UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM
REDES DE COMPUTADORES

ANÁLISE DOS MÉTODOS DE TRANSIÇÃO PARA O PROTOCOLO IPV6

TRABALHO DE CONCLUSÃO DE CURSO

Letícia da Silva Machado

Santa Maria, RS, Brasil

2015

CSTRC/UFSM, RS	MACHADO, Letícia da Silva	Graduada	2015

ANÁLISE DOS MÉTODOS DE TRANSIÇÃO PARA O PROTOCOLO IPV6

Letícia da Silva Machado

Trabalho de Conclusão de Curso (TCC) apresentado ao Curso Superior de Tecnologia em Redes de Computadores, do Colégio Técnico Industrial de Santa Maria, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de
Tecnólogo em Redes de Computadores.

Orientadora: Prof. Dra. Simone Regina Ceolin

Santa Maria, RS, Brasil

2015

Universidade Federal de Santa Maria
Colégio Técnico Industrial de Santa Maria
Curso Superior de Tecnologia em Redes de Computadores

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Conclusão de Curso

ANÁLISE DOS MÉTODOS DE TRANSIÇÃO PARA O PROTOCOLO IPV6

elaborado por
Letícia da Silva Machado

como requisito parcial para obtenção do grau de
Tecnólogo em Redes de Computadores

COMISSÃO EXAMINADORA:

Simone Regina Ceolin, Dra.
(Presidente/Orientadora)

Renato Preigschadt de Azevedo, Dr.(UFSM)

Tarcisio Ceolin Junior, Ms.(UFSM)

Santa Maria, 02 de julho de 2015.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus, por estar sempre comigo nos meus momentos de alegria e de angústia e por me dar saúde e coragem para buscar meus sonhos.

Aos meus pais, Serlen e Amauri da Silva Machado, que sempre me apoiaram e incentivaram a seguir em frente e não desistir, apesar das inúmeras dificuldades encontradas. Obrigada por todo amor incondicional e pela confiança que depositaram em mim.

Ás minhas irmãs Patrícia e Juliana, que sempre oraram e torceram por mim durante esta minha jornada.

Ao meu namorado José Dirlei de Oliveira, que esteve sempre ao meu lado, me auxiliando quando precisei de ajuda e me reerguendo quando estive desanimada. Obrigada pelo amor, pelo zelo e pela compreensão.

Á minha Orientadora Simone Regina Ceolin, pela confiança, pela ajuda e pelas valiosas ideias e sugestões que me deu durante a realização deste trabalho.

Ao professor Tarcísio Ceolin Junior, por ter se disponibilizado a me ajudar quando precisei.

A todos professores do Curso Superior de Tecnologia em Redes de Computadores, que foram essenciais para minha formação acadêmica, me passando conhecimentos que levarei para vida toda.

Às minhas colegas do curso Técnico em Administração que tanto me auxiliaram durante os períodos de avaliações e me apoiaram incessantemente.

A todos que não foram citados, mas que de uma forma ou de outra, contribuíram para que eu chegasse até aqui.

Obrigada a todos!

RESUMO

Trabalho de Conclusão de Curso
Colégio Técnico Industrial de Santa Maria
Curso Superior de Tecnologia em Redes de Computadores
Universidade Federal de Santa Maria

ANÁLISE DOS MÉTODOS DE TRANSIÇÃO PARA O PROCOTOLO IPV6

AUTORA: LETÍCIA DA SILVA MACHADO
ORIENTADORA: SIMONE REGINA CEOLIN
Data e Local da Defesa: Santa Maria, 02 de julho de 2015.

Com a evolução da Internet e o grande número de dispositivos que hoje em dia podem estar conectados a ela, os endereços IPv4 (*Internet Protocol versão 4*) se tornaram escassos.Por esta razão, uma nova versão foi criada, o IPv6 (*Internet Protocol versão 6*), para suplantar esta exaustão de endereços e também para inserir melhorias na utilização da *Internet* referentes ao desempenho e à qualidade dos serviços. Como estas duas versões do protocolo IP não são diretamente compatíveis, e tendo em vista que este processo de transição pode ser lento e complexo para ser feito imediatamente, torna-se necessário a utilização de técnicas que auxiliem na migração gradual para o novo protocolo. Este trabalho apresenta a implementação dos métodos de Pilha Dupla, tunelamento*IPv6-over-IPv4* e tradução NAT64/DNS64, que são técnicas de transição já existentes, em um cenário composto por máquinas virtuais, para que ao final deste estudo seja possível analisar e definir qual destes métodos é o melhor conforme a situação de cada rede.

Palavras-Chave: IPv6. Transição. Pilha Dupla. Tunelamento. Tradução.

ABSTRACT

Completion of Course Work
Colégio Técnico Industrial de Santa Maria
Superior Course of Technology in Computer Networks
Federal University of Santa Maria

ANALYSIS OF TRANSITION METHODS FOR PROCOTOL IPV6

AUTHOR: LETÍCIA DA SILVA MACHADO
ADVISER: SIMONE REGINA CEOLIN

Date and Place of Defense: Santa Maria, 02 de julho de 2015.

With the evolution of the Internet and the large number of devices nowadays can be connected to it, IPv4 addresses (Internet Protocol version 4) have become scarce. For this reason, a new version was created, the IPv6 (Internet Protocol version 6), to overcome this exhaustion of addresses and also to enter improvements in use of the Internet for the performance and quality of services. As these two IP protocol versions are not directly compatible, and given that this transition process can be slow and complex to be done immediately, it is necessary to use techniques that assist in the gradual migration to the new protocol. This work presents the implementation of dual stack methods, IPv6-over-IPv4 tunneling and NAT64/DNS64 translation, which are transition techniques existing in a scenario composed of virtual machines so that the end of this study it is possible to analyze and define which of these methods is the best according to situation of each network.

Keywords: IPv6. Transition. Dual stack. Tunneling. Translation.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

BGP	- *Border Gateway Protocol*
BIND	- *Berkeley Internet Name Domain*
CEPTRO.br	- Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações
CGI.br	- Comitê Gestor da Internet no Brasil
CIDR	- *Classless Inter-domain Routing*
DHCP	- *Dinamyc Host Configuration Protocol*
DHCPv6	- *Dynamic Host Configuration Protocol for IPv6*
DNS	- *Domain Name System*
DNS64	- *Domain Name System 64*
DOCSIS	- *Data Over Cable Service Interface Specification*
DSR	- Departamento de Infraestrutura de Serviços de Rede
FTP	- *File Transfer Protocol*
HTTP	- *Hypertext Transfer Protocol*
IANA	- *Internet Assigned Numbers Authority*
ICMP	- *Internet Control Message Protocol*
IETF	- *Internet Engineering Task Force*
IP	- *Internet Protocol*
IPng	- *Internet Protocol next generation*
IPv4	- *Internet Protocol version 4*
IPv6	- *Internet protocol version 6*
ISO	- *International Organization for Standardization*
ISP	- *Internet service provider*
LACNIC	- *Latin America and Caribbean Network Information Centre*
LAN	- *Local Area Network*
MPOG	- Ministério do Planejamento, Orçamento e Gestão
NAT	- *Network Address Translation*
NAT64	- *Network Address Translation 64*
NIC.br	- Núcleo de Informação e Coordenação do Ponto BR
OSI	- *Open Systems Interconnection*
OSPF	- *Open Shortest Path First*
RFC	- *Request for Coments*

RIP - *Routing Information Protocol*

RIR - *Regional Internet Registries*

SIPP - *Simple IP Plus*

SLTI - Secretaria de Logística e Tecnologia da Informação

SMTP - *Simple Mail Transfer Protocol*

TCP - *Transmission Control Protocol*

UDP - *User Datagram Protocol*

SUMÁRIO

1 INTRODUÇÃO

A *Internet* é uma das maiores conquistas tecnológicas dos últimos tempos. Através dela, as pessoas podem comunicar-se, trocar informações, ter acesso a inúmeros e variados conteúdos, compartilhar arquivos e recursos entre outras funcionalidades.

Kurose e Ross (2010, p. 2) definem a *Internet* como "uma rede de computadores que interconecta milhares de dispositivos computacionais ao redor do mundo".

Antigamente, nem se imaginava que poderia haver um computador em cada casa, e hoje isto é uma realidade bastante comum.

> Durante as duas primeiras décadas de sua existência, os sistemas computacionais eram altamente centralizados, em geral instalados em uma grande sala, muitas vezes com paredes de vidro, através das quais os visitantes podiam contemplar, embevecidos, aquela grande maravilha eletrônica. (TANENBAUM; WETHERALL, 2011, p.1).

O cenário atual demonstra a enorme evolução da indústria de informática, dado os inúmeros dispositivos que hoje podem estar conectados à *Internet*, como televisores, telefones celulares, automóveis, etc.

Kurose e Ross (2010, p. 2) destacam que "o termo *rede de computadores* está começando a soar um tanto desatualizado, dados os muitos equipamentos não tradicionais que estão sendo ligados à *Internet*".

Com o processo de evolução contínuo da *Internet* e seu rápido crescimento, vieram algumas consequências e uma delas é o esgotamento de endereços *Internet Protocol* (IP). Os endereços IPs são responsáveis por identificar computadores conectados à rede. A versão que vem sendo utilizada desde o final da década de 1970 até atualmente é o *Internet Protocol* versão 4 (IPv4).

> A longevidade da versão 4 mostra que o projeto é flexível e poderoso. Desde o momento em que o IPv4 foi projetado, o desempenho do processador aumentou por três ordens de grandeza, os tamanhos típicos da memória aumentaram por um fator maior que 400, a largura de banda dos enlaces de maior velocidade na *Internet* aumentaram por um fator de 150.000. As tecnologias de LAN emergiram e o número de hosts na *Internet* aumentou de alguns para centenas de milhões. (COMER, 2006, p.370)

Devido este aumento do número de *hosts*, torna-se necessário e inevitável a atualização para a nova versão do *Internet Protocol*, a versão 6, ou IPv6, que traz consigo muitas melhorias, entre elas o maior espaço para endereçamento, aspectos relacionados à qualidade dos serviços oferecidos, ao desempenho e à flexibilidade deste novo protocolo.

1.1 Objetivo Geral

Com a realização deste trabalho, objetiva-se conhecer melhor as características do protocolo IPv6, bem como, compreender as dificuldades encontradas atualmente em relação a sua implantação, para poder executar na prática algumas das técnicas de transição do IPv4 para o IPv6 em diferentes cenários e posteriormente divulgar os resultados obtidos.

1.2 Objetivos Específicos

- ✓ Implementar as técnica de Pilha Dupla (*Dual Stack*), tunelamento*IPv6-over-IPv4* e tradução NAT64/DNS64 em diferentes cenários, constituídos por máquinas virtuais, para análise posterior;
- ✓ Realizar testes em um ambiente virtualizado para que posteriormente seja possível especificar as características de cada técnica de transição escolhida para o trabalho, o modo de operação, como ocorre a implementação;
- ✓ Analisaras vantagens, desvantagens e vulnerabilidades de cada técnica de transição;
- ✓ Analisar a disponibilidade e o desempenho de conexões via IPv6 no cenário atual;
- ✓ Descrever quais técnicas são mais adequadas de acordo com cada cenário.

1.3 Justificativa e Motivação

Devido ao crescimento acelerado e contínuo da *Internet*, torna-se necessário que a migração para o novo protocolo ocorra o mais breve possível, visto que os endereços IPv4 encontram-se escassos e este processo está em atraso.

Segundo um cronograma elaborado pela Equipe IPv6.br (2012), com base no diálogo com provedores *Internet*, operadoras de telecom e provedores de conteúdo, o protocolo IPv6 deveria estar disponível para todos em janeiro de 2014, porém não foi o que ocorreu. A Figura 1 ilustra o cronograma de implantação.

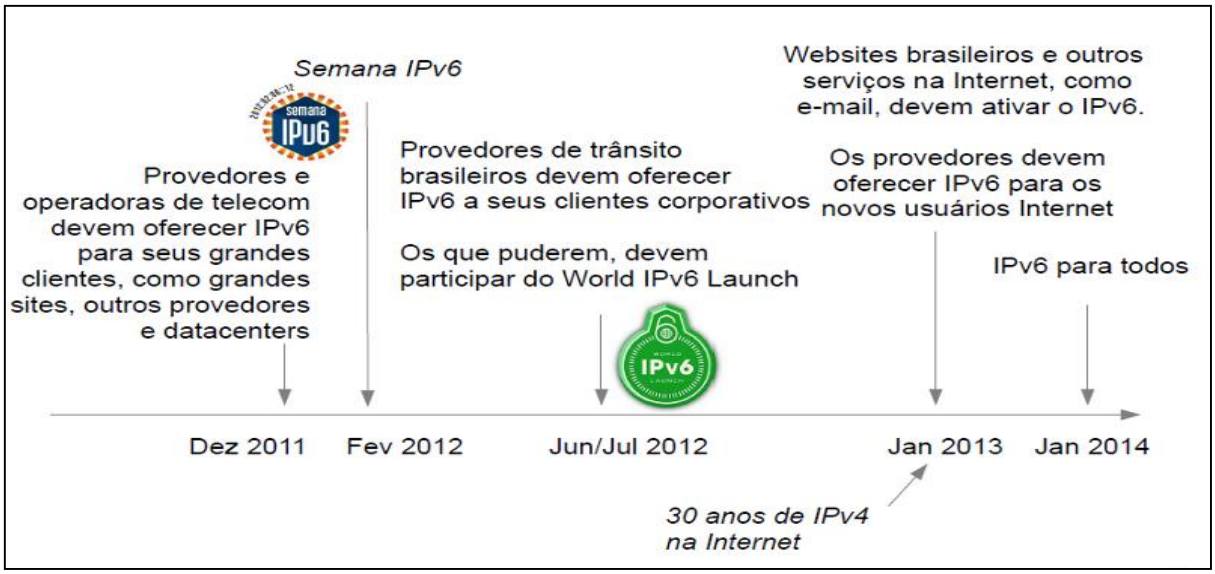


Figura 1: Cronograma de Implantação do IPv6.
Fonte:IPv6.br (http://ipv6.br/cronograma/).

A Equipe IPv6.br (2012) ainda destaca "O cronograma pode ser considerado uma recomendação técnica do NIC.br e um guia, mas não é um documento estático, pode evoluir com o tempo".

Em novembro de 2014, foi instituído um Plano de Disseminação do Uso do IPv6, elaborado pelo Ministério do Planejamento, Orçamento e Gestão (MP), pela Secretaria de Logística e Tecnologia da Informação (SLTI) e pelo Departamento de Infraestrutura de Serviços de Rede (DSR) com a colaboração do Comitê Gestor da *Internet* no Brasil (CGI.br), do Núcleo de Informação e Coordenação do Ponto BR (NIC.br) e do Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações (CEPTRO.br). Neste plano consta um novo cronograma que prevê que a implantação completa seja feita até setembro de 2018.

De acordo com estatísticas do *Latin American and Caribbean Internet Addresses Registry* (LACNIC), no ano de 2014 houve um aumento no número de blocos IPv6 alocados em sua área de cobertura, foram cerca de 1200 blocos, que corresponde a aproximadamente 695 blocos a mais que no ano anterior. Do total de blocos IPv6 alocados pelo LACNIC, cerca de 70% foram para o Brasil. Estes dados podem ser observados nos gráficos das Figuras 2 e 3.

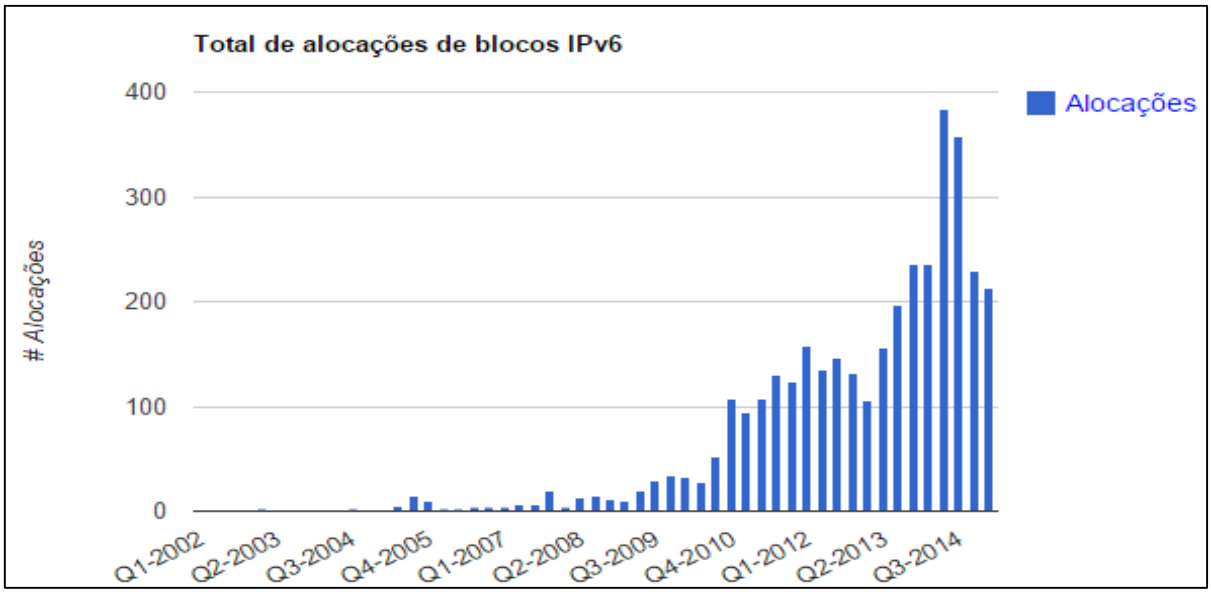


Figura 2: Total de blocos IPv6 alocados pelo LACNIC.
Fonte: LACNIC (http://www.lacnic.net/pt/web/lacnic/estadisticas-asignacion).

