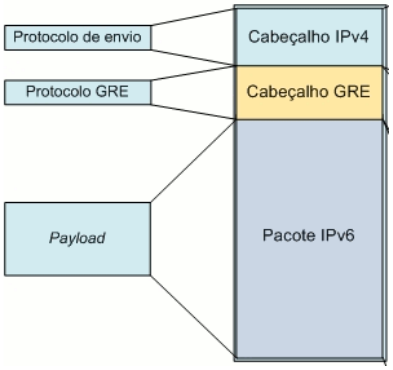

Imagem 16: Pacote com cabeçalho GRE. Fonte: ipv6.br.

Segurança em uma rede usando GRE deve ser relativamente semelhante a segurança em uma rede IPv4 normal, como roteamento usando GRE segue o mesmo encaminhamento que o IPv4 usa nativamente. A filtragem de rotas permanecerá inalterada. No entanto a filtragem de pacotes exige que um firewall olhe dentro do pacote GRE ou que a filtragem seja feita na extremidade do túnel. Nesses ambientes em que é considerado ter um problema de segurança, é desejável que o túnel seja encerrado em um firewall. (Dino Farinacci, RFC 2784).

3.4 6rd: Configuração de Relay e CPE (/64)

O 6rd permite que seja feita a implantação rápida do IPv6 até o usuário final. Existindo dois elementos, o Relay e o CPE. O CPE 6rd funciona exatamente da mesma forma que o CPE IPv4, atribuindo um endereço ao usuário, mas atribui também o IPv6, formado através do IPv4, já atribuído, e um prefixo que é fornecido pelo provedor. Enquanto o Relay 6rd se encontra no servidor de distribuição, e possui conectividade nativa IPv6 e IPv4, esse método utiliza o encapsulamento 6in4. (ipv6.br, 2012 & Remi Despres, RFC 5569).

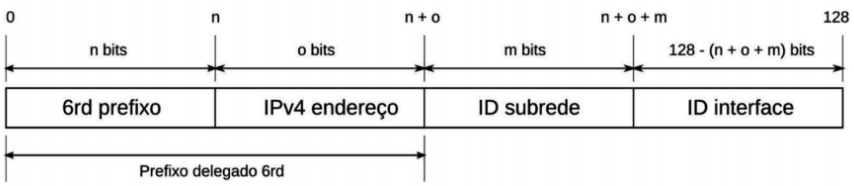

Imagem 17: Construção do endereço IPv6 para cliente 6rd (/64). Fonte: ipv6.br

Essa técnica não resolve o problema de escassez de endereços IPv4. A técnica só se pode ser utilizada quando se há disposição de endereços IPv4, caso contrário não se pode utilizar.

Essa técnica é utilizada em servidores dos quais não irão sofrer problemas de esgotamento de endereços em curto/médio prazo.

Tendo o ISP para trazer o IPv6 rapidamente, com a utilização do IPv4 e sem custo adicional, é uma maneira de quebrar o ciclo vicioso que vem atrasando a implementação do IPv6, ISPs esperam a demanda de clientes para implantar o IPv6, os clientes não exigem IPv6 enquanto os fornecedores ainda fornecerem infraestrutura IPv4, é investido as aplicações em compatibilidade de NAT Transversal, na qual alonga ainda mais a implementação do IPv6.

O 6rd tem por principio utilizar o 6to4 como base, melhorando e suprindo suas limitações. Primeiramente são modificadas as funções 6to4 para substituir o prefixo padrão 2002 :: /16 por um prefixo IPv6, que pertence ao espaço de endereço do ISP atribuído, e substituir o endereço anycast 6to4 por um endereço anycast escolhido pelo ISP. O ISP opera um ou mais gateways 6rd em sua infraestrutura entre a internet IPv4 e IPv6. CPEs suportam IPv6 no lado de seus clientes e suportam 6rd no lado dos provedores. (Remi Despres, RFC 5569).

6rd-relays Prefixo IPv6 do ISP	Endereço IPv4 do cliente
<- Menor ou Igual a 32 ->	<- 32 ->
<- Menor ou Igual a 64 ->	

Imagem 18: Formato do prefixo IPv6 atribuído ao 6rd do cliente.

4 Softwares para Simulação

Para o desenvolvimento do projeto, foram utilizados os softwares, Common Open Research Emulator (Core), o Iperf software para efetuar os testes de envio de pacotes e o software Wireshark 2.2.1 para a captura dos protocolos e analise do trafego de rede.

4.1 Common Open Research Emulator (Core)

O Core é uma ferramenta utilizada para simulação de rede em uma ou mais máquinas. O software oferece a possibilidade de fazer a emulação de uma rede e juntar essa emulação a uma rede real, já existente.

O Core pode oferece a possibilidade gráfica para o desenho e desenvolvimento de topologias de máquinas virtuais e o módulo Python, para a emulação de scripts desenvolvidos pelo próprio usuário ou scripts padrões, para funcionalidades do programa, como firewall, roteamente etc.

O software pode ser baixado no link http://downloads.pf.itd.nrl.navy.mil/core/, ou podem ser seguidos os seguintes passos para que seja instalado no linux.

Primeiramente é necessária a instalação dos pré-requisitos do software, possuir o Ubuntu 12.04 ou 14.04 ou superiores.

Utilizando a linha de comando pelo terminal “sudo apt-get install core-network”, será iniciado o download e instalação do Core.

Após o download e instalação do Core é necessária a instalação dos pacotes pré requisitados pelo sofware, para que todas suas funcionalidades sejam ativadas.

Através dos seguintes comandos será feita o download dos pacotes.

```
#sudo apt-get update
#sudo apt-get dist-upgrade
#sudo apt-get install bash bridge-utils ebtables iproute libev-dev python tcl8.5 tk8.5 libtk-img
```

Para o roteamento é necessária a instalação do Quagga.

```
#sudo apt-get install quagga
```

Após esses passos o software está pronto para utilização, o software após a instalação tem a seguinte forma:

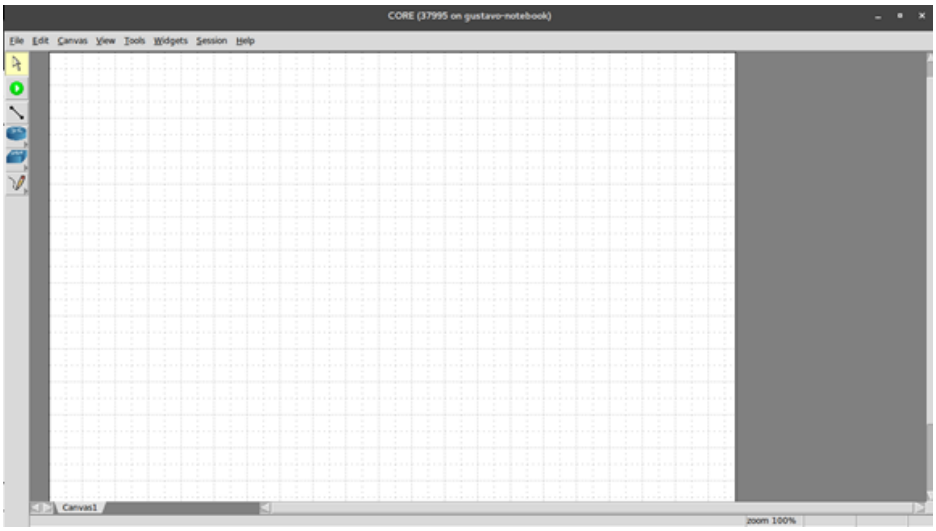

Imagem 19: Página inicial do CORE.

Para a criação de hosts, roteadores ou servidores, devemos ir até o menu lateral, do lado esquerdo, na quarta opção, onde todas essas opções se encontram.

Imagem 20: Ícone para criação de hosts, roteadores e servidores.

Após clicar no ícone no menu lateral, serão abertas as seguintes opções.

Imagem 21: Opções para criação no ícone Network-Layer Virtual Nodes

A opção Router, é onde os roteadores são criados; a opção Host é onde os servidores são criados, porém também podem fazer a função de computadores, devido à possibilidade de serem alteradas suas configurações; e a opção PC, é onde são criados os computadores da rede, segundo o próprio site do software, essa opção é basicamente para ser utilizada em simulação de rede sem fio, podendo fazer alteração do Host para computador.

Para a criação dos switches é necessário que clique na opção Link-Layer Nodes.

Imagem 22: Ícone para criação de switch.

Para criar o switch da rede, é necessário clicar na opção Ethernet Switch, que será a segunda opção.

Imagem 23: Criar switch na opção Link-Layer Nodes.

Para fazer a comunicação dos equipamentos, Link Tool esteja habilitado.

Imagem 24: Ícone para criação de comunicação entre equipamentos.

Após habilitar essa opção, é necessário somente clicar em qual equipamento irá se comunicar, arrastar até o outro equipamento que irá se comunicar e soltar.

Também é necessário que cada equipamento tenha suas configurações feitas corretamente, devido a diversas funcionalidades.

Quando se for criar Host, com função de computador, será necessário que algumas coisas sejam alteradas para que funcionasse a comunicação. Para alterar essas configurações é necessário clicar com o botão direito no host e clicar em Configure, irão se apresentar as seguintes opções:

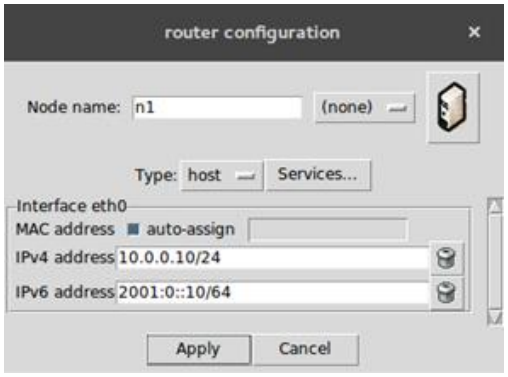

Imagem 25: Configuração Host.

