

Apostila

IPv6 Básico

Antonio Marcos Moreiras
Edwin Santos Cordeiro
Rodrigo Regis dos Santos
Alexandre Yukio Harano
Eduardo Barasal Morales
Heitor de Souza Ganzelli
Tiago Jun Nakamura
Rodrigo Mattos Carnier
Tuany Tabosa

Núcleo de Informação e Coordenação do Ponto BR - NIC.br

São Paulo

2012

Núcleo de Informação e Coordenação do Ponto BR

Diretor Presidente

Demi Getschko

Diretor Administrativo

Ricardo Narchi

Diretor de Serviços

Frederico Augusto de Carvalho Neves

Diretor de Projetos Especiais e Desenvolvimento

Milton Kaoru Kashiwakura

Centro de Estudos e Pesquisas em Tecnologia de Redes e Operações - CEPTRO.br

Antonio Marcos Moreiras

Coordenação Executiva e Editorial

Antonio Marcos Moreiras

Autores / Design / Diagramação

Antonio Marcos Moreiras

Edwin Santos Cordeiro

Rodrigo Regis dos Santos

Alexandre Yukio Harano

Eduardo Barasal Morales

Heitor de Souza Ganzelli

Tiago Jun Nakamura

Rodrigo Mattos Carnier

Tuany Tabosa

Sobre o CEPTRO

O Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações é a área do NIC.br responsável por iniciativas que visam melhorar a qualidade da Internet no Brasil e disseminar seu uso, com especial atenção para seus aspectos técnicos e de infraestrutura. A equipe do CEPTRO desenvolve soluções em infraestrutura de redes, software e hardware, além de gerenciar projetos executados por parceiros externos.

Os Pontos de troca de Tráfego do PTTMetro, espalhados por todo o Brasil, são o serviço mais importante do CEPTRO, ajudando a organizar a infraestrutura da Internet no país, tornando-a mais resiliente e diminuindo seus custos.

A equipe atua também na medição da qualidade da Internet, por meio de iniciativas como o SIMET (medição de qualidade na última milha) e SAMAS (medição de qualidade entre provedores).

Realiza ainda a divulgação da Hora Legal Brasileira via Internet, com o NTP.br, conduz estudos sobre a Web, atua em projetos para fomentar a criação de conteúdo multimídia em língua portuguesa na Internet, como o portal de vídeos Zappiens.br, além de outras iniciativas.

O trabalho do CEPTRO é financiado pelo registro de domínios ".br".

Sobre o IPv6.br

O IPv6.br engloba uma série de iniciativas do NIC.br para disseminar o IPv6 no Brasil, coordenadas pelo CEPTRO. Entre elas pode-se citar o sítio Web <http://ipv6.br>, o blog <http://ipv6.br/blog>, o validador de sítios Web para IPv6 <http://ipv6.br/validador>, o curso e-learning <http://ipv6.br/curso>.

O IPv6.br também oferece cursos presenciais gratuitos, com teoria e prática num laboratório multivendor, para provedores Internet e outros Sistemas Autônomos. Entre 2009 e 2011 cerca de 1700 pessoas foram capacitadas nesses treinamentos. O material didático do curso, do qual esta apostila faz parte, foi desenvolvido pelo NIC.br e está disponível sob uma licença Creative Commons bastante permissiva, podendo ser usado livremente para disseminar o conhecimento.

As iniciativas englobam ainda o fornecimento de transito IPv6 gratuitamente, em caráter experimental, para os membros do PTT Metro São Paulo, a realização de palestras em universidades, empresas e eventos de tecnologia, bem como a realização de eventos sobre o IPv6, como os “Fóruns Brasileiros de Implementadores IPv6”, e os diversos “IPv6 no Café da Manhã”.

Licença

Esta apostila está disponível sob a licença:

CREATIVE COMMONS

Attribution-ShareAlike 3.0 Brasil (CC BY-SA 3.0)

O texto integral da licença pode ser obtido em:

<http://creativecommons.org/licenses/by-sa/3.0/br/legalcode>

Você tem a liberdade de:

- **Compartilhar** — copiar, distribuir e transmitir a obra.
- **Remixar** — criar obras derivadas.
- Fazer **uso comercial** da obra

Sob as seguintes condições:

- **Atribuição** — Você deve creditar a obra da forma especificada pelo autor ou licenciante (mas não de maneira que sugira que estes concedem qualquer aval a você ou ao seu uso da obra). Este trabalho deve ser creditado ao NIC.br e deve ser sempre mencionado que o trabalho original pode ser encontrado em <http://ipv6.br>.
- **Compartilhamento pela mesma licença** — Se você alterar, transformar ou criar trabalhos derivados desta obra, você poderá distribuir a obra resultante apenas sob a mesma licença, ou sob uma licença similar à presente.

Ficando claro que:

- Renúncia — Qualquer das condições acima pode ser [renunciada](#) se você obiver permissão do titular dos direitos autorais. Para esta finalidade, entre em contato via email ipv6@nic.br
- Domínio Público — Onde a obra ou qualquer de seus elementos estiver em [domínio público](#) sob o direito aplicável, esta condição não é, de maneira alguma, afetada pela licença.
- Outros Direitos — Os seguintes direitos não são, de maneira alguma, afetados pela licença:
 - Limitações e exceções aos direitos autorais ou quaisquer [usos livres](#) aplicáveis;
 - Os [direitos morais](#) do autor;
 - Direitos que outras pessoas podem ter sobre a obra ou sobre a utilização da obra, tais como [direitos de imagem](#) ou privacidade.
- Aviso — Para qualquer reutilização ou distribuição, você deve deixar claro a terceiros os termos da licença a que se encontra submetida esta obra.

Índice

Sobre o CEPTRO.....	3
Sobre o IPv6.br.....	3
Licença.....	4
Capítulo 1: IPv6: Histórico e motivação.....	8
1. O desenvolvimento da Internet e o rápido esgotamento dos endereços.....	9
2. CIDR, NAT, DHCP e endereços privados.....	11
3. IPng, a nova geração de IPs.....	13
4. Entendendo a distribuição dos endereços na Internet.....	15
5. A transição do IPv4 para o IPv6.....	17
Referências:.....	22
Capítulo 2: Cabeçalho.....	23
1. Cabeçalho IPv4.....	23
2. Cabeçalho IPv6.....	24
3. Campos do Cabeçalho IPv6.....	27
3.1. Cabeçalhos de extensão.....	28
3.2. Hop-by-Hop.....	29
3.3. Destination Options.....	29
3.4. Routing.....	30
3.5. Fragmentation.....	31
3.6. Authentication Header e Encapsulating Security Payload.....	31
4. Referências.....	32
Capítulo 3: Endereçamento.....	33
1. Representação dos endereços	33
2. Tipos de endereços IPv6	34
2.1. Endereços Unicast	34
2.1.1. Identificadores de interface	36
2.1.2. Endereços especiais	36
2.2. Endereços Anycast	37
2.3. Endereços Multicast	37
3. Políticas de alocação e designação.....	40
4. Recomendação do NIC.br.....	41
Capítulo 4: Funcionalidades Básicas do IPv6.....	42
1.1. Código de Identificação e Localização.....	44
1.2. Formato do pacote.....	44
1.3. Classes.....	44
2. Neighbor Discovery Protocol (NDP).....	46
2.1. Mensagens.....	47
2.1.1. Router Solicitation (RS).....	47
2.1.2. Router Advertisement (RA).....	48
2.1.3. Neighbor Solicitation (NS).....	50
2.1.4. Neighbor Advertisement (NA).....	51
2.1.5. Redirect.....	53
2.2. Campo opções nas mensagens (Options).....	55
2.2.1. Source Link Layer Address.....	55
2.2.2. Target Link Layer Address.....	55
2.2.3. Prefix Information.....	56
2.2.4. Redirect Header.....	57
2.2.5. MTU.....	58
2.2.6. Recursive DNS Server Option (RDNSS).....	58
3. Funcionalidades Básicas baseadas no NDP.....	59

3.1. Duplicate Address Detection (DAD).....	59
3.1.1. Exemplo.....	60
3.2. Address Resolution.....	62
3.2.1. Exemplo.....	63
3.3. Router Discovery	64
3.3.1. Exemplo 1.....	65
3.3.2. Exemplo 2.....	66
3.4. Prefix Discovery.....	68
3.5. Parameter Discovery.....	68
3.6. Neighbor Unreachability Detection.....	68
3.6.1. Exemplo.....	69
3.7. Redirect.....	70
3.7.1. Exemplo.....	71
4. Funcionalidades Básicas com foco no mecanismo de autoconfiguração.....	72
4.1. Autoconfiguração.....	72
4.2. Autoconfiguração Sateless de endereços: interna do dispositivo.....	73
4.3. Autoconfiguração Stateless: Router Advertisement.....	73
4.3.1. Exemplo.....	74
4.4. Autoconfiguração: DHCPv6.....	75
4.4.1. Formato do pacote DHCPv6.....	76
4.4.2. Formato do pacote DHCPv6 Options	78
4.4.2.1. Client Identifier Option.....	79
4.4.2.2. Server Identifier Option.....	79
4.4.2.3. Identity Association for Non-temporary Addresses Option (OPTION_IA_NA)	80
4.4.2.4. Identity Association for Prefix Delegation (OPTION_IA_PD).....	81
4.4.2.5. Option Request Option.....	82
4.4.2.6. Elapsed Time Option.....	83
4.4.3. DHCPv6 em modo Stateful.....	83
4.4.3.1. Exemplo.....	84
4.4.4. DHCPv6 em modo stateless.....	85
4.4.4.1. Exemplo.....	86
4.4.5. DHCPv6 prefix delegation.....	87
4.4.5.1. Exemplo.....	87
4.4.6. Influência dos roteadores no DHCPv6.....	90
5. Estado dos Endereços	91
6. Referencias	92
Capítulo 5: Segurança em IPv6.....	93
1. Falhas de segurança, ataques e defesas para redes IPv6.....	94
2. Falhas na descoberta de vizinhos e autoconfiguração stateless (SEND e RA Guard).....	95
3. IPsec.....	97
4. Proteções contra varreduras de rede.....	101
5. Firewall.....	103
6. Técnicas de Transição de IPv4 para IPv6.....	105
7. Considerações Finais.....	106
8. Referências	106
Capítulo 6: Técnicas de Transição para o IPv6	107
1. Cenários de coexistência de IPv6 e IPv4.....	108
2. Classificação das técnicas de transição.....	111
3. Pilha Dupla: IPv6 e IPv4 em todos os dispositivos.....	113
4. Túneis 6over4 (IPv6-over-IPv4).....	115
5. Túneis GRE.....	118

6. Tunnel Broker.....	119
7. Dual Stack Lite (DS-Lite).....	123
8. dIVI, dIVI e dIVI-pd.....	126
9. NAT64 e DNS64.....	133
10. 464XLAT.....	136
11. 4rd.....	138
12. 6PE e 6VPE.....	139
13. 6rd.....	140
14. 6to4.....	142
15. Teredo.....	146
16. ISATAP.....	148
17. A+P.....	149
18. NAT444.....	150
19. Considerações Finais.....	151
20. Referências	153

Capítulo 1: IPv6: Histórico e motivação

A Internet é um fenômeno muito recente, mas que alterou, e continua alterando, a forma como nos comunicamos, trabalhamos, compramos, vendemos, nos divertimos, nos relacionamos com o Estado e mesmo como nos organizamos enquanto sociedade. Seu efeito tem sido, em geral, benéfico. Tanto, que muitos argumentam que o acesso a ela deveria ser considerado um direito fundamental de todo ser humano. A Internet quebra fronteiras geográficas, socializa a informação, permite a colaboração entre as pessoas numa escala antes inimaginável, e favorece o desenvolvimento, tanto dos indivíduos, como das organizações. Sem dúvida, a Internet é algo que a sociedade deve preservar, e mais, deve trabalhar para que seja algo de que todos possam usufruir, e cujo desenvolvimento seja sempre contínuo.

Do ponto de vista tecnológico, a Internet é uma rede de alcance mundial, que interliga computadores, tablets, celulares e uma infinidade de outros dispositivos. Na verdade, como seu próprio nome sugere, é formada por uma interconexão de um grande número de redes, mais ou menos independentes umas das outras. As diversas redes que compõem a Internet são administradas por diferentes instituições, que têm objetivos diversos, e usam equipamentos de vários fabricantes. Ainda sim todas são capazes de interoperar. Como isso é possível?

A Internet só é possível por que todos os seus participantes concordam em seguir um conjunto comum de padrões tecnológicos. Esses padrões são criados de forma aberta e colaborativa e aprovados por um processo de consenso aproximado pela IETF (Internet Engineering Task Force). Há literalmente milhares de padrões que definem como cada função deve ser realizada na rede, como cada serviço e aplicação devem funcionar. Um deles, contudo, pode ser destacado: o IP (Internet Protocol). Um protocolo é um conjunto de regras de comunicação. O IP é a base tecnológica da rede, o protocolo que empresta seu nome a ela: Internet.