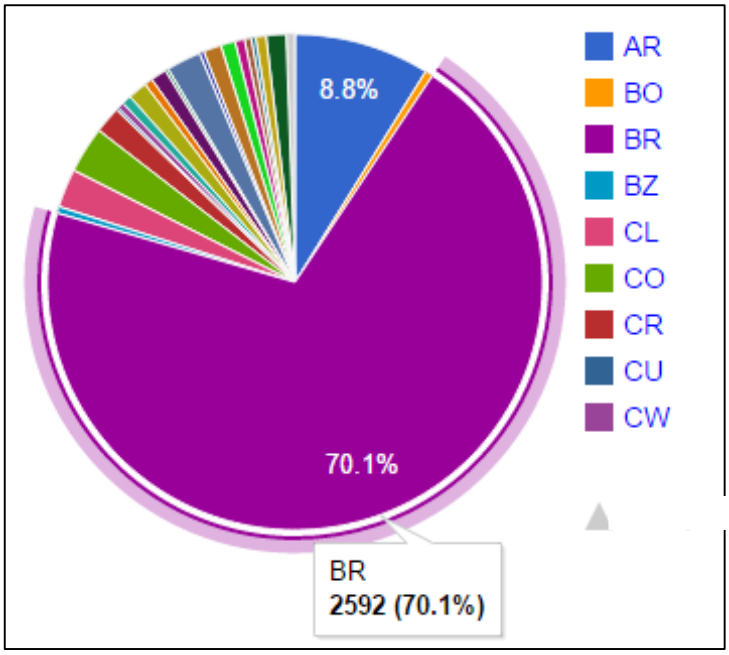


Figura 3: Distribuição de blocos IPv6 na área de cobertura do LACNIC.
Fonte: LACNIC (http://www.lacnic.net/pt/web/lacnic/estadisticas-asignacion).

Em relação a quantidade de usuários brasileiros que utilizam IPv6, segundo dados coletados pela Cisco (2015), nos anos anteriores, o percentual de internautas com IPv6 era bem pequeno e permanecia praticamente constante. Em 1º de janeiro de 2015, eram apenas 0,1% dos usuários que utilizavam IPv6 passando para mais de 2% em junho. O número ainda é bastante pequeno, porém a velocidade desse crescimento é considerável se for comparada com

os outros anos. Segundo Moreiras (2015) “o rápido crescimento tende a continuar e é resultado do trabalho sério dos provedores de acesso à *Internet* para implantação de IPv6 em suas redes”. No gráfico da Figura 4 é possível verificar este crescimento.

Figura 4: Usuários IPv6 no Brasil.
Fonte: Cisco (http://6lab.cisco.com/stats/).

Na Figura 5 está representado o cenário global atual de implantação do IPv6, onde os países representados em tons mais escuros são os que estão mais avançados na implantação do novo protocolo e os em tons mais claros estão menos avançados. Como pode-se notar, a maioria dos países ainda precisam evoluir bastante na implantação do IPv6.

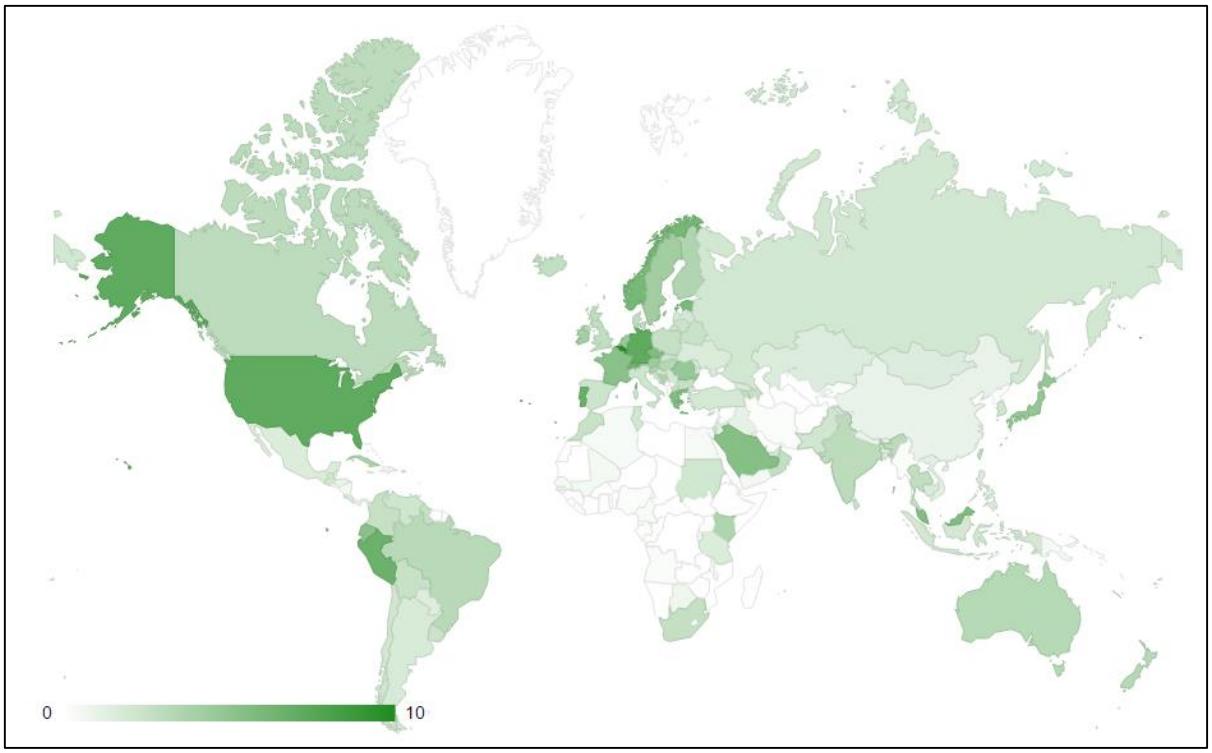


Figura 5: Cenário global de implantação do IPv6.
Fonte: Cisco (http://6lab.cisco.com/stats/).

A situação em que se encontra a *Internet* Global atualmente é o que motiva a realização deste trabalho, que é feito baseado nas técnicas que irão auxiliar na migração para o novo protocolo. Com o estudo e uma análise cuidadosa das técnicas de transição,será possível definir com maior propriedade qual é a mais adequada de acordo com cada rede, para que o funcionamento da mesma não seja prejudicado. Desta forma, a *Internet* poderá continuar crescendo e acomodando quantos usuários forem necessários para que seu processo evolutivo não permaneça estático.

1.4 Estruturação do Trabalho

O presente trabalho estáestruturado da seguinte forma: no Capítulo 2 consta a revisão de literatura, que traz algumas explicações sobre o protocolo IPv4 e as técnicas que ajudaram, temporariamente, a suplantar a escassez de seus endereços, explica também as principais características do protocolo IPv6, as mudanças que serão introduzidas com a sua implantação, a funcionalidade de autoconfiguração de endereços e também as técnicas de transição para o protocolo IPv6. O Capítulo 3 aborda alguns trabalhos já realizados que possuem relação com o

assunto deste. O Capítulo 4 apresenta os passos para a implantação de cada técnica de transição em cenários simulados, com explicações de como foram feitas as configurações. O Capítulo 5 apresenta os testes e os resultados obtidos são. E no Capítulo 6 estão as conclusões que foram obtidas a partir da realização deste trabalho.

2 REVISÃO DE LITERATURA

Neste capítulo serão apresentados alguns conceitos básicos necessários para um melhor entendimento do trabalho proposto. Na Seção 2.1 serão tratados os principais fundamentos a respeito de protocolos, na Seção 2.2 será tratado o protocolo IPv4, as principais motivações para o seu fim e os métodos que foram utilizados para o prolongamento de sua utilização, a Seção 2.3 abordará as mudanças introduzidas pelo IPv6, algumas de suas características e os benefícios de sua implantação, na Seção 2.4 é apresentada a estrutura de governança da *Internet*, a Seção 2.5 apresentará as técnicas de transição para o novo protocolo e na Seção 2.6 há a explicação de como ocorre o processo de autoconfiguração de endereços *stateless* e *stateful*.

2.1 Protocolos

Para início deste trabalho, é fundamental saber o que são protocolos, como são utilizados na comunicação inter-redes, e qual a importância dos mesmos, para assim, dar prosseguimento a este estudo.

Tanenbaum e Wetherall (2011, p. 25) definem protocolo como "um conjunto de regras que controla o formato e o significado dos pacotes ou mensagens que são trocadas pelas entidades pares contidas em uma camada".

Existem diversos tipos de protocolos que realizam diferentes tarefas de comunicação. É essencial que estes funcionem bem em conjunto para que a conexão seja estabelecida eficientemente, e assim, ocorra a comunicação.

Segundo Comer (2007, p. 244) para garantir que os protocolos funcionem bem juntos é feito um plano de projeto global, ou seja, os protocolos são projetados e desenvolvidos em conjuntos completos e cooperativos chamados simplesmente de conjuntos ou famílias, desta forma, cada protocolo em um conjunto resolve uma parte do problema de comunicação e juntos resolvem o problema de comunicação por completo.

Para planejar o conjunto de protocolos para resolver o problema de comunicação, os projetistas baseiam-se em um modelo de camadas (*layering model*).

A International Organization for Standardization (ISO), que como o nome já diz, é uma organização que desenvolve normas para padronização internacional, definiu um dos modelos

existentes, que se chama *Open Systems Interconnection* (OSI) *Reference Model*, ou simplesmente, Modelo OSI. Este modelo possui sete camadas: (1) Física, (2) Enlace, (3) Rede, (4) Transporte, (5) Sessão, (6) Apresentação e (7) Aplicação.

Existe também o Modelo de Referência TCP/IP, que segundo Tanenbaum e Wetherall (2011, p.28) "foi definido pela primeira vez em Cerf e Khan (1974), depois melhorado e definido como um padrão na comunidade da Internet (Branden, 1989)". Este modelo conta com quatro camadas: (1) Camada de Enlace, (2) Internet, (3) Transporte e (4) Aplicação.

Os modelos OSI e TCP/IP podem ser observados na Figura 6.

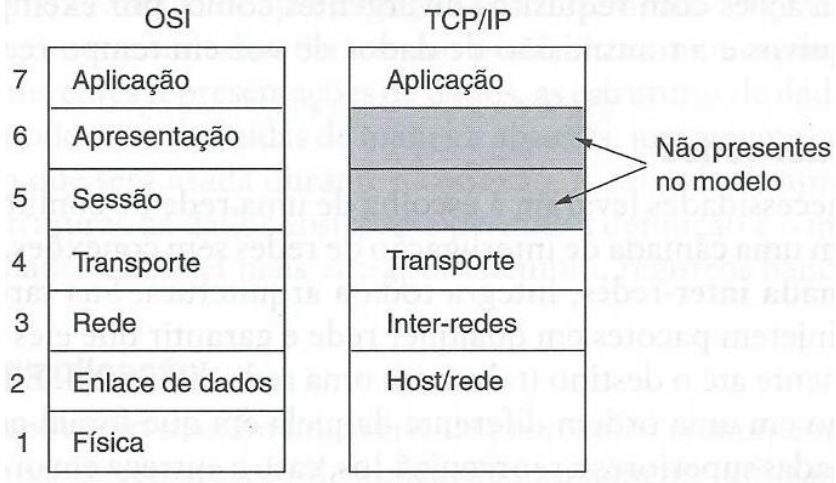


Figura 6: Modelo de referência OSI e TCP/IP.
Fonte:Tanenbaum (2003, p. 46).

Pode-se também observar que alguns autores, como Tanenbaum e Wetherall, costumam utilizar um modelo híbrido (Figura 7), que é composto por cinco camadas: (1) Física, (2) Enlace, (3) Rede, (4) Transporte e (5) Aplicação. Kurose e Ross chamam este modelo de Pilha de protocolos da *Internet.*

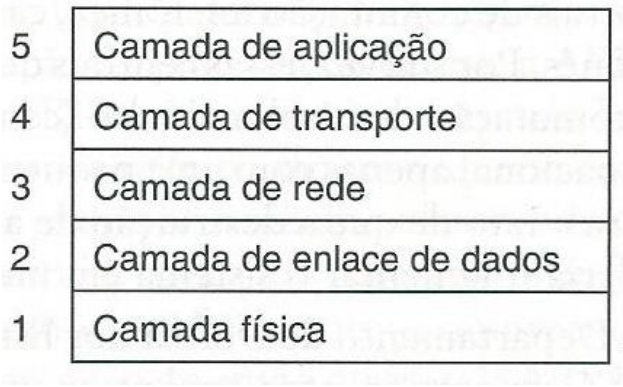


Figura 7: Modelo Híbrido.
Fonte: Tanenbaum (2003, p. 53).

Cada protocolo pertence a uma destas camadas. Cada camada possui diferentes funções e utiliza os serviços da camada abaixo dela para prover serviços à camada acima dela.

A descrição de cada camada é feita segundo Kurose e Ross (2014):

- Camada de Aplicação: é onde residem aplicações de rede e seus protocolos. A camada de aplicação da Internet inclui muitos protocolos, tais como o *Hypertext Transfer Protocol* (HTTP), o *Simple Mail Transfer Protocol* (SMTP) e o *File Transfer Protocol* (FTP).
- Camada de Transporte: a camada de transporte da *Internet* carrega mensagens da camada de aplicação entre os lados do cliente e servidor de uma aplicação. Há dois protocolos de transporte na Internet: o *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP), e qualquer um pode levar mensagens da camada de aplicação.
- Camada de Rede: a camada de rede da *Internet* é responsável pela movimentação, de um hospedeiro para outro, de pacotes da camada de rede, conhecidos como datagramas. Esta camada provê o serviço de entrega do segmento à camada de transporte no hospedeiro de destino. Nesta camada está o *Internet Protocol* (IP) e também os protocolos de roteamento, tais como o *Open Shortest Path First* (OSPF), o *Routing Information Protocol* (RIP) e o *Border Gateway Protocol* (BGP) entre outros. Para Kurose e Ross (2014) apesar de a camada de rede conter o protocolo IP e também numerosos outros de roteamento, o IP é o elemento fundamental que mantém a integridade da *Internet.*
- Camada de Enlace: Para levar um pacote de um nó ao nó seguinte na rota, a camada de rede depende dos serviços da camada de enlace. Os serviços prestados pela camada de enlace dependem do protocolo específico empregado no enlace. Por exemplo, alguns destes protocolos proveem entrega garantida entre enlaces, isto é, desde o nó transmissor, passando por um único enlace, até o nó receptor. Exemplos de protocolos da camada de enlace são

Ethernet, Wi-fi e o protocolo *Data Over Cable Service Interface Specification* (DOCSIS) da rede de acesso por cabo.

- Camada Física: Enquanto a tarefa da camada de enlace é movimentar quadros inteiros de um elemento da rede até um elemento adjacente, a da camada física é movimentar os *bits* individuais que estão dentro de um quadro de um nó para o seguinte. Os protocolos nessa camada dependem do enlace e, além disso, do próprio meio de transmissão do enlace (por exemplo, fios de cobre trançado ou fibra ótica monomodal). Por exemplo a Ethernet tem muitos protocolos de camada física: um para fios de cobre trançado, outro para cabo coaxial, mais um para fibra e assim por diante.

2.1.1 Protocolos da camada de rede

Neste trabalho será dado foco à camada de rede. Segundo Kurose e Ross (2014) a camada de rede envolve todos os roteadores da rede e por isso os protocolos da camada de rede estão entre os mais desafiadores a pilha de protocolos. Para eles, existem três componentes mais importantes nesta camada que são: o protocolo IP, os protocolos de roteamento, e o protocolo de comunicação de erro e de informações da *Internet*.

A Figura 8 contempla os principais componentes desta camada.

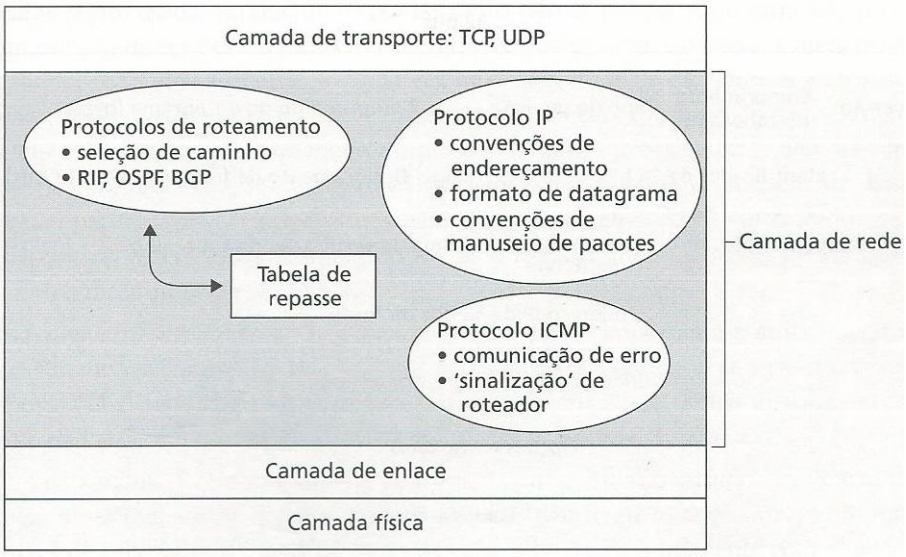


Figura 8: Protocolos da camada de rede da *Internet.*
Fonte: Kurose; Ross (2014, p. 245)

Segundo Tanenbaum (2003) a camada de rede controla a operação da sub-rede e uma questão fundamental de projeto é determinar a maneira como os pacotes são roteados da origem até o destino. Se houver muitos pacotes na sub-rede ao mesmo tempo, eles dividirão o mesmo caminho, provocando gargalos. O controle desse congestionamento também pertence à camada de rede. De modo mais geral, a qualidade do serviço fornecido também é uma questão da camada de rede. Quando um pacote tem de viajar de uma rede para outra até chegar a seu destino, podem surgir muitos problemas. O endereçamento utilizado pela segunda rede pode ser diferente do que é empregado pela primeira rede, os protocolos podem ser diferentes e a comunicação pode não ser estabelecida. Cabe a camada de rede superar todos esse problemas, a fim de permitir que redes heterogêneas sejam interconectadas.

2.2 *Internet Protocol* versão 4

As especificações do *Internet Protocol* versão 4 (IPv4), são tratadas na RFC 791 de 1981.Segundo consta no registro "*Version Numbers"* disponível no site da IANA, os números de versões 0 e 1 foram reservados e os números 2 e 3 não foram atribuídos.

De acordo com Comer (2007) o sucesso do IPv4 é incrível, pois acomodou mudanças em tecnologias de *hardware*, em redes heterogêneas e em escala extremamente grande. No entanto, a motivação primária para migrar para o IPv6, deve-se ao fato do espaço de endereçamento limitado, quando o IP foi criado existiam somente algumas redes de computadores e os 32 bits para endereçamento, que equivalem a mais de 4 bilhões de endereços distintos, eram suficientes. Porém, a *Internet* Global está crescendo exponencialmente e se não houver uma mudança no espaço de endereçamento, nenhum crescimento adicional será possível.

