Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Transição para o IPv6, uma implementação prática

Fernando Almeida Mayumi

JUIZ DE FORA
MARÇO, 2013

Transição para o IPv6, uma implementação prática

Fernando Almeida Mayumi

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Eduardo Pagani Júlio

JUIZ DE FORA

MARÇO, 2013

Transição para o IPv6, uma implementação prática

Fernando Almeida Mayumi

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Eduardo Pagani Júlio
Mestre em Computação/UFF

Francisco Henrique Cerdeira Ferreira
Mestre em Computação/UFJF

Alex Borges Vieira
Doutor em Computação/UFMG

JUIZ DE FORA

26 DE MARÇO, 2013

Resumo

O crescimento acelerado da internet e o expressivo aumento dos dispositivos que necessitam de um endereço global na rede exigem mudança do protocolo existente. O IPv6 foi criado com principal objetivo de suprir as necessidades de endereçamento e trazer novos recursos para as redes de computadores. A transição entre o antigo protocolo IPv4 para o IPv6 ocorre de forma lenta e gradual, de forma que os dois coexistirão no mercado por um período significativo. Esse trabalho demonstra a implantação do protocolo IPv6 na prática, abrangendo alguns dos principais serviços de rede utilizados na Universidade Federal de Juiz de Fora, e são demonstrados na prática. Será utilizado servidores DNS, web, dhcp, coexistência entre os dois protocolos, além de demonstrar as novas funcionalidades do IPv6.

Palavras-chave: IPv6, DNS, Serviços de redes.

Abstract

The fast growth of the Internet and the significant increase in devices that require a global address on the network require changing the existing protocol. IPv6 was created with the main objective to meet the needs of addressing and bringing new features to computer networks. The transition between the old IPv4 protocol to IPv6 occurs slowly and gradually, so that the two will coexist in the market for a significant period. This work demonstrates the implementation of IPv6 protocol in practice, covering some of the major network services used at the Federal University of Juiz de Fora, and are demonstrated in practice. It will be used DNS servers, web, dhcp, coexistence between the two protocols, and demonstrate the new features of IPv6.

Keywords: IPv6, DNS, Network Services.

Agradecimentos

Dedico este trabalho aos meus pais, exemplos de caráter, pelo amor incondicional e compreensão.

Ao meu orientador pelas idéias, paciência e apoio.

A todos meus amigos e familiares, pelo companheirismo e por sempre desejarem o meu bem.

"I know I was born and I know that I'll die, the in between is mine".

Pearl Jam (I Am Mine)

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

De acordo o Ibope Media, são 94,2 milhões de brasileiros conectados a Internet (Dezembro de 2012), sendo o Brasil o 5º mais conectado do mundo(ANTONIOLI, 2013).

Segundo pesquisa da *Cisco Visual Networking Index*, que prevê e analisa o crescimento de redes IP (*Internet Protocol*), o tráfego de Internet no Brasil alcançará 3,5 *exabytes* (ou 3,5 bilhões de gigabytes) por mês em 2016. Ainda segundo o estudo, no Brasil serão 617 milhões de aparelhos conectados até 2016 e o número de internautas no mundo deve crescer para 3,4 bilhões de pessoas, 45% da população mundial(CISCO, 2013).

O atual protocolo IPv4 põe em risco e pode limitar esse crescimento, já que foi projetado para atender aproximadamente 4 bilhões de endereços únicos e muito dessa faixa já tem endereços reservados para grandes empresas. Além disso tem-se o problema de tabelas de roteamento cada vez maiores e aplicações que necessitam de endereços fim-a-fim.

Em 1995, a IETF (*Internet Engineering Task Force*) desenvolveu o novo protocolo IPv6, que além do principal objetivo de resolver o problema da falta de endereços traz novas funcionalidades. O IPv6 tem novo formato de cabeçalho, endereçamento e encaminhamento eficiente e hierárquico, configuração de endereços com e sem estado, segurança incorporada (IPSEC), qualidade do serviço melhorada (QoS) e extensibilidade (TECHNET, 2013) .

O IPv4 e IPv6 são independentes entre e si e não são compatíveis, embora a coexistência entre os dois se dará por um bom tempo, já que os dois podem funcionar simultaneamente nos equipamentos, ao que chama-se de pilha dupla. No período de implantação será utilizada tecnologia auxiliares, conhecidas como técnicas de transição e inicialmente os cenários existentes serão estendidos para interoperabilidade entre os dois protocolos.

1.1 Objetivos

Este trabalho tem por objetivo demonstrar o funcionamento da rede IPv6 na prática, com a implantação de alguns dos principais serviços utilizados pelos servidores da Universidade Federal Juiz de Fora.

1.1.1 Objetivos específicos

Estudar o IPv6 e suas novas funcionalidades, montar um ambiente com o servidor DNS operando somente em IPv6 e em modo pilha dupla, mostrando as novas caraterísticas de segurança do protocolo (DNSSEC). Disponibilizar servidor web com respostas a IPv4 e IPv6 simultaneamente, demonstrar o funcionamento do DHCP na rede IPv6 e o novo modo de configuração e descoberta do *gateway*. Realizar testes no novo ambiente e avaliar o comportamento dos dois protocolos operando simultaneamente.

1.2 Justificativa

Através deste trabalho será demonstrado o funcionamento de alguns dos principais serviços IPv6 utilizados pelos servidores, que atualmente ainda operam em grande maioria em rede IPv4. Ele servirá de base para quem está em fase de transição para o IPv6, e através dos cenários montados introduzir as configurações mais utilizadas, demonstrar e evitar os erros mais proeminentes.

As empresas e instituições de ensino necessitam e precisam cada vez mais de serviços ligados a sua rede. Com a necessidade da transição para o novo protocolo e que ela continue operando de modo transparente, e facilitado para o usuário, será demonstrado como as novas configurações sem estado (*stateless*) vem pra ajudar e como o modo centralizado opera com o novo DHCPv6.

As resoluções de nome para navegação web, e-mail e outros, se mostram fundamentais através do DNS e as configurações de modo recursivo agilizam as respostas para o cliente, através do cache. A segurança, preocupação atual e eminente, será demonstrada pelo DNSSEC atuando com o IPv6.

No início do trabalho, os princípios de como opera o protocolo IPv6 e suas novas

funcionalidades foram pesquisados. Após isso, os ambientes montados, as ferramentas instaladas e os testes efetuados, de modo que se possa aproveitar a prática dos recursos estudados.

1.3 Divisão do Trabalho

Este trabalho será organizado da seguinte maneira. O Capítulo 2 contém o referencial teórico utilizado para o entendimento e explicação dos fundamentos do protocolo IPv6, contando com as referências bibliográficas para o melhor entendimento. No capítulo 3 estão as aplicações práticas, assim como as ferramentas utilizadas, resultados e erros encontrados. E por fim, no capítulo 4 estão as conclusões finais sobre o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta o material estudado do IPv6, e servirá de base para o entendimento do trabalho. São mostrados os detalhes de funcionamento, características e novas funcionalidades do protocolo, assim como a descrição dos serviços de rede que serão testados na prática e que utilizam o IPv6.

2.1 Protocolo

O protocolo é 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. As entidades utilizam protocolos com a finalidade de implementar suas definições de serviço, eles têm a liberdade de trocar seus protocolos, desde que não alterem o serviço visível para seus usuários (TANENBAUM, 2003).

Os protocolos de rede nasceram da necessidade de conectar equipamentos de fornecedores distintos, executando sistemas distintos, sem ter que escrever a cada caso programas específicos. Ambos os computadores devem estar configurados com os mesmos parâmetros e obedecer aos mesmos padrões para que a comunicação possa ser realizada sem erros. Existem diversos tipos de protocolos de rede, variando de acordo com o serviço a ser utilizado. De maneira geral há dois tipos de protocolos: Abertos e Proprietários. Os protocolos Abertos são a grande maioria dos utilizados na internet. Estes podem comunicar com outros protocolos que utilizam o mesmo padrão de protocolo. Um exemplo seria o protocolo TCP e o protocolo IP, pois ele pode comunicar com várias plataformas como Windows, Linux, Mac e outros (WIKIUNIVERSIDADE, 2013).

O protocolo tem várias funções, como (WIKIUNIVERSIDADE, 2013): detecção da conexão física, subjacente ou a existência de um nó, *handshaking* (estabelecimento de ligação), negociação de várias características de uma conexão, como iniciar e finalizar uma mensagem, como formatar uma mensagem, o que fazer com mensagens corrompidas ou mal formatadas, como detectar perda inesperada de conexão e o que fazer em seguida

e término de sessão ou conexão.

Os protocolos de comunicação via Internet são descritos nos documentos RFC da comunidade internacional IETF. O par TCP e IP são os principais e deram origem ao nome do conjunto de protocolos TCP/IP.

2.1.1 Protocolo de Internet

O protocolo de Internet (IP) é o principal na camada de rede e opera entregando os datagramas (pacotes) do *host* de origem ao *host* de destino com base em endereços IP. Este protocolo abstrai os protocolos das camadas superiores (transporte e aplicação) da rede em que se encontram, tratando várias redes interconectadas como apenas uma. Os datagramas do protocolo geralmente tem dois componentes, o cabeçalho e os dados úteis. O cabeçalho IP contém o endereço fonte, destino e outras informações necessárias para o roteamento e entrega do pacote. Os dados úteis representam o conteúdo que será transportado, e esse processo de juntar os dados úteis no pacote com o cabeçalho é chamado de encapsulamento (WIKILIVROS, 2013).

2.2 IPv6

Em meados de 1990, pesquisadores da área de redes começaram a questionar a viabilidade do número de endereços IPv4, visto a explosão na quantidade de usuários na rede global. Desenvolvida inicialmente para fins acadêmicos, para empresas de tecnologia de ponta e para o próprio governo, o protocolo IP não foi desenvolvido com pensamento de atender uma grande demanda. Com a explosão do uso por usuários domésticos e em todo o mercado, as pesquisas apontavam o crescimento de duas vezes o tamanho da Internet a cada nove meses (FREIRE, 1998).

O atual modelo IPv4 continua funcionando com *NAT*, que possibilita que vários hosts se comuniquem com o mundo externo com apenas um endereço único válido. Mas seu uso gera questionamento pois traz uma série de problemas como, acabar com o modelo de funcionamento fim-a-fim, trazendo complicações ou impedindo o funcionamento de uma série de aplicações, como por exemplo aplicações de voz sobre IP baseadas em SIP;

ele não escala bem, pois exige processamento pesado; ele não funciona com IPsec; ele funciona como um *stateful* firewall, dando uma falsa sensação de segurança a muitos administradores de rede e colaborando para a não adoção de boas práticas de segurança nas empresas; entre outros (IPV6ḂR, 2012).

Em 1990, com esses problemas no horizonte, a IETF começou a trabalhar em uma nova versão do IP, capaz de impedir que os endereços fossem esgotados e de resolver uma série de outros problemas, além de ser mais flexível e mais eficiente. Aqui estão alguns dos seus objetivos (TANENBAUM, 2003):

- Aceitar bilhões de *hosts*, de modo que a alocação de espaço seja suficiente;
- Reduzir o tamanho das tabelas de roteamento;
- Simplificar o protocolo, de modo a permitir que os roteadores processem os pacotes com mais rapidez;
- Oferecer mais segurança (autenticação e privacidade) do que o IP atual;
- Dar mais importância ao tipo de serviço, particularmente no caso de dados em tempo real;
- Permitir , possibilitando a especificação de escopos;
- Permitir que um *host* mude de lugar sem precisar mudar o endereço;
- Permitir que o protocolo evolua no futuro;
- Permitir a coexistência entre protocolos novos e antigos durante anos.