É possível verificar que o "n1" é o nome que foi dado ao host e que apresenta seus endereços IPv4 e IPv6 já associados automaticamente, assim que for feita a comunicação entre os equipamentos do cenário.

Após essa opção deve ser aberta a opção Services, para que seja feita a alteração dos serviços.

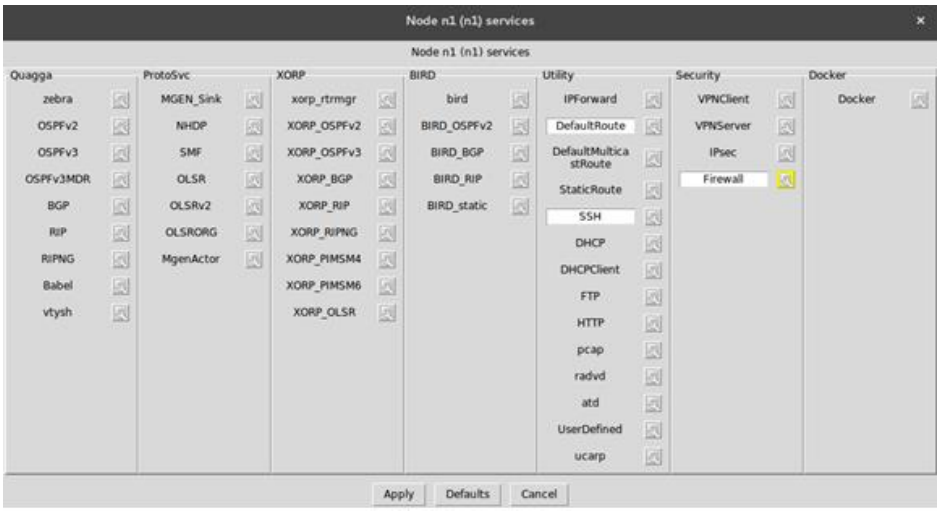

Imagem 26: Serviços Host.

Todos os equipamentos, exceto o switch, terão as mesmas opções e devem ter algumas delas alteradas, para que o cenário funcione.

Como o Host é utilizado como servidor, a opção Firewall está habilitada, é necessário

que seja desabilitada, somente clicando nela e logo após deve ser aplicada na opção Apply.

É necessário também que seja feito esse procedimento com o roteador, pois ele necessita que algumas configurações sejam habilitadas.

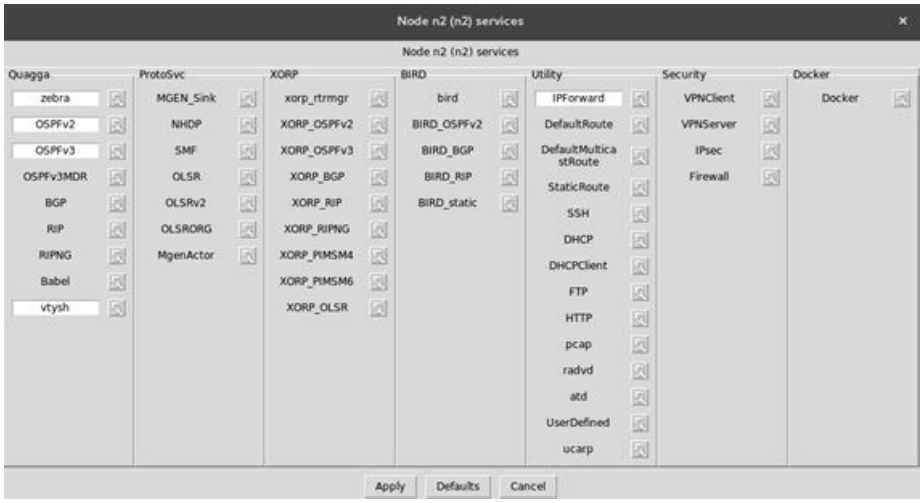

Imagem 27: Serviços router.

É possível verificar que por padrão, as configurações do roteador habilitadas são diferentes do Host, porém também devem ser habilitadas duas outras opções, as opções RIP e RIPNG na coluna QUAGGA, essa opções devem ser habilitadas, para que a comunicação dos roteadores e dos hosts possa ser feita, caso contrário não será possível fazer o envio de pacotes entre todos os hosts.

É necessário também que seja feita a configuração dos hosts através de linha de comando, para isso, será necessário o inicio da simulação, para que seja iniciado, basta clicar no ícone Start the Session.

Imagem 28: Iniciar simulação.

Após o inicio da simulação, clicando duas vezes com o botão esquerdo do mouse, sobre o equipamento, se inicia um terminal linux, no qual serão feitas todas as configurações das formas de transição.

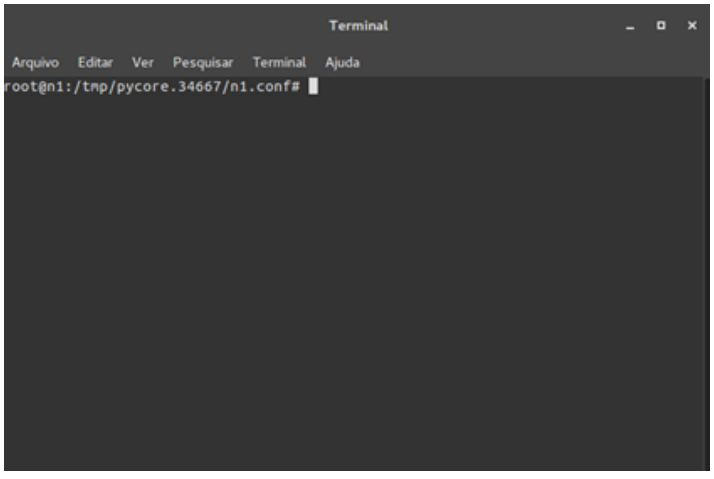

Imagem 29: Terminal Linux dos equipamentos.

Todos os comandos para que possa ser feita a configuração das formas de transição, serão feitos nesse terminal.

Para a iniciação do Wireshark, para que os pacotes sejam captados, é necessário que após a inicialização da simulação, clique com o botão direito sobre o equipamento no qual será feita a captação dos pacotes, selecione a opção Wireshark e clique na eth0, que irá aparecer assim que o mouse seja posicionado sobre a opção Wireshark. O software Wireshark fará sua inicialização automaticamente.

4.2 Iperf

O Iperf é um software desenvolvido pelo National Laboratory for Applied Network Research (NLANR) com intuito de ser utilizado como Cliente/Servidor. Com isso é possível efetuar teste de desempenho, fazendo o encaminhamento de pacotes, com o tamanho configurado, durante um tempo que seja configurado, com os parâmetros que sejam solicitados.

A instalação pode ser feita através do site https://iperf.fr/iperf-download.php, onde se encontra o download para Windows. Ou no caso de utilização do Linux, se pode fazer a instalação através dos seguintes comandos.

Ubuntu 64 bits / Debian 64 bits / Mint 64 bits:

```
#sudo apt-get remove iperf3 libiperf0
#wget https://iperf.fr/download/ubuntu/libiperf0_3.1.3-1_amd64.deb
#wget https://iperf.fr/download/ubuntu/iperf3_3.1.3-1_amd64.deb
#sudo dpkg -i libiperf0_3.1.3-1_amd64.deb iperf3_3.1.3-1_amd64.deb
#rm libiperf0_3.1.3-1_amd64.deb iperf3_3.1.3-1_amd64.deb
```

Ubuntu 32 bits / Debian 32 bits / Mint 32 bits:

```
#sudo apt-get remove iperf3 libiperf0
#wget https://iperf.fr/download/ubuntu/libiperf0_3.1.3-1_i386.deb
#wget https://iperf.fr/download/ubuntu/iperf3_3.1.3-1_i386.deb
#sudo dpkg -i libiperf0_3.1.3-1_i386.deb iperf3_3.1.3-1_i386.deb
#rm libiperf0_3.1.3-1_i386.deb iperf3_3.1.3-1_i386.deb
```

O Iperf possuí uma vasta tabela, para configuração dos pacotes que serão encaminhados, dentre essas todas opções as que serão utilizadas nesse projeto, são as seguintes:

-s : que indica que o host é o servidor;

-c : que indica que o host é um cliente;

-V : que indica que os pacotes enviados são IPv6;

-i : que indica intervalo de tempo em segundos entre a largura da banda periódica;

-t : o tempo em segundos para transmitir os pacotes;

-f : para o envio de um arquivo real na rede simulada.

Após a instalação do Iperf, para ser utilizado na simulação do CORE, somente é necessário que seja inserido o comando de iniciação do Iperf, indicando se o host é um servidor ou um host.

#iperf -s (inicia o host como servidor)

#iperf -c “ip do host servidor” (inicia o host como cliente)

4.3 Wireshark

O Wireshark é um software utilizado para a analise do tráfego de rede, fazendo a organização dos pacotes e gerando gráficos comparativos.

Para efetuar a instalação do programa na plataforma Windows, deve-se acessar o link https://www.wireshark.org/download.html, verificar qual a versão do Windows utilizado e fazer a instalação.

Em caso de utilização do Linux, deve se seguir as seguintes linhas de comando:

#sudo apt-get -y install wireshark wireshark-common wireshark-dev

Logo após o software pode ser aberto sem problema.