2.2.1 Mecanismos que prolongaram o ciclo de vida do IPv4

A escassez de endereços IPv4 foi prevista há muitos anos, porém com a ajuda de técnicas paliativas, entre elas o *Classless Inter-Domain Routing* (CIDR), o *Network Address Translation*

(NAT) e o *Dynamic Host Configuration Protocol* (DHCP), foi possível contornar este problema temporariamente.

- *Classless Inter-DomainRouting* (CIDR)

Inicialmente, o endereçamento IPv4 era feito por classes. Cada classe possuía um número fixo de blocos, de tamanhos fixos. A Tabela 1mostra o número e o tamanho dos blocos de cada classe de endereço.

Tabela 1: Número de blocos e tamanho dos blocos no endereçamento IPv4 com classes.

Classe	***Número de Blocos***	***Tamanho do Bloco***	***Aplicação***
A	128	16.777.216	*Unicast*
B	16.384	65.536	*Unicast*
C	2.097.152	256	*Unicast*
D	1	268.435.456	*Multicast*
E	1	268.435.456	Reservado

Fonte: Forouzan B.A. (2007, p.553).

Porém, segundo Forouzan (2007) um bloco de classe A é muito grande para praticamente qualquer organização, por isso a maioria dos endereços classe A era desperdiçada. Um bloco de classe B, provavelmente era muito grande para muitas das organizações que recebiam um destes blocos. Os de classe C, por sua vez, eram muito pequenos para muitas organizações.

A ineficiência desta divisão, motivou a criação de uma solução chamada *Classless Inter-Domain Routing* (CIDR), descrito inicialmente na RFC 1519de 1993, para suplantar o esgotamento de endereços e oferecer acesso à *Internet* a um número maior de organizações.

Segundo Tanenbaum (2003), a ideia básica por trás deste método, é alocar os endereços IP restantes em blocos de tamanho variável, sem levar em consideração as classes. Portanto, se um site precisar de 2.000 endereços, ele receberá um bloco de 2.048 endereços.

- *Network Address Translation* (NAT)

O NAT é especificado na RFC 1631 (1994)e pode ser definido basicamente como um protocolo que permite que vários computadores de uma rede interna estejam conectados à *Internet* através de um único endereço válido ou de um pequeno conjunto destes, ou seja, quando um host encaminha um pacote para a *Internet*, este passa por um roteador que possui NAT que substitui o endereço de origem por um endereço global. Quando o pacote retorna para

a rede interna ele passa novamente pelo roteador NAT, que substitui o endereço de destino que está no pacote pelo endereço privado original. Para saber de onde veio o pacote e para onde deve retornar, é mantida uma tabela de tradução no roteador NAT que contém o endereço os números de portas, o IP do *host* na *Internet* e o endereço IP interno do mesmo, com isso é possível fazer o mapeamento dos endereços. A Figura 9 exemplifica o processo de tradução com NAT.

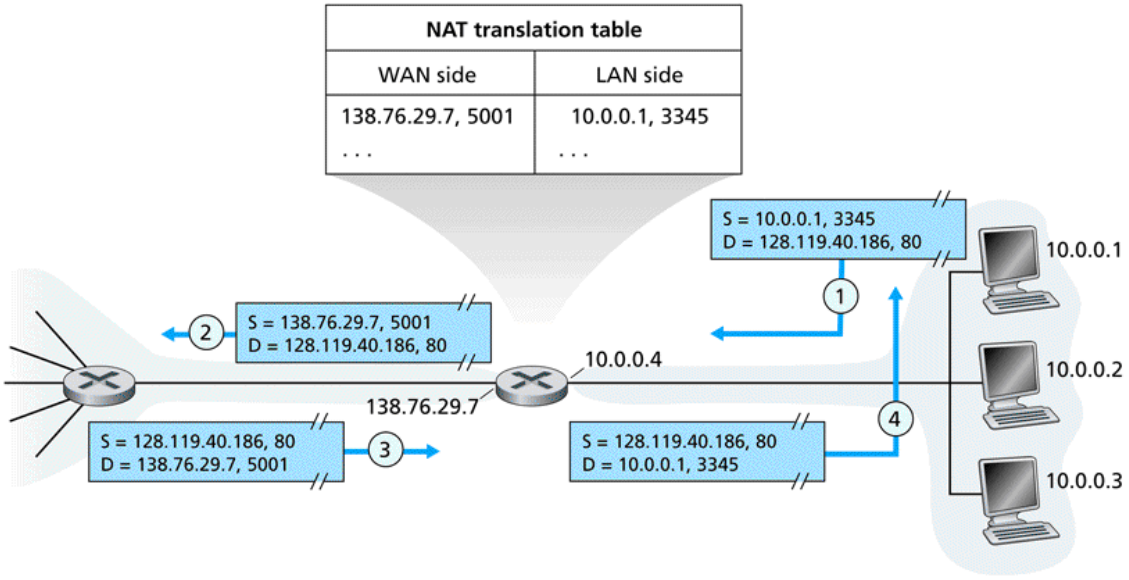


Figura 9: Tradução de endereços com NAT.
Fonte: Kurose e Ross (2010, p.261).

Segundo a RFC 1631(1994), esta solução quebra o princípio fim-a-fim de um endereço IP, além de prejudicar a segurança, escondendo a identidade de *hosts*, e o desempenho, pois aumenta o processamento nos dispositivos tradutores.

- *Dynamic Host Configuration Protocol* (DHCP)

O DHCP permite que um hospedeiro obtenha (ser alocado) um endereço IP automaticamente. Um administrador de rede pode configurar o DHCP de modo que um determinado hospedeiro receba o mesmo endereço IP toda vez que se conectar à rede, ou um hospedeiro pode receber um endereço IP temporário diferente sempre que se conectar à rede. Além de receber um endereço IP temporário, o DHCP também permite que o hospedeiro descubra informações adicionais, como a máscara de sub-rede, o endereço do primeiro roteador (comumente chamado de porta de comunicação padrão- *default gateway*) e o endereço de seu servidor DNS local.(KUROSE;ROSS, 2010, p. 257)

Quando o hospedeiro entra ou sai da rede, uma lista de endereços disponíveis, mantida pelo servidor DHCP, é atualizada. O processo para obtenção de endereços IP via DHCP é constituído por quatro etapas, conforme mostra a Figura 10.

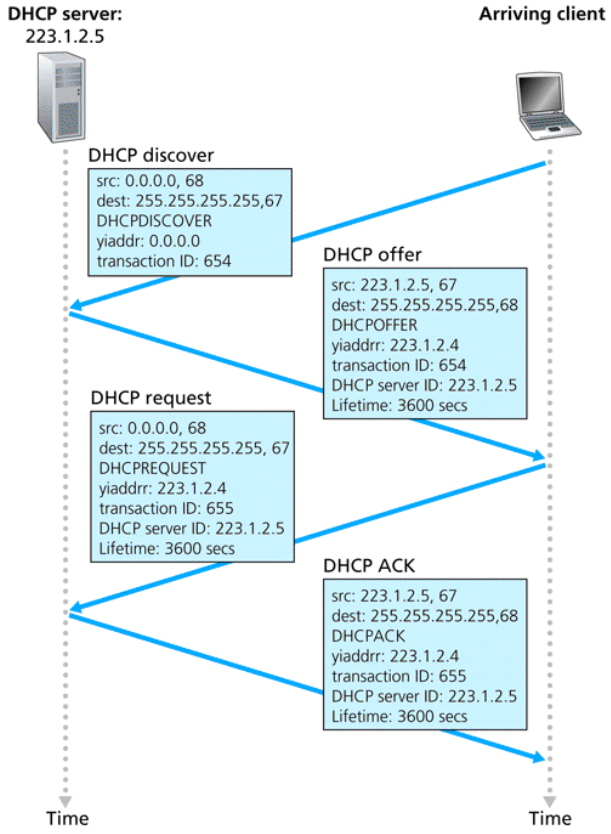


Figura 10: Interação cliente-servidor DHCP.
Fonte: Kurose; Ross (2010, p. 259).

Segundo a Equipe IPv6.br (2012), apesar das medidas citadas serem alternativas que temporariamente solucionaram a escassez de endereços IP e sustentaram a existência do protocolo IPv4, elas não foram suficientes para resolver os problemas decorrentes do crescimento da *Internet.* A adoção dessas técnicas reduziu em apenas 14% a quantidade de blocos de endereços solicitados à IANA e a curva de crescimento da *Internet* continuava apresentando um aumento exponencial. Estas medidas serviram para que houvesse mais tempo para se desenvolver uma nova versão do IP, que fosse baseada nos princípios que fizeram o sucesso do IPv4, porém, que fosse capaz de suprir as falhas apresentadas por ele, sendo então criado o IPv6.

2.2.2 Datagrama IPv4

Segundo Tanenbaum e Wetherall (2011) um datagrama IPv4 é constituído por uma parte de cabeçalho e uma parte de dados. O cabeçalho tem uma parte fixa de 20 *bytes* e uma parte opcional de tamanho variável. A Figura 11 ilustra o formato do datagrama IPv4.

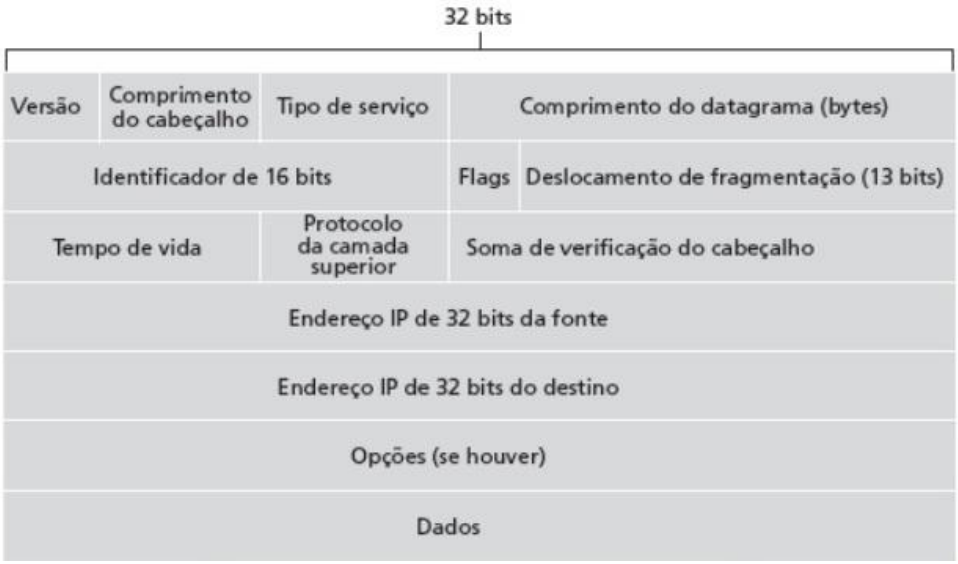


Figura11: Formato do datagrama IPV4.
Fonte: Kurose; Ross (2010, p. 248).

A descrição de cada um dos campos do datagrama IPv4 é feita segundo Kurose e Ross (2010):

- Versão: Quatro bits que especificam a versão do protocolo IP do datagrama.
- Comprimento do cabeçalho:Quatro bits que são necessários para determinar onde no datagrama IP, os dados realmente começam.
- Tipo de Serviço: Oito bits usados para diferenciar os diferentes tipos de datagramas IP que devem ser distinguidos uns dos outros.
- Comprimento do datagrama: É o comprimento total do datagrama IP (cabeçalho mais dados) medido em *bytes.* Possui 16 *bits*.
- Identificador, *Flags*, Deslocamento de fragmentação: Esses três campos têm a ver com a fragmentação do IP.

- Tempo de Vida: Serve para garantir que datagramas não fiquem circulando para sempre na rede. É decrementado em uma unidade cada vez que o datagrama é processado por um roteador. Se o campo chegar a zero, o mesmo deve ser descartado.
- Protocolo: é usado somente quando um datagrama IP chega ao seu destino final. Indica o protocolo da camada de transporte específico ao qual a porção de dados desse datagrama IP deverá ser passada.
- Soma de verificação do cabeçalho: a soma de verificação do cabeçalho auxilia um roteador na detecção de erros de bits em um datagrama IP recebido.
- Endereços IP de fonte e de destino: Endereço IP da fonte e endereço IP do destino final respectivamente.
- Opções: permite que um cabeçalho seja ampliado.
- Dados (carga útil): O campo de dados do datagrama IP contém o segmento da camada de transporte (TCP ou UDP) a ser entregue ao destino. Também pode carregar outros tipos de dados, como mensagens ICMP.

2.2.3 Endereçamento IPv4

Um endereço IPv4 é composto por 32 *bits*. Esses *bits* são segmentados em quatro campos de 8 *bits* que são expressados no formato decimal, de 0 a 255, separados por ".". Exemplo:

11000000 10101000 00000011 00011000 = 192.168.3.24

Com 32 *bits* para endereçamento pode-se obter 4.294.967.296 (2^{32}) endereços distintos. Quando o IPv4 foi criado, esta quantidade era suficiente, porém com o crescimento rápido e contínuo da *Internet* eles se tornaram escassos, surgindo a necessidade de um maior espaço de endereçamento, que é uma das melhorias proporcionados pelo IPv6.

2.3 *Internet Protocol* versão 6

As especificações *do Internet Protocol* versão 6(IPv6), ou também chamado de *IP Next Generation* (IPng), são tratadas na RFC 2460 de 1998.

Este protocolo foi formulado pelo *Internet Engineering Task Force*(IETF)que é uma comunidade internacional aberta, composta por diversos membros que possuem interesse em contribuir com a evolução e o funcionamento da *Internet.*

Para a formulação desta nova versão do protocolo IP, várias pessoas que possuíam interesse na evolução do protocolo, uniram esforços e fizeram inúmeras propostas e sugestões para a nova versão que seria a sucessora do protocolo IPv4.

Vários projetos foram propostos, porém um deles conhecido como *Simple IP Plus* (SIPP), que contava com ideias de outros projetos, foi a base fundamental para a criação do novo IP.

A IETF, então, decidiu atribuir a esta versão o número 6, pois segundo Comer (2006) o número de versão 5 foi pulado depois de uma série de erros e mal entendidos, por isso a escolha de numerar a nova versão com o número 6 serviu para eliminar a confusão e a ambiguidade.

Comer (2006) agrupou as mudanças introduzidas pelo IPv6 em sete categorias:

- *Endereços maiores.* O novo tamanho do endereço é a mudança mais observável. O IPv6 quadriplica o tamanho de um endereço IPv4 de 32 *bits* para 128 *bits*.
- *Hierarquia de endereço estendida.* O IPv6 usa o espaço de endereço maior para criar níveis adicionais de hierarquia de endereçamento (por exemplo, para permitir que um ISP aloque blocos de endereços a cada cliente).
- *Formato de cabeçalho flexível.* O IPv6 usa um formato de datagrama completamente novo e incompatível, que inclui um conjunto de cabeçalhos opcionais.
- *Opções avançadas.* O IPv6 permite que um datagrama inclua informações de controle opcionais; as opções do IPv6 fornecem facilidades adicionais não disponível no IPv4.
- *Provisão para extensão de protocolo.* Em vez de especificar todos os detalhes, a capacidade de extensão do IPv6 permite que o IETF adapte o protocolo ao novo *hardware* de rede e novas aplicações.
- *Suporte para autoconfiguração e renumeração.* O IPv6 permite que os computadores em uma rede isolada atribuam endereços locais automaticamente; o projeto também permite que um gerente renumere redes em um site dinamicamente.

- *Suporte para alocação de recursos.* O IPv6 inclui uma abstração de fluxos e bits para a especificação de serviço diferenciado (*Diff Serv)*. O último é idêntico ao *Diff Serv* do IPv4.

2.3.1 Datagrama IPv6

O datagrama IPv6 trouxe consigo uma estrutura mais simples que a do datagrama IPv4 e alguns aprimoramentos. Ele é constituído por um cabeçalho básico, que possui um tamanho fixo, seguido pelos dados. Ele pode ou não possuir cabeçalhos de extensão. Estes ficam entre o cabeçalho básico e os dados. A Figura 12 mostra o formato de um datagrama IPv6.

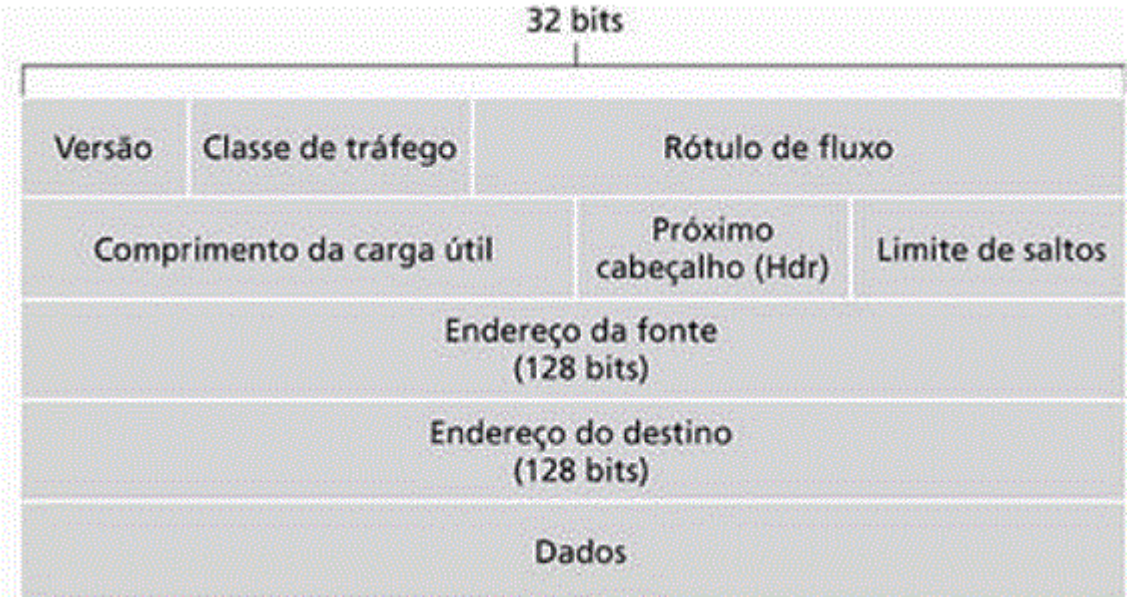


Figura 12: Formatobásico do datagrama IPv6.
Fonte: Kurose; Ross. Redes de computadores e a Internet (2010).