Os trabalhos de DEERING (1998), que contou com a ajuda de FRANCIS (1998), e com mais algumas modificações deram origem ao protocolo IPv6, bem aceito, o novo protocolo cumpre todos os objetivos propostos, enumerados anteriormente.

2.2.1 Cabeçalho IPv6

O cabeçalho IPv6 tem tamanho fixo de 40 *bytes*, e apesar dos endereços quatro vezes maior que o antigo protocolo ele tem apenas o dobro do tamanho. Isso se deve a flexibilidade e

eficiência dos cabeçalhos de extensão, além disso muitos campos foram removidos e outros renomeados.

O campo IHL (Tamanho do cabeçalho) não é mais necessário e foi removido, pois o IPv6 tem o comprimento de cabeçalho fixo, 40 bytes. Os campos *Identification* (Identificação), *Flags* e *Fragment Offset* (Deslocamento do Fragmento) também foram retirados e agora fazem parte do cabeçalho de extensão, o mesmo aconteceu com o campo *Header Checksum* (Soma de verificação) que foi retirado do novo protocolo e sua função de cálculo de CRC (verificação de redundância cíclica) e detecção nos erros de transmissão, são feitos pelas camadas superiores e até inferiores, como na camada de enlace pelo protocolo *ethernet*.

Com relação aos valores dos campos do cabeçalho, temos as seguintes mudanças:

- Campo *Version* (Versão) teve seu valor alterado de 4 para 6.
- O campo *Type of Service* (Tipo de Serviço) foi retirado para entrada do campos *Traffic Class* (Classe de Serviço) e *Flow Label* (Tipo de Fluxo) com objetivo de implementar novas funções de segurança de QoS (Qualidade de Serviço).
- *Payload Lenght* (Volume de Dados) substituiu *Total Length* (Tamanho Total) e o tamanho total do pacote é mensurado através dele.

Utiliza-se o campo TTL (*Tempo de Vida*) para limitar a vida de um datagrama em segundos na rede e esse já vinha sendo substituído pela contagem de saltos. Essa é a função original do campo *Hop Limit* (Limite de Saltos), em que é contabilizado a cada nó que o pacote passa na rede. O campo *Protocol* indica qual protocolo na camada superior (por exemplo TCP, UDP e ICMP) , receberá os dados incluídos no datagrama IP. Esse foi substituído pelo *Next Reader*, que pode continuar invocando um protocolo superior ou chamar um cabeçalho de extensão.

Os campos *Source Address* (Endereços de Origem) e *Destination Address* (Endereço de Destino) utilizam endereços de 32 bits no IPv4 e teve seu tamanho alterado para 128 bits no IPv6, as diferenças entre os dois cabeçalhos são demonstrados na Figura 2.1.

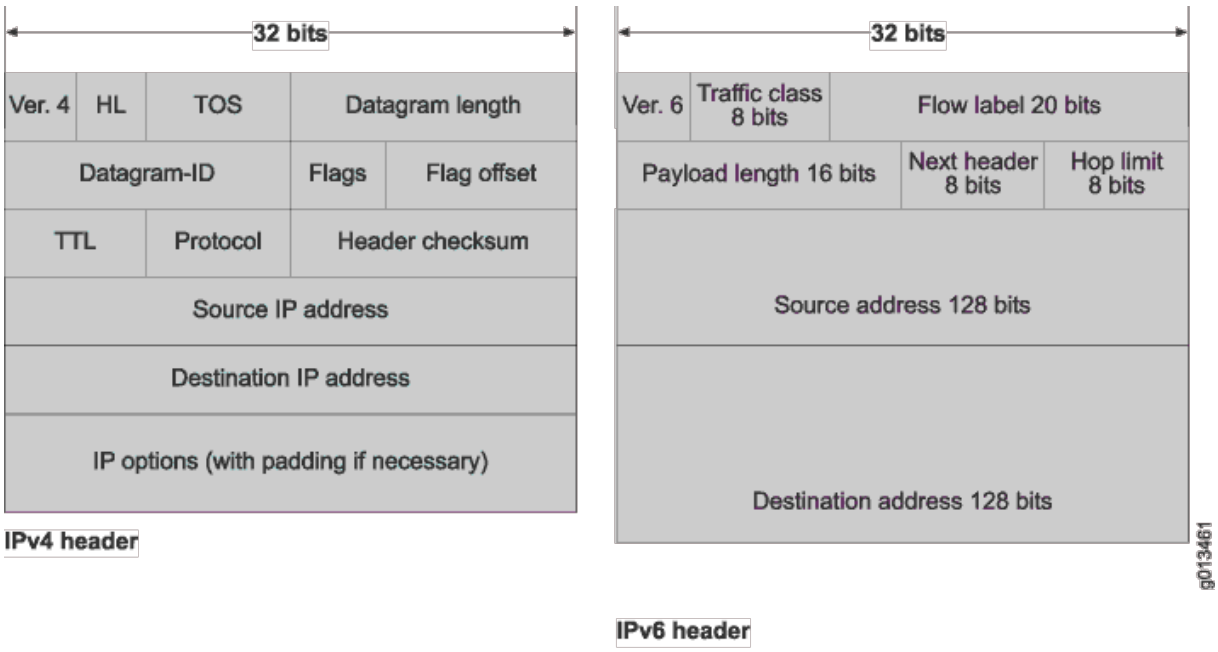

Figura 2.1: Cabeçalhos IPv4 e IPv6.

Fonte: http://www.juniper.net/techpubs/images/g013461.gif

2.2.1.1 Cabeçalhos de Extensão

O IPv6 trata as informações adicionais em seu cabeçalho diferentemente do protocolo IPv4, em que tudo estava incluso no cabeçalho base. As funcionalidades extras agora fazem parte dos cabeçalhos de extensão, que não possuem mais tamanho fixo e nem limite de quantidade. Esses estão localizados entre o cabeçalho base e o cabeçalho da camada imediatamente acima, podendo ser adicionados ilimitadamente e em série.

Os principais cabeçalhos de extensão são (IPV6ḂR, 2012):

- *Hop-by-Hop*: Contém opções que carregam informações para todos os dipositivos roteáveis entre a fonte e o destino. Ele deve ser colocado imediadamente após o cabeçalho base, é identificado pelo valor 00 no campo Próximo Cabeçalho;

- *Destination Options*: Utilizado na mobilidade IPv6, contém a opção *Home Address*, que contém o endereço de origem do nó móvel quando este está em transito. Deve ser processado apenas pelo nó de destino do pacote, é identificado pelo valor 60 no Próximo Cabeçalho;

- *Routing*: Define um método que permite que o dispositivo fonte especifique as rotas para um datagrama, esse cabeçalho permite a definição de múltiplos tipos de rotas.

Sua definição foi para auxiliar na mobilidade do IPv6, e deve carregar o endereço de origem do nó móvel em pacotes enviados pelo nó correspondente. Identificado pelo valor 43 no campo *Next Header*;

- *Fragmentation*: Quando o datagrama contém apenas um fragmento da mensagem original, esse cabeçalho é incluso. Ele contém o *Fragment Offset, Identification*, e mais fragmentos que foram removidos do cabeçalho principal. É necessário quando o pacote IPv6 a ser enviado é maior que o Path MTU, que representa a unidade máxima de transmissão, e refere-se ao tamanho do maior datagrama que uma camada de um protocolo de comunicação pode transmitir. É identificado pelo valor 44 no campo *Next Header*;

- *Authentication Header*: Utilizado pelo *IPsec*, contém informações usadas para verificar a autenticidade dos dados criptografados. Identificado pelo valor 51 no campo *Next Header*;

- *Encapsulating Security Payload*: Também utilizado pelo *IPSec*, garante a integridade e confidencialidade dos pacotes. Identificado pelo valor 52 no campo *Next Header*.

2.2.2 Endereçamento IPv6

O motivo principal da mudança do protocolo IPv4 para o IPv6 foi o esgotamento de endereços. Enquanto o IPv4 possui endereços de 32 bits, possibilitando 4.294.967.296 endereços únicos. O IPv6 tem espaço de endereçamento de 128 bits, possibilitando uma quantidade de 340.282.366.920.938.463.463.374.607.431.768.211.456 endereços únicos.

Os endereços passam a ser representados por números hexadecimais de 16 bits, separados por ":". É indiferente representar as letras com maiúsculas ou minúsculas, e algumas abreviações são possíveis, como a omissão de zeros à esquerda e a representação de um conjunto contínuo de zeros por "::". Como exemplo, temos:

- Zeros à esquerda em cada duocteto podem ser omitidos; Assim, 2001:12f0:0614:0000:0000:0000:0000:0001 pode ser representado por 2001:12f0:614:0:0:0:0:1;

- Além disso, é permitido substituir uma de sequencia zeros por "::", apenas uma vez no endereço, evitando-se assim a redundância na hora da representação do mesmo. Assim no exemplo utilizado, teríamos 2001:12f0:0614:0000:0000:0000:0000:0001 mais facilmente representado por 2001:12f0:614::1.

Para os prefixos de rede, elas são representadas como no CIDR (*Classless Inter-Domain Routing*), utilizado no IPv4, utilizando "/", seguida do número de bits representativos da sub-rede. Onde depois de "/" vem um valor decimal que especifica a quantidade de bits contíguos à esquerda do endereço que compreendem o prefixo. Como exemplo de prefixo de sub-rede temos (IPV6ḂR, 2012):

- Prefixo 2001:db8:3003:2::/64;

- Prefixo global 2001:db8::/32.

Esta representação também possibilita a agregação dos endereços de forma hierárquica, identificando a topologia da rede através de parâmetros como posição geográfica, provedor de acesso, identificação da rede, divisão da sub-rede, etc. Com isso, é possível diminuir o tamanho da tabela de roteamento e agilizar o encaminhamento dos pacotes.

Em relação a máscara de sub-rede, ela geralmente não é mais informada para fazer a operação de AND binário, como ocorria no IPv4, a notação de bit *COUNT*, em que a definição da sub-rede é feita por "/" seguido por um numeral, foi mantida (IPV6ḂR, 2012).

Para a representação de endereços IPv6 em URL *(Uniform Resource Locators)*, colocam-se os números do endereço envoltos por colchetes para diferenciação dos endereços IPv4. Exemplos desses, com e sem definição do número da porta:

- http://[2001:12f0:614::22]/home.html;

- http://[2001:12f0:614::22]:8080.

Quanto a faixa de endereços, o IPv6 possui endereços especiais, como representados na seguinte tabela (IPV6ḂR, 2012):

Tabela 2.1: Faixas de endereço IPv6.

::0	Endereço não-especificado
::1	Endereço loopback
::FFFF:wxyz	Endereço mapeado de um IPv4
2001(...)::/16	Endereço Global
FE80::/10	Endereços de *Link-Local*
FEC0::/10	Endereços de Site-Local
FF00::/8	Endereços *Multicast* - Toda a internet

2.2.3 Classificação dos endereços IPv6

O protocolo IPv6 possui três tipos de endereços: *unicast*, *anycast* e *multicast*.

