Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Um Modelo de Migração de Ambiente IPv4 para IPv6 em uma Rede Acadêmica Heterogênea

Juvenal dos Santos Barreto

Dissertação apresentada como requisito parcial
para conclusão do Mestrado Profissional em Computação Aplicada

Orientadora
Prof.ª Dr.ª Priscila América Solis Mendez Barreto

Brasília
2015

B268m Barreto, Juvenal dos Santos.
Um Modelo de Migração de Ambiente IPv4 para IPv6 em uma Rede Acadêmica Heterogênea / Juvenal dos Santos Barreto ; orientador Priscila América Solis Mendez Barreto. -- Brasília, 2015.
145 p.

Dissertação (Mestrado – Mestrado Profissional em Computação Aplicada) – Universidade de Brasília, 2015

1. Interconexão de redes. 2. Migração de IPv4 para Ipv6. 3. Pilha Dupla. 4. Redes Locais. 5. IPv6. I. Barreto, Priscila América Solis Mendez, orient. II. Título.

CDU 004.738

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Um Modelo de Migração de Ambiente IPv4 para IPv6 em uma Rede Acadêmica Heterogênea

Juvenal dos Santos Barreto

Dissertação apresentada como requisito parcial
para conclusão do Mestrado Profissional em Computação Aplicada

Prof.ª Dr.ª Priscila America Solis Mendez Barreto (Orientador)
CIC/UnB

Prof. Dr. Eduardo Adilio Pelinson Alchieri
CIC/UnB

Prof. Dr. Georges Daniel Amvame Nze
ENE/UnB

Prof. Dr. Marcelo Ladeira
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 23 de março de 2015

Dedicatória

Este trabalho é dedicado a minha família: minha mãe Helenita dos Santos Barreto, em memória do meu pai Aloisio Barreto, aos irmãos Ana Claudia, Carlos Luis, Humberto Lúcio e Neila Carla pelo amor incondicional dispensado a mim, e ainda por todo o esforço e dedicação dispensados desde a infância para que eu pudesse ter uma formação na qual me realizasse profissionalmente e pessoalmente.

Agradecimentos

A Deus, por todas as conquistas obtidas, por me dar força pra superar os obstáculos na vida, enfrentando minhas fraquezas e inseguranças.

A todos os professores do Mestrado Profissionalizante em Computação Aplicada, em especial, aos professores Marcelo Ladeira e Jacir Bordim pelo permanente apoio, e não diferente à minha orientadora, professora Priscila Solis, que me deu importantíssimo subsídio no desenvolvimento do estudo em questão e por ter acreditado na minha vontade de realizar este trabalho e na confiança demonstrada durante toda a pesquisa.

A todos os colegas do curso, em especial ao Andrei, Arthur, Antonio, Eduardo, Jackson, Jobe, Karam e Riane, pelo constante companheirismo.

Aos colegas de trabalho Alex Fidelis, Alessandro Caldeira, Alessandro Cordeiro, Antonio Vasconcelos, Claudio Garcia, Claudio Xavier, Domingos Costa, Erasmo Losi, Erivando, Fernando Brito, Justino Mendonça, Hugo Chaves, Ivan Viotti, Luiz Capdeville, Maurício Hiroaki, Samuel Oliveira e Vinícius Cesário, pela compreensão em momentos de pressões intensas e contribuições pelo zelo às funções confiadas.

Aos meus amigos Péricles Amador, Fabrício Gonçalves, Bruno Cardoso, Edson Roxo e outros aqui não mencionados, e a prima Eliene Santos, pelas constantes palavras de motivação e compartilhamento de bons momentos.

Enfim, a todos que contribuíram diretamente ou indiretamente neste trabalho com simples apoios e incentivos.

Resumo

A crescente demanda por informações e, particularmente, o aumento exponencial de redes conectadas à Internet, faz com que as instituições tenham que modernizar suas infraestruturas frequentemente. A sensível limitação do endereçamento disponível desta rede contribui para que essas instituições estudem a implantação da nova versão do protocolo da Internet, o IPv6. A Universidade de Brasília, como grande provedor de acesso e de informações, e procurando manter-se conectada ao maior número de usuários possível e em alta disponibilidade, vê a necessidade de introdução de novas soluções em seu ambiente, mas por ser um ambiente muito complexo e heterogêneo, precisa ater-se a um modelo de implementação que permita execução de uma transição para o IPv6 de forma segura, gradual e suave.

Neste trabalho de pesquisa é apresentada uma metodologia para criar um ambiente de experimentos dentro da REDUnB (Rede de Dados da UnB) para implementação do IPv6, analisando aspectos relacionados às técnicas de transição com análises de desempenho destas comunicações. Por meio desta metodologia busca-se uma base para um modelo de migração do ambiente de IPv4 para IPv6 em um ambiente de rede acadêmica heterogênea, com perspectivas concretas de implementação no ambiente REDUnB.

Palavras-chave: Transição para o IPv6, Migração de IPv4 para IPv6, Pilha Dupla, Dual Stack, IPv6

Abstract

The growing demand for information and, particularly, the exponential increase in the number of networks connected to the Internet, makes the institutions have to modernize their infrastructures often. The sensitive limitation of available addressing this network contributes to these institutions to study the implementation of the new version of Internet protocol, IPv6. The University of Brasilia, as leading provider of access and information, and trying to keep connected to the largest number of users as possible and high availability, see the need to introduce new solutions in your environment, but because it is a very complex environment and heterogeneous, need to concentre to a deployment model that allows implementation of a transition to IPv6 in a secure manner, gradually and smoothly.

In this research work presents a methodology to create an environment of experiments within the REDUnB (Data Network of UNB) for IPv6 implementation, analyzing aspects related to the techniques of transition with performance analysis of these communications. Through this methodology seeks to a basis for a model of migration from the environment of IPv4 to IPv6 in a heterogeneous academic network environment, with concrete prospects of implementation in REDUnB environment.

Keywords: *IPv6, transition techniques*

Lista de Siglas

6RD: *IPv6 Rapid Deployment*
ACL: *Access Control List*
APF: Administração Pública Federal
API: *Application Programming Interface*
ARP: *Address Resolution Protocol*
AS: *Autonomous System*
ASN: *Autonomous System Number*
BIS: *Bump in the Stack*
BGP: *Border Gateway Protocol*
BRs: *Borders Relay*
CERNET2: *China Education and Research Network 2*
CGI.br: Comitê Gestor de Internet no Brasil
CIDR: *Classless Inter-Domain Routing*
CLNS: *Connectionless-mode Network Service*
CPD: Centro de Informática
CPU: *Central Processing Unit*
DDoS: *Distributed Denial of Service*
DHCP: *Dynamic Host Configuration Protocol*
DHCPv4: *Dynamic Host Configuration Protocol for IPv4*
DHCPv6: *Dynamic Host Configuration Protocol for IPv6*
DNS: *Domain Name System*
DNS64: *Domain Name System IPv6-IPv4*
DSCP: *DiffServ Code Point*
e-PING: Padrões de Interoperabilidade de Governo Eletrônico
ECN: *Explicit Congestion Notification*
EGP: *Exterior Gateway Protocol*
FINATEC: Fundação de Empreendimentos Científicos e Tecnológicos
FT: Faculdade de Tecnologia
FTP: *File Transfer Protocol*

HTTP: *Hypertext Transfer Protocol*
IANA: *Internet Assigned Numbers Authority*
ICC: Instituto Central de Ciências
ICMP: *Internet Control Message Protocol*
ICMPv6: *Internet Control Message Protocol version 6*
IETF: *Internet Engineering Task Force*
IGMP: *Internet Group Management Protocol*
IGPs: *Interior Gateway Protocols*
IP: *Internet Protocol*
IPSec: *Internet Protocol Security*
IPv4: *Internet Protocol version 4*
IPng: *Internet Protocol next generation*
IPv6: *Internet Protocol version 6*
IS-IS: *Intermediate System to Intermediate System*
ISATAP: *Intra-Site Automatic Tunnel Addressing Protocol*
ISP: *Internet Service Provider*
ITU-T: *International Telecommunication Union*
LAN: *Local Area Network*
LSA: *Link State Advertisement*
MAC: *Media Access Control*
MAP-E: *Mapping of Address and Port-Encapsulation*
MOS: *Mean Opinion Score*
MTU: *Maximum Transmission Unit*
NAT: *Network Address Translator*
NAT44: *Network Address Translator from IPv4 to IPv4*
NAT-PT: *Network Address Translation-Protocol Translation*
ND: *Neighbor Discovery*
NDP: *Neighbor Discovery Protocol*
NIC.br: Núcleo de Informação e Coordenação do Ponto BR
NLPID: *Network Layer Protocol IDentifier*
NTP: *Network Time Protocol*
OSPF: *Open Shortest Path First*
PAMS: *Perceptual Analysis Measurement System*
PDU: *Protocol Data Unit*
PESQ: *Perceptual Evaluation of Speech Quality*
PoC: *Proof of Concept*
QoS: *Quality of Service*

RARP: *Reverse Address Resolution Protocol*
REDECOMEP: Redes Comunitárias de Educação e Pesquisas
REDUnB: Rede de Dados da UnB
RFC: *Request for Comments*
RIP: *Routing Information Protocol*
RNP: Rede Nacional de Ensino e Pesquisa
SEND: *Secure Neighbor Discovery*
SIIT: *Stateless IP/ICMP Translation*
SIP: *Session Initiation Protocol*
SLAAC: *Stateless Address Autoconfiguration*
SPF: *Shortest Path First*
SSH: *Secure Shell*
SSL: *Secure Socket Layer*
TCP: *Transmission Control Protocol*
TCP/IP: *Transmission Control Protocol/ Internet Protocol*
ToS: *Type of Service*
TTL: *Time to Live*
UDP: *User Datagram Protocol*
UnB: Universidade de Brasília
URL: *Uniform Resource Locator*
VLAN: *Virtual Local Area Network*
VoIP: *Voice over Internet Protocol*
VPN: *Virtual Private Network*

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Este capítulo expõe a evolução das redes de computadores, e com esse objetivo, a suíte de protocolos TCP/IP (*Transmission Control Protocol/Internet Protocol*) desempenha um papel fundamental na conectividade de redes de tecnologias distintas, o que contribuiu substancialmente para o crescimento da Internet, dada a transparência que este protocolo dá às redes, facilitando o desenvolvimento contínuo de novas aplicações e serviços.

A denominação TCP/IP provém dos nomes dos dois protocolos essenciais da sequência de protocolos, os protocolos TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*). Este último, o protocolo IP, foco do estudo em questão, tem por propósito conceder aos dispositivos na grande rede um endereço único, possibilitando que sejam identificados e encontrados e, consequentemente, que a comunicação possa ocorrer.

Os sistemas que se comunicam com a rede pública necessitam do protocolo IP para compartilharem arquivos e recursos, mas este vem apresentando um sensível esgotamento de endereçamento IP na versão 4 [2]. Para o enfrentamento deste obstáculo, uma nova versão de protocolo foi desenvolvida e já se apresenta como parte das necessidades de comunicação. A versão 6 do protocolo IP, mais conhecida como IPv6 (*Internet Protocol version 6*), se torna um avanço importante devido a implementações de novas características, permitindo mais eficiência e segurança, além de solucionar o problema da insuficiência de endereços do protocolo IPv4 (*Internet Protocol version 4*). Com sua utilização uma nova questão deverá ser discutida, a forma como as versões diferentes poderão se comunicar sem a geração de incompatibilidades.

Esta comunicação deverá ser alcançada por meio de mecanismos de transição, mas é necessário avaliar com prudência se estes mecanismos de transição conseguirão minimizar o impacto e as dificuldades que o processo de migração ocasionará.

Com isso, o propósito deste trabalho de pesquisa é aprofundar no conhecimento sobre a versão mais recente do protocolo IP, subsidiando a elaboração de um modelo de migração do ambiente IPv4 para IPv6 em rede acadêmica heterogênea. Para tal, é imprescindível a

comprovação de viabilidade do modelo por meio de avaliação em ambiente experimental de rede confinada em laboratório. Posteriormente, pretende-se implementar um ambiente piloto em produção na REDUnB, com base no modelo de referência técnica produzido neste trabalho.

1.1 Justificativa

Com a grande evolução da computação e da Internet, as redes de informação ganharam espaço nas atividades mais simples do dia-a-dia, ensejando interação entre os dispositivos ligados à grande rede pública, e com imprescindível papel nas empresas, com capacidade de trafegar dados, imagem, voz, vídeo, por meio de uma infraestrutura única. Isso foi possível em função do protocolo IP, que permite a comunicação entre *hardwares* e sistemas de diferentes arquiteturas, o que o tornou muito difundido, fazendo-se necessária a criação de mecanismos de convergência de tecnologias.

Mesmo com muitos mecanismos disponíveis, inclusive para melhoria da alocação dos endereços públicos, a demanda pelo uso da Internet segue crescendo expressivamente, com a iminência do esgotamento de alocação de novos endereços IPv4, o que inibe o desenvolvimento da chamada Internet das Coisas (*Internet of Things*). Por outro lado, o IPv4 não foi projetado para suportar serviços como os que atualmente tem sido muito demandados, como serviços móveis, de tempo real, multimídia, dentre outros, apresentando como um desafio para a Internet do Futuro [2]. A fim de integrar plenamente essas novas tecnologias, a rede deve suportar recursos altamente variáveis dentro de curtos períodos de tempo, ou ainda atrasos de propagação extremamente longos.

Abordagens para uma Internet do Futuro vão de pequenos passos evolutivos incrementais até uma remodelagem completa nos princípios arquiteturais, onde as tecnologias aplicadas não podem ser limitadas por normas existentes ou paradigmas. Nesse contexto, o protocolo IPv6 pode ser visto como um passo na evolução das redes de computadores, sendo necessário na infraestrutura da Internet, uma questão de continuidade de negócios, para provedores, empresas e instituições.

Um grande benefício da adesão ao IPv6 é a disponibilidade de um número extremamente maior de endereços se comparado ao IPv4. A alta disponibilidade de endereços e prefixos de rede fornece uma flexibilidade na arquitetura de redes que permite uma organização hierárquica e inclusive geográfica, onde um prefixo de rede pode ser usado para endereçar um país ou até mesmo um continente e segmentá-los em diversos níveis, permitindo que seja feita hierarquização da estrutura com objetivo de reduzir o tamanho das tabelas de roteamento, aumentando a escalabilidade.

Um outro ponto chave da adesão ao protocolo IPv6 é quanto a segurança, pois na arquitetura deste protocolo, este aspecto já é provido de forma nativa, sobretudo sob suporte do protocolo IPSec (*Internet Protocol Security*). Mecanismos de autenticação e encriptação passaram a fazer parte do protocolo IPv6, disponibilizando para qualquer par de dispositivos de uma conexão fim-a-fim, métodos que visam garantir a segurança dos dados que trafegam pela rede, no entanto, o aprimoramento do aspecto segurança no IPv6 continua sendo um desafio. Contudo, o IPv6 traz novidades para as quais as equipes técnicas e os equipamentos de segurança ainda não estão bem preparados.

Com base em aspectos diversos relacionados à área de tecnologia, incluindo a migração para o protocolo IPv6, foi elaborado um conjunto de premissas, políticas e especificações técnicas que regulamentam a utilização da Tecnologia da Informação e Comunicação no governo federal do Brasil, resultando em um documento de referência denominado como e-PING (Padrões de Interoperabilidade de Governo Eletrônico). Este documento estabelece as condições de interação com os demais poderes e esferas de governo e com a sociedade em geral, proporcionando a operação integrada entre equipamentos, programas e sistemas de informação, visando o aproveitamento irrestrito dos potenciais de intercâmbio de dados e informações na esfera da APF (Administração Pública Federal) direta, autárquica e fundacional [3].

No contexto das políticas técnicas para interconexão de ativos das redes de dados, o documento de referência expõe que os órgãos da APF deverão se interconectar utilizando IPv4 e planejar sua futura migração para IPv6. Novas contratações e atualizações de redes devem prever suporte à coexistência dos protocolos IPv4 e IPv6 e a produtos que suportem ambos os protocolos.

Somando a estas orientações advindas do Comitê Executivo de Governo Eletrônico desde os idos anos de 2004, a RNP (Rede Nacional de Ensino e Pesquisa) que coordena infraestrutura de rede Internet voltada para a comunidade brasileira de ensino e pesquisa, vem fazendo coro a esta linha de pensamento, muito embora não tenha avançado o suficiente em ações no sentido de prover a seus clientes, como a própria UnB, a alternativa de saída pela Internet IPv6.

Todavia, a transição do protocolo IPv4 para o IPv6 é vista como um passo importante para o futuro da Internet, mas ambos coexistirão por um bom tempo ainda, e esta transição deverá ocorrer de forma gradual e transparente para o usuário final.

1.2 Contribuição Esperada

A proposta espera contribuir com:

- A criação de um ambiente de rede experimental dentro da REDUnB onde seja possível estudar e avaliar o protocolo IPv6 como PoC (*Proof of Concepts*);
- Elaboração de um modelo técnico-administrativo consistente de migração gradual do protocolo IPv4 para o IPv6 em ambiente de rede heterogêneo da REDUnB, que possa ser tomado como caso de sucesso para aplicação;
- Produção de documento final que sirva como referência teórica para compreender o IPv6 e guia de implementação prática do protocolo como meta de transferência de tecnologia para profissionais de tecnologia da informação;
- Avaliação de conjunto básico de serviços sobre IPv6 como DNS (*Domain Name System*) e HTTP (*Hypertext Transfer Protocol*).

1.3 Resumo do Capítulo

Esta seção abordou de forma breve o atual contexto em que se insere os protocolos TCP/IP nas redes de computadores, mencionando as principais versões do protocolo IP e sua intercomunicação. Neste ensejo é citada uma proposta de trabalho para um modelo de migração, seguida de justificativas que fundamentam o aprofundamento do conhecimento desta nova versão do protocolo IP e as contribuições que se espera deste estudo.

Capítulo 2

Revisão de Literatura

O propósito aqui é realizar um detalhado estudo sobre o protocolo IPv6 visando a praticidade de sua aplicação nas redes atuais, descrevendo os recursos e funcionalidades básicas, apontando diferenças significativas e suas vantagens em relação ao protocolo IPv4, além de buscar uma compreensão das técnicas de transição e o comportamento dos protocolos IPv4 e IPv6 quanto à interoperabilidade.

2.1 Protocolo IP

A Internet é um grande aglomerado de computadores espalhados ao redor do mundo disponibilizando serviços diversos a quem tiver interesse e autorização para acessá-los. Para que computadores distintos rodando sistemas operacionais diferentes possam se comunicar, é preciso que tenham os mesmos padrões de comunicação, e é neste âmbito que o protocolo IP desempenha um papel fundamental, concedendo aos dispositivos na grande rede um endereço IP globalmente único e de formato uniforme, possibilitando que sejam identificados e, consequentemente, que a comunicação possa ocorrer [4].

Sucintamente, considerando o uso dos mesmos padrões de comunicação, quando um pacote IP é recebido por um roteador, seu endereço de destino é procurado na tabela de roteamento. Se o destino for uma rede distante, o pacote será encaminhado para o

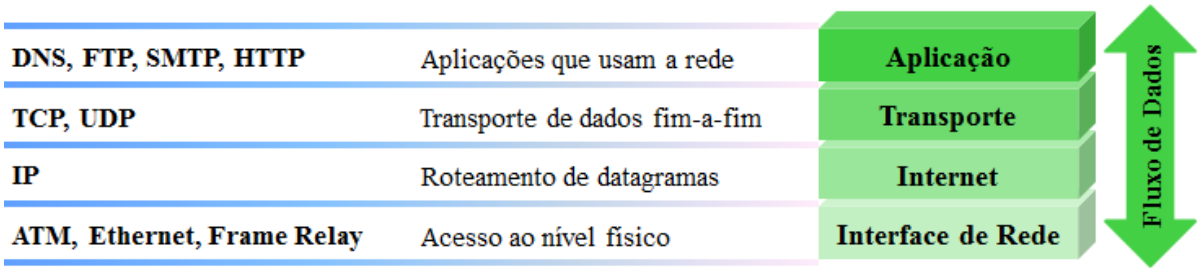


Figura 2.1: Modelo TCP/IP em camada com suas respectivas funções e protocolos

próximo roteador da interface fornecida na tabela. Caso o destino seja um *host* local, por exemplo, na LAN (*Local Area Network*) do roteador, o pacote será enviado diretamente para lá. Se a rede não estiver presente, o pacote será enviado para um roteador predefinido que tenha tabelas mais abrangentes [5].

2.2 Protocolo IPv4

A versão 4 do protocolo IP não sofreu alterações substanciais desde a sua criação na década de 1980, e passou a ser um dos protocolos mais amplamente difundidos e implementados em todo o mundo por se tratar de um projeto flexível e poderoso no qual foi possível conciliar as constantes mudanças tecnológicas, provando ser robusto, de fácil implementação e muito escalável [6]. O protocolo IPv4 é o mecanismo responsável pela comunicação da pilha TCP/IP, tendo relacionamento basicamente com a camada de Internet do modelo TCP/IP.

Como citado anteriormente, para que os dispositivos de uma rede possam trocar informações é necessário que todos adotem os mesmos padrões de comunicação para o envio e recebimento de informações, podendo ser entendidos também como um conjunto de regras, estas denominadas como protocolo de comunicação. Neste contexto, o TCP/IP tornou-se padrão de fato na Internet e utiliza um esquema de comunicação concebido em quatro camadas: Aplicação, Transporte, Internet e Interface de Rede, como se vê na ilustração adaptada Figura 2.1 [5].

A camada de Aplicação contém todos os protocolos para um serviço específico, utilizada pelos programas para enviar e receber informações de outros programas através da rede. Nesta camada são identificadas aplicações para resolução de nomes de domínios em endereços IP, para transferência de arquivos, correio eletrônico, navegação na Internet, dentre outras, cada tipo de programa se comunicando com um protocolo de aplicação diferente, dependendo da finalidade do programa.

Após processar a requisição do programa, o protocolo na camada de Aplicação se comunicará com um outro protocolo na camada de Transporte, usando TCP ou UDP (*User Datagram Protocol*). A camada de Transporte é responsável por pegar os dados enviados pela camada superior, dividi-los em pacotes e enviá-los para a camada inferior, a camada Internet. Além disso, a camada de Transporte é responsável por ordenar os pacotes recebidos da rede e também verificar se o conteúdo dos pacotes está intacto.

Na camada de Internet há o protocolo IP que pega os pacotes recebidos da camada de Transporte e adiciona informações de endereços IP de origem e destino, gerando assim, o que chamamos de datagrama. Em seguida os datagramas são enviados para a camada imediatamente inferior, a camada Interface de Rede.

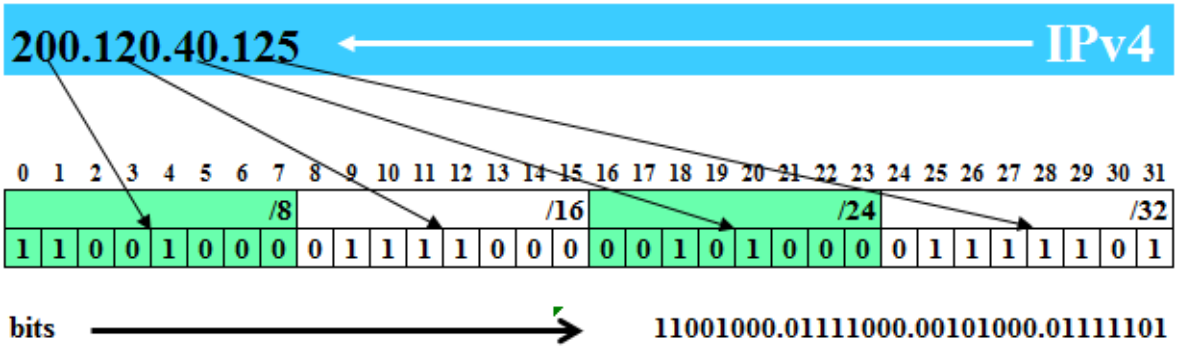


Figura 2.2: Endereçamento IPv4 representado em *bits*

Na camada Interface de Rede os datagramas serão enviados para a rede, ou receberá os dados da rede, caso o computador esteja recebendo dados. Os protocolos inclusos nesta camada dependerão do tipo de rede que o computador estiver usando, com uma grande predominância do protocolo Ethernet.

Como visto, o protocolo IP reside na camada Internet do modelo TCP/IP, e em sua versão 4, o IPv4, a cada computador é associado um endereço de 32 *bits*, chamado endereço IP, como pode ser visualizado na Figura 2.2. No exemplo, o endereço 11001000011110000010100001111101 em notação binária pode ser difícil de compreender, então dividi-lo em 4 partes de 8 dígitos binários, ou seja, 4 octetos 11001000.01111000. 00101000.01111101, e então converter estes endereços binários em formato decimal facilitará bastante. Estes endereços IP são habitualmente escritos com 4 números decimais entre 0 e 255 separados por pontos, por exemplo, o endereço IP 200.120.40.125 [5].

Para tornar eficaz o encaminhamento dos datagramas os endereços IP são arranjados em duas partes, uma que é o prefixo que identifica a rede e outra que identifica o nó nessa mesma rede. As decisões de encaminhamento dos datagramas baseiam-se na parte que identifica a rede. Tomando o exemplo do endereço IP citado, todos os endereços dos nós dessa rede começam por 200.120.40, só o último *byte* é diferente, então endereços 200.120.40.125 e 200.120.40.135 correspondem a dois nós distintos da rede 200.120.40.

Uma rede IP 200.120.40.0 com máscara de subrede 255.255.255.0 é composta por endereços que vão de 200.120.40.1 até 200.120.40.254, sendo que o endereço 200.120.40.0 é usado para designar a rede inteira e o endereço 200.120.40.255 é usado para o *broadcast* da rede. Esta rede só se comunica com as máquinas que estejam nesta faixa de IP e para isso não precisam de nenhum dispositivo para auxiliá-los a encontrar essas máquinas, tal como um roteador.

2.2.1 Cabeçalho do IPv4

Um datagrama IP consiste em duas partes, o cabeçalho e os dados. O cabeçalho inclui campos adicionais à mensagem a ser transmitida. O cabeçalho de um pacote IPv4 tem tamanho que varia entre 20 e 60 *bytes*, e é apresentado mediante ilustração adaptada na Figura 2.3 [5]. Uma breve descrição de cada campo no cabeçalho IPv4 segue abaixo conforme descreve [7]:

- ***Version*** (4 *bits*): aponta a versão do protocolo, no caso, a versão 4 do protocolo IP;
- ***Internet Header Length*** (4 *bits*): como um datagrama IPv4 pode conter um número variável de opções incluídas no seu cabeçalho, esses quatro *bits* são necessários para determinar onde, no datagrama IP, os dados realmente começam;
- ***Type of Service*** (8 *bits*): especifica informações especiais para diferenciar os distintos tipos de datagramas IP, como tipos que requerem particularmente, baixo atraso, alta vazão ou confiabilidade, devendo ser distinguidos uns dos outros;
- ***Total Length*** (16 *bits*): número de identificação de cada datagrama enviado, utilizado na remontagem dos fragmentos do datagrama;
- ***Identification*** (16 *bits*): identifica um datagrama;
- ***Flags*** (3 *bits*): identificam a transmissão de sinais de controle;
- ***Fragment Offset*** (13 *bits*): um valor numérico sucessivo atribuído a cada fragmento do datagrama. O IP no destino utiliza este campo para remontar os fragmentos do datagrama na ordem correta;
- ***Time to Live*** (8 *bits*): indica o número máximo de roteadores pelos quais um datagrama pode passar. É decrementado de 1 em cada roteador, e o datagrama é descartado quando o TTL (*Time to Live*) atinge zero;
- ***Protocol*** (8 *bits*): indica qual protocolo de alto nível foi usado para criar a mensagem que está sendo transportada na área de dados do datagrama;
- ***Header Checksum*** (16 *bits*): destina-se à verificação da validade do cabeçalho. É recalculado em cada roteador à medida que o campo TTL é decrementado;
- ***Source Address*** (32 *bits*): informa o endereço de origem;
- ***Destination Address*** (32 *bits*): informa o endereço de destino;
- ***Options*** (entre 0 e 320 *bits*): de tamanho variável com informações de segurança, roteamento, relatórios de erro, etc. Pode aparecer ou não em um datagrama.

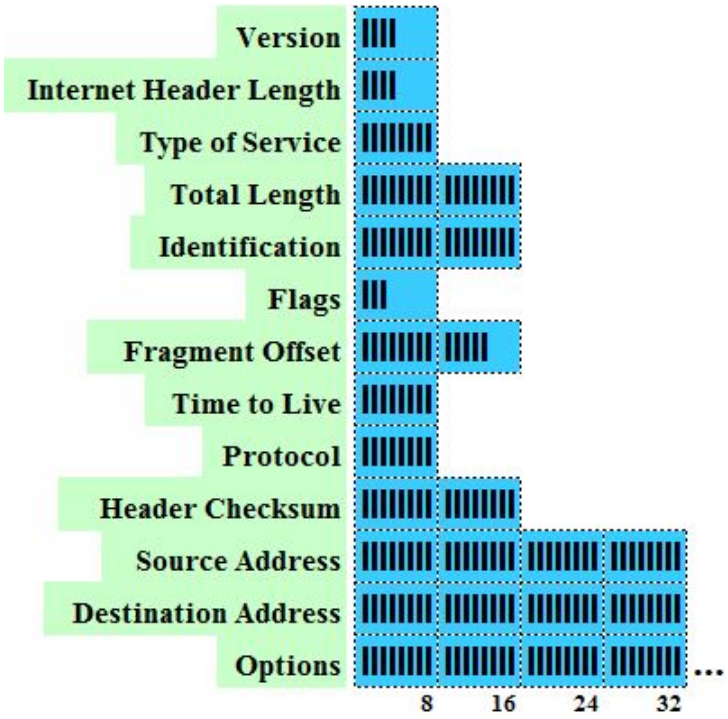


Figura 2.3: Cabeçalho IPv4

Apesar de ser um projeto flexível, poderoso e de grande êxito desde seu surgimento, o IPv4 tem sofrido alguns problemas. Um deles é referente ao crescimento exponencial da Internet e a ameaça de exaustão do espaço de endereçamento, onde os endereços IPv4 se tornaram relativamente escassos, forçando as organizações a adotarem soluções paliativas como o CIDR (*Classless Inter-Domain Routing*) [8], o DHCP (*Dynamic Host Configuration Protocol*) [9], o NAT (*Network Address Translator*) [10] e por fim, a alocação de endereços privados com três faixas de endereços, não válidos na Internet, para uso em redes corporativas [11].

Outro problema enfrentado pelo IPv4 refere-se à necessidade de um suporte melhorado para a entrega de dados em tempo real. Nessa circunstância, o serviço padrão oferecido pela rede IP é conhecido como serviço *Best Effort* (melhor esforço), fazendo sempre o melhor possível para encaminhar os pacotes de acordo com os recursos que ele tem disponíveis naquele instante de tempo, mas sem qualquer garantia de entrega. O serviço de melhor esforço consiste em oferecer o mesmo tratamento aos pacotes, sem nenhuma distinção entre eles. Com isso, o QoS dá o aporte necessário às redes IP com um conjunto de algoritmos capazes de fornecer vários níveis de tratamentos para diferentes tipos de tráfego na rede. O propósito dessa tecnologia é otimizar o uso da banda passante provendo um tráfego fim-a-fim eficaz e econômico.

Com objetivo de resolver estes e os demais problemas enfrentados pelo protocolo IPv4, no começo da década de 1990, a IETF (*Internet Engineering Task Force*) iniciou um

esforço para desenvolver o sucessor do protocolo IPv4 [7]. Esta nova versão do protocolo IP, hoje conhecida como IPv6 tenta causar o mínimo impacto nos protocolos das camadas acima e abaixo, eliminando a adição aleatória de novas características.

2.3 Protocolo IPv6

Considerando as limitações do protocolo IP na versão 4, justificou-se a evolução deste protocolo para uma versão mais avançada. No início foi utilizada a designação IPng (*Internet Protocol next generation*) como referência à geração seguinte do protocolo IP, entretanto substituída pela designação IPv6 atualmente adotada [12].

Os aspectos essenciais do IPv4 que estão na base do sucesso do protocolo foram mantidos no IPv6. O cerne da evolução se concentrou em reformular as deficiências do protocolo IPv4, as funcionalidades que não têm um bom desempenho ou que não são usadas com frequência foram tornadas opcionais ou simplesmente excluídas. Algumas novas características que se consideram necessárias foram adicionadas. No IPv6 foram introduzidas novas funcionalidades tais como suporte a mobilidade, segurança de forma nativa, suporte melhorado para cabeçalhos de extensão além de promover a simplificação do cabeçalho base, dentre outras mudanças.

2.3.1 Endereçamento IPv6

A principal razão para a reestruturação do formato de endereçamento da nova versão do protocolo IP foi de atender a carência de alocações públicas, passando de endereçamentos de 32 *bits* do IPv4 para 128 *bits* do IPv6 [13]. Estes endereços de 128 *bits* são tipicamente representados em notação hexadecimal divididos em 8 grupos de 16 *bits* separados por dois pontos (:), representando cada grupo com um hexadecimal (base 16) número de 0 a FFFF, podendo usar letras maiúsculas ou minúsculas para dígitos hexadecimais, e denominado como Hexadecateto. Exemplo de um endereço IPv6 é mostrado a Figura 2.4.

Assim, o IPv6 aumenta substancialmente o número de endereços em relação ao IPv4, admitindo alocar em torno de 340 undecilhões de endereços possíveis.

Com propósito de melhor representar a compreensão do formato do endereço, é adotada uma simplificação de notação em que quando houver grupos de zeros, apenas um deles é necessário ser escrito e, os zeros à esquerda de grupos com outros valores, não necessitam ser representados. Assim, o endereço apresentado na Figura 2.4 pode ser representado por FEAB:0:0:0:8:800:400B:335C. De forma ainda mais simplificada, pode ser utilizado um par de dois pontos (::) para representar grupos de zeros consecutivos,

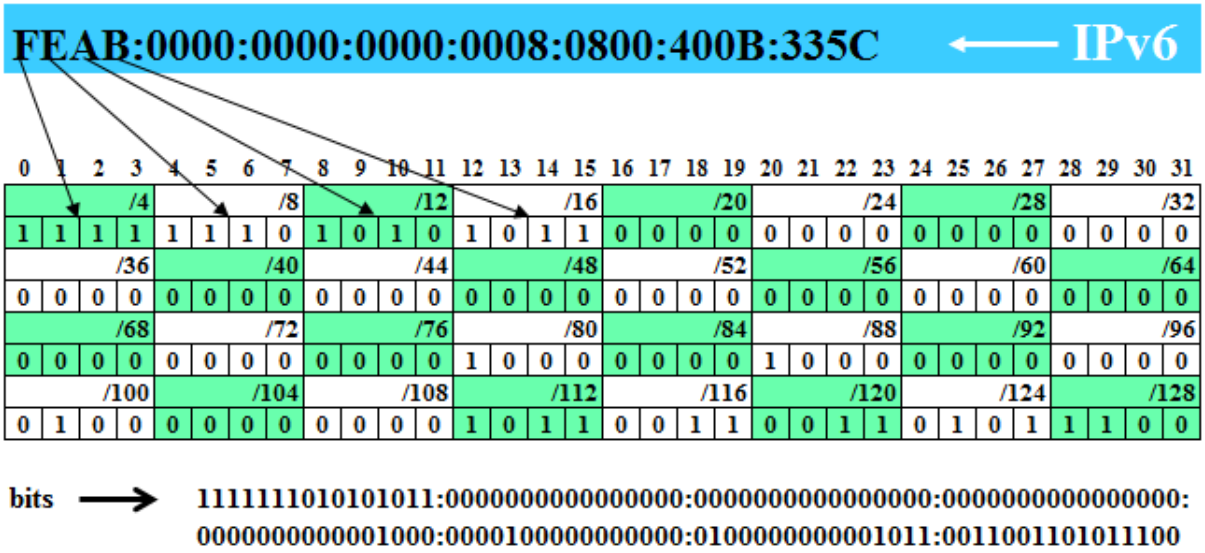


Figura 2.4: Endereçamento IPv6 representado em *bits*

conforme pode ser observado na notação FEAB::8:800:400B:335C. Cabe enfatizar que somente uma supressão de zeros por um par de dois pontos (::) é admitida. Caso ocorram duas sequências de zeros, apenas uma deverá receber esta representação [14].

Com referência à representação dos endereços IPv6 em URLs (*Uniform Resource Locators*), estes agora passam a ser incluídos entre colchetes. Assim, é evitado que possíveis ambiguidades ocorram caso seja necessário indicar o número de uma porta juntamente com a URL. Segue abaixo exemplo de URLs com endereços IPv6 que seguem essa convenção:

- **http://[FEAB::8:800:400B:335C]/index.html**
- **http://[FEAB::8:800:400B:335C]:8080**

O IPv6 não utiliza máscaras de rede, mas emprega a notação de prefixo que é comum também no roteamento IPv4 ao usar a notação CIDR. Assim, um *range* de endereços IPv6 de 2001:DB8:31:1:: a 2001:DB8:31:1:FFFF:FFFF:FFFF:FFFF pode ser escrito como 2001:DB8:31:1::/64, onde o tamanho do prefixo é um valor decimal que especifica a quantidade de *bits* contíguos à esquerda do endereço que compreendem o prefixo. Assim, o prefixo /64 apresentado indica que dos 128 *bits* do endereço, 64 *bits* mais à esquerda são utilizados para identificar a subrede e os 64 *bits* restantes para identificar a interface. A parte de endereço em um prefixo deve ser um endereço IPv6 válido com todos os *bits* que não fazem parte do prefixo definido para zero (0). Então, 2001:DB8:31:1::/64 e 2001:DB8:31:1::/127 são prefixos válidos, mas 2001:DB8:31:1/64 e 2001:DB8:31:1::/48 não são. No primeiro caso, a parte do endereço não é um endereço IPv6 válido de 128

bits, e no segundo caso, a parte :1:: fica fora dos 48 *bits* de prefixo, por isso deve ser zero, escrito como 2001:DB8:31::/48 [15].

Diferente do IPv4, no IPv6 não existe endereço *broadcast*, este responsável por direcionar um pacote para todos os nós de um mesmo domínio. Entretanto, os endereços do tipo *Unicast*, *Multicast* e *Anycast* são adotados, e na sequência seguem algumas características destes tipos de endereços.

O tipo de endereço *Unicast* permite que um datagrama seja entregue exclusivamente para a interface que possui o endereço especificado. Estes endereços *Unicast* ainda podem ser segmentados nas categorias descritas a seguir:

- ***Globals***: semelhantes aos endereços públicos do IPv4, sendo roteáveis globalmente na parte IPv6 da Internet;
- ***Link-Local***: por meio deste tipo de endereçamento executa-se a comunicação entre nós pertencentes à mesma rede local. O escopo deste endereço é o enlace local, de modo que os roteadores nunca encaminham para outro enlace pacotes com este endereço. Ele é usado também pelos processos do *Neighbor Discovery* e é sempre automaticamente configurado, mesmo na ausência de todos os outros tipos de endereços *Unicast*. Os dez primeiros *bits* deste endereço sempre começam com FE80::/10;
- ***Site-Local***: são equivalentes aos endereços IPv4 privados, ou seja, está restrito a um domínio sem ligação com a Internet. Eles podem ser usados em conjunto com os endereços *Global Unicast*. Estes endereços não são automaticamente configurados e precisam ser designados com configuração *stateless* ou *stateful*. Os 10 primeiros *bits* deste endereço são fixos em FEC0::/10;
- ***Unspecified***: é usado para identificar a ausência de um endereço e é representado por 0:0:0:0:0:0:0:0 ou ainda por ::;
- ***Loopback***: é usado quando um nó envia um datagrama para ele mesmo. É representado por 0:0:0:0:0:0:0:1 ou ainda por ::1;
- ***IPv4 Compatible***: utilizado quando se necessita encaminhar um datagrama de uma rede IPv6 para outra utilizando tunelamento em redes IPv4. São representados como um endereço IPv6 com os últimos 32 *bits* correspondendo a um endereço IPv4 e os 96 *bits* iniciais acrescentados de zeros.

Um datagrama que é destinado a um endereço *Multicast* é entregue a todas as interfaces que constam daquele grupo de endereços. Como no IPv6 o endereço *broadcast* encontra-se indisponível, serviços conseguem utilizar característica semelhante ao *broadcast* por meio de endereços do tipo *Multicast* [16].

No ambiente de redes IPv4 o *Multicast* também existe, sendo executado pelo protocolo IGMP (*Internet Group Management Protocol*) [17]. No entanto, o *Multicast* no IPv4, embora útil, ele é opcional. Em contrapartida, em IPv6, *Multicast* é obrigatório, na verdade, fundamental para operação do IPv6. O protocolo IGMP foi incorporado pelo ICMPv6, conforme RFC 2710 [18], e o *Multicast* é usado para implementar o ARP (*Address Resolution Protocol*) equivalente do IPv6.

Todos os endereços *Multicast* derivam do bloco FF00::/8, onde o prefixo FF identifica um endereço *Multicast*, e este prefixo antecede quatro *bits*, que representam quatro *flags*, seguido por outros quatro *bits* que define o escopo do grupo de endereços *Multicast*. O restante dos 112 *bits* ficam à cargo da identificação de grupo *Multicast*. As informações apresentadas na Tabela 01, demonstram alguns endereços *Multicast* permanentes.

Tabela 2.1: Alguns Endereços Multicast Permanentes

Endereço	**Escopo**	**Descrição**
FF01::1	*Node-Local*	Todos os nós
FF01::2	*Node-Local*	Todos os roteadores
FF02::1	*Link-Local*	Todos os nós
FF02::2	*Link-Local*	Todos os roteadores
FF02::5	*Link-Local*	Roteadores OSPF
FF02::6	*Link-Local*	Roteadores OSPF designados
FF02::9	*Link-Local*	Roteadores RIP
FF02::D	*Link-Local*	Roteadores PIM
FF02::1:2	*Link-Local*	Agentes DHCP
FF05::2	*Site-Local*	Todos os roteadores
FF05::1:3	*Site-Local*	Servidores DHCP em um Site
FF05::1:4	*Site-Local*	Agentes DHCP em um Site

Uma lista completa de endereços permanentes do *Multicast* pode ser encontrada no endereço *web* da IANA (*Internet Assigned Numbers Authority* - http://www.iana.org), a qual é relativamente extensa. Entretanto, dois endereços *Multicast* são muito importantes, recomendável conhecê-los, trata-se dos endereços FF02::1 e FF02::2. O primeiro é o *Link-Local* (interface) dos endereços de todos os nós, algo próximo da equivalência com endereço de *broadcast* 255.255.255.255 do protocolo IPv4. O segundo é o *Link-Local* (interface) de todos os endereços de roteadores, sendo estes dois de fundamental importância para o processo de autoconfiguração no IPv6 [19].