Cada campo do cabeçalho básico é descrito abaixo e são especificados na RFC 2460 (1998):

- Versão: 4-bit. *Internet Protocol* versão número = 6.
- Classe de Tráfego: 8-bit. Usado para identificar e distinguir entre diferentes classes ou prioridades de pacotes IPv6.
- Rótulo de fluxo: 20-bit. Usado para identificar um fluxo de datagramas.

- Comprimento da carga útil: 16-bit. É o número de octetos transportados no datagrama, excluindo o próprio cabeçalho.
- Próximo cabeçalho: 8-bit. Identifica o tipo de cabeçalho imediatamente a seguir ao cabeçalho-base IPv6.
- Limite de saltos: Reduzido em 1 por cada nó que encaminha o pacote. O pacote é descartado se o limite de saltos é reduzido para zero.
- Endereço da fonte:128-bit. Endereço do remetente do pacote.
- Endereço do destino:128-bit. Endereço do destinatário do pacote (possivelmente não o destinatário final, se um cabeçalho *Routing* está presente)

2.3.2 Endereçamento IPv6

Um endereço IPv6 é composto por 128 *bits*. Esses bits são segmentados em oito campos de 16 *bits* que são expressados no formato hexadecimal de 4 dígitos, de 0 a F, separados por ":". Exemplo:

0010000111011010000000001101001100000000000000000101111001110110000001010101010000000001111111111111110001010001001110001011010

=

21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A

Pode-se simplificar a representação dos endereços ocultando-se os zeros à esquerda de cada um dos blocos de 16 *bits* e no caso de blocos representados por uma sequência de zeros, pode-se representá-los com "::". Deve-se lembrar que a representação "::" só pode ser usada uma vez em cada endereço. Considerando o endereço IPv6 exemplificado anteriormente, a simplificação ficaria do seguinte modo:

21DA:D3::2F3B:2AA:FF:FE28:9C5A

Com 128 *bits* é possível obter 340.282.366.920.938.463.463.374.607.431.768.211.456 endereços (2^{128}) que equivalente a aproximadamente 79 octilhões de vezes a quantidade de endereços IPv4.

2.4 Governança da *Internet*

A *Internet* Global possui uma estrutura extremamente grande e complexa de ser gerenciada. Por isso, esta tarefa é delegada à autoridades regionais que têm como responsabilidade coordenar e manter o bom funcionamento da *Internet* .

> A governança da *Internet* no contexto mundial é hierarquizada, sendo que existem diversas "autoridades" (assim são denominadas) que têm por objetivo assegurar o bom funcionamento da rede. Dentre outras atribuições, compete a essas autoridades, por exemplo, a distribuição de endereços IPv4 e IPv6, distribuição de ASNs, registro de nomes DNS, entre outras atividades fundamentais para manter a *Internet* em operação.No topo dessa hierarquia está a IANA (*Internet Assigned Numbers Authority*), vinculada ao ICANN, que coordena as atividades globalmente. A IANA, no poder de suas atribuições, delega parte dessas atividades para autoridades com abrangência menor, normalmente da área de continentes que são denominadas RIR (acrônimo de *Regional Internet Registry*). Atualmente existem 5 entidades regionais que são: ARIN, RIPE NCC, APNIC, LACNIC e AfriNIC. (BRITO, 2013, Texto on-line)

A área coberta por cada RIR é ilustrada na Figura 13.

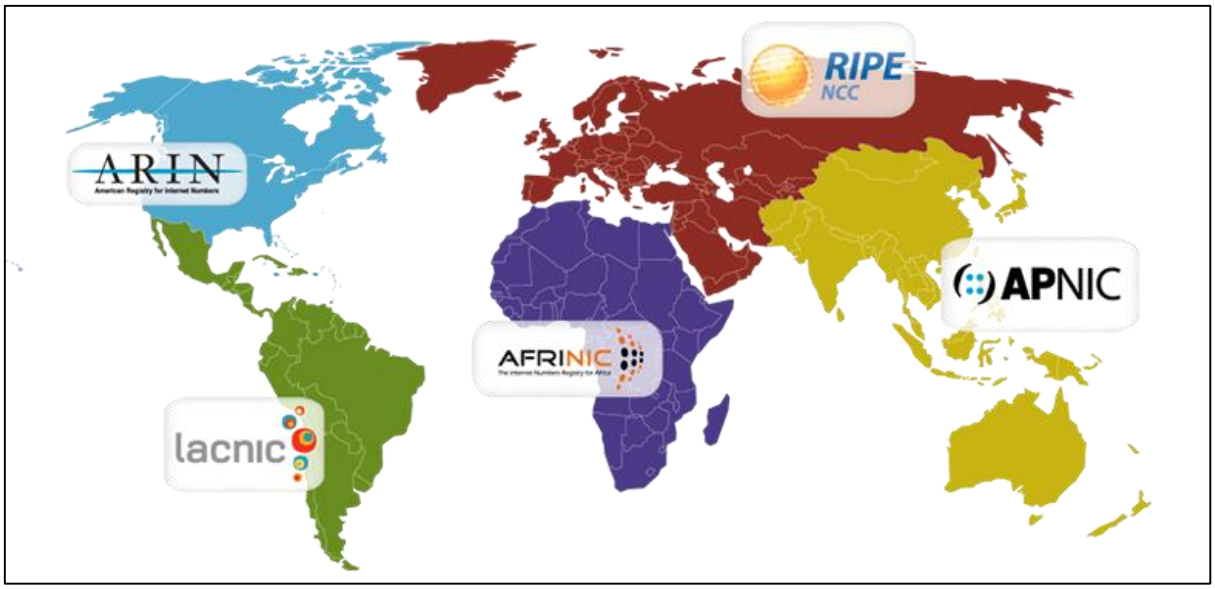


Figura 13: Áreas de cobertura dos Registros Regionais da *Internet.*
Fonte: http://labcisco.blogspot.com.br/2013/01/governanca-da-internet-no-mundo.html.

Como pode-se perceber, o LACNIC é o RIR responsável pela nossa região. O LACNIC é uma organização não governamental que serve à América Latina e o Caribe e foi estabelecida no Uruguai no ano de 2002.

2.5 Técnicas de Transição

Devido as mudanças introduzidas pelo IPv6, ele é incompatível com IPv4, por isso é necessário migrar para o novo protocolo. Durante um período de tempo eles deverão coexistir. As técnicas de transição foram criadas para que a migração do protocolo IPv4 para o IPv6 ocorra gradualmente.

Segundo o IPv6.br (2012) as técnicas de transição podem ser classificadas de acordo com sua funcionalidade em:

Pilha dupla: consiste na convivência do IPv4 e do IPv6 nos mesmos equipamentos, de forma nativa, simultaneamente. Essa técnica é a técnica padrão escolhida para a transição para IPv6 na Internet e deve ser usada sempre que possível.

Túneis: Permitem que diferentes redes IPv4 comuniquem-se através de uma rede IPv6, ou vice-versa.

Tradução: Permitem que equipamentos usando IPv6 comuniquem-se com outros que usam IPv4, por meio da conversão dos pacotes.

2.5.1 Pilha Dupla

Conforme consta na RFC 4213 (2005, p. 1) "manter a compatibilidade com o IPv4, enquanto implanta-se o IPv6 irá agilizar a tarefa de transição da *Internet* para o IPv6". Com base nesta afirmação, devemos considerar sempre que possível a implantação de pilha dupla, que permite uma transição gradual com os dois protocolos em operação.

De acordo com o Ipv6.br (2012), esta técnica consiste em IPv4 e IPv6 coexistindo no mesmo equipamento, ou seja, os dispositivos e roteadores são equipados com pilhas de ambos protocolos, tendo capacidade de enviar e receber os dois tipos de pacotes. Com isso, um nó Pilha Dupla, se comportará como um nó IPv6 na comunicação com outro nó IPv6 e se comportará como um nó IPv4 na comunicação com outro nó IPv4.

Para tornar possível a comunicação com ambos os protocolos, é necessário que sejam utilizados mecanismos para configuração de endereços, para que cada dispositivo tenha um endereço IPv4 e IPv6.

De acordo com a RFC 4213 (2005) nós IPv6/IPv4 utilizam mecanismos IPv4 (por exemplo, DHCP) para adquirir os seus endereços IPv4, e mecanismos do protocolo IPv6 (por exemplo, autoconfiguração de endereços *stateless* e/ou DHCPv6) para adquirir os seus endereços IPv6. O funcionamento da pilha dupla é ilustrado na Figura 14.

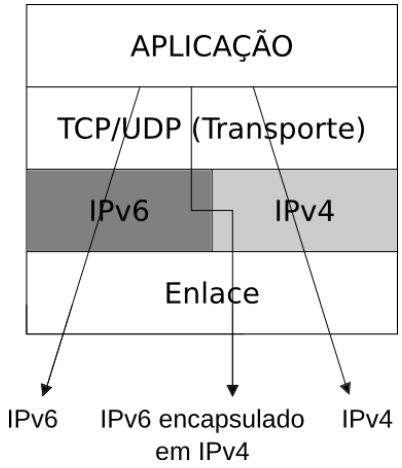


Figura 14: Funcionamento da pilha dupla.
Fonte: Ipv6.br. Transição.

Este método de transição permite uma implantação gradual, com a configuração de pequenas seções do ambiente de rede de cada vez. Além disso, caso no futuro o IPv4 não seja mais usado, basta simplesmente desabilitar a pilha IPv4 em cada nó. (IPv6.br,2012)

2.5.2 Túnel IPv6-over-IPv4

O tunelamento *IPv6-over-IPv4*, segundo a RFC 4213 (2005), é uma técnica utilizada para estabelecer túneis ponto-a-ponto através do encapsulamento de pacotes IPv6 dentro de cabeçalhos IPv4 para carregá-los sobre as infraestruturas de roteamento IPv4. A Figura 15 ilustra como ocorre o encapsulamento.

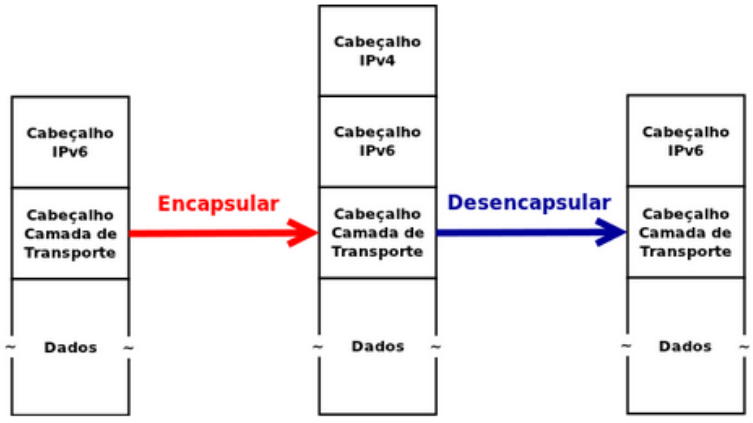


Figura 15: Encapsulamento 6in4.
Fonte: Ipv6.br (http://ipv6.br/entenda/transicao/#tecnicas-6o4).

Este encapsulamento se chama *6in4*ou *IPv6-in-IPv4*e é definido na RFC 4213(2005). Ao encapsular o pacote IPv6 dentro de um pacote IPv4, seus endereços de origem e destino são adequados para o IPv4 e no cabeçalho é colocado o tipo 41, por isto este encapsulamento também pode ser chamado de "protocolo 41". Quando o destino recebe o pacote com o tipo 41 ele irá remover o cabeçalho IPv4 e tratar o pacote como IPv6.

2.5.3 Tradução NAT64/DNS64

O NAT64 é definido na RFC 6146 (2011) como uma técnica *stateful* de tradução de pacotes IPv6 em IPv4 e vice-versa. O DNS64, definido na RFC 6147 (2011), auxilia o NAT64 sintetizando registro AAAA para registros A. As duas técnicas são utilizadas em conjunto para permitir que clientes somente IPv6 se comuniquem com um servidor somente IPv4, ou também com um nó IPv4. A RFC 6052 (2010) aborda o processo de tradução algorítmica de endereços IPv6 para endereços IPv4 e vice-versa. Este algoritmo também é usado em outros tradutores IPv4/IPv6. O NAT64 permite simultaneamente o compartilhamento de endereços IPv4 e usa o DNS64 para auxiliar a mapear os nomes de domínio. Com isso, *hosts* somente IPv6 podem acessar dispositivos IPv4 usando este mecanismo de tradução. Este processo é totalmente transparente ao usuário. No processo de tradução todos os endereços IPv4 da *Internet* são mapeados na rede do provedor de acesso para um prefixo IPv6 pré-definido. Este prefixo pode ser definido pelas operadoras, porém na RFC 6052 há um bloco de endereços reservado exclusivamente para esta finalidade que é o 64:ff9b::/96. Quando um *host* IPv6 desejar acessar conteúdos IPv6, esse acesso será direto, porém quando desejar acessar conteúdos em IPv4, será realizada uma consulta ao DNS, que é responsável por mapear nomes de domínio em endereços IP. O DNS64 funciona como um recursivo comum, mas caso o nome consultado não possua um endereço IPv6 atrelado ao nome, este será acrescentado na resposta usando o prefixo 64:ff9b::/ 96 mais os 32 *bits* do endereço IPv4 (Figura 16), ou seja, será retornado um endereço no formato "64:ff9b::endereço_ipv4".

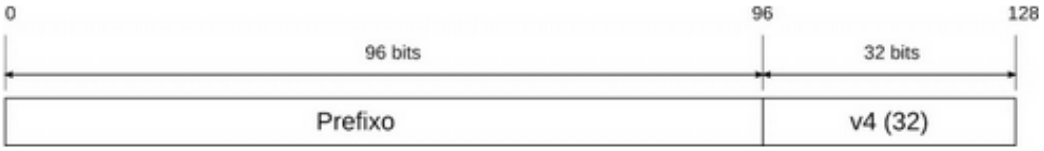


Figura 16: Endereço IPv4 traduzido para IPv6 pelo NAT64.
Fonte: IPv6.br (http://ipv6.br/entenda/transicao/#tecnicas-64).

Para o usuário, é como se o conteúdo que ele acessou já estivesse em IPv6 e é iniciada a comunicação utilizando este protocolo. Como o endereço de destino possui o prefixo de mapeamento NAT64 os pacotes são encaminhados para o dispositivo responsável por fazer a tradução *stateful* do IPv6 para o IPv4. Na tradução, o endereço IPv6 de origem do usuário é substituído por um IPv4 do *pool* de endereços disponíveis para o NAT. O endereço IPv4 de destino já está embutido no endereço IPv6 de destino e é facilmente obtido. Os dados do mapeamento são armazenados, para que seja possível o retorno dos pacotes. Na resposta a tradução inversa é realizada, utilizando os dados que foram armazenados. O processo realizado pelo NAT64/DNS64 é ilustrado sequencialmente na topologia da Figura 17.

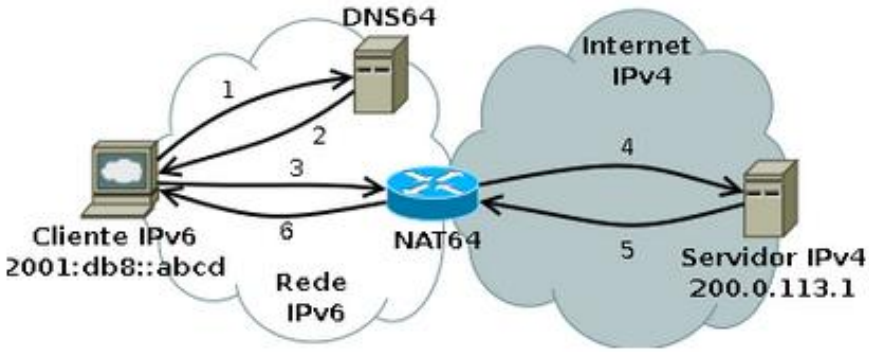


Figura 17: Topologia de rede do NAT64/DNS64.
Fonte: IPv6.br (http://ipv6.br/entenda/transicao/#tecnicas-64).

2.6 Autoconfiguração de endereços

Para o protocolo IPv6, são definidos tanto mecanismos *stateful* como *stateless* para autoconfiguração de endereços.

Segundo a RFC 4862 (2007, p. 2) a autoconfiguração de endereços *stateless* não requer nenhuma configuração manual nos *hosts*, porém nos roteadores são feitas configurações

mínimas, não havendo necessidade de servidores para esta tarefa. Na autoconfiguração *stateful*, os endereços das interfaces e outras informações de configuração e parâmetros são atribuídos por um servidor que mantém um banco de dados para controlar quais endereços foram atribuídos e para quais hosts.

Na RFC 3315 (2003) é especificado o *Dynamic Host Configuration Protocol* para IPv6 (DHCPv6) que é um protocolo utilizado por realizar a autoconfiguração de endereços *stateful*. Nas Seções 2.6.1 e 2.6.2 serão descritos cada um destes mecanismos com mais detalhes.

2.6.1 Autoconfiguração de endereços IPv6 *stateless*

A autoconfiguração de endereços IPv6 *stateless* é descrita na RFC 4862 de 2007, que é uma versão atualizada das RFC's 1971 (1996) e 2462 (1998).

O processo de autoconfiguração automática inclui a geração de um endereço *link-local*, geração de endereços globais via autoconfiguração de endereços *stateless* e o processo de detecção de endereço duplicado para verificar a unicidade dos endereços em um *link*. (RFC 4862, 2007, p.2, tradução nossa)

De acordo com a RFC 4862 (2007) o mecanismo *stateless* permite que um *host* gere seus próprios endereços usando uma combinação de informações localmente disponíveis e informações anunciadas por roteadores. Os roteadores anunciam prefixos que identificam a(s) sub-rede(s) associadas com o *link*, enquanto os *hosts* geram um "identificador de interface" que identifica unicamente uma interface na sub-rede. Um endereço é formado pela combinação dos dois.

Neste modo de configuração, quando não existem roteadores para anunciarem o prefixo da sub-rede, o *host* só gera o endereço de *link-local*, que são suficientes para que os *hosts* se comuniquem com os outros nós ligados ao mesmo *link*.

Na Figura 18, é possível visualizar como ocorre a autoconfiguração *stateless* de endereços quando a iniciativa provém do próprio *host*. Este envia uma requisição com a mensagem *Router Solicitation* aos roteadores da rede. A partir daí é gerada uma resposta *Router Advertisement* com as informações para a autoconfiguração do dispositivo.