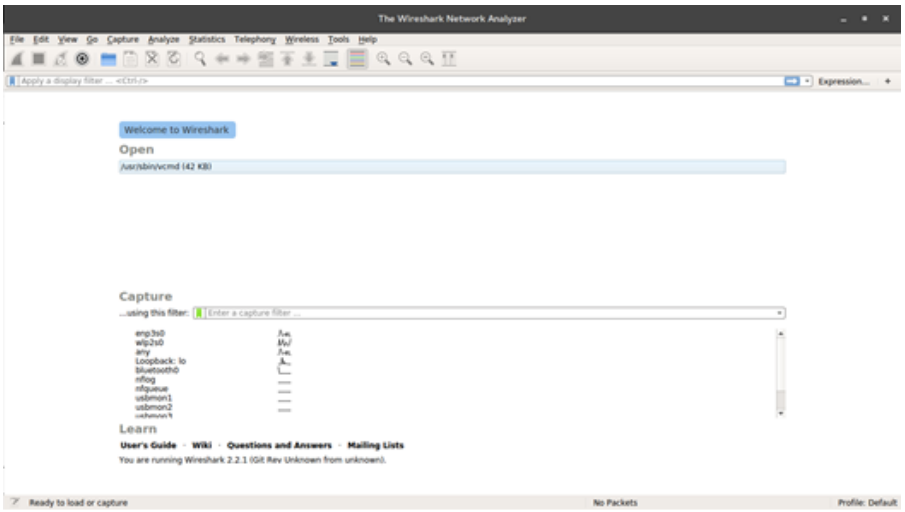

Imagem 30: Tela inicial do Wireshark.

Para a inicialização do Wireshark para capturar os pacotes no CORE, se deve abrir o Wireshark diretamente após a simulação do cenário, clicando com o botão direito sobre o host que será feita a analise de pacotes e clicando na opção wireshark > eth0, após isso será iniciado o programa e apresentará a analise dos pacotes já sendo enviados.

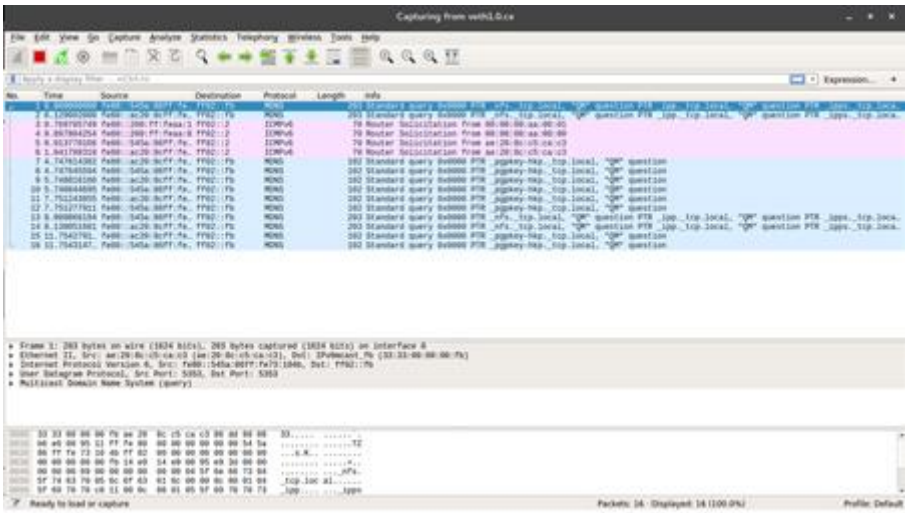

Imagem 31: Wireshark analisando pacotes do host.

Para gerar os gráficos nos quais foram feitas comparações do projeto, se deve acessar o menu superior Statistics > I/O Graph. Automaticamente será feita a analise em forma gráfica dos pacotes analisados, é possível fazer a alteração dessas verificações de acordo com o que será analisado, por padrão a configuração são dos pacotes enviados e dos pacotes que ocorreram algum erro de envio.

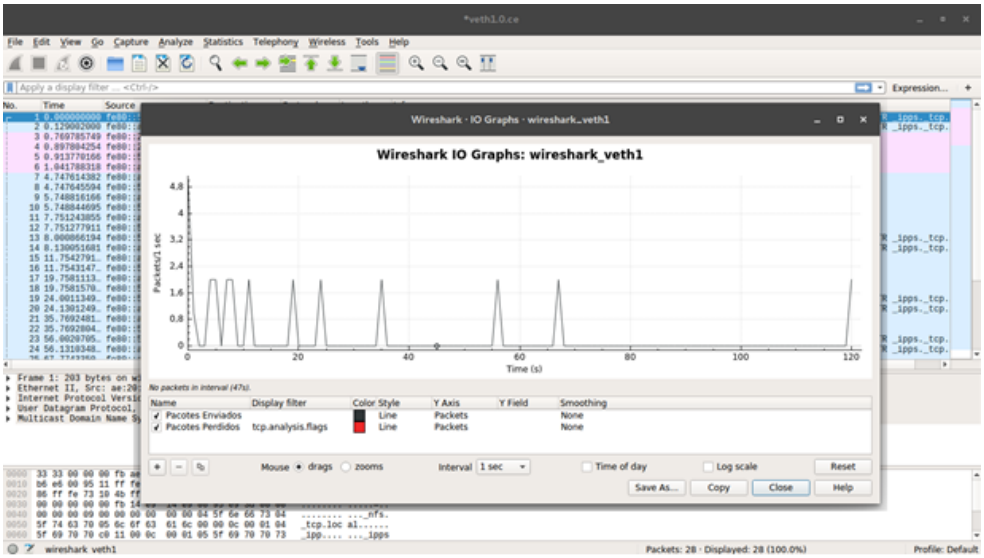

Imagem 32: Gerar gráfico no wireshark.

5 Implementação

A implementação foi divida em 4 partes, primeiro foi feita a alteração de cenário necessária, para que seja feita a implantação da forma de transição proposta. A segunda parte são feitas as configurações necessárias para a troca de pacotes nas formas de transição. A última parte é a captação e analise dos resultados através do Wireshark.

5.1 Pilha Dupla

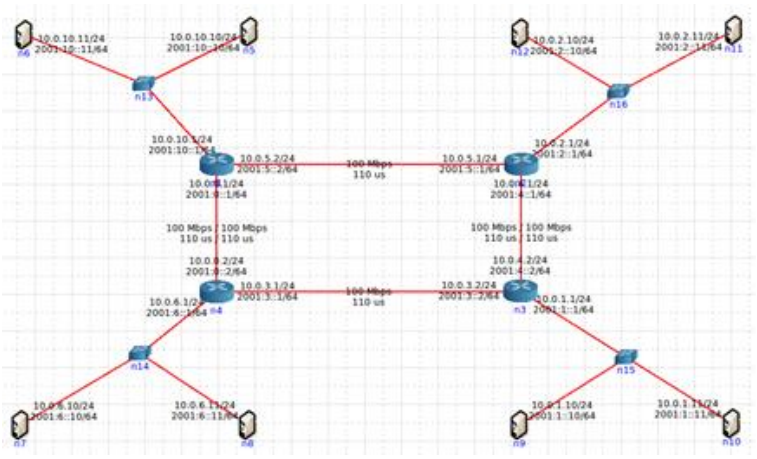

Imagem 33: Cenário de implementação.

Na imagem a cima, podemos verificar que todos os componentes, aceitam os dois protocolos (IPv4 e IPv6), nesse caso, a implementação do IPv6 não teria problema, já que todos os equipamentos trabalham com a dualidade dos protocolos.

5.2 Túneis 6in4

O túnel 6in4 tem como grande diferença, que somente os equipamentos de utilização direta do usuário, computadores, celulares, notebooks ou outros dispositivos, o suporte para o protocolo IPv4 e IPv6, conforme a imagem a seguir.

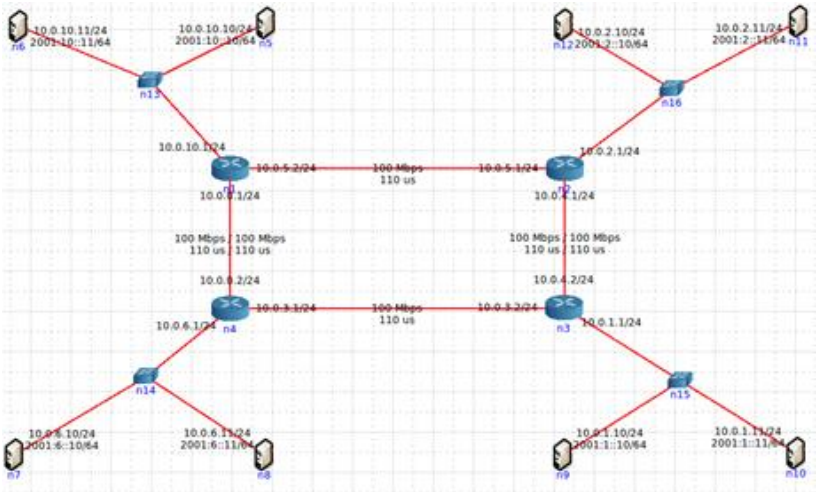

Imagem 34: Ambiente de implementação.

Nesse cenário, é necessária a configuração do túnel 6in4, do host N6 até o host N10.

Lembrando que todos os túneis que forem ser construídos na rede, devem ser configurados nas duas pontas.

```
root@n6:/tmp/pycore.32837/n6.conf# ip addr add 2001:10::11 dev lo
root@n6:/tmp/pycore.32837/n6.conf# ip tunnel add toN10 mode sit ttl 64 remote 10.0.1.11 local 10.0.10.11
root@n6:/tmp/pycore.32837/n6.conf# ip link set dev toN10 up
root@n6:/tmp/pycore.32837/n6.conf# ip -6 route add 2001:1::11 dev toN10
root@n6:/tmp/pycore.32837/n6.conf#
```

Imagem 35: Configuração túnel 6in4 host N6.

A configuração funciona da seguinte maneira, a linha “ip addr add 2001:10::22 dev to” indica de qual endereço o túnel irá partir; na linha “ip tunnel add toN10 mode sit ttl 64 remote 10.0.1.11 local 10.0.10.11” indica qual será o nome do túnel “toN10”, qual será o túnel que será criado “sit”(faz referencia no linux para o túnel 6in4) e quais os endereços IPv4 de destino e local que o túnel irá encapsular os pacotes; na linha “ip link set dev toN10 up” é feita a ativação desse túnel; e na linha “ip -6 route add 2001:1::11 dev toN10” indica qual o endereço IPv6 de destino dos pacotes, onde eles serão desencapsulados.