Um endereço *Anycast* é designado para comunicação com múltiplas interfaces. Pacotes endereçados a um endereço *Anycast* são encaminhados pela infraestrutura de roteamento para a interface mais próxima do endereço *Anycast* designado. Recordando, endereços *Unicast* são atribuídos a uma máquina e cada pacote é entregue a essa máquina. Endereços *Multicast* são atribuídos a várias máquinas e cada pacote é entregue a todas essas

máquinas. Já os endereços *Anycast* são atribuídos a muitas máquinas, mas cada pacote é entregue a apenas uma dessas máquinas. Complementando a compreensão, um endereço *Unicast* atribuído a mais de uma interface transforma-se em um endereço *Anycast*, com a devida explicitação do uso de endereço *Anycast* nas configurações dos nós.

Os endereços *Anycast* são projetados para fornecer redundância e balanceamento de carga em situações onde múltiplos *hosts* ou roteadores proveem o mesmo serviço, e eles utilizam o mesmo *range* de endereços *Globals Unicast* e são indistinguíveis a partir deles. No entanto, um nó que é atribuído a um endereço *Anycast* deve ser configurado para estar ciente deste fato. Endereços *Multicast* e *Anycast* podem ser usados como endereços de destino em pacotes, mas somente endereço *Unicast* pode ser usado como endereço de origem. Além disso, somente os roteadores podem ser configurados com um endereço *Anycast* no IPv6 [2].

Para descomplicar a entrega, a infraestrutura de roteamento deve estar vigilante às interfaces designadas como *Anycast* e a suas distâncias em termos de métricas de roteamento. Exemplos de utilização mais básicos estão relacionados a serviços UDP, principalmente DNS, quando se tem muitos servidores publicados em diferentes localidades com o mesmo número IP [20].

Plano de Alocação de Endereços IPv6

O endereçamento IPv6 possui estrutura flexível para atribuição de endereço, conferindo redes baseadas em diferentes critérios, tais como tamanho da rede e taxa de crescimento estimado. Em casos frequentes, uma atribuição inicial pode não ser tão escalável se uma rede pequena torna-se maior do que o esperado e, portanto, precisa de mais endereços.

A solução mais fácil, mas menos flexível é adotar a atribuição de endereço de bloco IPv6 em ordem desde o início do bloco alocado para a organização. Mas esta ação não leva em consideração as necessidades futuras e não pondera a respeito do agrupamento de redes por área que possibilite a sumarização de roteamento. Além disso, este método torna muito difícil ou quase impraticável fazer uma atribuição para aumentar a rede existente e manter seu espaço de endereço contíguo.

Para que a rede lógica esteja mais organizada possível, o espaço de endereçamento deverá ser distribuído de forma hierárquica, de acordo com a topologia e a infraestrutura física da rede. Existem vários fatores que deverão ser considerados na atribuição de endereçamento a cada unidade, dos quais se destacam a dimensão, a localização, a importância ou o contexto dentro da instituição.

Contudo, é muito importante que o planejamento da atribuição de endereço de bloco IPv6 ocorra, e a RFC 3531 [21], denominada como *A Flexible Method for Managing the*

Assignment of Bits of an IPv6 Address Block, sugere alternativa que auxilia no melhor uso do bloco de endereços.

2.3.2 Estrutura do Cabeçalho IPv6

O cabeçalho IPv6 é uma versão simplificada do cabeçalho IPv4, tendo sido projetado com sensíveis mudanças em relação ao seu antecessor. No IPv6, cinco campos do cabeçalho IPv4 foram removidos conforme pode ser visto abaixo:

- ***Header Length***;
- ***Identification***;
- ***Flags***;
- ***Fragment Offset***;
- ***Header Checksum***.

O campo *Header Length* foi removido por ter se tornado desnecessário, uma vez que seu valor foi fixado. Os campos *Identification*, *Flags* e *Fragment Offset* passaram a ter suas informações indicadas em cabeçalhos de extensão apropriados. Por fim, o campo *Header Checksum* foi eliminado com o objetivo de aumentar a velocidade de processamento já que outras validações são realizadas pelos protocolos das camadas superiores da rede [2].

Outros campos do IPv4 passaram por alterações no IPv6, sendo renomeados respectivamente, como o campo *Type of Service* que foi alterado para *Traffic Class*, o campo *Total Length* para *Payload Length*, o campo *Time to Live* para *Hop Limit* e o campo *Protocol* para *Next Header*.

A estrutura do cabeçalho IPv6 como bem descrito na RFC 2460 [22] que especifica o protocolo IPv6 norteando toda a comunidade da Internet bem como suscitando discussões e sugestões para sua melhoria, é apresentada na ilustração adaptada na Figura 2.5 [22].

O cabeçalho tem comprimento fixo de 40 *bytes*. Considerando que os dois campos de endereços de origem e destino usam 16 *bytes* cada, fica restando só 8 *bytes* para informação geral do cabeçalho.

Uma breve descrição de cada campo no cabeçalho IPv6 segue abaixo:

- ***Version*** (4 *bits*): indica a versão do protocolo, no caso, a versão 6 do protocolo IP;
- ***Traffic Class*** (8 *bits*): indica a classe ou prioridade do pacote IPv6, funcionalidade semelhante ao campo ToS (*Type of Service*) do cabeçalho IPv4. Os 6 primeiros *bits* do campo *Traffic Class* representam o campo DSCP (*DiffServ Code Point*) conforme especifica a RFC 2474 [23], e os últimos 2 *bits* são usados para ECN (*Explicit Congestion Notification*) conforme define a RFC 3168 [24];

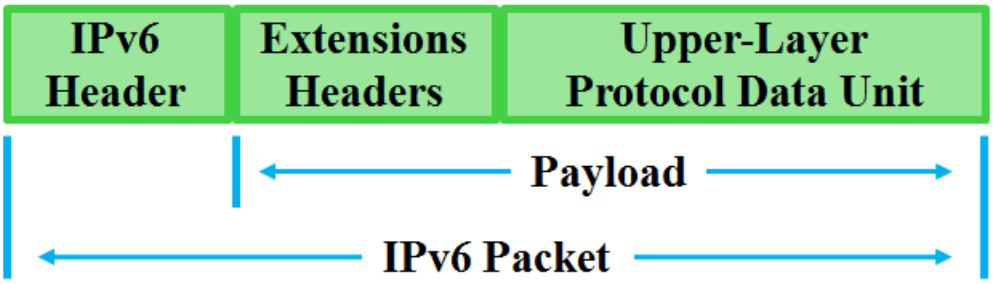


Figura 2.5: Estrutura de um pacote IPv6

- ***Flow Label*** (20 *bits*): é o identificador de fluxo. Este campo consiste num valor arbitrário que pode ser utilizado pelo emissor para identificar pacotes, para os quais tenha requerido uma determinada qualidade de serviço. Um fluxo é uma sequência de pacotes enviados por um determinado emissor para um destino específico para o qual, o emissor deseja um tratamento especial e análogo por parte dos roteadores intermediários no encaminhamento dos pacotes;
- ***Payload Length*** (16 *bits*): indica o comprimento da carga útil de dados do pacote IPv6, combinando os cabeçalhos de extensão e a PDU da camada superior. Estes *Payloads* normalmente possuem comprimento de até 65535 *bytes*, mas podem exigir cargas maiores, sendo assim designados como *jumbograms*;
- ***Next Header*** (8 *bits*): indica o tipo do cabeçalho que se encontra imediatamente após o cabeçalho base IPv6. Ele usa os mesmos valores do campo *Protocol* do IPv4, como definido na RFC 1700 [25]. Se este valor contém o código para o TCP, então o cabeçalho TCP e o *Payload* iniciam imediatamente depois do pacote cabeçalho IPv6. De outro modo, um ou mais cabeçalhos de extensão do IPv6 podem ser encontrados antes do início do cabeçalho TCP e de *Payload*. Assim, desde que cada cabeçalho de extensão possua outro campo *Next Header* e um campo *Header Length*, isto constitui em uma lista de cabeçalho antes do final do cabeçalho de extensão, seguido pelo *Payload*;
- ***Hop Limit*** (8 *bits*): é semelhante ao campo TTL do IPv4, exceto pelo fato de não ter relação histórica com a quantidade de tempo (em segundos) que o pacote aguarda na fila do roteador. Ele indica o número máximo de saltos que o pacote IPv6 pode dar antes de ser descartado;
- ***Source Address*** (128 *bits*): identifica o endereço IPv6 do nó de origem;
- ***Destination Address*** (128 *bits*): identifica o endereço IPv6 do nó de destino. Este campo é definido para o endereço de destino final, no entanto, se um cabeçalho

de extensão de roteamento está presente, o campo *Destination Address* pode ser configurado para o endereço do próximo destino intermediário.

Cabeçalhos de Extensão

Os cabeçalhos de extensão podem ser ou não encontrados no pacote IPv6. Se cabeçalhos de extensão estão presentes nos pacotes IPv6, o campo *Next Header* no cabeçalho IPv6 indica o primeiro cabeçalho de extensão. Dentro de cada cabeçalho de extensão tem outro campo *Next Header*, indicando o próximo cabeçalho de extensão. O último cabeçalho de extensão aponta para o cabeçalho dos protocolos da camada superior como o TCP, UDP ou o ICMPv6 (*Internet Control Message Protocol version 6*), contidos na PDU da camada superior. O novo formato de cabeçalho de extensão permite que o IPv6 suporte novas funcionalidades e necessidades futuras.

Os cabeçalhos de extensão têm tamanhos variáveis, não têm tamanho máximo e podem expandir-se para acomodar todos os dados de extensão necessários para a comunicação IPv6. A PDU da camada superior compreende normalmente do cabeçalho do protocolo da camada superior e seu *Payload* - carga útil de dados [14].

Os tipos básicos de Cabeçalhos de Extensão são definidos pela RFC 2460 [22], e são os seguintes:

- ***Hop-by-Hop Options Extension Header***: utilizado para transportar informações opcionais que devem ser examinadas por todos os nós ao longo do caminho de entrega do pacote. Identificado pelo valor 0 (zero) no campo *Next Header*;
- ***Routing Extension Header***: utilizado pelo nó de origem para listar um ou mais nós intermediários que devem ser visitados até que o pacote chegue ao destino. Identificado pelo valor 43 no campo *Next Header*;
- ***Fragment Extension Header***: utilizado quando o pacote IPv6 a ser enviado é maior que o *Path MTU* (*Maximum Transmission Unit*) para o seu destino. No IPv6, a fragmentação de pacotes é feita somente no nó de origem, devendo usar uma descoberta de MTU para delimitar o tamanho máximo dos pacotes ao longo do caminho até o destino. Identificado pelo valor 44 no campo *Next Header*;
- ***Destination Options Extension Header***: utilizado para transportar informações opcionais que precisam ser examinadas apenas pelo nó de destino do pacote. Identificado pelo valor 60 no campo *Next Header*.

2.3.3 Funcionalidades Básicas do IPv6

Esta sessão tem o objetivo de apresentar aspectos teóricos sobre as funcionalidades básicas do IPv6. No decorrer da exposição são feitas comparações em relação ao IPv4, com intuito de enfatizar as principais diferenças e seus motivos.

ICMPv6

A especificação do IPv6 redefine o ICMP (*Internet Control Message Protocol*) do IPv4 com algumas mudanças, resultando na denominação de protocolo ICMPv6 [26]. No IPv4, uma prática comum de alguns administradores, é o bloqueio total de mensagens ICMP na operação normal da rede. No âmbito de redes IPv6, em situação normal de operação, a prática de bloqueio total de mensagens ICMPv6 não é recomendada, uma vez que estas mensagens são de uso fundamental para as funcionalidades básicas do protocolo IPv6 [14].

O protocolo ICMPv6 é muito mais poderoso do que ICMPv4 e contém novas funcionalidades. Esta nova versão tem a capacidade de informar erros se os pacotes não podem ser processados corretamente e enviar mensagens informativas sobre o status da rede. A função IGMP que gerencia as adesões do grupo *multicast* com IPv4 foi incorporada ao ICMPv6. O mesmo ocorreu para os protocolos ARP/RARP (*Address Resolution Protocol/Reverse Address Resolution Protocol*), funções usadas em IPv4 para mapear endereços da camada 2 para endereços IP e vice-versa. Foi incorporado também o ND (*Neighbor Discovery*), utilizando mensagens ICMPv6 para determinar endereços de camada de enlace para vizinhos ligados ao mesmo escopo de subrede, para encontrar roteadores, acompanhar quais vizinhos são alcançáveis e detectar os endereços de camada de enlace alterados. Novos tipos de mensagens foram definidas para tornar mais simples a renumeração de redes e atualização de informações de endereço entre *hosts* e roteadores [2].

Todas as mensagens ICMPv6 possuem a mesma estrutura geral de cabeçalho como se vê na ilustração adaptada Figura 2.6 [2]:

- **Campo *Type* (1 *byte*)** Especifica o tipo de mensagem, o qual determina o formato do restante da mensagem;
- **Campo *Code* (1 *byte*)** Identifica o subtipo de mensagem dentro de cada valor do tipo da mensagem ICMP;
- **Campo *Checksum* (2 *bytes*)** Usado para detectar dados corrompidos no cabeçalho ICMPv6 e em parte do cabeçalho IPv6. Para calcular o *checksum*, um nó deve determinar os endereços de origem e destino no cabeçalho IPv6. Um campo pseudo-cabeçalho do cabeçalho IPv6 é anexado para o cálculo da soma de verificação;

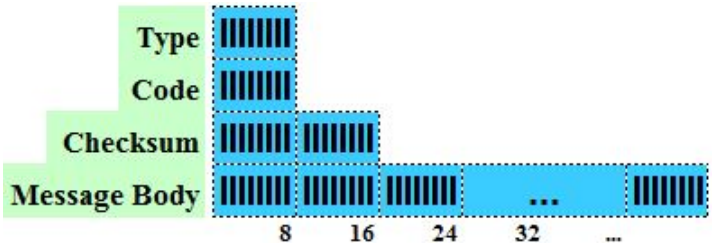


Figura 2.6: Estrutura geral de cabeçalho de mensagem ICMPv6

- **Campo *Message Body* (tamanho variável)** Depende dos valores em *Type* e *Code*, o corpo da mensagem conterá dados diversos. No caso de uma mensagem de erro, para ajudar na solução, ela vai conter, quanto possível do pacote solicitado na mensagem. O tamanho total do pacote ICMPv6 não deve exceder o mínimo do MTU IPv6, que é 1280 *bytes* [2].

NDP (*Neighbor Discovery Protocol*)

O NDP (*Neighbor Discovery Protocol*) é um novo protocolo dentro do IPv6 para a descoberta de vizinhança, concebido com o objetivo de sanar os problemas de relacionamento entre os nós vizinhos de uma rede. O seu uso possibilita que nós de uma rede consigam verificar a presença uns dos outros, apontar os endereços de seus vizinhos, descobrir roteadores e manter informações atualizadas sobre rotas a serem utilizadas na transmissão de pacotes [14].

Para a comunicação entre nós em uma rede, os nós precisam de algumas informações além do endereço de destino. Para obter essas informações alguns procedimentos são utilizados:

- ***Router Discovery***: atua na descoberta de roteadores pertencentes ao enlace;
- ***Prefix Discovery***: atua na descoberta de prefixos de redes do enlace, com objetivo de decidir para onde os pacotes serão mandados numa comunicação, se para um roteador específico ou direto para um nó do enlace;
- ***Parameter Discovery***: *hosts* podem descobrir os parâmetros IPv6 corretos para qualquer enlace em que ele esteja inserido, como o MTU e o *Hop Limit*;
- ***Stateless Address Autoconfiguration***: procedimento de autoconfiguração de endereço *Stateless* na camada de enlace;
- ***Address Resolution***: atua na descoberta de endereço físico de interfaces de rede por meio de seu endereço lógico IPv6;

- ***Next-Hop Determination***: algoritmo utilizado para mapear endereços IP de nós vizinhos para os quais pacotes devem ser enviados mediante seus endereços de destino;
- ***Neighbor Unreachability Detection***: recurso utilizado para detectar se um nó vizinho é ou continua sendo acessível;
- ***Duplicate Address Detection***: atua na tarefa de descobrir se o endereço que se deseja configurar já está sendo utilizado por um outro nó na rede;
- ***Redirect***: permite que o roteador oriente um nó na rede a respeito de uma melhor rota a ser utilizada no encaminhamento de pacotes a um destino específico.

Dessa forma, o NDP age sobre dois aspectos principais da comunicação IPv6, a autoconfiguração de nós na rede e a transmissão de pacotes entre os nós na rede. Assim, as funcionalidades *Parameter Discovery*, *Address Autoconfiguration* e *Duplicate Address Detection* influenciam na autoconfiguração de nós na rede, enquanto as funcionalidades *Router Discovery*, *Prefix Discovery*, *Address Resolution*, *Neighbor Unreachability Detection*, *Next-Hop Determination* e *Redirect* influenciam na transmissão de pacotes entre os nós na rede.

Por meio de 5 mensagens do ICMPv6 o NDP consegue executar estas funcionalidades citadas. Existem duas classes de mensagens ICMPv6, uma conhecida como mensagem de erro ICMP que utiliza identificação dentro do *range* de 0 a 127, e a outra mensagem informativa ICMP, identificada pelo *range* de 128 a 255 [15]:

- ***Router Solicitation* (*Type* 133)**: enviada por um dispositivo para requisitar que roteadores da rede imediatamente se apresentem através da resposta *Router Advertisement*;
- ***Router Advertisement* (*Type* 134)**: enviada pelo roteador para anunciar sua presença no enlace e suas configurações, isto periodicamente ou como resposta a uma mensagem *Router Solicitation*;
- ***Neighbor Solicitation* (*Type* 135)**: enviada por um dispositivo para requisitar que um vizinho se apresente imediatamente através da resposta *Neighbor Advertisement*, atuando na descoberta de um endereço físico através de um endereço lógico como o papel do ARP no IPv4, no teste de acessibilidade de nós vizinhos do enlace, além da detecção de endereços IPv6 duplicados na vizinhança;
- ***Neighbor Advertisement* (*Type* 136)**: enviada tanto em resposta a uma mensagem *Neighbor Solicitation* quanto para anunciar de forma voluntária, a alteração

de alguma característica de um dispositivo na rede. Assim como no *Neighbor Solicitation*, também atua na resolução de endereços físicos, no teste de acessibilidade de nós vizinhos e na detecção de endereços duplicados;

- ***Redirect* (*Type* 137)**: enviada por roteadores para avisar a um nó da rede sobre uma melhor rota para alcançar um destino específico.

Autoconfiguração

A capacidade de autoconfiguração do IPv6 disponibiliza um substancial auxílio aos administradores de rede. Por meio desta importante característica, os diversos dispositivos podem adquirir informações da rede, do enlace e de endereçamento. Isto leva a um grande dinamismo à Internet, visto que permite dispositivos se interconectarem sem a necessidade de configurações manuais.

Existem dois modos de divulgação de informações para a autoconfiguração dos dispositivos:

Stateless: conhecido também pela sigla SLAAC (*Stateless Address Autoconfiguration*), o equipamento que fornece informações de configuração não mantém o registro do estado e das características do nó destinatário, ou seja, o nó de destino se responsabiliza por se autoconfigurar enquanto o nó origem apenas informa as características da rede. Por padrão, endereço autoconfigurado por este modo criará um endereço *Unicast Link-Local* formado pela junção do prefixo (FE80::/64) com o identificador da interface física da máquina. Há várias implementações para a geração desse identificador, a mais comum é baseada no endereço MAC. Se houver um *daemon Router Advertisement* configurado e executando em um enlace, indica que é um procedimento utilizado por roteadores para transmitir informações aos dispositivos, o que engloba desde propriedades do enlace, da rede, de DNS, MTU, de prefixos, dentre outras, as quais serão processadas e adicionadas às configurações dos dispositivos.

O nó ao enviar uma requisição com mensagem *Router Solicitation* aos roteadores deste enlace, também criará automaticamente um endereço *Unicast* do tipo *Globals*, utilizando o prefixo de subrede de 64 *bits*, características informadas pela mensagem *Router Advertisement*. A geração do identificador de interface, os 64 *bits* restantes, pode também ser obtida do endereço MAC do nó, podendo ainda usar uma geração aleatória desses 64 *bits* [14]. Ainda se tratando de mensagens *Router Advertisement*, outra alternativa é por iniciativa dos próprios roteadores que periodicamente enviam estas mensagens para anunciar sua presença na rede e orientar os nós para autoconfiguração. Com o recebimento da mensagem *Router Advertisement*, em qualquer dos modos, é iniciado o processo de autoconfiguração, caso este processo não tenha ocorrido anteriormente. Após a conclusão do processo de geração do endereço, este necessita ser confirmado como único no enlace

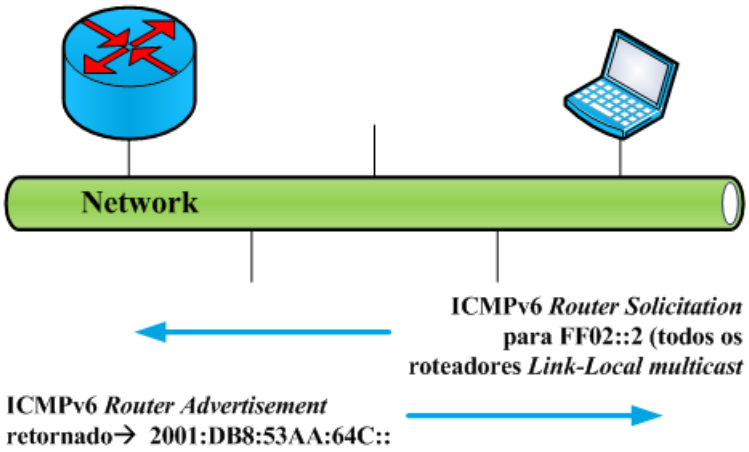


Figura 2.7: Exemplo de Autoconfiguração de *host*

antes de ser adicionado à interface. Assim, o método de detecção de endereços duplicados entra em ação, para só a partir desta confirmação, permitir que o nó se comunique no enlace [19].

Como pode ser notado por meio de ilustração adaptada na Figura 2.7 [27], os *hosts* enviam uma mensagem *Router Solicitation* para todos os roteadores usando o endereço *multicast* (FF02::2) que indica todos os roteadores, e uma mensagem *Router Advertisement* é devolvida com informações sobre a rede em questão, para posterior autoconfiguração do dispositivo.

Stateful: os dispositivos obtêm endereços ou configurações de um servidor que mantêm uma base de dados com todos os endereços que foram distribuídos na rede. Esse tipo de autoconfiguração permite que dispositivos clientes obtenham endereços, bem como outras configurações, de um servidor centralizado, como ocorre no protocolo IPv4 com a utilização de um servidor DHCP, e no protocolo IPv6 com o DHCPv6 (*Dynamic Host Configuration Protocol for IPv6*). Configurações *Stateful.* são frequentemente empregadas quando há uma necessidade de maior rigor no controle com referência aos endereços alocados nos dispositivos, tendo a preocupação principal de manter os endereços únicos. Dependendo das políticas da administração de redes, pode ser necessário que alguns endereços sejam alocados para dispositivos específicos de modo permanente [27].

A autoconfiguração de endereços em modos *Stateless* e *Stateful* podem ser combinados. Por exemplo, um *host* pode utilizar autoconfiguração de endereço *Stateless* para gerar um endereço IPv6, mas, em seguida, usar o DHCPv6 para os parâmetros adicionais [2].

DHCPv6

O DHCPv6 definido na RFC 3315 [28] é a versão DHCP para o protocolo IPv6. Devido a recursos de autoconfiguração de endereços *Stateless* no IPv6, o DHCPv6 apresenta

algumas importantes diferenças se comparado ao seu predecessor, o DHCPv4 (*Dynamic Host Configuration Protocol for IPv4*). As duas versões fornecem autoconfiguração *Stateful* e registro automático de *host* DNS. O DHCPv6 usa as portas 546 e 547 UDP, já o DHCPv4 utiliza as portas 67 e 68 UDP [2].

Em cada rede deve haver ao menos um servidor DHCPv6 capaz de enviar dados para os clientes se configurarem. Normalmente, os dispositivos comunicam em seu *Link-Local* com o servidor DHCPv6 ou por meio de *relay agents* (*All_DHCP_Relay_Agents_and_Servers*), usando o endereço FF02::1:2 de *Link-Local*, mas, outros endereços podem ser utilizados dependendo do servidor. O *relay agent* supracitado corresponde a um endereço *Multicast* com escopo *Node-Local* usado para que clientes enviem mensagens aos roteadores, e aos servidores de destino que se localizam na vizinhança. Em algumas redes simples, não há

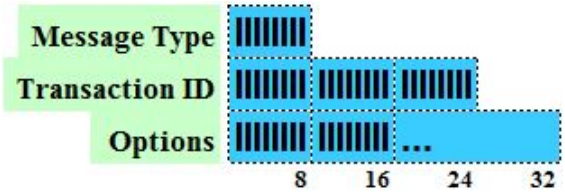


Figura 2.8: Formato do pacote DHCPv6

necessidade do uso do DHCPv6 por causa do recurso de autoconfiguração de endereços *Stateless*. No entanto, DHCPv6 é uma solução que possibilita que nós IPv6 aprendam automaticamente os endereços IPv6 e os servidores DNS. Para as redes que trabalham simultaneamente com as duas versões do protocolo IP, não há conflito entre DHCPv4 e DHCPv6, e ambos podem existir mesmo em um único nó. Neste caso, o lado IPv4 de um nó teria sua configuração IPv4 do servidor DHCPv4, e do lado IPv6 do nó teria sua configuração IPv6 do servidor DHCPv6.

Com o DHCPv6, o administrador de rede consegue melhorar significativamente o controle sobre a distribuição de identificadores de interface, muito mais eficiente que com a autoconfiguração de endereços *Stateless*. E nem todas as comunicações DHCPv6 precisam ocorrer dentro do mesmo enlace, precisando para tal, utilizar roteadores para retransmitir tanto as mensagens do cliente quanto as do servidor.

Tal como acontece com DHCPv4, *relay agents* são usados para permitir que os dispositivos se comuniquem com servidores DHCPv6 remotos. Isso ainda é feito via UDP, mas usando um endereço *Multicast* de escopo *Site-Local* (FF05::1:3), que é usado apenas por *relay agents* denominados por *All_DHCP_Servers*, utilizados pelos roteadores para se comunicarem com os servidores DHCPv6 ao retransmitirem as mensagens recebidas dos clientes.

Há 13 tipos de mensagens no protocolo DHCPv6 que podem ser utilizadas para troca de informação entre clientes e servidores, com ou sem roteadores no meio do caminho:

- ***SOLICIT*** (1): enviada por um cliente para localizar um servidor DHCPv6;
- ***ADVERTISE*** (2): enviada pelo servidor DHCPv6 como resposta à mensagem *SOLICIT* de cliente;
- ***REQUEST*** (3): enviada por um cliente a um servidor DHCPv6 para solicitar dados de configuração;
- ***CONFIRM*** (4): enviada por um cliente a um servidor DHCPv6 para verificar se endereço e parâmetros de configuração permanecem válidos para uso no enlace;
- ***RENEW*** (5): enviada por um cliente a um servidor DHCPv6 para estender o tempo de vida do seu endereço e atualizar outros parâmetros de configuração;
- ***REBIND*** (6): enviada por um cliente a qualquer servidor DHCPv6 para estender o tempo de vida do seu endereço e atualizar outros parâmetros de configuração, isto quando sua alocação estiver próximo de expirar e não ter recebido uma resposta da mensagem *RENEW*;
- ***REPLY*** (7): enviada pelo servidor DHCPv6 como resposta às mensagens *SOLICIT*, *REQUEST*, *RENEW* e *REBIND* de cliente com um *Rapid Commit Option*. Um *REPLY* a uma mensagem *INFORMATION-REQUEST* contém somente parâmetros de configurações, mas nenhum endereço IP. Um *REPLY* a uma mensagem *CONFIRM* contém uma confirmação ou negação de que o endereço IP do cliente ainda é válido para o enlace. Por fim, um servidor DHCPv6 envia uma mensagem *REPLY* para informar que recebeu as mensagens *RELEASE* e *DECLINE*;
- ***RELEASE*** (8): enviada por um cliente a um servidor DHCPv6 que lhe concedeu endereço IP, para indicar que deixará de usar o endereço alocado;
- ***DECLINE*** (9): enviada por um cliente a um servidor DHCPv6 para informar que um ou mais endereços que foram transmitidos para autoconfiguração já está(ão) sendo utilizado(s) no enlace;
- ***RECONFIGURE*** (10): enviada pelo servidor DHCPv6 a cliente já configurado para informar que o servidor possui novas informações de configuração ou sofreu atualização. Assim, o cliente inicia atualização por meio de transações *RENEW/-REPLY* ou *INFORMATION-REQUEST/REPLY*;
- ***INFORMATION-REQUEST*** (11): enviada por um cliente a um servidor DHCPv6 solicitando parâmetros adicionais de configurações, sem informações de endereço IP;
- ***RELAY-FORW*** (12): é enviada por um roteador por meio de *relay agent* para encaminhar mensagens para os servidores DHCPv6, seja diretamente ou através de

outro *relay agent*. A mensagem recebida, uma mensagem de um cliente ou uma mensagem RELAY-FORW de outro relay agent, é encapsulada em uma opção na mensagem de *RELAY-FORW*;

- ***RELAY-REPL*** (13): enviada pelo servidor DHCPv6 aos clientes por meio de *relay*. Esta mensagem pode ser retransmitida entre roteadores até alcançar o cliente, uma vez que, a mensagem do cliente apresenta-se encapsulada nas opções da mensagem *RELAY-REPL*. O último roteador deve extraí-la e enviá-la ao cliente.

O pacote DHCPv6 é muito simples, todas as mensagens trocadas entre clientes e servidores que se encontrem no mesmo enlace utilizam um formato geral, um cabeçalho fixo com uma parte variável para opções, como pode ser notado na ilustração adaptada pela Figura 2.8 [2].

O campo *Message Type* é composto de 8 *bits* e define o tipo de mensagem dentro do protocolo, delimitando assim as opções da mensagem no campo *Options*. Para cada *REQUEST* o cliente gera um novo código de identificação da transação e registra no campo *Transaction ID*, campo composto de 24 *bits*, possibilitando que em um fluxo de mensagens seja possível saber se a mensagem é uma resposta a uma solicitação específica. Quanto ao campo *Options*, é formado por um tamanho variável, sendo usado para fornecer informações de parâmetros de configuração.

Entretanto, quando na comunicação entre clientes e servidores há roteadores utilizando *relay agents*, os pacotes precisam passar por uma transformação antes de serem enviados pelo servidor. Os pacotes alterados possuem o formato demonstrado na ilustração adaptada pela Figura 2.9 [2].

A composição do campo *Message Type* continua sendo de 8 *bits*, determinando o tipo de mensagem dentro do protocolo, o que permite delimitar as opções da mensagem no campo *Options*. Neste campo, quando especificado o valor 12, indica o uso de mensagem *RELAY-FORW*, e quando especificado o valor 13, a mensagem a ser usada passa a ser *RELAY-REPL*.

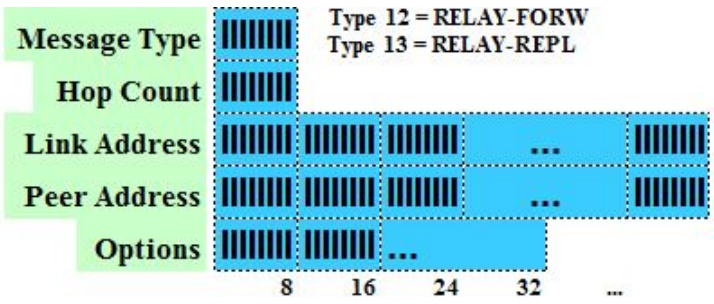


Figura 2.9: Formato do pacote DHCPv6 em *Relay Agents* e mensagens de servidores

O campo *Hop Count* composto de 8 *bits* é responsável por contabilizar a quantidade de roteadores atravessados antes que a solicitação alcance o servidor DHCPv6. Baseado no campo *Link Address* (128 *bits*), em uma mensagem *RELAY-FORW*, o servidor pode identificar o enlace de localização do cliente que executou uma solicitação. Neste campo contém o endereço *Global* ou *Site-Local* para localização do cliente. Já no campo *Peer Address* (128 *bits*), contém o endereço do cliente ou do roteador que enviou a mensagem. Por fim, o campo *Options* que também tem tamanho variável, é utilizado para encaminhar informações extras que auxiliam no mecanismo de autoconfiguração [14].

2.3.4 Roteamento no IPv6

O roteamento é o processo utilizado na Internet para encaminhamento de pacotes entre redes. Os protocolos de roteamento IP se dividem entre IGP (*Interior Gateway Protocol*) que foi projetado para uso dentro de um AS (*Autonomous System*), ou seja, entre os roteadores que são controlados pela mesma empresa ou organização, e que incluem protocolos como RIP (*Routing Information Protocol*), IS-IS (*Intermediate System to Intermediate System*) e OSPF (*Open Shortest Path First*). O outro tipo é o EGP (*Exterior Gateway Protocol*), que foi projetado para troca de rotas entre ASs, como entre operadoras de rede, onde encontra-se inserido o protocolo BGP (*Border Gateway Protocol*) [29].

Para suportar o IPv6, estes protocolos de roteamento supracitados precisaram passar por adequações, principalmente a respeito da acomodação do tamanho dos endereços IP. Esta seção cobrirá sucintamente os protocolos de roteamento OSPF e BGP, uma vez que são os que mais contribuirão para o desenvolvimento do estudo em questão.

OSPF

Projetado para ambientes de rede TCP/IP, o OSPF é um protocolo do tipo *link-state* que envia avisos sobre o estado da conexão a todos os outros roteadores em uma mesma área hierárquica [7]. O OSPF usa o algoritmo SPF (*Shortest Path First*), baseado no algoritmo de Dijkstra para a escolha do melhor caminho e permite agrupar os roteadores em áreas, trabalhando de forma hierárquica, dividindo os roteadores de uma rede em diversas áreas, como se vê na Figura 2.10. A cada uma dessas áreas é atribuído um identificador único (*Area-ID*) de 32 *bits* e todos os roteadores de uma mesma área mantém um banco de dados de estado separado, de modo que a topologia de uma área seja desconhecida fora dela. Isso reduz o volume de tráfego de roteamento entre diferentes partes da rede. A área identificada pelo ID 0 (ou 0.0.0.0) área de *backbone* é a responsável por difundir as informações de roteamento. Em rede onde não existem tais divisões, a área de *backbone* é a única a ser configurada.

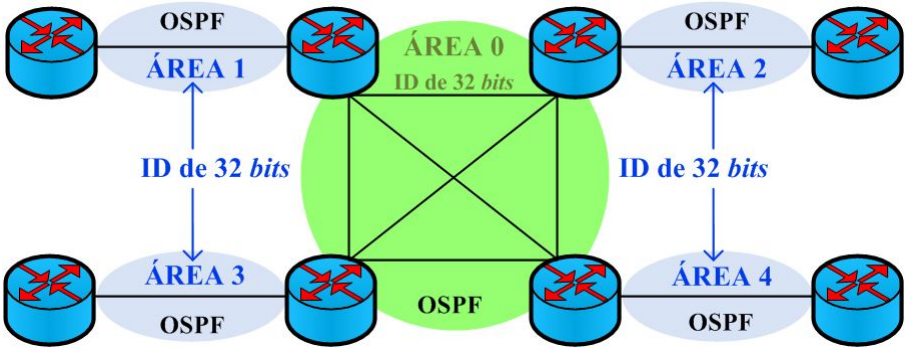


Figura 2.10: Exemplo de topologia de rede sob protocolo OSPF

O OSPF versão 3 definido na RFC 5340 [30] é um protocolo utilizado unicamente em redes IPv6 e foi baseado na versão do OSPF versão 2 , utilizada em redes IPv4. Deste modo, em uma rede com Pilha Dupla, é necessário utilizar tanto OSPFv2, para o roteamento IPv4, quanto o OSPFv3, para o roteamento IPv6. A maioria dos algoritmos do OSPFv2 foram preservados no OSPFv3. No entanto, algumas mudanças foram necessárias referentes à semântica do protocolo entre IPv4 e IPv6 ou simplesmente quanto ao aumento do tamanho do endereço IPv6 [30]. O protocolo OSPFv3 é processado por *link*, e não por

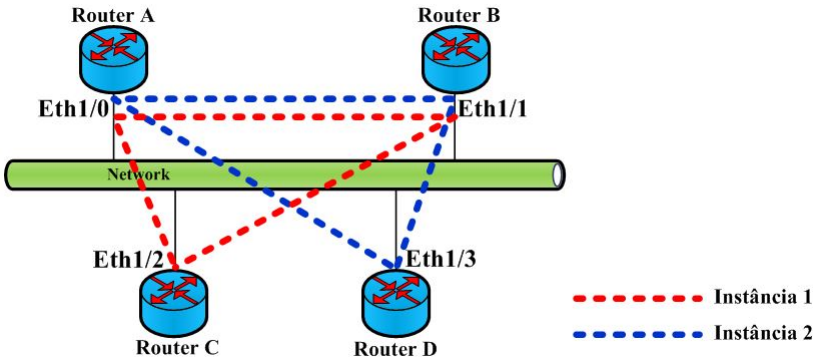


Figura 2.11: Exemplo de múltiplas instâncias do OSPF em execução em um *Link*

subrede como no OSPFv2. O termo *link* é usado para indicar um meio sobre o qual os nós podem se comunicar na camada de enlace [12]. Múltiplas instâncias do OSPFv3 podem ser executadas em um *link*, como pode ser visto na Figura 2.11 e exemplificado na sequência.

Conectados à mesma rede os *routers* A, B, C e D compartilham o mesmo *link* e podem estabelecer relações de vizinhança entre eles. Na primeira instância do OSPFv3 são configuradas as interfaces Eth1/0, Eth1/1 e Eth1/2 respectivamente nos *routers* A, B e C. Já na segunda instância do OSPFv3 configuradas as interfaces Eth1/0, Eth1/1 e Eth1/3 respectivamente nos *routers* A, B e D. As relações de vizinhança da primeira

instância são estabelecidas entre os *routers* A, B e C, enquanto que na segunda instância são estabelecidas entre os *routers* A, B e D.

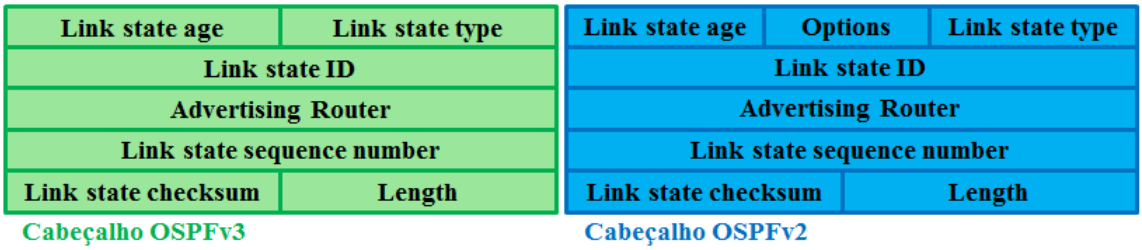


Figura 2.12: Formatos de cabeçalhos dos protocolos OSPFv3 e OSPFv2

O OSPFv3, assim como o OSPFv2, processa pacotes que são enviados somente ao longo da vizinhança, com exceção do pacote *Hello*, que é usado para descobrir vizinhos. As funções e tipos de pacotes são os mesmos em ambos IPv4 e IPv6, codificada pelo campo *Link state Type* do cabeçalho padrão do pacote OSPF, este apresentado por meio de ilustração adaptada pela Figura 2.12 [14], na qual nota-se que o campo *Options* do OSPFv2 foi removido do OSPFv3. Na sequência seguem relacionados os campos do cabeçalho OSPFv3 [30]:

- ***Link state age***: aponta o tempo em segundos desde que o LSA foi originado;
- ***Link state type***: aponta a função desempenhada pelo LSA;
- ***Link state ID***: identificador de origem do roteador para o LSA. A combinação de *Link state ID*, *Link state type* e *Advertising Router* identificam unicamente o LSA na base de dados de estado de *link*;
- ***Advertising Router***: indica o *Router ID* do roteador que originou o LSA;
- ***Link state sequence number***: instâncias sucessivas de um LSA são dadas por sucessivos números de sequências de estado de *link*. O número de seqüência pode ser usado para detectar casos de LSA antigos ou duplicados;
- ***Link state checksum***: aponta a verificação pelo algoritmo Fletcher *checksum* do conteúdo completo do LSA;
- ***Length***: aponta o tamanho em *bytes* do LSA.

Assim como no OSPFv2, o OSPFv3 opera com 5 tipos de pacotes [30]:

- ***Hello packets***: pacote periodicamente enviado para estabelecer e manter o relacionamento com os vizinhos;
- ***Database Description***: estes pacotes são trocados depois que uma vizinhança é iniciada e descreve o conteúdo da base de dados do estado de *link*;

- ***Link State Request***: após a troca de pacotes *Database Description* com um roteador vizinho, um roteador pode encontrar alguns LSAs faltantes. Para obter esses LSAs, o roteador envia pacotes *Link State Request* que carregam o resumo desses LSAs para o vizinho;
- ***Link State Update***: cada pacote transporta uma coleção de LSAs;
- ***Link State Acknowledgment***: são enviados para reconhecer os LSAs recebidos.