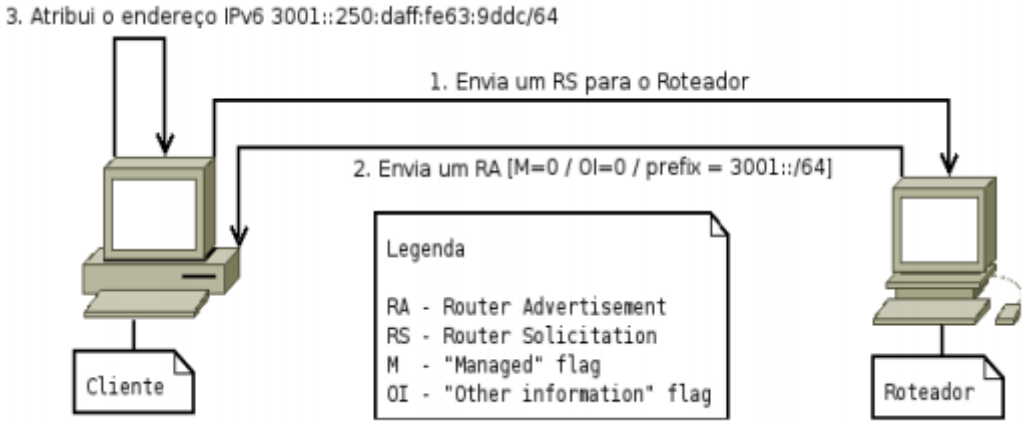


Figura 18: Autoconfiguração de endereços *stateless* IPv6.
Fonte: Autoconfiguração do protocolo IPv6 (http://www.pucrs.br/).

Ainda de acordo com a RFC 4862(2007) a abordagem *stateless* é usada quando não há muita preocupação de quais endereços exatamente os hosts vão usar, contanto que eles sejam endereços únicos e devidamente roteáveis. Portanto, quando se deseja manter um controle mais rígido sobre quais endereços estão sendo atribuídos exatamente, é necessário utilizar o DHCPv6 que possui uma abordagem *stateful*.

2.6.2 Autoconfiguração de endereços IPv6 *stateful*

O *Dynamic Host Configuration Protocol for* IPv6 é especificado na RFC 3315(2003), e consiste em um protocolo cliente-servidor de autoconfiguração *stateful*. Ele pode ser utilizado tanto para distribuir endereços IPv6 quanto para divulgar informações da rede. O DHCPv6 e DHCPv4 atuam de forma independente. Para efetuar a troca de mensagens o primeiro utiliza as portas UDP 546 para clientes e 547 para roteadores (*relay agents)* e servidores, já o segundo utiliza a UDP 68 e a UDP 67. Normalmente, os clientes utilizam seus endereços de *link-local* para comunicarem-se com o servidor DHCPv6, porém outros endereços podem ser utilizados dependendo do servidor. O endereço de destino das mensagens vindas dos clientes é o *All_DHCP_Relay_Agents_and_Server*, definido na RFC 3315 como "FF02::1:2". Este endereço é um endereço *multicast* com escopo de enlace usado para clientes enviarem mensagens aos *relay agents* e aos servidores se localizam na vizinhança. Outro endereço *multicast* utilizado pelo DHCPv6 é o *All_DHCP_Servers*, definido na RFC 3315 como FF05::1:3. Este é um endereço *multicast* de escopo de site, usado pelos *relay agentes* para se

comunicarem com os servidores DHCPv6 ao retransmitirem as mensagens recebidas dos clientes.

A Figura 19 ilustra as trocas de mensagens para autoconfiguração *stateful* via DHCPVv6. O cliente envia uma mensagem *"Solicit"* para a rede procurando por servidores habilitados e o Servidor DHCPv6 responde com a mensagem *"Advertise"* anunciando-se para fornecer informações. Nesta mensagem está contido o endereço solicitado para auxiliar o cliente na escolha dos servidores que responderam. O cliente então, elege um servidor e envia a mensagem "*Request"* para requisitar permissão de uso do endereço global passado e o Servidor DHCPv6 armazena, num registro, o endereço passado ao cliente e manda uma mensagem "*Reply"* como confirmação.

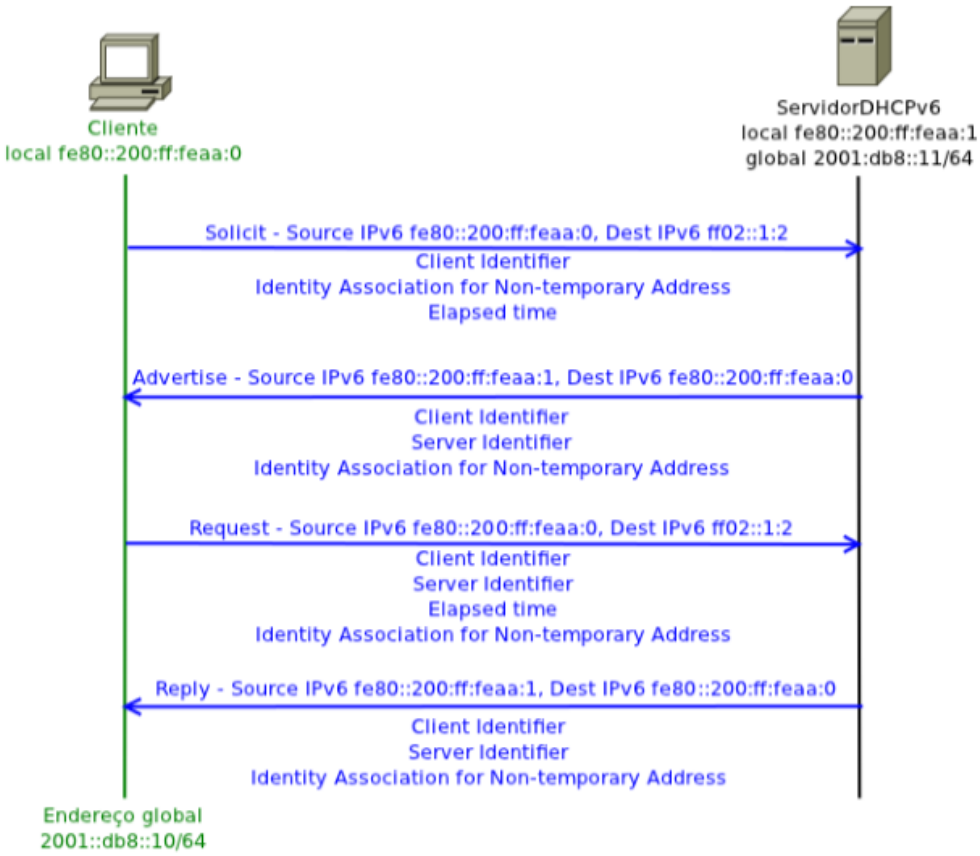


Figura 19: Trocas de mensagens para autoconfiguração *stateful* via DHCPVv6.
Fonte:NIC.br. Apostila Curso básico IPv6 (http://ipv6.br).

Depois de recebida a última mensagem, o cliente DHCPv6 inicia um procedimento de detecção de endereços duplicados no enlace para saber se pode utilizar este endereço em suas comunicações.

O DHCPv6 também pode ser utilizado em modo *stateless.* Neste caso, ele não guarda registros de informações e também não delega endereços para *hosts* clientes. Ele apenas passa

informações adicionais, tais como *Domain Name System*(DNS) e *Network Time Protocol* (NTP) entre outras informações.

3 TRABALHOS RELACIONADOS

Devido a importância da implantação do IPv6 para que seja possível sustentar o crescimento contínuo da Internet, existem atualmente estudos relacionados ao protocolo e às diversas formas de se realizar a transição para o mesmo. Portanto, nesta sessão serão elencados alguns estudos que possuem, assim como este trabalho, o objetivo de analisar as técnicas de transição do protocolo IPv4 para o IPv6.

Monego (2014) analisa o cenário atual e propõe a implantação de túneis para que clientes IPv6 possam comunicar-se via Internet IPv4 até que os provedores de *Internet* incorporem totalmente o novo protocolo. Dentre as técnicas de tunelamento existentes, foram escolhidas as técnicas *TunnelBroker*, *6over4* e *Tunnel* GRE, para um estudo mais aprofundado. A autora sugere, para trabalhos futuros, a implementação da Pilha Dupla a fim de proporcionar à rede IPv6 implementada, acesso a sítios IPv4, tornando os dois protocolos interoperantes.

Silveira (2012) concentra seu estudo na convivência entre os dois protocolos, IPv4 e IPv6, ou seja, possibilitar uma comunicação transparente entre redes IPv6 e demais redes em uma *Internet* majoritariamente IPv4. O autor focou-se nas técnicas de Pilha Dupla, e de Tradução (NAT64/DNS64 e IVI), pois possibilitam a comunicação de ilhas IPv6 com redes IPv4. No entanto, é ressaltada a importância da migração para o novo protocolo, visto o esgotamento dos endereços IPv4, e considerando este processo inevitável.

Pletsch (2012) aplicou seu estudo à realidade de uma operadora de serviços de comunicação multimídia. Ele destaca que várias questões devem ser consideradas para que a implementação não comprometa a segurança, a disponibilidade e o desempenho das redes, que com o IPv4, ainda trabalham de forma aceitável. Foram analisadas várias técnicas de transição e coexistência para que fosse possível definir a melhor a ser implantada em um ISP levando em consideração a viabilidade de cada método. Constatou-se também neste estudo que muitos equipamentos ainda não possuem suporte para IPv6, mas na maioria deles, atualizações de *softwares* são disponibilizadas pelos fabricantes tornando estes equipamentos capacitados para IPv6.

4 TRABALHO PROPOSTO

Neste Capítulo serão descritas as etapas desenvolvidas para a realização deste trabalho. Na Seção 4.1 será descrito o processo para implantação da Pilha Dupla em uma rede local, na Seção 4.2 será descrito o processo para criação de um túnel *6over4* para interconectar redes IPv6 e; por fim, na Seção 4.3 será descrita a implantação do mecanismo de tradução NAT64/DNS64.

4.1 Implantação da Pilha Dupla

Como foi mencionado na Seção 2.5 do Capítulo 2, a técnica de pilha dupla consiste na convivência da pilha de ambos os protocolos, IPv4 e IPv6, no mesmo equipamento, funcionando simultaneamente. Para que isto seja possível é necessário que eles possuam tanto endereçamento IPv4 como IPv6. Nesta implantação de Pilha Dupla, será utilizado um servidor DHCP para atribuição de endereços IPv4 e IPv6 aos *hosts* clientes, e no servidor serão atribuídos endereços estáticos manualmente. Será utilizado também o *radvd* para divulgação do roteador, pois ao contrário do DHCPv4, o DHCPv6 não divulga o *default gateway*. O *radvd* é um mecanismo *stateless* que além de divulgar o roteador também pode ser utilizado para autoconfiguração de endereços IPv6, porém esta funcionalidade não será utilizada no presente trabalho.

A Pilha Dupla (ou *Dual Stack*) será implantada em um cenário que possui uma máquina virtual atuando como servidor DHCP e duas atuando como clientes que obterão seus endereços IPv4 e IPv6 através do servidor. Assume-se que estas máquinas utilizam sistema operacional *Linux*, distribuição *Ubuntu* 12.04 LTS *Desktop*. A Figura 20 ilustra o cenário inicial de implantação.

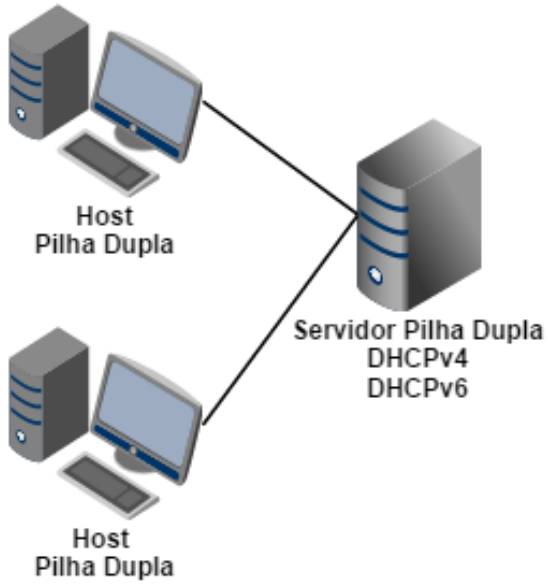


Figura 20: Cenário de implantação da Pilha Dupla.
Fonte: Do Autor.

Neste cenário, a interface que faz a comunicação do Servidor com a rede interna é a eth1, e a interface que possibilita a comunicação com o ambiente externo é a eth0.

A interface eth0 da máquina virtual Servidor foi configurada como *Bridge*, com endereços IPv4 e IPv6 estáticos. Já a interface eth1, foi configurada como Rede Interna e lhe foram atribuídos endereços IPv4 e IPv6 também estáticos. A Figura 21 mostra como ficaram configuradas as interfaces do Servidor.

```
root@leticiaTcc:/home/leticia# ifconfig
eth0      Link encap:Ethernet  Endereço de HW 08:00:27:44:f7:30
          inet end.: 192.168.0.109  Bcast:192.168.0.255  Masc:255.255.255.0
          endereço inet6: fe80::a00:27ff:fe44:f730/64 Escopo:Link
          endereço inet6: 2001:db8:dcba::109/64 Escopo:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
          pacotes RX:3383 erros:0 descartados:0 excesso:0 quadro:0
          Pacotes TX:3149 erros:0 descartados:0 excesso:0 portadora:0
          colisões:0 txqueuelen:1000
          RX bytes:3194616 (3.1 MB) TX bytes:317005 (317.0 KB)

eth1      Link encap:Ethernet  Endereço de HW 08:00:27:03:91:3c
          inet end.: 192.168.1.1  Bcast:192.168.1.255  Masc:255.255.255.0
          endereço inet6: fe80::a00:27ff:fe03:913c/64 Escopo:Link
          endereço inet6: 2001:db8:abcd::1/64 Escopo:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
          pacotes RX:0 erros:0 descartados:0 excesso:0 quadro:0
          Pacotes TX:71 erros:0 descartados:0 excesso:0 portadora:0
          colisões:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:10264 (10.2 KB)

lo        Link encap:Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          endereço inet6: ::1/128 Escopo:Máquina
          UP LOOPBACK RUNNING  MTU:65536  Métrica:1
          pacotes RX:618 erros:0 descartados:0 excesso:0 quadro:0
          Pacotes TX:618 erros:0 descartados:0 excesso:0 portadora:0
          colisões:0 txqueuelen:0
          RX bytes:48176 (48.1 KB) TX bytes:48176 (48.1 KB)
```

Figura 21:Interfaces do Servidor Pilha Dupla.
Fonte: Do Autor.

Primeiramente, na criação do cenário, foi editado o arquivo *"/etc/sysctl.conf"* do Servidor Pilha Dupla, e habilitado o encaminhamento de pacotes IPv4 e IPv6. Para permitir o encaminhamento foi alterado o valor de "0" para "1". Esta configuração pose ser visualizada na Figura 22.

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
```

Figura 22: Habilitando o encaminhamento de pacotes IPv4 e IPv6.
Fonte: Do Autor.

Após, o Servidor foi configurado para permitir que os *hosts* da LAN com endereços IPv4 privados comuniquem-se com redes públicas externas. Para isto, é necessário fazer

mascaramento dos pedidos dos hosts da LAN com o endereço IPv4 do dispositivo externo do *firewall* (neste caso, eth0). O seguinte comando foi feito:

```
root@leticiaTcc:/home/leticia# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Para configurar a atribuição de endereços, é necessário instalar o pacote "*isc-dhcp-server*" e realizar as seguintes configurações:

- Acessar o arquivo *dhcpd.conf*, que é o arquivo de configuração do servidor DHCP para atribuição de endereços IPv4, localizado no diretório " */etc/dhcp*";
- Adicionar neste arquivo informações da subrede, que serão divulgadas aos *hosts* clientes. Pode-se notar que neste arquivo existem vários exemplos de configurações possíveis para subrede. O arquivo *dhcpd.conf* completo encontra-se no Apêndice A. Na Figura 23 estão algumas configurações básicas para atribuição de endereços dinâmicos.

```
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
option routers 192.168.1.1;
option domain-name-servers 8.8.8.8;
range 192.168.1.50 192.168.1.199;
}
```

Figura 23: Configurações básicas *dhcpd.conf.*
Fonte: Do Autor.

A saber:

- *default-lease-time 600*:define que o tempo padrão de empréstimo do endereço IP será de 600 segundos. Este tempo é determinado pelo gerente da rede.
- *max-lease-time 7200 :* define que, caso o cliente solicite um tempo maior de "empréstimo" do endereço IP, o tempo máximo será de 7200 segundos.
- *subnet 192.168.1.0 netmask 255.255.255.0 :* especifica que o equipamento pertencerá a sub-rede 192.168.1.0 e que a máscara de rede é 255.255.255.0.
- *option routers 192.168.1.1:* especifica qual será o *gateway* da sub-rede.
- *option domain-name-servers 8.8.8.8*: especifica o endereço do servidor DNS, utilizado para resolução de nomes. Pode ser definido mais de um servidor DNS.
- *range 192.168.1.50 192.168.1.199 :* especifica a faixa de endereços IP que podem ser atribuídos aos clientes. Neste caso, de 192.168.1.50 à 192.168.1.199.

O *Dynamic Host Configuration Protocol for IPv6* (DHCPv6), utilizado para atribuir endereços IPv6, está contido no pacote "*isc-dhcp-server*", portanto, basta fazer o seguinte:

- Criar um arquivo *"dhcpd6.conf"* no diretório *"/etc/dhcp/"*;
- Adicionar ao arquivo *"dhcpd6.conf"*, as informações da sub-rede que serão divulgadas aos *hosts* clientes (Figura 24).

```
#Arquivo de configuração dhcpd6.conf

default-lease-time 600;
max-lease-time 7200;
subnet6 2001:db8:abcd::/64 {
range6 2001:db8:abcd::50 2001:db8:abcd::254;
option dhcp6.name-servers 2001:4860:4860::8888;
}
```

Figura 24: Configurações básicas dhcpd6.conf.
Fonte: Do Autor.

Terminadas as configurações é necessário reiniciar o serviço com o comando "*service isc-dhcp-server restart*".