Dessa mesma forma é feita a configuração no host N10, porém fazendo as devidas alterações.

```
root@n10:/tmp/pycore.32837/n10.conf# ip addr add 2001:1::11 dev lo
root@n10:/tmp/pycore.32837/n10.conf# ip tunnel add toN6 mode sit ttl 64 remote 10.0.10.11 local 10.0.1.11
root@n10:/tmp/pycore.32837/n10.conf# ip link set dev toN6 up
root@n10:/tmp/pycore.32837/n10.conf# ip -6 route add 2001:10::11 dev toN6
root@n10:/tmp/pycore.32837/n10.conf#
```

Imagem 36: Configuração túnel 6in4 host N10.

5.3 Túnel GRE

No túnel GRE é possível verificar que somente os hosts finais da rede, computadores, celulares, notebook, etc, suportam o protocolo IPv4 e IPv6. Os equipamentos intermediários da rede, suportam apenas o protocolo IPv4.

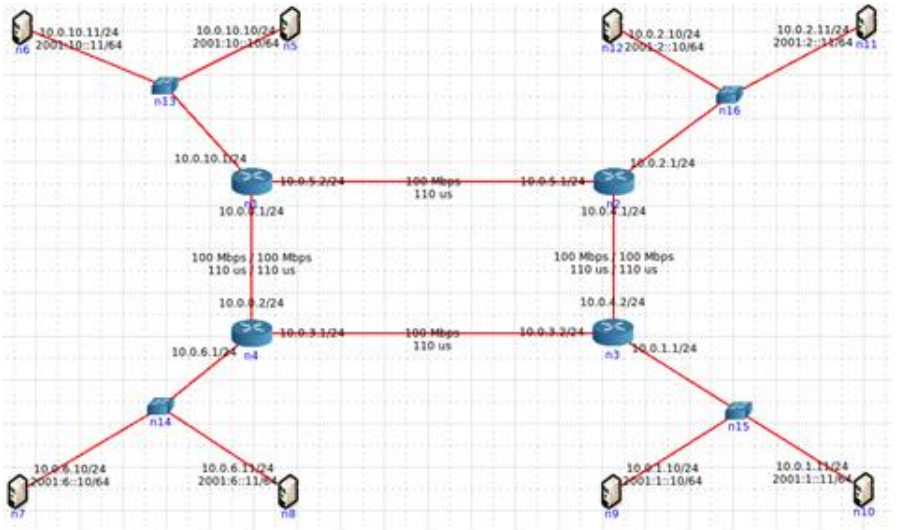

Imagem 37: Ambiente de implementação.

A configuração do túnel GRE é feita entre todos os hosts que irão se comunicar. A configuração deve ser feita da seguinte maneira:

```
root@n7:/tmp/pycore.46280/n7.conf# ip addr add 2001:6::10 dev lo
root@n7:/tmp/pycore.46280/n7.conf# ip tunnel add toN11 mode gre ttl 64 remote 10.0.2.11 local 10.0.6.10
root@n7:/tmp/pycore.46280/n7.conf# ip link set dev toN11 up
root@n7:/tmp/pycore.46280/n7.conf# ip -6 route add 2001:2::11 dev toN11
```

Imagem 38: Configuração túnel GRE host N7.

A configuração funciona da seguinte maneira, a linha “ip addr add 2001:6::10 dev to” indica de qual endereço o túnel irá partir; na linha “ip tunnel add toN11 mode gre ttl 64 remote 10.0.2.11 local 10.0.6.10” indica qual será o nome do túnel “toN11”, qual será o túnel que será criado “GRE” (faz referencia no linux para o túnel GRE) e quais os endereços IPv4 de destino e local que o túnel irá encapsular os pacotes; na linha “ip link set dev toN11 up” é feita a ativação desse túnel; e na linha “ip -6 route add 2001:2::11 dev toN11” indica qual o endereço IPv6 de destino dos pacotes, onde eles serão desencapsulados.

Dessa mesma forma é feita a configuração no host N10, porém fazendo as devidas alterações.

```
root@n11:/tmp/pycore.46280/n11.conf# ip addr add 2001:2::11 dev lo
root@n11:/tmp/pycore.46280/n11.conf# ip tunnel add toN7 mode gre ttl 64 remote 10.0.6.10 local 10.0.2.11
root@n11:/tmp/pycore.46280/n11.conf# ip link set dev toN7 up
root@n11:/tmp/pycore.46280/n11.conf# ip -6 route add 2001:6::10 dev toN7
```

Imagem 39: Configuração túnel GRE host N11.

5.4 6rd: Configuração de Relay e CPE (/64)

No túnel 6rd é possível verificar que todos os hosts finais e intermediários da rede, exceto o servidor, tem suporte apenas para o protocolo IPv4.

Nessa forma de transição é necessário que o provedor forneça um equipamento para o cliente, que suporte essa configuração do túnel e o próprio servidor do cliente tenham algumas configurações feitas para que possa encaminhar os pacotes corretamente.

A grande diferença nesse caso, é que o provedor tem a necessidade de fazer uma configuração de roteamento para cada cliente, devido a necessidade de configuração das duas pontas do túnel.

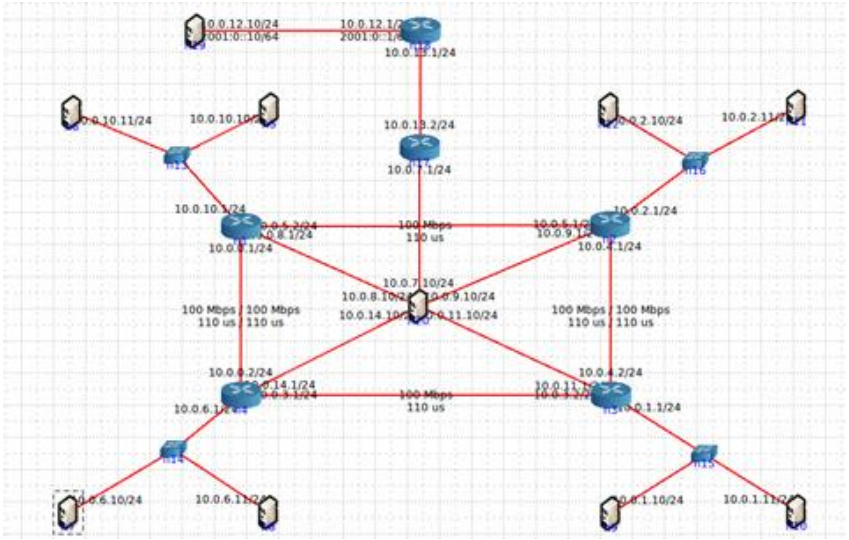

Imagem 40: Ambiente de implementação.

A configuração do túnel 6rd é feita, primeiramente no equipamento de distribuição do

provedor de endereço IPv6, verificando na figura acima que o equipamento recebe IPv4 e IPv6 e somente encaminha IPv4 para o próximo equipamento.

A configuração do equipamento do servidor é feita da seguinte forma:

```
root@n18:/tmp/pycore.43882/n18.conf# ip tunnel add toClient mode sit local 10.0.13.1 ttl 64
root@n18:/tmp/pycore.43882/n18.conf# ip tunnel 6rd dev toClient 6rd-prefix 2001:db8::/32
root@n18:/tmp/pycore.43882/n18.conf# ip link set toClient up
root@n18:/tmp/pycore.43882/n18.conf# ip -6 addr add 2001:db8::1/128 dev toClient
root@n18:/tmp/pycore.43882/n18.conf# ip -6 route add 2001:db8::/32 dev toClient
root@n18:/tmp/pycore.43882/n18.conf# ip -6 route add 2000::/3 dev eth1
```

Imagem 41: Configuração equipamento do provedor.

A configuração funciona da seguinte maneira, a linha “ip tunnel toClient mode sit local 10.0.13.1 ttl 64” indica que esta sendo criado túnel com destino ao cliente, esse túnel utiliza a forma de transição 6in4 (sit) e está se iniciando no IPv4 10.0.13.1; na linha “ip tunnel 6rd dev toClient 6rd-prefix 2001:db8::/32” indica que o túnel 6rd recebe o prefixo de rede 2001:db8:: para como endereço local IPv6; na linha “ip link set toClient up” é feita a ativação desse túnel; na linha “ip -6 add 2001:db8::1/128 dev toClient” indica qual o endereço IPv6 local do inicio do túnel; na linha “ip -6 route add 2001:db8::/32 dev toClient” é criado o roteamento até o cliente; na linha “ip -6 route add 2000::/3 dev eth1” é indicada a interface que a comunicação será feita.

O próximo passo é a configuração do equipamento destino do túnel.

```
root@n17:/tmp/pycore.43882/n17.conf# ip -6 addr add 2001:db8:cb00:7182::/64 dev eth0
root@n17:/tmp/pycore.43882/n17.conf# ip tunnel add toInternet mode sit local 10.0.13.2 ttl 64
root@n17:/tmp/pycore.43882/n17.conf# ip tunnel 6rd dev toInternet 6rd-prefix 2001:db8::/32
root@n17:/tmp/pycore.43882/n17.conf# ip link set toInternet up
root@n17:/tmp/pycore.43882/n17.conf# ip -6 addr add 2001:db8:cb00:7182::1/128 dev toInternet
root@n17:/tmp/pycore.43882/n17.conf# ip -6 route add ::/96 dev toInternet
root@n17:/tmp/pycore.43882/n17.conf# ip -6 route add 2000::/3 via ::10.0.13.1
```

Imagem 42: Configuração equipamento destino do túnel.