BGP

O BGP como referido anteriormente, é o acrônimo de *Border Gateway Protocol*, usado para roteamento entre Sistemas Autônomos. Ele trabalha transmitindo informações sobre quem pode alcançar qual prefixo CIDR, em essência quais endereços, e por qual rede [13]. Na ilustração adaptada Figura 2.13 [7], mostra a interação entre protocolos IGP e EGP, no qual o BGP é utilizado entre 02 (dois) ASs e o OSPF para comunicação entre roteadores internos nas redes. Não há atualmente uma versão específica de protocolo BGP para

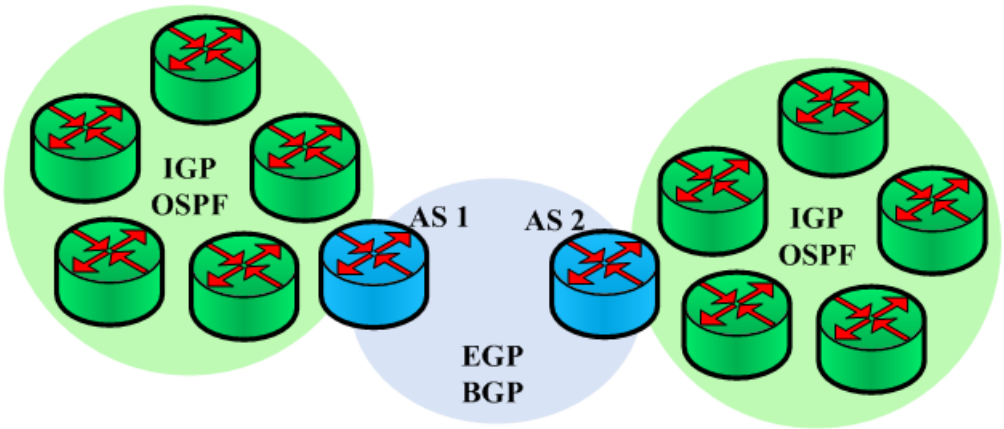


Figura 2.13: Interação entre protocolos de roteamento BGP e OSPF

operação em IPv6. A versão 4 do BGP (BGP-4) definido na RFC 4271 [31], suporta somente o IPv4, mas há definições de extensões de multiprotocolos do BGP-4 (BGP-4+) designados na RFC 4760 [32] que permitem o suporte ao IPv6 e outros protocolos [2].

2.3.5 Segurança com IPv6

Para o propósito do estudo em questão, e por ser um tema que exige uma abordagem pontual e aprofundada, a segurança no tráfego IPv6 será tratada de forma sucinta e baseada em alguns focos de grande relevância, como Endereços Temporários, os Endereços Criptograficamente Gerados, e o IPSec.

Endereços Temporários

Os endereços temporários podem auxiliar na segurança da arquitetura de rede, possuem curta duração e são alterados de tempos em tempos. Em geral, não é possível distingui-los dos endereços públicos. Por meio da Figura 2.14, imagem gerada no ambiente de laboratório, com o prefixo de rede 2001:DB8:CAFE::/64 vê-se a autoconfiguração de dois endereços IPv6 com sufixos diferentes, sendo um originalmente gerado através do endereço MAC (vermelho) e outro temporário gerado aleatoriamente (amarelo).

A RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, define formas de gerar e alterar esses endereços temporários através de extensões de privacidade. As condições importantes são que a sequência de endereços temporários escolhidos para uma interface deve ser totalmente imprevisível, sem um esquema específico, e ter baixa probabilidade de colidir com as escolhas feitas por outras interfaces [29].

Em alguns sistemas operacionais estas extensões de privacidade são adotadas por entender que usar o endereço MAC das interfaces de rede no próprio endereço IPv6 é um risco de segurança que atinge a privacidade dos usuários, uma vez que fica mais fácil rastrear a máquina do usuário independente da rede em que ele esteja [15]. O uso de

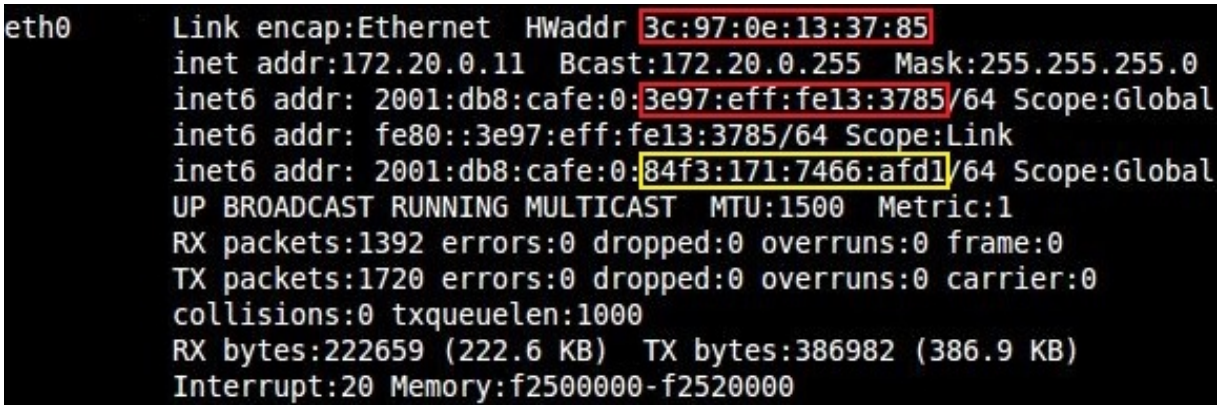


Figura 2.14: Exemplo de geração de endereço temporário em sistema operacional Linux

atribuições de endereços temporários pode ser de muita utilidade para a privacidade. Em muitos casos em que as empresas administram suas próprias redes, atribuições de endereço por meio do DHCPv6 pode ser preferível.

Endereços Criptograficamente Gerados

Os Endereços Criptograficamente Gerados, referenciados constantemente pelo acrônimo CGA (*Cryptographically Generated Addresses*), também conhecidos como endereços baseados em *hash*, fornece um método de garantir que o originador de uma mensagem de

Neighbor Discovery é o dono do endereço contido na mensagem. Como se vê na ilustração adaptada pela Figura 2.15 [29], a idéia é escolher um par de chave pública e privada adequado para criar uma assinatura digital com uma chave privada e então as verificar com a chave pública. A chave pública, junto a outros parâmetros, são usados para gerar uma identificação de interface, a chave pública é inserida dentro da mensagem, e a mensagem é marcada com a chave privada. A chave pública pode ser usada para verificação de endereços e assinatura. Um atacante sem a chave privada não pode forjar a assinatura da mensagem.

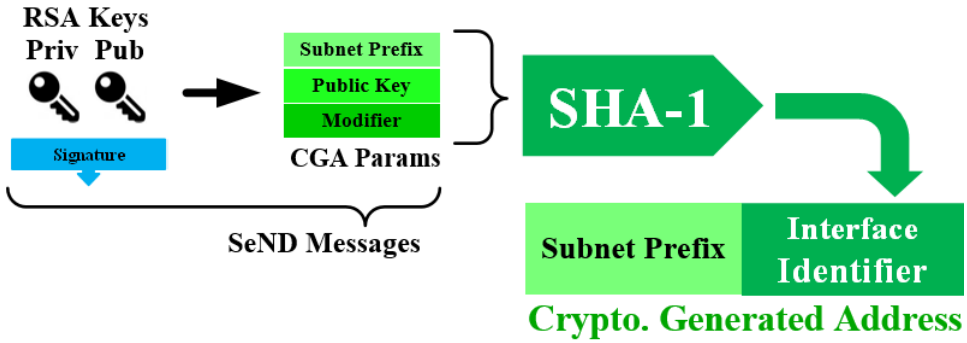


Figura 2.15: Geração de endereço criptográfico com par de chave público-privada

O CGA pode ser usado para aumentar a segurança em uma rede IPv6, mas em muitos casos este método não tem sido percebido como vantajoso, além de gerar mais *overhead* do que o IPSec ou protocolos de segurança de camadas mais altas. O CGA tem sido padronizado como a principal segurança na criação de blocos para o *Secure Neighbor Discovery* do IPv6.

IPSec

O modelo de segurança com IPSec, já é familiar para muitos por ela ser a base de muitos sistemas VPN (*Virtual Private Network*) que já encontram-se implantados. A arquitetura do IPSec é de considerável complexidade, e é descrita na RFC 2401 [33]. No IPv6, o IPSec é implementado com o uso de cabeçalho de extensão, e utilizado para encriptar e autenticar pacotes IP. Ele é projetado para proteger qualquer tráfego da aplicação por estar estabelecido na camada de Internet (rede). Há três principais protocolos que o IPSec usa para realizar suas funções [34]:

- ***Security Association*** (SA): este gera as chaves de encriptação e autenticação que serão usadas pelo IPSec. Uma vez executado papel do SA, a segurança dos serviços é garantida pela utilização dos protocolos de segurança (AH, ESP, ou ainda de ambos);

- ***Authentication Header*** (AH): este fornece autenticação, integridade e proteção anti-repetição para o pacote inteiro (o cabeçalho IP e o *payload*). Ele não fornece confidencialidade, o que significa que ele não criptografa os dados. Os dados são legíveis, mas protegidos contra modificação. O cabeçalho AH usa algoritmos de *hash* com chave para assinar o pacote para fins de integridade;
- ***Encapsulating Security Payload*** (ESP): este provê a garantia de privacidade dos dados, utilizando criptografia para que apenas o destino seja capaz de decodificar as informações. Também fornece um mecanismo opcional próprio de autenticação, para o caso de o AH não estar sendo utilizado. É formado por três campos, sendo eles um cabeçalho ESP, uma cauda ESP e um campo referente a autenticação opcional.

Dependendo das necessidades da rede, o AH e o ESP podem ser usados em conjunto para fornecer tanto autenticação quanto privacidade, no entanto, eles podem operar separadamente, sendo apenas um deles suficiente para atender a maioria dos casos. Além disso, o IPSec pode ser implementado seguindo dois modos diferentes de operação, denominados modo de transporte e modo de tunelamento.

No IPv6, no modo de transporte, os cabeçalhos de segurança são adicionados depois do cabeçalho IP, encriptando apenas do cabeçalho de transporte em diante. Dessa forma, a origem e destino do pacote ficam visíveis na rede pública. Já no modo de tunelamento os cabeçalhos de segurança são adicionados na frente de todo o pacote original, que é totalmente encriptado, e então é adicionado na frente dos cabeçalhos de segurança um novo cabeçalho IP. Portanto, um túnel IPSec é estabelecido entre a origem e destino deste novo cabeçalho IP, por onde irão trafegar os pacotes IP encriptados. As extremidades do túnel IPSec podem ser roteadores encarregados de lidar com os protocolos do IPSec e posteriormente encaminhar os pacotes desencriptados aos seus sistemas finais, que não precisam estar a par da utilização do IPSec [14].

2.4 Resumo do Capítulo

Neste capítulo foram abordadas as principais características a respeito do protocolo IP em suas versões 4 e 6, com exposição teórica sucinta sobre suas características e as fundamentais evoluções da versão 4 para a 6. Em conformidade com o estudo em questão, o protocolo IPv6 foi alvo de uma dedicação e exposição de maior abrangência, com atenção especial ao conjunto de protocolos que amparam desde suas funcionalidades básicas a particularidades concernentes ao IPv6.

Capítulo 3

Transição entre os protocolos IPv4 e IPv6

Neste capítulo serão abordadas as principais técnicas disponíveis e amplamente utilizadas na transição entre os protocolos IPv4 e IPv6. Uma palavra chave na fase de transição é a interoperabilidade entre estes protocolos, as duas versões devem coexistir na rede simultaneamente se comunicando. Nesse intuito, nos idos anos de 2005 a IETF por meio da RFC 4038 [35] intitulado por *Aspectos de Aplicação da Transição IPv6* apresentou o desenvolvimento de mecanismos que intentam na colaboração para a transição de forma gradativa e suave [35]. O objetivo não é promover uma transição de forma abrupta de todo o ambiente de rede IPv4 para IPv6, isso levaria a um transtorno incomensurável, e com pouca probabilidade de sucesso.

Nesse documento, são definidos mecanismos em 3 grupos principais, o conhecido como Tunelamento ou *Tunneling* que permite o transporte de tráfego IPv6 sobre a infraestrutura de IPv4 existente o Tradução ou *Translation* que permite nós somente com IPv6 se comunicarem com nós somente IPv4 e por fim o Pilha Dupla ou *Dual Stack* que permite que IPv4 e IPv6 coexistam nos mesmos dispositivos e redes.

3.1 Tunelamento ou *Tunneling*

O mecanismo de Tunelamento pode ser usado para implantar uma infraestrutura de encaminhamento IPv6 enquanto a infraestrutura global de IPv4 ainda é a base [2]. Esta técnica pode ser usada para transportar o tráfego IPv6 através do encapsulamento em pacotes IPv4 de forma a viabilizar a transmissão sobre a infraestrutura IPv4. Por exemplo, se um provedor de acesso à Internet tem uma infraestrutura somente com IPv4, o Tunelamento permite que se tenha uma rede IPv6 e túnel através dessa rede de provedor IPv4 para chegar a outros nós ou redes IPv6.

O Tunelamento é um mecanismo pelo qual um protocolo é encapsulado em outro protocolo para ser transportado através de uma rede onde o protocolo original normalmente não é suportado ou que seja processado de alguma forma indesejada. O tunelamento do IPv6 no IPv4 (*6in4*) é normalmente feito simplesmente adicionando um cabeçalho IPv4 antes do pacote IPv6. O resultado do pacote é então encaminhado para o endereço de destino informado no cabeçalho IPv4, e nesse destino, o cabeçalho externo é retirado, e o pacote é processado como se tivesse sido recebido através de um ambiente IPv6 habilitado [15]. Uma ilustração adaptada desse processo pode ser visto na Figura 3.1. Também é possível, de forma análoga, encapsular pacotes IPv4 em pacotes IPv6, técnica conhecida como IPv4 em IPv6 (*4in6*).

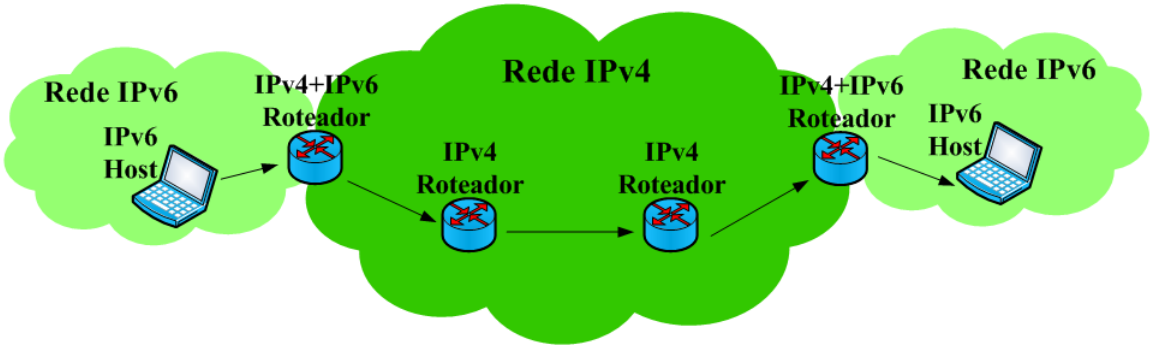


Figura 3.1: Tunelamento de pacotes IPv6 através de rede IPv4

Definições em RFCs descrevem dois tipos de Tunelamento. Um é o Tunelamento Configurado e o outro Automático, respectivamente de natureza estática e dinâmica. O primeiro necessita de administradores dos sistemas para configurar os pontos finais do túnel. Com o segundo, os nós são autoconfiguráveis. Geralmente, em ambientes corporativos é usado o Tunelamento Configurado para infraestruturas de túneis, enquanto a forma de Tunelamento Automático são aplicados em *hosts* para tunelar o retorno das ilhas IPv4 ou IPv6. Atualmente, existem diferentes mecanismos de Tunelamento Automático, abaixo seguem alguns:

- ***6over4***: para uso de *host* para roteador ou de roteador para *host*;
- ***6to4 e 6rd***: para uso de roteador para roteador;
- ***ISATAP***: para uso de Tunelamento *intra-sites*;
- ***Teredo***: para encapsulamento UDP destinados para tunelamento através de NAT IPv4.

No aspecto segurança do Tunelamento, é necessário fazer algumas considerações, o primeiro é quanto às extremidades do túnel, estas são sempre um ponto de foco para a

segurança, são alvos frequentes de ataques. Potencialmente, o tráfego de qualquer lugar pode chegar a pontos de extremidade do túnel. O segundo é a respeito de inspeção do tráfego encapsulado, pois não se deve encará-los como confiáveis sem uma devida inspeção, isto requer examinar o tráfego IPv6 dentro de pacotes IPv4 e submetê-lo ao controle de segurança local. Aplicando as políticas de segurança em ambas as entradas e saídas do túnel, irá proteger o túnel de prováveis ataques.

Por último, as ACLs (*Access Control Lists*) nos roteadores IPv4 podem ser incapazes de bloquear o acesso à rede de endereços IPv6 específicos ou de blocos inteiros de endereços IPv6, eles podem não ter habilidade de reconhecer os números dos campos *next header* do IPv6, o tipo de mensagem ICMPv6, ou número de porta para TCP ou UDP sobre IPv6. O motivo é que o cabeçalho IPv6 e endereços não são visíveis para o roteador, eles são parte do pacote de *payload*, que é a intenção funcional de qualquer processo de encapsulamento [29].

3.2 Tradução ou *Translation*

No domínio das redes IPv4, a Tradução, conhecida há muitos anos pela técnica NAT foi definida para mapeamento entre endereços privados dentro de uma rede e um endereço público em direção ao mundo exterior, afim de sanar a crescente escassez de endereços IPv4 públicos.

Assim, com objetivo de fornecer conectividade para muitos dispositivos, o NAT não mapeia apenas os endereços dos dispositivos, mas também utiliza portas para cada endereço para mapear múltiplos endereços privados para um endereço público. Esta é uma técnica tipo *stateful*, pois o *gateway* precisa manter o estado para rotear corretamente pacotes de retorno. Atualmente, em uma orbe onde os protocolos IPv4 e IPv6 necessitam de interação, este tipo de NAT é referenciado como NAT44 (*Network Address Translator from IPv4 to IPv4*).

O fato é que a técnica de Tradução apresenta-se como uma importante tecnologia para auxiliar na transição do IPv4 para o IPv6, como suporte à administração do acelerado crescimento da Internet. Mesmo que novos usuários da Internet só obtenham endereços IPv6, eles não são capazes de acessar conteúdos sobre as predominantes redes IPv4, exceto, se existir um elemento com papel de Tradução. Com isso, a IETF decidiu definir métodos padrões de Tradução para impedir a proliferação de métodos independentes e sem padrões globais. O mecanismo de Tradução consiste em transformar pacotes IPv4 em IPv6 e vice-versa, para que possam ser roteados ou transmitidos em uma rede [2].

A técnica de Tradução pode ocorrer em diferentes camadas da pilha de protocolo, dentre elas, Internet, Transporte e Aplicação, e nesta seção, diferentes métodos para implementações da Tradução serão abordadas [27].

O método SIIT (*Stateless IP/ICMP Translation*) que encontra-se na camada Internet e definido na RFC 2765 [36], permite a substituição de cabeçalhos IPv4 para IPv6 de modo recíproco. A Tradução no nível IP é relativamente simples, o campo *Time to Live* é copiado para o *Hop Limit*, os *bits* do campo *Type of Service* copiados para *Traffic Class*, o comprimento do *payload* é recalculado e campos fragmentados podem ser copiados para um cabeçalho fragmentado se necessário. Este método também traduz mensagens ICMP e adiciona um pseudo-cabeçalho ICMP com intuito de verificação da integridade, se necessário, ele fragmenta e reagrupa as mensagens.

O NAT-PT (*Network Address Translation-Protocol Translation*), usa o método SIIT e possibilita a comunicação entre sistemas e serviços de ambos protocolos IPv6 e IPv4, e assim amplia o conjunto de possibilidade de interoperação entre estes protocolos. Como define a RFC 4966, o NAT-PT tem aplicação limitada, proporcionando mais benefícios na transição por tornar possível a conectividade de aplicações legadas que podem nunca terem suporte a IPv6.

Outro método que atua na camada Internet é o BIS (*Bump in the Stack*), definido na RFC 2767 [37], é muito similar ao NAT-PT combinado com o SIIT, mas a motivação é ligeiramente diferente. Usado em casos onde há a necessidade de migração para o IPv6 de algum sistema, mas não se consegue uma versão do sistema compatível com o IPv6.

O BIS, no entanto, apresenta algumas limitações, a comunicação é restrita a um único sentido, partindo de um *host* IPv4 à outro IPv6, além de que, em comunicação entre *hosts* IPv4, há a necessidade de uso de Tradução em algum ponto das aplicações envolvidas. Outro fator de limitação, é que não funciona em comunicações *multicast*.

A operação do método BIS é desempenhada de forma que, o *software* ao realizar uma consulta DNS sobre um *host* IPv6, recebe do BIS um endereço IPv4 privado para representar tal *host*, e o *software* pode utilizar o endereço designado normalmente. Os pacotes destinados a este endereço IPv4 são interceptados pelo BIS, que utiliza o mecanismo de SIIT para traduzi-los e enviá-los à seu destino IPv6. A ação de retorno do pacote acontece de forma semelhante, com o BIS traduzindo os pacotes para IPv4.

O TRT (*Transport Relay Translator*) proposto na RFC 3142 [38], atua na camada de Transporte, e permite que *hosts* IPv6 troquem tráfego (TCP ou UDP) com *hosts* IPv4, tendo como vantagem o fato de poder ser implementado sem modificações extras tanto nos *hosts* IPv6 quanto em *hosts* IPv4. Para tal, apenas necessita de um elemento que trabalhe com ambos protocolos IPv4 e IPv6 inserido em um ponto intermediário da rede. Apesar de ser um método simples, este suporta a maioria das aplicações amplamente

utilizadas, além de possuir alta escalabilidade.

No método TRT, quando um *host* IPv6 precisa se comunicar com um *host* IPv4, o *host* IPv6 deve inserir um prefixo falso ao endereço IPv4 que deseja alcançar. O pacote contendo tal prefixo é interceptado quando passa pelo TRT, para ser traduzido e encaminhado ao destino [27].

Por fim, o método BIA (*Bump in the API*) definido pela RFC 3338 [39], opera de forma similar ao BIS, exceto pelo fato de que a Tradução ocorre em uma camada mais alta, a de Aplicação. Seu objetivo principal é o mesmo do BIS, isto é, permitir que aplicações em IPv4 se comuniquem com *hosts* IPv6 sem qualquer modificação da aplicação IPv4. Entretanto, enquanto o BIS opera em sistemas apenas em IPv4, o BIA requer que o sistema possua as duas versões do protocolo IP.

O método BIA atua incluindo uma API (*Application Programming Interface*) de Tradução entre o *socket* API e os módulos TCP/IP da pilha dupla de um *host*, com a intenção de traduzir as funções do *socket* IPv4 em funções do *socket* IPv6, e vice-versa, promovendo a comunicação entre aplicações IPv4 e IPv6 [27].

Em se tratando de métodos de Tradução, é relevante citar também o IVI, criado inicialmente para possibilitar que servidores somente em IPv6 pudessem comunicar-se com a Internet em IPv4. Para tal, um endereço IPv4 é atribuído virtualmente ao dispositivo, utilizando-se um mecanismo de Tradução de pacotes em modo *stateless* [14].

A compreensão do conceito IVI fica mais claro ao se pressupor que ele cria um *host* IPv6 espelho para o IPv4 e um *host* IPv4 espelho para o IPv6, considerando que um *host* espelho é um endereço que simula a presença do dispositivo na rede, mas que na verdade encaminha os pacotes enviados a ele para o *host* real por meio da Tradução *stateless*.

O servidor ou usuário IPv6 nativo na rede onde o IVI encontra-se implementado, embora não possua um endereço IPv4 atribuído a si, é alcançado por um *host* IPv4 na Internet através de seu endereço espelho e, de forma semelhante, alcança um *host* IPv4 qualquer na Internet através de seu endereço IPv6 espelho.

Algumas considerações são necessárias a respeito da Tradução, ela não é recomendada como estratégia para conduzir uma transição do IPv4 para o IPv6 por várias razões. Traduzir endereços IPv6 para IPv4, efetivamente rejeita muitos dos motivos convincentes para migração para o IPv6. A Tradução não soluciona, por exemplo, o problema da exaustão do espaço de endereçamento IPv4. No entanto, um sistema implementado apenas sobre o IPv6 não pode comunicar com outro sistema em operação apenas sobre o IPv4, a menos que a Tradução ocorra em algum lugar, por isso, a Tradução é necessária para manter aplicações legadas em IPv4 isoladas executando de forma diferente no universo do IPv6 [29].

É importante salientar que a Tradução pode resultar em perda de características relevantes quando não há um mapeamento claro entre os recursos fornecidos pelo mecanismo. Por exemplo, a Tradução de um cabeçalho IPv6 em um cabeçalho IPv4 pode levar à perda da etiqueta de fluxo IPv6 que acompanha a funcionalidade.

Embora apresente como uma tecnologia eficaz, a técnica de Tradução não suporta algumas características avançadas do IPv6, tal como os serviços de segurança, controle de acesso, confidencialidade e integridade de dados, além de impor algumas limitações à estrutura topológica da rede, pois qualquer mensagem enviada pelo elemento tradutor deverá retornar pelo mesmo elemento.

3.3 Pilha Dupla ou *Dual Stack*

Um *host* em Pilha Dupla tem suporte para os protocolos IPv4 e IPv6, sendo também referido como um *host* IPv6/IPv4 e tem, pelo menos, um endereço para cada versão de protocolo. Em comunicação com outro *host* IPv6, tal *host* se comporta como um *host* somente IPv6, e em comunicação com um *host* IPv4, ele se comporta como um *host* somente IPv4. Um *host* em Pilha Dupla utiliza mecanismos IPv4 para configurar um endereço IPv4 (configuração estática ou DHCP) e utiliza mecanismos IPv6 para configurar um endereço IPv6 (configuração estática, SLAAC ou DHCPv6).

Implementações ainda podem permitir habilitar ou desabilitar um dos protocolos de um *host* em Pilha Dupla. Por meio da ilustração adaptada na Figura 3.2 [40] é apresentado de forma sucinta um cenário de rede em Pilha Dupla.

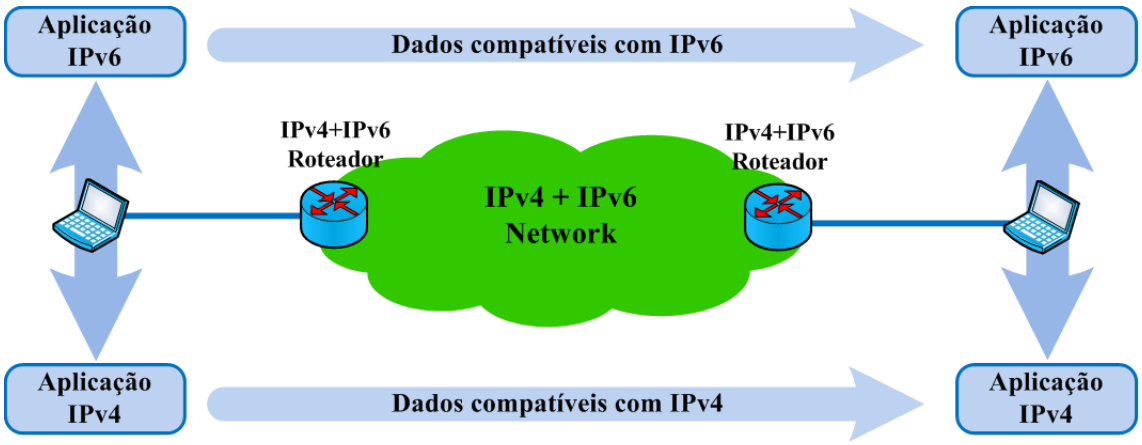


Figura 3.2: Cenário de rede em Pilha Dupla

O DNS é utilizado em ambas versões do protocolo para resolver nomes e endereços IP. Nos *hosts* em Pilha Dupla é necessário que o DNS seja capaz de resolver diferentes tipos de registros de endereços IPv4 e IPv6.

Dependendo de como um serviço é alcançado, sobre IPv4, ou IPv6 ou ainda por ambos, o DNS pode retornar somente em IPv4, ou somente em IPv6 ou ambos. Mecanismos de seleção de endereços padrão e perfis podem garantir que as conexões sejam estabelecidas de forma eficiente em qualquer caso, tendo como exemplo, o *Happy Eyeballs*, que será abordado a frente [2].

Um *host* em Pilha Dupla deve incluir o código na camada Internet da pilha de protocolo TCP/IP para processar ambos pacotes IPv4 e IPv6. Normalmente, há um único enlace disponível para enviar e receber pacotes IPv4 ou IPv6, e neste enlace, ainda encontra-se os protocolos ARP do IPv4 e o ND do IPv6.

Na camada de Transporte há pequenas diferenças na forma como os pacotes IPv4 e IPv6 são tratados, principalmente relacionado à forma de cálculo *checksum* dos protocolos TCP e UDP, que abrange os endereços de origem e destino do cabeçalho IP, o que obviamente é diferente nas duas versões do protocolo IP.

Já na camada de Aplicação, o código pode fazer chamadas para rotinas no *socket* API do IPv4, nos *sockets* API básicos e avançados do IPv6. Funções de *sockets* IPv4 irão acessar o lado IPv4, enquanto as funções de *socket* IPv6 acessarão o lado IPv6. Na Figura 3.3 [27], é apresentada uma ilustração adaptada do modelo de Pilha Dupla [14].

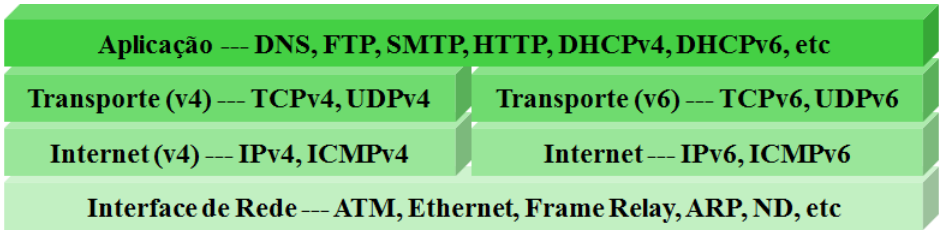


Figura 3.3: Modelo de Pilha Dupla

Uma vantagem significativa desta tecnologia é que ela executa os protocolos IPv4 e IPv6 de modo nativo. Uma vez a infraestrutura habilitada para execução de Pilha Dupla, é viável dar início à migração das aplicações do IPv4 para o IPv6 gradualmente, o que possibilita atualização por pequenas partes.

O propósito do método de Pilha Dupla é reduzir quanto possível o número de túneis usados no processo de transição [29]. Uma implementação sem o uso de Tunelamento e nem Tradução, é o melhor cenário para o desempenho, escalabilidade e eficiência, além de não haver necessidade de projetar, testar e executar mecanismos temporários de transição que precisam ser removidos mais tarde. Uma vez toda a rede ter sido atualizada para o protocolo IPv6, basta desabilitar o protocolo antecessor.

Esta tecnologia exige uma atualização de rede completa para possibilitar a execução das duas pilhas de protocolos IP. As tabelas de roteamento são mantidas simultaneamente com roteamento IPv4 e IPv6. No aspecto segurança é preciso também existir dois con-

ceitos, um para o IPv4 e outro para o IPv6, pois, são duas incursões na rede, e estas precisam ser tratadas no requisito política de segurança.

Quanto ao gerenciamento da rede, apresenta ser mais complicado executar *troubleshooting*, e em alguns sistemas operacionais, dependendo do protocolo utilizado, os comandos podem ser separados, e é preciso mais memória e poder de processamento para execução de Pilha Dupla [2].

Utilizar pilha dupla pode não ser possível em todas as ocasiões. Por exemplo, quando não há mais IPv4 disponíveis e o provedor precisa atender a usuários novos com IPv6 e IPv4. Para redes corporativas que já utilizam NAT isso não é um impendimento, o IPv6 nativo pode ser utilizado em conjunto com o IPv4 compartilhado. Outra situação que dificulta a implantação do IPv6 usando pilha dupla é a existência de equipamentos que não o suportam e que não podem ser facilmente substituídos.

Entretanto, vantagens da técnica de Pilha Dupla como novos elementos de rede já poderem ser endereçados em IPv6 e os elementos já existentes poderem ser migrados em fases sem grandes impactos, o gerenciamento da técnica ser mais fácil, além da infraestrutura interna oferecer condições favoráveis à implementação da técnica, estes já são aspectos de grande relevância para uma escolha segura.

3.4 Considerações sobre as Tecnologias de Transição

Com o teor já exposto, é notável que o principal objetivo na abordagem de Transição entre os protocolos IPv4 e IPv6 é de permitir que as duas versões se comuniquem sem afetar o funcionamento da Internet, porém este fato não é algo tão simplista, pois não se pode mudar a estrutura atual com o IPv4 para a nova estrutura com o IPv6 de uma só vez. As organizações ou operam redes em Pilha Dupla ou precisarão acomodar tanto o IPv4 e o IPv6 em rede através de outros meios. No entanto, o objetivo central da Transição é de alcançar uma única rede IPv6, ou pelo menos com o IPv6 como o protocolo principal.

Dentre as tecnologias de Transição apresentadas, nenhuma cobre todas as necessidades ou é ideal para todos os cenários de redes, mas possibilitam muitas combinações e maneiras de se garantir conectividade entre as versões do protocolo IP. Designar uma ou várias delas está diretamente relacionado ao seu domínio de aplicabilidade, suas propriedades, a infraestrutura existente e as metas estabelecidas para transição, devendo ser capaz de prover a interoperabilidade entre os protocolos e satisfazer às necessidades da rede relacionadas à disponibilidade, desempenho, escalabilidade e a segurança [2].

3.5 IPv6 em *Hardwares* e Sistemas Operacionais

Um planejamento antecipado de migração para o IPv6 com ciclos de atualizações, estes geralmente realizados por meio de atualizações com o término do ciclo de vida regular do produto, pode levar a um custo final menor nos investimentos em *hardware* e *softwares*. Em muitos casos, o IPv6 não é parte do *hardware* e pode ser instalado como parte do sistema operacional ou como uma atualização de *software*. Se as funções do protocolo IP são implementadas puramente em *hardware* ou se o *software* do sistema não pode ser atualizado, o *hardware* deve ser substituído para suportar o IPv6.

Atualmente, os sistemas operacionais são em sua maioria fornecidos com IPv6 e sem custo extra. Nos sistemas operacionais modernos, o IPv6 já vem habilitado por padrão, por exemplo, uma versão atualizada de Windows, Linux, MacOS, AIX ou BSD, o IPv6 já vem ativado.

No âmbito dos *hardwares*, mais especificamente dos relacionados à infraestrutura de rede, os roteadores e *switches* devem fornecer um nível equivalente de suporte ao IPv6, não sacrificando os recursos avançados do IPv6 nem comprometendo o *throughput* da rede.

3.6 Migração de Aplicações para o protocolo IPv6

A implementação do protocolo IPv6 na rede é perfeitamente factível, e muitas aplicações de uso abrangente e regular já executam sobre este novo protocolo, tais como, *web*, correio eletrônico, FTP, SSH, dentre outros. Entretanto, muitas aplicações não executam sobre esta nova realidade do protocolo IP, sejam elas desenvolvidas pela própria organização ou por terceiros.

Aplicações desenvolvidas por terceiros se tiverem suporte a IPv6, normalmente as utiliza em redes IPv4 com o IPv6 desabilitado, em caso de não suportar o IPv6, é preciso que conste em *roadmap* a previsão de disponibilidade nas versões de atualização.

Os cenários elencados abaixo são frequentemente encontrados nas organizações que passam pelo processo de migração:

- Aplicações IPv4 sobre nós em Pilha Dupla;
- Aplicações IPv6 sobre nós em Pilha Dupla;
- Aplicações que suportam IPv4 e IPv6 sobre nós em Pilha Dupla;
- Aplicações que suportam IPv4 e IPv6 sobre nós somente IPv4;
- Aplicações que suportam IPv4 e IPv6 sobre nós somente IPv6.

O desafio para os desenvolvedores é a criação de aplicativos que funcionam bem em todas as situações. Nomes DNS devem ser usados sempre que um serviço tem que ser chamado, então, é preciso entrar com nomes de serviços no DNS, com os correspondentes tipos de registros de recursos para ambiente de rede IPv4 e IPv6 no DNS. Um nó resolvendo um nome DNS e obtendo vários endereços na resposta deve julgá-los e ter um mecanismo para escolher a conexão com o melhor desempenho, a exemplo do *Happy Eyeballs* [2].

Assim, ter habilitado o IPv6 na rede mas não ter aplicações para uso sobre esta nova rede IPv6, é inútil. Então, é preciso iniciar o trabalho com aplicações no início do processo de habilitação do IPv6 na rede. Se a idéia central é de que a migração para o IPv6 precisa ocorrer em larga escala, as aplicações executadas em sistemas de *desktops*, *laptops* e em muitos aparelhos móveis, precisam trabalhar tão bem com o IPv6 quanto com o IPv4 [40].

3.7 Resolução de Nomes no IPv6

Para o protocolo IPv6, é mais importante do que nunca que os nomes, em vez de endereços, sejam usados para fazer referência a recursos de rede. O endereço IPv4 já é suficientemente difícil de ser lembrado com uma série de quatro números decimais, como um endereço IPv6 pode ter até 32 dígitos hexadecimais, se tornou uma tarefa ainda mais difícil. Aliás, com uma mistura de ambos endereços IPv4 e IPv6, ao especificar um nome a um recurso, permite que o sistema operacional escolha o melhor conjunto de endereços com os quais se comunicar. Logo, o suporte a resolução de nomes para endereços IPv6 com o DNS é uma parte de extrema importância de uma implantação IPv6.

A RFC 3596 [41] define extensões DNS para suporte ao IPv6, adicionando o registro **AAAA**, também conhecido como **quad-A** para a resolução de um nome de domínio totalmente qualificado para um endereço IPv6. Os registros **AAAA** são comparáveis com endereços de *hosts* **A** que designam registros de recursos usados pela resolução de nomes do IPv4. O tipo de registro de recurso é nomeado **AAAA** porque os endereços IPv6 de 128 *bits* são quatro vezes maior que o tamanho do endereço IPv4 de 32 *bits*.

Um registro de recurso **AAAA** em um arquivo base de um DNS típico tem uma estrutura composta de um nome de domínio totalmente qualificado e um endereço IPv6 associado ao nome [42]. Abaixo segue um exemplo de um registro de recurso **AAAA**: **host.alpha.com IN AAAA 2001:DB8::1:DD48:AB34:D07C:3914**

Registros **A** e **AAAA** podem tranquilamente coexistir lado a lado, assim, possibilita a implantação generalizada de ambos registros para o mesmo nome de *host*, e esta é a direção em que os registros DNS caminham. Desta forma, se um *host* opera em pilha dupla, po-

dem ser anexados os registros **A** e **AAAA** ao seu nome de domínio, como mostrado abaixo: **host.alpha.com IN A 64.4.10.56 IN AAAA 2001:DB8::1:DD48:AB34:D07C:3914**

No entanto, é preciso de cuidado com essa configuração, alguns resolvedores de nomes atuais sempre irão procurar registros **AAAA** antes de registros *A*, mesmo se o *host* que está executando o resolvedor não tenha a capacidade de se comunicar com todos os endereços IPv6, por exemplo, um *host* que tem somente um endereço IPv6 *Link-Local*, ou usa alguma tecnologia de transição que lhe dá conectividade IPv6 limitada.

Contudo, é prudente anexar os registros **A** e **AAAA** para nomes de domínios diferentes, pelo menos aos *hosts* que proveem serviços, como mostrado abaixo: **host.alpha.com IN A 64.4.10.56 host-v6.alpha.com IN AAAA 2001:DB8::1:DD48:AB34:D07C:3914**

Uma solução potencial para os problemas de priorização e proposto pela IETF é o referido na RFC 6555 [43] e conhecido como **Happy Eyeballs**, e que é uma forma de tentar corrigir o problema de decisão sobre qual conexão preferir, em casos que há a presença de ambos registros **A** e **AAAA**. Se o endereço IPv6 está disponível e capaz de responder rapidamente, a comunicação acontecerá sobre IPv6. Se uma conexão IPv6 não estiver disponível, a comunicação ocorrerá sobre IPv4. Similarmente, se a comunicação IPv6 for mais lenta, talvez por causa de estar tunelada sobre IPv4, a aplicação usa a conexão IPv4 mais rápida [40].

Quanto ao *troubleshooting* relacionado ao DNS, no ambiente IPv6 não é muito diferente do ambiente IPv4, as ferramentas principais no auxílio a esta tarefa continuam sendo o **nslookup** e o **dig** [44].

3.8 Resumo do Capítulo

Neste capítulo as principais técnicas de transição entre protocolos IPv4 e IPv6 foram abordadas. São apresentadas considerações sobre o domínio de aplicabilidade de uma ou mais técnicas nos ambientes de rede, expostas considerações a respeito do protocolo IPv6 em *hardwares*, sistemas operacionais, em aplicações e como a resolução de nomes pode auxiliar na referência a recursos de rede.

Capítulo 4

Estado da Arte

Existem muitas publicações de estudos que colocam em pauta a discussão do processo de transição entre os protocolos IPv4 e IPv6, muitas abordando questões práticas de monitoramento de protocolo IPv6, de tunelamento e tradução, problemas de segurança e desafios fundamentais, dentre outros. Entretanto, poucos trabalhos podem ser encontrados referentes a modelos de implementação em ambientes de redes heterogêneas, sobretudo, no âmbito da esfera de órgãos públicos. Entretanto, três artigos foram selecionados e apresentam correlação clara com a linha de estudo da proposta em questão, sendo apresentados nos parágrafos subsequentes.

Em [1] são apresentados fundamentos da tecnologia de Tradução IVI seguido de exposição de um modelo para implementação do método de transição entre os protocolos IPv4 e IPv6 em servidores de um ISP (*Internet Service Provider*). No artigo [45], foram expostos os principais mecanismos de transição, propondo uma solução para suavizar a transição para IPv6 baseado em túneis e tecnologia de tradução. Já em [46], foram abordados mecanismos principais de transição IPv6 que têm sido propostos introduzindo fundamentos de técnicas de tunelamento e tradução, e analisando o objetivo principal dos túneis e mecanismos de tradução, mapeando por fim, os mecanismos adequados em cenários heterogêneos, discutindo ainda como escolher e implantar os mecanismos de transição.

Na continuação serão descritos os três modelos que foram usados como base para o estudo e produção do modelo objetivo deste trabalho:

Modelo 1

Em [1] foi proposta uma transição do protocolo IPv4 para o IPv6 dos servidores de ISP por meio de um método denominado por Tradução IVI, que sugere uma transição consistente e suave entre as gerações do protocolo IP. Na implantação deste método IVI, o ISP é exigido a ter um ambiente de rede em Pilha Dupla, operando com os protocolos IPv4 e IPv6. Com base na regra de mapeamento de endereço IVI, o encaminhamento ocorre de forma simples, como mostrado na ilustração adaptada na Figura 4.1 [1].