Como pode-se observar na Figura 16, não há nenhuma configuração que defina o roteador *default* da sub-rede. Por isso, foi realizada a seguinte configuração no Servidor para que ele se auto-divulgue como *default gateway*:

- Foi instalado o *radvd* com o comando " *apt-get install radvd"*;
- No arquivo "*/etc/radvd.conf"* (Figura 25) foram adicionadas as seguintes configurações:

```
# Configuração do radvd.conf para divulgação do roteador (sem divulgação de
 prefixo para autoconfiguração).

interface eth1 {
        AdvSendAdvert on; #manda os anúncios de tempos em tempos
        MinRtrAdvInterval 3; #tempo mínimo para retrasmissão dos anúncios
        MaxRtrAdvInterval 10; #tempo máximo para retrasmissão dos anúncios
        AdvManagedFlag on; #permite que seja usado DHCPv6 em conjunto
};
```

Figura 25: Arquivo de configuração *radvd.conf*.
Fonte: Do Autor.

Para que os clientes recebam seus endereços IPv4 e IPv6 via DHCP, é necessário que conste no arquivo "*/etc/network/interfaces*" (Figura 26) a seguinte configuração na interface que deverá receber o endereço (neste caso, utilizou-se a eth1):

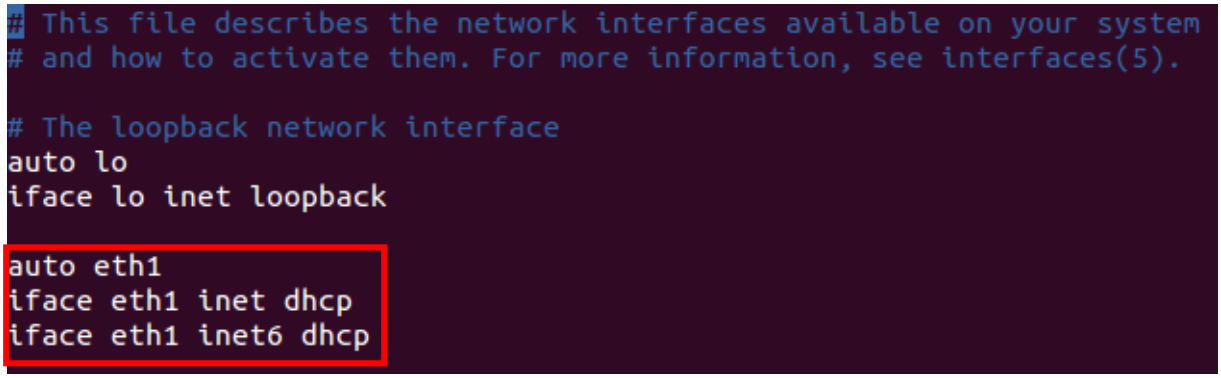


Figura 26: Arquivo */etc/network/interfaces* de um *host* cliente.
Fonte: Do Autor.

Com a realização destas configurações, já é possível haver comunicação entre os dispositivos da rede local. Esta comunicação pode ser testada através do comando "*ping endereço_ipv4*" para verificar a comunicação IPv4 entre os *hosts*; e com o comando "*ping6 endereço_ipv6*" é possível verificar a comunicação IPv6 entre os *hosts*.

Para saber quais endereços IPv4 que foram atribuídos pelo servidor DHCP, basta inserir o comando "*less /var/lib/dhcp/dhcpd.leases*"; e para saber os endereços IPv6 atribuídos o comando é "*less/var/lib/dhcp/dhcpd6.leases"* Desta forma, pode-se haver um maior controle sobre os endereços que são atribuídos, para quem são atribuídos e por quanto tempo este endereço permanecerá em uso.

Com o método de Pilha Dupla, não é necessário configurar nenhum mecanismo de encapsulamento em redes internas, porém, para haver comunicação com redes externas, muitas vezes é necessário utilizar outros mecanismos como o tunelamento e a tradução. Na Seção 4.2 será descrito como estabelecer uma comunicação em IPv6 entre duas redes via infraestrutura IPv4.

4.2 Implantação do Túnel IPv6-over-IPv4

Ao cenário existente (REDE A), foi adicionada uma nova rede (REDE B), para que seja possível demonstrar como ocorre a comunicação entre redes IPv6 através de uma infraestrutura IPv4. Nesta Seção, a proposta é implantar um túnel que possibilite esta comunicação. Será usado um túnel *IPv6-over-IPv4*, também chamado de *6over4*, que encapsulará os pacotes IPv6 em pacotes IPv4, estabelecendo-se assim, a comunicação. O cenário de implantação do túnel é ilustrado na Figura 27.

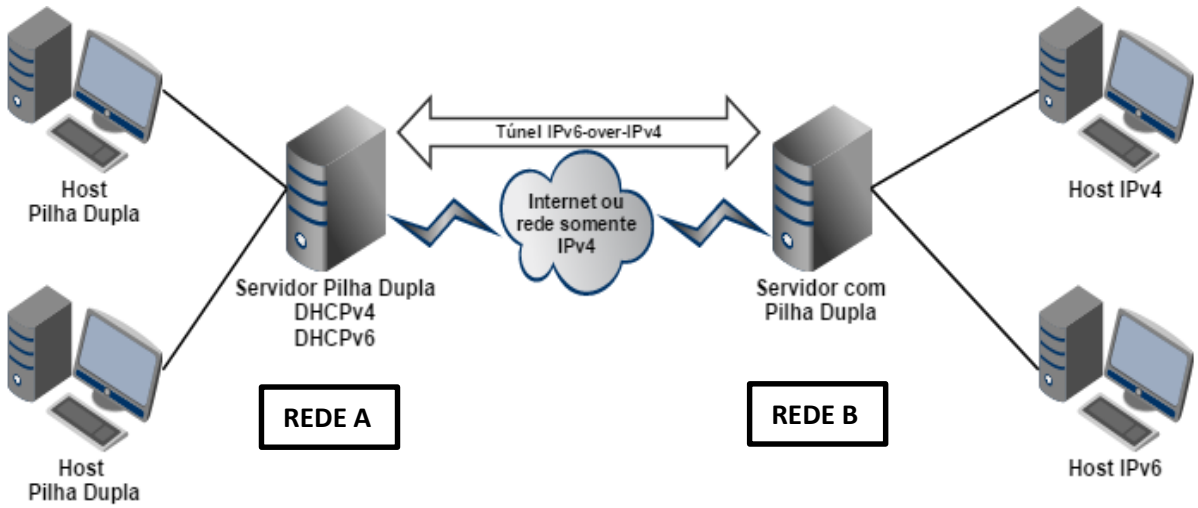


Figura 27: Cenário de implantação do Túnel 6over4.
Fonte: Do Autor.

As interfaces do servidor da rede B, foram configuradas de forma semelhante às do servidor da rede A, com endereços IPv4 e IPv6 estáticos.

Para criar um túnel estático entre os servidores da rede A e da rede B foram realizadas configurações manualmente conforme a Figuras 28 e a Figura 29.

- No servidor da rede A:

```
root@leticiaTcc:/home/leticia# ip tunnel add tun6over4 mode sit ttl 64
 remote 192.168.0.103 local 192.168.0.109
root@leticiaTcc:/home/leticia# ip link set dev tun6over4 up
root@leticiaTcc:/home/leticia# ip -6 route add 2001:db8:cdba::103 dev
tun6over4
```

Figura 28: Comandos para criação do túnel 6over4 no servidor da rede A.
Fonte: Do Autor.

A saber:

- 192.168.0.103 é o endereço IPv4 do servidor da rede B.
- 192.168.0.109 é o endereço IPv4 do servidor da rede A.
- 2001:db8:cdba::103 é o endereço IPv6 do servidor da rede B.

- No servidor da rede B:

```
root@servidor2:/home/servidor2# ip tunnel add tun6over4 mode sit ttl 64
remote 192.168.0.109 local 192.168.0.103
root@servidor2:/home/servidor2# ip link set dev tun6over4 up
root@servidor2:/home/servidor2# ip -6 route add 2001:db8:dcba::109 dev
tun6over4
```

Figura 29: Comandos para criação do túnel 6over4 no servidor da rede B.
Fonte: Do Autor.

A saber:

- 192.168.0.109 é o endereço IPv4 do servidor da rede A.
- 192.168.0.103 é o endereço IPv4 do servidor da rede B.
- 2001:db8:dcba::109 é o endereço IPv6 do servidor da rede A.

Nesta técnica de tunelamento ponto-a-ponto, é necessário saber os endereços IPv4 das duas pontas do túnel para que seja possível criá-lo. As pontas do túnel também devem possuir endereçamento IPv6 para que os pacotes recebidos possam ser tratados adequadamente.

4.3 Implantação do NAT64/DNS64

Para implantar o NAT64/DNS64 foi utilizada a REDE B do cenário de implantação do túnel 6over4 (Figura 30). No servidor, foram configurados endereços IPv4 e IPv6 estáticos na

interface de rede interna. Este servidor, utilizando o NAT64/DNS64, intermediará a comunicação entre o Host IPv4 e o Host IPv6.

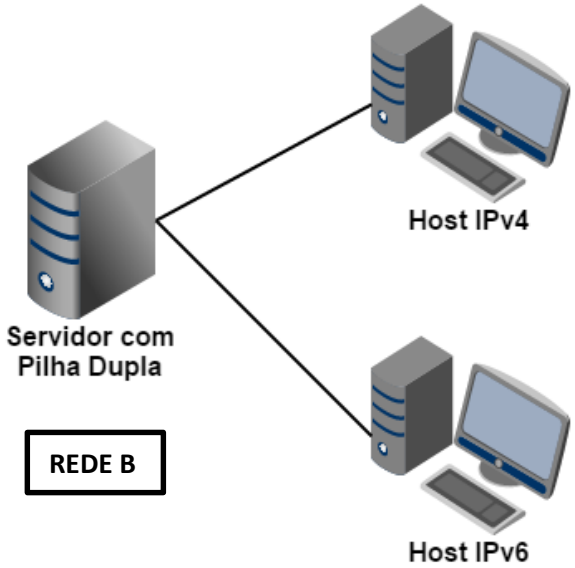


Figura 30: Cenário de implantação do NAT64/DNS64.
Fonte: Do Autor.

Segundo a Equipe IPv6.br (2012), o NAT64 possui implementações para *hosts* Linux, Windows, roteadores de marcas renomadas no mercado e roteadores domésticos baseados em Linux. Para configurar o NAT64 no servidor, que possui sistema operacional Linux, distribuição Ubuntu 12.04 LTS *Desktop*, foi utilizada a implementação *open-source* desenvolvida pelo projeto *Ecdysis*, pois foi a que se obteve maior quantidade de informações. Deve ser realizado o *download* do arquivo fonte em <http://ecdysis.viagenie.ca/download/ecdysis-nat64-20140422.tar.gzf>. Esta versão do NAT64 requer um *kernel* superior a 2.6.31. Para realizar o *download* é necessário o preenchimento de um formulário (Figura 31).

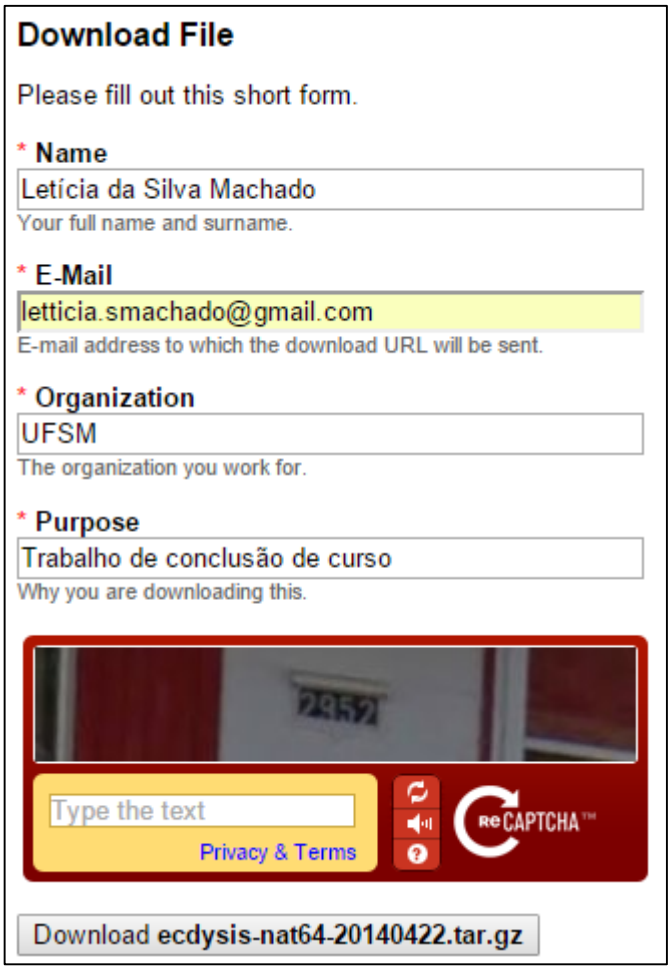


Figura 31: Formulário de preenchimento para *download* do NAT64.
Fonte: Do Autor.

Após concluído o *download*, deve-se extrair os arquivos para uma pasta adequada. Neste caso, o arquivo foi descompactado em "*/home/servidor2/Downloads*". Após, foram realizados os seguintes comandos:

- Foi compilado o módulo do *kernel:*

```
root@servidor2:/home/servidor2/Downloads/ecdysis-nat64-20140422# make && make install
```

- No arquivo *nat64-config.sh* (Figura 32) foram comentadas as linhas abaixo para que os comandos contidos nelas não sejam executados automaticamente.

```
# Load the nf_nat64 module
#modprobe -r nf_nat64
#modprobe nf_nat64 nat64_ipv4_addr=$IPV4_ADDR nat64_prefix_addr=$PREFIX_ADDR
nat64_prefix_len=$PREFIX_LEN
```

Figura 32: Arquivo nat64-config.sh do NAT64
Fonte: Do Autor.

- Em seguida, habilitou-se o módulo do *kernel*:

```
root@servidor2:/home/servidor2/Downloads/ecdysis-nat64-20140422# insmod nf_nat64.ko
nat64_ipv4_addr=192.168.0.103 nat64_prefix_addr=64:ff9b:: nat64_prefix_len=96
```

A saber:

- "192.169.0.103" é o endereço IPv4 da interface que está conectada à *Internet.*
- "64:ff9b::" é o prefixo recomendado para fins de mapeamento. Segundo a Equipe IPv6.br (2012) o prefixo IPv6 pode ser escolhido pela operadora, mas é recomendada a utilização do prefixo 64:ff9b::/96, reservado especificamente para a utilização em algoritmos de mapeamento de endereços IPv4 em IPv6. Por exemplo, o IPv4 203.0.113.1 seria convertido para o endereço IPv6 64:ff9b::203.0.113.1.
- "94" é o tamanho do prefixo.

- Após a habilitação do módulo deve-se executar o arquivo de configuração do NAT64 com o seguinte comando:

```
root@servidor2:/home/servidor2/Downloads/ecdysis-nat64-20140422# ./nat64-config.sh
192.168.0.103
```

A saída deste comando deve ser o seguinte:

```
**************
nf_nat64 setup
**************

Info: Using 192.168.0.103 as the NAT64 IPv4 address.
Info: Using 64:ff9b::/96 as the NAT64 Prefix.


+ ifconfig nat64 up
+ ip -6 route add 64:ff9b::/96 dev nat64
+ sysctl -w net.ipv4.conf.all.forwarding=1
net.ipv4.conf.all.forwarding = 1
+ sysctl -w net.ipv6.conf.all.forwarding=1
net.ipv6.conf.all.forwarding = 1
```

Figura 33: NAT64 habilitado.
Fonte: Do Autor.

Pode-se verificar que ao executar o comando "*ifconfig*" no servidor da REDE B, aparecerá uma interface do NAT64 (Figura 34).

```
nat64     Link encap:Não Especificado  Endereço de HW 00-00-00-00-00-00-00-00-00-00
-00-00-00-00-00-00
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Métrica:1
          pacotes RX:0 erros:0 descartados:0 excesso:0 quadro:0
          Pacotes TX:0 erros:0 descartados:0 excesso:0 portadora:0
          colisões:0 txqueuelen:0
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Figura 34: Interface NAT64 no servidor da Rede B.
Fonte: Do Autor.

Para configurar o DNS64 é necessário fazer *download* do BIND9, que é um servidor DNS, disponível no *site* <http://www.isc.org/software/bind>. Após o *download*, deve-se editar o arquivo *named.conf* (Figura 35), localizado no diretório "*/etc/bind",* e adicionar as seguintes configurações:

```
options {
        listen-on-v6 {any;};
        dns64 64:ff9b::/96 {
                clients {any;};
                mapped {any;};
                sufix ::;
                recursive-only yes;
                break-dnssec yes;
        };
};
```

Figura 35: Arquivo named.conf do BIND9.
Fonte: Do Autor

Após editado, deve-se salvar as configurações e executar o arquivo com o seguinte comando:

```
root@servidor2:/etc/bind# named -c /etc/bind/named.conf
```

No *host IPv6* é necessário adicionar uma rota para o prefixo de rede usado pelo NAT64, neste caso para o prefixo 64:ff9b::/96, para que os endereços IPv4 sejam convertidos em IPv6. Foi utilizado o comando "*ip route add 64:ff9b::/96 via 2001:db8:b0ca::1 dev eth4"*para adicionar a rota.

5 TESTES E RESULTADOS

Neste capítulo serão demonstrados os testes realizados em cada cenário de implantação. Foi utilizada a ferramenta *ping*, presente em praticamente todos os sistemas operacionais, para verificar a conectividade entre os *hosts*, a latência mínima, média e máxima (parâmetros *rtt min/avg/max*), o *jitter* (parâmetro *mdev*) e também a porcentagem de perda de pacotes. E o Wireshark, que é um *software* gratuito utilizado para analisar pacotes que trafegam na rede, foi empregado para verificar informações dos cabeçalhos dos pacotes. Na Seção 5.1 consta o teste de validação da técnica Pilha Dupla, na Seção 5.2 consta o teste de validação do túnel *6over4* e na Seção 5.3 consta o teste de validação da técnica de tradução NAT64/DNS64.

5.1 Teste da técnica Pilha Dupla

A partir do comando *ping* realizado entre os hosts do cenário de implantação da Pilha Dupla (Figura 20), pode-se perceber que a comunicação em IPv4 é estabelecida com sucesso e com a utilização do comando *ping6*, pode-se verificar que a comunicação em IPv6 também foi estabelecida. Os testes de comunicação entre os *hosts* podem ser visualizados na Figura 36 e na Figura 37.

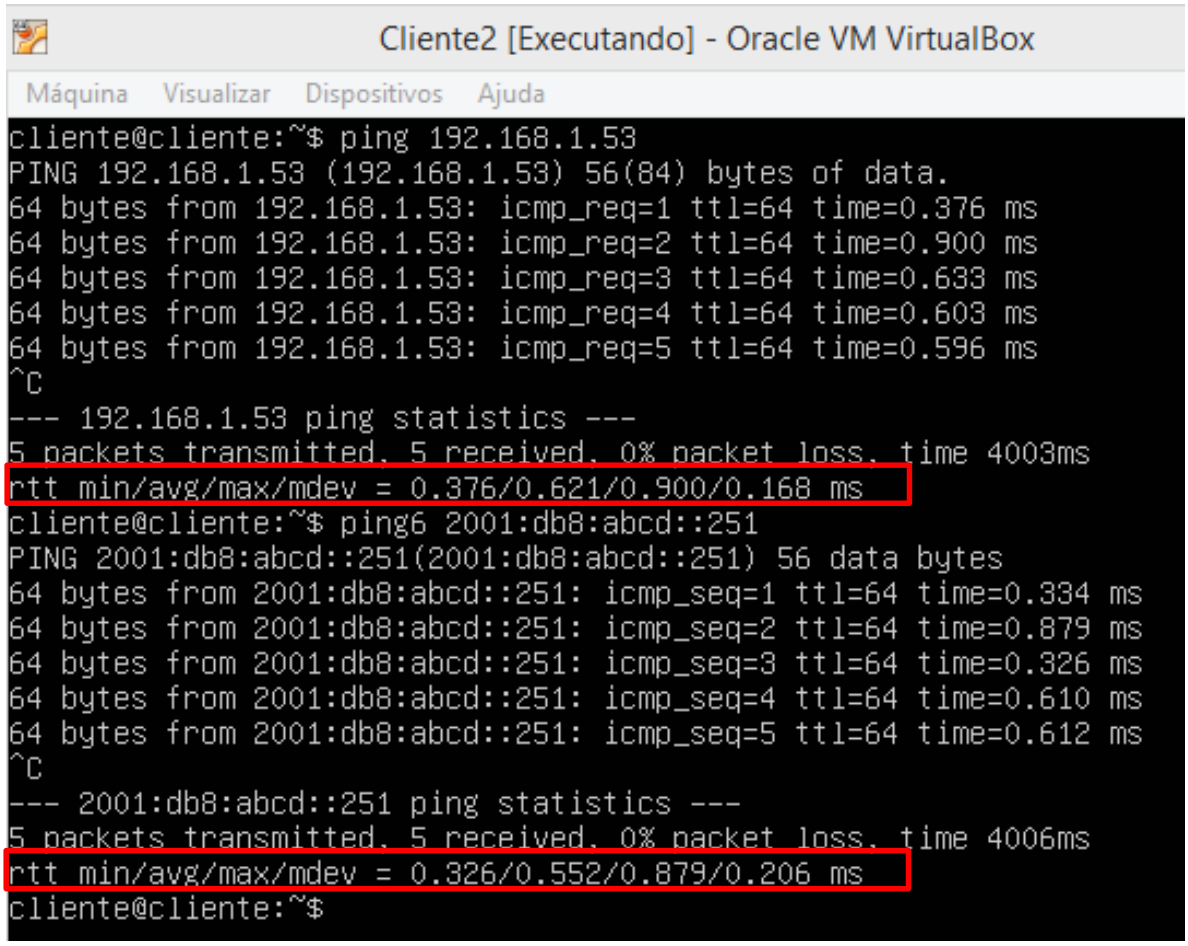


Figura 36: Teste de ping de um host pilha dupla para outro host pilha dupla.
Fonte: Do Autor.

Neste caso da Figura 36, analisando os resultados obtidos com *ping e ping6* e comparando-os, pode-se notar que o tempo médio de latência (*avg*) foi menor utilizando IPv6 e o *jitter* (*mdev*), que é a variação da latência no tempo, foi menor com IPv4. Isso significa que o tempo médio de atraso com IPv6 foi menor em relação ao IPv4, porém com o IPv6 houve maior variação no tempo de latência.

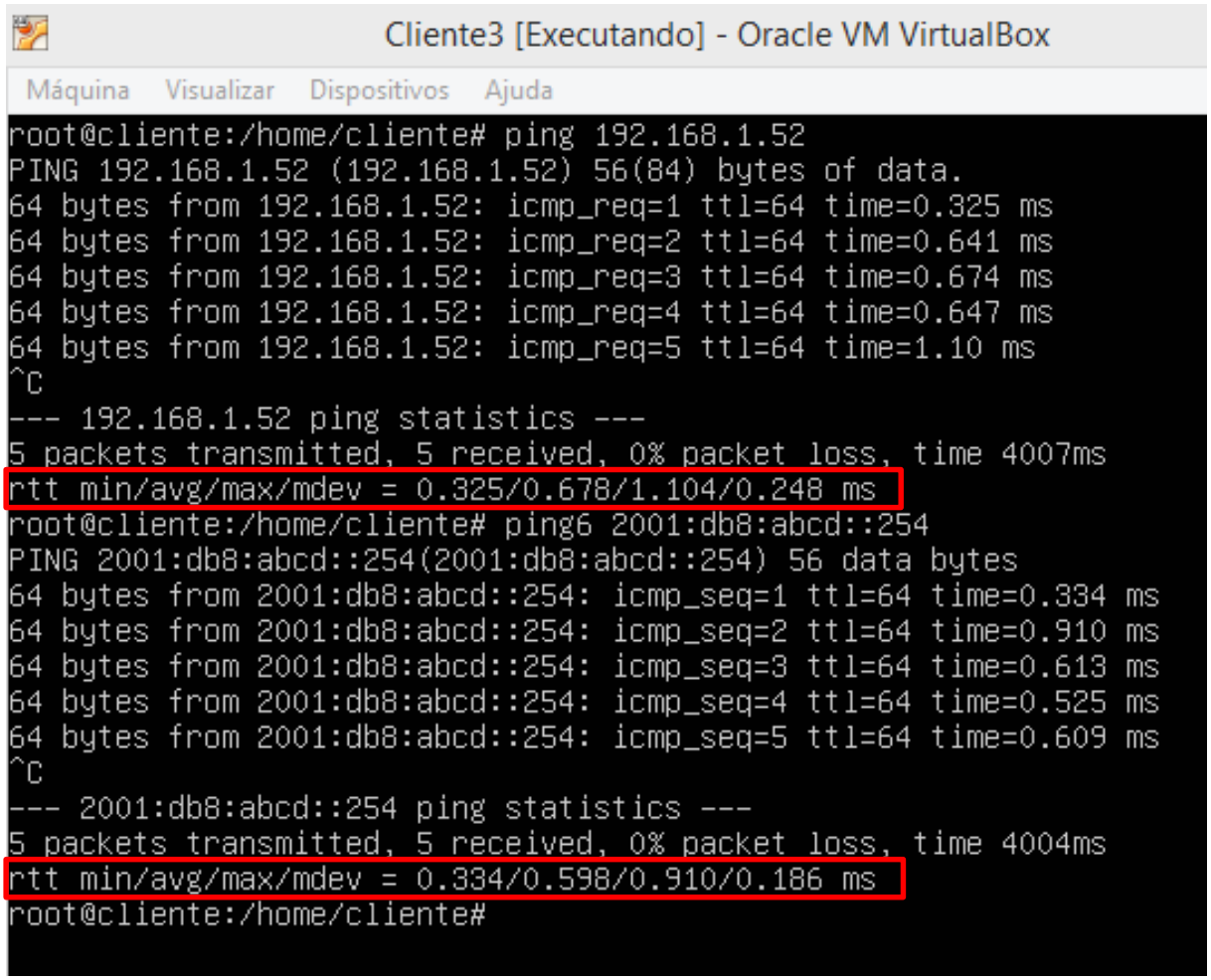


Figura 37: Teste de ping de um host pilha dupla para outro host pilha dupla.
Fonte: Do Autor.

Neste caso da Figura 37, analisando os resultados obtidos com *ping e ping6* e comparando-os, pode-se notar que o tempo médio de latência (*avg*) foi menor utilizando IPv6 e o *jitter* (*mdev*), também foi menor com IPv6. Isso significa que o tempo médio de atraso com IPv6 foi menor e a variação também. Deve-se observar que na Figura 36 e na Figura 37 foram transmitidos poucos pacotes, apenas 5.

No entanto, quando foi executado o *ping* e o *ping6* por um tempo maior, ou seja, ao transmitir um número maior de pacotes, houve um aumento da latência média utilizando-se IPv6 superando a latência média utilizando-se IPv4. Como pode-se observar na Figura 38 e na Figura 39, foram transmitidos 50 pacotes e, ainda que não seja muito, o desempenho do IPv4 foi melhor que do IPv6.

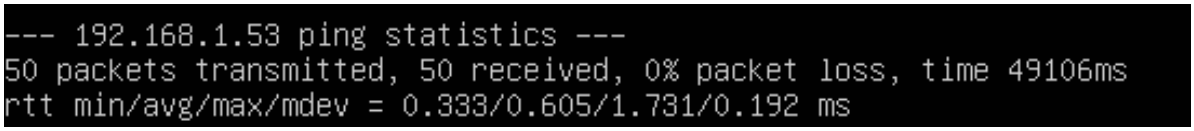


Figura 38: Estatísticas do teste de *ping* de um *host* pilha dupla para outro *host* pilha dupla.

Fonte: Do Autor.

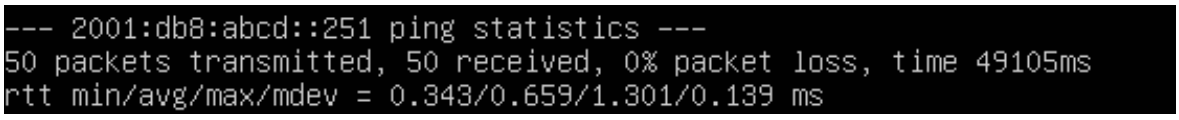


Figura 39: Estatísticas do teste de *ping6* de um *host* pilha dupla para outro *host* pilha dupla.

Fonte: Do Autor.

´ Nota-se que ainda é necessário um maior amadurecimento do protocolo IPv6 para que ele se torne efetivamente mais eficiente que o IPv4. O protocolo IPv4, por existir há mais tempo e por ser o mais amplamente utilizado possui uma infraestrutura mais consolidada.

Ao tentar comunicar os *hosts* da LAN com um sítio na *Internet*, pode-se perceber esta situação: com a pilha IPv4 foi possível acessar o sítio, porém com a pilha IPv6 não foi possível porque a maior parte dos provedores de acesso ainda não fornecem IPv6 nativamente para os usuários. Este problema pode ser contornado com a utilização de serviços gratuitos de túneis, chamados de *Tunnel Brokers.*

O funcionamento da técnica de Pilha Dupla é bastante simples, e esta é a mais recomendada para a transição, porém o que torna sua implantação mais complexa é o esgotamento de endereços IPv4 para serem atribuídos à novos clientes e as configurações de rotas, regras de *firewall* e registros de DNS que devem ser feitas de forma independente para cada protocolo. Principalmente as configurações de regras de *firewall*, devem ser feitas com cuidado para que não sejam deixadas brechas de segurança na rede.

5.2 Teste do Túnel *6over4*

Antes da implantação do túnel foi feito, a partir do Servidor da rede A, o comando *ping6 2001:db8:cdba::103* (endereço IPv6 do Servidor da rede B); e como pode-se observar na Figura 40, não foi possível a comunicação, pois a rede está inacessível. Isso ocorre devido a infraestrutura de rede que ainda é majoritariamente IPv4.

```
root@leticiaTcc:/home/leticia# ping6 2001:db8:cdba::103
connect: Network is unreachable
```

Figura 40: Teste ping6 do Servidor da rede A para o Servidor da rede B antes da implantação do túnel.
Fonte: Do autor.

Após a implantação do túnel *6over4,* foi realizado o *ping6* novamente e, como mostra a Figura 41, constatou-se que a comunicação entre as redes foi estabelecida.