É importante lembrar que a Internet é construída sobre a infraestrutura de telecomunicações tradicional. A mesma infraestrutura que é usada para telefonia, rádio e TV. Ainda assim, a Internet é normalmente muito mais flexível e barata do que as demais tecnologias que fazem uso dos mesmos recursos de telecomunicações. Isso é possível porque ela faz um uso muito mais eficiente dos recursos. No lugar de utilizar a comunicação por circuitos, que faz a reserva antecipada dos recursos necessários para a comunicação entre emissor e receptor, a Internet utiliza a comutação de pacotes, dividindo a informação em pequenos blocos, que podem ser enviados de forma independente pela rede, seguindo seu caminho até o destino. A comunicação de pacotes permite o compartilhamento dos recursos de telecomunicações de forma muito eficiente, e a construção de redes extremamente resilientes, em que pode haver muitos caminhos diferentes entre dois pontos quaisquer.

Quem separa a Internet das telecomunicações é justamente o IP. O protocolo Internet é o responsável por identificar cada dispositivo presente na rede, por meio de números que chamamos de endereços, e por encapsular todos os dados que fluem através dela, agregando a eles informações suficientes para que cheguem a seus destinos. O IP pode fazer uso de diversos tipos de redes de telecomunicações diferentes, criando uma camada padronizada sobre a qual todos os demais protocolos e serviços na Internet funcionam.

O IPv6 é a versão mais recente do protocolo IP. Ela tem de ser implantada rapidamente na Internet, porque a versão anterior, o IPv4, não é mais capaz de suportar o crescimento da rede: não há mais endereços livres.

Neste capítulo será apresentado um breve histórico do desenvolvimento da Internet, para contextualizar a necessidade do IPv6, e um breve histórico de seu próprio desenvolvimento. Em seguida, será discutida a transição do IPv4 para o IPv6, como havia sido planejada inicialmente, e

como de fato vem acontecendo, com a apresentação de algumas estatísticas. Por fim, será discutida a situação atual.

1. O desenvolvimento da Internet e o rápido esgotamento dos endereços

O Departamento de Defesa (DoD - Department of Defense) do governo estadunidense iniciou em 1966, através de sua Agencia de Pesquisas e de Projetos Avançados (ARPA - Advanced Research Projects Agency), um projeto para a interligação de computadores em centros militares e de pesquisa. Este sistema de comunicação e controle distribuído com fins militares recebeu o nome de ARPANET, tendo como principal objetivo criar uma arquitetura de rede sólida e robusta, baseada na comutação de pacotes, e que pudesse lidar com a indisponibilidade de alguns de seus nós, funcionando com os computadores e ligações de comunicação restantes.

Em 1969, foram instalados os primeiros quatro nós da rede, na Universidade de Los Angeles (UCLA), na Universidade da Califórnia em Santa Bárbara (UCSB), no Instituto de Pesquisas de Standford (SRI) e na Universidade de Utah.

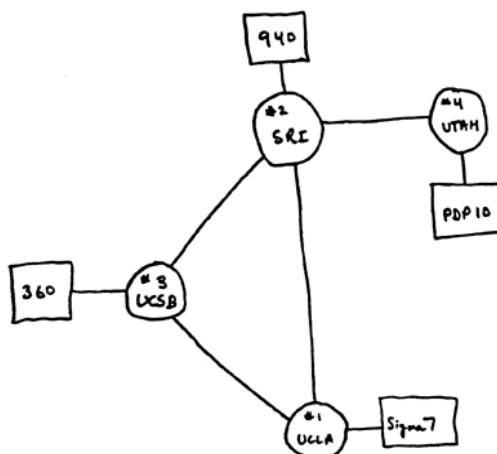


Figura 1: Mapa da ARPANET em 1969

fonte: http://www.computerhistory.org/internet_history

No início, a ARPANET trabalhava com diversos protocolos de comunicação, sendo o principal o NCP (Network Control Protocol). No entanto, em 10. de Janeiro de 1983, quando a rede já possuía 562 hosts, todas as máquinas da ARPANET passaram a adotar como padrão o conjunto de protocolos conhecido por TCP/IP, permitindo o crescimento ordenado da rede e eliminando restrições dos protocolos anteriores. Nesse mesmo ano, a ARPANET foi dividida em duas, uma rede fechada para os militares chamada MILNET, e uma rede aberta, com então 45 hosts, que evoluiu para a rede que hoje conhecemos como a Internet. Não há uma data formal na qual a ARPANET passou a ser a Internet e, ao discutir quando a Internet realmente começou, há diversos critérios diferentes que podem ser adotados. Uma das possibilidades é considerar a data de 10. de Janeiro de 1983 como o nascimento da Internet.

Um ponto interessante a ser notado é que, apesar de ter nascido como um projeto militar, desde o início a ARPANET foi utilizada para conectar instituições de pesquisa, e sempre foi baseada em padrões abertos. Talvez esse tenha sido um fator fundamental para sua evolução para a Internet que todos conhecem hoje, um ambiente aberto e propício a inovação, com um gerenciamento que não é centralizado, mas dividido entre diversas instituições, com a participação de operadores e usuários, incluindo a iniciativa privada, o governo e a sociedade civil.

O IP versão 4, definido na RFC 791, é, como visto, uma das principais bases tecnológicas, sobre as quais se sustentam a Internet. Esse protocolo mostrou-se bastante robusto, de fácil implantação e

interoperabilidade. No entanto, seu projeto remonta à década de 1970 e não previu alguns aspectos hoje importantes como:

- O crescimento das redes e um possível esgotamento dos endereços IP;
- O aumento da tabela de roteamento, que é a tabela onde estão relacionadas todas as interconexões entre as redes que compõem a Internet, permitindo identificar os possíveis caminhos para os pacotes, até seus destinos;
- Problemas relacionados a segurança dos dados transmitidos;
- Prioridade na entrega de determinados tipos de pacotes.

As especificações do IPv4 reservam 32 bits para endereçamento, possibilitando gerar mais de 4 bilhões de endereços distintos. Inicialmente, estes endereços foram divididos em três classes principais de tamanhos fixos, da seguinte forma:

- **Classe A:** definia o bit mais significativo como 0, utilizava os 7 bits restantes do primeiro octeto para identificar a rede, e os 24 bits restantes para identificar o host. Esses endereços utilizavam a faixa de 1.0.0.0 até 126.0.0.0;
- **Classe B:** definia os 2 bits mais significativo como 10, utilizava os 14 bits seguintes para identificar a rede, e os 16 bits restantes para identificar o host. Esses endereços utilizavam a faixa de 128.1.0.0 até 191.254.0.0;
- **Classe C:** definia os 3 bits mais significativo como 110, utilizava os 21 bits seguintes para identificar a rede, e os 8 bits restantes para identificar o host. Esses endereços utilizavam a faixa de 192.0.1.0 até 223.255.254.0;

O intuito dessa divisão tenha era tornar a distribuição de endereços flexível, abrangendo redes de tamanhos variados, mas essa classificação mostrou-se na verdade rígida e muito ineficiente, levando a um grande desperdício de endereços. A classe A, por exemplo, atendia a um número muito pequeno de redes, apenas 128 delas, mas ocupava metade de todos os endereços disponíveis. Já se houvesse a necessidade de endereçar uma rede relativamente pequena, com 300 dispositivos, seria necessário obter um bloco de endereços da classe B, desperdiçando assim quase o total dos 65 mil endereços. Os 256 endereços da classe C, por sua vez, não supriam as necessidades da grande maioria das redes.

Algumas dezenas de faixas classe A foram atribuídas integralmente a grandes instituições como IBM, AT&T, Xerox, HP, Apple, MIT, Ford, Departamento de Defesa Americano, entre muitas outras, disponibilizando para cada uma 16.777.216 endereços. Além disso, 35 faixas de endereços classe A foram reservadas para usos específicos como multicast, loopback e uso futuro.

Em 1990, existiam cerca de 313.000 hosts conectados à rede e estudos já apontavam para um colapso devido a falta de endereços. Outros problemas também tornavam-se aparentes, conforme a Internet evoluía, como o aumento da tabela de roteamento. Devido ao ritmo de crescimento da Internet e da política de distribuição de endereços, em maio de 1992, 38% das faixas de endereços classe A, 43% da classe B e 2% da classe C, já estavam alocados. Nesta época, a rede já possuía aproximadamente 1.136.000 hosts conectados.

Em 1993, com a criação da Web e com a liberação por parte do Governo estadunidense para a utilização comercial da Internet, houve um salto ainda maior em sua taxa de crescimento. O número de hosts passou de aproximadamente 2.056.000 em 1993, para mais de 26.000.000 em 1997.

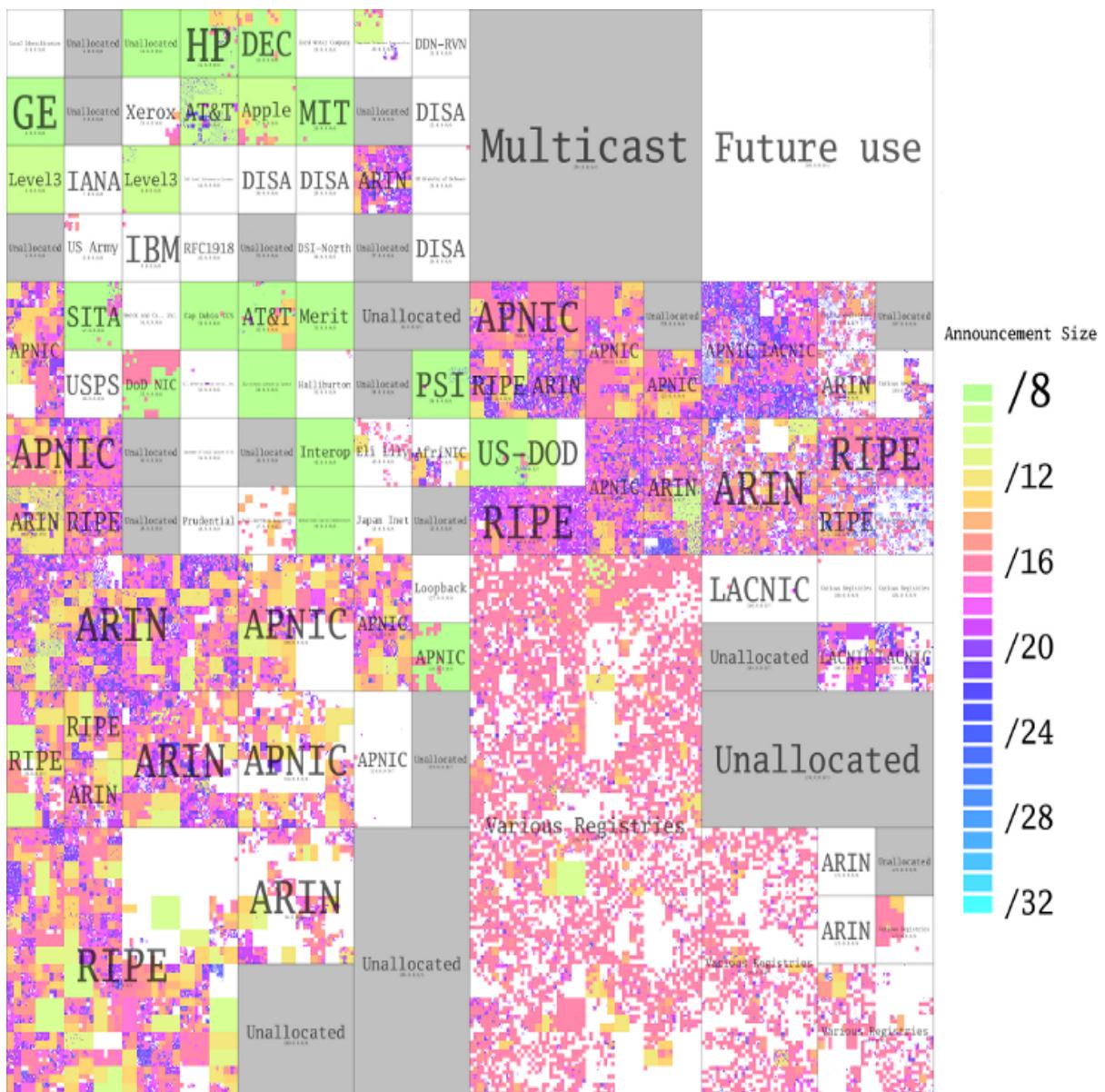


Figura 2: Mapa da tabela BGP em 2008

fonte: <http://maps.measurement-factory.com/gallery/RoutevIEWS/>

A figura 2 mostra uma visualização das informações da tabela de roteamento BGP extraída do projeto RoutevIEWS. Nela, o espaço de endereço IPv4 unidimensional é mapeado em uma imagem de bidimensional, onde blocos CIDR sempre aparecem como quadrados ou retângulos.