A configuração funciona da seguinte maneira, a linha “ip -6 addr add 2001:db8:cb00:7182::/64 dev eth0” indica que a interface de rede tem como endereço IPv6 2001:db8:cb00:: e que esta ligado na interface 0; na linha “ip tunnel add toInternet mode sit local 10.0.13.2 ttl 64” indica que o túnel criado tem o nome “toInternet” e utiliza a forma de transição 6in4 (sit) com endereço local 10.0.13.2 IPv4; na linha “ip tunnel 6rd dev toInternet 6rd-prefix 2001:db8::/32” indica o prefixo desse dessa ponto do túnel; na linha “ip link set toInternet up” é feita a ativação desse túnel; na linha “ip -6 addr add 2001:db8:cb00:7182:1/128 dev toInternet” indica qual o endereço IPv6 que o equipamento está recebendo, para recebimento dos pacotes; na linha “ip -6 route add ::/96 dev toInternet” é criado o roteamento do cliente; na linha “ip -6 route add 2000::/3 via ::10.0.13.1” indica de onde o inicio do túnel vem.

Após a configuração dos equipamentos de borda, agora é necessário ser feita a configuração do servidor, para que todos os pacotes sejam encaminhados para o equipamento na qual esta feita a criação do túnel.

```
root@n20:/tmp/pycore.43882/n20.conf# ip -6 addr add 2001:db8:cb00:7182::b1c0/64 dev eth0
root@n20:/tmp/pycore.43882/n20.conf# ip -6 route add default via 2001:db8:cb00:7182::
```

Imagem 43: Configuração equipamento do cliente.

É necessário também a configuração do servidor do cliente, pois deve encaminhar os

pacotes IPv6 para o túnel criado.

Na linha “ip -6 addr add 2001:db8:cb00:7182::b1c0/64 dev eth0” é indicado o endereço do servidor, na qual irá se comunicar com o equipamento onde esta configurado o túnel; na linha “ip -6 route add defaul via 2001:db8:cb00:7182::” indica cria a rota na qual o endereço de rede IPv6 foi criado.

5.5 Resultados

Após as implementações, os resultados gerados são a Latência da rede, medido pelo ping. É feita a verificação das rotas, por onde os pacotes estão passando. É feita uma medição de pacotes enviados e perdidos, quando toda é exposta a seu limite de envio, com todos os hosts se comunicando, enviando e recebendo pacotes e uma medição de pacotes enviados e perdidos apenas por um dos hosts enviando um arquivo real.

A implementação foi feita em uma máquina Core i3 clock de 2,70 GHz, 8 gigas de memória, 500 giga de HD, com sistema operacional Linux Ubuntu 15.04.

5.5.1 Ping

O ping é utilizado para fazer a verificação dos caminhos que o pacote percorre, até chegar ao seu destino, é importante saber o caminho, para que possamos ter certeza se o tunelamento foi feita corretamente e esta passando pelos devidos caminhos.

A latência da rede na utilização da pilha dupla tem em média um tempo de resposta de 0.615 ms. E reconhece todos os equipamentos por onde o pacote passa, conforme as imagens abaixo.

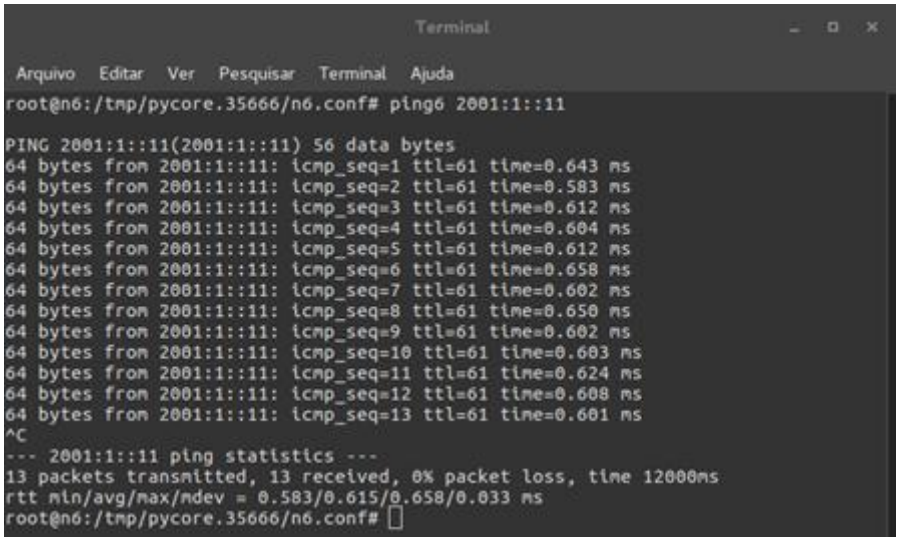

Imagem 44: Latência da implementação Pilha Dupla.

É possível verificar que o teste de latência da rede utilizando o túnel 6in4, apresenta uma média de tempo de resposta de 0.669 ms, levando em consideração a ideal, um tempo de resposta um pouco mais lento.

```
root@n6:/tmp/pycore.32837/n6.conf# ping6 2001:1::11
PING 2001:1::11(2001:1::11) 56 data bytes
64 bytes from 2001:1::11: icmp_seq=1 ttl=64 time=0.615 ms
64 bytes from 2001:1::11: icmp_seq=2 ttl=64 time=0.653 ms
64 bytes from 2001:1::11: icmp_seq=3 ttl=64 time=0.652 ms
64 bytes from 2001:1::11: icmp_seq=4 ttl=64 time=0.661 ms
64 bytes from 2001:1::11: icmp_seq=5 ttl=64 time=0.656 ms
64 bytes from 2001:1::11: icmp_seq=6 ttl=64 time=0.709 ms
64 bytes from 2001:1::11: icmp_seq=7 ttl=64 time=0.653 ms
64 bytes from 2001:1::11: icmp_seq=8 ttl=64 time=0.728 ms
64 bytes from 2001:1::11: icmp_seq=9 ttl=64 time=0.634 ms
64 bytes from 2001:1::11: icmp_seq=10 ttl=64 time=0.714 ms
64 bytes from 2001:1::11: icmp_seq=11 ttl=64 time=0.697 ms
64 bytes from 2001:1::11: icmp_seq=12 ttl=64 time=0.656 ms
^C
--- 2001:1::11 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 10997ms
rtt min/avg/max/mdev = 0.615/0.669/0.728/0.033 ms
```

Imagem 45: Latência da implementação Túnel 6in4.

É possível verificar que o teste de latência da rede utilizando o túnel GRE, apresenta uma média de tempo de resposta de 0.656 ms, levando em consideração a ideal, um tempo de resposta um pouco mais lento.

```
root@n7:/tmp/pycore.46280/n7.conf# ping6 2001:2::11
PING 2001:2::11(2001:2::11) 56 data bytes
64 bytes from 2001:2::11: icmp_seq=1 ttl=64 time=0.538 ms
64 bytes from 2001:2::11: icmp_seq=2 ttl=64 time=0.664 ms
64 bytes from 2001:2::11: icmp_seq=3 ttl=64 time=0.651 ms
64 bytes from 2001:2::11: icmp_seq=4 ttl=64 time=0.671 ms
64 bytes from 2001:2::11: icmp_seq=5 ttl=64 time=0.682 ms
64 bytes from 2001:2::11: icmp_seq=6 ttl=64 time=0.637 ms
64 bytes from 2001:2::11: icmp_seq=7 ttl=64 time=0.639 ms
64 bytes from 2001:2::11: icmp_seq=8 ttl=64 time=0.710 ms
64 bytes from 2001:2::11: icmp_seq=9 ttl=64 time=0.675 ms
64 bytes from 2001:2::11: icmp_seq=10 ttl=64 time=0.700 ms
64 bytes from 2001:2::11: icmp_seq=11 ttl=64 time=0.652 ms
64 bytes from 2001:2::11: icmp_seq=12 ttl=64 time=0.677 ms
64 bytes from 2001:2::11: icmp_seq=13 ttl=64 time=0.629 ms
64 bytes from 2001:2::11: icmp_seq=14 ttl=64 time=0.651 ms
64 bytes from 2001:2::11: icmp_seq=15 ttl=64 time=0.635 ms
64 bytes from 2001:2::11: icmp_seq=16 ttl=64 time=0.686 ms
^C
--- 2001:2::11 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 14997ms
rtt min/avg/max/mdev = 0.538/0.656/0.710/0.039 ms
```

Imagem 46: Latência da implementação Túnel GRE.

É possível verificar que o tempo de resposta entre o servidor e o provedor de internet, na implementação do 6rd, é bom, com 0.194 ms. Esse tempo de resposta, não indica o tráfego interno, pois essa forma de transição não se aplica diretamente ao tráfego interno e sim se cria o túnel a partir dos equipamentos de borda.

```
root@n20:/tmp/pycore.43882/n20.conf# ping6 2001:0::10
PING 2001:0::10(2001::10) 56 data bytes
64 bytes from 2001::10: icmp_seq=1 ttl=62 time=0.475 ms
64 bytes from 2001::10: icmp_seq=2 ttl=62 time=0.149 ms
64 bytes from 2001::10: icmp_seq=3 ttl=62 time=0.154 ms
64 bytes from 2001::10: icmp_seq=4 ttl=62 time=0.210 ms
64 bytes from 2001::10: icmp_seq=5 ttl=62 time=0.179 ms
64 bytes from 2001::10: icmp_seq=6 ttl=62 time=0.175 ms
64 bytes from 2001::10: icmp_seq=7 ttl=62 time=0.173 ms
64 bytes from 2001::10: icmp_seq=8 ttl=62 time=0.199 ms
64 bytes from 2001::10: icmp_seq=9 ttl=62 time=0.161 ms
64 bytes from 2001::10: icmp_seq=10 ttl=62 time=0.175 ms
64 bytes from 2001::10: icmp_seq=11 ttl=62 time=0.189 ms
64 bytes from 2001::10: icmp_seq=12 ttl=62 time=0.159 ms
64 bytes from 2001::10: icmp_seq=13 ttl=62 time=0.173 ms
64 bytes from 2001::10: icmp_seq=14 ttl=62 time=0.166 ms
64 bytes from 2001::10: icmp_seq=15 ttl=62 time=0.176 ms
^C
--- 2001:0::10 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14000ms
rtt min/avg/max/mdev = 0.149/0.194/0.475/0.077 ms
```

Imagem 47: Latência da implementação 6rd.