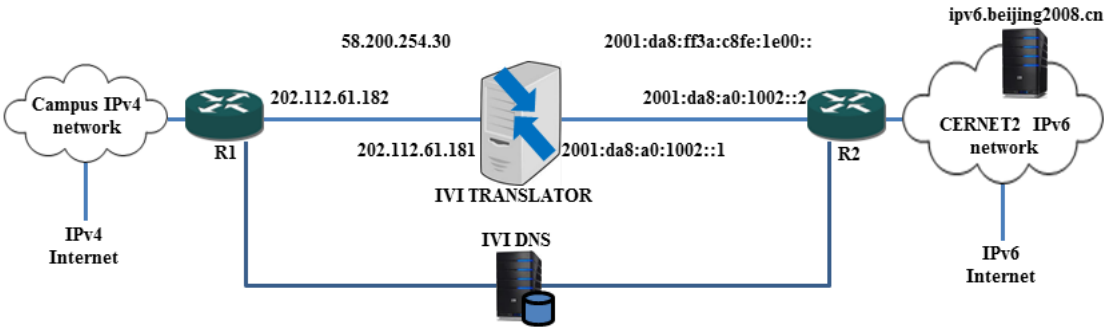


Figura 4.1: Implantação do IVI em ambiente descrito pelo primeiro modelo [1]

Um endereço específico como ipv6.beijing2008.cn é um servidor *web* IPv6 nativo estabelecido no CERNET2 (*China Education and Research Network 2*) com um endereço IPv6 IVI (IVI6) 2001:da8:ff3a:c8fe:1e00::. O endereço virtual IPv4 IVI (endereço IVI4) correspondente é 58.200.254.30 pela regra de mapeamento de endereço. O Tradutor IVI é um roteador em Pilha Dupla com suporte a protocolos de roteamentos estático e dinâmico com duas interfaces, uma da rede IPv4 e a outra para a rede IPv6, sendo possível também, ter uma única interface configurada para ambos.

O Roteador R1 tem uma rota estática IPv4 para endereço IVI4 do servidor *web* com o próximo salto para o IP 202.112.61.181. O Roteador R2 tem uma rota *default* IPv6 ::/0 com o próximo salto para o IP 2001:da8:a0:1002::1. Enquanto isso, o Tradutor IVI tem uma rota IPv6 para endereço IVI6 do servidor *web* com o próximo salto para o IP 2001:db8:a0:1002::2 e outra rota IPv4 para endereço IVI4 0.0.0.0/0 com o próximo salto para o IP 202.112.61.182.

Na solução, um DNS IVI em Pilha Dupla também foi configurado entre os cenários IPv4 e IPv6 no auxílio à resolução de nomes. Quando um cliente IPv6 faz uma consulta a um registro **AAAA** no servidor DNS, este retornará 2001:da8:ff3a:c8fe:1e00::. Quando um cliente IPv4 faz consulta a um registro **A** do servidor, o DNS IVI tentará retornar primeiro o registro **A** do endereço IVI4. Se o registro **A** não existe, o DNS IVI irá consultar o registro **AAAA** e mapeá-lo para um endereço IVI4 e retornar um registro **A** para o cliente IPv4. Assim, permitindo que os usuários dos ambientes IPv4 e IPv6 acessem o servidor **web** ipv6.beijing2008.cn sem qualquer problema.

O estudo levou os autores a algumas conclusões, a citar:

- Pode ser implantado gradativamente para se alcançar a transição completa entre os protocolos IPv4 e IPv6 dos numerosos servidores de ISPs;
- A dependência de uso de DNS foi compreendida como uma deficiência;
- Ainda assim, concluíram que o método de Tradução IVI é uma boa opção para transição dos servidores de ISPs.

Modelo 2

Em [45] foi exposto que diferentes tecnologias de transição devem ser usadas durante diferentes fases de transição. Foi proposta uma solução de transição para o IPv6 com base nas tecnologias de Tunelamento, Tradução e Pilha Dupla, com processo de evolução em 3 fases:

- **Fase 1**: Alguns serviços IPv6 começaram a ser implantados continuamente na rede IPv6, mas não há usuários somente IPv6. Na rede IPv4, *hosts* em Pilha Dupla podem acessar esses novos serviços IPv6, estabelecendo túnel IPv6 sobre IPv4. IPv4-*only* e *hosts* em Pilha Dupla podem acessar os serviços IPv4 em IPv4 nativo;
- **Fase 2**: Nesta fase houve um avanço no número de serviços IPv6 implantados na rede IPv6. Alguns usuários IPv4 começaram a migrar da rede IPv4 para a rede IPv6 como usuários IPv6 IVI com endereços IVI6 específicos. Na rede IPv4, *hosts* em Pilha Dupla podem acessar esses novos serviços IPv6, estabelecendo o túnel IPv6 sobre IPv4. IPv4-*only* e *hosts* em Pilha Dupla acessam os serviços IPv4 em IPv4 nativo. Na rede IPv6, usuários IVI6 podem acessar os serviços IPv4 através da solução de IVI, podendo também acessar serviços IPv6 em IPv6 nativo.
- **Fase 3**: Na fase final alguns antigos serviços IPv4 foram migrados da rede IPv4 para IPv6 como serviços de rede IPv6 IVI, com endereços IVI6 específicos para estes serviços IPv6. Aumentou o número de usuários IVI6 na rede IPv6, correspondente ao número de endereços IPv4 IVI, atingindo o objetivo de migração da rede IPv4 para IPv6. Na rede IPv4, os *hosts* em Pilha Dupla podem acessar esses serviços IPv6 configurando o túnel IPv6 sobre IPv4. Se os serviços IPv6 estão implantados em IVI6, os usuários IPv4 podem acessar esses serviços IVI6 por meio do Tradutor IVI. O IPv4-*only* e os *hosts* em Pilha Dupla podem acessar os serviços IPv4 por IPv4 nativo. Na rede IPv6, usuários IVI6 podem acessar os serviços IPv4 através do Tradutor IVI, comumente usuários IPv6 acessam serviços IPv4 pelo túnel IPv4 sobre IPv6. Todos os usuários IPv6 podem acessar serviços IPv6 em IPv6 nativo. Gradualmente todo o espaço de endereços IPv4 foi convertido em espaço de endereços IPv4 IVI. Nesse caso, usuários e serviços IPv4 foram todos migrados para a rede IPv6, completando o processo de transição.

O servidor de túnel é um *gateway* de rede virtual para cada cliente do túnel, sendo ele responsável pelo roteamento no túnel dos pacotes do cliente e tendo a função de *push*, empurrar informações específicas de configuração de rede. Este servidor ainda precisa configurar um *pool* de endereços IPv6, pois, quando um cliente do túnel se conecta ao servidor de túnel, em primeiro lugar, o servidor precisa verificar a legitimidade do cliente de acordo com o nome de usuário e a senha ou com base em certificado, e então empurra

um endereço IPv6 ou o prefixo IPv6 do *pool* de endereços IPv6 para o cliente de acordo com as informações do usuário.

Foram projetados e realizados dois modos diferentes de trabalho em túnel, os modos *Host* e *Gateway*. O modo *Host* é aquele quando usado processo de autenticação do cliente do túnel, ele terá um endereço IPv6 do servidor túnel, simultaneamente, o servidor do túnel adicionará um registro de rota para este endereço IPv6 através do túnel. No modo *Gateway* um cliente do túnel não necessita de processo de autenticação, ele terá um prefixo IPv6 do servidor de túnel, simultaneamente, o servidor de túnel adiciona um registro de rota para esse prefixo IPv6 através do túnel. Como um *gateway*, clientes do túnel podem distribuir endereços IPv6 com o prefixo IPv6 para os *hosts* em suas subredes IPv4 por meio, por exemplo, de serviço DHCPv6 (*Dynamic Host Configuration Protocol for IPv6*).

Na subrede IPv4, quando um *host* tem um endereço IPv6 do servidor de túnel por meio do DHCPv6, ele pode acessar os serviços IPv6. Os pacotes são roteados através do adaptador de rede virtual, o processo de túnel terá esses pacotes IPv6 a partir do adaptador de rede virtual, usando o protocolo SSL (*Secure Socket Layer*) para montar esses pacotes IPv6 em dados da camada de aplicação, e os pacotes serão enviados como pacotes IPv4 através do adaptador de rede físico. Finalmente um canal de comunicação é estabelecido entre o servidor de túnel e o cliente do túnel e é transparente para os usuários.

Sob a análise dos autores no âmbito do sucesso da transição, um sistema completo de Tradução IVI deve incluir um Tradutor IVI e um DNS IVI [45]. Sendo assim, foi desenvolvido um guia descrevendo as principais configurações de Tradutor IVI e DNS IVI implementadas nos testes, além de sucintamente descrever as ações para configuração de Tunelamento.

Alguns entendimentos finais os autores alcançaram, os quais seguem abaixo:

- O conjunto de soluções propostas e executadas deixou a visão de que a transição ocorreu de forma suave e gradual;
- Num âmbito mais técnico, a Tradução IVI não pode traduzir o endereço IP na camada de aplicação.

Modelo 3

Em [46] os autores centralizaram esforços iniciais numa descrição objetiva mas bem esclarecedora das características do protocolo IPv6 sempre fazendo referências ao seu antecessor IPv4 com abordagem dos problemas de conectividade em redes heterogêneas, bem como expôs também, uma visão geral dos mecanismos principais de transição IPv6 que têm sido propostas. De forma sintética, mapearam os mecanismos adequados em todos os cenários de interconexão heterogênea e travessia heterogênea. Diferenciando-se da maioria das referências encontradas, este estudo abordou mecanismo de uso e estratégias

de implementação de transição para IPv6 na esfera de redes ISPs, redes de *backbones* e redes de borda.

Então para demandas de acesso IPv4, o ISP pode implantar *Dual Stack Lite*, IPv4 sobre IPv6 ou MAP-E (*Mapping of Address and Port-Encapsulation*). O ISP deve escolher entre os três mecanismos candidatos com base na sua situação de endereços IPv4 excedentes e condições da rede. Se a alta taxa de compartilhamento de endereços é a principal exigência, ou a questão NAT em larga escala não é uma grande preocupação, então *Dual Stack Lite* seria uma solução adequada. Por outro lado, IPv4 sobre IPv6 e MAP-E são as opções quando o ISP tem endereços IPv4 suficiente para o conjunto de provisionamento de porta. Se o acesso IPv4 é um requisito comum de assinantes e o ISP é capaz de renumerar a rede IPv6, então MAP-E seria uma escolha melhor, com a grande vantagem de ser *stateless*. Se o ISP quer manter IPv6 e IPv4 desacoplados e trazer nenhuma mudança para a rede IPv6, ou fornecer o acesso IPv4 em um estilo sob demanda, então IPv4 sobre IPv6 se torna a solução mais apropriada.

Ainda em [46], se considera que as redes de *backbones* são responsáveis por encaminhamento IP de alta velocidade, enquanto os serviços de redes específicas são de responsabilidade das redes de borda. Isso reflete também na implantação de transição, o *backbone* se concentra em fornecer transporte IPv4 e IPv6 para redes de borda, levando à atualização do *backbone* para Pilha Dupla, ou a construção de um *backbone* IPv6 independente ao lado do IPv4. A desvantagem também é óbvia, o custo tanto de *upgrade* de *hardware* e operação e gerenciamento são muito altos.

Outra alternativa sugerida para migração em *backbones* consiste em implantar *Softwire Mesh* e fornecer transporte Pilha Dupla na parte superior do *Single Stack* do *backbone*. Para o *backbone* IPv4 existente, pode-se implantar rede IPv6 sobre IPv4 *Mesh*. Para isso, deve-se atualizar os roteadores de borda do *backbone* com funcionamento para os protocolos IPv4 e IPv6 para suportar Pilha Dupla, bem como IPv6 sobre IPv4 *Mesh*. Tráfegos IPv6 podem ser encaminhados no *backbone* por túnel IPv6 sobre IPv4 entre roteadores que suportem as duas versões de protocolos IP. Os roteadores dentro do *backbone* podem permanecer com IPv4-*only* e evitar a atualização. Quanto ao *backbone* IPv6 pode-se implantar IPv4 sobre IPv6 *Mesh* para preservar o transporte IPv4. Apenas um *backbone* físico é usado enquanto o transporte IPv4 e IPv6 são alcançados simultaneamente.

Para redes de borda, por problemas de complexidade e escalabilidade, mecanismos de tradução devem ser implantados em seu interior, em vez de no *backbone*. Na sequência, diferentes estratégias de transição são apresentadas de acordo com diferentes requisitos de comunicação e tipos de rede de borda:

- A rede de borda IPv4 oferece um acesso IPv4 nativo para os usuários finais, e se estes desejam acesso à rede IPv6, o 6RD (IPv6 *Rapid Deployment*) pode ser usado

com o objetivo de permitir ao usuário final ter conexão com as redes IPv6 apesar da rede que provê acesso continue funcionando em IPv4. Nós finais devem trabalhar em Pilha Dupla e suportar funções 6RD em cliente de borda. No lado do ISP, um ou mais 6RD BRs (*Borders Relay*) devem ser implantados como concentradores de túneis;

- Usuários em redes de borda IPv4 também podem querer acessar serviços em redes IPv6 usando IPv4. Entre os mecanismos de transição, o NAT-PT (*Network Address Port Translation-Protocol Translation*) pode satisfazer a demanda, sendo mais utilizado em um ambiente de pequena escala e controlável;

- Usuários em redes de borda IPv6 também podem acessar a rede IPv4 com IPv6, o que exige uma tradução IPv6 para IPv4, o que pode ser realizada pelas técnicas IVI ou NAT64 com função adicional DNS64 (*Domain Name System IPv6-IPv4*) ao sistema DNS. Ao escolher entre IVI e NAT64, os critérios também encontram-se em situação de excesso de endereços IPv4. O IVI fornece comunicação bidirecional com o custo do consumo de endereço por *host*, enquanto NAT64 somente garante comunicação iniciada a partir do lado IPv6, mas alcança compartilhamento de endereços IPv4 dinâmicos.

Demandas dos servidores também devem ser consideradas, estes precisam prestar serviços para clientes IPv4 e IPv6. Se o servidor suporta tanto IPv4 quanto IPv6 na camada de aplicação, então não é necessário qualquer outro mecanismo, caso contrário, deve ser fornecido suporte a transição a nível da aplicação. Se o servidor opera em IPv4, pode-se implantar um tradutor NAT64 para traduzir conexões de clientes IPv6 para IPv4, caso contrário, o servidor executa em IPv6, e pode ser implantado um Tradutor IVI para traduzir conexões de clientes IPv4 para IPv6.

Contudo, algumas conclusões foram alcançadas:

- Para as técnicas de tradução uma questão crítica é a ausência de mecanismo de tradução *stateful* de IPv4 para IPv6;

- Os mecanismos de tradução existentes apresentam problemas de escalabilidade, endereçamento heterogêneo e tradução da camada de aplicação;

- Antes de as técnicas de transição poderem ser aplicadas numa implantação em larga escala, análises de desempenhos quantitativos e sistemáticos devem ser efetuadas antes de qualquer ação;

- Em meio a vários mecanismos de transição para o IPv6, provoca de início no administrador da rede uma certa confusão na escolha da melhor alternativa.

4.1 Conclusões sobre as técnicas utilizadas nos artigos

Mediante análise de trabalhos executados na transição para o IPv6, foi possível notar que os grupos principais de técnicas de transição, a citar Pilha Dupla, o Tunelamento e a Tradução, continuam sendo fortes referências para o alcance de uma transição segura. Dada a constatação, estas técnicas podem ser utilizadas ou combinadas para que em ambiente de rede heterogênea, se possa atingir a meta de experimentos bem sucedidos seguido de um plano de transição gradual e suave tanto para os clientes quanto para os servidores de aplicações.

A proposta elaborada e executada no artigo [1], baseada nas técnicas de Tradução IVI sobre infraestrutura em Pilha Dupla, apresenta-se como uma boa alternativa de transição para o protocolo IPv6 em servidores, principalmente por oferecer implantação *stateless*, altamente incremental e independente de AS. Mesmo sob uma perspectiva de implementação direcionada a ISPs, a linha de trabalho adotada é facilmente adaptável a outros cenários, inclusive ao ambiente acadêmico vivenciado na UnB no âmbito dos servidores que disponibilizam serviços acadêmicos e administrativos.

O plano de implementação demonstrado no artigo [45], concentrou-se numa solução com uso das técnicas de Pilha Dupla, de Tunelamento e Tradução IVI. Mediante constatação de seus executores, mostrou-se viável. Entretanto, é preciso levar em consideração que a implementação de tunelamento apresenta carga adicional colocada no roteador, já que cada ponto de entrada e de saída precisa de tempo e poder de processamento para encapsular e desencapsular pacotes, além de tornar mais complexo processo de *troubleshooting* de rede [2].

O artigo [46] focou em sugerir estratégias de implementação de transição para IPv6 nos cenários de redes ISPs, redes de *backbones* e redes de borda. Dentre estes cenários, as técnicas sugeridas para as redes de *backbones* e borda permitem melhor adequação ao ambiente de rede heterogênea no qual se enquadra a REDUnB, em especial, o uso da técnica de Pilha Dupla em todo os cenários de rede.

4.2 Resumo do Capítulo

Neste capítulo são relatados e analisados três modelos de migração do protocolo IPv4 para o IPv6. Os três modelos em questão referem-se a registros em artigos, seguidos de conclusões sobre as metodologias utilizadas.

Capítulo 5

Proposta de Modelo para Migração Gradual

Este capítulo descreve de forma detalhada a metodologia de migração do protocolo IPv4 para o IPv6 na REDUnB, temática que é a proposta principal deste trabalho. É essencial a compreensão do atual cenário no qual este estudo se baseia, pois, as redes têm a tendência para se tornar mais complexas, devido ao seu crescimento e heterogeneidade de tecnologias utilizadas.

5.1 Ambiente de Aplicação

A REDUnB integrada ao projeto REDECOMEP (Redes Comunitárias de Educação e Pesquisas) é estruturada atualmente sobre uma ampla capilaridade de fibra óptica que alcança regiões significativamente afastadas do campus Darcy Ribeiro, como pode ser notado na Figura 5.1. Como demonstrado, seus campi geograficamente deslocados, como a Faculdade de Ceilândia, Faculdade do Gama, Faculdade de Planaltina, o Núcleo de Práticas Jurídicas em Taguatinga, o Hospital Veterinário na Granja do Torto, a Fazenda Águas Limpas, os Edifícios Anápolis e OK (ambos no Setor Comercial Sul), a Estação Experimental de Biologia, o Centro de Ensino à Distância, estes dois últimos localizados na Asa Norte.

A REDUnB atende a toda a comunidade acadêmica, e atua como um grande provedor de acesso à rede pública e de acesso aos sistemas acadêmicos e administrativos. Por meio da infraestrutura de rede, mantém conectividade permanente com a Internet com endereçamento público.

Esta infraestrutura atende a aproximadamente 17000 (dezessete mil) pontos de acesso cabeados. Como se vê na Figura 5.2, esta rede é alicerçada em seu *backbone* por 4 (quatro) robustos *switches core* (de núcleo) com capacidade de *backplane* de 9.5Tbps, 2.56Tbps em

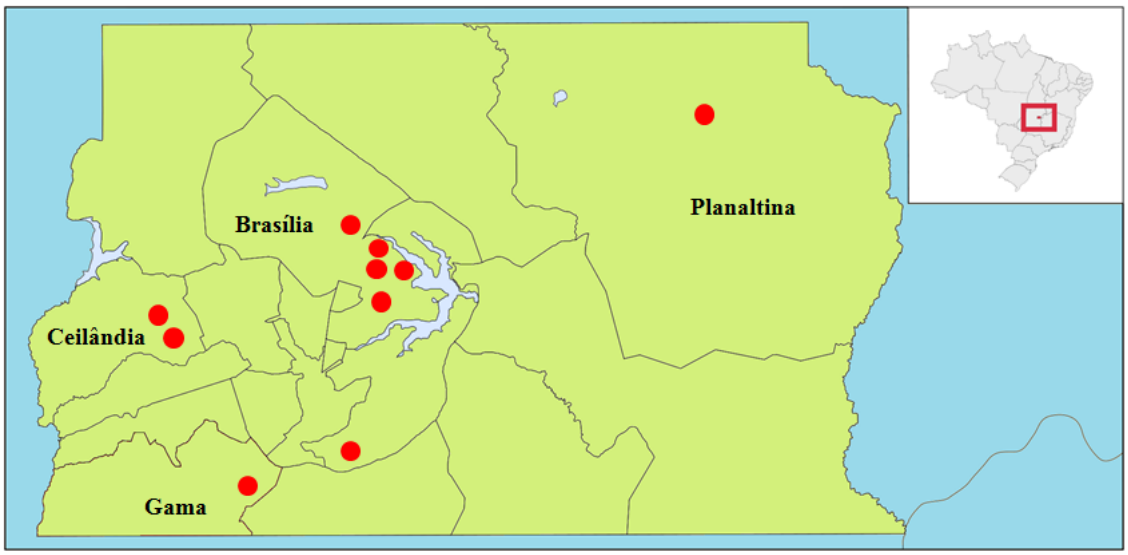


Figura 5.1: Alcance da REDUnB

capacidade de *switching* e suporta *throughput* de 1920 Mpps. Estes são interligados em topologia *full mesh* com enlaces de 10Gbps provendo redundância e tolerância a falhas, característica referenciada nas literaturas com frequência como resiliência. O *backbone* é composto pelos nós identificados por ICC (Instituto Central de Ciências), FT (Faculdade de Tecnologia), FINATEC (Fundação de Empreendimentos Científicos e Tecnológicos) e o CPD (Centro de Informática).

Dos 4 (quatro) *switches core* são distribuídos enlaces de 1Gbps para 93 (noventa e três) unidades de *switches layer* 3 de agregação que possuem alta performance em portas Gigabit Ethernet com capacidade de comutação de 264Gbps. Estes *switches* ficam localizados em salas de distribuição de acesso, de onde partem todo cabeamento de rede para os pontos de acesso. O *switch core* do ICC provê acesso à REDUnB a 37 (trinta e sete) locais, o da FT a 22 (vinte e dois), o da FINATEC a 20 (vinte) e o do CPD a 14 (quatorze) locais. Os *switches* do *backbone* e de agregação são todos equipamentos do mesmo fabricante operando com roteamento estabelecido em OSPF versão 2. Os *switches* de acesso possuem enlaces de 1Gbps com os *switches layer* 3 de agregação. Vale destacar que desde o ano de 2008 as direções do Centro de Informática da UnB têm seguido as orientações do e-PING, onde todos os investimentos em equipamentos do parque de redes de dados precisam já disponibilizar em seu *datasheet* a comprovação de suporte à coexistência dos protocolos IPv4 e IPv6 e a produtos que suportem ambos os protocolos. Contudo, todos os equipamentos envolvidos no cenário supradescrito já possuem características que atendam à orientação do documento de referência do governo federal.

Ainda no contexto da infraestrutura descrita, esta é constituída sobre uma rede IPv4

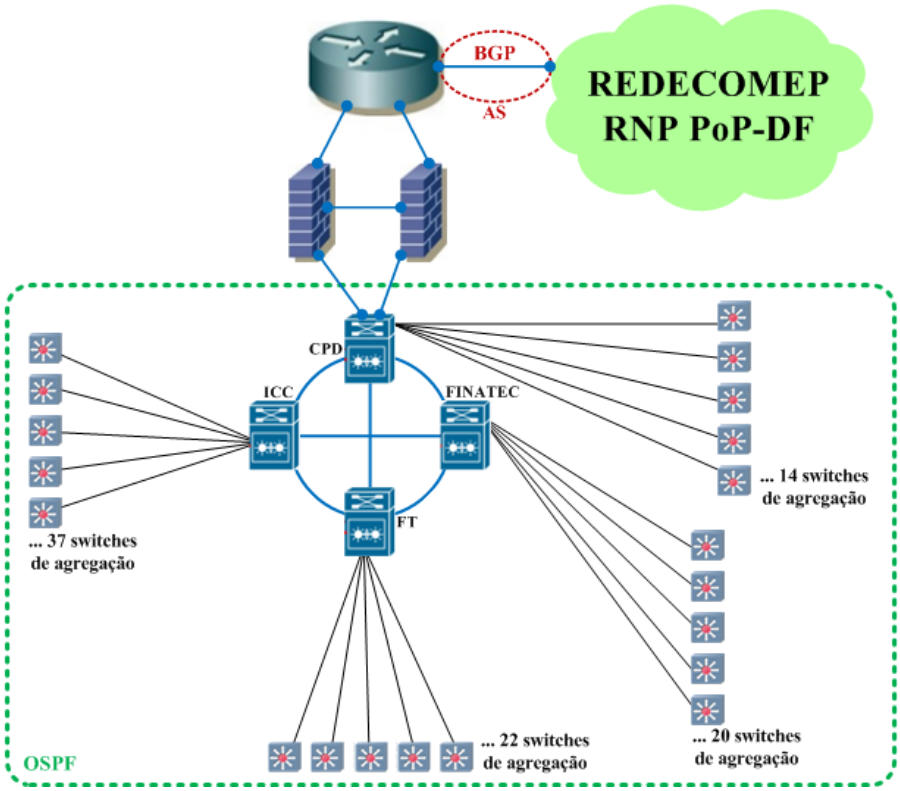


Figura 5.2: Topologia da REDUnB

com endereços públicos classe B (164.41.0.0/16), a qual disponibiliza a alocação de 65.534 (sessenta e cinco mil e quinhentos e trinta e quatro) *hosts* na rede. Este bloco de endereços IPv4 atualmente é segmentado em outros 4 (quatro) blocos uniformes (164.41.0.0/18, 164.41.64.0/18, 164.41.128.0/18 e 164.41.192.0/18), cada um reservado respectivamente para distribuição de redes pelos nós ICC, FT, FINATEC e CPD. Deste bloco de endereços IPv4 /16 sob controle da UnB, em torno de 67% do total encontra-se reservado e em uso dentro da REDUnB, as tendo ainda uma reserva técnica importante de endereços livres para uso.

Quanto ao contexto de endereçamento IPv6, o Registro.br que é um departamento do NIC.br (Núcleo de Informação e Coordenação do Ponto BR) responsável pela distribuição de endereços IPv4 e IPv6 no Brasil, no mês de julho do ano corrente (2014), à pedido do CPD da UnB, atribuiu um bloco IPv6 com prefixo /48 identificado como 2801:80:b90::/48. O bloco de endereços IPv6 em questão disponibiliza 65.536 (sessenta e cinco mil e quinhentos e trinta e seis) redes /64, redes que correspondem a uma quantidade de *hosts* na ordem de 18.446.744.073.709.551.616, um número naturalmente complicado de grafar.

Na saída da REDUnB, há 2 (dois) enlaces de 1Gbps entre o *switch core* do CPD e 2 (dois) *firewalls* operando em *cluster*, ficando estes últimos encarregados pela política de segurança na borda da rede. Concluindo a topologia em questão, os *firewalls cluster* são

interligados a um roteador de borda por 2 (dois) enlaces de 1Gbps, enquanto que a conexão entre o roteador de borda e o ambiente externo (Internet) é por meio da REDECOMEP, em um *link* de 1Gbps, enlace designado como um AS, ou seja, Sistema Autônomo, que utiliza o protocolo de roteamento BGP para a Internet.

O Centro de Informática da UnB, mais especificamente o núcleo de Suporte à Rede e Serviços, é responsável pelo gerenciamento da REDUnB e tem sob sua gerência no que compete a esta rede de dados precisamente 1277 (um mil e duzentos e setenta e sete) equipamentos em operação e gerenciados por uma equipe de 6 (seis) analistas de sistemas. Neste número de equipamentos estão incluídos, 748 (setecentos e quarenta e oito) *switches*, 6 (seis) *firewalls*, 3 (três) roteadores, 10 (dez) controladoras *wireless* e 510 (quinhentos e dez) *access points wireless*.

Ao se tratar da *server farm* da UnB, ou seja, o grupo de equipamentos servidores mantidos pelo Centro de Informática, atualmente sob plataformas que além de possuírem poder de processamento de bom nível, já encontram-se com sistemas operacionais para servidores que suportam o protocolo IPv6. Esta *server farm* mantida em sala cofre, disponibiliza o acesso a diversos serviços (correio eletrônico, HTTP, FTP, DNS, dentre outros), além dos sistemas administrativos e acadêmicos que encontram-se em fase de plena atualização de plataforma. Outro fator importante a ser exposto é que uma pequena parte de sistemas legados encontra-se sobre equipamentos servidores com sistemas operacionais bastante antigos, e consequentemente não suportando o protocolo IPv6.

Ainda com referência aos sistemas operacionais, aqui concernentes aos *hosts* dos usuários, a grande maioria das empresas que os fabricam ou distribuem já oferecem suporte para que seus produtos trabalhem com o protocolo IPv6. Dentro da REDUnB, mediante registro do *Helpdesk* que é o núcleo do Centro de Informática da UnB que dá suporte técnico mais próximo aos usuários da rede cabeada, atualmente a predominância é de sistemas operacionais Windows, e não inferiores ao Windows XP *Service Pack* 1 que já suporta a nova versão do protocolo IP. Em outros sistemas operacionais também frequentemente encontrados na rede, como MacOs, Linux e FreeBSD, estes também já oferecem suporte ao IPv6.

Como já citado anteriormente, as técnicas de transição mencionadas no estudo podem ser implementadas em infraestrutura de rede isoladamente ou em conjunto, dependendo da dimensão e do grau de complexidade do ambiente de rede e sistemas de informática da instituição em questão. Assim, é indispensável analisar o cenário da REDUnB para uma migração suave, segura e que seja exequível.

5.2 Proposta de Implementação

Conforme exposto, destacam-se dois aspectos relevantes para a proposta de modelo, um é que todos os equipamentos (*switches*) que compõem o núcleo da REDUnB já podem operar sobre o protocolo IPv6. O outro é quanto a alta disponibilidade de endereços IPv4 na REDUnB.

Além dos dois aspectos supracitados, outros fatores bases para a escolha do modelo refere-se à alta recomendação dos órgãos gestores da Internet quanto ao uso de Pilha Dupla e a facilidade de gerenciamento do ambiente que é característica da técnica.

Assim, o modelo de migração do protocolo IPv4 para IPv6 dentro da REDUnB consistirá de uma implementação voltada à coexistência das duas versões de protocolos IP nos mesmos equipamentos, de forma nativa, simultaneamente. Inicialmente a intenção concentra-se na implementação da técnica de transição de Pilha Dupla no *backbone* da REDUnB e nos *switches layer* 3 de agregação que roteiam redes para os pontos de acesso.

5.2.1 Fase 1: Endereçamento e Roteamento

Esta fase, começa basicamente com os planejamentos da topologia lógica e física do ambiente de rede, dos endereçamentos IPv4 e IPv6 envolvidos no núcleo da rede, dos blocos de endereços IPv4 (164.41.0.0/16) e IPv6 (2801:80:b90::/48) que estarão disponíveis na borda da rede e do esquema de roteamento das redes deste núcleo em OSPF nas versões 2 e 3, ilustrados pela Figura 5.3. Como se vê, a idéia é manter os ativos de rede

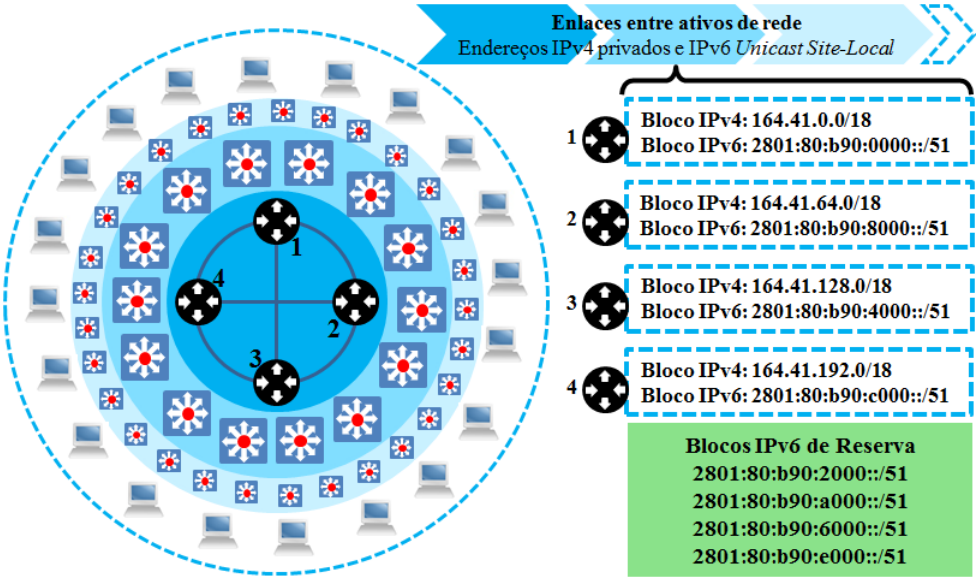


Figura 5.3: Atribuição de blocos de endereços IPv4 e IPv6 na REDUnB

(*switches*, roteadores, *firewalls*, dentre outros) com enlaces de rede utilizando endereços IPv4 privados e no escopo IPv6 usar endereços *Unicast Site-Local*.

No que compete ao roteamento interno da rede, este continua utilizando o protocolo OSPF versão 2 mas com o acréscimo da versão 3 para suporte ao tráfego IPv6, pois, como já exposto, trabalharão de forma independente. Na Figura 5.4 é apresentado como as áreas de roteamento OSPF serão organizadas.

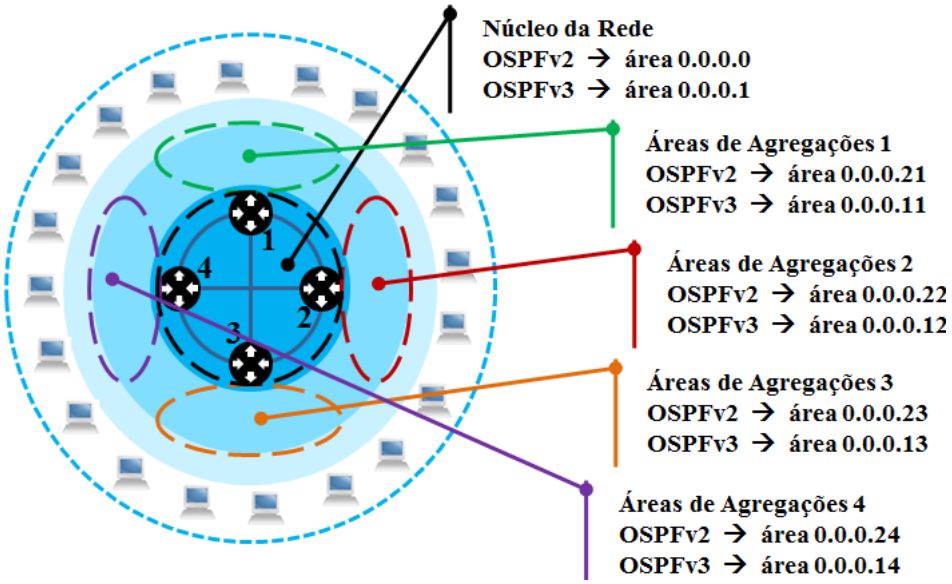


Figura 5.4: Atribuição de áreas de roteamento OSPF versões 2 e 3 na REDUnB

No núcleo da rede (*backbone*) os nós se comunicarão por meio de roteamento OSPF nas áreas 0 e 1, respectivamente para as versões 2 e 3 do protocolo OSPF. Os nós 1, 2, 3 e 4 do *backbone* receberão também respectivamente os pares de áreas 11 e 21, 12 e 22, 13 e 23 e ainda 14 e 24, para propagações de rotas dos *switches* de agregação. Nas áreas de agregações, onde atuam os *switches* da categoria de agregação, serão mantidas as áreas 21, 22, 23 e 24 para o protocolo OSPFv2, respectivamente para os nós 1, 2, 3 e 4 do *backbone*. Para o protocolo OSPFv3, as áreas 11, 12, 13 e 14 nessa ordem para os nós 1, 2, 3 e 4 do *backbone*.

O esquema de divisão do bloco de endereços IPv4 público conforme já praticado dentro da REDUnB, em 4 (quatro) grandes blocos de endereços IPv4 /18 continuará o mesmo. Quanto ao bloco de endereços IPv6 disponível (2801:80:b90::/48), será utilizado método *leftmost* orientado na RFC 3531 [21] e segmentado em 8 (oito) partes como pode ser visto na Figura 5.5.

Este método permite por meio de atribuições esparsas que blocos de endereços reservas permaneçam entre as atribuições, facilitando o futuro crescimento das redes.

Com a divisão em questão, 8 (oito) novos blocos são disponibilizados para uso, e respeitando a orientação do método, como a REDUnB possui 4 (quatro) grandes áreas de

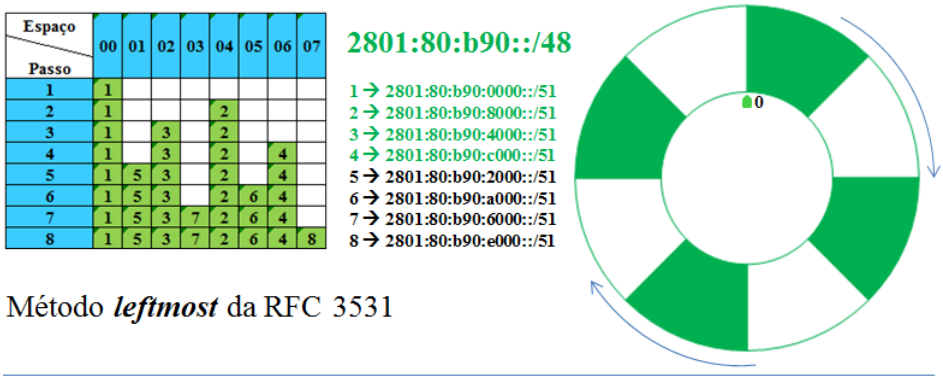


Figura 5.5: Alocação de prefixos IPv6 e áreas de concentradores de redes

concentração de redes (ICC, FT, FINATEC e CPD), os 4 (quatro) primeiros prefixos /51 serão utilizados inicialmente por estas áreas. Cada prefixo /51 possibilita a criação de 8.192 (oito mil e cento e noventa e duas) redes com prefixo /64, este último, é um tamanho mínimo de prefixo para redes orientada pelo CGI.br e por literaturas correlatas. Assim, cada um dos nós do *backbone*, distribuirá também acesso a redes IPv6 para os Institutos, Faculdades, Departamentos mediante a atribuição de endereços IPv6 submetidas ao método *leftmost*.

5.2.2 Fase 2: Organização do Ambiente de Rede

Com o planejamento concluído, é iniciada a etapa de implementação das configurações nos *switches*, com atribuições dos endereços de ambos os protocolos às interfaces VLANs (*Virtual Local Area Networks*) e as configurações de roteamento necessárias que cada protocolo necessita para operar de forma desejada, conforme consta no capítulo sobre a descrição do laboratório.

A habilitação de um serviço de resolução de nomes (servidor DNS e DNS reverso) com suporte a resolução de nomes para os protocolos IPv4 e IPv6 é muito importante para o êxito desse novo universo que se apresenta. Para casos em que um único nome de domínio possua endereços do tipo **A** e **AAAA**, o DNS pode ser configurado de modo a responder utilizando uma ordem já prenunciada. Esta habilidade força a ocorrência de maior tráfego do protocolo escolhido como a opção prevalecente. Em nível de aplicação, alguns ajustes podem ser realizados a fim de priorizar o tráfego de uma versão.

Com a adoção do DNS na solução de migração para o IPv6 puramente em Pilha Dupla, fica muito mais simples para as aplicações decidirem como estabelecer a conexão. Se ela recebe um registro **AAAA**, então a conectividade é estabelecida pelo *socket* v6, e se recebe apenas um registro **A**, ela usa o *socket* v4.

Um aspecto imprescindível para o sucesso da migração para o IPv6 e que precisa estar na linha de ação, é a disponibilização de conteúdos aos clientes da rede de forma a divulgar a disponibilidade de acesso IPv6. Este processo precisa ocorrer de forma gradual e segura. Ainda ao referenciar o cenário descrito sobre o parque tecnológico da UnB, a *server farm* já apresenta características bastante favoráveis para a implementação do protocolo IPv6 por meio de Pilha Dupla, uma vez que os serviços e sistemas disponíveis são alcançados mediante endereços IPv4 públicos, e a REDUnB atualmente não vivencia o tão incômodo esgotamento do seu bloco IPv4.

No processo de migração de conteúdos, os serviços básicos (correio eletrônico, HTTP, FTP, dentre outros) já disponíveis em IPv4, podem receber uma atenção especial e serem ofertados também em IPv6 tão logo este esteja acessível na *server farm*. Os sistemas administrativos e acadêmicos como passam por fase de profundas mudanças estruturais, é desaconselhável qualquer intervenção momentânea em termos de alterações de linhas de código, permanecendo acessível pela rede IPv4 até que se tenha definições mais assertivas à respeito do futuro. Quanto aos sistemas legados e antigos que representam uma fração ínfima do montante dos sistemas, estão na iminência do desuso por ocasião de descontinuidade da solução e substituição por ambiente *web*.

Como a técnica de Pilha Dupla é suportada de forma nativa pelos atuais sistemas operacionais, inclusive os em uso dentro da REDUnB, do lado do cliente a implementação do seu ingresso na rede consiste em atribuir os endereços de ambos os protocolos às interfaces de rede com os seus correspondentes servidores de resolução de nomes (DNS). Uma vantagem do uso de Pilha Dupla também nos clientes é que novos membros na rede já podem ser endereçados com IPv4 e IPv6, enquanto os membros já existentes podem ser incluídos em fases, em pequenas seções de redes, sem grandes impactos.

Assim, o cliente dentro da REDUnB que esteja com equipamento configurado apenas com o protocolo IPv4, quando for comunicar com clientes em Pilha Dupla estabelecerá a comunicação por meio do protocolo IPv4, o mesmo ocorrendo no sentido inverso. De outro modo, quando o cliente estiver operando em Pilha Dupla e o protocolo IPv6 configurado na rede como prioritário, o equipamento o utiliza para comunicar com clientes que estejam também em Pilha Dupla, mantendo a compatibilidade e comunicação durante o período de coexistência, em servidores e em clientes.

Como endereços dos protocolos IPv4 e IPv6 serão atribuídos a cada cliente, tal tarefa pode ser executada de forma manual ou automática (DHCPv4, DHCPv6, mecanismo de autoconfiguração). É inegável a necessidade de administrar a atribuição de endereços IPv4 e IPv6 na REDUnB por meio de mecanismos automáticos, principalmente em extensos ambientes de rede, e certamente o uso dos protocolos DHCPv4 e DHCPv6 podem sustentar uma solução mais escalável e segura. Entretanto, o fato em questão exige um estudo amplo

dos heterogêneos ambientes e clientes que a REDUnB atende, o que cabe a um esforço futuro.