É possível observar, na figura, os grandes blocos classe A, no quadrante esquerdo superior, distribuídos para instituições como HP, DEC, AT&T, Apple, MIT, GE, Level3, IBM, Merit, etc, e como, desproporcionalmente, regiões como a do LACNIC usaram pouco espaço.

2. CIDR, NAT, DHCP e endereços privados

Dante desse cenário, a IETF (Internet Engineering Task Force) passa a discutir estratégias para solucionar a questão do esgotamento dos endereços IP e o problema do aumento da tabela de roteamento. Para isso, em novembro de 1991, é formado o grupo de trabalho ROAD (ROuting and Addressing), que apresenta como solução a estes problemas a utilização do CIDR (Classless Inter-domain Routing).

Definido na RFC 4632 (tornou obsoleta a RFC 1519), o CIDR tem como idéia básica o fim do uso de classes de endereços, permitindo a alocação de blocos de tamanho apropriado a real necessidade de cada rede; e a agregação de rotas, reduzindo o tamanho da tabela de roteamento. Com o CIDR os blocos são referenciados como prefixo de redes. Por exemplo, no endereço a.b.c.d/x, os x bits mais significativos indicam o prefixo da rede. Outra forma de indicar o prefixo é através de máscaras, onde a máscara 255.0.0.0 indica um prefixo /8, 255.255.0.0 indica um /16, e assim sucessivamente.

Outra solução, apresentada na RFC 2131 (tornou obsoleta a RFC 1541), foi o protocolo DHCP (Dynamic Host Configuration Protocol). Através do DHCP um host é capaz de obter um endereço IP automaticamente e adquirir informações adicionais como máscara de sub-rede, endereço do roteador padrão e o endereço do servidor DNS local.

O DHCP tem sido muito utilizado por parte dos ISPs por permitir a atribuição de endereços IP temporários a seus clientes conectados. Desta forma, torna-se desnecessário obter um endereço para cada cliente, devendo-se apenas designar endereços dinamicamente, através de seu servidor DHCP. Este servidor terá uma lista de endereços IP disponíveis, e toda vez que um novo cliente se conectar à rede, lhe será designado um desses endereços de forma arbitrária, e no momento que o cliente se desconecta, o endereço é devolvido.

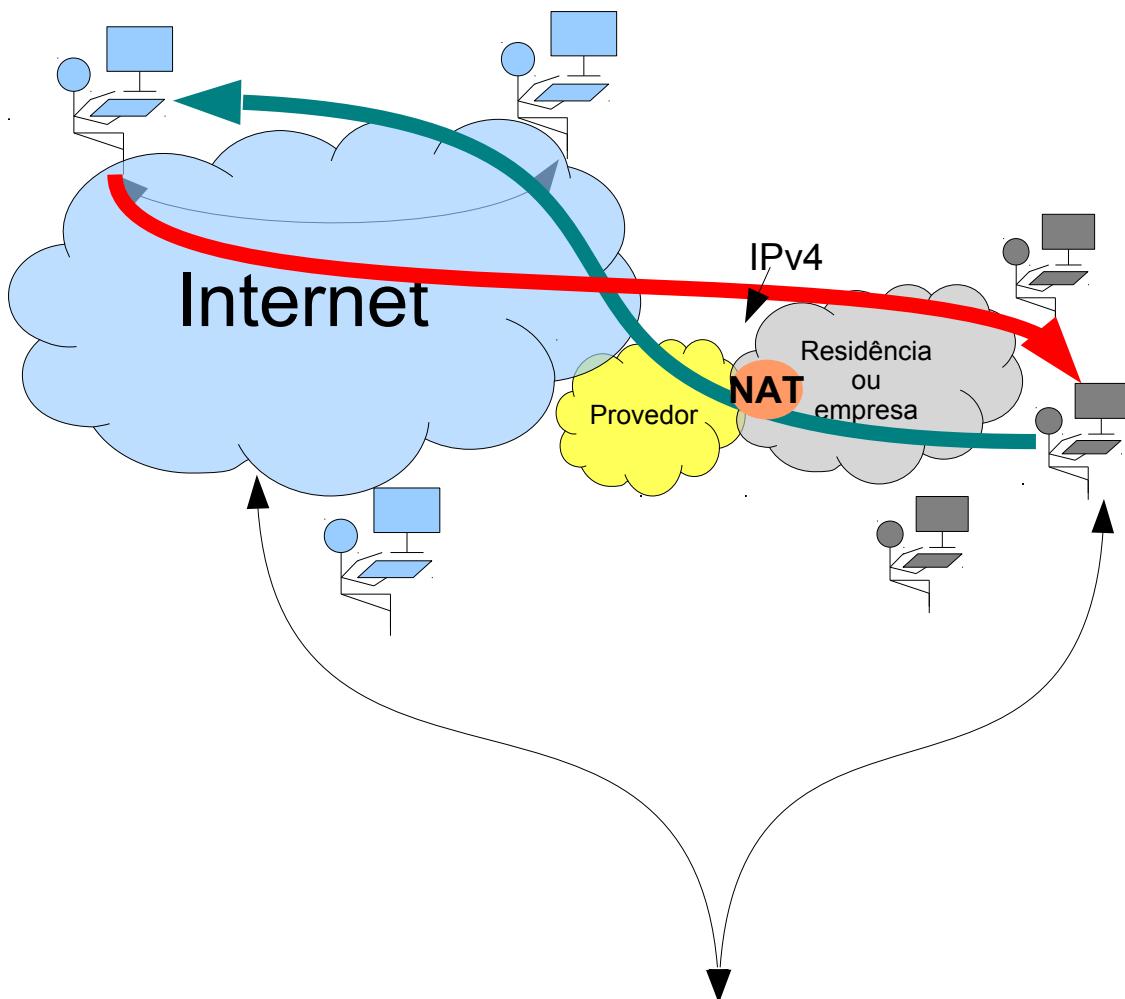


Figura3: Network Address Translation

A NAT (Network Address Translation), foi outra técnica paliativa desenvolvida para resolver o problema do esgotamento dos endereços IPv4. Definida na RFC 3022 (tornou obsoleta a RFC 1631), tem como ideia básica permitir que, com um único endereço IP, ou um pequeno número deles, vários hosts possam trafegar na Internet. Dentro de uma rede, cada computador recebe um endereço IP privado único, que é utilizado para o roteamento do tráfego interno. No entanto, quando um pacote

precisa ser roteado para fora da rede, uma tradução de endereço é realizada, convertendo endereços IP privados em endereços IP públicos globalmente únicos.

Para tornar possível este esquema, utiliza-se os três intervalos de endereços IP declarados como privados na RFC 1918, sendo que a única regra de utilização, é que nenhum pacote contendo estes endereços pode trafegar na Internet pública. As três faixas reservadas são:

A utilização da NAT mostrou-se eficiente no que diz respeito a economia de endereços IP, além de apresentar alguns outros aspectos positivos, como facilitar a numeração interna das redes, ocultar a topologia das redes e só permitir a entrada de pacotes gerados em resposta a um pedido da rede. No entanto, o uso da NAT apresenta inconvenientes que não compensam as vantagens oferecidas.

A NAT quebra o modelo fim-a-fim da Internet, não permitindo conexões diretas entre dois hosts, o que dificulta o funcionamento de uma série de aplicações, como P2P, VoIP e VPNs. Outro problema é a baixa escalabilidade, pois o número de conexões simultâneas é limitado, além de exigir um grande poder de processamento do dispositivo tradutor. O uso da NAT também impossibilita rastrear o caminho de pacote, através de ferramentas como traceroute, por exemplo, e dificulta a utilização de algumas técnicas de segurança como IPSec. Além disso, seu uso passa uma falsa sensação de segurança, pois, apesar de não permitir a entrada de pacotes não autorizados, a NAT não realiza nenhum tipo de filtragem ou verificação nos pacotes que passa por ela.

Embora estas soluções tenham diminuído a demanda por IPs, 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.

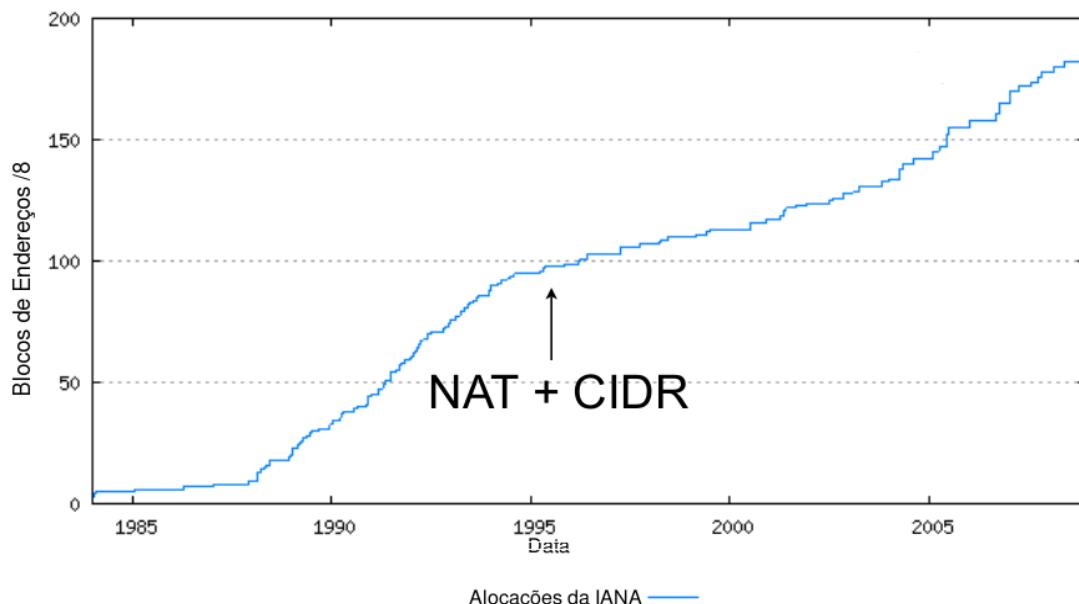


Figura 4: Efeito da introdução das medidas paliativas no consumo de IPs
fonte: <http://potaroo.net>

Essas medidas, na verdade, 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.

3. IPng, a nova geração de IPs

Deste modo, em dezembro de 1993 a IETF formalizou, através da RFC 1550, as pesquisas a respeito da nova versão do protocolo IP, solicitando o envio de projetos e propostas para o novo protocolo.

Esta foi uma das primeiras ações do grupo de trabalho da IETF denominado Internet Protocol next generation (IPng). As principais questões que deveriam ser abordadas na elaboração da próxima versão do protocolo IP foram:

- Escalabilidade;
- Segurança;
- Configuração e administração de rede;
- Suporte a QoS;
- Mobilidade;
- Políticas de roteamento;
- Transição.

Diversos projetos começaram a estudar os efeitos do crescimento da Internet, sendo os principais o CNAT, o IP Encaps, o Nimrod e o Simple CLNP. Destas propostas surgiu o TCP and UDP with Bigger Addresses (TUBA), que foi uma evolução do Simple CLNP, e o IP Address Encapsulation (IPAE), uma evolução do IP Encaps. Alguns meses depois foram apresentados os projetos Paul's Internet Protocol (PIP), o Simple Internet Protocol (SIP) e o TP/IX. Uma nova versão do SIP, que englobava algumas funcionalidades do IPAE, foi apresentada pouco antes de agregar-se ao PIP, resultando no Simple Internet Protocol Plus (SIPP). No mesmo período, o TP/IX mudou seu nome para Common Architecture for the Internet (CATNIP).