2.2.3.1 Unicast

Tipo de endereço que identifica somente uma interface, de forma única e exclusiva, de modo que seja possível a comunicação entre dois nós. Ele se beneficia da grande variedade de endereços oferecidos pelo IPv6 e viabiliza que todos os *hosts* existentes no planeta possam estabelecer uma conexão fim-a-fim. Ideal para redes privadas virtuais, serviços de voz sobre IPv6, entre outros.

- **Endereços Globais** - São aqueles que são visíveis na Internet, assemelham-se aos IPs públicos do IPv4. Foram alocados apenas 13% dos possíveis endereços pela IANA. A faixa liberada até o momento engloba apenas um oitavo do total de endereços, na faixa 2000::/3.

- **Endereços *Link Local*** - Requerido por todas as *interfaces* no IPv6, é criado com base nos últimos 64 bits do endereço MAC da interface. É utilizado para identificar apenas um *host* dentro do enlace específico onde ele está conectado. O prefixo fe80::/64 foi designado para a tarefa.

- **Endereços *Unique Local*** - Criado com o intuito de designar endereços locais para a rede, não visíveis ao mundo exterior, semelhante ao utilizado no IPv4 para redes internas. Seu bloco de endereços é fc00::/7.

2.2.3.2 Multicast

Identifica um grupo de *interfaces*, um pacote enviado a um endereço *multicast* é entregue a todas as interfaces associadas a esse endereço. Funcionamento semelhante ao *broadcast* (extinto no IPv6), ele se diferem no fato de que no *multicast* o pacote é enviado apenas a um grupo de *hosts* e no *broadcast*, todos os *hosts* recebem o pacote, sem filtragem no conjunto de *interfaces*.

Endereços *multicast* são identificados pelo prefixo FF e seu bloco pertence a ff00::/8. Endereços *multicast* são geralmente formado por grupos de 4 bits. O campo de endereço *flag* é definido apenas pelo 3º e 4º bit, os demais foram reservados para uso futuro. Os outros três bits são conhecidos como R, P e T, explicados na Tabela 2.2 a seguir:

Tabela 2.2: Representação dos bits R,P e T.

R	R=0, Não carrega o endereço de um ponto de encontro R=1, Carrega o endereço de um ponto de encontro
P	P=0, Endereço *multicast* não baseado em prefixo de rede P=1, Endereço *multicast* baseado no prefixo de rede
T	T=0, Endereço *multicast* não é permanente(dinãmico T=1, Endereço *multicast* é permanente(atribuído pela IANA)

Os bits do campo escopo do endereço *multicast* são representados por 4 bits e tem a função de limitar a abrangência de cada grupo.

Tabela 2.3: Abrangência dos valores do campo escopo.

Valor do Campo Escopo	Escopo
1	Interface Local
2	Nós de um enlace
5	Nós de um site
8	Organização Local
E	Global - Toda a internet

2.2.3.3 *Anycast*

Os endereços *anycast* são atribuídos a mais de uma *interface*, com a propriedade que um pacote enviado a um endereço *anycast* é roteado para a *interface* mais próxima que tem esse endereço, de acordo com a medida de distância do protocolo de roteamento. Os serviços UDP são os principais serviços que o utilizam, principalmente DNS.

Além disso, endereços *anycast* não podem ser utilizados como endereço de origem de qualquer pacote IPv6. E também não pode ser configurado em um *host*, são exclusivos para associação em roteadores.

2.2.4 Segurança no IPv6

Desde o seu surgimento, o IPv4 foi criado sem focar a segurança no protocolo, não por negligência mas por ter sido projetado para atender ambientes menores ao que a realidade da Internet se encontra atualmente.

A utilidade e a vontade de implementar características de segurança e autenticação na camada de rede é discutida por anos e desde as origens do desenvolvimento do protocolo IPv6, foi pensado novas funcionalidades para o mesmo. O protocolo de segurança IP(IPSEC) é o método padrão que visa oferecer autenticação, segurança, incluindo transmissão segura de senha, encriptação e assinatura digital são implementados através do cabeçalho de autenticação(AH) e do encapsulamento de segurança de carga útil (ESP).

O IPSec é de uso obrigatório e integrado ao IPv6. É uma extensão do protocolo IP e uma combinação de variadas técnicas que foram criadas para oferecer uma melhor segurança. A criptografia e autenticação são feitas na camada de rede e portanto oferece segurança fim-a-fim. São possíveis dois modos de utilização, modo de transporte e modo de tunelamento, descritos a seguir.

2.2.4.1 Modo de Transporte

O modo de transporte do IPSec protege os protocolos de camadas superiores e é utilizado para segurança fim-a-fim, pois o *host* que origina o pacote também é capaz de verificar a segurança, tanto decriptando o pacote ou certificando a autenticação. Nesse modo apenas apenas o segmento da camada de transporte é processado, como se pode ver na Figura 2.2, o roteamento fica inalterado, desde que o cabeçalho IP não seja modificado.

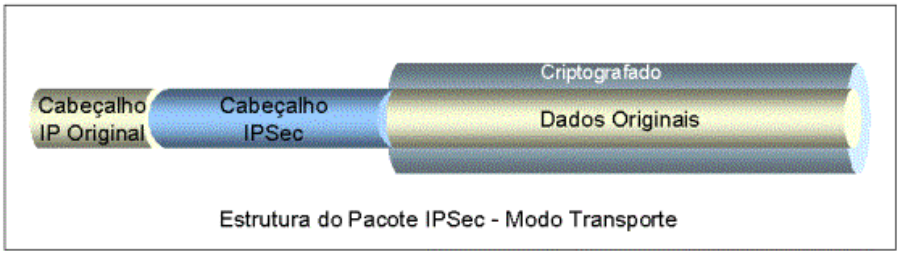

Figura 2.2: Pacote IPv6 protegido pelo modo túnel.

Fonte:UNICAMP.BR (2009)

2.2.4.2 Modo de Tunelamento

Neste modo, todo o conteúdo interno fica criptografado e apenas a parte do cabeçalho com o endereço de destino e de origem fica visível, informando o destino do *gateway*. O cabeçalho IPSec é colocado na frente do cabeçalho IP, é necessário portanto um novo pacote IP para fazer sua distribuição, como ilustrado na Figura 2.3. Esse modo é particularmente ´útil quando o túnel não inicia e termina nos sistemas finais, como por exemplo entre *firewalls*.

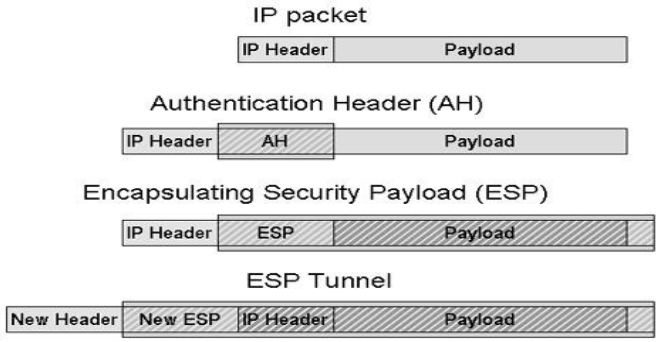

Figura 2.3: Pacote Ipv6 com os modos *AH*, *ESP* e novo pacote transportando túnel *ESP*.

Fonte: http://i.technet.microsoft.com/dynimg/IC197162.gif

2.3 Planejamento e Transição Para o IPv6

A implementação do IPv6 traz uma nova realidade para os profissionais da área de redes e remete a novas preocupações em relação a tempo, custos e riscos de implementar a nova tecnologia. Para se ter a rede funcionando toda em modo IPv6 teríamos que atualizar todos os *hosts* de uma rede mais os equipamentos que ainda não suportam a nova tecnologia. Algo inviável para empresas com inúmeros servidores e estações.

Portanto a transição deve vir de forma lenta e gradual acompanhando a realidade financeira e da necessidade gradual de cada empresa. Para ajudar na tarefa, já existem existem vários métodos e estratégias para facilitar a coexistência dos dois protocolos.

Ainda tem-se no mundo a escassez de provedores que oferecem o serviço, dificultando a propagação e a instalação dos novos ambientes. O protocolo foi criado há mais de

quinze anos e era esperado uma transição mais rápida entre os dois. Com a coexisteência entre eles temos que conviver com a necessidade da pilha dupla(*Dual Stack*), túneis ou tradução entre os dois (FLORENTINO, 2012).

2.3.1 Pilha Dupla

A pilha dupla traz a conveniência do funcionamento entre ambos os protocolos, nos mesmos equipamentos nativamente, conforme Figura 2.3. Apesar disso, tem-se vários aspectos negativos com a pilha dupla em atividade, como (FLORENTINO, 2012): Dois planos de endereçamento, duas gerências, duas tabelas de roteamento distintas, duas resoluções de problemas (pois o fato de uma pilha estar funcionando corretamente não implica que a outra esteja).

Suas vantagens são poder dividir a rede em conglomerados menores em que, nos seus nós será oferecido serviços de DHCP, DNS, entre outros com configurações para ambos os endereços, que trabalharão independentemente em todos os equipamentos. Dividindo assim, a rede em regiões, poderemos aos poucos ir desligando os equipamentos que possuem o antigo protocolo em funcionamento a medida que necessário.

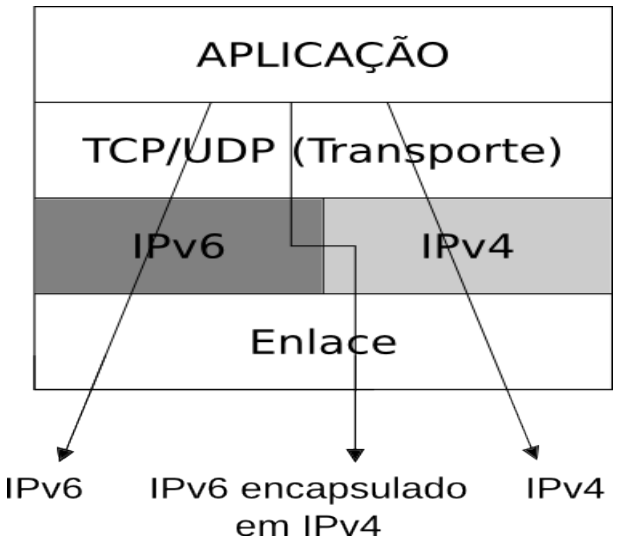

Figura 2.4: Representação do funcionamento da Pilha Dupla.

Fonte: http://www.teleco.com.br/imagens/tutoriais/tutorialredeip2_figura03.gif

2.3.2 Túneis

Os túneis são necessários quando precisamos que o protocolo de rede encapsule o datagrama de um protocolo diferente. Para a transição do IPv6 para o IPv4 tem-se várias técnicas diferentes, como *Tunnel Brokers*, *NAT64 e DNS64*.

2.3.2.1 *Tunnel Brokers*

Oferecidos pelos provedores de acesso, os *Tunnel Brokers* oferecem conectividade IPv6 aos usuários finais que possuem somente conexão IPv4, conforme ilustra a Figura 2.4. É recomendado para usuários desejam testar a nova tecnologia em lugares que os provedores ainda não disponibilizam acesso nativo ao IPv6.

Possui a desvantagem de deixar o acesso lento devido ao fato procurar um servidor público na internet para criar um túnel. Fala-se em equivalência de estar conectado através de uma *VPN*. Por isso a importância de buscar um servidor que esteja mais próxima da sua localidade. Exemplos de serviços que oferecem túneis são a *Hurricane Eletric* e a *SIXXS*, última será utilizada no presente trabalho (FLORENTINO, 2012).