```
root@leticiaTcc:/home/leticia# ping6 2001:db8:cdba::103
PING 2001:db8:cdba::103(2001:db8:cdba::103) 56 data bytes
64 bytes from 2001:db8:cdba::103: icmp_seq=1 ttl=64 time=0.703 ms
64 bytes from 2001:db8:cdba::103: icmp_seq=2 ttl=64 time=1.45 ms
64 bytes from 2001:db8:cdba::103: icmp_seq=3 ttl=64 time=0.946 ms
64 bytes from 2001:db8:cdba::103: icmp_seq=4 ttl=64 time=0.891 ms
64 bytes from 2001:db8:cdba::103: icmp_seq=5 ttl=64 time=1.90 ms
64 bytes from 2001:db8:cdba::103: icmp_seq=6 ttl=64 time=1.35 ms
^C
--- 2001:db8:cdba::103 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 0.703/1.209/1.909/0.409 ms
```

Figura 41: Teste ping6 do Servidor da rede A para o Servidor da rede B após a implantação do túnel.
Fonte: Do Autor.

Analisando os pacotes ICMPv6 trocados entre os servidores da Rede A e da Rede B no *Wireshark* (Figura 42), podemos notar que na camada *Internet Protocol*, o campo "*Protocol*" sinaliza que a mensagem encapsula um pacote IPv6, utilizando o protocolo 41 (6in4), logo abaixo aparecem os endereços IPv4 de origem e de destino e na camada *Internet Protocol Version 6* estão contidos os endereços IPv6 de origem e de destino.

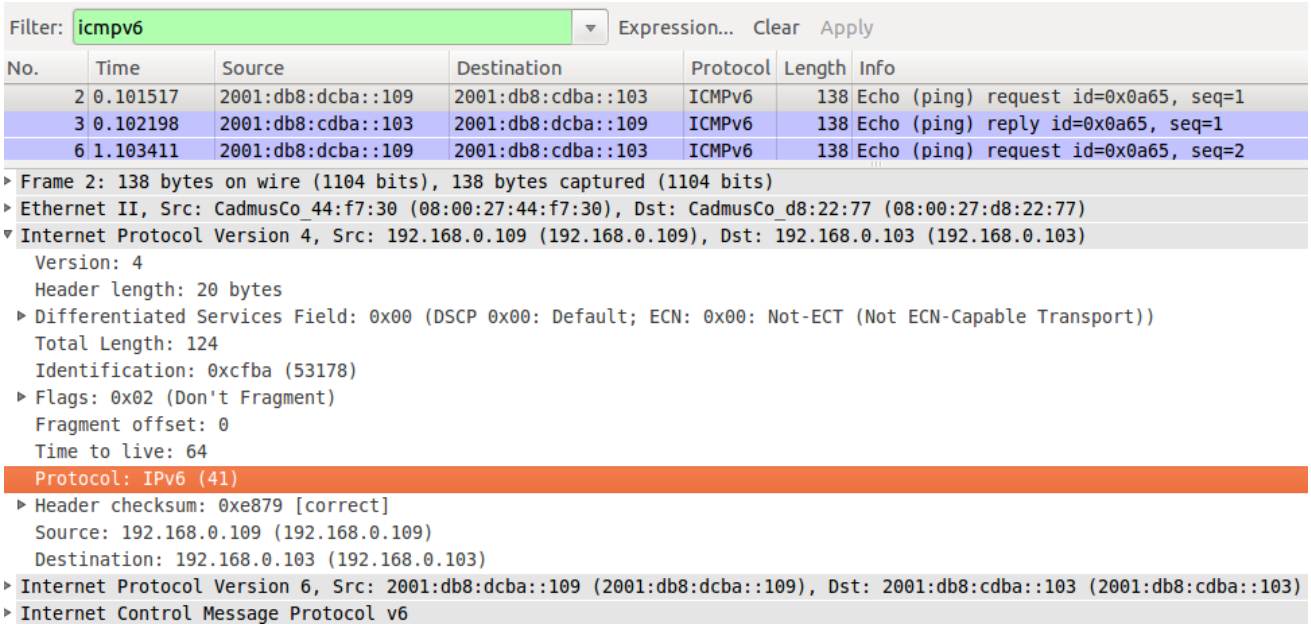


Figura 42: Mensagens ICMPv6 no *Wireshark*..
Fonte: Do Autor.

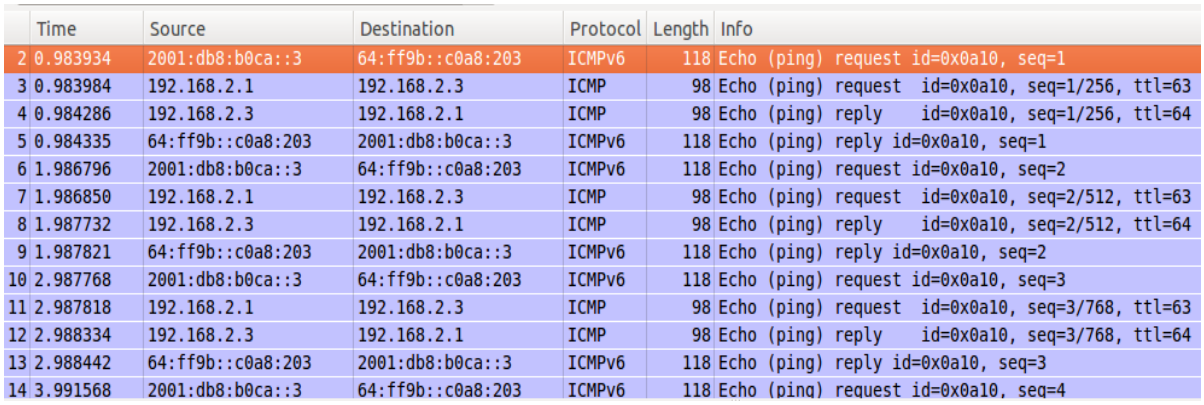
Constata-se então, que apesar da infraestrutura de rede ainda não suportar IPv6, o túnel *6over4* permite que os pacotes de uma rede IPv6 sejam transmitidos através da infraestrutura IPv4, pois encapsula os pacotes IPv6 em IPv4 para transferi-los de uma rede para outra. Este processo se dá de forma simples e para isso são necessários poucos comandos, a partir do momento que a infraestrutura de rede suportar IPv6, este túnel pode ser desabilitado facilmente.

Uma desvantagem é que este túnel só pode ser utilizado para comunicar uma rede IPv6 com outra rede IPv6 através de uma infraestrutura IPv4 não se aplicando para conectar usuários IPv6 à Internet IPv6. Deve-se observar também que devido ao processo de encapsulamento há um aumento na latência da transmissão.

5.3 Teste do NAT64/DNS64

Para simular o processo de tradução, foi realizado um teste de *ping* de um *host* puramente IPv6 [2001:db8:b0ca::3] para um *host* com suporte apenas àIPv4 [192.168.0.3]. Ao analisar as mensagens *Request e Reply no Wireshark* (Figura 43), pode-se perceber que o *host* IPv6 vê somente o endereço IPv6 traduzido do *host* IPv4 [64:ff9b::c0a8:203], enquanto o nó IPv4 só vê o endereço IPv4 do NAT64 [192.168.2.1]. Deve-se observar que no endereço

64:ff9b::c0a8:203, o “c0a8:203” equivale a 192.168.2.3 em hexadecimal, e esta conversão é realizada pelo NAT64. Portanto, esta técnica fica totalmente invisível ao usuário.



	Time	Source	Destination	Protocol	Length	Info
2	0.983934	2001:db8:b0ca::3	64:ff9b::c0a8:203	ICMPv6	118	Echo (ping) request id=0x0a10, seq=1
3	0.983984	192.168.2.1	192.168.2.3	ICMP	98	Echo (ping) request id=0x0a10, seq=1/256, ttl=63
4	0.984286	192.168.2.3	192.168.2.1	ICMP	98	Echo (ping) reply id=0x0a10, seq=1/256, ttl=64
5	0.984335	64:ff9b::c0a8:203	2001:db8:b0ca::3	ICMPv6	118	Echo (ping) reply id=0x0a10, seq=1
6	1.986796	2001:db8:b0ca::3	64:ff9b::c0a8:203	ICMPv6	118	Echo (ping) request id=0x0a10, seq=2
7	1.986850	192.168.2.1	192.168.2.3	ICMP	98	Echo (ping) request id=0x0a10, seq=2/512, ttl=63
8	1.987732	192.168.2.3	192.168.2.1	ICMP	98	Echo (ping) reply id=0x0a10, seq=2/512, ttl=64
9	1.987821	64:ff9b::c0a8:203	2001:db8:b0ca::3	ICMPv6	118	Echo (ping) reply id=0x0a10, seq=2
10	2.987768	2001:db8:b0ca::3	64:ff9b::c0a8:203	ICMPv6	118	Echo (ping) request id=0x0a10, seq=3
11	2.987818	192.168.2.1	192.168.2.3	ICMP	98	Echo (ping) request id=0x0a10, seq=3/768, ttl=63
12	2.988334	192.168.2.3	192.168.2.1	ICMP	98	Echo (ping) reply id=0x0a10, seq=3/768, ttl=64
13	2.988442	64:ff9b::c0a8:203	2001:db8:b0ca::3	ICMPv6	118	Echo (ping) reply id=0x0a10, seq=3
14	3.991568	2001:db8:b0ca::3	64:ff9b::c0a8:203	ICMPv6	118	Echo (ping) request id=0x0a10, seq=4

Figura 43: Análise das mensagens *request e reply* no Wireshark.
Fonte: Do Autor.

Em relação ao tempo de atraso, quando uma requisição é feita de um *host* IPv6 nativo para outro *host* IPv6 nativo o tempo médio de latência é bem menor, comparando com uma requisição feita de um *host* IPv6 nativo para um *host* IPv4 traduzido. Portanto, pode-se afirmar que a melhor alternativa é que todos os *hosts* possuam um endereço IPv6 nativo e quando isto não é possível, devido a incapacidade de um equipamento suportar IPv6, este deve ser substituído para haver uma melhor performance nas conexões de rede. Este fato pode ser observado na Figura 44, onde o primeiro teste de *ping* foi feito para um *host* IPv6 nativo [2001:db8:b0ca::1] e o segundo foi feito para um *host* IPv4 traduzido para IPv6 [64:ff9b::192.168.2.3]. Foi definida a mesma quantidade de pacotes a serem transmitidos (parâmetro -c 4).

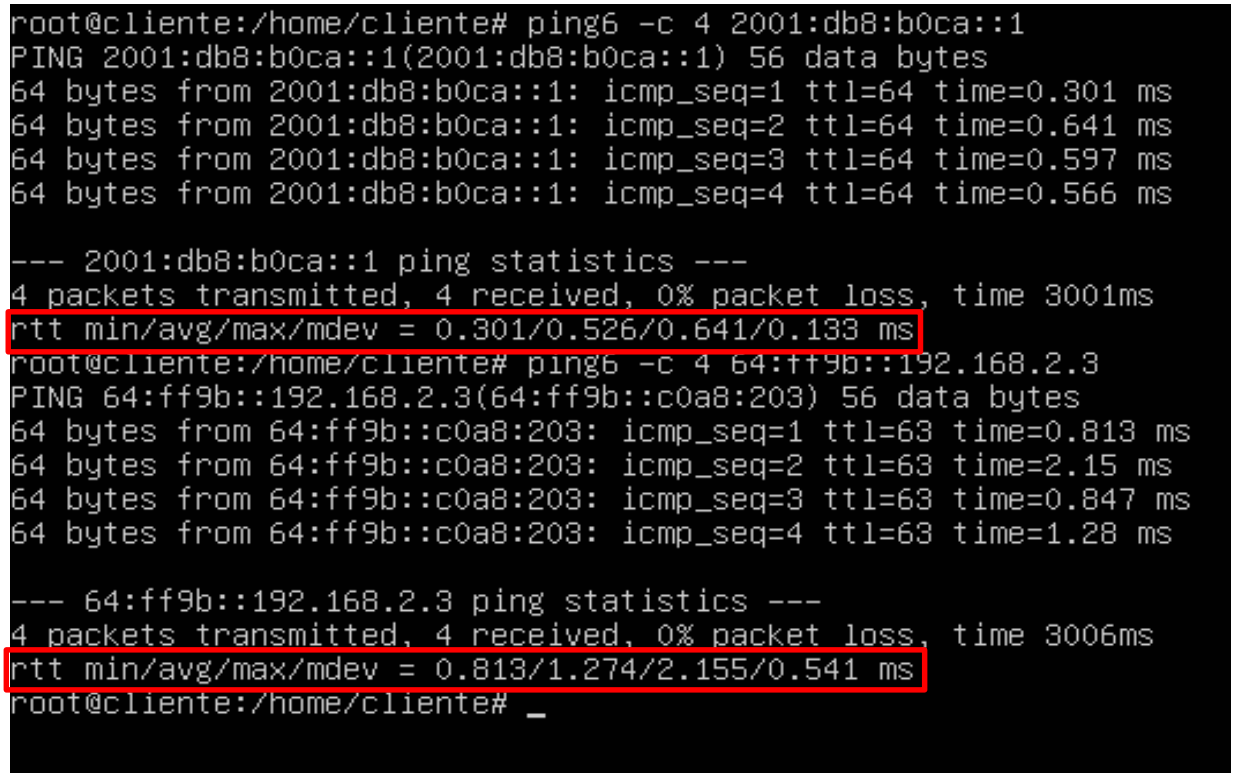


Figura 44: Teste de *ping* para IPv6 nativo e para IPv4 traduzido.
Fonte: Do Autor.

Esta técnica deve ser sempre a última alternativa a ser escolhida, pois o processo tradução causa a perda de muitos dados dos cabeçalhos que são traduzidos, aumenta a latência,a complexidade do núcleo da rede, o custo de processamento e quebra a conectividade fim-a-fim como qualquer NAT. Estes fatores prejudicam muito o desempenho e a segurança da rede.

O NAT64 deve ser usado somente quando as alternativas anteriores, como pilha dupla e túneis, não sejam possíveis de serem implantadas, em situações em que os dispositivos não possuam suporte ao protocolo IPv6 e quando os mesmos não podem ser substituídos facilmente. Esta técnica também pode ser uma alternativa para quem utiliza *softwares* legados que não possuam atualizações para o novo protocolo. Estas situações são dificilmente encontradas hoje em dia, pois segundo Pletsch (2012), a maior parte dos equipamentos possui suporte ao IPv6 ou atualizações de *software* que são oferecidas pelos próprios fabricantes para torná-los capacitados para IPv6.

6 CONCLUSÃO

Com a rapidez que *Internet* Global vem crescendo, é inevitável a transição para o novo protocolo e é necessário que as pessoas, principalmente os profissionais que atuam na área de redes tenham conhecimento sobre os benefícios de implantação deste protocolo e sobre as técnicas existentes para fazer a transição. Neste momento, em que a implantação ocorre de forma gradual, são necessários investimentos para adequações estruturais e para treinamentos dos profissionais da área de TI, para que os mesmos possam executar as técnicas adequadamente conforme a necessidade de cada ambiente de rede. Deve-se ter consciência que se estes investimentos não forem feitos, as empresas e provedores podem vir a perder muitos clientes, prejudicando seus negócios. Além disto, podem haver problemas para acessar determinados recursos na *Internet*, o desempenho e a segurança das redes podem ser comprometidos devido a utilização de NATs de grande porte, entre outros prejuízos. Com as grandes empresas e provedores de acesso implantando o novo protocolo, o acesso via IPv6 para os usuários finais chegará mais rapidamente, e futuramente, o protocolo IPv4 não será mais tão utilizado e cairá em desuso. É esta a intenção da criação do IPv6, porém sabe-se que este processo de transição ainda durará muito tempo.

Com a realização deste trabalho, pode-se obter um maior conhecimento e domínio das dos tipos de técnicas de transição, das diferenças que apresentam e da utilidade de cada para cada rede específica. Foram vivenciadas as dificuldades de implantação que são enfrentadas por muitos profissionais e pode-se observar as peculiaridades de cada técnica. A Pilha Dupla, por exemplo, se mostra muito simples no seu funcionamento e sua implantação pode ser feita gradualmente, porém por possuir os dois protocolos implantados nos dispositivos, é necessária uma maior atenção dos gerentes da rede que devem configurar IPv4 e IPv6 separadamente, ou seja, é necessário do dobro de trabalho para configurar as redes e maior preocupação com a segurança.

Os túneis, por sua vez, são ótimas alternativas quando a estrutura de rede não suporta algum dos protocolos, possibilitando contornar estas estruturas através do encapsulamento de pacotes e também para usuários que desejam conectar-se à *Internet* IPv6 e que os provedores de acesso ainda na provêm esta conectividade.

Já as técnicas de tradução, se tornam vantajosas em caso de sistemas legados que possuem alto custo e complexidade e que não podem ser atualizados para suportarem o IPv6.

Também são boas alternativas para redes IPv6 nativas que ainda necessitam acessar conteúdos na Internet IPv4.

É visto que nenhuma técnica é perfeita, porém analisando-as detalhadamente pode-se escolher a que mais se adequa às características da rede e que trará mais benefícios para as empresas e para os usuários.

REFERÊNCIAS

BRITO, S. H. B. **Governança da Internet no Mundo.** Disponível em:<http://labcisco.blogspot.com.br/2013/01/governanca-da-internet-no-mundo.html> Acesso em: 22 mai. 2015.

CISCO. **6lab - The place to monitor IPv6 adoption.** Disponível em: <http://6lab.cisco.com/stats/cible.php?country=BR&option=all>. Acesso em 21 jun. 2015.

COMER, D. E. **Redes de computadores e internet :** abrange transmissão de dados, ligação inter-redes, web e aplicações.4.Ed. Porto Alegre: Bookman, 2007.

COMER, D. E. **Interligação de Redes com TCP/IP:** princípios, protocolos e arquitetura.5.ed. São Paulo: Elsevier, 2006.

EQUIPE IPV6.BR. **Cronograma de Implantação.** Disponível em: <http://ipv6.br/cronograma/>. Acesso em: 10 mar. 2015.

EQUIPE IPV6.BR. **Transição.** Disponível em: <http://ipv6.br/entenda/transicao/>. Acesso em: 10 mar. 2015.

FOROUZAN, B. A. **Comunicação de dados e redes de computadores**. 4. Ed. São Paulo: McGraw-Hill, 2008.

IANA. ***Version Numbers.*** Disponível em: <http://www.iana.org/assignments/version-numbers/version-numbers.xhtml#version-numbers-1>. Acesso em: 11 jul. 2015

KUROSE., J. F. ; ROSS, K. W. **Redes de Computadores e a Internet :**Uma abordagem top-down . 5.ed. São Paulo: Pearson, 2010.

LACNIC. **Estatísticas de alocações do LACNIC.** Disponível em: <http://www.lacnic.net/pt/web/lacnic/estadisticas-asignacion>. Acesso em: 24 mai. 2015

MINISTÉRIO DO PLANEJAMENTO, ORÇAMENTO E GESTÃO; SECRETARIA DE LOGÍSTICA E TECNOLOGIA DA INFORMAÇÃO(Brasil). **Plano de Disseminação do Uso do IPv6.** Disponível em: <http://www.governoeletronico.gov.br/biblioteca/arquivos/plano-de-disseminacao-do-uso-ipv6/view>. Acesso em: 10 mar. 2015

MONEGO, R. P. **Implantação de uma rede utilizando os padrões do protocol IPv6.** Disponível em: <http://www.redes.ufsm.br/docs/tccs/Raissa_Monego.pdf>. Acesso em: 15 mar. 2015.

MOREIRAS, A. M.**1% dos usuários brasileiros com IPv6.** Disponível em**:** <http://ipv6.br/um-porcento-dos-usuarios-brasileiros-com-ipv6/>. Acesso em: 24 abr.2015

PLETSCH, V. **Transição para o protocol IPv6: Um estudo de caso aplicado a uma provedora de serviços de comunicação multimedia.** Disponível em:<http://www.inf.furb.br/~pericas/orientacoes/IPv62012.pdf> Acesso em: 15 mar. 2015.

RFC 1631.**The IP Network Address Translator (NAT).** Disponível em:<https://www.ietf.org/rfc/rfc1631.txt>. Acesso em: 15 mar. 2015.

RFC 2460.**Internet Protocol, Version 6 (IPv6) Specification.** Disponível em: <https://tools.ietf.org/html/rfc2460>. Acesso em: 12 mar 2015.

RFC 3315.**Dynamic Host ConfigurationProtocol for IPv6 (DHCPv6).**Disponível em: <http://www.rfc-base.org/txt/rfc-3315.txt>. Acesso em: 22 abr. 2015.

RFC 4213. **Basic Transition Mechanisms for IPv6 Hosts and Routers.** Disponível em: <https://tools.ietf.org/html/rfc4213>. Acesso em: 21abr. 2015.

RFC 4862. **IPv6 Stateless Address Autoconfiguration.** Disponível em: <https://tools.ietf.org/html/rfc4862>. Acesso em: 23 abr. 2015.

RFC 6146.**Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers.** Disponível em:<https://tools.ietf.org/html/rfc6146>. Acesso em: 1 jun. 2015.

RFC 6147.**DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers.** Disponível em: <https://tools.ietf.org/html/rfc6147>. Acesso em: 1 jun. 2015.

RFC 6052. **IPv6 Addressing of IPv4/IPv6 Translators.** Disponível: <https://tools.ietf.org/html/rfc6052>. Acesso em: 1 jun. 2015.

RUI, F. F.; Machado, G. S. **Autoconfiguração do protocolo IPv6.** Disponível em: <http://www.pucrs.br/research/salao/2006VIISalaoIC/Arquivos2006/CienciasExatasedaTerra/37364%20-%20GUILHERME%20SPERB%20MACHADO.pdf> Acesso em: 22abr. 2015.

SILVEIRA, A. M. **Redes IPv6 com Integração IPv4.** Disponível em: <http://wiki.sj.ifsc.edu.br/wiki/images/e/e3/TCC_AndreManoeldaSilveira.pdf>. Acesso em: 15 mar. 2015.

TANENBAUM, A. S. **Redes de Computadores.** 5. ed. Rio de Janeiro: Elsevier, 2003.

TANENBAUM, A. S.; WETHERALL, D. J. **Redes de Computadores.** 5. ed. Rio de Janeiro: Elsevier, 2011.

APÊNDICE A- Arquivo *dhcpd.conf*

```
# Sample configuration file for ISC dhcpd for Debian
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
option routers 192.168.1.1;
option domain-name-servers 192.168.0.1;
range 192.168.1.50 192.168.1.199;
}

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}
# This is a very basic subnet declaration.

#subnet 10.254.239.0 netmask 255.255.255.224 {
```

```
#  range 10.254.239.10 10.254.239.20;
#  option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
#}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
#subnet 10.5.5.0 netmask 255.255.255.224 {
#  range 10.5.5.26 10.5.5.30;
#  option domain-name-servers ns1.internal.example.org;
#  option domain-name "internal.example.org";
#  option routers 10.5.5.1;
#  option broadcast-address 10.5.5.31;
#  default-lease-time 600;
#  max-lease-time 7200;
#}

# Hosts which require special configuration options can be listed in

# host statements.   If no address is specified, the address will be

# allocated dynamically (if possible), but the host-specific information

# will still come from the host declaration.


#host passacaglia {

#  hardware ethernet 0:0:c0:5d:bd:95;

#  filename "vmunix.passacaglia";

#  server-name "toccata.fugue.com";

#}


# Fixed IP addresses can also be specified for hosts.   These addresses

# should not also be listed as being available for dynamic assignment.

# Hosts for which fixed IP addresses have been specified can boot using

# BOOTP or DHCP.   Hosts for which no fixed address is specified can only
```

```
# be booted with DHCP, unless there is an address range on the subnet
# to which a BOOTP client is connected which has the dynamic-bootp flag
# set.
#host fantasia {
#  hardware ethernet 08:00:07:26:c0:a5;
#  fixed-address fantasia.fugue.com;
#}
# You can declare a class of clients and then do address allocation
# based on that.   The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and all
# other clients get addresses on the 10.0.29/24 subnet.

#class "foo" {
#  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
#}

#shared-network 224-29 {
#  subnet 10.17.224.0 netmask 255.255.255.0 {
#    option routers rtr-224.example.org;
#  }
#  subnet 10.0.29.0 netmask 255.255.255.0 {
#    option routers rtr-29.example.org;
#  }
#  pool {
#    allow members of "foo";
#    range 10.17.224.10 10.17.224.250;
#  }
#  pool {
#    deny members of "foo";
#    range 10.0.29.10 10.0.29.230;
#  }
#}
```