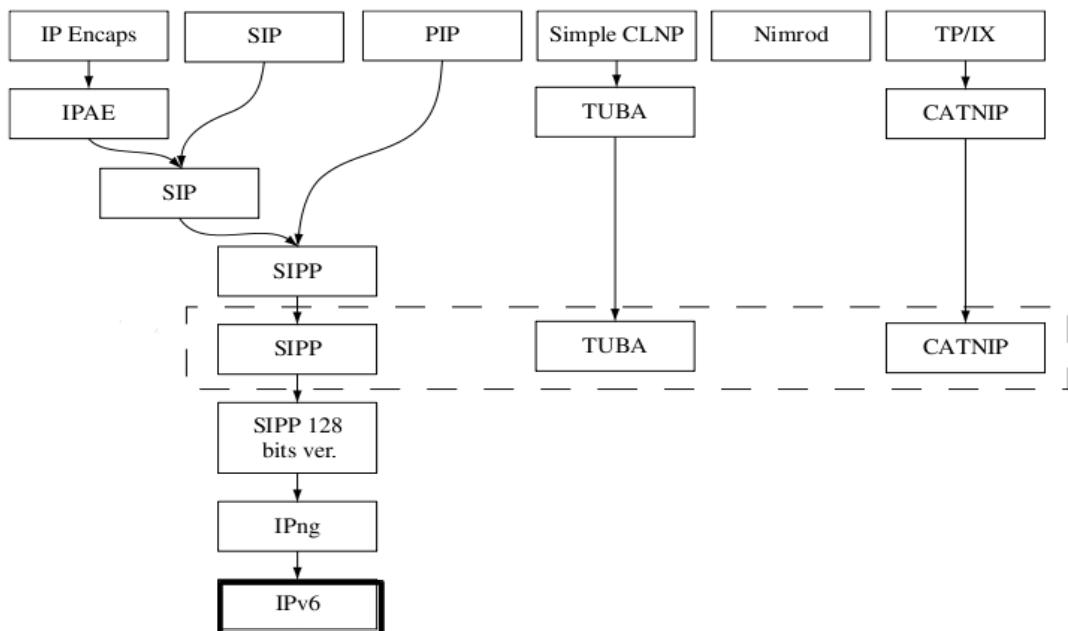


Figura 5: IPng: evolução do processo de criação do novo protocolo IP.

Fonte: Blanchet, Marc. Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks. John Wiley & Sons Ltd, 2006, Québec, Canada, p. 29.

Em janeiro de 1995, na RFC 1752 o IPng apresentou um resumo das avaliações das três principais propostas:

CATNIP - foi concebido como um protocolo de convergência, para permitir a qualquer protocolo da camada de transporte ser executado sobre qualquer protocolo de camada de rede, criando um ambiente comum entre os protocolos da Internet, OSI e Novell;

TUBA - sua proposta era de aumentar o espaço para endereçamento do IPv4 e torná-lo mais hierárquico, buscando evitar a necessidade de se alterar os protocolos da camada de transporte

e aplicação. Pretendia uma migração simples e em longo prazo, baseada na atualização dos host e servidores DNS, entretanto, sem a necessidade de encapsulamento ou tradução de pacotes, ou mapeamento de endereços;

SIPP - concebido para ser uma etapa evolutiva do IPv4, sem mudanças radicais e mantendo a interoperabilidade com a versão 4 do protocolo IP, fornecia uma plataforma para novas funcionalidades da Internet, aumentava o espaço para endereçamento de 32 bits para 64 bits, apresentava um nível maior de hierarquia e era composto por um mecanismo que permitia “alargar o endereço” chamado cluster addresses. Já possuía cabeçalhos de extensão e um campo flow para identificar o tipo de fluxo de cada pacote.

Entretanto, conforme relatado também na RFC 1752, todas as três propostas apresentavam problemas significativos. Deste modo, a recomendação final para o novo Protocolo Internet baseou-se em uma versão revisada do SIPP, que passou a incorporar endereços de 128 bits, juntamente com os elementos de transição e autoconfiguração do TUBA, o endereçamento baseado no CIDR e os cabeçalhos de extensão. O CATNIP, por ser considerado muito incompleto, foi descartado.

Após esta definição, a nova versão do Protocolo Internet passou a ser chamado oficialmente de IPv6.

As especificações da IPv6 foram apresentadas inicialmente na RFC 1883 de dezembro de 1995, no entanto, em dezembro de 1998, esta RFC foi substituída pela RFC 2460. Como principais mudanças em relação ao IPv4 destacam-se:

- **Maior capacidade para endereçamento:** no IPv6 o espaço para endereçamento aumentou de 32 bits para 128 bits, permitindo: níveis mais específicos de agregação de endereços; identificar uma quantidade muito maior de dispositivos na rede; e implementar mecanismos de autoconfiguração. A escalabilidade do roteamento multicast também foi melhorada através da adição do campo "escopo" no endereço multicast. E um novo tipo de endereço, o anycast, foi definido;
- **Simplificação do formato do cabeçalho:** alguns campos do cabeçalho IPv4 foram removidos ou tornaram-se opcionais, com o intuito de reduzir o custo do processamento dos pacotes nos roteadores;
- **Suprimento a cabeçalhos de extensão:** as opções não fazem mais parte do cabeçalho base, permitindo um roteamento mais eficaz, limites menos rigorosos em relação ao tamanho e a quantidade de opções, e uma maior flexibilidade para a introdução de novas opções no futuro;
- **Capacidade de identificar fluxos de dados:** foi adicionado um novo recurso que permite identificar de pacotes que pertençam a determinados tráfegos de fluxos, para os quais podem ser requeridos tratamentos especiais;
- **Suprimento a autenticação e privacidade:** foram especificados cabeçalhos de extensão capazes de fornecer mecanismos de autenticação e garantir a integridade e a confidencialidade dos dados transmitidos.

4. Entendendo a distribuição dos endereços na Internet

Como não pode haver repetição de endereços, eles são um recurso que tem de ser gerenciado de forma centralizada na Internet. Desde a época da ARPANET existe, na rede, uma autoridade com o objetivo de efetuar esse controle, chamada Internet Assigned Numbers Authority, ou IANA.

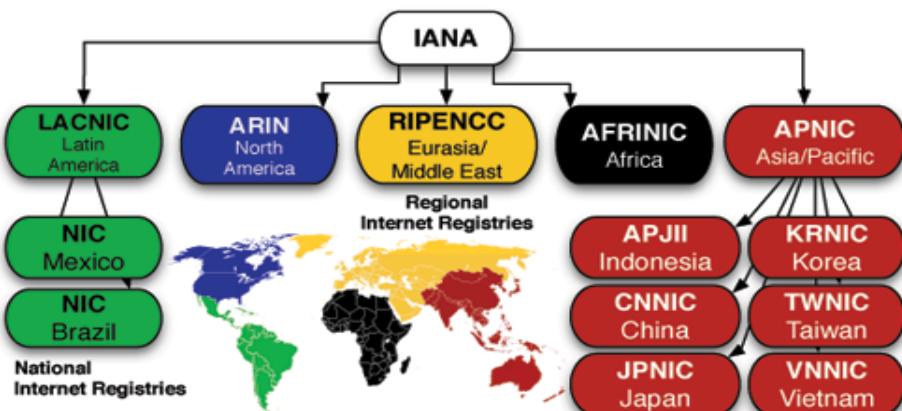


Figura 6: Estrutura para distribuição de IPs na Internet.

Fonte: <http://caida.org>

Atualmente, a função da IANA é realizada pela ICANN (Internet Corporation for Assigned Names and Numbers) e a estrutura de distribuição de IPs é hierárquica, contando também com organizações regionais, chamadas de Regional Internet Registries, ou RIRs, e, em alguns casos, estruturas nacionais, chamadas de National Internet Registries, ou NIRs.

Há cinco RIRs: o ARIN, na América do Norte, o LACNIC, na América Latina e Caribe, o RIPE, abrangendo a Europa e parte da Ásia, o AFRINIC, na África e o APNIC, na região da Ásia e Oceania. Cada uma dessas organizações é responsável por definir as regras de distribuição dos endereços em sua respectiva área de atuação, e por implementá-las. Essa definição de políticas é feita por meio de processos bottom-up, com a participação dos próprios operadores da Internet, que utilizam os recursos de numeração.

Em alguns países há entidades nacionais para a distribuição dos IPs. É o caso do Brasil, por exemplo, onde é o NIC.br quem gerencia esse recurso. Isso acontece aqui por dois motivos principais. Primeiro, por razões históricas. A Internet chegou ao Brasil muito cedo, e o LACNIC ainda não existia quando o Registro.br, hoje um departamento do NIC.br, começou a fazer a distribuição dos endereços. O NIC.br foi uma das instituições que ajudou a fundar o LACNIC. Além disso, há a questão da linguagem. Enquanto praticamente todo o restante da América Latina fala espanhol, nós falamos português.

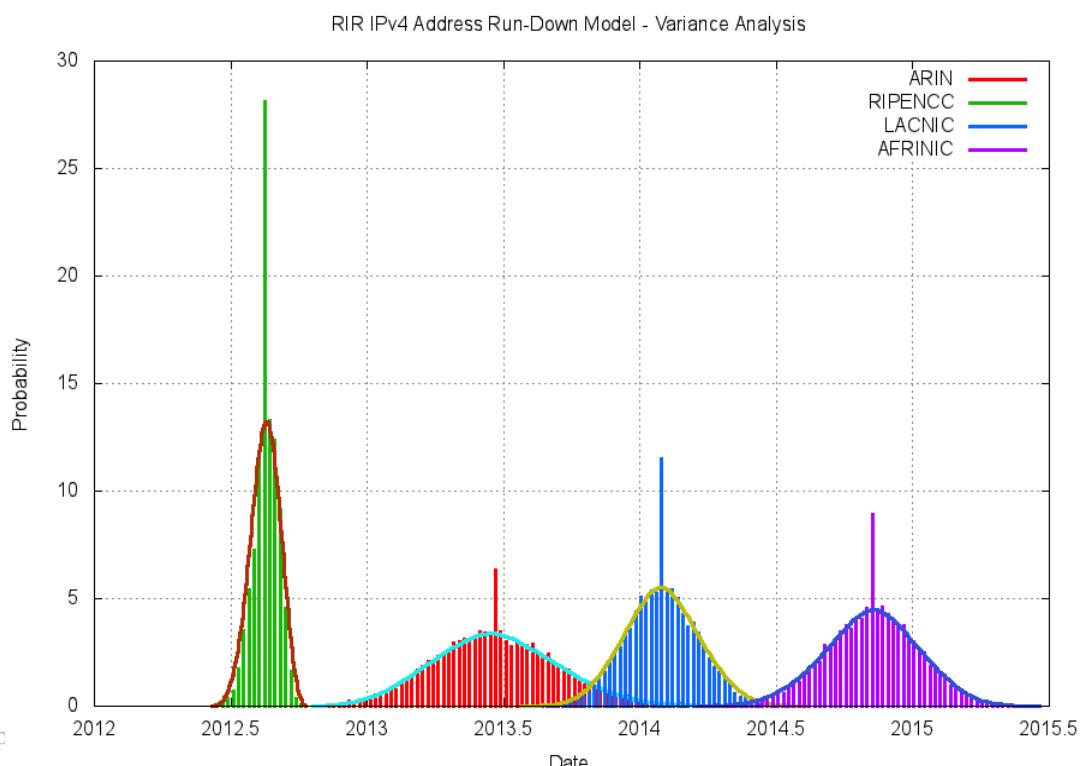


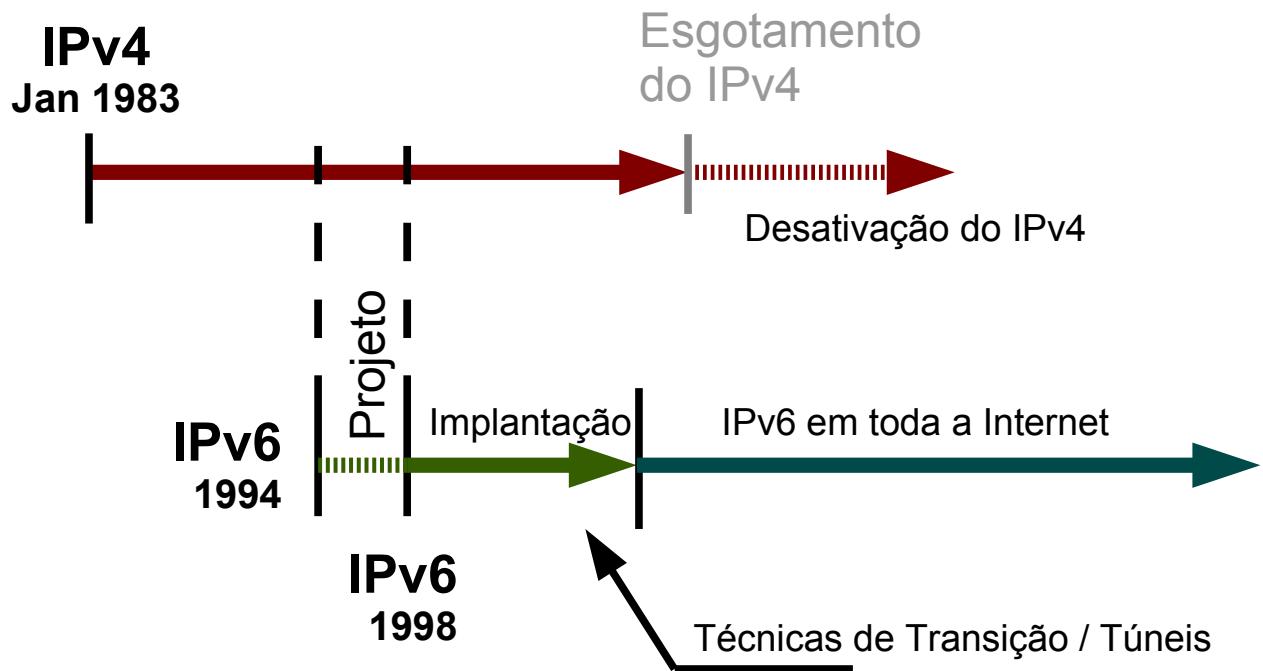
Figura 7: Esgotamento dos IPs versão 4.fonte: <http://potaroo.net>

No Brasil, quando os provedores Internet e outros Sistemas Autônomos necessitam ampliar suas redes, solicitam os recursos de numeração ao NIC.br. Este fornece os recursos enquanto tiver em seu estoque, que é compartilhado com o estoque do LACNIC. Uma vez que o estoque do LACNIC estiver com níveis muito baixos, ele solicita a IANA que lhe forneça mais endereços para gerenciar.