Os *Tunnel Brokers* podem usar tecnologias diversas para prover os túneis. Podem usar, por exemplo, túneis *6in4*, encapsulamento em UDP, o protocolo AYIYA (que significa *Anything in Anything*), ou TSP (*Tunnel Setup Protocol*), definido na RFC 5572 (IPV6ḂR, 2012).

Os passos básicos para estabelecimento da conexão pelo túnel estão descritos a seguir (IPV6ḂR, 2012).

- Cliente pilha dupla solicita túnel (pode ser solicitada autenticação) via IPv4;
- *Broker* cadastra usuário no Servidor de túnel;
- *Broker* informa cliente parâmetros para criação do túnel;
- Túnel estabelecido.

A figura 2.5 ilustra o cenário:

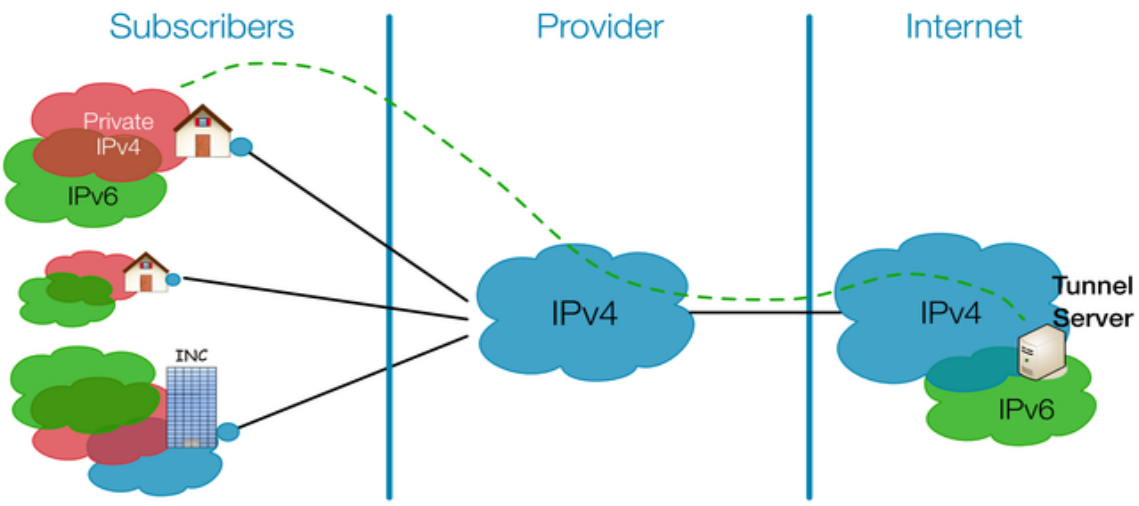

Figura 2.5: Demonstração básica do funcionamento do *Tunnel Broker*

Fonte: http://www.ipv6actnow.org/wp-content/uploads/2012/02/6in4.png

2.3.2.2 Tradução - NAT64 e DNS64

O NAT64 foi criado com o intuito de facilitar quem já possui o protocolo IPv6 de acessar a rede IPv4. O caso já é valido quando se tem pelo menos duas interfaces, uma de cada protocolo e conectada a sua respectiva rede. A tradução não é simétrica, já que o espaço do endereço IPv6 é muito maior que o do IPv4.

Quando uma máquina somente IPv6 tenta acessar uma rede IPv4 por uma pesquisa DNS em um registro AAAA, o DNS64 responde a solicitação mascarando o prefixo IPv4 com um prefixo qualquer. O *host* IPv6 envia o pacote para o *host* IPv4, que por sua vez é traduzido pelo NAT64 (tradução do tipo de pacote) (FLORENTINO, 2012).

Existem dois casos de tradução por NAT64, a configuração *Stateless* e a com estado (*Statefull*), a primeira é uma tradução 1 para 1, não conserva o endereço IPv4, requer atribuição de endereços IPv4 traduzíveis para IPv6, feito manualmente por uso de DHCPv6. A configuração *Statefull*, como mostrado na Figura 2.6, é uma tradução 1

para N e conserva o endereço IPv4, não requer alguma exigência em relação a atribuição de endereços IPv6, e é livre na hora da escolha do modo de atribuir os endereços.

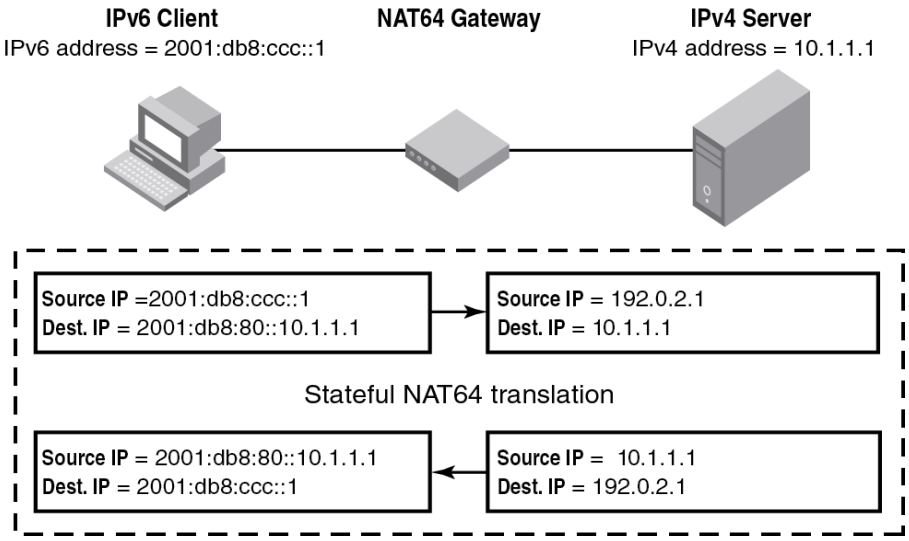

Figura 2.6: Demonstração da tradução por NAT64.

Fonte: https://labs.ripe.net/Members/raimis/NAT64DNS64image.png

2.4 Funcionalidades e novidades do IPv6

A substituição do antigo protocolo, apesar de movida pelo principal motivo de espaço de endereçamento, também visa trazer novos benefícios para os usuários e administradores de redes. As principais novidades vem acompanhadas do protocolo ICMPv6, que exerce papel fundamental no funcionamento do mesmo. E são elas, a configuração de endereço (*Stateless*, novas características como a mobilidade IPv6, suporte melhorado na qualidade de serviço(QoS), e suporte *multicast*, como visto anteriormente.

2.4.1 ICMPv6

O protocolo ICMPv6 é obrigatório na camada de rede na arquitetura TCP/IP, e é fundamental para a arquitetura, já que a mesma depende de muitas de suas funções. Originalmente criado para reportar alguns erros pequenos, hoje as mensagens ICMP são usadas para reportar diversos erros e mensagens de retorno.

Equipamentos como *gateways*, ao receber datagramas que provoquem algum erro, devem enviar mensagens ICMP reportando o que ocorreu. Exemplos do que pode ocorrer:

Tabela 2.4: Tipos de mensagens IMCPv6.

Valor do Tipo	Nome da mensagem	Descrição
1	Destino Inacessível	Indica que o datagrama não pode ser entregue ao seu destino.
2	Pacote excedeu tamanho	Ocorre quando um datagrama é muito grande para o permitido pela transmissão máxima de unidade(MTU), exclusivo para IPv6 que não podem fragmentar mensagens com tamanho excedido.
3	Tempo excedido	Enviado quando o tempo limite do salto do pacote é excedido.
4	Problema de parâmetro	Indica alguns erros diversos, como erros no campo do cabeçalho, próximo campo do cabeçalho e opção IPv6 irreconhecíveis.

O protocolo ICMPv6 permite funções importantes como *ping*, descoberta de vizinhança e atribuição de endereços *stateless*, pela descoberta de roteadores e *gateways* na rede IPv6. O ICMPv6 incorporou as funções dos protocolos ARP, RARP e IGMP. As mensagens de informação são vistas a seguir:

Tabela 2.5: Mensagens de informação ICMPv6.

Valor do Tipo	Nome da mensagem	Descrição
128	Requisição *Echo*	Enviado pelos dispositivos para testar a conectividade.
129	Resposta ao *Echo*	Enviado em resposta a requisição *Echo*.
133	Solicitação do Roteador	Enviado por uma estação que solicita as configurações de um roteador na subrede. Permite fechar uma conexão sem a necessidade do DHCPv6.
134	Anúncio do Roteador	Enviado pelos roteadores com o objetivo de anunciar aos *hosts* da rede local que ele está disponível e quais as suas características e configurações.
135	Solicitação de vizinhança	Enviado por um *host* quando deseja encontrar o endereço MAC de um endereço IPv6, se assemelha ao ARP do IPv4, ou com o objetivo de averiguar a unicidade de um endereço atribuído ao próprio anteriormente ao *host* de origem.(*Discover address Duplicate - DAD*).
136	Anúncio de vizinhança	Enviado pelos *host*s para informar a mudança de endereço ou em resposta a requisição da solicitação de vizinhança.

Para que se possa ter a exata noção de sua importância, se deixarmos o *firewall* das estações de trabalho bloquearem toda e qualquer mensagem ICMPv6, a rede simples-

mente irá parar (FLORENTINO, 2012), pois foram incorporadas funções através de suas mensagens, vitais ao funcionamento do protocolo.

2.4.2 Configuração *Stateless*

A configuração sem estado é permitida devido as novas funcionalidades que o protocolo IPv6 oferece, como os endereços de *link*-local, descoberta de vizinhança e a capacidade de gerar o identificador de interface pela camada de enlace.

A idéia geral do processo é gerar um endereço temporário até receber as características da rede e criar o endereço definitivo através dessas informações. Os passos básicos para um *host* adquirir um endereço pelo modo *Stateless* (PEPELNJAK, 2011):

1. O *host* primeiramente gera um endereço de link-local, com o prefixo FE80::/64 combinado com derivação do endereço MAC;

2. O nó testa se o endereço anteriormente gerado não está previamente usado na rede, usando a descoberta de vizinhança e pela resposta do anúncio de vizinhança;

3. Se passar no teste de unicidade, o endereço de link-local é atribuído para a interface e pode ser utilizado para comunicação local apenas;

4. No próximo passo, o *host* tenta estabelecer conexão com o roteador para continuar a configuração, isso ocorre através das mensagens de solicitação de roteador ou recebendo o anúncio;

5. O roteador responde como proceder com a auto-configuração, sendo esta direcionada para uma configuração com estado através de um servidor DHCPv6 ou informar para o *host* como conseguir seu IP global de outra maneira;

6. Ao final, assumindo que a configuração sem estado foi utilizada, o *host* irá ter sido configurado com um endereço global, geralmente formado pelo prefixo informado pelo *host* combinado com o identificador gerado no primeiro passo.

2.4.3 Mobilidade IPv6

No IPv4 Móvel, a transmissão de pacotes de dados baseia-se geralmente num roteamento triangular, onde os pacotes são enviados a um servidor proxy antes de chegar ao seu destino final e gerava muitos problemas como a sobrecarga da rede, atraso na entrega de pacotes e maior probabilidade de perda de pacotes. O IPv6 trouxe novas funcionalidades afim de evitar essas situações, como a notificação confiável e atualizada de IPs temporários.