É presumível que neste contexto da migração para o IPv6 que algumas atualizações de *softwares* ou substituições de equipamentos sejam necessárias. Ainda assim, há outras importantes medidas a serem tomadas pelo Centro de Informática da UnB, que garantirão resultados satisfatórios nesse processo de migração:

1. Habilitar a técnica de Pilha Dupla em roteadores e devida adequação de configurações em *firewalls*;
2. Mediar junto à REDECOMEP, mais especificamente ao PoP-DF, o suporte para BGP em IPv6 e garantir redundância deste serviço;
3. Desenvolver avaliações de segurança e implantar políticas de segurança nos diferentes níveis da rede sobre o protocolo IPv6.

Com a técnica de Pilha Dupla implementada, o impacto nas redes atuais é menor, o gerenciamento da rede se torna mais fácil e permite uma implantação gradual, com a configuração de pequenas seções do ambiente de rede de cada vez. Além de ser uma técnica altamente recomendada pelo CGI.br (Comitê Gestor de Internet no Brasil), quando possível.

Na atual fase de implantação do IPv6, não é aconselhável ter clientes com suporte somente à versão 6 do protocolo IP, visto que muitos serviços e dispositivos na Internet ainda trabalham somente com IPv4. Assim, manter o IPv4 já existente funcionando de forma estável e implantar o IPv6 nativamente, para que coexistam nos mesmos equipamentos, é a forma básica escolhida para a transição na Internet.

A técnica de transição de Pilha Dupla pode facilitar ainda o gerenciamento entre as versões dos protocolos IP, por possuir pilhas distintas e funcionais cada uma em seu método de comunicação. Por outro lado, assim que a pilha do protocolo IPv4 e suas redes forem se tornando obsoletas, a técnica de Pilha Dupla simplifica a descontinuação do protocolo IPv4 na rede, necessitando somente desabilitá-lo.

5.3 Resumo do Capítulo

Este capítulo abordou detalhadamente a descrição do ambiente de aplicação do estudo em questão com objetivo de subsidiar a decisão na escolha da técnica de migração do protocolo IPv4 para o IPv6. Na sequência, com o já conhecido ambiente, foi apresentada a proposta de implementação do IPv6 na REDUnB, com a inclusão do planejamento do endereçamento IPv4 e IPv6, o esquema de roteamento e a ordenação da implementação no ambiente de rede.

Capítulo 6

Cenário de Avaliação

Esta seção possui o objetivo de avaliar a proposta descrita. A escolha da avaliação em laboratório real se deve principalmente por já permitir uma visão realística da implementação das versões do protocolo IP na linha de equipamentos (fabricante Enterasys/Extreme) com a qual a REDUnB já opera em nível de ambientes de rede em *backbone*, agregação e acesso.

6.1 Caracterização do Laboratório

O laboratório em questão foi estruturado com topologias física e lógica de rede, de forma a simular o cenário descrito da REDUnB, o qual corresponde a como ela se encontra neste momento, como pode ser notado na Figura 6.1. Como se vê, são 04 (quatro) *switches* que compõem o *backbone*, e simplificadamente 1 (um) *switch* de agregação originado por cada nó do *backbone*. Os nós do *backbone* do laboratório foram interligados fisicamente por topologia de rede *full mesh*. Entre os nós do *backbone* do laboratório foram estabelecidos enlaces de 1Gbps, assim como entre estes nós e os *switches* de agregação.

Os *switches* utilizados no laboratório prático são todos do mesmo fabricante (Enterasys/Extreme), correspondendo ao modelo C5G124-24P2. Este modelo de equipamento é normalmente destinado dentro da REDUnB a perfil de *switch* de agregação, por possuírem alta performance em portas Gigabit Ethernet, além de oferecer suporte a roteamento IPv4 e IPv6. Para o melhor funcionamento dos *switches* em questão bem como para alcançar o comportamento adequado dos protocolos utilizados no laboratório, os seus *firmwares* foram atualizados para uma versão mais atual e estável (c5-series-06.81.02.0007).

Os *switches* do laboratório foram configurados inicialmente com a implementação do *backbone* em IPv4 e posteriormente agregando a interoperação deste com o protocolo IPv6 com o escopo de técnica de Pilha Dupla. Esta fase de implementação das configurações

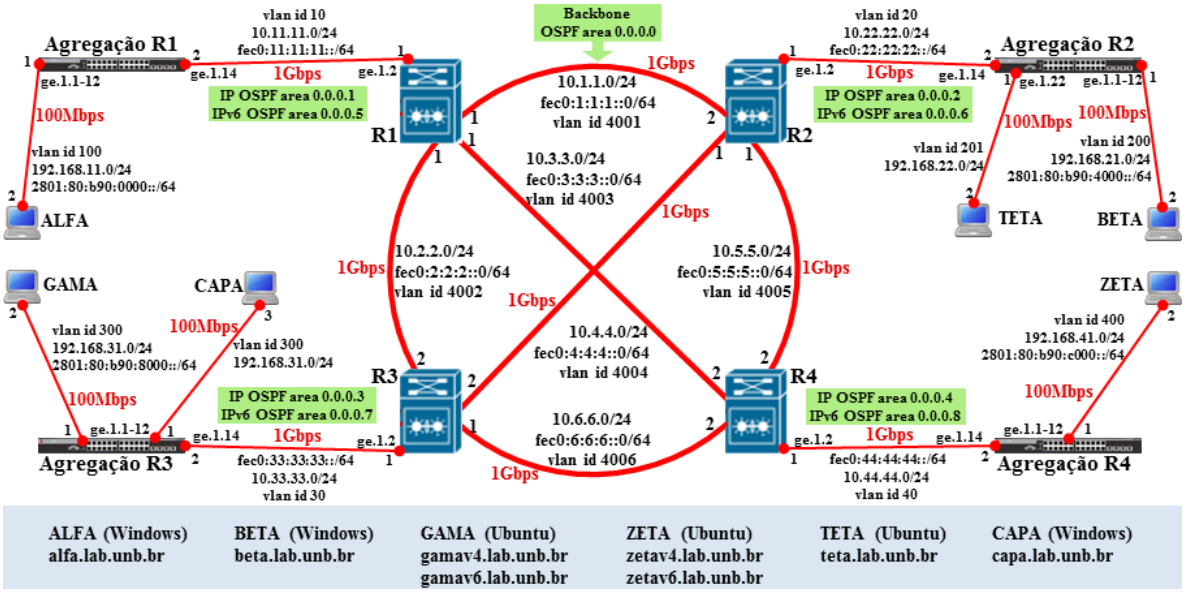


Figura 6.1: Topologia do Laboratório - Pilha Dupla

foi baseada em planejamento prévio a respeito do ambiente de laboratório, planejamento este que também é semelhante ao do cenário REDUnB.

Ainda com foco no que apresenta a Figura 6.1, os *hosts* que integram o laboratório recebem registros de nomes de domínio com objetivo de facilitar o acesso a serviços e de demonstrar o uso de resolução de nomes em ambiente de rede em Pilha Dupla. Assim, os *hosts* receberam registros no formato que segue: **ALFA** com registro alfa.lab.unb.br, **BETA** com registro beta.lab.unb.br, **GAMA** com registros gamav4.lab.unb.br e gamav6.lab.unb.br, **ZETA** com registros zetav4.lab.unb.br e zetav6.lab.unb.br, **TETA** com registro teta.lab.unb.br e por fim **CAPA** com registro capa.lab.unb.br.

6.2 Configurações de Equipamentos no Laboratório

No âmbito da preparação dos equipamentos no laboratório, as tarefas foram divididas em duas fases, a primeira para a configuração dos equipamentos do *backbone* e a segunda para a configuração dos *switches* de agregação. No Apêndice A, as configurações destes equipamentos são apresentadas de forma detalhada. Os passos de configuração dos *switches* de *backbone* são definidos abaixo:

1. Atualização dos *firmwares* para versão (c5-series-06.81.02.0007);
2. Criação de identificações de VLANs;
3. Atribuições destas identificações de VLANs a interfaces virtuais, denominadas de interfaces VLAN;

4. Habilitação do protocolo IPv6;
5. Configurações nas interfaces VLANs dos endereços IPv4 e IPv6 dos enlaces entre os nós do *backbone*, habilitação dos protocolos de roteamento OSPF nas duas versões (2 e 3) separadamente, atribuição da identificação de área para o OSPF e habilitação da interface VLAN para operar;
6. Configurações nas interfaces VLANs dos endereços IPv4 e IPv6 dos enlaces entre os nós do *backbone* e os *switches* de agregação, habilitação dos protocolos de roteamento OSPF nas duas versões (2 e 3) separadamente, atribuição da identificação de área para o OSPF e habilitação da interface VLAN para operar;
7. Atribuída ao *switch* característica de operação com protocolo IPv6 em endereço tipo *Unicast*;
8. Criação de identificações de roteamento OSPF para as duas versões (2 e 3) nas áreas que interligam os nós do *backbone* aos *switches* de agregação;
9. Atribuição de identificações de VLANs a interfaces físicas nos *switches*;
10. Execução de configurações do protocolo *Spanning Tree* para evitar os *loops* em caminhos redundantes na rede.

Quanto aos *switches* de agregação, os passos de configuração são também apresentados na sequência:

1. Atualização dos *firmwares* para versão (c5-series-06.81.02.0007);
2. Criação de identificações de VLANs;
3. Atribuições destas identificações de VLANs a interfaces virtuais, denominadas de interfaces VLAN;
4. Habilitação do protocolo IPv6;
5. Configurações nas interfaces VLANs dos endereços IPv4 e IPv6 dos enlaces entre *switches* de agregação e nós do *backbone*, habilitação dos protocolos de roteamento OSPF nas duas versões (2 e 3) separadamente, atribuição da identificação de área para o OSPF e habilitação da interface VLAN para operar;
6. Configurações nas interfaces VLANs dos endereços IPv4 e IPv6 dos segmentos de redes distribuídos aos clientes, habilitação dos protocolos de roteamento OSPF nas duas versões (2 e 3) separadamente, atribuição da identificação de área para o OSPF e habilitação da interface VLAN para operar;

7. Atribuída ao *switch* característica de operação com protocolo IPv6 em endereço tipo *Unicast*;
8. Criação de identificações de roteamento OSPF para as duas versões (2 e 3) nas áreas que interligam os nós do *backbone* aos *switches* de agregação;
9. Redistribuição de redes conectadas;
10. Atribuição de identificações de VLANs a interfaces físicas nos *switches*;
11. Execução de configurações do protocolo *Spanning Tree* para evitar os *loops* em caminhos redundantes na rede;
12. Configuração de espelhamento de interface para coleta de tráfego.

Ainda na composição do laboratório, na borda da rede encontram-se *hosts* conforme disposição apresentada na Figura 6.1. Os *switches* de agregações **Agregação R1** e **Agregação R4** exibem cenários idênticos, roteando redes somente em Pilha Dupla, com os respectivos *hosts* **ALFA** e **ZETA** podendo se comunicar tanto em IPv4 quanto com IPv6. Os endereços IPv4 e IPv6 configurados são respectivamente 192.168.11.2 e 2801:80:B90:0000::2 para o *host* **ALFA** e 192.168.41.2 e 2801:80:B90:c000::2 para o *host* **ZETA**.

O *switch* de agregação **Agregação R2** possui 2 (duas) identificações de interfaces VLANs configuradas, uma conferida à rede em IPv4 mais IPv6 com *host* **BETA** e a outra somente IPv4 com *host* **TETA**. Os endereços IPv4 e IPv6 designados ao *host* **BETA** são respectivamente 192.168.21.2 e 2801:80:B90:4000::2, enquanto que o *host* **TETA** por operar apenas em IPv4 teve designado o endereço 192.168.22.2. É importante ponderar que o *host* **BETA** não se encontra na mesma subrede IPv4 do *host* **TETA**, o que permite concluir que um *switch* que opere em Pilha Dupla pode perfeitamente prover acesso a redes somente em IPv4, estas produtos de redes legadas.

Quanto ao *switch* de agregação **Agregação R3**, configurada apenas uma identificação de interface VLAN preparada para prover rede IPv4 e IPv6 com *hosts* **GAMA** operando em Pilha Dupla e **CAPA** apenas em IPv4. Os endereços IPv4 e IPv6 designados ao *host* **GAMA** são respectivamente 192.168.31.2 e 2801:80:B90:8000::2, enquanto que o *host* **CAPA** por operar apenas em IPv4 teve designado o endereço 192.168.31.2. Neste ponto é relevante atentar ao fato de que o *host* **CAPA** se encontra na mesma subrede IPv4 do *host* **GAMA**, o que também permite concluir que um *switch* que opere em Pilha Dupla pode perfeitamente permitir que *hosts* somente em IPv4 compartilhem o mesmo domínio de *broadcast* com os *hosts* em Pilha Dupla.

Os *hosts* no escopo do laboratório apresentam características de *hardwares* e sistemas operacionais frequentemente encontradas em equipamentos de clientes na REDUnB. To-

davia, a estes *hosts* foram adicionados *softwares* auxiliares para composição do ambiente de avaliação. Na Tabela 02 seguem descritas as particularidades de cada *host*:

Tabela 6.1: Particularidades de *hosts* no Laboratório

Host	***Hardware***	***Software***
01-ALFA	CPU Core 2 Duo, 2.53GHz, 4GB RAM, 297GB HD, interface Ethernet 100Mbps	Sistema operacional de 64 bits (Windows Seven), XAMPP versão 1.8.3, iperf v-2.0.2-2, jperf v-1.0, Java 7 Update 25, Windows Media Player v-11.0, Windows Media Encoder 9 Series, VLC v-2.0.7, Linphone
02-BETA	CPU Celeron, 1.60GHz, 2GB RAM, 73.4GB HD, interface Ethernet 100Mbps	Sistema operacional de 64 bits (Windows Seven), XAMPP versão 1.8.3, iperf v-2.0.2-2, jperf v-1.0, Java 7 Update 25, Windows Media Player v-11.0, Windows Media Encoder 9 Series, VLC v-2.0.7
03-GAMA	CPU Core i3, 3.07GHz x4, 3.7GB RAM, 488.1GB HD, interface Ethernet 100Mbps	Sistema Operacional de 64 bits (Ubuntu), XAMPP versão 1.8.3, iperf, VLC, Asterisk
04-ZETA	CPU Core 2 Duo, 2.53GHz x2, 3.9GB RAM, 310.7GB HD, interface Ethernet 100Mbps	Sistema Operacional de 64 bits (Ubuntu), XAMPP versão 1.8.3, iperf, VLC, BIND 9.9.5, HTTPING, Linphone
05-TETA	CPU Celeron, 1.30GHz, 2GB RAM, 80.4GB HD, interface Ethernet 100Mbps	Sistema Operacional de 64 bits (Windows Seven)

continua na próxima página

Tabela 6.1: Particularidades de *hosts* no Laboratório (continuação)

Host	***Hardware***	***Software***
06-CAPA	CPU Celeron, 1.60GHz, 2GB RAM, 100.4GB HD, interface Ethernet 100Mbps	Sistema Operacional de 64 bits (Windows Seven)

De forma auxiliar, no laboratório foi configurado no *host* **ZETA** um servidor de resolução de nomes (DNS) ao se considerar a premissa de que nomes e endereços têm finalidades distintas, com nomes para identificar recursos, enquanto endereços para localizar recursos. Com a proposta de implementação da coexistência de endereços IPv4 e IPv6 na rede, a infraestrutura torna-se relativamente complexa para a tradução de nomes em endereços, e vice-versa. Essa infraestrutura de rede com o trabalho de servidores DNS é fundamental para o funcionamento correto da maioria dos serviços. No Apêndice B seguem detalhes das configurações aplicadas no laboratório.

Figura 6.2: Laboratório prático

Importante observar na topologia do laboratório que todos os *hosts* envolvidos nos testes já possuem entradas de registros no servidor DNS. Os *hosts* **ALFA** e **BETA**

encontram-se com 01 (um) nome de domínio para cada com 02 (dois) endereços, 01 (um) em IPv4 e o outro em IPv6. Para os *hosts* **GAMA** e **ZETA** existem 02 (dois) nomes de domínios para cada, 01 (um) nome de domínio para o endereço IPv4 e o outro para o endereço IPv6. Quanto aos *hosts* **TETA** e **CAPA**, por operarem apenas com o protocolo IPv4, foi adicionado apenas 01 (um) nome de domínio e 01 (um) endereço IPv4 para cada. Ao adicionar registros de nomes de domínios com um endereço IPv4 e outro IPv6 como no caso dos *hosts* **ALFA** e **BETA**, se apresenta com o objetivo de comprovação da prioridade da comunicação pelo IPv6 em detrimento do IPv4.

O ambiente do laboratório prático descrito acima foi montado nas dependências do Centro de Informática da UnB. A Figura 6.2 demonstra o ambiente em questão composto de 1 (um) *rack* com os *switches* e os 6 (seis) *hosts* usados nos testes.

De forma complementar, a este cenário foi acrescentado um equipamento denominado por ***Fluke Optiview XG*** apontado na Figura 6.2 com identificação 07 que é um *tablet* completo para a análise de desempenho em redes. O equipamento possui característica que permite extrair relatórios detalhados do comportamento do tráfego de rede, com subsídio à análise e diagnóstico de redes cabeadas e sem fio [47]. Esta ferramenta foi utilizada no laboratório para a tarefa de capturas de pacotes e análise destas capturas por meio do *software* de análise embarcado na solução, o ***ClearSight Analyzer***.

6.3 Certificações do Ambiente

Preliminar à fase de avaliações foram executadas certificações de ambiente com objetivo de verificar a correta configuração dos equipamentos do laboratório. Inicialmente foi ratificada a conectividade entre todos os *hosts* (**ALFA**, **BETA**, **GAMA**, **ZETA**, **TETA** e **CAPA**) e a confirmação do pleno funcionamento da resolução de nomes de domínio. Comprovar a coexistência harmônica dos protocolos IPv4 e IPv6 foi uma tarefa elementar e ainda a verificação da prioridade de comunicação pelo protocolo IPv6 entre *hosts* em Pilha Dupla sendo experimentado por meio da execução de ICMP entre os *hosts* **ALFA** e **BETA** utilizando o nome de domínio beta.lab.unb.br. Outro aspecto constatado refere-se à possibilidade de *host* somente em IPv4, no caso o *host* **TETA**, executar consulta de registro DNS ao nome de domínio alfa.lab.unb.br com retorno de detalhes de endereços IPv4 e IPv6. Foi confirmada também a comunicação entre *host* somente em IPv4 com *host* em Pilha Dupla.

O esquema de roteamento OSPF foi validado, apresentada tabela de roteamento de um dos *switches routers* do *backbone* e traçada rota entre dois *hosts* **BETA** e **CAPA**, nos extremos da rede. Por fim, com o tráfego de fundo em execução, foi averiguado o

nível de utilização de CPU nos *switches* do laboratório. Os detalhes relacionados aos experimentos estão contidos no Apêndice D.

6.4 Avaliação de Desempenho no Ambiente de Experimentos

Os procedimentos sistemáticos de experimentos possuem o objetivo de coletar e avaliar informações que municiem o amplo processo de transição, sendo executados com os protocolos IPv4 e IPv6 em *hosts* em Pilha Dupla do laboratório. Nesta finalidade, são descritos dois planos de avaliações de desempenho:

- Avaliação 1: Comunicação de serviço HTTP;
- Avaliação 2: Serviço VoIP.

Como exposto, na avaliação 1 o alvo é prover acesso a um serviço amplamente utilizado na REDUnB e na Internet, o serviço HTTP implementado no laboratório, com intuito de obter a latência entre dois pontos da rede. Na avaliação 2 foi implementado serviço VoIP (*Voice over Internet Protocol*) na comunicação entre dois *hosts*, com análise de desempenho com metas que apontam taxas de pacotes fora da sequência, perdas de pacotes, *jitter*, latência, índice MOS (*Mean Opinion Score*) e indicador *R-Value*.

A latência é o termo usado para descrever a quantidade de tempo que leva para os dados serem processados ou movidos de um ponto a outro. A latência em enlaces de redes é a combinação da propagação de *delay* e o processamento de *delay* [13]. Os requisitos para a navegação na *web* convencional (HTTP) são influenciados principalmente pelo tempo de resposta, que é limitada a não mais que 5 segundos, em matéria de acesso otimizado não mais que 4 segundos [48].

Em redes locais cabeadas e bem estruturadas é essencial que a comunicação ocorra com ótimo desempenho ao considerar a proximidade entre os dispositivos e a qualidade da infraestrutura de cabeamento. Em teste de *Ping* em redes locais é desejável que a latência não ultrapasse 10ms, ou é tolerável e não recomendado até 30ms. Quanto mais dispositivos intermediários existirem entre dois *hosts*, naturalmente aumenta a latência porque cada dispositivo intermediário tem algum mecanismo de tratamento dos quadros [48].

Em termos de voz sobre IP, latência é o tempo que a fala leva pra sair do locutor e chegar ao receptor. Latência não tem nada a ver com o rendimento, largura de banda, ou a velocidade de um enlace. Latência é influenciada pela distância, a velocidade de propagação do sinal e a quantidade de tempo que o *hardware* leva para processar os

dados. No serviço VoIP a latência deve possuir um valor abaixo do patamar de 150ms [48].

Pacotes fora de sequência correspondem a entrega de pacotes de dados em uma ordem diferente da que foi enviada. Este indicador de desordenação pode ser causado pelo fato de os pacotes seguirem vários caminhos através de uma rede, ou através de caminhos de processamento paralelo dentro de equipamentos de rede que não são projetados para garantir que a ordenação de pacotes seja preservada [13].

A perda de pacote, também frequentemente referenciada como *packet loss* ocorre quando um ou mais pacotes que navegam sobre uma rede de computadores falha em alcançar o destinatário, sendo um dos erros previstos na transmissão de dados [7]. Contudo, uma taxa de perda de pacotes aceitável enquadra-se em algo menor que um porcento ($<$1%) da quantidade de pacotes trafegados [49].

O *jitter*, que fundamenta-se na variação do atraso de transmissão, é um dos principais fatores que causa degradação da qualidade em uma comunicação de voz sobre IP (VoIP). As aplicações VoIP geram pacotes em intervalos regulares, mas após passarem pelos roteadores da rede intermediária, os intervalos de tempo entre os pacotes se tornam totalmente irregulares. Entre as causas do *jitter* estão a interferência eletromagnética e a interferência com outros sinais [50]. Uma taxa menor que 400 milissegundos ($<$400ms) é sugerida para um diálogo de VoIP com qualidade.

Para a avaliação de desempenho da qualidade de voz trafegada por uma rede IP existem alguns métodos, dentre estes o MOS. Este método representa uma das mais conhecidas medidas da qualidade de voz, embora seja de modo subjetivo. No cenário de teste do laboratório prático onde é apresentado o indicador MOS, o índice é calculado sob a recomendação P.800 do ITU-T (*International Telecommunication Union*) aliado a avaliações objetivas de métodos usando modelos de percepção PAMS (*Perceptual Analysis Measurement System*), PESQ (*Perceptual Evaluation of Speech Quality*) e ainda recomendações ITU-T G.107 com *R-Value*, os quais permitem a medição da qualidade fim a fim de uma comunicação de voz em condições de rede reais. A escala de valores MOS vai de 1 a 5, sendo que uma pontuação 4 ou maior indica o som de voz adequado ao serviço de telefonia e VoIP [51].

O indicador *R-Value* é um número ou pontuação que é usado para expressar quantitativamente a qualidade subjetiva do discurso em sistemas de comunicações, especialmente redes digitais que trafegam VoIP, ou para as quais o serviço de VoIP está sob avaliação. A pontuação *R-Value*, que é utilizada em conjunção com processos de teste de voz, pode variar do indicador 1 (pior) a 100 (o melhor), tendo a margem entre 80 e 90 indicada como uma taxa satisfatória e de 90 a 100 indicada com uma taxa muito satisfatória. A relação de 1 (um) ponto MOS é equivalente a cerca de 20 (vinte) pontos de *R-Value*, embora a

relação não seja perfeitamente linear.

O método de pontuação *R-Value* é o preferido em relação a outros métodos de pontuação pelas empresas de telecomunicações, por ser considerado método que retrata com precisão, e assim, pode ser utilizado para prever os efeitos de perda de pacotes e atrasos em redes digitais que transportam os sinais de voz [51].

Uma vez estabelecidas as métricas de avaliações de desempenho para caracterizar um ambiente próximo do real, foi definida a utilização de tráfegos de fundo no ambiente de experimentos. Foram injetados tráfegos de fundo, ou seja, tráfegos que não fazem parte do foco na análise de desempenho, mas podem ser utilizados para simular o congestionamento de uma rede operacional. Dessa forma, foram utilizados *software* ***Windows Media Encoder*** para transmissão de vídeo sob demanda e *Broadcasting Live Event* e *software* ***iPerf***.

Na transmissão de vídeo sob demanda o *host* **BETA** provê acesso a arquivo ***.avi*** de 207MB em IPv4 e IPv6 para os *hosts* **ALFA**, **GAMA** e **ZETA**. Enquanto que o *host* **GAMA** provê acesso ao mesmo arquivo ***.avi*** em IPv4 e IPv6 para os *hosts* **ALFA**, **BETA** e **ZETA**. As transmissões citadas são configuradas na aplicação com *Bit Rate* de 2137Kbps e com 29.97 *frames* por segundo, executados em *loop* e acessadas nos clientes pelo *software* ***VLC***.

Na transmissão de *Broadcasting Live Event* o *host* **BETA** provê acesso à aplicação em IPv4 e IPv6 para os *hosts* **ALFA**, **GAMA** e **ZETA**. Enquanto que o *host* **GAMA** provê acesso em IPv4 e IPv6 para os *hosts* **ALFA**, **BETA** e **ZETA**. Os conteúdos encaminhados em *broadcasting* são captados por câmeras acopladas aos *hosts* **BETA** e **GAMA** e enviados com *Bit Rate* de 2137Kbps e com 29.97 *frames* por segundo, acessadas nos clientes pelo *software* ***VLC***.

Por meio do *software* ***iPerf*** foi aplicado na direção de **ALFA** para **ZETA** tráfego em IPv6, e no sentido inverso, de **ZETA** para **ALFA** tráfego em IPv4, tráfegos estes com as mesmas características. O *host* **ALFA** com comando *iperf -c 2801:80:b90:c000::2 -V -len 1450 -t 1000000000* envia tráfego para o *host* **ZETA** que habilita o serviço com o comando *iperf -s -V*. No sentido inverso, o *host* **ZETA** com comando *iperf -c 192.168.11.2 -l 1450 -t 1000000000* envia tráfego para o *host* **ALFA** que habilita o serviço com o comando *iperf -s*.

Ainda utilizando o *software* ***iPerf*** foi aplicado na direção de **BETA** para **GAMA** tráfego em IPv6, e no sentido inverso, de **GAMA** para **BETA** tráfego em IPv4, tráfegos estes mais uma vez com as mesmas características. O *host* **BETA** com comando *iperf -c 2801:80:b90:8000::2 -V -len 1450 -t 1000000000* envia tráfego para o *host* **GAMA** que habilita o serviço com o comando *iperf -s -V*. No sentido inverso, o *host* **GAMA** com comando *iperf -c 192.168.21.2 -l 1450 -t 1000000000* envia tráfego para o *host* **BETA**

que habilita o serviço com o comando *iperf -s*.

Os argumentos dos comandos do *software* ***iPerf*** usados no cenário incluem *-c* apontando que o *host* trabalhará como cliente, o *-V* designa a nova versão do protocolo IP, o *-len 1450* e *-l 1450* indicam o tamanho de *buffer* a ser trafegado, o *-t 1000000000* informa o tempo em segundos da comunicação e o *-s* informa que o *host* trabalhará como servidor.

6.4.1 Resultados

Nesta seção são apresentados os resultados dos dois planos de avaliações de desempenho:

Resultados da Avaliação 1

A mensuração do desempenho no acesso a serviço HTTP através da latência nos protocolos IPv4 e IPv6 foi realizada do *host* **ZETA** para o *host* **ALFA**. Neste cenário foram empregados os utilitários ***Ping*** e ***HTTPing*** para subsidiar na coleta de resultados e análises.

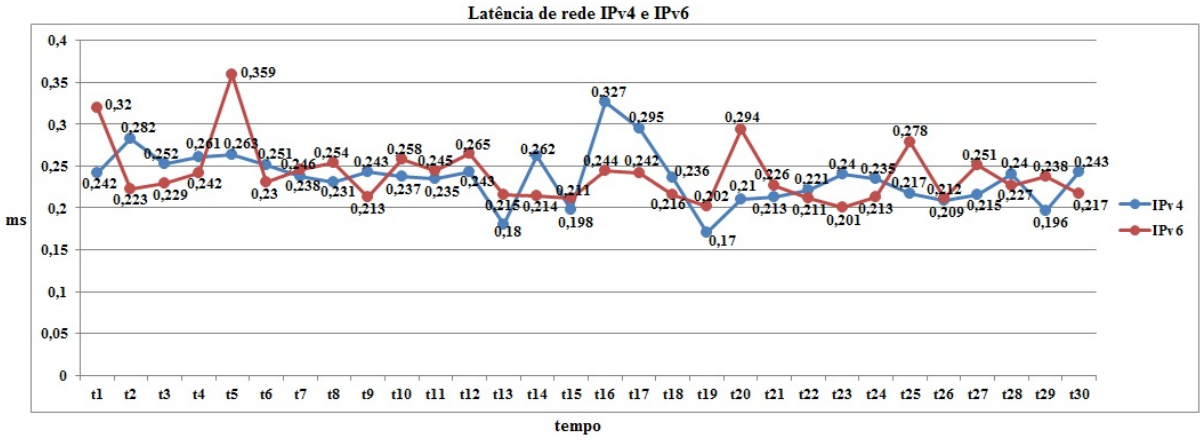


Figura 6.3: Desempenho da latência de rede na comunicação entre *hosts* ZETA e ALFA

O pequeno utilitário ***Ping*** foi usado para testar a conectividade entre os *hosts* **ZETA** e **ALFA**, extraindo a latência e *jitter* desta comunicação no âmbito da camada de rede com finalidade de comprovar o nível de conformidade do meio de comunicação. Com o uso dos comandos *ping -c 5 alfa.lab.unb.br* e *ping6 -c 5 alfa.lab.unb.br*, tendo em ambos o argumento *-c 5* para determinar a quantidade de pacotes enviados na comunicação. A partir do *host* **ZETA** foram executados três grupos de dez capturas de estatísticas sumarizadas em diferentes momentos, totalizando trinta capturas. Os três grupos de capturas foram executados em dias diferentes e com intervalo de pelo menos 5 minutos entre cada captura individual.

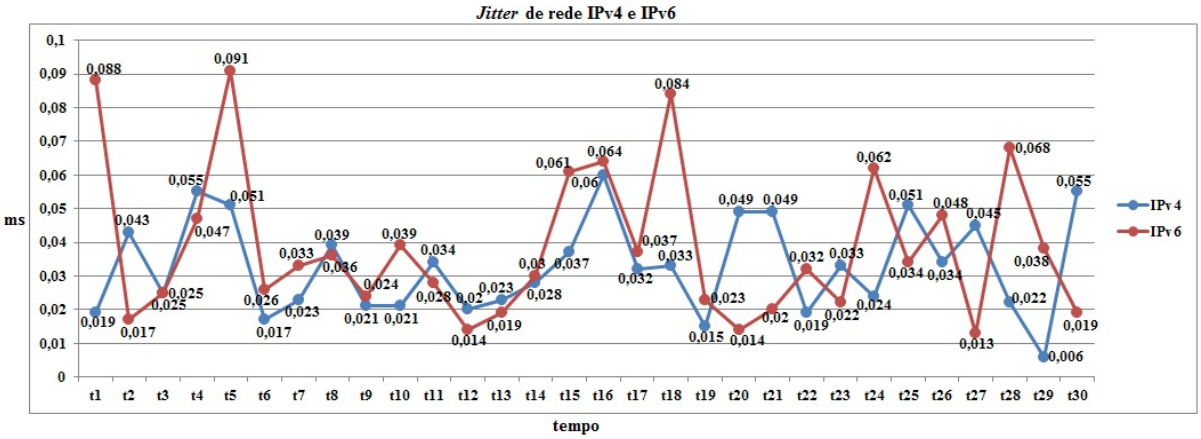


Figura 6.4: Desempenho do *Jitter* de rede na comunicação entre *hosts* ZETA e ALFA

O retorno das informações sobre o tempo de ida e volta dos pacotes de dados ocorreram sem perdas, gerando estatísticas finais de perda de pacotes, latência e *jitter* como se encontra no Anexo A e sumarizado nas Figuras 6.3 e 6.4 os índices das médias de latências e *jitters*.

Os índices que compõem o gráfico da Figura 6.3 apontam na comunicação IPv4 média de latência alcançando a máxima de 0,327ms no instante 16 e mínima de 0,17ms no instante 19 enquanto que na comunicação em IPv6 média de latência alcançando a máxima de 0,359ms no instante 5 e mínima de 0,201ms no instante 23. Ao considerar que na prática um excelente desempenho da latência em redes locais não é recomendável ultrapassar 10ms e que é tolerável até 30ms, os índices apresentados evidenciam um ótimo nível de latência se considerado o ambiente interno da rede do laboratório.

A partir dos resultados obtidos quanto às médias de latências de rede, ao considerar cada um dos três grupos de capturas, somadas as taxas médias de latência e divididas por dez que é um número de capturas de cada grupo, foi possível alcançar os seguintes indicadores:

- No primeiro grupo o desempenho do protocolo IPv6 apresentou uma taxa de 2,96% acima da taxa média da latência alcançada pelo IPv4;
- No segundo grupo o desempenho do protocolo IPv4 apresentou uma taxa de 0,34% acima da taxa média da latência alcançada pelo IPv6;
- E no terceiro grupo o desempenho do protocolo IPv6 apresentou uma taxa de 0,807% acima da taxa média da latência alcançada pelo IPv4.

Se considerado todo o conjunto de trinta capturas, somadas as médias de latências e extraídas taxas médias gerais referentes a cada protocolo IP, com isso, o desempenho do

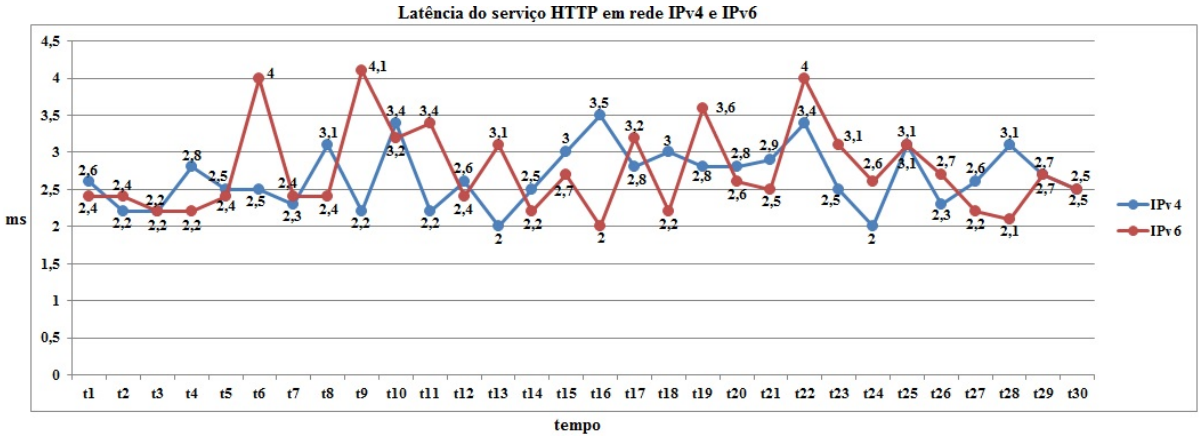


Figura 6.5: Desempenho da latência de rede na comunicação do serviço HTTP entre *hosts* ZETA e ALFA

protocolo IPv6 apresentou uma taxa de 1,56% acima da taxa média latência alcançada pelo IPv4.

No quesito *jitter* de rede apresentado na Figura 6.4, os índices que compõem o gráfico apontam na comunicação IPv4 taxa máxima de 0,06ms no instante 16 e mínima de 0,006ms no instante 29 enquanto que na comunicação em IPv6 taxa máxima de 0,091ms no instante 5 e mínima de 0,013ms no instante 27. A partir dos resultados obtidos quanto ao *jitter* de rede, ao considerar cada um dos três grupos de capturas, somadas as taxas de *jitter* e divididos por dez que é um número de capturas de cada grupo, foi possível alcançar os seguintes indicadores:

- No primeiro grupo o desempenho do protocolo IPv6 apresentou uma taxa de 35,66% acima da taxa de *jitter* alcançada pelo IPv4;
- No segundo grupo o desempenho do protocolo IPv6 apresentou uma taxa de 13% acima da taxa de *jitter* alcançada pelo IPv4;
- E no terceiro grupo o desempenho do protocolo IPv6 apresentou uma taxa de 5,32% acima da taxa de *jitter* alcançada pelo IPv4.

Se considerado todo o conjunto de trinta capturas, somadas as taxas de *jitter* e dividido por trinta que é a quantidade total de capturas, extraindo assim taxas médias gerais de *jitter* referentes a cada protocolo IP, é possível inferir que o IPv6 obteve um resultado inferior ao do seu antecessor, com um índice de 17,38% maior que o *jitter* na rede IPv4.

Quanto ao ***HTTPing***, este é semelhante ao ***Ping***, mas para requisições HTTP. Ao fornecer uma URL, este vai mostrar quanto tempo leva para conectar, enviar um pedido e obter a resposta, com apenas os cabeçalhos. Com este utilitário, é importante ter em mente que este não está apenas testando o tempo para o servidor *web* responder, mas

também o tempo que leva para enviar o pedido através da rede e o servidor *web* retornar com os cabeçalhos de volta. Basicamente, ele mede a latência do servidor *web* somado com a latência da rede e opera até a camada de aplicação.

A partir do *host* **ZETA** foram executados três grupos de dez capturas de estatísticas sumarizadas em diferentes momentos, totalizando trinta capturas. Os três grupos de capturas foram executados em dias diferentes e com intervalo de pelo menos 5 minutos entre cada captura individual. Foram utilizados os comandos *httping -c 5 alfa.lab.unb.br* e *httping -6 -c 5 alfa.lab.unb.br*, tendo em ambos o argumento *-c 5* para determinar a quantidade de pacotes enviados na comunicação e no segundo comando o argumento *-6* para especificar o uso de IPv6.

O retorno das informações sobre o tempo de ida e volta dos pacotes de dados ocorreram sem perdas, gerando estatísticas finais de falhas na comunicação e latência como se encontra no Anexo B e sumarizado na Figura 6.5 os índices de médias de latências. Os índices que compõem o gráfico em foco apontam na comunicação IPv4 média de latência alcançando a máxima de 3,5ms no instante 16 e mínima de 2ms nos instantes 13 e 24 enquanto que na comunicação em IPv6 média de latência alcançando a máxima de 4,1ms no instante 9 e mínima de 2ms no instante 16. Mais uma vez os resultados levam a compreender que as taxas apresentadas por ambos protocolos IP estão dentro da margem considerada aceitável (entre 4 e 5 segundos [5]).

A partir dos resultados obtidos quanto às médias de latências de rede, ao considerar cada um dos três grupos de capturas, somadas as taxas médias de latência e divididas por dez que é um número de capturas de cada grupo, foi possível alcançar os seguintes indicadores:

- No primeiro grupo o desempenho do protocolo IPv6 apresentou uma taxa de 7,36% acima da taxa média da latência alcançada pelo IPv4;
- No segundo grupo o desempenho do protocolo IPv6 apresentou uma taxa de 0,735% acima da taxa média da latência alcançada pelo IPv4;
- E no terceiro grupo o desempenho do protocolo IPv6 apresentou uma taxa de 1,476% acima da taxa média da latência alcançada pelo IPv4.

Se considerado todo o conjunto de trinta capturas, somadas as médias de latências do serviço HTTP na rede e extraídas taxas médias gerais referentes a cada protocolo IP, com isso, o desempenho do protocolo IPv6 apresentou uma taxa de 3,12% acima da taxa média de latência do serviço HTTP alcançada pelo IPv4, ainda assim em nível satisfatório para o tipo de serviço HTTP.

Entretanto, conforme análises apresentadas sobre os desempenhos dos protocolos IPv4 e IPv6 quanto à latência de rede, o *jitter* de rede e a latência do serviço HTTP, quando

analisados os três grupos separadamente, é possível inferir que o protocolo IPv4 possui uma performance superior se comparado ao protocolo IPv6, mesmo que moderado. Se considerado o desempenho dos protocolos IPv4 e IPv6 nas mesmas circunstâncias citadas, e com um maior número de amostras, ou seja, um número três vezes superior ao da análise individual, é possível identificar uma atenuação da diferença de desempenho entre os protocolos IPv4 e IPv6, o que permite compreender uma tendência à atenuação entre a diferença entre os protocolos IPv4 e IPv6 ao longo do tempo.

Resultados da Avaliação 2

O objetivo na segunda avaliação é demonstrar sobre as redes IPv4 e IPv6 a utilização de serviço VoIP entre 02 (dois) *hosts* no laboratório e a execução de análise de desempenho da qualidade de voz sobre a comunicação em questão em ambas versões do protocolo IP. Neste cenário o *host* **GAMA** desempenha um papel fundamental ao prover o serviço de VoIP na rede por meio do *software* ***Asterisk***, atribuindo os números 1001 e 1002 para os respectivos *hosts* **ALFA** e **ZETA**, configurações que constam no Apêndice C.

Nesta circunstância os *hosts* **ALFA** e **ZETA** têm os seus *softwares* clientes VoIP ***Linphone*** configurados para atrelar o serviço VoIP sobre protocolo SIP (*Session Initiation Protocol*) ao servidor ***Asterisk***. Assim, os *hosts* são registrados respectivamente com os números 1001 e 1002. A comunicação VoIP é iniciada através do *software* ***Linphone*** do *host* **ALFA** para o *host* **ZETA** em uma ligação ininterrupta durante todo o procedimento de experimento.

Uma vez concluída a ativação do serviço VoIP nos *hosts* **GAMA** (servidor), **ALFA** e **ZETA** (clientes), é o momento de posicionar o equipamento ***Fluke Optiview XG*** no cenário para coleta de tráfego. O equipamento em questão foi inserido no contexto do laboratório por meio de conexão física em cabo de par trançado na interface **ge.1.15** do *switch* de **Agregação R4**. Como bem consta no Apêndice I, a interface **ge.1.14** do *switch* **Agregação R4** espelha para a interface **ge.1.15** todo o tráfego *ingress* ou *egress* que trafega pela interface **ge.1.14**.