Para IPv4, contudo, o estoque da IANA terminou em 03 de fevereiro de 2011. O estoque do APNIC acabou pouco tempo depois, em 14 de abril de 2011. Na Figura 7 é possível visualizar as previsões atuais para que os demais estoques regionais esgotem-se. Para o Brasil, que está na região do LACNIC, a previsão é de que seja entre meados de 2013 e meados de 2014.

5. A transição do IPv4 para o IPv6

O IPv6 foi projetado de tal forma que não é compatível com o IPv4. Eles não podem interoperar diretamente. Ainda assim, o plano inicial para a transição era muito simples: ambos os protocolos podem conviver, sem problemas, nos mesmos equipamentos e softwares. O plano inicial era, então, fazer uma transição gradual, mantendo o IPv4, e adicionando o IPv6 em todos os dispositivos da Internet ao longo do tempo, de forma que, antes dos endereços livres IPv4 esgotarem-se, o IPv6 estivesse instalado em toda a Internet.

**Figura 8:** Plano inicial de transição para o IPv6

Tecnicamente, o plano ilustrado na Figura 8 era muito simples de ser executado. Contudo, o mesmo não levou em consideração aspectos econômicos e políticos em toda a sua complexidade. Embora o IPv6 tenha sido projetado para ser um protocolo melhor do que o IPv4, a característica que realmente se sobressai é a grande capacidade de endereçamento. Ele pode até ser superior em outros aspectos, mas não de forma que seja considerado um grande diferencial. Os gestores de TI viram-se ante a seguinte situação, na última década ou pouco mais: sabiam da importância do IPv6 e de sua implantação, mas sabiam também que teriam de gastar recursos para isso, sem benefícios a curto prazo, e que se não fizessem nada o problema real só se manifestaria após muito tempo. Ou seja,

poderiam adiar a implantação do IPv6, aparentemente sem consequências negativas, e assim a maioria o fez.

O atraso na implantação do IPv6 na Internet nos fez chegar em um ponto em que o esgotamento do IPv4 já é uma realidade, e o IPv6 está longe de estar difundido em toda a Internet. Nessa situação os novos usuários da rede necessitam ainda de IPs versão 4, para acessar conteúdo e serviços que ainda não implantaram o novo protocolo.

Nessa situação, são necessárias tecnologias para a transição que permitam que, ao mesmo tempo em que se implanta o IPv6, se compartilhe também endereços IPv4.

É uma situação transitória, e espera-se que dure muito pouco. Em alguns lugares pode ser mesmo que ainda possa ser evitada, caso se acelere suficientemente a implantação do IPv6 por parte dos provedores de serviço na rede.

Espera-se, atualmente (esse texto foi escrito em julho de 2012) que a transição para o IPv6 seja muito rápida. Provavelmente ele estará plenamente difundido na Internet nos próximos 2 a 4 anos.

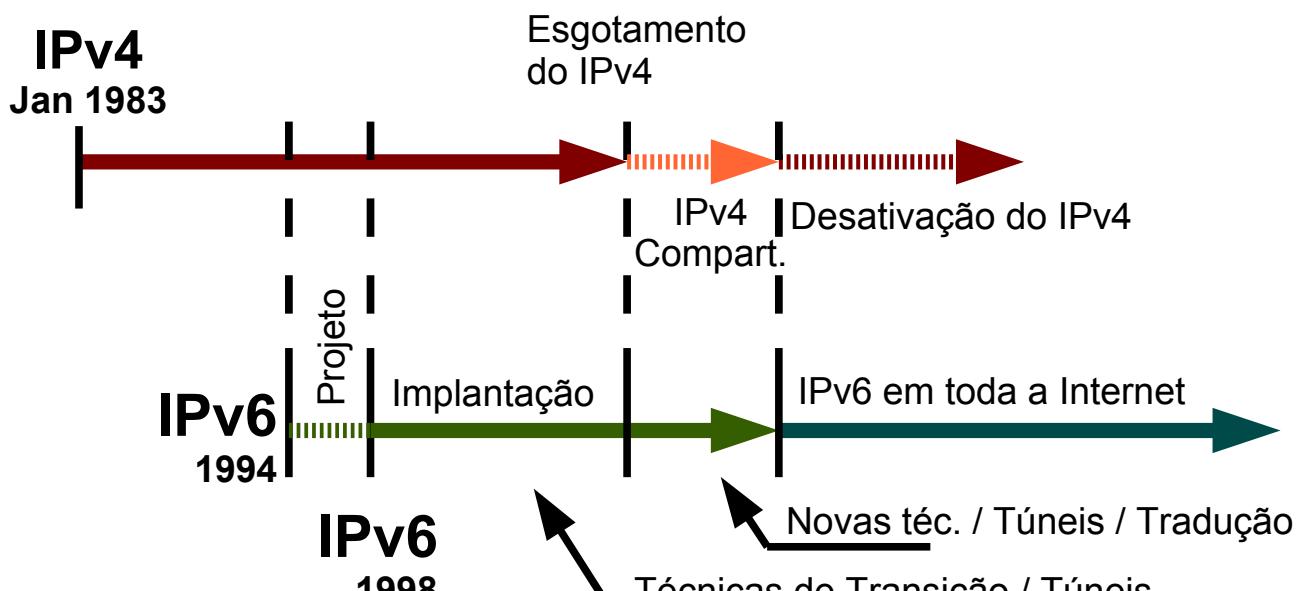
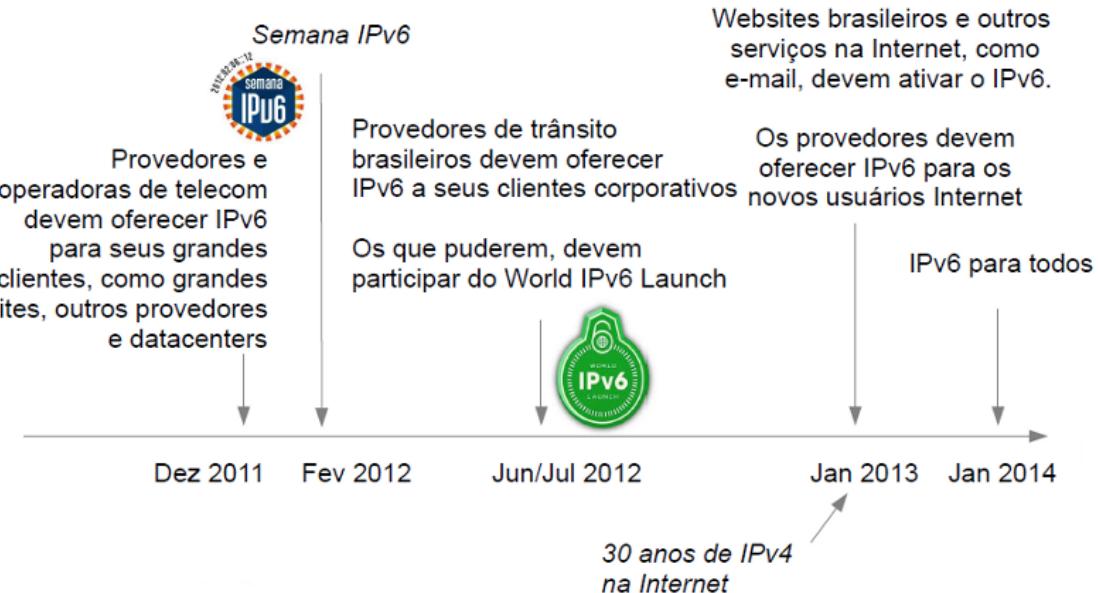


Figura 9: Novo plano de transição

No Brasil, construiu-se o cronograma ilustrado na Figura 10, como referência para a implantação do protocolo no país. Ele foi construído com base no diálogo com diversos provedores de acesso, de serviços e operadoras de telecomunicações, em diversas reuniões de coordenação ao longo dos anos de 2011 e 2012. Ele tem por trás de si, uma lógica muito simples:

Figura 10: Cronograma Brasileiro de migração para o IPv6



- Em primeiro lugar os provedores de trânsito Internet (as grandes operadoras de telecom e Sistemas Autônomos em geral que oferecem trânsito para outros ASes) devem preparar-se.
- Uma vez que os provedores de serviços e companhias em geral tenham como conseguir conectividade Internet, os serviços (sites em geral, comércio eletrônico, bancos, governos, etc) devem migrar para IPv6. Essa migração deve ser feita rapidamente. Quanto mais rápida e efetiva for essa migração, melhor será para todos: os sites não terão o risco de perderem audiência, ou terem usuários que os acessarão com potenciais problemas, e os provedores de acesso terão menos motivos para utilizarem o compartilhamento de IPv4 numa fase de transição
- Por fim os provedores de acesso devem fazer chegar o IPv6 aos usuários domésticos. Primeiro aos novos usuários, e depois àqueles já conectados à Internet via IPv4.

Com base nesse cronograma, o Comitê Gestor da Internet no Brasil faz a seguinte recomendação:

O COMITÊ GESTOR DA INTERNET NO BRASIL – CGI.br, em sua 4ª Reunião Ordinária de 2012, realizada em 18 de maio de 2012, na sede do NIC.br, e no uso das atribuições que lhe confere o Decreto nº 4.829/2003, resolve aprovar esta Resolução, da seguinte forma:

Resolução CGI.br/RES/2012/007/P – Recomendação para Implantação do Protocolo IPv6

Considerando que:

- a Internet vem se expandindo e desenvolvendo continuamente, desde sua criação, e que essa expansão necessita ser preservada, tornando universal o acesso à rede;
- que à Internet estão sendo incorporados cada vez mais, diferentes tipos de dispositivos e equipamentos;

- que o IP (Internet Protocol) é o protocolo responsável por identificar cada um dos dispositivos presentes na Internet e por encapsular toda a informação que flui pela mesma, podendo ser considerado como a base tecnológica que separa a Internet dos outros serviços existentes sobre as redes de telecomunicações, sendo, portanto, de vital importância para seu funcionamento;
- que a versão 4 do IP, ou IPv4, que vem sendo usada na Internet desde janeiro de 1983, está em vias de se esgotar, não podendo mais sustentar o crescimento da rede e desenvolvimento futuros;
- que mesmo soluções paliativas, para a compartilhamento e preservação dos endereços IPv4, e que vêm sendo usadas com sucesso desde 1994 também alcançaram o seu limite de aplicação;
- que desde 1998 com o padrão RFC 2460 o IETF (Internet Engineering Task Force) desenvolveu uma nova versão do protocolo IP, a versão 6 – IPv6;
- que o IPv6 foi testado com sucesso em diversos ambientes de laboratório e em produção;
- que hoje o suporte ao IPv6 está disponível na maioria dos equipamentos usados no núcleo das redes, que os principais sistemas operacionais o suportam, e que muitos provedores de conectividade, acesso e serviços já o implantaram com sucesso;
- que o NIC.br tem capacitado técnicos, engenheiros e administradores de redes para que implantem e operem redes com suporte ao IPv6 ao longo dos últimos anos, por meio de diversas ações, como palestras, cursos e publicação de material pertinente em português;
- que o NIC.br tem coordenado esforços com provedores de acesso, serviços e conteúdo na Internet, no sentido de realizar a implantação do IPv6 e que, como resultado destes esforços delineou-se um cronograma para nortear a implantação do IPv6 no Brasil.