5.5.2 Trace Route

O trace route, indica por onde todos os pacotes estão passando, do seu host de origem até seu host de destino. É importante sabermos qual o caminho esta percorrendo, pois é a partir do caminho, que vamos poder saber se o túnel foi feito corretamente e esta passando por todos os pontos nos quais deveria passar.

É possível verificar que na implantação da Pilha Dupla, todos os caminhos que o pacote passa, são todos os equipamentos que existem entre o host de origem e o host de destino.

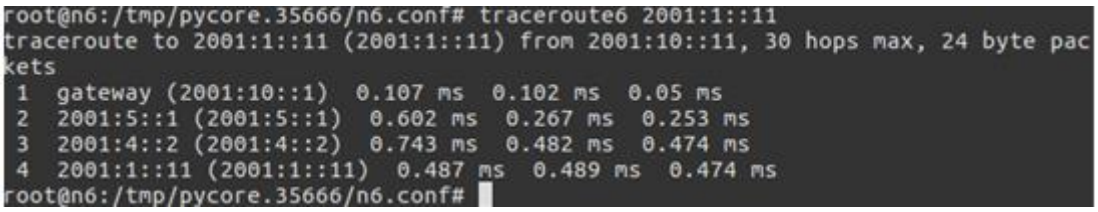

Imagem 48: Trace route da implementação Pilha Dupla.

É possível verificar no trace route na implantação do túnel 6in4, faz apenas 1 salto até chegar ao host destino.

Imagem 49: Trace route N6-N10 da implementação 6in4.

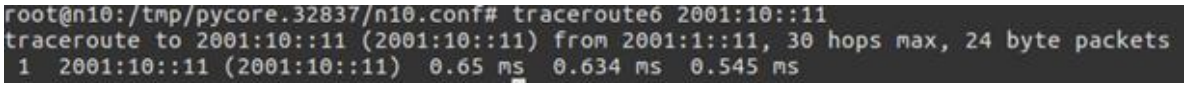

Imagem 50: Trace route N10-N6 da implementação 6in4.

É feito somente 1 salto, pois a criação do túnel, acaba ignorando todo o caminho que é feito, devido a utilização do protocolo IPv6, nesse caso como esta sendo feito o trace route do IPv6, só é possível se ver o túnel criado.

É possível verificar no trace route na implantação do túnel GRE, faz apenas 1 salto até chegar ao host destino.

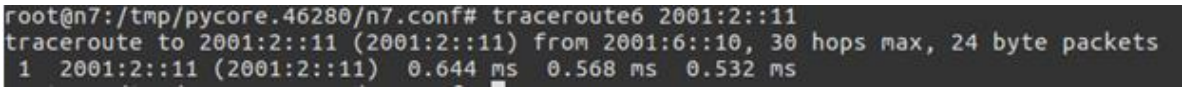

Imagem 51: Trace route N7-N11 da implementação GRE.

Imagem 52: Trace route N11-N7 da implementação GRE.

É feito somente 1 salto, pois a criação do túnel, acaba ignorando todo o caminho físico que é feito, devido a utilização do protocolo IPv6, nesse caso como esta sendo feito o trace route do IPv6, só é possível se ver o túnel criado.

É possível verificar na figura abaixo, que o pacote sai pelo servidor e passa inicialmente pelo seu gateway configurado, o IPv6 da rede, logo após segue para o endereço do equipamento, que foi configurado na interface de comunicação entre os dois pontos do túnel, e finalmente

chega até a interface na qual o túnel foi criado.

```
root@n20:/tmp/pycore.43882/n20.conf# traceroute6 2001:0::10
traceroute to 2001:0::10 (2001::10) from 2001:db8:cb00:7182::b1c0, 30 hops max, 24 byte packets
 1  gateway (2001:db8:cb00:7182::)  0.201 ms  0.114 ms  0.095 ms
 2  2001:db8::1 (2001:db8::1)  0.129 ms  0.111 ms  0.105 ms
 3  2001::10 (2001::10)  0.085 ms  0.081 ms  0.054 ms
```

Imagem 53: Trace route entre servidor e provedor.

5.5.3 Consumo total de banda

Abaixo é possível verificar o teste feito com a utilização de toda a banda da rede, no ambiente de pilha dupla, com envio de pacotes para todos os computadores, efetuando assim o máximo consumo de banda interna que a rede suporta.

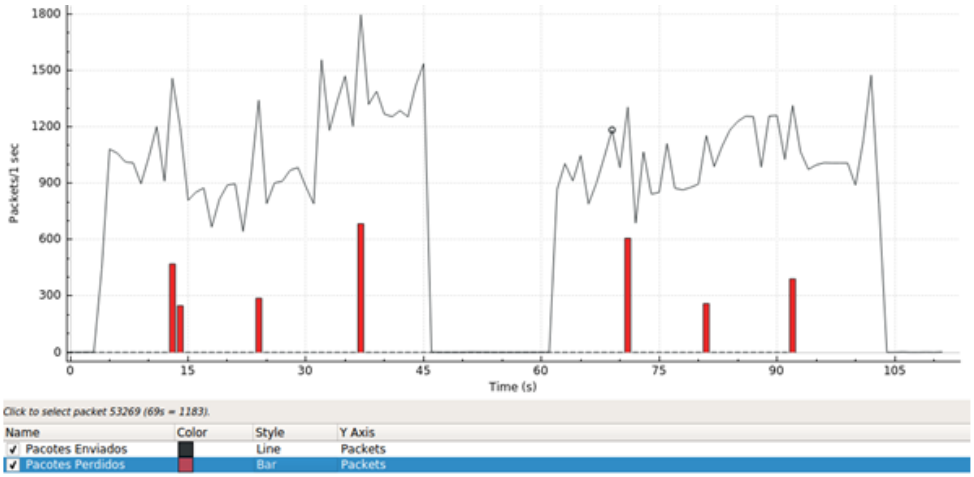

Imagem 54: Resultado consumo total da rede Pilha Dupla.

É possível verificar o desempenho comum da rede, como todos os hosts enviam arquivos ao mesmo tempo, os momentos nos quais a rede perde pacotes é quando se atinge a utilização máxima da rede, fazendo com que os pacotes tenham que ser reenviados, por isso um pico de pacotes enviados.

Abaixo é possível verificar o teste feito com a utilização de toda a banda da rede, no ambiente utilizando túnel 6in4, com envio de pacotes para todos os computadores, efetuando assim o máximo consumo de banda interna que a rede suporta.

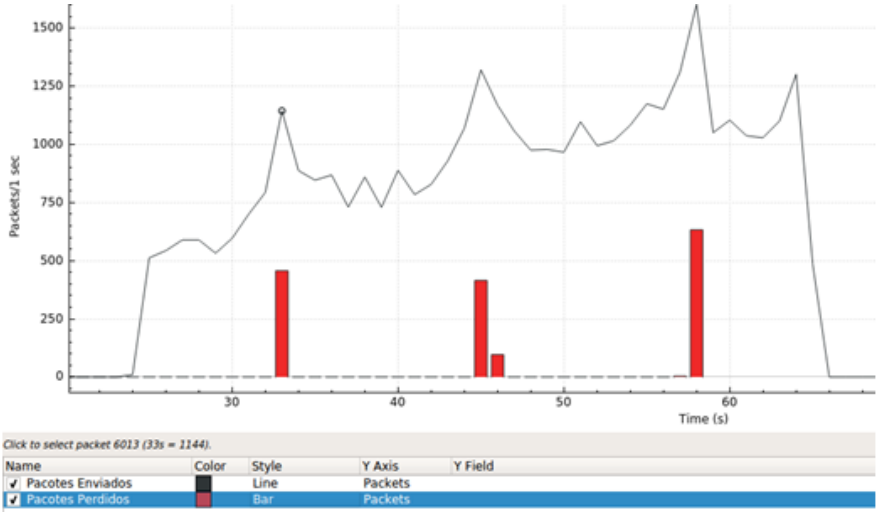

Imagem 55: Resultado consumo total da rede túnel 6in4.

É possível verificar, que o teste apresentou um desempenho bom, bem próximo ao ideal, com perda de pacotes no consumo total de banda da rede, onde há picos de pacotes enviados.

Abaixo é possível verificar o teste feito com a utilização de toda a banda da rede, no ambiente implementado com túnel GRE, com envio de pacotes para todos os computadores, efetuando assim o máximo consumo de banda interna que a rede suporta.

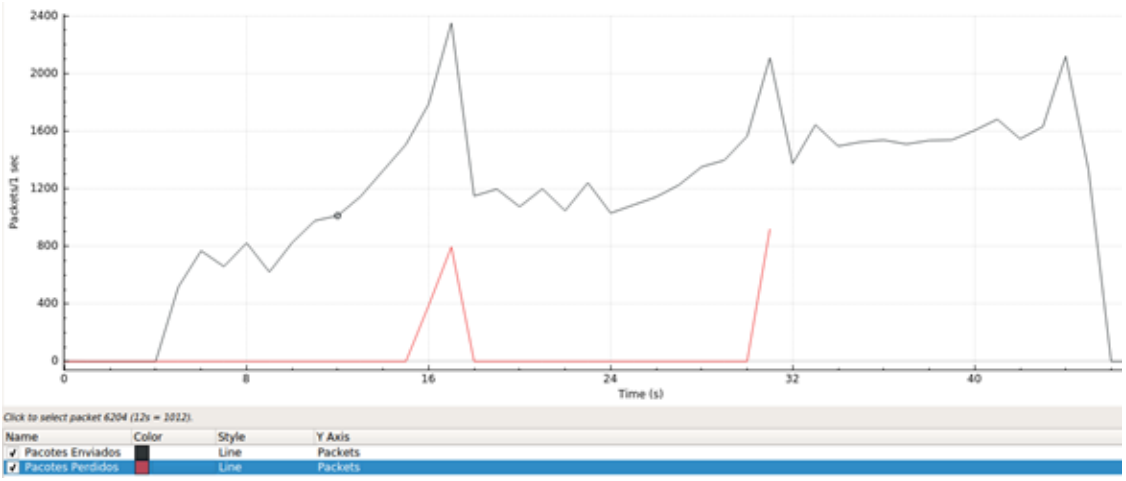

Imagem 56: Resultado consumo total da rede túnel GRE.