Desta forma, o equipamento ***Fluke Optiview XG*** é ajustado com parâmetros correspondentes ao protocolo UDP porta 5060 com opção SIP habilitada na seção *Combined Flows*. Já com filtros direcionados para coleta de todo trânsito de pacote a respeito do VoIP entre os elementos de interesse **ALFA** e **ZETA**, a coleta é realizada e armazenada em arquivos com extensão ***.pcap*** para posterior análise com o *software* ***ClearSight Analyzer*** do fabricante **Fluke**. Vale ressaltar que foram realizadas 05 (cinco) capturas individuais em diferentes momentos para os tráfegos em protocolo IPv4 e em IPv6.

Antes de expor os resultados obtidos das capturas, é importante destacar aspectos relevantes à respeito do serviço VoIP. Os 02 (dois) principais parâmetros de qualidade de serviço de voz mais afetados pelo desempenho da rede IP e processamento de VoIP são a

Protocolos IPv4 / IPv6	Packet Loss		Jitter (ms)		Out of Sequence		Latency (ms)	
	Client	Server	Client	Server	Client	Server	Client	Server
192.168.11.2 --> 192.168.41.2	0.00%	0.00%	20.095	15.264	0.00%	0.00%	16.778	16.778
2801:80:b90:0000::2 --> 2801:80:b90:c000::2	0.00%	0.24%	19.918	15.116	0.00%	0.00%	22.212	22.212

Figura 6.6: Índices de desempenhos de tráfego VoIP nos protocolos IPv4 e IPv6

clareza ou qualidade e atraso de voz. A melhoria da voz depende de muitos fatores, além de perda de pacotes e *jitter*, e os vários fatores influenciam um no outro.

Por ser sensível ao tempo, a voz tem uma baixa tolerância a atrasos (<100 milissegundos) [52]. Uma tolerância para variância do atraso ou *jitter* (<400 milissegundos). Além disso, as aplicações de voz geralmente têm uma baixa tolerância para a perda de pacotes (<1%). Por frequentemente utilizarem protocolo UDP, um pacote perdido significa perda de dados, não há retransmissões [49].

Protocolos IPv4 / IPv6	Packets		Packet Loss		Packet Loss Rate	
	Client	Server	Client	Server	Client	Server
192.168.11.2 --> 192.168.41.2	896	892	0	0	0.00%	0.00%
2801:80:b90:0000::2 --> 2801:80:b90:c000::2	836	833	0	2	0.00%	0.24%

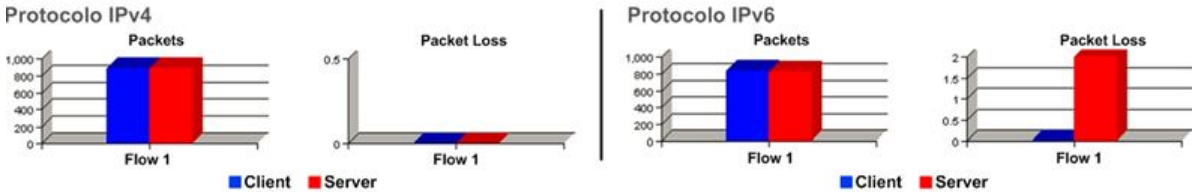


Figura 6.7: Relatório de perda de pacotes no tráfego VoIP nos protocolos IPv4 e IPv6

Conforme sumarização executada pelo *software* ***ClearSight Analyzer*** sobre captura obtida por meio do equipamento ***Fluke Optiview*** de tráfego VoIP entre os *hosts* **ALFA** e **ZETA** sobre os protocolos IPv4 e IPv6, a Figura 6.6 extraída e adaptada de relatório do *software* ***ClearSight Analyzer*** já expõe de forma sucinta alguns indicadores relevantes à análise. É identificado que não houve perda de pacotes na comunicação VoIP sobre o protocolo IPv4 em nenhum dos *hosts*, enquanto que na comunicação sobre o protocolo IPv6 foi registrada uma perda de 0,24% no lado do *host* **ZETA**, taxa ainda assim considerada muito boa mediante referência supracitada. A variância de atraso ou *jitter* no tráfego VoIP se manteve em patamares sensivelmente abaixo do limite recomendável, tanto na comunicação pelo protocolo IPv4 como com o IPv6. Isso permitiu o alcance dos fluxos de pacotes de voz do *host* **ALFA** para o destino *host* **ZETA**, reciprocamente, numa harmonia constante, e em um ritmo de recebimento de pacotes proximamente com que foram gerados.

Ainda com referência à Figura 6.6, nos tráfegos em protocolos IPv4 e IPv6, os pacotes não sofreram atrasos significativos a ponto de culminar em desordem, como se vê no

campo *Out of Sequence* com taxas percentuais de 0,00%, o que aponta que mesmo com o tráfego de fundo (concorrente) na rede, o ambiente de rede apresenta-se adequado ao tráfego de aplicações sensíveis como VoIP. O campo *Out of Sequence* pode ainda apontar que o enlace sofre de perdas, duplicação ou reordenação. O valor da latência está na média em um patamar bastante aceitável para aplicações de VoIP, tanto no protocolo IPv4 quanto IPv6, dado que, como pode ser observado na Figura 6.6 a latência encontra-se sensivelmente abaixo do limite recomendado para serviço VoIP, que é de 150 milissegundos [48]. Entretanto, esta taxa de latência apresentada pelo protocolo IPv6 foi 32,38% maior que a taxa do protocolo IPv4.

Um relatório de pacotes perdidos foi gerado no intuito de demonstrar graficamente os dados já conhecidos à respeito de perda de pacote, dados estes demonstrados na Figura 6.7 extraída e adaptada de relatório do *software* ***ClearSight Analyzer***. De forma complementar, a este relatório foi adicionada para cada protocolo IPv4 e IPv6 a quantidade de pacotes envolvida na comunicação VoIP entre os *hosts* **ALFA** e **ZETA** e a taxa percentual de pacotes perdidos. Nota-se que não houve perda de pacotes para a comunicação realizada em IPv4, com a taxa percentual de perda de pacotes de 0,00%, um excelente indicativo. Enquanto que na comunicação realizada em IPv6, embora tenha ocorrido perda de pacotes da ordem de 0,24%, ainda assim representa uma ótima qualidade do diálogo.

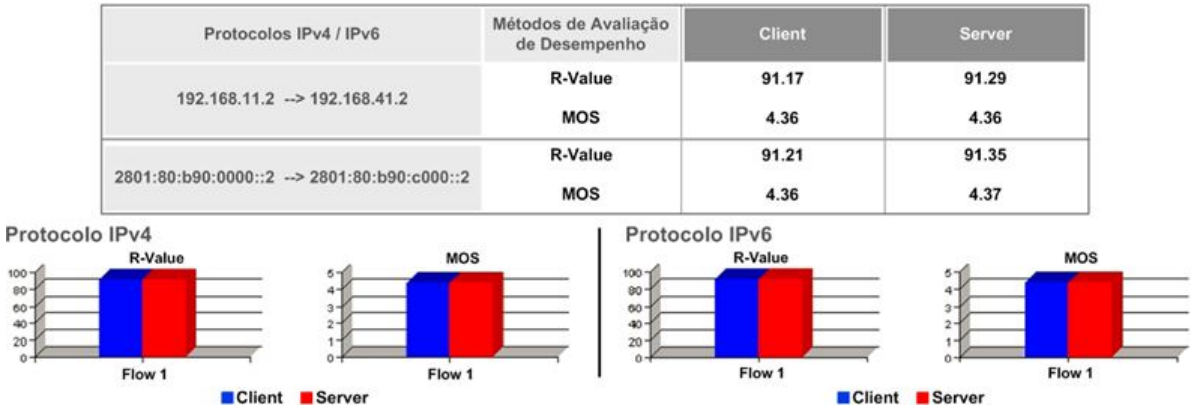


Figura 6.8: Indicadores de métodos de avaliação de desempenho do tráfego VoIP nos protocolos IPv4 e IPv6

Ao se tratar de métodos de avaliação de desempenho, por meio da Figura 6.8 extraída e adaptada de relatório do *software* ***ClearSight Analyzer***, fica comprovado que o tráfego de voz tomado como experimento no laboratório prático ocorreu em alta qualidade nos protocolos IPv4 e IPv6, e isto se baseia nos indicadores expostos. Os *R-Values* alcançados em ambas versões foram identificados acima de 90 pontos e o índices MOS acima de 4.34 pontos, o que representam pontuações de um nível de satisfação do cliente VoIP denominada de *Very satisfied* ou Muito Satisfatório [53].

6.4.2 Análise dos Resultados

Os resultados apresentados nas avaliações 1 e 2 levam à compreensão de que os protocolos IPv4 e IPv6 demonstram uma pequena diferença de desempenho, consubstanciando na equivalência entre as versões do protocolo. Na avaliação 1 o protocolo IPv4 apresentou resultados melhores se comparados aos resultados do IPv6, enquanto que na avaliação 2 os resultados das duas versões do protocolo IP não apresentaram significativas diferenças de desempenho, mas ambas avaliações com resultados satisfatórios para as aplicações em análise. Em meio a estas avaliações, é relevante que em um processo contínuo de implementação do novo protocolo nos ambientes de rede em produção, que sejam executadas mais avaliações sobre os serviços de amplo uso na REDUnB.

Em complemento à análise dos resultados, vale julgar o ambiente de laboratório para execução dos experimentos no estudo em questão, considero extremamente proveitoso o uso de ambiente real com equipamentos de alto desempenho. Com esta escolha tornou possível notar o comportamento na prática da coexistência dos protocolos IPv4 e IPv6 sem ter a considerável limitação de *hardware* quando se usa ambientes virtuais. O laboratório foi montado com topologias de rede lógica e física correspondentes ao ambiente da REDUnB, com exceção da *bandwidth* de cada enlace do *backbone* em topologia *full mesh*, que no laboratório foram estabelecidos em 1Gbps ao contrário do ambiente real que é de 10Gbps. Contudo, a dissemelhança entre os referidos enlaces do ambiente de *backbone* da REDUnB com o do laboratório não influenciou nos resultados obtidos nos experimentos.

Ainda na composição do laboratório, foram utilizados computadores com configurações frequentemente encontradas com os clientes na REDUnB. Estes computadores com interfaces de rede com *bandwidth* de 100Mbps, limitou sensivelmente a possibilidade de exaustão do ambiente de rede. Os *switches* designados para o laboratório por serem de alto desempenho conforme especificado neste documento, necessitariam de uma infraestrutura com características superiores para esgotamento de sua capacidade. Contudo, o ambiente conforme implementado apresentou condições bastante favoráveis aos experimentos escolhidos bem como a possibilidade de muitos outros experimentos de aplicabilidades distintas.

É importante considerar sobre os resultados apresentados, que se observado o ambiente de laboratório, é possível atribuir seguramente validade aos experimentos e aos resultados, principalmente por se tratar de tarefas executadas em ambiente real e com inserção de tráfego de fundo para simular congestionamentos de rede em operação. Mesmo ao julgar válidos e seguros os resultados obtidos, não são dispensados experimentos com maior número de serviços e em diferentes formatos de coletas de informações, diferentes metodologias, métricas de avaliações, e em ambiente de produção. O fato é que o tempo exíguo com o equipamento ***Fluke Optiview XG*** no processo de produção deste projeto

não permitiu a execução de experimentos com um maior número de serviços.

6.5 Resumo do Capítulo

Este capítulo apresentou o cenário de avaliação executado em laboratório real. Na sequência foram expostas as características do laboratório e as configurações utilizadas para implementação do modelo descrito neste estudo. Por fim, foram descritas as métricas perseguidas no ambiente de testes, a certificação de ambiente e os testes com seus resultados.

Capítulo 7

Conclusões

Por meio da análise dos resultados obtidos foi possível concluir que o protocolo IPv4 possui um melhor desempenho do que o protocolo IPv6, mesmo que módico. Mesmo com desempenho inferior, a transição se faz necessária e já vem sendo executada de forma gradual, mas não há como precisar uma data para migração total. O IPv4 ainda deve funcionar por muitos anos.

O processo de migração de versão do protocolo IP nas redes atuais, incluindo a REDUnB, precisa ocorrer, um processo imperioso para o futuro da Internet. A crescente escassez do IPv4 possivelmente submeterá as empresas e os profissionais de tecnologias de redes a esta ação. Conforme abordado, o IPv6 não somente tem a vantagem de disponibilizar endereços válidos para toda a necessidade atual, mas também com a previsão futura do uso crescente com a Internet das Coisas. Embora a REDUnB atualmente não vivencie a referida escassez, não é inteligente ignorar o movimento registrado em todo o mundo quanto à migração para o IPv6, senão esta migração poderá ocorrer de forma repentina no momento em que a escassez atingir pontos críticos.

Em alguns ambientes de rede a utilização do protocolo IPv6 pode parecer algo ainda distante de acontecer, por existir barreira a ser vencida quanto a atualização de *hardwares* e *softwares*, o que representa custos elevados. De forma propícia, o ambiente da REDUnB também já possui sua infraestrutura completa sobre *hardwares* e *softwares* (sistemas operacionais) que já suportam o novo protocolo, o que representa um significativo subsídio para investimento no estudo da migração para o protocolo IPv6.

Nesta circunstância, o modelo escolhido para migração do ambiente IPv4 para IPv6 foi o de ampla utilização da técnica de Pilha Dupla. Neste modelo são configuradas as duas versões de protocolos nos equipamentos de rede, permitindo aos usuários o acesso a redes em IPv4 utilizando seu endereço IPv4 e redes IPv6 utilizando seu endereço IPv6, o que proporciona a transição um processo suave e sem grandes transtornos. Para simulação da implementação do cenário de rede em Pilha Dupla foi utilizado um laboratório em

ambiente real, no qual os testes apresentaram resultados equivalentes entre os protocolos IPv4 e IPv6 quanto às métricas escolhidas. Outro aspecto relevante nas avaliações é que os *switches routers* envolvidos no laboratório não apresentaram aumento considerável de processamento da CPU, mesmo com os testes ocorrendo em meio ao tráfego de fundo.

A viabilidade para realização desta migração na REDUnB não requer grandes modificações da estrutura das redes instaladas. Mas como implantar IPv6 afeta várias áreas, é importante ter o apoio de instâncias superiores da instituição, não basta só a equipe de profissionais de redes, os profissionais que trabalham na área de sistemas e os provedores de serviços também precisam estabelecer plano de adaptação de sistemas para suporte ao IPv6. Ademais, a maioria dos grandes provedores de Internet ainda não oferecem suporte nativo nem serviço IPv6 para os clientes, exigindo assim, o condicionamento do serviço prestado à demanda pela entrega de tráfego IPv6. De forma complementar, é preciso que os profissionais de redes e sistemas, além de equipes de apoio técnico em redes, sejam treinados, pois, não possuem conhecimento suficiente de IPv6 para suporte técnico.

Contudo, a administração da atribuição de endereços IPv4 e IPv6 na REDUnB precisa ocorrer preferencialmente por meio de mecanismos automáticos, em especial por se tratar de ambiente de rede com grande número de clientes. Nesta tarefa seguramente o uso dos protocolos DHCPv4 e DHCPv6 oferecem uma solução mais escalável e segura, e que cabe como sugestão para estudo futuro, além de esforço aplicado sobre a segurança no protocolo IPv6.

Ademais, este trabalho pode ser melhorado com a disponibilização no ambiente de laboratório de computadores mais robustos para os experimentos, com a elaboração de avaliações de outros serviços de amplo uso na REDUnB além de implementação de tráfego IPv6 no âmbito da saída para a Internet.

O atual panorama de implantação permite concluir que grande parte das redes ainda não está apta a receber o novo protocolo IPv6. Ainda por um tempo indefinido, as duas versões continuarão existindo, de forma que os mecanismos de transição garantirão a interoperabilidade entre elas.

Porém, espera-se que este trabalho possa servir de incentivo a discussões sobre a adoção do IPv6, sendo também útil à quem deseja obter informações a respeito da transição, mas que cada ambiente de rede possui suas peculiaridades e exigem soluções igualmente com suas particularidades.

Referências

[1] L. ZIMU; P. WEI e L. YUJUN. An innovative ipv4-ipv6 transition way for internet service provider. http://ieeexplore.ieee.org/stamp/stamp.jsp-arnumber-6219279, nov 2012. xiii, 44, 45, 50

[2] Silvia HAGEN. *IPv6 Essentials – Integrating IPv6 into Your IPv4 Network*. O'Reilly Media Inc., 3rd edition, 2014. 1, 2, 14, 15, 18, 19, 22, 23, 25, 29, 33, 35, 39, 40, 42, 50

[3] Secretaria de Logística e Tecnologia da Informação Ministério do Planejamento Orçamento e Gestão. Padrões de interoperabilidade de governo eletrônico e-ping, versão 2015. http://eping.governoeletronico.gov.br, jun 2015. 3

[4] Bruce HARTPENCE. *Packet Guide to Core Network Protocols*. O'Reilly Media Inc., 1st edition, 2011. 5

[5] Andrew S. TANENBAUM. *Computer Networks*. Editora Campus, 4th edition, 2003. 6, 7, 8, 73

[6] Douglas E. COMER. *Interligação em Rede com TCP/IP: Princípios, protocolos e arquitetura*. Editora Elsevier, 3rd edition, 1998. 6

[7] James F. KUROSE e Keith W. ROSS. *Redes de computadores e a Internet: uma abordagem top-down*. Pearson Education, 6th edition, 2013. 8, 10, 26, 29, 68

[8] V. FULLER e T. LI. Request for comments 4632. classless inter-domain routing (cidr): The internet address assignment and aggregation plan. https://tools.ietf.org/html/rfc4632, aug 2006. 9

[9] R. DROMS. Request for comments 2131. dynamic host configuration protocol. https://tools.ietf.org/html/rfc2131, mar 1997. 9

[10] K. EGEVANG e P. FRANCIS. Request for comments 1631. the ip network address translator (nat). https://tools.ietf.org/html/rfc1631, may 1994. 9

[11] Y. REKHTER; B. MOSKOWITZ; G. J. de GROOT; D. KARRENBERG e E. LEAR. Request for comments 1918. address allocation for private internets. https://tools.ietf.org/html/rfc1918, feb 1996. 9

[12] Pete LOSHIN. *IPv6: Theory, Protocol, and Practice*. Morgan Kaufmann Publishers, 2nd edition, 2004. 10, 27

[13] Gary A. DONAHUE. *Network Warrior*. O'Reilly Media, 2nd edition, 2011. 10, 29, 67, 68

[14] Lawrence E. HUGHES. *The Second Internet - Reinventing Computer Networking with IPv6*. InfoWeapons, 1st edition, 2010. 11, 17, 18, 19, 21, 26, 28, 32, 37, 39

[15] Iljitsch van BEIJNUM. *Running IPv6*. Apress, 1st edition, 2006. 12, 20, 30, 34

[16] Qing LI; Tatuya JINMEI e Keiichi SHIMA. *IPv6 Core Protocols Implementation*. Morgan Kaufmann Publishers, 1st edition, 2007. 12

[17] B. CAIN; S. DEERING; I. KOUVELAS; B. FENNER e A. THYAGARAJAN. Request for comments 3376. internet group management protocol, version 3. https://tools.ietf.org/html/rfc3376, oct 2002. 13

[18] S. DEERING; W. FENNER e B. HABERMAN. Request for comments 2710. multicast listener discovery (mld) for ipv6. https://tools.ietf.org/html/rfc2710, oct 1999. 13

[19] Niall Richard MURPHY e David MALONE. *IPv6 Network Administration*. O'Reilly Media, 1st edition, 2005. 13, 22

[20] Adilson Aparecido FLORENTINO. *IPv6 na Prática*. New Media do Brasil Editora Ltda, 1st edition, 2012. 14

[21] M. BLANCHET. Request for comments 3531. a flexible method for managing the assignment of bits of an ipv6 address block. https://tools.ietf.org/html/rfc3531, apr 2003. 14, 56

[22] S. DEERING e R. HINDEN. Request for comments 2460. internet protocol, version 6 (ipv6) specification. https://tools.ietf.org/html/rfc2460, dec 1998. 15, 17

[23] K. NICHOLS; S. BLAKE; F. BAKER e D. BLACK. Request for comments 2474. definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. https://tools.ietf.org/html/rfc2474, dec 1998. 15

[24] K. RAMAKRISHNAN; S. FLOYD e D. BLACK. Request for comments 3168. the addition of explicit congestion notification (ecn) to ip. https://tools.ietf.org/html/rfc3168, sep 2001. 15

[25] J. REYNOLDS e J. POSTEL. Request for comments 1700. assigned numbers. https://tools.ietf.org/html/rfc1700, oct 1994. 16

[26] A. CONTA; S. DEERING e M. GUPTA. Request for comments 4443. internet control message protocol (icmpv6) for the internet protocol version 6 (ipv6) specification. https://tools.ietf.org/html/rfc4443, mar 2006. 18

[27] Daniel MINOLI e John J. AMOSS. *Handbook of IPv4 to IPv6 Transition, Methodologies for Institutional and Corporate Networks*. Auerbach Publications Taylor e Francis Group, 1st edition, 2008. 22, 36, 37, 39

[28] R. DROMS; J. BOUND; B. VOLZ; T. LEMON; C. PERKINS e M. CARNEY. Request for comments 3315. dynamic host configuration protocol for ipv6 (dhcpv6). https://tools.ietf.org/html/rfc3315, jul 2003. 22

[29] Sheila FRANKEL; Richard GRAVEMAN; John PEARCE e Mark ROOKS. *Guidelines for the Secure Deployment of IPv6 - Recommendations of the National Institute of Standards and Technology*. NIST, 1st edition, 2010. 26, 30, 31, 35, 37, 39

[30] R. COLTUN; D. FERGUSON; J. MOY e A. LINDEM. Request for comments 5340. ospf for ipv6. https://tools.ietf.org/html/rfc5340, jul 2008. 27, 28

[31] S. Y. REKHTER; LI, T. e HARES. Request for comments 4271. a border gateway protocol 4 (bgp-4). https://tools.ietf.org/html/rfc4271, jan 2006. 29

[32] T. BATES; R. CHANDRA; D. KATZ e Y. REKHTER. Request for comments 4760. multiprotocol extensions for bgp-4. https://tools.ietf.org/html/rfc4760, jan 2007. 29

[33] S. KENT e R. ATKINSON. Request for comments 2401. security architecture for the internet protocol. https://tools.ietf.org/html/rfc2401, nov 1998. 31

[34] John e Sons WILEY. *Networking Fundamentals*. Wiley, 1st edition, 2011. 31

[35] M-K. SHIN; Y-G. HONG; J-I I. HAGINO; P. SAVOLA e E. M. CASTRO. Request for comments 4038. application aspects of ipv6 transition. https://tools.ietf.org/html/rfc4038, mar 2005. 33

[36] E. NORDMARK. Request for comments 2765. stateless ip/icmp translation algorithm (siit). https://tools.ietf.org/html/rfc2765, feb 2000. 36

[37] C. AOUN e E. DAVIES. Request for comments 4966. reasons to move the network address translator – protocol translator (nat-pt) to historic status. https://tools.ietf.org/html/rfc4966, jul 2007. 36

[38] J. HAGINO e K. YAMAMOTO. Request for comments 3142. an ipv6-to-ipv4 transport relay translator. https://tools.ietf.org/html/rfc3142, jun 2001. 36

[39] S. LEE; M-K. SHIN; Y-J. KIM; E. NORDMARK e A. DURAND. Request for comments 3338. dual stack hosts using "bump-in-the-api" (bia). https://tools.ietf.org/html/rfc3338, oct 2002. 37

[40] Dan YORK. *Migrating Applications to IPv6*. O'Reilly Media, 1st edition, 2011. 38, 42, 43

[41] S. THOMSON; C. HUITEMA; V. KSINANT e M. SOUISSI. Request for comments 3596. dns extensions to support ip version 6. https://tools.ietf.org/html/rfc3596, oct 2003. 42

[42] Joseph DAVIES. *Understanding IPv6*. O'Reilly Media, 3rd edition, 2012. 42

[43] D. WING e A. YOURTCHENKO. Request for comments 6555. happy eyeballs: Success with dual-stack hosts. https://tools.ietf.org/html/rfc6555, apr 2012. 43

[44] Cricket LIU. *DNS and BIND on IPv6*. O'Reilly Media, 1st edition, 2011. 43

[45] H. HOU; Q. ZHAO e Y. MA. Design and implementation of a solution to smooth ipv6 transition. http://ieeexplore.ieee.org/xpls/absall.jsp-arnumber-5696883tag-1, jul 2010. 44, 46, 47, 50

[46] P. WU; Y. CUI; J. WU; J. LIU e C. METZ. Transition from ipv4 to ipv6, a state-of-the-art survey. http://ieeexplore.ieee.org/stamp/stamp.jsp-arnumber-6380492, jun 2013. 44, 47, 48, 50

[47] Fluke Corporation. Datasheet: Optiview xg network analysis tablet - traffic and packet analysis. http://pt.flukenetworks.com/content/optiview-xg-network-analysis-tablet, oct 2014. 66

[48] Yan CHEN; Toni FARLEY e Nong YE. Qos requirements of network applications on the internet. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.475.9277, sep 2004. 67, 68, 76

[49] F. FLUCKIGER. *Understanding Networked Multimedia*. Prentice Hall, 1st edition, 1995. 68, 75

[50] N. YE. Qos-centric stateful resource management in information systems. information systems frontiers. http://dl.acm.org/citation.cfm?id=595230, oct 2002. 68

[51] Sivannarayana NAGIREDDI. *VoIP Voice and Fax Signal Processing*. Wiley, 1st edition, 2008. 68, 69

[52] B. O. SZUPROWICZ. *Multimedia Networking*. McGraw-Hill, 1st edition, 1995. 75

[53] J. A. BERGSTRA e C. A. MIDDELBURG. Itu-t recommendation g.107. the e-model, a computational model for use in transmission planning. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.103.3547, nov 2003. 76

Parte I

Apêndices

Apêndice A

Configurações de *switch* **R1** no *backbone* do laboratório prático:

R1

begin
set vlan create 10
set vlan create 701
set vlan create 4001
set vlan create 4002
set vlan create 4003
set vlan name 10 "ENLACE-R1-AGREGACAO"
set vlan name 701 "GER"
set vlan name 4001 "ENLACE-R1-R2"
set vlan name 4002 "ENLACE-R1-R3"
set vlan name 4003 "ENLACE-R1-R4"
clear vlan egress 1 ge.1.1-2;ge.1.22-24
set vlan egress 10 ge.1.2 untagged
set vlan egress 701 ge.1.1 untagged
set vlan egress 4001 ge.1.24 untagged
set vlan egress 4002 ge.1.23 untagged
set vlan egress 4003 ge.1.22 untagged
set host vlan 701
router
enable
configure
ipv6 unicast-routing
interface vlan 10
ip address 10.11.11.1 255.255.255.0
ip ospf areaid 0.0.0.1
ip ospf enable

```
ipv6 address fec0:11:11:11::1/64
ipv6 ospf areaid 0.0.0.5
ipv6 ospf enable
no shutdown
exit
interface vlan 4001
ip address 10.1.1.1 255.255.255.0
ip ospf enable
ipv6 address fec0:1:1:1::1/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
interface vlan 4002
ip address 10.2.2.1 255.255.255.0
ip ospf enable
ipv6 address fec0:2:2:2::1/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
interface vlan 4003
ip address 10.3.3.1 255.255.255.0
ip ospf enable
ipv6 address fec0:3:3:3::1/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
router id 10.10.10.10
router ospf 1
area 0.0.0.1 nssa
exit
ipv6 router id 110.110.110.110
ipv6 router ospf
area 0.0.0.5 nssa
exit
```

exit
exit
exit
set ipv6 enable
set port vlan ge.1.1 701
set port vlan ge.1.2 10
set port vlan ge.1.22 4003
set port vlan ge.1.23 4002
set port vlan ge.1.24 4001
set spantree version rstp
set spantree portadmin ge.1.2 disable
set spantree portadmin ge.1.22 disable
set spantree portadmin ge.1.23 disable
set spantree portadmin ge.1.24 disable
set spantree priority 0 0
end

Configurações de *switch* **R2** no *backbone* do laboratório prático:

R2

begin
set vlan create 20
set vlan create 701
set vlan create 4001
set vlan create 4004
set vlan create 4005
set vlan name 20 "ENLACE-R2-AGREGACAO"
set vlan name 701 "GER"
set vlan name 4001 "ENLACE-R2-R1"
set vlan name 4004 "ENLACE-R2-R3"
set vlan name 4005 "ENLACE-R2-R4"
clear vlan egress 1 ge.1.1-2;ge.1.22-24
set vlan egress 20 ge.1.2 untagged
set vlan egress 701 ge.1.1 tagged
set vlan egress 4001 ge.1.24 untagged
set vlan egress 4004 ge.1.23 untagged
set vlan egress 4005 ge.1.22 untagged
set host vlan 701
router

```
enable
configure
ipv6 unicast-routing
interface vlan 20
ip address 10.22.22.1 255.255.255.0
ip ospf areaid 0.0.0.2
ip ospf enable
ipv6 address fec0:22:22:22::1/64
ipv6 ospf areaid 0.0.0.6
ipv6 ospf enable
no shutdown
exit
interface vlan 4001
ip address 10.1.1.2 255.255.255.0
ip ospf enable
ipv6 address fec0:1:1:1::2/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
interface vlan 4004
ip address 10.4.4.1 255.255.255.0
ip ospf enable
ipv6 address fec0:4:4:4::1/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
interface vlan 4005
ip address 10.5.5.1 255.255.255.0
ip ospf enable
ipv6 address fec0:5:5:5::1/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
router id 20.20.20.20
```

router ospf 1
area 0.0.0.2 nssa
exit
ipv6 router id 120.120.120.120
ipv6 router ospf
area 0.0.0.6 nssa
exit
exit
exit
exit
set ipv6 enable
set port vlan ge.1.1 701
set port vlan ge.1.2 20
set port vlan ge.1.22 4005
set port vlan ge.1.23 4004
set port vlan ge.1.24 4001
set spantree version rstp
set spantree portadmin ge.1.2 disable
set spantree portadmin ge.1.22 disable
set spantree portadmin ge.1.23 disable
set spantree portadmin ge.1.24 disable
set spantree priority 0 0
end

Configurações de *switch* **R3** no *backbone* do laboratório prático:

R3

begin
set vlan create 30
set vlan create 701
set vlan create 4002
set vlan create 4004
set vlan create 4006
set vlan name 30 "ENLACE-R3-AGREGACAO"
set vlan name 701 "GER"
set vlan name 4002 "ENLACE-R3-R1"
set vlan name 4004 "ENLACE-R3-R2"
set vlan name 4006 "ENLACE-R3-R4"
clear vlan egress 1 ge.1.1-2;ge.1.22-24

set vlan egress 30 ge.1.2 untagged
set vlan egress 701 ge.1.1 tagged
set vlan egress 4002 ge.1.24 untagged
set vlan egress 4004 ge.1.23 untagged
set vlan egress 4006 ge.1.22 untagged
set host vlan 701
router
enable
configure
ipv6 unicast-routing
interface vlan 30
ip address 10.33.33.1 255.255.255.0
ip ospf areaid 0.0.0.3
ip ospf enable
ipv6 address fec0:33:33:33::1/64
ipv6 ospf areaid 0.0.0.7
ipv6 ospf enable
no shutdown
exit
interface vlan 4002
ip address 10.2.2.2 255.255.255.0
ip ospf enable
ipv6 address fec0:2:2:2::2/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
interface vlan 4004
ip address 10.4.4.2 255.255.255.0
ip ospf enable
ipv6 address fec0:4:4:4::2/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
interface vlan 4006
ip address 10.6.6.1 255.255.255.0

ip ospf enable
ipv6 address fec0:6:6:6::1/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
router id 30.30.30.30
router ospf 1
area 0.0.0.3 nssa
exit
ipv6 router id 130.130.130.130
ipv6 router ospf
area 0.0.0.7 nssa
exit
exit
exit
exit
set ipv6 enable
set port vlan ge.1.1 701
set port vlan ge.1.2 30
set port vlan ge.1.22 4006
set port vlan ge.1.23 4004
set port vlan ge.1.24 4002
set spantree disable
set spantree version rstp
set spantree portadmin ge.1.2 disable
set spantree portadmin ge.1.22 disable
set spantree portadmin ge.1.23 disable
set spantree portadmin ge.1.24 disable
set spantree priority 0 0
end

Configurações de *switch* **R4** no *backbone* do laboratório prático:

R4

begin
set vlan create 40
set vlan create 701
set vlan create 4003

```
set vlan create 4005
set vlan create 4006
set vlan name 40 "ENLACE-R4-AGREGACAO"
set vlan name 701 "GER"
set vlan name 4003 "ENLACE-R4-R1"
set vlan name 4005 "ENLACE-R4-R2"
set vlan name 4006 "ENLACE-R4-R3"
clear vlan egress 1 ge.1.1-2;ge.1.22-24
set vlan egress 40 ge.1.2 untagged
set vlan egress 701 ge.1.1 tagged
set vlan egress 4003 ge.1.24 untagged
set vlan egress 4005 ge.1.23 untagged
set vlan egress 4006 ge.1.22 untagged
set host vlan 701
router
enable
configure
ipv6 unicast-routing
interface vlan 40
ip address 10.44.44.1 255.255.255.0
ip ospf areaid 0.0.0.4
ip ospf enable
ipv6 address fec0:44:44:44::1/64
ipv6 ospf areaid 0.0.0.8
ipv6 ospf enable
no shutdown
exit
interface vlan 4003
ip address 10.3.3.2 255.255.255.0
ip ospf enable
ipv6 address fec0:3:3:3::2/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
interface vlan 4005
ip address 10.5.5.2 255.255.255.0
```

```
ip ospf enable
ipv6 address fec0:5:5:5::2/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
interface vlan 4006
ip address 10.6.6.2 255.255.255.0
ip ospf enable
ipv6 address fec0:6:6:6::2/64
ipv6 ospf areaid 0.0.0.0
ipv6 ospf enable
no shutdown
exit
router id 40.40.40.40
router ospf 1
area 0.0.0.4 nssa
exit
ipv6 router id 140.140.140.140
ipv6 router ospf
area 0.0.0.8 nssa
exit
exit
exit
exit
set ipv6 enable
set port vlan ge.1.1 701
set port vlan ge.1.2 40
set port vlan ge.1.22 4006
set port vlan ge.1.23 4005
set port vlan ge.1.24 4003
set spantree version rstp
set spantree portadmin ge.1.2 disable
set spantree portadmin ge.1.22 disable
set spantree portadmin ge.1.23 disable
set spantree portadmin ge.1.24 disable
set spantree priority 0 0
```

end

Configurações de *switch* de agregação **Agregação R1** do laboratório prático:

Agregação R1

```
begin
set vlan create 10
set vlan create 701
set vlan name 10 "ENLACE-AGREGACAO-R1"
set vlan name 100 "REDE-1"
set vlan name 701 "GER"
set vlan egress 10 ge.1.14 untagged
set vlan egress 100 ge.1.1-12 untagged
set vlan egress 701 ge.1.13 tagged
set host vlan 701
router
enable
configure
ipv6 unicast-routing
interface vlan 10
ip address 10.11.11.2 255.255.255.0
ip ospf areaid 0.0.0.1
ip ospf enable
ipv6 address fec0:11:11:11::2/64
ipv6 ospf areaid 0.0.0.5
ipv6 ospf enable
no shutdown
exit
interface vlan 100
ip address 192.168.11.1 255.255.255.0
ip ospf areaid 0.0.0.1
ip ospf enable
ipv6 address 2801:80:b90::1/64
ipv6 ospf areaid 0.0.0.5
ipv6 ospf enable
no shutdown
exit
interface loopback 1
ip address 10.1.1.1 255.255.255.255
```

ip ospf areaid 0.0.0.1
ip ospf enable
no shutdown
exit
router id 10.1.1.1
router ospf 1
area 0.0.0.1 nssa
redistribute connected subnets
exit
ipv6 router id 110.101.101.101
ipv6 router ospf
area 0.0.0.5 nssa
redistribute connected
exit
exit
exit
exit
set ipv6 enable
set port vlan ge.1.1 100
set port vlan ge.1.2 100
set port vlan ge.1.3 100
set port vlan ge.1.4 100
set port vlan ge.1.5 100
set port vlan ge.1.6 100
set port vlan ge.1.7 100
set port vlan ge.1.8 100
set port vlan ge.1.9 100
set port vlan ge.1.10 100
set port vlan ge.1.11 100
set port vlan ge.1.12 100
set port vlan ge.1.13 701
set port vlan ge.1.14 10
set port vlan ge.1.15 10
set port vlan ge.1.18 10
set port vlan ge.1.21 10
set port vlan ge.1.22 10
set port vlan ge.1.23 10

set port vlan ge.1.24 10
set port mirroring create ge.1.14 ge.1.15
set port mirroring enable ge.1.14 ge.1.15
end

Configurações de *switch* de agregação **Agregação R2** do laboratório prático:

Agregação R2

begin
set vlan create 20
set vlan create 200
set vlan create 201
set vlan create 701
set vlan name 20 "ENLACE-AGREGACAO-R2"
set vlan name 200 "REDE-2"
set vlan name 201 "REDE-2-IPv4Only"
set vlan name 701 "GER"
set vlan egress 20 ge.1.14 untagged
set vlan egress 200 ge.1.1-12 untagged
set vlan egress 201 ge.1.22-24 untagged
set vlan egress 701 ge.1.13 tagged
set host vlan 701
router
enable
configure
ipv6 unicast-routing
interface vlan 20
ip address 10.22.22.2 255.255.255.0
ip ospf areaid 0.0.0.2
ip ospf enable
ipv6 address fec0:22:22:22::2/64
ipv6 ospf areaid 0.0.0.6
ipv6 ospf enable
no shutdown
exit
interface vlan 200
ip address 192.168.21.1 255.255.255.0
ip ospf areaid 0.0.0.2
ip ospf enable

ipv6 address 2801:80:b90:4000::1/64
ipv6 ospf areaid 0.0.0.6
ipv6 ospf enable
no shutdown
exit
interface vlan 201
ip address 192.168.22.1 255.255.255.0
ip ospf areaid 0.0.0.2
ip ospf enable
no shutdown
exit
interface loopback 1
ip address 10.2.2.1 255.255.255.255
ip ospf areaid 0.0.0.2
ip ospf enable
no shutdown
exit
router id 10.2.2.1
router ospf 1
area 0.0.0.2 nssa
redistribute connected subnets
exit
ipv6 router id 110.102.102.101
ipv6 router ospf
area 0.0.0.6 nssa
redistribute connected
exit
exit
exit
exit
set ipv6 enable
set port vlan ge.1.1 200
set port vlan ge.1.2 200
set port vlan ge.1.3 200
set port vlan ge.1.4 200
set port vlan ge.1.5 200
set port vlan ge.1.6 200

set port vlan ge.1.7 200
set port vlan ge.1.8 200
set port vlan ge.1.9 200
set port vlan ge.1.10 200
set port vlan ge.1.11 200
set port vlan ge.1.12 200
set port vlan ge.1.13 701
set port vlan ge.1.14 20
set port vlan ge.1.22 201
set port vlan ge.1.23 201
set port vlan ge.1.24 201
set port mirroring create ge.1.14 ge.1.15
set port mirroring enable ge.1.14 ge.1.15
end

Configurações de *switch* de agregação **Agregação R3** do laboratório prático:

Agregação R3

begin
set vlan create 30
set vlan create 300
set vlan create 301
set vlan create 701
set vlan name 30 "ENLACE-AGREGACAO-R3"
set vlan name 300 "REDE-3"
set vlan name 301 "REDE-3-IPv6Only"
set vlan name 701 "GER"
set vlan egress 30 ge.1.14 untagged
set vlan egress 300 ge.1.1-12 untagged
set vlan egress 301 ge.1.22-24 untagged
set vlan egress 701 ge.1.13 tagged
set host vlan 701
router
enable
configure
ipv6 unicast-routing
interface vlan 30
ip address 10.33.33.2 255.255.255.0
ip ospf areaid 0.0.0.3

ip ospf enable
ipv6 address fec0:33:33:33::2/64
ipv6 ospf areaid 0.0.0.7
ipv6 ospf enable
no shutdown
exit
interface vlan 300
ip address 192.168.31.1 255.255.255.0
ip ospf areaid 0.0.0.3
ip ospf enable
ipv6 address 2801:80:b90:8000::1/64
ipv6 ospf areaid 0.0.0.7
ipv6 ospf enable
no shutdown
exit
interface vlan 301
ipv6 address 2801:80:b90:8001::1/64
ipv6 ospf areaid 0.0.0.7
ipv6 ospf enable
no shutdown
exit
interface loopback 1
ip address 10.3.3.1 255.255.255.255
ip ospf areaid 0.0.0.3
ip ospf enable
no shutdown
exit
router id 10.3.3.1
router ospf 1
area 0.0.0.3 nssa
redistribute connected subnets
exit
ipv6 router id 110.103.103.101
ipv6 router ospf
area 0.0.0.7 nssa
redistribute connected
exit

exit
exit
exit
set ipv6 enable
set port vlan ge.1.1 300
set port vlan ge.1.2 300
set port vlan ge.1.3 300
set port vlan ge.1.4 300
set port vlan ge.1.5 300
set port vlan ge.1.6 300
set port vlan ge.1.7 300
set port vlan ge.1.8 300
set port vlan ge.1.9 300
set port vlan ge.1.10 300
set port vlan ge.1.11 300
set port vlan ge.1.12 300
set port vlan ge.1.13 701
set port vlan ge.1.14 30
set port vlan ge.1.22 301
set port vlan ge.1.23 301
set port vlan ge.1.24 301
set port mirroring create ge.1.14 ge.1.15
set port mirroring enable ge.1.14 ge.1.15
end

Configurações de *switch* de agregação **Agregação R4** do laboratório prático:

Agregação R4

begin
set vlan create 40
set vlan create 400
set vlan create 701
set vlan name 40 "ENLACE-AGREGACAO-R4"
set vlan name 400 "REDE-4"
set vlan name 701 "GER"
set vlan egress 40 ge.1.14 untagged
set vlan egress 400 ge.1.1-12 untagged
set vlan egress 701 ge.1.13 tagged
set host vlan 701

```
router
enable
configure
ipv6 unicast-routing
interface vlan 40
ip address 10.44.44.2 255.255.255.0
ip ospf areaid 0.0.0.4
ip ospf enable
ipv6 address fec0:44:44:44::2/64
ipv6 ospf areaid 0.0.0.8
ipv6 ospf enable
no shutdown
exit
interface vlan 400
ip address 192.168.41.1 255.255.255.0
ip ospf areaid 0.0.0.4
ip ospf enable
ipv6 address 2801:80:b90:c000::1/64
ipv6 ospf areaid 0.0.0.8
ipv6 ospf enable
no shutdown
exit
interface loopback 1
ip address 10.4.4.1 255.255.255.255
ip ospf areaid 0.0.0.4
ip ospf enable
no shutdown
exit
router id 10.4.4.1
router ospf 1
area 0.0.0.4 nssa
redistribute connected subnets
exit
ipv6 router id 110.104.104.101
ipv6 router ospf
area 0.0.0.8 nssa
redistribute connected
```

```
exit
exit
exit
exit
set ipv6 enable
set port vlan ge.1.1 400
set port vlan ge.1.2 400
set port vlan ge.1.3 400
set port vlan ge.1.4 400
set port vlan ge.1.5 400
set port vlan ge.1.6 400
set port vlan ge.1.7 400
set port vlan ge.1.8 400
set port vlan ge.1.9 400
set port vlan ge.1.10 400
set port vlan ge.1.11 400
set port vlan ge.1.12 400
set port vlan ge.1.13 701
set port vlan ge.1.14 40
set port vlan ge.1.24 40
set port mirroring create ge.1.14 ge.1.15
set port mirroring enable ge.1.14 ge.1.15
end
```

Apêndice B

Configurações do servidor de resolução de nomes (DNS) no laboratório prático:
Arquivo **named.conf**

```
options {
directory "/etc/named/";
listen-on { any; };
listen-on-v6 { any; };
allow-query { any; };
recursion no;
};
zone "."{
type hint;
file "named.root";
};
zone "localhost"{
type master;
file "/etc/bind/db.local";
};
zone "lab.unb.br"{
type master;
file "lab.unb.br.zone";
};
zone "168.192.in-addr.arpa"{
type master;
file "192-168.db";
};
```

Arquivo **lab.unb.br.zone**

```
$TTL 1s
lab.unb.br. IN SOA ns.lab.unb.br. root.lab.unb.br. (
```

19 ; serial
28800 ; refresh
7200 ; retry
604800 ; expire
1s ; ttl
)
;; Servidor que responde pelo dominio
IN NS ns.lab.unb.br.
ns IN A 192.168.41.2
ns IN AAAA 2801:80:b90:c000::2
;; Hosts do LAB
alfa.lab.unb.br. IN A 192.168.11.2
alfa.lab.unb.br. IN AAAA 2801:80:b90:0000::2
beta.lab.unb.br. IN A 192.168.21.2
beta.lab.unb.br. IN AAAA 2801:80:b90:4000::2
gamav4.lab.unb.br. IN A 192.168.31.2
gamav6.lab.unb.br. IN AAAA 2801:80:b90:8000::2
zetav4.lab.unb.br. IN A 192.168.41.2
zetav6.lab.unb.br. IN AAAA 2801:80:b90:c000::2
teta.lab.unb.br. IN A 192.168.22.2
capa.lab.unb.br. IN A 192.168.31.3
Comando para iniciar serviço do servidor de resolução de nomes (DNS)
sudo named -c /etc/named.conf

Apêndice C

Configurações do serviço de voz sobre IP em *software* **Asterisk**:
Comandos para iniciar e parar serviço do *software* **Asterisk**:
Para iniciar: 'sudo /etc/init.d/asterisk start';
Para parar: 'sudo /etc/init.d/asterisk stop';
Arquivo **sip.conf**
;contexto geral, configuracoes gerais dos canais sip
[general]
;iudpbindaddr=0.0.0.0
;disallow=all
;allow=ulaw
;session-timers=refuse
;session-expires=3600
;session-minse=90
;session-refresher=uac
bindaddr=::
;useragent=unbvoip
srvlookup=yes
maxexpirey=120
dafaultexpirey=80
language=pt_BR
nat=force_rport,comedia instead
;qualify=yes
rtcachefriends=yes
canreinvite=no
realm=asterisk
call-limit = 1000
;textsupport=yes
;accept_outofcall_message=yes

;outofcall_message_context=astsms
;TESTES
disallow=all
allow=ulaw
allow=alaw
allow=gsm
;TEMPLATE DE CANAIS SIP SEM TLS
[sip](!)
context=internal
type=friend
host=dynamic
callgroup=10
pickupgroup=10
encryption=no
;;;;;;;;;;;;;;;;;
;CANAIS SIP
[1001](sip)
defaultuser=1001
secret=1001
callerid = Cliente1
[1002](sip)
defaultuser=1002
secret=1002
callerid = Cliente2

Arquivo **sip.conf**

;SECAO QUE DEFINE VALORES PADROES PARA OUTRAS SESSOES
[general]
static=yes
writeprotect=no
autofallthrough=yes
priorityjumping=yes
language=pt_BR
[default]
include => internal
;EXTENSAO GENERICA QUE FAZ LIGACOES PARA TODOS OS RAMAIS
[internal]
exten => _X.,1,Macro(discar,SIP,$EXTEN)

```
    exten => _X.,n,Hangup()
    ;MACRO QUE REALIZA LIGACOES PARA RAMAIS DO ASTERISK
    [macro-discar]
    exten => s,1,Dial($ARG1$ARG2,20,tTwW)
    exten => s,n,Goto(s-$DIALSTATUS,1)
    exten => s-BUSY,1,VoiceMail($ARG2@caixa_msg,b) ;ocupado
    exten => s-NOANSWER,1,VoiceMail($ARG2@caixa_msg,u) ;ele nao atende
    exten => s-CHANUNAVAIL,1,ExecIf($[MAILBOX_EXISTS($ARG2@caixa_msg)]?VoiceMail
($ARG2@caixa_msg,u):Playback(erro))
```

Apêndice D

Com o objetivo de verificar a correta configuração dos equipamentos do laboratório, esta seção expõe os detalhes das certificações de ambiente aplicadas. A partir do *host* **ALFA** foi comprovado o alcance deste aos demais *hosts* do laboratório prático tanto no protocolo IPv4 quanto com o IPv6 como pode ser visto na Figura D.1. O teste de alcance foi executado por meio do comando 'Ping' (ping <endereço IPv4 ou IPv6> -t) que para sistemas operacionais Windows pode ser usado para ambas versões do protocolo IP.