Recomenda:

- que todas as redes conectadas à Internet no Brasil considerem, com a urgência necessária, a implantação do IPv6;
- que Sistemas Autônomos que provêem trânsito Internet para outros Sistemas Autônomos suportem o protocolo IPv6 à partir da segunda metade de 2012, em caráter de produção e em todas as localidades onde operam;
- que provedores de hospedagem, conteúdo e serviços na Internet, que incluem sítios Web, serviços de “e-mail”, comércio eletrônico, serviços bancários e de governo prestados pela Internet, suportem o IPv6 antes de 01 de Janeiro de 2013;
- que provedores de acesso Internet ofereçam conectividade IPv6 de forma nativa, para todos os seus novos usuários, à partir de 01 de Janeiro de 2013, juntamente com conectividade IPv4, usando sempre que possível números IPv4 válidos ou, em caso de carência, IPv4 compartilhados ou mesmo técnicas de tradução que permitam ao usuário nativo de IPv6 acesso aos serviços que só respondem a IPv4;
- que provedores de acesso Internet ofereçam suporte ao IPv6 para todos os usuários antes de 01 de Janeiro de 2014;
- que provedores de acesso não utilizem técnicas para preservação e compartilhamento de IPs versão 4 de forma isolada sem a implantação concomitante do IPv6;
- que os fabricantes de equipamentos usados na Internet, incluindo-se mas não limitados a modems, roteadores e roteadores sem fio, ofereçam equipamentos compatíveis com IPv6, à partir da segunda metade de 2012;

- que as empresas usuárias da Internet realizem a implantação do IPv6 tanto em seus serviços expostos na Internet, como em sua rede interna, conforme as datas recomendadas, de forma planejada e com a urgência possível;
- que o governo, considerando-os aqui os três poderes e em suas diversas instâncias, estabeleça normas internas com cronograma conforme as datas aqui previstas e com metas claras para a implantação do IPv6, em especial nos serviços oferecidos aos cidadãos através da Internet;
- que as universidades e centros de pesquisa, em especial os relacionados às disciplinas de redes, computação e Internet, implantem o IPv6 em suas redes com urgência.

Referências:

- RFC 1380 - IESG Deliberations on Routing and Addressing
- RFC 1918 - Address Allocation for Private Internets
- RFC 2131 - Dynamic Host Configuration Protocol
- RFC 2775 - Internet Transparency
- RFC 2993 - Architectural Implications of NAT
- RFC 3022 - Traditional IP Network Address Translator (Traditional NAT)
- RFC 3027 - Protocol Complications with the IP Network Address Translator
- RFC 4632 - Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan.
- RFC 1550 - IP: Next Generation (IPng) White Paper Solicitation
- RFC 1752 - The Recommendation for the IP Next Generation Protocol
- Migrating to IPv6 : A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks . Autor: Marc Blanchet.
- 6 Net: An IPv6 Deployment Guide.
Autor: Martin Dunmore.
- IPv6 Essentials.
Autor: Silvia Hagem.
- Global IPv6 Strategies.
Autores: Patric Grossetete; Ciprian popovicius; Fred Wetting.
- Planning and Accomplishing the IPv6 Integration: Lessons Learned from a Global Construction and Project-Management Company .
Autor: Cisco Public Information.
- Technical and Economic Assessment of Internet Protocol Version 6.
Autor: U.S. DEPARTMENT OF COMMERCE.
- Introducción a IPv6.
Autor: Roque Gagliano.
- Planificando IPv6.
Autor: Roque Gagliano.
- Deliverable D 6.2.4: Final report on IPv6 management tools, developments and tests.

Autor: 6 Net.

IPv6 Security: Are You Ready? You Better Be!

Autor: Joe Klein.

IPv6 and IPv4 Threat Comparison and Best- Practice Evaluation (v1.0)

Autores: Sean Convery; Darrin Miller.

BGP Routing Table Analysis Reports - <http://bgp.potaroo.net/>

Autores: Tony Bates; Philip Smith; Geoff Huston.

Measuring IPv6 Deployment

Autores: Geoff Huston; George Michaelson.

IPv6 at Google.

Autores: Angus Lees; Steinar H. Gunderson.

Resumo do Barômetro Cisco Banda Larga Brasil 2005-2010

Autores: Mauro Peres; João Paulo Bruder.

Tracking the IPv6 Migration. Global Insights From the Largest Study to Date on IPv6 Traffic on the Internet.

Autor: Craig Labovitz.

ICE: Uma solução geral para a travessia de NAT

Autor: José Henrique de Oliveira Varanda

RFC 1287 - Towards the Future Internet Architecture.

RFC 1296 - Internet Growth (1981-1991)

Solensky F., 'Continued Internet Growth', Proceedings of the 18th Internet Engineering Task Force, Agosto 1990, <http://www.ietf.org/proceedings/prior29/IETF18.pdf>

IANA IPv4 Address Space Registry - <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>

RFC 3330 - Special-Use IPv4 Addresses.

Capítulo 2: Cabeçalho

Nesta seção, serão apresentadas as principais características do IPv6 a começar pela análise das mudanças ocorridas na estrutura de seu cabeçalho, seguido da explicitação das diferenças entre os cabeçalhos de ambas as versões, ressaltando o que foi aprimorado no funcionamento do protocolo. Também, será detalhada a utilização dos cabeçalhos de extensão e, o porquê dela melhorar o desempenho dos roteadores.

Serão abordados os seguintes tópicos

- Cabeçalho IPv4
- Cabeçalho IPv6
 - Campos do Cabeçalho IPv6
 - Cabeçalhos de extensão
 - Hop-by-Hop
 - Destination Options
 - Routing
 - Fragmentation
 - Authentication Header e Encapsulating Security Payload
 - Aspectos dos cabeçalhos de extensão

1. Cabeçalho IPv4

Versão (Version)	Tamanho do Cabeçalho (IHL)	Tipo de Serviço (ToS)	Tamanho Total (Total Length)					
Identificação (Identification)			Flags	Deslocamento do Fragmento (Fragment Offset)				
Tempo de Vida (TTL)	Protocolo (Protocol)		Soma de verificação do Cabeçalho (Checksum)					
Endereço de Origem (Source Address)								
Endereço de Destino (Destination Address)								
Opções + Complemento (Options + Padding)								

O cabeçalho IPv4 é composto por 12 campos fixos, que podem ou não conter opções responsáveis por fazer com que o tamanho varie de 20 a 60 Bytes. Estes campos são destinados transmitir informações sobre:

- a versão do protocolo;
- o tamanho do cabeçalho e dos dados;

- a fragmentação dos pacotes;
- o tipo dos dados sendo enviados;
- o tempo de vida do pacote;
- o protocolo da camada seguinte (TCP, UDP, ICMP);
- a integridade dos dados;
- a origem e destino do pacote.

2. Cabeçalho IPv6

Versão (Version)	Classe de Tráfego (Traffic Class)	Identificador de Fluxo (Flow Label)				
Tamanho dos Dados (Payload Length)		Próximo Cabeçalho (Next Header)	Limite de Encaminhamento (Hop Limit)			
Endereço de Origem (Source Address)						
Endereço de Destino (Destination Address)						

Algumas mudanças foram realizadas no formato do cabeçalho base do IPv6 de modo a torná-lo mais simples. O número de campos foi reduzido para apenas oito e o tamanho foi fixado de 40 Bytes. Além disso, ele ficou mais flexível e eficiente com a adição de cabeçalhos de extensão que não precisam ser processados por roteadores intermediários. Tais alterações permitiram que, mesmo com um espaço de endereçamento quatro vezes maior que o do IPv4, o tamanho total do cabeçalho IPv6 fosse apenas duas vezes.

Dentre essas mudanças, destaca-se a remoção de seis dos campos existentes cabeçalho IPv4, como resultado tanto da inutilização de suas funções quanto de sua reimplementação com o uso de cabeçalhos de extensão. A figura a seguir identifica esses campos.

A primeira remoção foi a do campo “Tamanho do Cabeçalho” que tornou-se desnecessário uma vez que seu valor foi fixado. A seguir, os campos “Identificação”, “Flags”, “Deslocamento do Fragmento” e “Opções e Complementos” passaram a ter suas informações indicadas em cabeçalhos de extensão apropriados. Por fim, o campo “Soma de Verificação” foi descartado com o objetivo de deixar o protocolo mais eficiente já que outras validações são realizadas pelos protocolos das camadas superiores da rede.

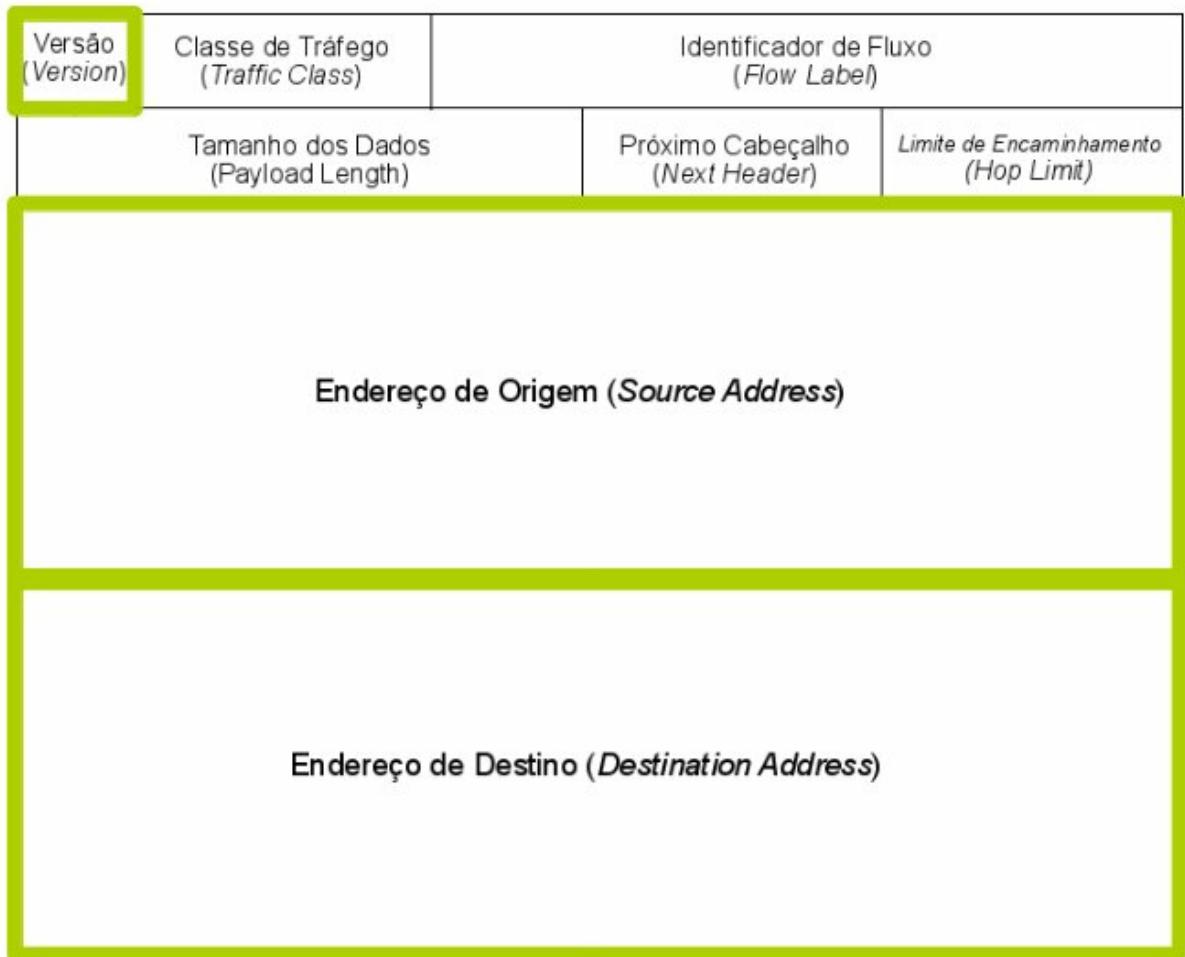
Outra alteração realizada com o intuito de agilizar o processamento foi a renomeação e reposicionamento de quatro campos conforme a tabela abaixo:

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

Além disso, o campo “Identificador de Fluxo” foi adicionado para possibilitar o funcionamento de um mecanismo extra de suporte a QoS (Quality of Service). Mais detalhes sobre este campo e mecanismo serão apresentados nas próximas seções.



Por fim, os campos “Versão”, “Endereço de Origem” e “Endereço de Destino” foram mantidos e apenas tiveram seus tamanhos alterados.