É possível verificar, que o teste apresentou um desempenho bom, bem próximo ao ideal, com perda de pacotes no consumo total de banda da rede, onde há picos de pacotes enviados.

Abaixo é possível verificar o teste feito com a utilização de toda a banda da rede, no ambiente implementado com 6rd, com envio de pacotes para todos os computadores, efetuando assim o máximo consumo de banda interna que a rede suporta.

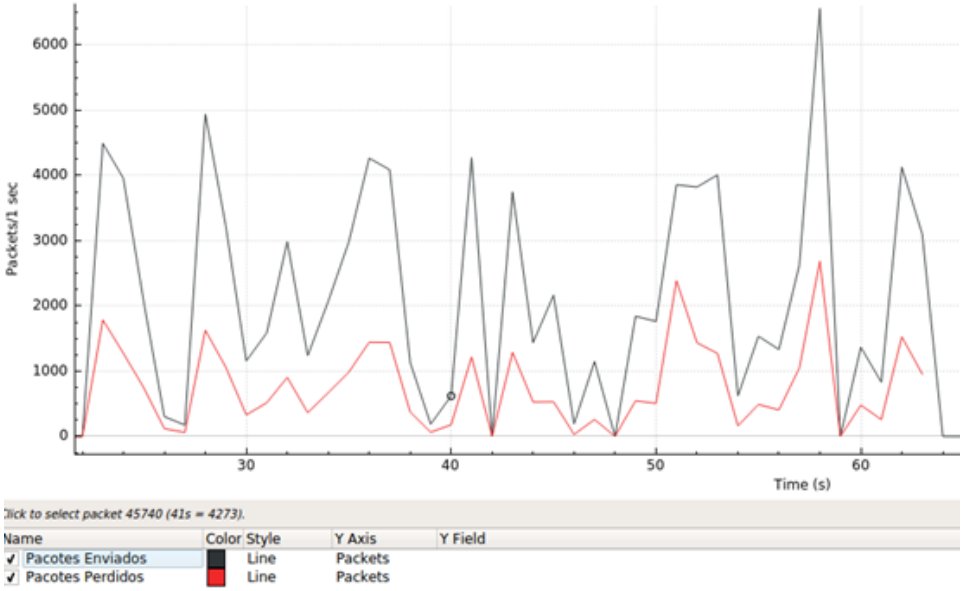

Imagem 57: Resultado consumo total da rede 6rd.

É possível verificar que o consumo da rede é bem maior, como a rede interna tem seu trafego inteiro em IPv4, o número de pacotes enviados é bem maior, consequentemente a rede tem uma média bem maior de perda de pacotes.

5.5.4 Envio arquivo real

O envio de apenas um arquivo na rede é importante para que possamos verificar se sem utilizar toda sua banda, os arquivos serão enviados e não terão grandes problemas de perda de pacote e tempo de envio.

Abaixo é possível verificar o teste feito com o envio de um arquivo real através do protocolo TCP, no ambiente implementado com pilha dupla.

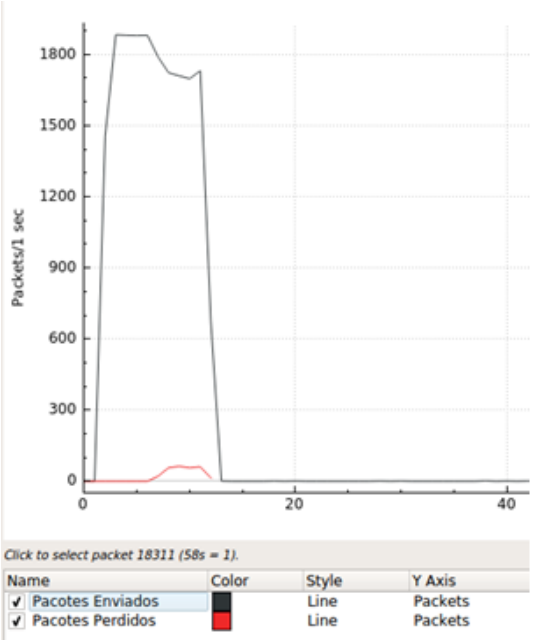

Imagem 58: Resultado envio de arquivo único na rede pilha dupla.

É possível verificar que os pacotes perdidos são praticamente zero. Indicando assim qual deve ser o parâmetro para comparação com todas as formas de transição. Devem ter resultados próximos a esses, para que não haja problemas.

Abaixo é possível verificar o teste feito com o envio de um arquivo real, no ambiente implementado com túnel 6in4.

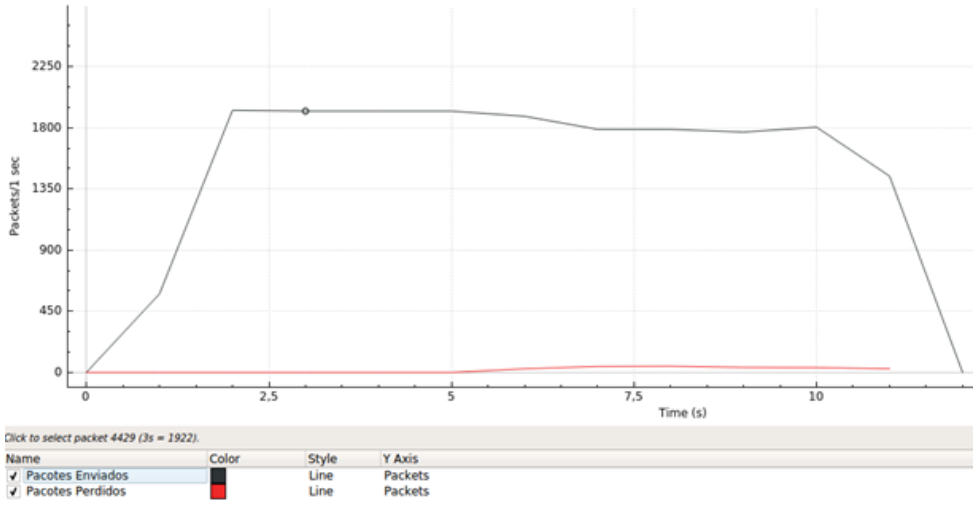

Imagem 59: Resultado envio de arquivo único na rede túnel 6in4.

O resultado apresentado pelo envio de arquivo na rede foi bom, apresentando um perda de pacotes mínima, com um tempo de envio de 14 segundos, um resultado bem próximo ao

resultado esperado.

Abaixo é possível verificar o teste feito com o envio de um arquivo real, no ambiente implementado com túnel GRE.

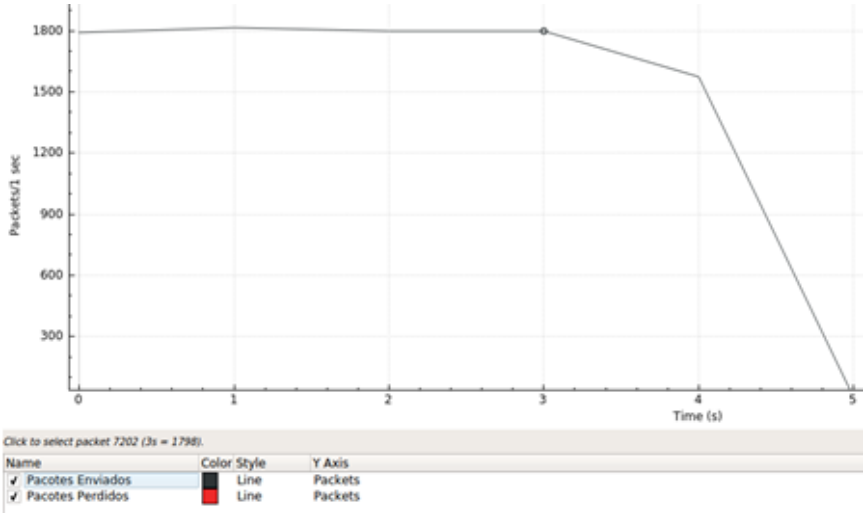

Imagem 60: Resultado envio de arquivo único na rede túnel GRE.

Abaixo é possível verificar o teste feito com o envio de um arquivo real, no ambiente implementado com 6rd.

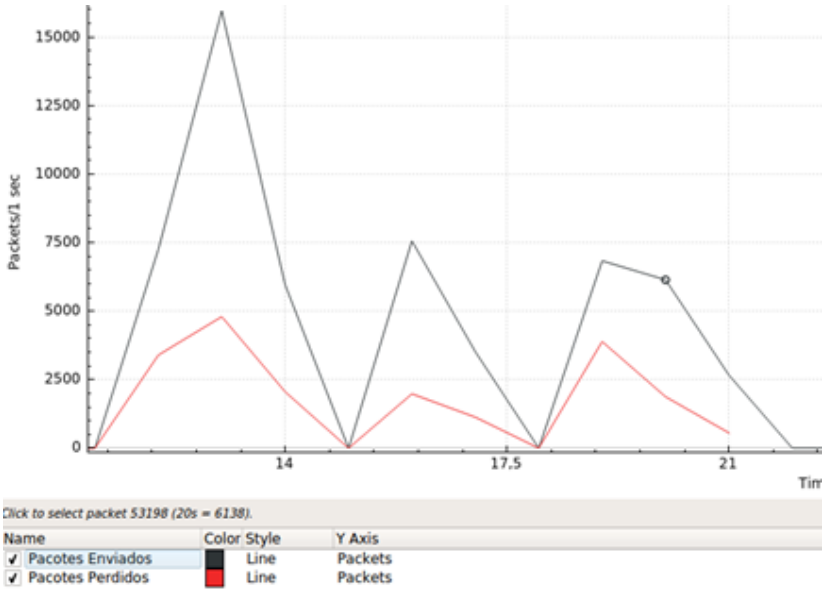

Imagem 61: Resultado envio de arquivo único na rede 6rd.