A mobilidade IPv6 (MIPv6) tem o propósito de permitir que um dispositivo movimente-se por uma ou mais redes, sem perder a conexão com sua rede de origem, e mantenha seu IP original. Para isso a conexão entre eles deve ser intermediada por *hosts* que ainda podem ter origem IPv4, usando para isso o protocolo de IP móvel (MALKI, 2003). Os principais elementos que compõem a mobilidade IPv6 são:

- Agente móvel: Nó que faz parte da rede original e pode mudar de subrede e assim mesmo pertencer a sua rede original ou local(*Home network*);
- Agente local: Roteador que mantém os dados de conexão do dispositivo móvel, assim como o redirecionamento dos dados que foram passados para o mesmo quando ele se encontra fora da rede local, conexão realizada por túnel;
- Agente estrangeiro: Roteador que entrega aos nós móveis que estão conectados em rede estrangeira, pacotes oriundos da conexão estabelecida por túnel entre a rede local e a rede estrangeira na qual estão alocados no momento.

O funcionamento básico da MIPv6 entrega dois endereços IP ao dispositivo móvel, um deles é fixo(*home address*), e faz parte da rede local no qual foi registrado primariamente. O segundo IP é temporário (*care-of-address*) e determinado pelo agente estrangeiro, que contém a posição atual do dispositivo na rede estrangeira.

Na primeira etapa o dispositivo fica encarregado de descobrir quais as redes ele pode conectar e as redes que disponibilizam *care-of-address*, assim como as características de encapsulamento disponível. Quando o nó móvel é registrado na rede estrangeira, é enviado um aviso para sua rede local com o objetivo de que os dados do celular (ligações, mensagens, e etc) sejam repassados para sua localização atual. Os dados são repassados para o

host móvel através do túnel entre o agente local e o agente estrangeiro(JOBSTRAIBIZER, 2011), como ilustrado na Figura 2.7 a seguir.

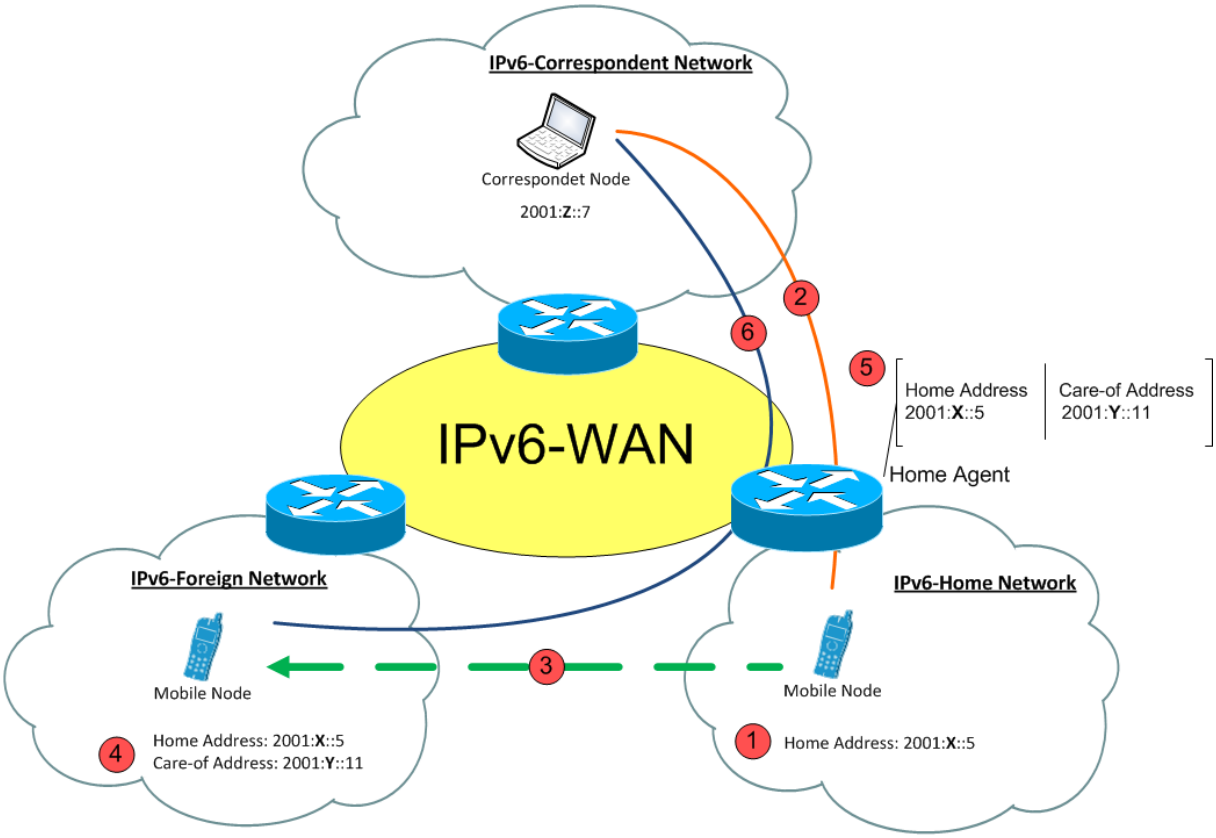

Figura 2.7: Passo a passo do serviço MIPv6.

Fonte: http://web.fhnw.ch/technik/projekte/i/bachelor11/borer-tribelhorn/einfuehrungmipv6.png

2.4.4 DHCPv6

O DHCP é um protocolo que é usado para configurar automaticamente IPv6 nos *hosts* localizados dentro de uma rede local. O protocolo é composto pelo lado do servidor que irá entregar os endereços, pelos clientes que recebem automaticamente as configurações e para retransmissão do DHCP (*relay*).

O servidor DHCPv6 é um servidor distinto, totalmente independente do IPv4 e entregam além do endereço, inúmeros parâmetros para o cliente, que são definidas no campo *options* como descrito no manual (DROMS, 2003). Esse trabalho utilizará apenas as principais funções para o funcionamento básico da rede.

2.4.5 DNS

O serviço de DNS (*Domain Name Resolution*) é responsável pela resolução de nomes para os endereços IP e se torna ainda mais fundamental com a chegada do protocolo IPv6. Isso se da ao fato da grande quantidade de endereços e da dificuldade de reprentação e memorização deles.

Os principais servidores DNS são do tipo primário(*master*), secundário(*slave*) e recursivo(*cache*). No trabalho atual o servidor recursivo será usado com o objetivo de efetuar consultas em IPv6, mantendo a cache local e melhorando o tempo de respostas das resoluções de nome.

Os RR(*Resource Records*) descrevem as características da zona ou domínio, cada um deles tem um tempo de expiração, classe, e possuem um tipo principal. No caso do IPv6 são utilizados o tipo A6 ou AAAA, o segundo é o padrão recomendado pela IETF e utilizado nesse trabalho.

O servidor DNS funciona operando em modo IPv4, IPv6 ou os dois protocolos ao mesmo tempo. Em todos modos, ele é capaz de resolver nomes para registros AAAA(IPv6) e A(IPv4), ou seja, não é necessário pilha dupla para a resolução de nomes dos dois protocolos.

2.4.5.1 DNSSEC

O DNSSEC surgiu como extensão para a tecnologia DNS existente para proporcionar maior segurança na Internet. Ele tem o objetivo de garantir a segurança na resolução de endereços, autenticidade e integridade. Contanto ele não é capaz de proteger contra ataques de negação de serviço e prover confidencialidade dos dados.

Para atingir o objetivo de autenticidade nas respostas do DNS, assinaturas digitais foram incluídas nas respostas, através de novos *resource records*. Os servidores recursivos de *cache*, como o utilizado no presente trabalho, podem validar as assinaturas digitais para prover que os dados DNS são autênticos, conforme mostrado na Figura 2.7.

O servidor DNS recursivo irá conter uma chave pública ancorada que servirá como inicio da cadeia de confiança, ao iniciar uma consulta, a chave ancorada será comparada com a *DNSKEY* do servidor *root* primeiramente, caso seja válida, continua com

as requisições. O servidor *root* não retorna resposta mas continua com referência para o próximo domínio de nível inferior, de modo que ela contém a delegação de zona, qual o servidor de nome e o RRSIG (representa a assinatura de um RR específico com uma determinada chave DNSKEY) dele. O servidor recursivo irá verificar a assinatura (RRSIG) através da DNSKEY e continuará assim a requisição até receber a resposta, formando assim uma cadeia de confiança, como ilustrado na Figura 2.8.

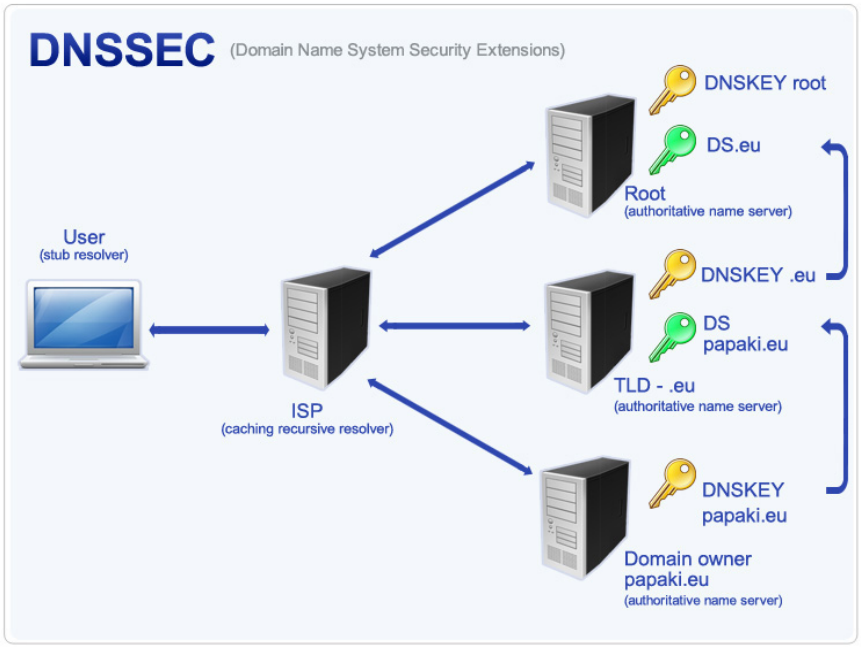

Figura 2.8: Demostração dos servidores recursivos com DNSSEC.

Fonte: http://imgs.gr/sites/all/themes/papaki2/img/dnssec-grafix.jpg

3 Estudo de Caso

O desenvolvimento do trabalho contou com a implatação da nova faixa de rede IPv6 pelo CGCO, disponibilizada para o NRC (em caráter de teste). Foi utilizado o sistema operacional Linux, ditribuição Fedora 17, Windows 7, assim como ferramentas e testes específicos para o funcionamento dos serviços voltados para a rede IPv6. Esse capítulo mostra os principais métodos utilizados para a realização do objetivo proposto, juntamente com o material utilizado.

3.1 Ferramentas e metodologia

O ambiente montado contou com uma máquina física principal, utilizada como servidor, e uma máquina virtual, ambas instaladas com sistema operacional Fedora 17, mais uma máquina virtual com Windows 7, todas dentro da faixa de IPv6 da UFJF. E contou também com uma máquina física instalada com Arch Linux em ambiente residencial para teste de túnel. Foram utilizados o Apache como servidor web, DHCP para servidor DHCPv6, *BIND* como servidor DNS, *RADVD* versão para anúncio do *gateway* e Wireshark para análise dos pacotes IPv6. Além dos pacotes principais, foram utilizadas ferramentas de rede do terminal bash Linux, como *dig*, *traceroute*, para auxilio e verificação de problemas.