3. Campos do Cabeçalho IPv6

Versão (Version)	Classe de Tráfego (Traffic Class)	Identificador de Fluxo (Flow Label)	
	Tamanho dos Dados (Payload Length)	Próximo Cabeçalho (Next Header)	Limite de Encaminhamento (Hop Limit)
Endereço de Origem (Source Address)			
Endereço de Destino (Destination Address)			

Conforme a observado na figura acima, o cabeçalho do IPv6 está dividido nos seguintes campos:

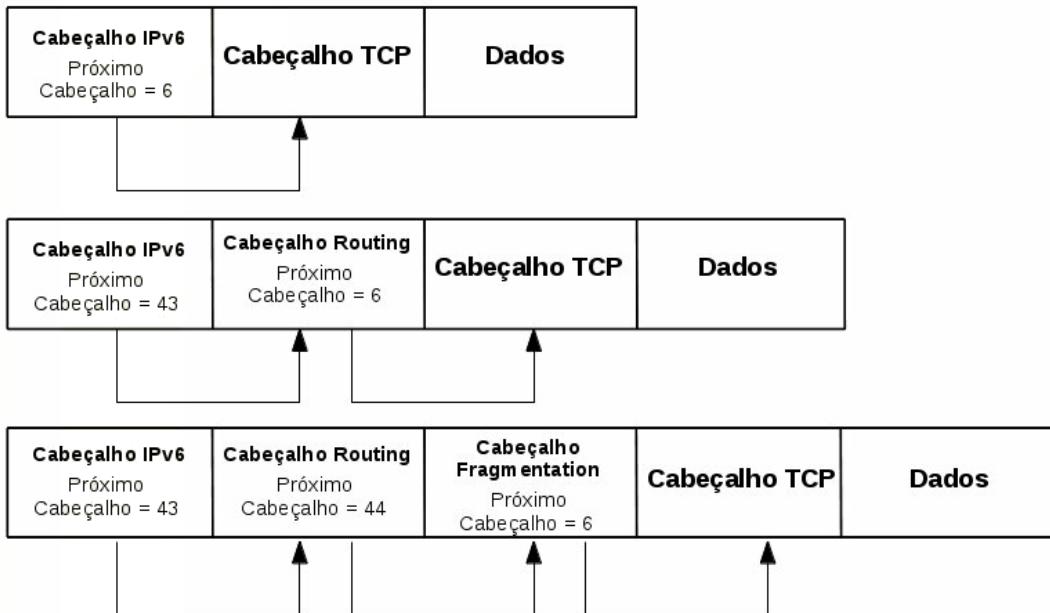
- **Versão (4 bits)** - Identifica a versão do protocolo utilizado. No caso, o valor desse campo é 6.
- **Classe de Tráfego (8 bits)** – Identifica os pacotes por classes de serviços ou prioridade. Ele provê as mesmas funcionalidades e definições do campo “Tipo de Serviço do IPv4”.
- **Identificador de Fluxo (20 bits)** – Identifica pacotes do mesmo fluxo de comunicação. Idealmente esse campo é configurado pelo endereço de destino para separar os fluxos de cada uma das aplicações e os nós intermediários de rede podem utilizá-lo de forma agregada com os endereços de origem e destino para realização de tratamento específico dos pacotes.
- **Tamanho do Dados (16 bits)** – Indica o tamanho, em Bytes, apenas dos dados enviados junto ao cabeçalho IPv6. Substituiu o campo Tamanho Total do IPv4, que indicava o tamanho do cabeçalho mais o tamanho dos dados transmitidos. Contudo, o tamanho dos cabeçalhos de extensão também são somado nesse novo campo.
- **Próximo Cabeçalho (8 bits)** – Identifica o cabeçalho de extensão que segue o atual. Ele foi renomeado (no IPv4 chamava-se Protocolo) para refletir a nova organização dos pacotes IPv6, uma vez que ele deixou de conter os valores referentes a outros protocolos, para indicar os tipos dos cabeçalhos de extensão.
- **Limite de Encaminhamento (8 bits)** – Esse campo é decrementado a cada salto de roteamento e indica o número máximo de roteadores pelos quais o pacote pode passar antes

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

- **Endereço de origem (128 bits)** – Indica o endereço de origem do pacote.
- **Endereço de Destino (128 bits)** – Indica o endereço de destino do pacote.

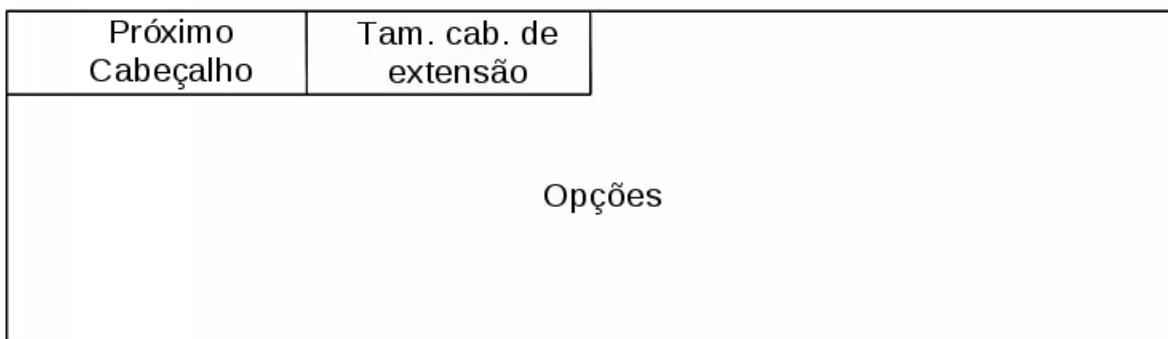
3.1. Cabeçalhos de extensão

Diferente do IPv4, que inclui no cabeçalho base todas as informações opcionais, o IPv6 trata essas informações através de cabeçalhos de extensão. Estes, localizam-se entre o cabeçalho base e o cabeçalho da camada de imediatamente acima e, não possuem quantidade ou tamanho fixo. Caso existam múltiplos cabeçalhos de extensão no mesmo pacote, eles serão adicionados em série formando uma “cadeia de cabeçalhos”. A figura abaixo exemplifica essa situação.



As especificações do IPv6 definem seis cabeçalhos de extensão: Hop-by-Hop Options, Destination Options, Routing, Fragmentation, Authentication Header e Encapsulating Security Payload.

A criação dos cabeçalhos de extensão do IPv6 teve a finalidade de aumentar a velocidade de processamento nos roteadores, visto que o único que deve ser processado em cada roteador é o Hop-



by-Hop, enquanto que os demais são tratados apenas pelo nó de destino. Além disso, novos cabeçalhos podem ser definidos no protocolo sem a necessidade alterações no cabeçalho base. O esquema abaixo mostra o template de um cabeçalho de extensão.

3.2. Hop-by-Hop

Identificado pelo valor 00 no campo Próximo Cabeçalho, o cabeçalho de extensão Hop-by-Hop deve ser colocado imediatamente após o cabeçalho base IPv6. Suas informações devem ser examinadas por todos os nós intermediários do caminho do pacote até o destino. E, em sua ausência, os roteadores não precisam processar nada além do cabeçalho base, o que agiliza o encaminhamento de pacotes.

Os seguintes campos estão presentes nesse cabeçalho:

- **Próximo Cabeçalho (1 Byte):** Identifica o tipo de cabeçalho que segue ao *Hop-by-Hop*.
- **Tamanho do Cabeçalho (1 Byte):** Indica o tamanho seu tamanho (em unidades de 8 Bytes) excluídos os oito primeiros bits.
- **Opções:** Contém uma ou mais opções e seu tamanho é variável. Neste campo, o primeiro Byte contém informações sobre como estas opções devem ser tratadas caso o nó que as esteja processando, não as reconheça. Desse byte, o valor dos primeiros dois bits especifica qual das seguintes ações a devem ser tomadas:
 - 00: ignorar e continuar o processamento.
 - 01: descartar o pacote.
 - 10: descartar o pacote e enviar uma mensagem ICMP Parameter Problem para o endereço de origem do pacote.
 - 11: descartar o pacote e enviar uma mensagem ICMP Parameter Problem para o endereço de origem do pacote, apenas se o destino não for um endereço de multicast.

O terceiro bit indica se a informação opcional pode mudar de rota (valor 1) ou não (valor 0). Até o momento existem dois tipos definidos para o cabeçalho *Hop-by-Hop*: Router Alert e Jumbogram:

- **Router Alert:** Utilizado para informar aos nós intermediários que a mensagem a ser encaminhada exige tratamento especial. Esta opção é utilizada pelos protocols MLD (Multicast Listener Discovery) e RSVP (Resource Reservation Protocol).
- **Jumbogram:** Utilizado para informar que o tamanho do pacote IPv6 é maior do que 64KB.

3.3. Destination Options

Identificado pelo valor 60 no campo Próximo Cabeçalho, o cabeçalho de extensão *Destination Options* deve ser processado apenas pelo nó de destino do pacote. A definição de seus campos é igual as do cabeçalho *Hop-by-Hop*.

Ele é utilizado no suporte ao mecanismo de mobilidade do IPv6 através da opção *Home Address*, que contém o Endereço de Origem do Nó Móvel quando este está em transito.

3.4. Routing

Próximo Cabeçalho	Tam. cab. de extensão	Tipo de Routing	Saltos restantes
Reservado			
Endereço de Origem			

Identificado pelo valor 43 no campo Próximo Cabeçalho, o cabeçalho de extensão *Routing* foi desenvolvido inicialmente para listar um ou mais nós intermediários que deveriam ser visitados até o pacote chegar ao destino, de forma semelhante às opções *Loose Source* e *Record Route* do IPv4. No entanto, esta função tornou-se obsoleta pela RFC5095 devido a problemas de segurança.

Um novo cabeçalho *Routing*, *Type 2*, foi definido para ser utilizado como parte do mecanismo de suporte a mobilidade do IPv6. Segundo essa nova definição, ele deve carregar o Endereço de Origem do Nô Móvel em pacotes enviados pelo Nô Correspondente.

As definições de cada campo desse cabeçalho são as seguintes:

- **Próximo Cabeçalho (1 Byte):** Identifica o tipo de cabeçalho que segue ao cabeçalho Routing.
- **Tamanho do Cabeçalho (1 Byte):** Indica o tamanho seu tamanho (em unidades de 8 Bytes) excluídos os oito primeiros bits.
- **Routing Type (1 Byte):** Identifica o tipo de cabeçalho Routing. Atualmente apenas o Type 2 está especificado.
- **Saltos restantes:** Definido para ser utilizado com o Routing Type 0, indica o número de saltos a serem visitados antes do pacote atingir seu destino final.
- **Endereço de Origem:** Carrega o Endereço de Origem de um Nô Móvel.

3.5. Fragmentation

Próximo Cabeçalho	Reservado	Deslocamento do Fragmento	Res	M
Identificação				

Identificado pelo valor 44 no campo Próximo Cabeçalho, o cabeçalho de extensão *Fragmentation* é utilizado quando o pacote IPv6 a ser enviado é maior que o Path MTU.

As definições de cada campo do cabeçalho são as seguintes:

- **Próximo Cabeçalho (1 Byte):** Identifica o tipo de cabeçalho que segue ao cabeçalho *Fragmentation*.
- **Deslocamento do Fragmento (13 bits):** Indica, em unidades de oito Bytes, a posição dos dados transportados pelo fragmento atual em relação ao início do pacote original.
- **Flag M (1 bit):** Se marcado com o valor 1, indica que há mais fragmentos. Se marcado com o valor 0, indica que é o fragmento final.
- **Identificação (4 Bytes):** Valor único gerado pelo nó de origem, para identificar o pacote original. É utilizado para detectar os fragmentos de um mesmo pacote.

3.6. Authentication Header e Encapsulating Security Payload

Os cabeçalhos de extensão *Authentication Header (AH)* e *Encapsulating Security Payload (ESP)*, indicados respectivamente pelos valores 51 e 52 no campo Próximo Cabeçalho, fazem parte do cabeçalho IPSec.

Embora as funcionalidades do IPSec sejam idênticas tanto no IPv4 quanto no IPv6, sua utilização com IPv6 é facilitada pelo fato de seus principais elementos integrarem essa nova versão do protocolo. Outros aspectos que também facilitam sua utilização são a inexistência de NAT IPv6 e o detalhamento dos cabeçalhos AH e ESP.

Aspectos dos cabeçalhos de extensão

Alguns aspectos sobre os cabeçalhos de extensão devem ser observados. Primeiramente, estes cabeçalhos devem ser enviados segundo uma determinada ordem com o intuito de evitar que os nós intermediários tenham que processar toda a cadeia de cabeçalhos para decidir quais eles deverão tratar. Assim, os cabeçalhos importantes para todos os nós envolvidos no roteamento devem ser colocados em antes daqueles que são relevantes apenas para o destinatário final. A vantagem, é que um nó pode parar de analisar cabeçalhos assim que encontrar algum dedicado ao destino. Isso, melhora significativamente o desempenho dos roteadores pacotes, porque, em geral, apenas o processamento do cabeçalho base é necessário. Deste modo, a sequência a ser seguida é:

Hop-by-Hop Options

Routing

Fragmentation

Authentication Header

Encapsulating Security Payload

Destination Options

Vale também observar que, se um pacote for enviado para um endereço multicast, os cabeçalhos de extensão serão examinados por todos os nós do grupo.