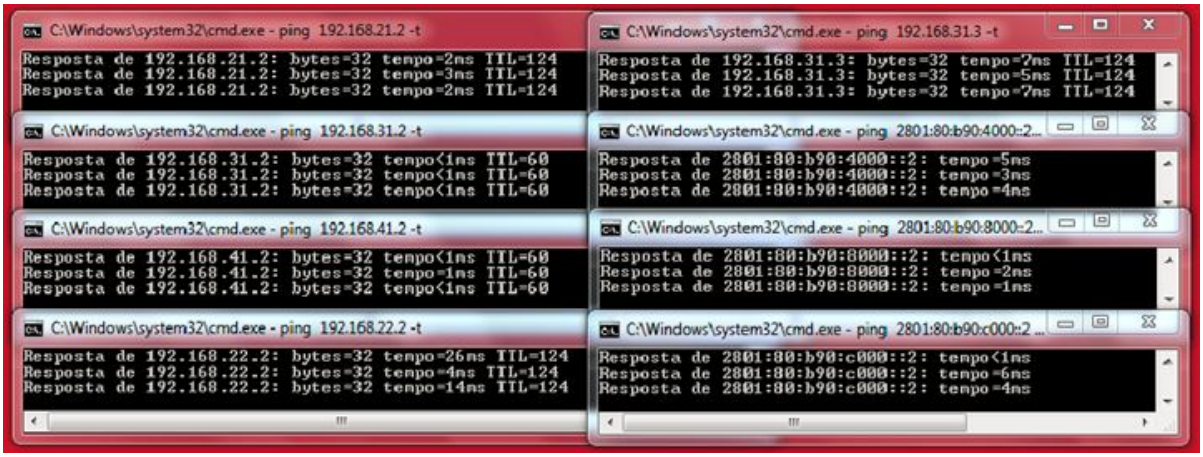


Figura D.1: Alcance do *host* **ALFA** para os demais *hosts* do ambiente de teste por meio do comando 'Ping'

Como se vê na Figura D.2 o *host* **GAMA** alcança o *host* **ALFA**. Importante conferir que o *host* **ALFA** possui registrado no servidor de resolução de nomes 01 (um) nome de domínio (alfa.lab.unb.br) e 02 (dois) registros de endereços IP (192.168.11.2 e 2801:80:b90:0000::2). Constata-se por meio da figura apresentada que embora o protocolo IPv6 possua prioridade de comunicação sobre o protocolo IPv4, na resolução de nome para o *host* **ALFA** foi traduzido para um endereço IPv4. Isto se deve pelo fato de que nos sistemas operacionais, algumas aplicações utilizam o mesmo comando para as 02 (duas) versões do protocolo IP, enquanto em outras não. Como se vê ainda na imagem em ques-

tão, o comando 'Ping6' é utilizado no sistema operacional Ubuntu para alcançar endereços IPv6 e o comando 'Ping' para alcançar endereços IPv4.

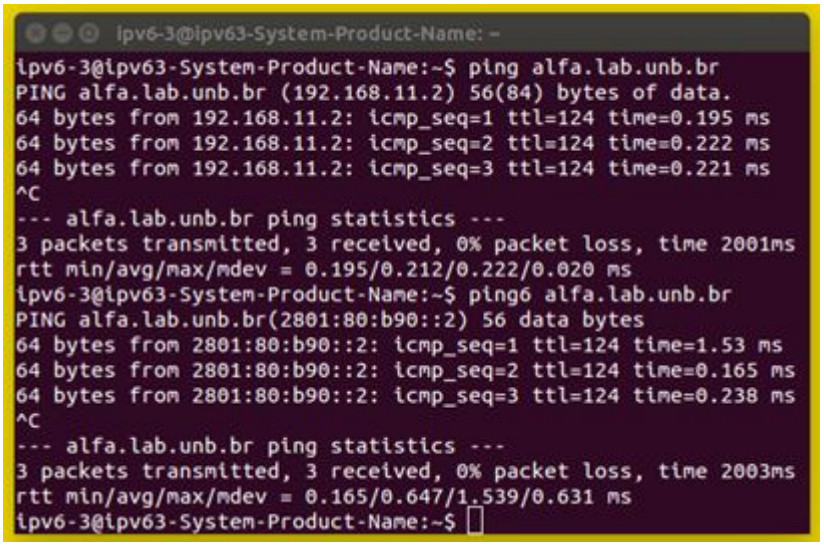


Figura D.2: Alcance do *host* **GAMA** para o *host* **ALFA** por meio do comando 'Ping' e 'Ping6'

Ainda no propósito de demonstrar a prioridade da comunicação pelo protocolo IPv6 entre *hosts* em Pilha Dupla, na Figura D.3 vê-se um exemplo. O *host* **ALFA** ao tentar alcançar o *host* **BETA** por meio do comando 'Ping' e utilizando neste comando o nome de domínio do *host* a alcançar (ping beta.lab.unb.br), ao consultar o servidor de resolução de nomes, a prioridade é dada ao protocolo IPv6, assim, o endereço 2801:80:b90:4000::2 tem prioridade sobre o endereço 192.168.21.2. Pode ser notado também que no sistema operacional Windows o comando 'Ping' pode ser utilizado para ambas versões do protocolo IP.

Figura D.3: *Host* **ALFA** alcança o *host* **BETA** por meio do comando 'Ping'

Como já citado neste documento, compete aos servidores de resolução de nomes (DNS) um papel elementar para o sucesso das redes, sejam elas concernentes ao universo IPv4

ou IPv6. Com a coexistência das 02 (duas) versões do protocolo IP, sua função se torna ainda mais valorosa, facilitando a localização de recursos na rede.

Na Figura D.4 é apresentada uma consulta do *host* **TETA** ao servidor DNS do laboratório (*host* **ZETA**) por informações sobre domínio ou *host*. A consulta ocorre através do comando 'Nslookup' com o objetivo de obter detalhes sobre o nome de domínio alfa.lab.unb.br, retornando além do endereço do servidor DNS (192.168.41.2) consultado, os endereços IPv4 e IPv6 (2801:80:b90:0000::2 e 192.168.11.2) correspondentes ao nome de domínio de interesse. Outra constatação a ser extraída é que *hosts* apenas com IPv4 podem receber detalhes de informações de endereços IPv6, como é o caso do *host* **TETA** que apenas possui endereço IPv4.

Figura D.4: Resolução de nome do *host* **TETA** para o *host* **ALFA**

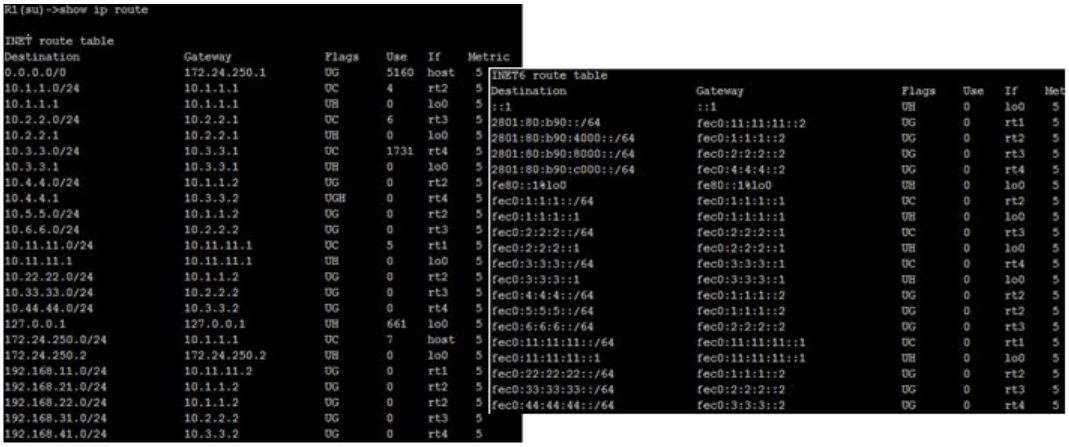
Um aspecto muito importante do uso da técnica de Pilha Dupla nos *hosts* se deve ao fato de que a comunicação precisa ocorrer destes com *hosts* que operam apenas no universo de redes IPv4, a comunicação precisa ser estabelecida de forma estável para o usuário. Vê-se pela Figura D.5 que o *host* **BETA** em Pilha Dupla alcança o *host* **CAPA** somente em IPv4, obviamente, por meio deste último protocolo IP, circunstância que se replica normalmente nas demais comunicações entre *hosts* em Pilha Dupla e somente em IPv4, seja no ambiente de rede local ou deste com a Internet.

Figura D.5: *Host* **BETA** traça rota para o *host* **CAPA** por meio do comando 'Tracert'

Ainda com referência à Figura D.5, é relevante explicitar que a comunicação do *host* **BETA** para o *host* **CAPA** foi executada por meio do comando 'Tracert', o qual expõe os endereços IPs dos *switches* (roteadores) intermediários entre a origem e o destino, isto é, traça a rota entre ambos.

Na perspectiva do roteamento das redes do laboratório prático, como já informado, foram configurados os protocolos OSPF versões 2 e 3. A Figura D.6 demonstra a tabela de

roteamento OSPF do *switch router* **R1**, consulta executada por meio do comando *show ip route*. Embora o comando citado apresente as tabelas das 02 (duas) versões do protocolo OSPF, estas operam de forma independente.



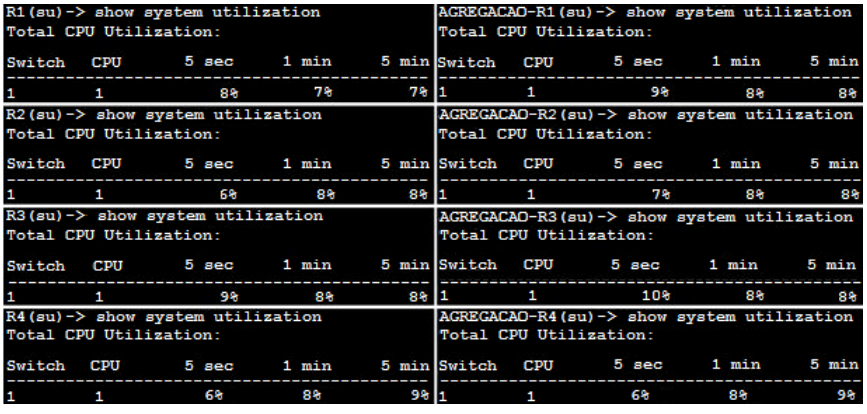
```
R1(su)->show ip route

INET route table
Destination             Gateway           Flags    Use   If    Metric
0.0.0.0/0               172.24.250.1      UG       5160  host  5
10.1.1.0/24             10.1.1.1          UC       4     rt2   5
10.1.1.1                10.1.1.1          UH       0     lo0   5
10.2.2.0/24             10.2.2.1          UC       6     rt3   5
10.2.2.1                10.2.2.1          UH       0     lo0   5
10.3.3.0/24             10.3.3.1          UC       1731  rt4   5
10.3.3.1                10.3.3.1          UH       0     lo0   5
10.4.4.0/24             10.1.1.2          UG       0     rt2   5
10.4.4.1                10.3.3.2          UGH      0     rt4   5
10.5.5.0/24             10.1.1.2          UG       0     rt2   5
10.6.6.0/24             10.2.2.2          UG       0     rt3   5
10.11.11.0/24           10.11.11.1        UC       5     rt1   5
10.11.11.1              10.11.11.1        UH       0     lo0   5
10.22.22.0/24           10.1.1.2          UG       0     rt2   5
10.33.33.0/24           10.2.2.2          UG       0     rt3   5
10.44.44.0/24           10.3.3.2          UG       0     rt4   5
127.0.0.1               127.0.0.1         UH       661   lo0   5
172.24.250.0/24         10.1.1.1          UC       7     host  5
172.24.250.2            172.24.250.2      UH       0     lo0   5
192.168.11.0/24         10.11.11.2        UG       0     rt1   5
192.168.21.0/24         10.1.1.2          UG       0     rt2   5
192.168.22.0/24         10.1.1.2          UG       0     rt2   5
192.168.31.0/24         10.2.2.2          UG       0     rt3   5
192.168.41.0/24         10.3.3.2          UG       0     rt4   5

INET6 route table
Destination                             Gateway                 Flags  Use  If   Met
::1                                     ::1                     UH     0    lo0  5
2801:80:b90::/64                        fec0:11:11:11::2        UG     0    rt1  5
2801:80:b90:4000::/64                   fec0:1:1:1::2           UG     0    rt2  5
2801:80:b90:8000::/64                   fec0:2:2:2::2           UG     0    rt3  5
2801:80:b90:c000::/64                   fec0:4:4:4::2           UG     0    rt4  5
fe80::1%lo0                             fe80::1%lo0             UH     0    lo0  5
fec0:1:1:1::/64                         fec0:1:1:1::1           UC     0    rt2  5
fec0:1:1:1::1                           fec0:1:1:1::1           UH     0    lo0  5
fec0:2:2:2::/64                         fec0:2:2:2::1           UC     0    rt3  5
fec0:2:2:2::1                           fec0:2:2:2::1           UH     0    lo0  5
fec0:3:3:3::/64                         fec0:3:3:3::1           UC     0    rt4  5
fec0:3:3:3::1                           fec0:3:3:3::1           UH     0    lo0  5
fec0:4:4:4::/64                         fec0:1:1:1::2           UG     0    rt2  5
fec0:5:5:5::/64                         fec0:1:1:1::2           UG     0    rt2  5
fec0:6:6:6::/64                         fec0:2:2:2::2           UG     0    rt3  5
fec0:11:11:11::/64                      fec0:11:11:11::1        UC     0    rt1  5
fec0:11:11:11::1                        fec0:11:11:11::1        UH     0    lo0  5
fec0:22:22:22::/64                      fec0:1:1:1::2           UG     0    rt2  5
fec0:33:33:33::/64                      fec0:2:2:2::2           UG     0    rt3  5
fec0:44:44:44::/64                      fec0:3:3:3::2           UG     0    rt4  5
```

Figura D.6: Tabela de rotas consultada no *switch router* **R1**

No decorrer dos testes com o tráfego de fundo em execução, é relevante ter uma visão do nível de utilização da CPU (*Central Processing Unit*) dos *switches* do laboratório prático com o advento do uso dos protocolos IPv4 e IPv6 em Pilha Dupla. A Figura D.7 exibe as respostas ao comando *show system utilization* utilizado em todos os *switches* envolvidos no cenário de testes enquanto o tráfego de fundo é executado. Como se vê, os níveis de utilização de CPU em todos os *switches* apresentam-se baixos, o que aponta comportamento adequado para a operação onde lhe é conferida a habilidade de trabalhar com protocolos adicionais, como o IPv6 e OSPF versão 3.



```
R1(su)-> show system utilization
Total CPU Utilization:

Switch   CPU       5 sec      1 min      5 min
-----------------------------------------------
1        1           8%         7%         7%

AGREGACAO-R1(su)-> show system utilization
Total CPU Utilization:

Switch   CPU       5 sec      1 min      5 min
-----------------------------------------------
1        1           9%         8%         8%

R2(su)-> show system utilization
Total CPU Utilization:

Switch   CPU       5 sec      1 min      5 min
-----------------------------------------------
1        1           6%         8%         8%

AGREGACAO-R2(su)-> show system utilization
Total CPU Utilization:

Switch   CPU       5 sec      1 min      5 min
-----------------------------------------------
1        1           7%         8%         8%

R3(su)-> show system utilization
Total CPU Utilization:

Switch   CPU       5 sec      1 min      5 min
-----------------------------------------------
1        1           9%         8%         8%

AGREGACAO-R3(su)-> show system utilization
Total CPU Utilization:

Switch   CPU       5 sec      1 min      5 min
-----------------------------------------------
1        1          10%         8%         8%

R4(su)-> show system utilization
Total CPU Utilization:

Switch   CPU       5 sec      1 min      5 min
-----------------------------------------------
1        1           6%         8%         9%

AGREGACAO-R4(su)-> show system utilization
Total CPU Utilization:

Switch   CPU       5 sec      1 min      5 min
-----------------------------------------------
1        1           6%         8%         9%
```

Figura D.7: Nível de utilização de CPU nos *switches* do laboratório

Na complementação de certificações de ambiente, o *host* **ALFA** acessa a página *web* do *host* **ZETA** e este a página *web* do *host* **ALFA** como se vê na Figura D.8. Este experimento possui objetivo de primeiramente demonstrar o comportamento da resolução

de nomes entre *hosts* em Pilha Dupla que possuem diferentes entradas de registros no servidor DNS.

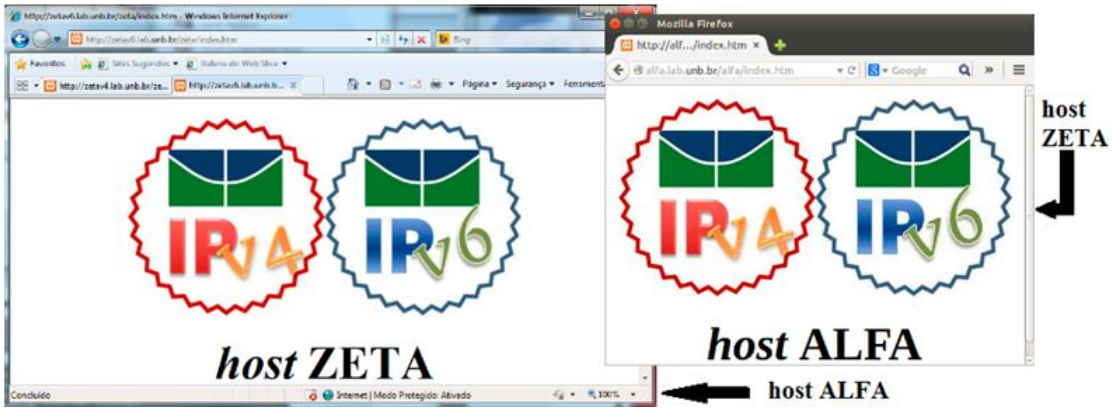


Figura D.8: Acesso mútuo a serviço HTML entre *hosts* **ALFA** e **ZETA**

Importante observar que no servidor de resolução de nomes (DNS) o *host* **ALFA** possui apenas 1 (um) nome de domínio para registros **A** e outro **AAAA**, enquanto que o *host* **ZETA** possui 2 (dois) nomes de domínios, 1 (um) para cada tipo de registro **A** e **AAAA**, como consta na Figura D.9.

```
ipv6-4@ipv6: /etc/named
ipv6-4@ipv6:/etc/named$ cat lab.unb.br.zone
$TTL 1s
lab.unb.br.     IN      SOA     ns.labunb.br.   root.lab.unb.br. (
        19      ;       serial
        28800   ;       refresh
        7200    ;       retry
        604800  ;       expire
        1s      ;       ttl
        )

;; Servidor que responde pelo dominio
                        IN      NS      ns.lab.unb.br.
ns                      IN      A       192.168.41.2
ns                      IN      AAAA    2801:80:b90:c000::2

;; Hosts do LAB
alfa.lab.unb.br.                IN      A       192.168.11.2
alfa.lab.unb.br.                IN      AAAA    2801:80:b90:0000::2
beta.lab.unb.br.                IN      A       192.168.21.2
beta.lab.unb.br.                IN      AAAA    2801:80:b90:4000::2
gamav4.lab.unb.br.              IN      A       192.168.31.2
gamav6.lab.unb.br.              IN      AAAA    2801:80:b90:8000::2
zetav4.lab.unb.br.              IN      A       192.168.41.2
zetav6.lab.unb.br.              IN      AAAA    2801:80:b90:c000::2
teta.lab.unb.br.                IN      A       192.168.22.2
capa.lab.unb.br.                IN      A       192.168.31.3
```

```
ipv6-4@ipv6:/etc/named$ cat named.conf
options {
 directory "/etc/named/";
 listen-on { any; };
 listen-on-v6 { any; };
 allow-query { any; };
 recursion no;
};
```

Figura D.9: Configurações de resolução de nome para a **zona lab.unb.br** e habilitação de IPv6 no DNS

Observa-se ainda na Figura D.9 as configurações do arquivo **named.conf** onde é habilitada a resolução de nomes para endereços IPv4 *listen-on { any; };* e IPv6 *listen-on-v6 { any; };*.