3.1.1 Configuração da rede IPv6

Foi disponibilizada a faixa de endereçamento IPv6 2001:12f0:614::/64 para a UFJF, oferecendo 18,446,744,073,709,551,616 de *hosts* com endereço IP único. A máquina física utilizada como servidor, teve o endereço 2001:12f0:614::22 e *gateway* 2001:12f0:614::1 setado manualmente, as máquinas virtuais obtiveram suas configurações de endereço através da configuração sem estado e na faixa definida pelo *DHCPv6*, de 2001:12f0:614::30 até 2001:12f0:614::100. A máquina física fora do domínio da UFJF teve seu endereço definido pelo provedor CTBC de Uberlândia, 2001:1291:200:41a::2, *gateway* 2001:1291:200:41a::1.

3.1.2 Configuração sem estado(SLAAC) - radvd

O *radvd* é o *daemon* de anúncio de roteador para os *hosts* que atuam dessa forma, ele funciona mandando mensagens de anúncio de roteador(*router advertisement*) para a rede local através de endereços *multicast*, faixa ff02::1, ou quando um nó faz uma requição, enviando solicitações de roteador(*router solicitations*) através da faixa ff02::2. O *daemon* é utilizado por administradores de rede com o objetivo de efetivar a configuração sem estado em redes IPv6.

Para o funcionamento correto do servidor atuando como *gateway*, foram necessárias modificações em alguns paraâmetros no *kernel*, como descritos a seguir:

Tabela 3.1: Parâmetros do *Kernel* modificados.

Valor	Caminho	Descrição
1	*/proc/sys/net/ipv6/conf/p2p1/forwarding*	Opção ativada de modo que permita o encaminhamento de pacotes entre as interfaces.
2	*/proc/sys/net/ipv6/conf/p2p1/accept_ra*	Aceita anúncios de roteadores, opção setada como 2 indica que são aceitos mesmo se a opção *forwarding* descrita acima esteja ativada.
1	*/proc/sys/net/ipv6/conf/p2p1/accept_ra_defrtr*	Aprende o roteador padrão no anúncio do mesmo.

Os seguintes cenários são possíveis na implementação da autoconfiguração de endereço sem estado(*SLAAC*):

1. Configuração SLAAC sem configuração de DNS;

2. Configuração SLAAC com adição do DNS;

Como o presente trabalho visa a configuração da rede de modo funcional e dinâmico, serão utilizadas as configurações SLAAC com a adição do DNS. As opções que serão utilizadas no arquivo de configuração estão descritas a seguir.

Tabela 3.2: Opções de configuração *radvd*

Opção	Descrição
AdvSendAdvert	*Flag* que indica se o roteador deve ou não enviar solicitações de roteador e responder e também responde-las.
AdvManagedFlag	Indica que os *hosts* devem utilizar a configuração *statefull* para autoconfiguração do endereço.
AdvOtherConfigFlag	Quando setada a *flag*, os *hosts* utilizam configuração *statefull* para o restante de suas configurações, como endereço DNS.
prefix	Indica o prefixo da rede ou endereço da interface.
AdvAutonomous	Indica se o prefixo (*prefix*) pode ser utilizado para configuração autônoma de endereço.
RDNSS	Possibilita a entrega do endereço DNS quando setado.

3.1.2.1 Configuração SLAAC com RDNSS

A configuração sem estado tem a vantagem de entregar a rede de modo funcional, com solução única, e é o melhor exemplo em que as novas funcionalidades do IPv6 são implementadas, como *multicast* e anúncio de roteador com RDNSS. O servidor *radvd* foi configurado através do arquivo de configuração */etc/radvd.conf*, com as opções setadas conforme mostrado na Figura 3.1.

```
interface p2p1
{
        AdvSendAdvert on;
#       AdvOtherConfigFlag on;
#       AdvManagedFlag on;
#       MinRtrAdvInterval 30;
#       MaxRtrAdvInterval 100;
        prefix 2001:12f0:614::/64
        {
#               AdvOnLink on;
#               AdvAutonomous off;
#               AdvRouterAddr on;
        };
        RDNSS 2001:12f0:614::22 {};

};
```

Figura 3.1: Arquivo de configuração do *radvd*.

O endereço é gerado de modo que a primeira parte é composta pelo prefixo e os *bits* restantes composto pelo endereço MAC. As faixas de subrede estão geralmente alocadas no bloco "/64", para preservar funcionalidades, como a autoconfiguração. Apesar da facilidade de configuração, não seria possível limitar uma faixa de endereços, deixando uma imensa faixa de endereços aberta, e além disso o fabricante da interface pode ser identificado no endereço, e dependendo até o modelo da placa, como mostra a Figura 3.2.

```
[fmayumi1@vm-monografia ~]$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 2001:12f0:614:0:20c:29ff:feab:ac16  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::20c:29ff:feab:ac16  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:ab:ac:16  txqueuelen 1000  (Ethernet)
        RX packets 70944  bytes 9574412 (9.1 MiB)
        RX errors 0  dropped 7  overruns 0  frame 0
        TX packets 3158  bytes 636351 (621.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figura 3.2: Configuração da interface por SLAAC, endereço composto pelo prefixo e MAC.

No ambiente Windows a configuração *stateless* é recebida com sucesso com excessão do DNS, que ainda não tem no sistema operacional, ferramenta nativa para setar o mesmo através do RDNSSD. Nota-se também que o endereço é composto pelo prefixo da rede mas não segue o padrão mandatório da RFC2464 (CRAWFORD, 1998), o restante do endereço é gerado por *hashing* do MAC usando MD5 (NARTEN, 2001).

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-26-40-44
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:12f0:614:0:d9c8:4d1f:17b7:c67e(Preferred)
   Temporary IPv6 Address. . . . . . : 2001:12f0:614:0:3cda:6ad5:60f7:2f7a(Preferred)
   Link-local IPv6 Address . . . . . : fe80::d9c8:4d1f:17b7:c67e%11(Preferred)
   Default Gateway . . . . . . . . . : fe80::223:5aff:fe66:1031%11
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

Figura 3.3: Configuração SLAAC, endereço composto pelo prefixo e *hash*, DNS não é setado.

3.1.3 Servidor DHCPv6

A ferramenta DHCP para IPv6 usada será o *open source* DHCPv6, proposto pela ISC. A configuração proposta de gerenciamento centralizado tem algumas vantagens em relação a configuração *stateless*, mas ainda sim depende dela para configuração de rota. Com a escolha do DHCPv6 para a autoconfiguração pode-se escolher a faixa de endereços IPv6 e o acesso aos mesmos, e entregar serviços além do DNS, como o NTP.

As opções de configuração utilizadas no trabalho estão representadas pela tabela a seguir:

Tabela 3.3: Opções de configuração DHCPv6.

Opção	Descrição
default-lease-time	Tempo em segundos da concessão que será entregue para o cliente caso ele não especifique uma.
max-lease-time	Tempo máximo em segundos permitido para uma concessão.
subnet6	Indica o prefixo da rede local que os endereços estão alocados.
range6	Indica a faixa mínima e máxima para a entrega dos endereços IPv6.
dhcp6.name-servers	Escolha de quais endereços DNS serão entregues.

3.1.3.1 DHCPv6 (*Stateless*)

Nesse cenário, o DHCPv6 foi configurado de modo que irá entregar apenas o endereço local da rede, as configurações de *gateway* e DNS são responsabilidades da configuração sem estado. O servidor DHCPv6 foi configurado através do arquivo de configuração */etc/dhcp/dhcpd6.conf*, com as seguintes opções setadas, demonstradas na Figura 3.4.

```
default-lease-time 600;
max-lease-time 7200;
subnet6 2001:12f0:614::/64
{
        range6 2001:12f0:614::30 2001:12f0:614::100;
#       option dhcp6.name-servers 2001:12f0:614::22;
#       option routers 2001:12f0:614::1;//Nao funciona no DHCPv6
}
```

Figura 3.4: Arquivo de configuração *dhcpd6.conf*.

No lado do cliente, ainda não é possível a autoconfiguração do DNS pelo Network-Manager, pois a ferramenta ainda não é capaz de atualizar o arquivo */etc/resolv.conf* pelo DNS recursivo(*RDNSS*). Portanto será utilizado uma *daemon* auxiliar para a função. A *rdnssd* foi criada com o intuito de permitir a atualização do arquivo de configuração através do DNS recursivo(RDNSS) recebido.

Após configurar e iniciar os serviços do DHCPv6 no servidor e *rdnssd* no cliente, os endereços foram recebidos corretamente e dentro da faixa indicada, de modo que nos testes realizados o endereço assinalado na interface sempre começa pelo final da faixa, no exemplo final temos o final "::100", como mostra a Figura 3.5.

```
[root@vm-monografia fmayumi1]# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::20c:29ff:feab:ac16  prefixlen 64  scopeid 0x20<link>
        inet6 2001:12f0:614::100  prefixlen 128  scopeid 0x0<global>
        ether 00:0c:29:ab:ac:16  txqueuelen 1000  (Ethernet)
        RX packets 292533  bytes 42689546 (40.7 MiB)
        RX errors 0  dropped 2477  overruns 0  frame 0
        TX packets 12477  bytes 2689539 (2.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figura 3.5: Configuração com uso do *rdnssd*

No Windows o endereço é recebido com sucesso e dentro da faixa indicada do servidor DHCPv6, mas tem o mesmo problema de não ser possível setar o DNS, descrito na seção 3.1.2.1.

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-26-40-44
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:12f0:614::100(Preferred)
   Lease Obtained. . . . . . . . . . : Saturday, March 23, 2013 12:24:03 AM
   Lease Expires . . . . . . . . . . : Saturday, March 23, 2013 12:34:03 AM
   Link-local IPv6 Address . . . . . : fe80::d9c8:4d1f:17b7:c67e%11(Preferred)
   Default Gateway . . . . . . . . . : fe80::223:5aff:fe66:1031%11
   DHCPv6 IAID . . . . . . . . . . . : 234884137
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-18-DF-04-7F-00-0C-29-26-40-44
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

Figura 3.6: Endereço setado pelo DHCPv6 e DNS incompleto.

3.1.3.2 DHCPv6 (*Statefull*)

O modo de configuração *statefull* pode utilizar todas as opções que o DHCPv6 pode oferecer, deixando apenas configuração do endereço do *gateway* para ser setada manualmente ou por meio de complemento com configuração *stateless*. No arquivo de configuração do DHCP, ilustradas pela Figura 3.7, escolhe-se as seguintes configurações:

```
default-lease-time 600;
max-lease-time 7200;
subnet6 2001:12f0:614::/64
{
        range6 2001:12f0:614::30 2001:12f0:614::100;
        option dhcp6.name-servers 2001:12f0:614::22;
#       option routers 2001:12f0:614::1;//Nao funciona no DHCPv6
}
```

Figura 3.7: Arquivo de configuração *dhcpd6.conf*.