O tempo de envio é maior, comparando com as outras formas de transição, devido a toda rede funcionar com IPv4, a perda de pacote também é maior, pois com o aumento de envio de pacotes, existe também o aumento de erros.

6 Conclusão

É possível verificar que após todas as formas de transição, a que obteve o melhor desempenho, com menos perda de pacotes, latência bem próxima a ideal seria a transição com túnel GRE.

FORMAS DE TRANSIÇÃO	PING	TRACE ROUTE	PACOTES ENVIADOS (MÉDIA)	PACOTES PERDIDOS (MÉDIA)	ENVIO ARQUIVO (TEMPO)
PILHA DUPLA	0.615MS	4 SALTOS	900 PK/S	3%	18 S
TÚNEL 6IN4	0.669MS	1 SALTO	750 PK/S	4%	12,5 S
TÚNEL GRE	0.656MS	1 SALTO	1100 PK/S	3%	5 S
6RD	0.194MS	3 SALTOS	2200 pk/s	37%	21,5 S

Imagem 62: Tabela de resultados.

Porém a analise não se trata somente desses parâmetros, a analise deve também ser feita considerando quantos hosts a rede possuí, em casos de diversos hosts o servidor deverá ter diversas rotas de tunelamento criado, podendo assim gerar diversos problemas em caso de problemas com o servidor, já que a dependência de todos esses tuneis esta exclusivamente dependente do servidor. É necessário a analise do provedor de fornecimento, pois se o provedor não oferecer a possibilidade de utilização dos equipamentos necessários, no caso da implantação do 6rd, o usuário não pode simplesmente fazer a implantação, pois há a necessidade do servidor ter a rota criada para o usuário e fornecer o equipamento necessário para que seja implantado. Outra verificação necessária é o custo, se o custo da implantação for muito alto, talvez não seja tão interessante que seja feito um trabalho intenso de criação e configuração de tuneis, sendo que com a compra de placas de rede para suporte aos dois protocolos sairia mais em conta e seria a forma correta de ser feita essa implantação.

No ambiente proposto, a melhor forma de transição, visando não ter gastos altos e ter o melhor desempenho possível, seria a implantação do túnel GRE.

O túnel GRE apresentou o melhor desempenho de todas as formas de transição, tem uma implantação razoavelmente fácil de ser aplicada, porém mais trabalhoso que as demais.

Considerações Finais

Após o desenvolvimento do projeto, os objetivos propostos de fazer a comparação entre as formas de transição de IPv4 para IPv6, em um cenário especifico, foram bem sucedidos.

Obtendo um resultado satisfatório, porém que ainda pode ser alterado, variando entre outras necessidades do cenário de implantação.

O trabalho busca facilitar e ajudar na escolha de uma forma de transição, para empresas que fazem da internet uma ferramenta de extrema importância para o trabalho fornecido. O trabalho também visa ajudar empresas fornecedoras de internet, podendo analisar a implantação necessária nos usuários.

Em um futuro próximo, será necessária a implantação de transição de IPv6 para IPv4, em cenários nos quais o IPv6 será predominante e o IPv4 ainda existirá. O trabalho poderá auxiliar nessas formas de transição, pois a base utilizada para todas as formas de transição, foi apresentada.

Referências Bibliográficas

Brito, S. H. B. **Servidores DHCPv6 em Redes IPv6**. Disponível em: http://labcisco.blogspot.com.br/2013/05/servidores-dhcpv6-em-redes-ipv6.html. 2013.

Castro, J. e Ribeiro, L. **Manual de Funcionamento Wireshark**. Disponível em: http://www.cleberjean.com.br/downloads/Manual_Wireshark.pdf. 2008.

Conta, A. e Deering, S. **Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification**. Disponível em: https://tools.ietf.org/html/rfc2463. 1998.

Core-dev. **Core Documentation**.. Disponível em: http://downloads.pf.itd.nrl.navy.mil/docs/core/core_manual.pdf. 2015.

Conta, A. e Deering, S. **Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification**. Disponível em: https://tools.ietf.org/html/rfc4443. 2006.

Conta, A. e Deering, S. **Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification**. Disponível em: https://tools.ietf.org/html/rfc2463. 1998.

Crawford, M. **Router Renumbering for IPv6**. Disponível em: https://tools.ietf.org/html/rfc2894. 2000.

Deering, S. e Hinden, R. **Internet Protocol, Version 6 (IPv6) Specification**. Disponível em: https://tools.ietf.org/html/rfc2460. 1998.

Deering, S. **Host Extensions for IP Multicasting**. Disponível em: https://www.ietf.org/rfc/rfc1112.txt. 1989.

Despres, R. **IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)**. Disponível em: https://tools.ietf.org/html/rfc5569. 2010.

Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C. e Carney, M. **Dynamic Host Configuration Protocol for IPv6 (DHCPv6)**. Disponível em:

https://www.ietf.org/rfc/rfc3315.txt. 2003.

Droms, R. **Dynamic Host Configuration Protocol**. Disponível em: https://www.ietf.org/rfc/rfc2131.txt. 1997.

Erik Nordmark & Robert E. Gilligan. **Basic Transition Mechanisms for IPv6 Hosts and Routers**. Disponível em: https://tools.ietf.org/html/rfc4213. 2005.

Esli-nux. **Teste de Performance de Rede com Iperf**. Disponível em: http://www.esli-nux.com/2012/08/teste-de-performance-de-rede-com-iperf.html.

Farinacci, D., Hanks, S., Meyer, D. e Traina, P. **Generic Routing Encapsulation (GRE)**. Disponível em: https://tools.ietf.org/html/rfc2784. 2000.

Fazzanaro, L. P., **Protocolo IPv6 Uma Abordagem Geral**. 1ª Edição, Leme-SP. 2013.

Fuller,V. Li, T. **Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan**. Disponível em: https://tools.ietf.org/html/rfc4632. 2006.

Hinden, R. e Deering, S. **IP Version 6 Addressing Architecture**. Disponível em: https://www.ietf.org/rfc/rfc2373.txt. 1998.

Iperf. **iPerf - The ultimate speed test tool for TCP, UDP and SCTP**. Disponível em: https://iperf.fr/iperf-doc.php.

IPv6.br. **Introdução.** Disponível em: http://ipv6.br/post/introducao/. Acessado em: 16/05/2016.

IPv6.br. **Cabeçalho.** Disponível em: http://ipv6.br/post/cabecalho/. Acessado em: 16/05/2016.

IPv6.br. **Endereçamento.** Disponível em: http://ipv6.br/post/enderecamento/. Acessado em: 16/05/2016.

IPv6.br. **Funcionalidades básicas.** Disponível em: http://ipv6.br/post/funcionalidades-basicas/. Acessado em: 16/05/2016.

IPv6.br. **Transição.** Disponível em: http://ipv6.br/post/transicao/. Acessado em: 16/05/2016.

Johnson, D., Perkins, C. e Arkko, J. **Mobility Support in IPv6**. Disponível em: https://www.ietf.org/rfc/rfc3775.txt. 2004.

Kaoru, M., Moreiras, M. A. **IPv6**. Disponível em: http://ipv6.br/. 2012.

Kent, S. **IP Authentication Header**. Disponível em: https://tools.ietf.org/html/rfc4302. 2005.

Kent, S. **IP Encapsulating Security Payload (ESP)**. Disponível em: https://www.ietf.org/rfc/rfc4303.txt. 2005.

Narten, T., Nordmark, E., Simpson, W. e Soliman, H. **Neighbor Discovery for IP version 6 (IPv6)**. Disponível em: https://tools.ietf.org/html/rfc4861. 2007.

Nascimento, M. B. **IPv6 e as Principais Mudanças em Relação ao IPv4**. Disponível em: http://www.dltec.com.br/blog/redes/ipv6-versus-ipv4-para-o-ccna/. 2011.

Ortega, A. **Testando a rede com o Iperf (gerador de tráfego)**. Disponível em: http://brainwork.com.br/2010/06/21/testando-a-rede-com-o-iperf-gerador-de-trfego/. 2010.

Rekhter, Y., Moskowitz, B., Groot, G. J. e Lear, E. **Address Allocation for Private Internets**. Disponível em: https://tools.ietf.org/html/rfc1918. 1996.

Santos, R. R., Moreiras, A. M., Reis, E. A. e Rocha, A. S. **Curso IPv6 básico**. São Paulo, Núcleo de Informação e Coordenação do ponto BR. 2010.

Soliman. H. **Problem Statement: Dual Stack Mobility**. Disponível em: https://www.ietf.org/rfc/rfc4977.txt. 2007.

Srisuresh, P. e Egevang, K. B. **Traditional IP Network Address Translator (Traditional NAT)**. Disponível em: https://www.ietf.org/rfc/rfc3022.txt. 2001.

Teleco. **Comparativo entre IPv4 e IPv6**. Disponível em: http://www.teleco.com.br/tutoriais/tutorialredeip1/pagina_4.asp.

Wing. D., e Yourtchenko. A. **Happy Eyeballs: Success with Dual-Stack Hosts**. Disponível

em: https://tools.ietf.org/html/rfc6555. 2012.