Parte II

Anexos

Anexo A

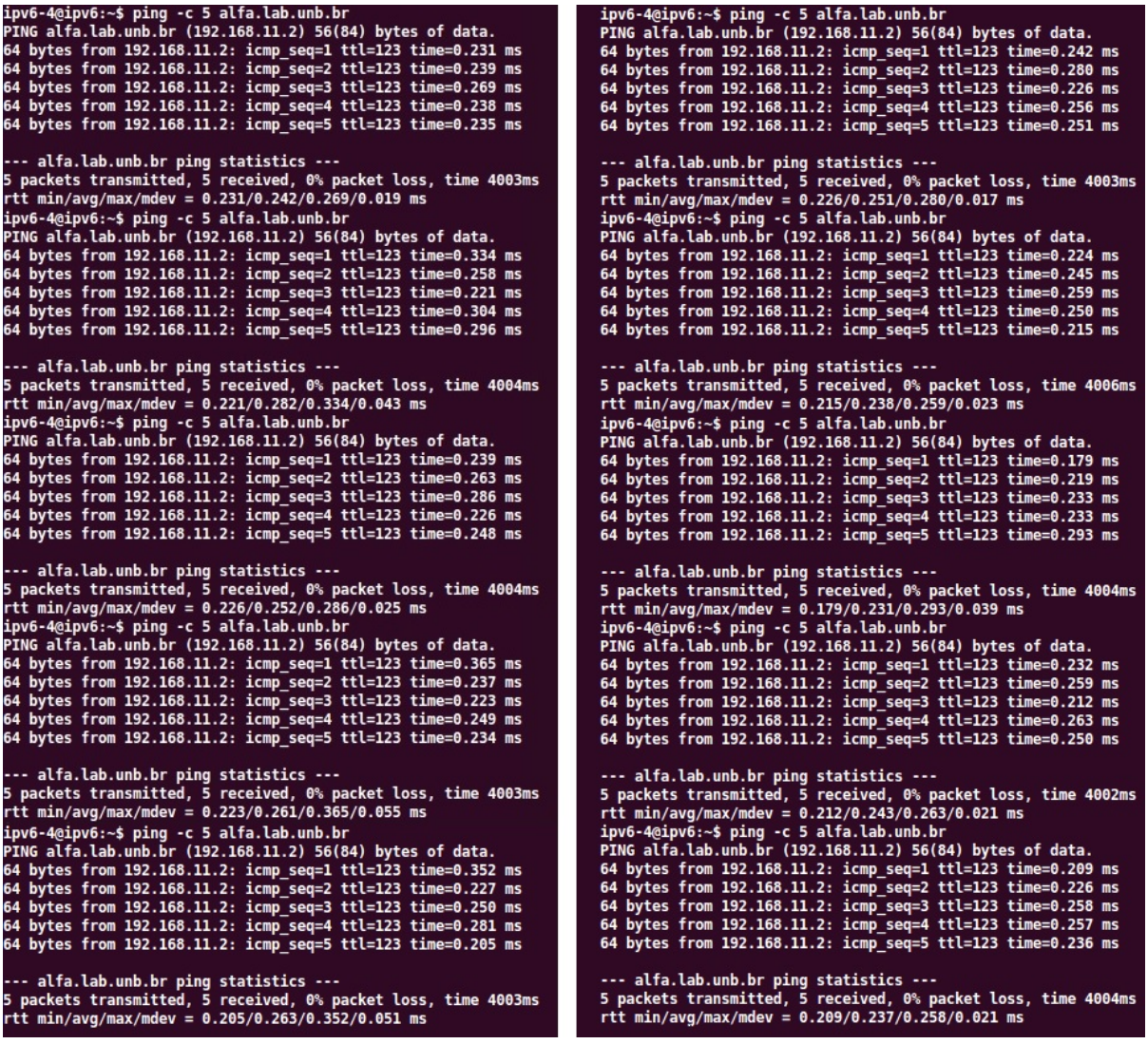


Figura A.1: Grupo de coleta 01 - IPv4

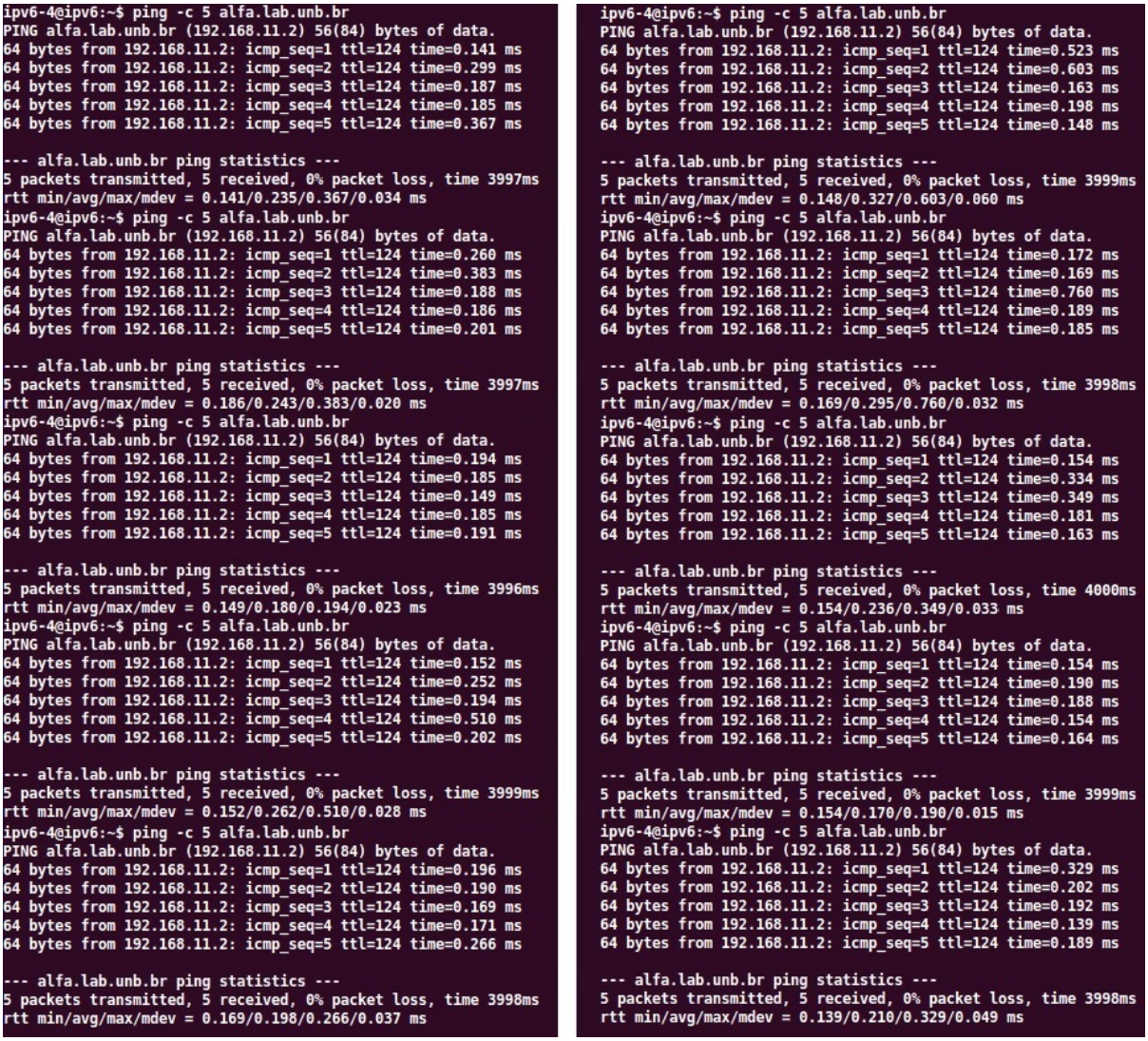


Figura A.2: Grupo de coleta 02 - IPv4

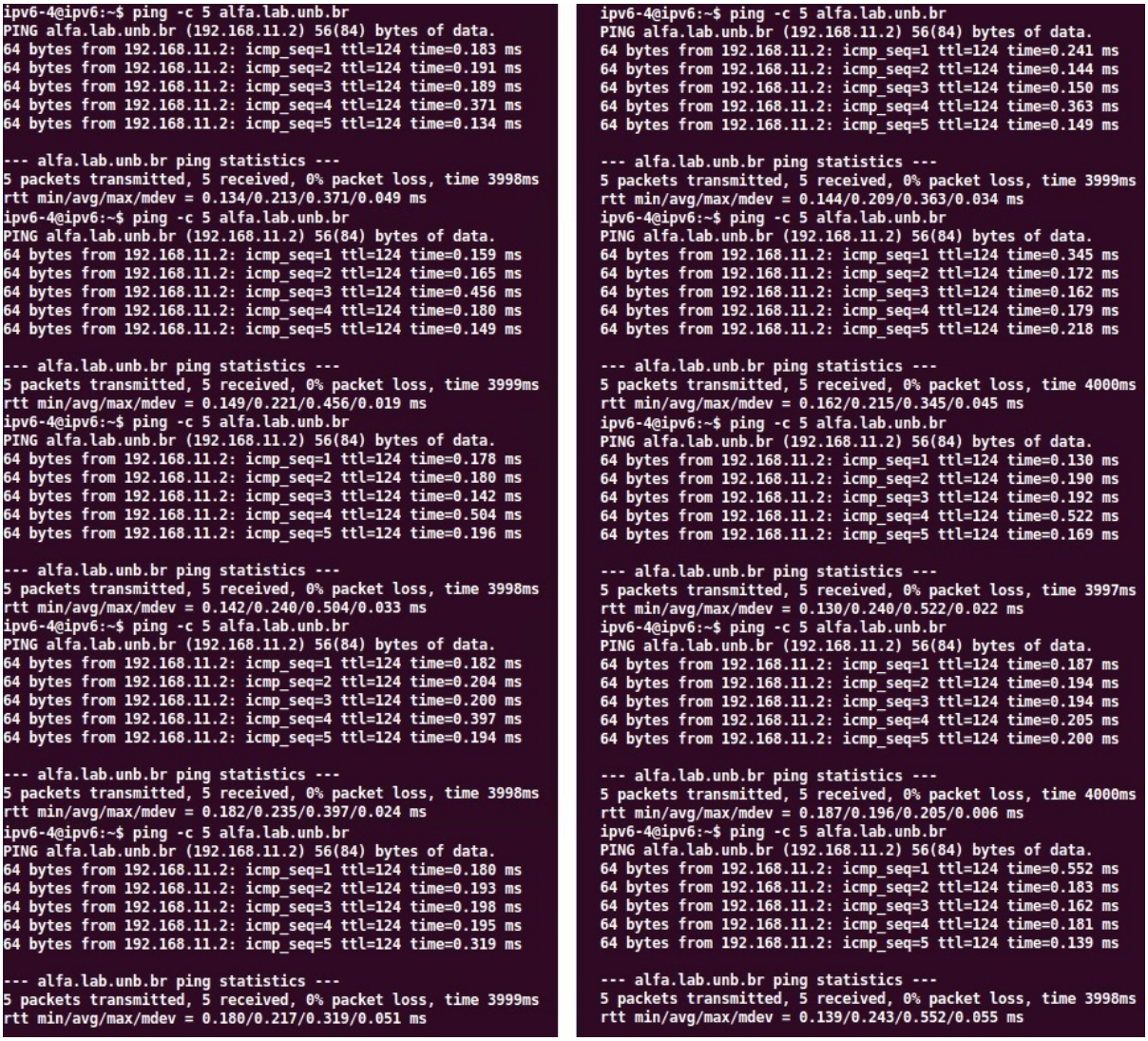


Figura A.3: Grupo de coleta 03 - IPv4

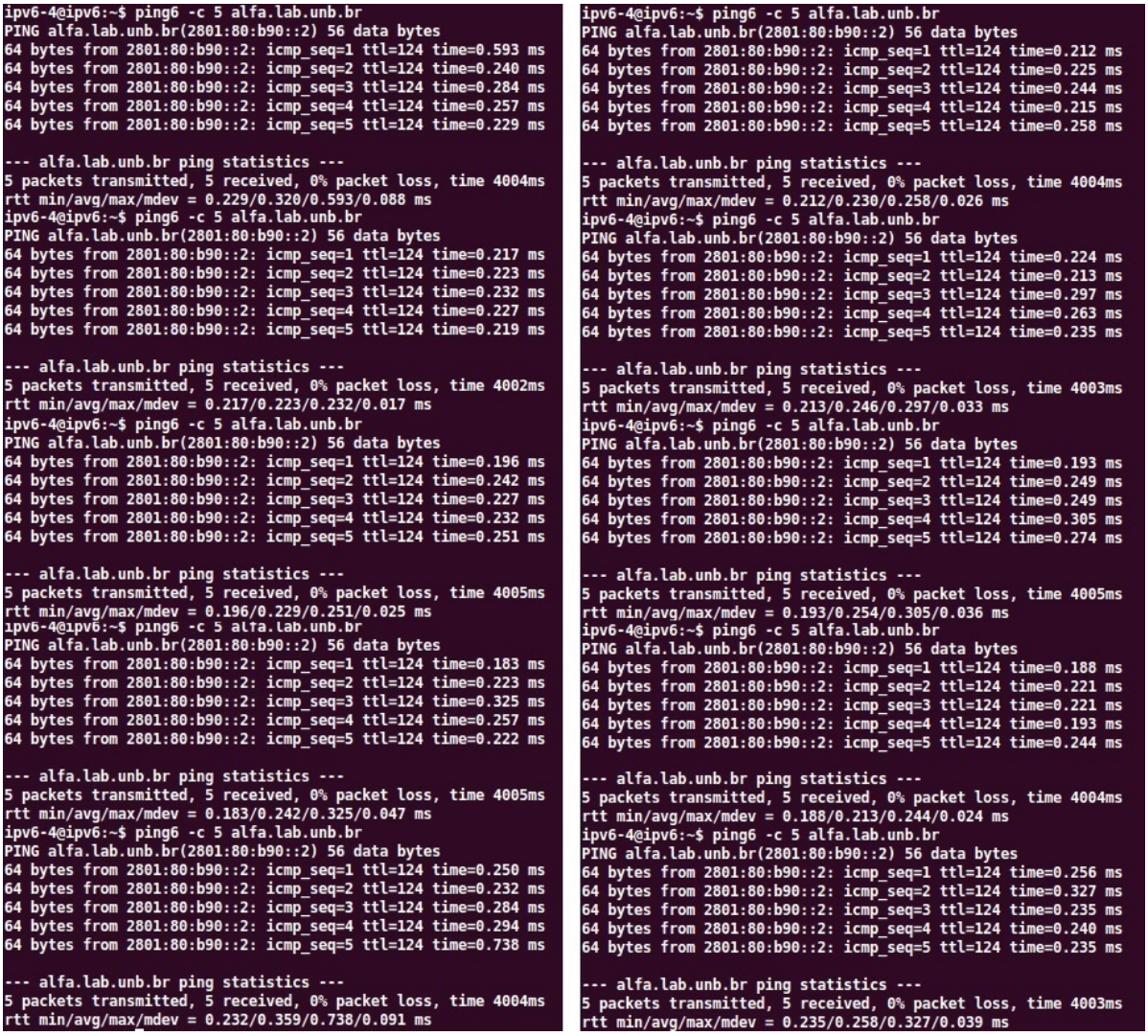


Figura A.4: Grupo de coleta 04 - IPv6

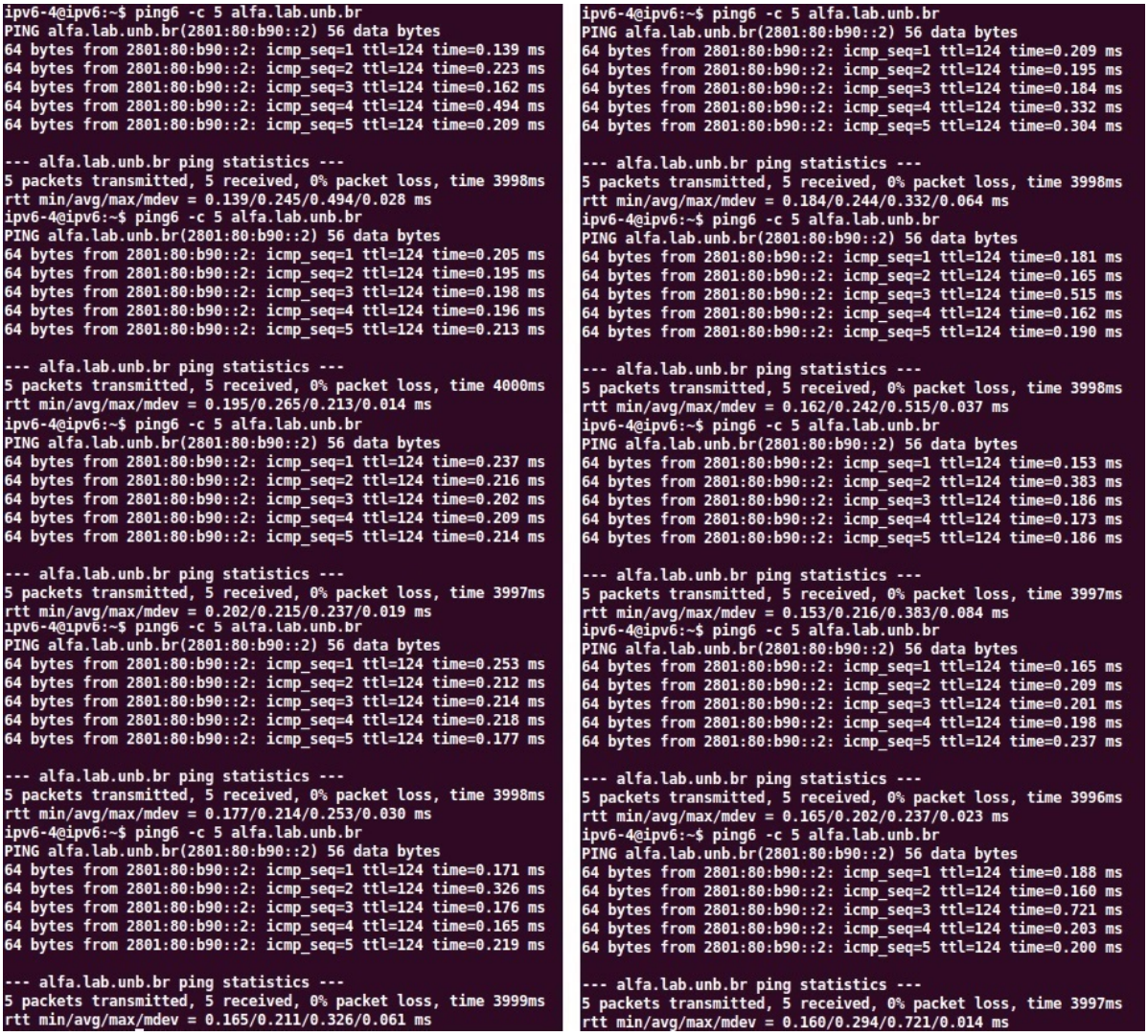


Figura A.5: Grupo de coleta 05 - IPv6

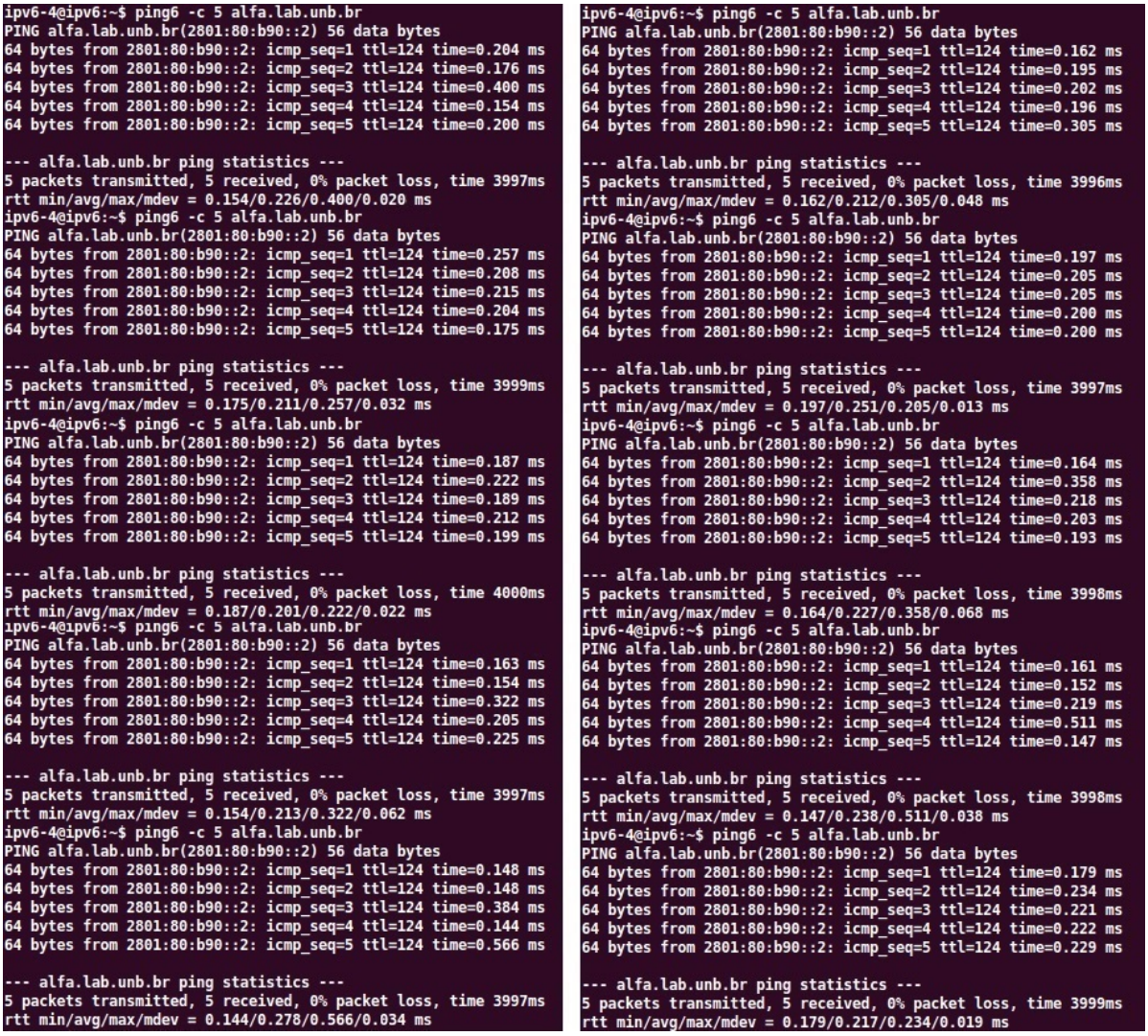


Figura A.6: Grupo de coleta 06 - IPv6

Anexo B

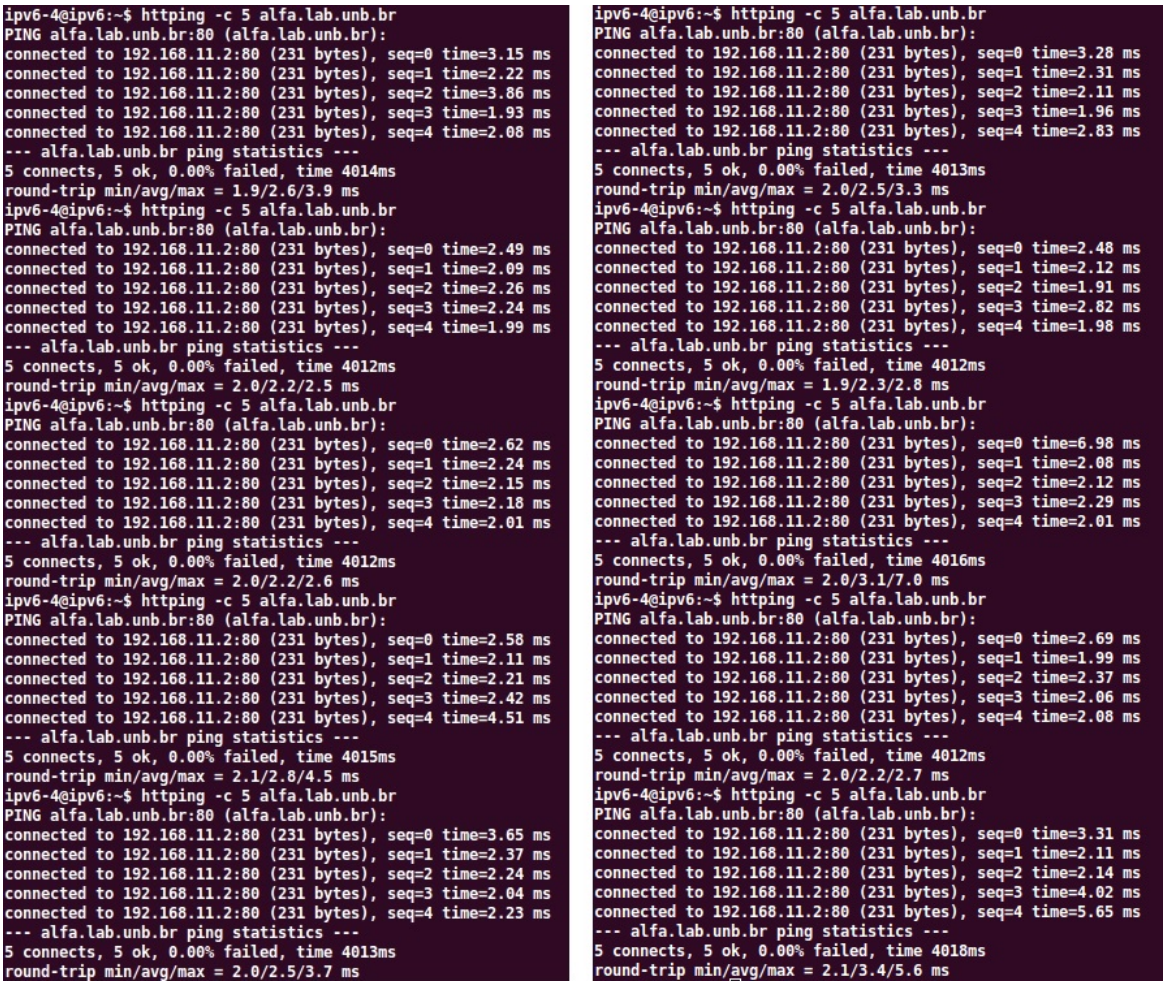


Figura B.1: Grupo de coleta 07 - IPv4

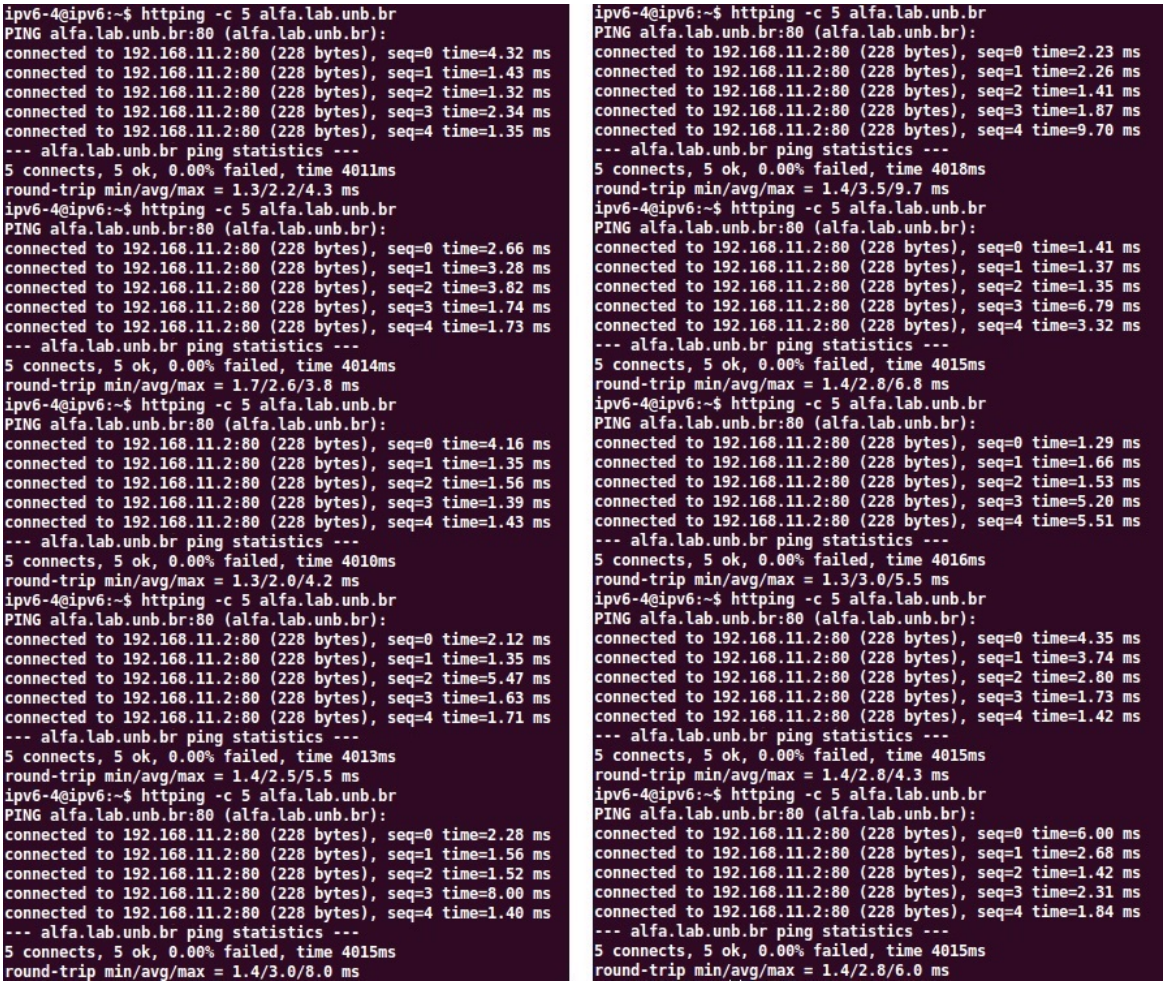


Figura B.2: Grupo de coleta 08 - IPv4

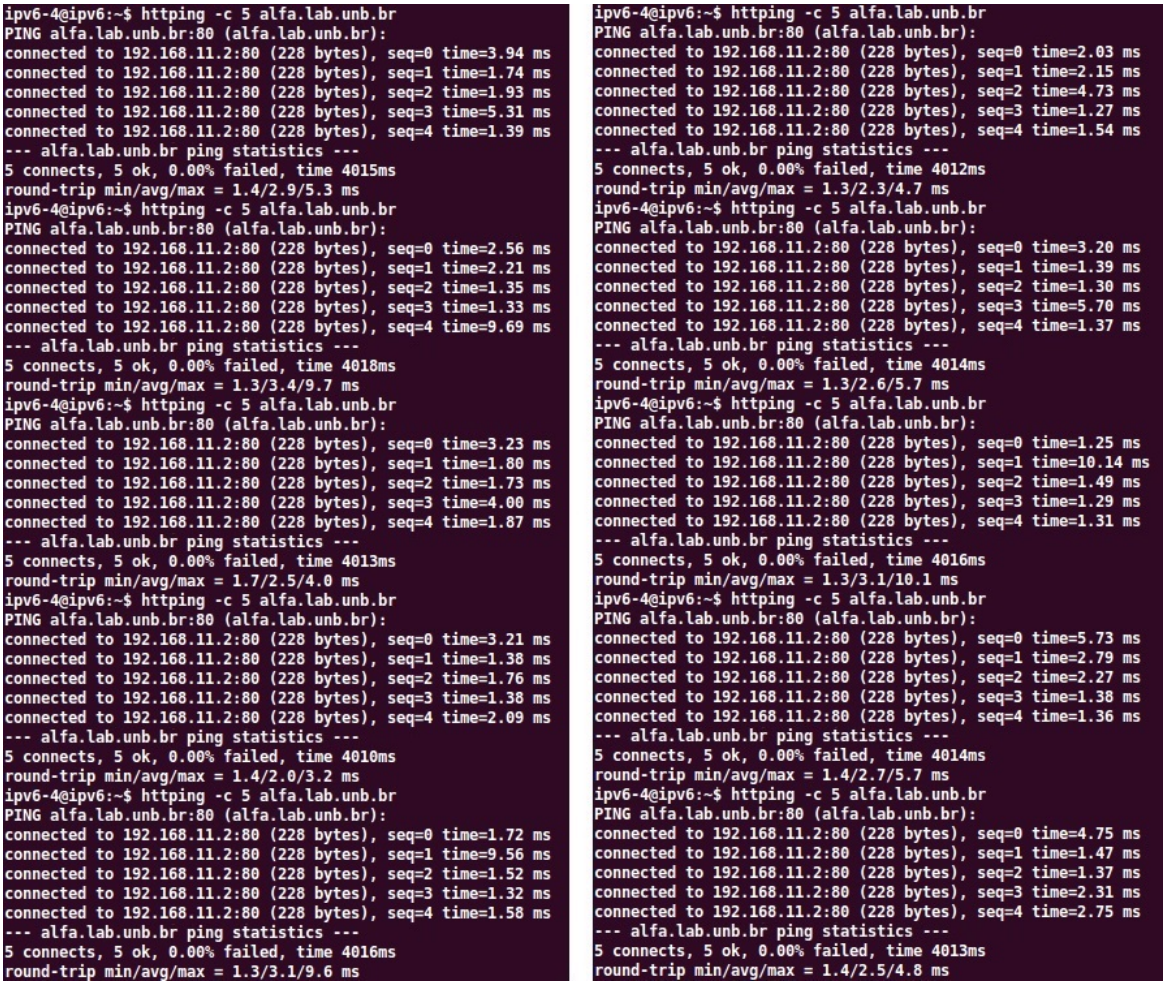


Figura B.3: Grupo de coleta 09 - IPv4

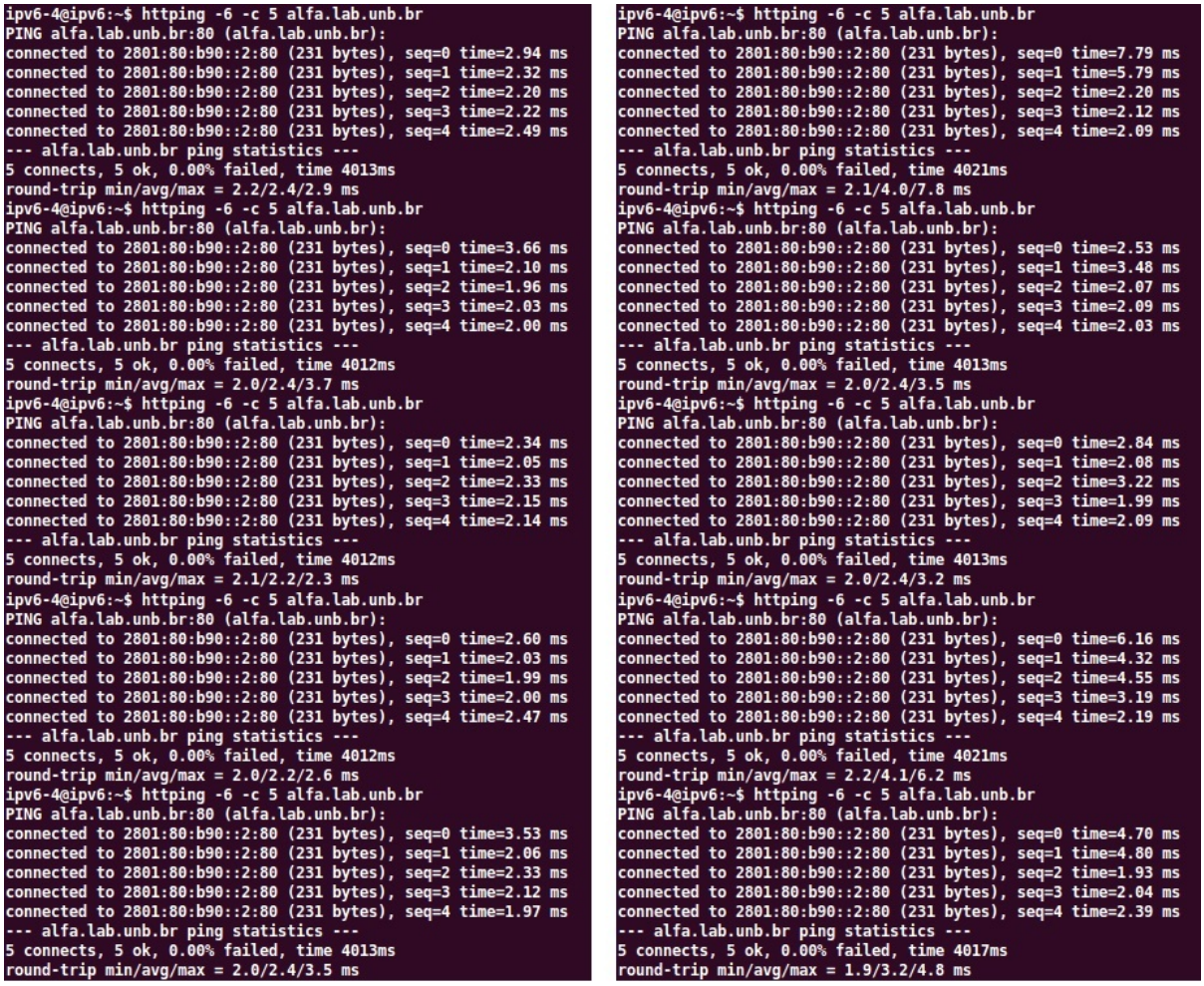


Figura B.4: Grupo de coleta 10 - IPv6

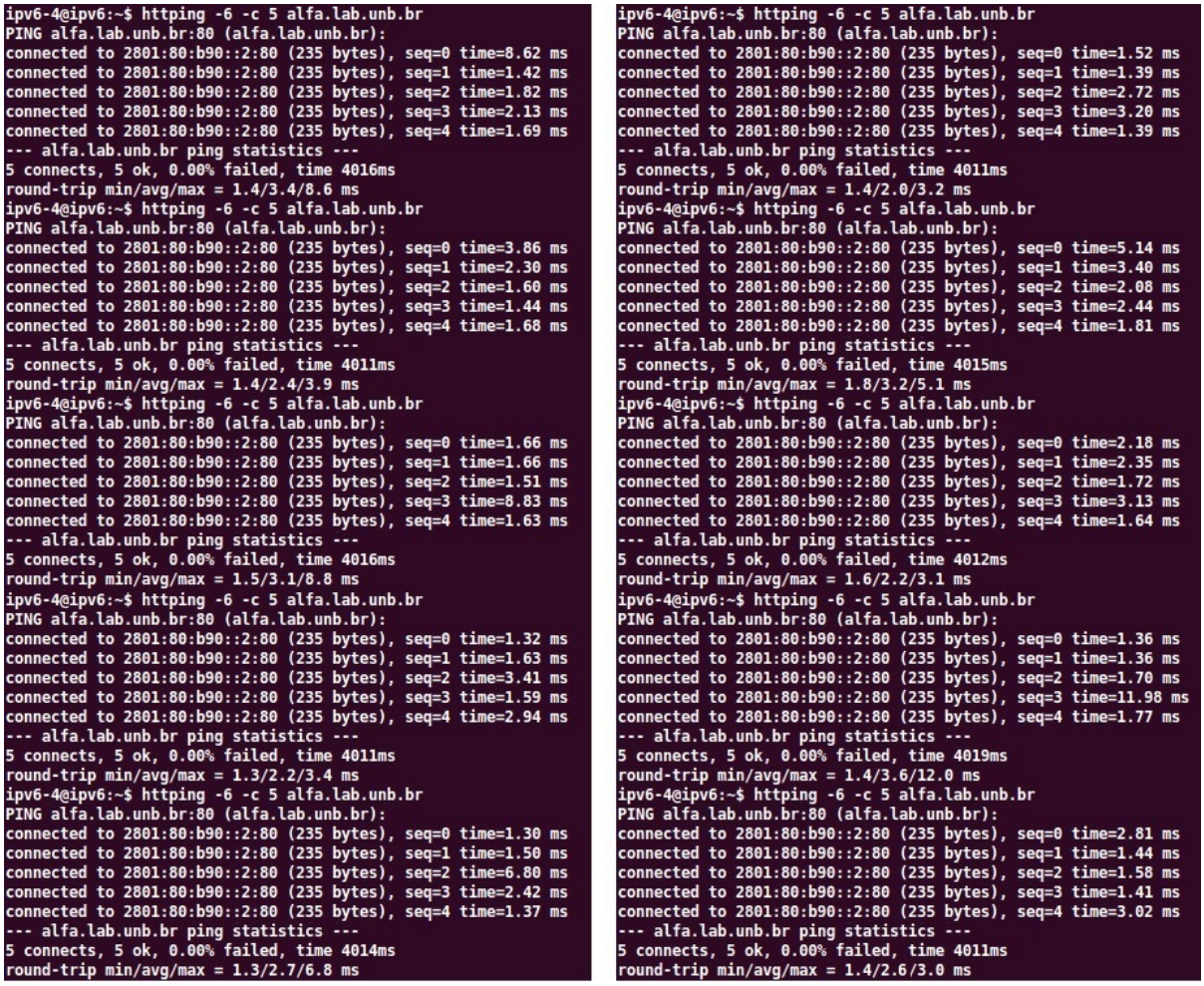


Figura B.5: Grupo de coleta 11 - IPv6

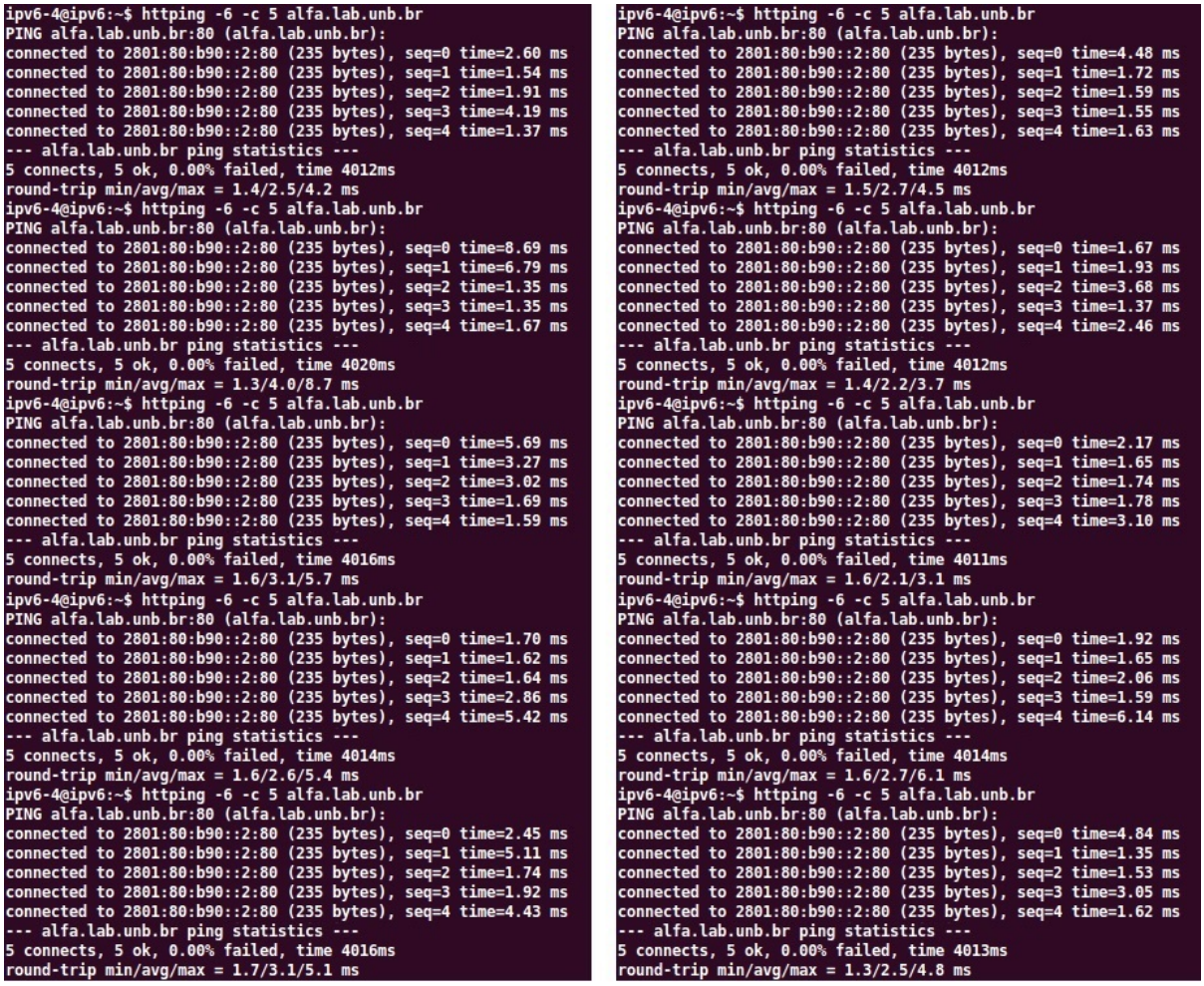


Figura B.6: Grupo de coleta 12 - IPv6

Anexo C

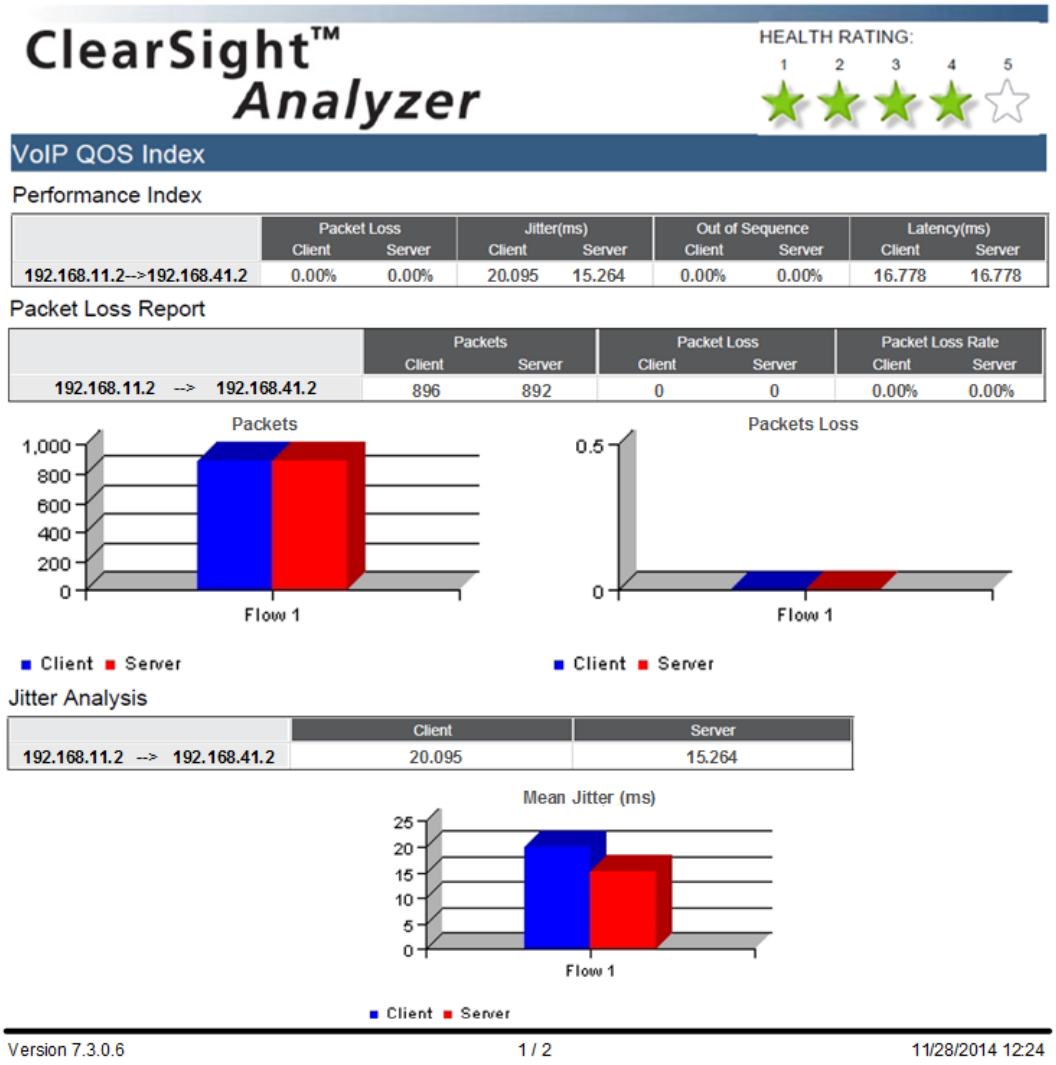


Figura C.1: Relatório VoIP - IPv4 pag 01 de 02

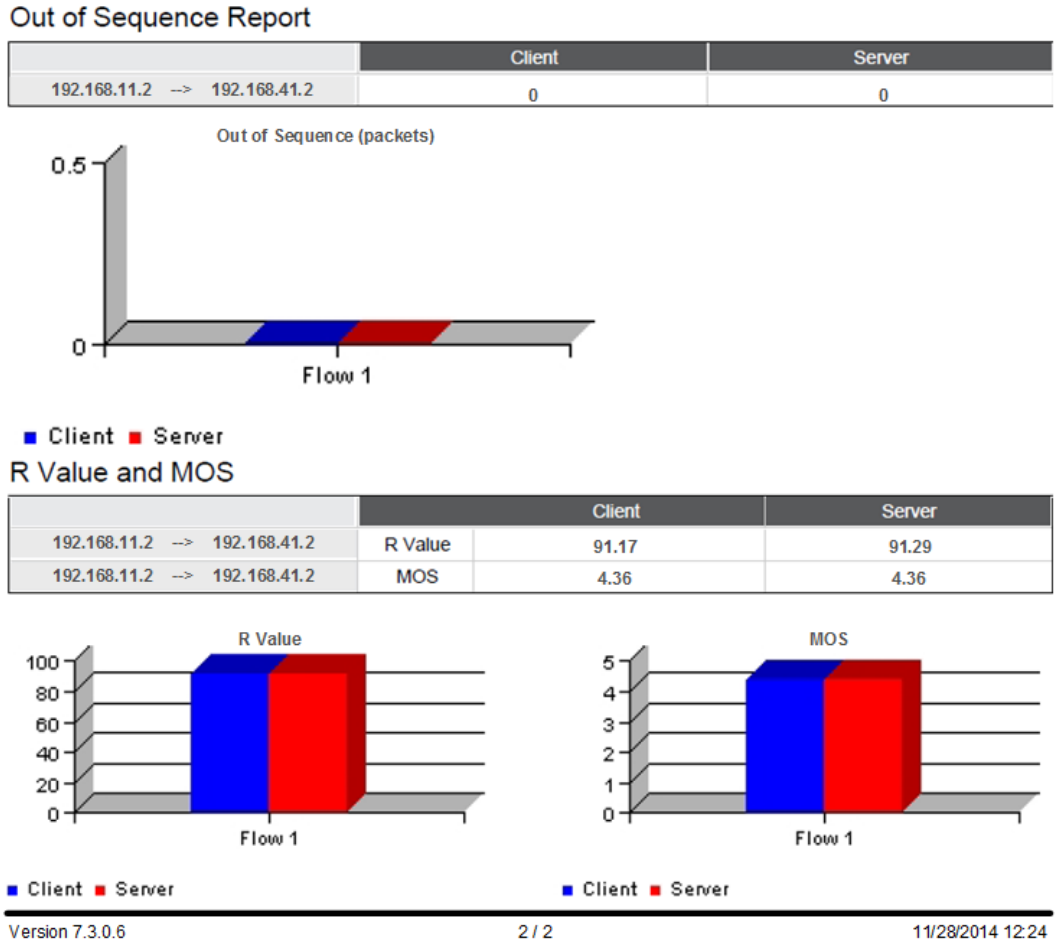


Figura C.2: Relatório VoIP - IPv4 pag 02 de 02

Anexo D

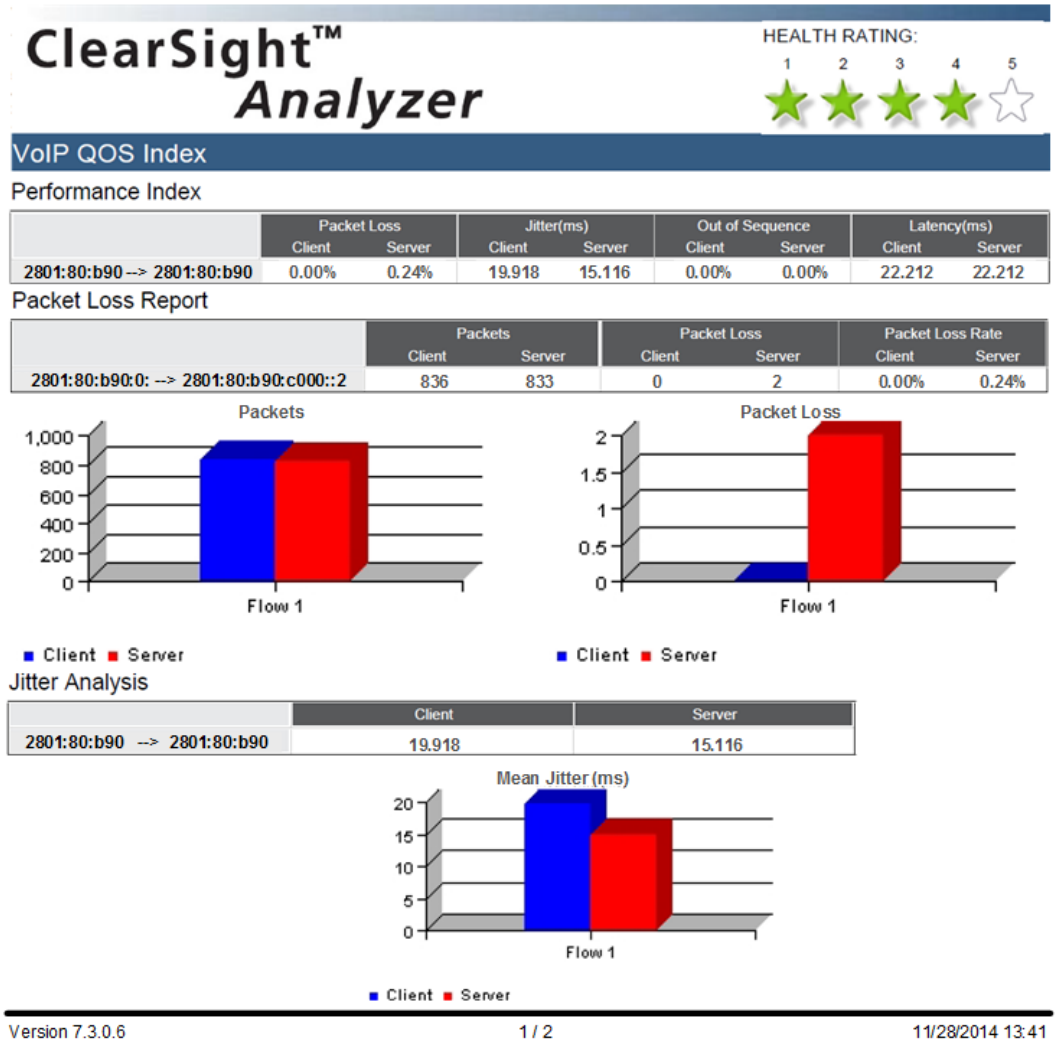


Figura D.1: Relatório VoIP - IPv6 pag 01 de 02

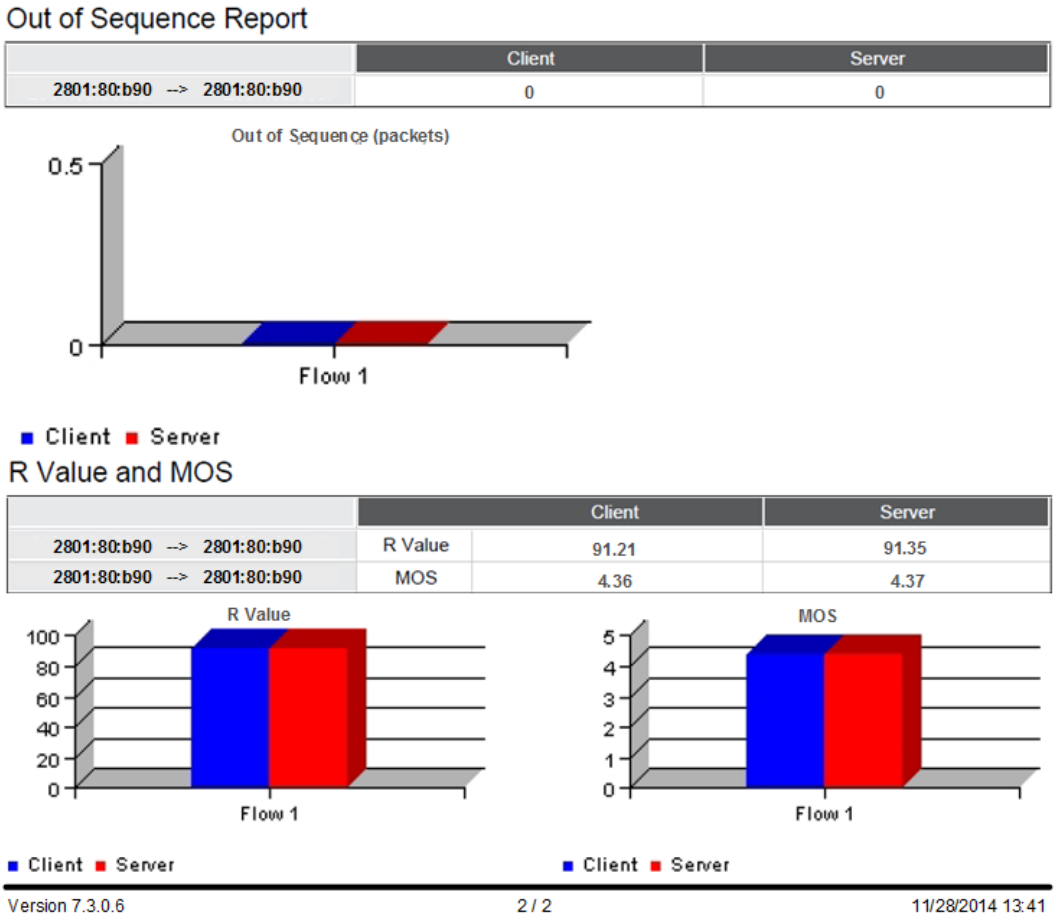


Figura D.2: Relatório VoIP - IPv6 pag 02 de 02