Após configurar o *NetworkManager* no cliente para receber as as configurações automáticas por DHCP, todas elas foram recebidas com sucesso, mas o *gateway* não foi setado. Para verificar as configurações, foi executado o comando *route -6 -n* e constatado que as *flags* indicando que o roteamento está ativo(U) em conjunto com uso do *gateway*(G) não foram encontradas, conforme Figura 3.8.

```
[root@vm-monografia fmayumi1]# route -6 -n
Kernel IPv6 routing table
Destination                    Next Hop                   Flag Met Ref Use If
::1/128                        ::                         U    256 0     0 lo
fe80::/64                      ::                         !n   256 0     0 lo
fe80::/64                      ::                         U    256 0     0 eth0
::/0                           ::                         !n   -1  1   895 lo
::1/128                        ::                         Un   0   1     3 lo
fe80::/128                     ::                         Un   0   1     0 lo
fe80::20c:29ff:feab:ac16/128   ::                         Un   0   1     8 lo
ff00::/8                       ::                         U    256 0     0 eth0
::/0                           ::                         !n   -1  1   895 lo
```

Figura 3.8: Imagem demonstra que a opção UG não foi setada

Realizando uma análise detalhada dos pacotes ICMPv6 de *router advertisement* recebidos no cliente, pode-se ver no Wireshark que o *gateway* envia os pacotes corretamente e com as flags AdvManagedFlag(*bit* M) e AdvOtherConfigFlag(*bit* O) setadas, demonstrado na Figura 3.9.

```
▽ Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0x567f [correct]
    Cur hop limit: 64
  ▽ Flags: 0xc0
      1... .... = Managed address configuration: Set
      .1.. .... = Other configuration: Set
      ..0. .... = Home Agent: Not set
      ...0 0... = Prf (Default Router Preference): Medium (0)
      .... .0.. = Proxy: Not set
      .... ..0. = Reserved: 0
    Router lifetime (s): 1800
    Reachable time (ms): 0
    Retrans timer (ms): 0
  ▷ ICMPv6 Option (Prefix information : 2001:12f0:614::/64)
  ▷ ICMPv6 Option (Source link-layer address : 00:23:5a:66:10:31)
```

Figura 3.9: Pode-se ver que o pacote *RA* foi recebido e com as *flags* setadas corretamente

Portanto foi constatado que o NetworkManager não suporta esse tipo de cenário

ainda, deixando o *gateway* para ser setado manualmente ou por um dos modos já vistos acima (REDHAT, 2011).

No Windows, o cliente recebeu e setou todos os parâmetros de configuração com sucesso.

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-26-40-44
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:12f0:614::100(Preferred)
   Lease Obtained. . . . . . . . . . : Saturday, March 23, 2013 12:26:43 AM
   Lease Expires . . . . . . . . . . : Saturday, March 23, 2013 12:36:43 AM
   Link-local IPv6 Address . . . . . : fe80::d9c8:4d1f:17b7:c67e%11(Preferred)
   Default Gateway . . . . . . . . . : fe80::223:5aff:fe66:1031%11
   DHCPv6 IAID . . . . . . . . . . . : 234884137
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-18-DF-04-7F-00-0C-29-26-40-44
   DNS Servers . . . . . . . . . . . : 2001:12f0:614::22
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

Figura 3.10: Endereço setado corretamente pelo DHCPv6.

3.1.4 *Tunnel Broker* - SIXXS

Para o teste do novo protocolo para locais que somente possuem acesso ao IPv4, usa-se a alternativa chamada de *Tunnel Broker* para obter a conectividade IPv6. No presente trabalho será utilizado a opção gratuita oferecida pelos site da SIXXS. Para obter acesso aos serviços é necessário cadastrar, e após receber o e-mail de confirmação, loga-se no site e requisita-se um túnel, como demonstrado na Figura 3.11.

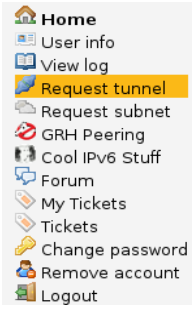

Figura 3.11: Requisição de túnel na página inicial.

Escolhe-se o túnel de localidade mais próxima, de modo que a latência seja a menor possível e obtenha-se melhores resultados. No Brasil, apenas a CTBC em Uberlândia oferece o serviço, que será o escolhido, juntamente justificado pela razão da utilização, conforme mostra a Figura 3.12.

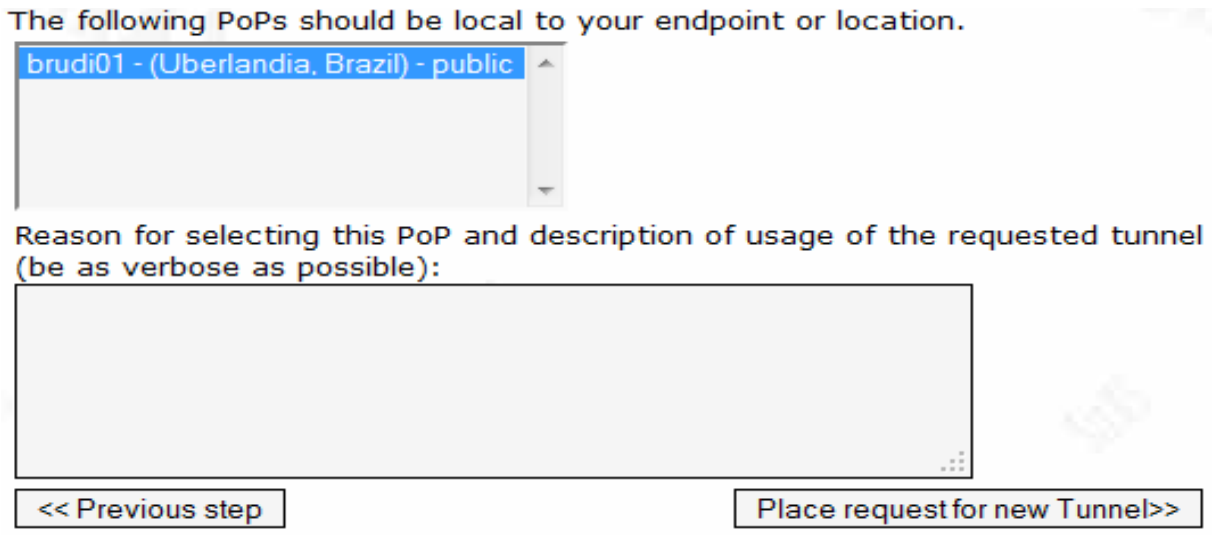

Figura 3.12: Escolha da região do Túnel a ser utilizado.

Após enviar os requisitos, a análise será realizada e se aprovada, um e-mail com as configurações será enviado. Ele será a base para a configuração do túnel. A daemon *aiccu* será utilizada para a configuração, arquivo localizado em */etc/aiccu.conf*. As configurações são ilustradas pela Figura 3.13 e seta-se o usuário, senha, nome da interface e ID do túnel, conforme enviado no e-mail.

```
# AICCU Configuration

# Login information (defaults: none)
username usuario-SIXXS
password senhaaqui

# Protocol and server to use for setting up the tunnel (defaults: none)
protocol tic
#<tic|tsp|l2tp>
#server <server to use>

# Interface names to use (default: aiccu)
# ipv6_interface is the name of the interface that will be used as a tunnel inte
rface.
# On *BSD the ipv6_interface should be set to gifX (eg gif0) for proto-41 tunnel
s
# or tunX (eg tun0) for AYIYA tunnels.
ipv6_interface sixxs

# The tunnel_id to use (default: none)
# (only required when there are multiple tunnels in the list)
tunnel_id T115066
```

Figura 3.13: Configurações do arquivo *aiccu.conf*

Reinicia-se o serviço *aiccu* e as configurações são setadas, na interface de escolha, no caso a interface *SIXXS* foi o nome dado. O host está pronto para conectar a rede IPv6 e as configurações de endereço são ilustradas a seguir, na figura 3.14.

```
[root@arch-casa fmayumi]# ifconfig sixxs
sixxs: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1280
        inet6 2001:1291:200:41a::2  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::1091:200:41a:2  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500
```

Figura 3.14: Configurações do endereço do túnel.

3.1.5 Servidor Web - Apache

O servidor *Apache*, criado em 1995, é o *open-source* multiplataforma amplamente utilizado. Possui inúmeras funcionalidades, estendidas atravé de módulos compilados. Suas configurações estão localizadas em */etc/httpd/conf/httpd.conf*. Para ativar o IPv6 é relativamente simples, é necessário apenas alterar as configurações do *Listen*.

Tabela 3.4: Opções de configuração do *Apache*

Modo	*Listen*
Somente IPv4	Não será utilizado, por exemplo Listen IP:80
Somente IPv6	Escuta no IPv6 e porta 80, Listen [2001:12f0:614::22]:80.
Pilha Dupla	Escuta os dois protocolos na porta 80, Listen *.

Servidor Web foi reiniciado e acesso pelo próprio servidor e pela máquina virtual retornou a página inicial com sucesso.

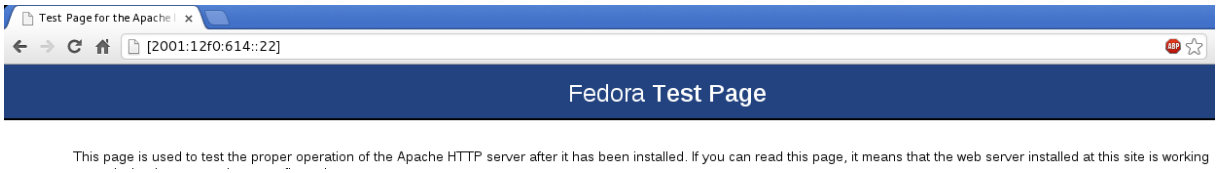

Figura 3.15: Página inicial do *Apache* acessado por IPv6.

3.1.6 Servidor DNS - BIND

O servidor utilizará o *software* BIND para implementação do serviço de DNS. O BIND é o *open-source* criado em 1986 por alunos da Universidade de Berkeley e o servidor DNS mais utilizado atualmente. O arquivo de configuração principal está localizado em */etc/named.conf* e as seguintes opções principais serão utilizadas:

Tabela 3.5: Opções de configuração do *BIND*

Opção	Descrição
listen-on	Ativa ou desativa consultas de nome para a rede IPv4, desligada e setada para *none*.
listen-on-v6	Escutará por consultas IPv6, e não será alterada a porta padrão, 53.
allow-recursion	Indica para quais os endereços será permitido a recursão, ::1 e 2001:12f0:614/64.
allow-transfer	Proibe transferências de zona a não ser pelos IPs definidos, setado para *none*.
directory	Diretório padrão dos arquivos de zona e extras de configuração.
dnssec-enable	Ativa as funções DNSSEC.
dnssec-validation	Usado em conjunto com o dnssec-enable, é utilizado apenas em servidores recursivos para ativar a criptografia e validar as zonas assinadas.
managed-keys	Gerência das chaves e onde irão os dados da âncora de confiança(*trust anchor*).

Os arquivos foram configurados de tal forma que o servidor responderá as consultas de nome que estão no cache, e caso não estejam presentes, ele as buscará recursivamente, utilizando o arquivo de zona */var/named/root.ca* para iniciar as consultas. As configurações foram montadas para que o cenário respondesse apenas a solicitações pela rede IPv6, tanto para registros A e AAAA, já que os servidores DNS podem entregar ambos os tipos de registro, independente do protocolo, assim a rede IPv4 foi desligada. Desse modo subiu-se o daemon e as consultas foram feitas através do utilitário do BIND, o *dig*. As seguintes pesquisas foram realizadas:

```
[root@fernando-note fmayumi]# dig -6 AAAA @2001:12f0:614::22 ipv6.br +short
2001:12ff:0:4::22
[root@fernando-note fmayumi]# dig -6 AAAA @2001:12f0:614::22 ipv6dobrasil.com +short
2001:470:0:207::403e:9056
[root@fernando-note fmayumi]# dig -6 AAAA @2001:12f0:614::22 ipv6.google.com +short
[root@fernando-note fmayumi]# dig -6 AAAA @2001:12f0:614::22 www.v6.facebook.com +short
[root@fernando-note fmayumi]#
```

Figura 3.16: Consultas de nome ao Google e Facebook não retornam endereços para rede IPv6 nativa.

Pelos resultados foi possível perceber que as consultas aos nomes do Google e do Facebook não foram recebidas com sucesso. Analisando a rota mais detalhadamente com o a adição da opção *+trace* percebe-se que, ainda para alguns endereços, é necessário pilha dupla para que a consulta seja retornada com sucesso. O teste realizado mostra que

mesmo para o DNS do Google para rede nativa IPv6, nada é retornado quando traçamos a rota pela rede IPv6, como mostra a Figura 3.17

```
[root@fernando-note fmayumi]# dig -6 AAAA @2001:4860:4860::8888 +trace +short ip
v6.google.com
NS h.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS i.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS g.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS j.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS m.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS b.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS d.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS f.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS a.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS c.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS l.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS k.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
NS e.root-servers.net. from server 2001:4860:4860::8888 in 185 ms.
RRSIG NS 8 0 518400 20130402000000 20130325230000 40323 . aTGYFsYGkrlO2C8ZtYh7RS
B7WHfU0aqVZfCtIS/+wAN9tIkATPQbQ8sn rAKMf1tgtCpqerd2P31MHHpc9xmd3pq9hrcc366nQ/oo5
JGwmlTg4eYY R0+MKctqht0hLhqL0mS+jnHkNkMHKbcsj4D9l40cYZzyVh6vJsygG/wh V28= from s
erver 2001:4860:4860::8888 in 185 ms.
dig: couldn't get address for 'ns2.google.com': no more
```

Figura 3.17: Traçando a rota para a consulta de nome pelos servidor DNS do Google.

Servidores de cache recursivos são a parte mais importante na parte de implatação do DNSSEC, isso ocorre porque eles irão validar as respostas para as consultas DNS efetuadas pelos clientes. O servidor será configurado com uma *trust anchor*, e através dela será capaz de validar as assinaturas usadas pelo DNSSEC (RIJSWIJK, 2012). Primeiramente pega-se a chave do servidor da IANA, a de valor 257(*security entry point*).

```
[root@fernando-note fmayumi]# dig DNSKEY . | grep 257
.                       45167   IN      DNSKEY  257 3 8 AwEAAagAIKlVZrpC6Ia7gEza
h0R+9W29euxhJhVVLOyQbSEW0O8gcCjF FVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL
2MTJRkxoX bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD X6RS6CXpoY68L
svPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz W5h0A2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sG
IcGOYl7OyQdXfZ57relS Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq Qx
A+Uk1ihz0=
```

Figura 3.18: Geração da chave de segurança do *trust anchor*.

Após gerar a *trust anchor* mais recente, ela é incluída no arquivo de configuração */etc/named.conf*, na opção *managed-keys* e o servidor DNS é reiniciado. Para conferir o funcionamento do DNSSEC foi executado novamente uma consulta em um site que disponiliza a funcionalidade de segurança DNSSEC, a ferramenta escolhida foi novamente o *dig*, com as opções escolhidas conforme demonstra a Figura 3.19. Verifica se a *flag* AD

(dados autenticados) está presente.

```
[root@fernando-note fmayumi]# dig @2001:12f0:614::22 +dnssec +noauthority +noadd
itional AAAA ipv6.br

; <<>> DiG 9.9.2-rl.028.23-P1-RedHat-9.9.2-5.P1.fc17 <<>> @2001:12f0:614::22 +dn
ssec +noauthority +noadditional AAAA ipv6.br
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39150
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 6, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;ipv6.br.                       IN      AAAA

;; ANSWER SECTION:
ipv6.br.                86253   IN      AAAA    2001:12ff:0:4::22
ipv6.br.                86253   IN      RRSIG   AAAA 5 2 86400 20130506163818 20
130225163818 35077 ipv6.br. kjztMXUbvgaKdZzqTTnNxJEdeRLMP3/m+fGgsTsLBwUkafClZNqz
1OzC DYM9/jUYRDD1h+LgFe265CcKbBZojnOUKiLt9bXCyr1H7Mv6FR2hu353 orN9wAwl2xA3ybRtOX
Mx7qzojY+tUc7gxJruAzR8ekutGx1L5+pktUzm Uw37GVqOQZ4OFPRbnQiQNUWI

;; Query time: 1 msec
;; SERVER: 2001:12f0:614::22#53(2001:12f0:614::22)
;; WHEN: Mon Mar 25 22:35:38 2013
;; MSG SIZE  rcvd: 678
```

Figura 3.19: Consulta com DNSSEC em funcionamento.

4 Considerações Finais

O esgotamento dos endereços IPv4 é eminente e pode prejudicar o crescimento da Internet, a IANA, órgão responsável pela coordenação global de endereços distribuiu os últimos cinco blocos de IP para os órgãos regionais no mundo, encontrando-se assim oficialmente sem endereços IPv4 para distribuição, os remanescentes que foram distribuídos, agora estão por responsabilidade dos órgãos regionais e inicia-se a contagem regressiva para seu fim. Portanto a subtituição pelo novo protocolo IPv6 já é realidade para os administradores de redes. O endereçamento de 128 *bits* atende as conformidades em questão e por ser um protocolo desenvolvido em uma fase mais atual traz novos benefícios para antigas preocupações.

Pelos cenários montados e testes efetuados, nota-se que a melhor solução ainda é a pilha dupla, pois é o único modo em que os serviços continuam rodando sem restrições e oferecendo todas as funcionalidades para a rede. A técnica mostrou eficiência nos testes realizados quando juntou os dois mundos, mas traz a desvantagem de duplas configurações para os protocolos e que os equipamentos rodem os dois modos em conjunto.

A maioria dos grandes provedores de Internet ainda não oferecem o serviço IPv6 para os clientes, principalmente para os usuários domésticos. E mesmo testes realizados em redes externas mostraram que até os grandes serviços e provedores ainda não oferecem suporte nativo para a rede IPv6.

Muito ainda tem que ser melhorado em relação ao suporte e funcionamento das ferramentas, as novas funcionalidades ainda estão longe de ter cenários favoráveis com o que se tem disponível nos sistemas operacionais, e para que se possa ser montados de forma eficiente pelos especialistas da área. No caso prático em redes Linux, o *NetworkManager* se mostrou ineficiente para alguns cenários e para alguns casos precisa de software terceiros para funcionamento estável.

Com relação a segurança não se pode afirmar que o protocolo IPv6 é mais seguro do que o IPv4, o antigo protocolo já está no mercado há mais tempo e sofre de constantes ataques e pesquisas por longo tempo, enquanto o novo protocolo ainda terá que passar

por testes em larga escala. O que se fala é que o protoclo IPv6 engloba soluções para falhas de segurança com erros aprendidos no passado.

Mesmo com a transição estar se dando de maneira lenta e gradual, vai ser imprescindível preparar a rede para receber o IPv6. A melhor solução é já ir começando a capacitação dos funcionários da área. Deste modo poderão ser feitas análise dos erros de forma que impactue a rede o menos possível e seja feita a transição de software e hardware de maneira eficiente.

Referências Bibliográficas

ANTONIOLI, L. **Estatísticas, dados e projeções atuais sobre a internet no brasil**. http://tobeguarany.com/internet_no_brasil.php, 2012. [Online; acessado em 06 de Dezembro de 2012].

CISCO. **Cisco visual networking index prevê que o tráfego global de dados móveis crescerá 13 vezes até 2017**. http://globalnewsroom.cisco.com/easyir/BR/pt/local/press-release/Cisco-Visual-Networking-Index-preve-que-o-trafego-de-dados-moveis.html, 2013. [Online; acessado em 14 de Fevereiro de 2013].

CRAWFORD, M. **Transmission of IPv6 Packets over Ethernet Networks**. RFC, December 1998.

DEERING, S. **IP Version 6 Addressing Architecture**. RFC, July 1998.

DROMS, R. **Dynamic Host Configuration Protocol for IPv6 (DHCPv6)**. RFC, July 2003. [Online; acessado em 07 de Março de 2013].

FLORENTINO, A. A. **IPv6 na prática**. Linux Magazine, 2012.

et al., M. A. F. Ipv6. **V Workshop de Administração e Integraç ao de Sistemas**, v.1, p. 6, 1998.

IPV6ḂR, E. **Faq**. http://ipv6.br/faq/, Maio 2012. [Online; acessado em 10 de Fevereiro de 2013].

IPV6ḂR, E. **Cabeçalho**. http://ipv6.br/entenda/cabecalho/, Maio 2012. [Online; acessado em 29 de Novembro de 2012].

JOBSTRAIBIZER, F. Mobilidade com ipv6. **Linux Magazine**, v.1, p. 3, 2011.

MALKI, K. E. **Mobile ipv6 tutorial**. http://www.usipv6.com/ppt/MobileIPv6_tutorial_SanDiegok.pdf, 2003. [Online; acessado em 03 de Março de 2013].

NARTEN, T. **Privacy Extensions for Stateless Address Autoconfiguration in IPv6**. RFC, January 2001.

PEPELNJAK, I. **Ipv6 stateless autoconfiguration 101**. http://blog.ioshints.info/2011/10/ipv6-stateless-autoconfiguration-101.html, 2011. [Online; acessado em 14 de Março de 2013].

REDHAT. **Networking/addressing**. https://fedoraproject.org/wiki/Networking/Addressing#SLAAC_with_DHCPv6_Information_Request, 2011. [Online; acessado em 1 de Março de 2013].

RIJSWIJK. **Deploying dnssec**. http://www.surfnet.nl/Documents/rapport_Deploying_DNSSEC_v20.pdf., 2012. [Online; acessado em 20 de Fevereiro de 2013].

TANENBAUM, A. S. **Redes de Computadores**. Elsevier Brasil, 2003.

TECHNET. **Funcionalidades do ipv6**. http://technet.microsoft.com/pt-pt/library/cc780593(v=ws.10).aspx, 2013. [Online; acessado em 29 de Janeiro de 2013].

WIKIUNIVERSIDADE, C. **Introdução as redes de computadores/protocolos e serviços de rede**. http://pt.wikiversity.org/wiki/Introdu{\cc}\~ao_as_Redes_de_Computadores/Protocolos_e_servi{\cc}os_de_rede, 2013. [Online; acessado em 18 de Dezembro de 2012].

WIKIUNIVERSIDADE, C. **Protocolo (ciência da computação)**. http://pt.wikipedia.org/wiki/Protocolo_(ci\^encia_da_computa{\cc}\~ao), 2013. [Online; acessado em 19 de Dezembro de 2012].

WIKILIVROS, C. **Redes de computadores/protocolo ip**. http://pt.wikibooks.org/wiki/Redes_de_computadores/Protocolo_IP, 2013. [Online; acessado em 02 de Dezembro de 2013].