Em relação à flexibilidade oferecida pelos cabeçalhos de extensão, merece destaque o desenvolvimento do cabeçalho Mobility, que é utilizado por nós com suporte ao mecanismo de mobilidade IPv6.

4. Referências

RFC 2711 – IPv6 Router Alert Option

RFC 3775 – Mobility Support in IPv6 – 6.4. Type 2 Routing Header

RFC 5095 – Deprecation of Type 0 Routing Headers in IPv6

Capítulo3: Endereçamento

O protocolo IPv6 apresenta como principal característica e justificativa maior para o seu desenvolvimento, o aumento no espaço para endereçamento. Por isso, é importante conhecemos as diferenças entre os endereços IPv4 e IPv6, saber reconhecer a sintaxe dos endereços IPv6 e conhecer os tipos de endereços IPv6 existentes e suas principais características.

No IPv4, o campo do cabeçalho reservado para o endereçamento possui 32 bits. Este tamanho possibilita um máximo de 4.294.967.296 (232) endereços distintos. A época de seu desenvolvimento, está quantidade era considerada suficiente para identificar todos os computadores na rede e suportar o surgimento de novas sub-redes. No entanto, com o rápido crescimento da Internet, surgiu o problema da escassez dos endereços IPv4, motivando a criação de uma nova geração do protocolo IP.

O IPv6 possui um espaço para endereçamento de 128 bits, sendo possível obter 340.282.366.920.938.463.463.374.607.431.768.211.456 endereços (2128). Este valor representa aproximadamente 79 octilhões ($7,9 \times 10^{28}$) de vezes a quantidade de endereços IPv4 e representa, também, mais de 56 octilhões ($5,6 \times 10^{28}$) de endereços por ser humano na Terra, considerando-se a população estimada em 6 bilhões de habitantes.

Serão abordados os seguintes tópicos:

- Representação dos endereços
- Tipos de endereços IPv6
- Endereços Unicast
 - Identificadores de interface
 - Endereços especiais
- Endereços Anycast
- Endereços Multicast
- Políticas de alocação e designação

1. Representação dos endereços

Os 32 bits dos endereços IPv4 são divididos em quatro grupos de 8 bits cada, separados por “.”, escritos com dígitos decimais. Por exemplo: 192.168.0.10.

A representação dos endereços IPv6, divide o endereço em oito grupos de 16 bits, separando-os por “:”, escritos com dígitos hexadecimais (0-F). Por exemplo:

- 2001:0DB8:AD1F:25E2:CAFE:F0CA:84C1

Na representação de um endereço IPv6, é permitido utilizar tanto caracteres maiúsculos quanto minúsculos.

Além disso, regras de abreviação podem ser aplicadas para facilitar a escrita de alguns endereços muito extensos. É permitido omitir os zeros à esquerda de cada bloco de 16 bits, além de substituir uma sequência longa de zeros por “::”.

Por exemplo, o endereço **2001:0DB8:0000:0000:130F:0000:0000:140B** pode ser escrito como **2001:DB8:0:0:130F::140B** ou **2001:DB8::130F:0:0:140B**. Neste exemplo é possível observar que a abreviação do grupo de zeros só pode ser realizada uma única vez, caso contrário poderá haver ambiguidades na representação do endereço. Se o endereço acima fosse escrito como

2001:DB8::130F::140B, não seria possível determinar se ele corresponde a **2001:DB8:0:0:130F:0:0:140B**, a **2001:DB8:0:0:130F:0:140B** ou **2001:DB8:0:130F:0:0:140B**.

Esta abreviação pode ser feita também no fim ou no início do endereço, como ocorre em **2001:DB8:0:54:0:0:0:0** que pode ser escrito da forma **2001:DB8:0:54::**.

Outra representação importante é a dos prefixos de rede. Em endereços IPv6 ela continua sendo escrita do mesmo modo que no IPv4, utilizando a notação CIDR. Esta notação é representada da forma “endereço-IPv6/tamanho do prefixo”, onde “tamanho do prefixo” é um valor decimal que especifica a quantidade de bits contíguos à esquerda do endereço que compreendem o prefixo. O exemplo de prefixo de sub-rede apresentado a seguir indica que dos 128 bits do endereço, 64 bits são utilizados para identificar a sub-rede.

- Prefixo **2001:db8:3003:2::/64**
- Prefixo global **2001:db8::/32**
- ID da sub-rede **3003:2**

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.

Com relação a representação dos endereços IPv6 em URLs (Uniform Resource Locators), estes agora passam a ser representados entre colchetes. Deste modo, não haverá ambiguidades caso seja necessário indicar o número de uma porta juntamente com a URL. Observe os exemplos a seguir:

- [http://\[2001:12ff:0:4::22\]/index.html](http://[2001:12ff:0:4::22]/index.html)
- [http://\[2001:12ff:0:4::22\]:8080](http://[2001:12ff:0:4::22]:8080)

2. Tipos de endereços IPv6

Existem no IPv6 três tipos de endereços definidos:

- **Unicast** – este tipo de endereço identifica uma única interface, de modo que um pacote enviado a um endereço unicast é entregue a uma única interface;
- **Anycast** – identifica um conjunto de interfaces. Um pacote encaminhado a um endereço anycast é entregue a interface pertencente a este conjunto mais próxima da origem (de acordo com distância medida pelos protocolos de roteamento). Um endereço anycast é utilizado em comunicações de um-para-um-de-muitos.
- **Multicast** – também identifica um conjunto de interfaces, entretanto, um pacote enviado a um endereço multicast é entregue a todas as interfaces associadas a esse endereço. Um endereço multicast é utilizado em comunicações de um-para-muitos.

Diferente do IPv4, no IPv6 não existe endereço broadcast, responsável por direcionar um pacote para todos os nós de um mesmo domínio. No IPv6, essa função foi atribuída à tipos específicos de endereços multicast.

2.1. Endereços Unicast

Os endereços unicast são utilizados para comunicação entre dois nós, por exemplo, telefones VoIPV6, computadores em uma rede privada, etc., e sua estrutura foi definida para permitir agregações com prefixos de tamanho flexível, similar ao CIDR do IPv4.

Existem alguns tipos de endereços unicast IPv6: Global Unicast; Unique-Local; e Link-Local por exemplo. Existem também alguns tipos para usos especiais, como endereços IPv4 mapeados em IPv6, endereço de loopback e o endereço não-especificado, entre outros.

- **Global Unicast** – equivalente aos endereços públicos IPv4, o endereço global unicast é globalmente roteável e acessível na Internet IPv6. Ele é constituído por três partes: o prefixo de roteamento global, utilizado para identificar o tamanho do bloco atribuído a uma rede; a identificação da sub-rede, utilizada para identificar um enlace em uma rede; e a identificação da interface, que deve identificar de forma única uma interface dentro de um enlace. Sua estrutura foi projetada para utilizar os 64 bits mais à esquerda para identificação da rede e os 64 bits mais à direita para identificação da interface. Portanto, exceto casos específicos, todas as sub-redes em IPv6 tem o mesmo tamanho de prefixo, 64 bits (/64), o que possibilita $2^{64} = 18.446.744.073.709.551.616$ dispositivos por sub-rede.

Atualmente, está reservada para atribuição de endereços a faixa 2000::/3 (001), que corresponde aos endereços de **2000::** a **3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff**. Isto representa 13% do total de endereços possíveis com IPv6, o que nos permite criar

$$2^{(64-3)} = 2.305.843.009.213.693.952 \text{ (2,3} \times 10^{18} \text{) sub-redes (/64) diferentes ou } 2^{(48-3)} = 35.184.372.088.832 \text{ (3,5} \times 10^{13} \text{) redes /48.}$$

- **Link Local** – podendo ser usado apenas no enlace específico onde a interface está conectada, o endereço link local é atribuído automaticamente utilizando o prefixo FE80::/64. Os 64 bits reservados para a identificação da interface são configurados utilizando o formato IEEE EUI-64. Vale ressaltar que os roteadores não devem encaminhar para outros enlaces, pacotes que possuam como origem ou destino um endereço link-local
- **Unique Local Address (ULA)** – endereço com grande probabilidade de ser globalmente único, utilizado apenas para comunicações locais, geralmente dentro de um mesmo enlace ou conjunto de enlaces. Um endereço ULA não deve ser roteável na Internet global. Um endereço ULA, criado utilizando um ID global alocado pseudo-randomicamente, é composto das seguintes partes:
 - **Prefixo:** FC00::/7.
 - **Flag Local (L):** se o valor for 1 (FD) o prefixo é atribuído localmente. Se o valor for 0 (FC), o prefixo deve ser atribuído por uma organização central (ainda a definir).
 - **Identificador global:** identificador de 40 bits usado para criar um prefixo globalmente único.
 - **Identificador da Interface:** identificador da interface de 64 bits.

Deste modo, a estrutura de um endereço ULA é **FDUU:UUUU:UUUU::** onde U são os bits do identificador único, gerado aleatoriamente por um algoritmo específico.

Sua utilização permite que qualquer enlace possua um prefixo /48 privado e único globalmente. Deste modo, caso duas redes, de empresas distintas por exemplo, sejam interconectadas, provavelmente não haverá conflito de endereços ou necessidade de renumerar a interface que o esteja usando. Além disso, o endereço ULA é independente de provedor, podendo ser utilizado na comunicação dentro do enlace mesmo que não haja uma conexão com a Internet. Outra vantagem, é que seu prefixo pode ser facilmente bloqueado, e caso um endereço ULA seja anunciado acidentalmente para fora do enlace, através de um roteador ou via DNS, não haverá conflito com outros endereços.

2.1.1. Identificadores de interface

Os identificadores de interface (IID), utilizados para distinguir as interfaces dentro de um enlace, devem ser únicos dentro do mesmo prefixo de sub-rede. O mesmo IID pode ser usado em múltiplas interfaces em um único nó, porém, elas dever estar associadas a diferentes sub-redes.

Normalmente utiliza-se um IID de 64 bits, que pode ser obtido de diversas formas. Ele pode ser configurado manualmente, a partir do mecanismo de autoconfiguração stateless do IPv6, a partir de servidores DHCPv6 (stateful), ou formados a partir de uma chave pública (CGA). Estes métodos serão detalhados no decorrer deste curso.

Embora eles possam ser gerados randomicamente e de forma temporária, recomenda-se que o IID seja construído baseado no endereço MAC da interface, no formato EUI-64.

Um IID baseado no formato EUI-64 é criado da seguinte forma:

- Caso a interface possua um endereço MAC de 64 bits (padrão EUI-64), basta complementar o sétimo bit mais a esquerda (chamado de bit U/L – Universal/Local) do endereço MAC, isto é, se for 1, será alterado para 0; se for 0, será alterado para 1. Caso a interface utilize um endereço MAC de 48 bits (padrão IEEE 802), primeiro adiciona-se os dígitos hexadecimais FF-FE entre o terceiro e quarto Byte do endereço MAC (transformando no padrão EUI-64), e em seguida, o bit U/L é complementado. Por exemplo:
- Se endereço MAC da interface for:
 - 48-1E-C9-21-85-0C
- adiciona-se os dígitos FF-FE na metade do endereço:
 - 48-1E-C9-FF-FE-21-85-0C
- complementa-se o bit U/L:
 - 48 = 01001000
 - 01001000 → 01001010
 - 01001010 = 4A
- IID = 4A-1E-C9-FF-FE-21-85-0C

Um endereço link local atribuído à essa interface seria **FE80::4A1E:C9FF:FE21:850C**.

2.1.2 Endereços especiais

Existem alguns endereços IPv6 especiais utilizados para fins específicos:

- **Endereço Não-Especificado (Unspecified):** é representado pelo endereço **0:0:0:0:0:0:0:0** ou **::0** (equivalente ao endereço IPv4 unspecified **0.0.0.0**). Ele nunca deve ser atribuído a nenhum nó, indicando apenas a ausência de um endereço. Ele pode, por exemplo, ser utilizado no campo Endereço de Origem de um pacote IPv6 enviado por um host durante o processo de inicialização, antes que este tenha seu endereço exclusivo determinado. O endereço unspecified não deve ser utilizado como endereço de destino de pacotes IPv6;
- **Endereço Loopback:** representado pelo endereço unicast **0:0:0:0:0:0:0:1** ou **::1** (equivalente ao endereço IPv4 loopback **127.0.0.1**). Este endereço é utilizado para referenciar a própria máquina, sendo muito utilizado para teste internos. Este tipo de endereço não deve ser atribuído a nenhuma interface física, nem usado como endereço de origem em pacotes IPv6 enviados para outros nós. Além disso, um pacote IPv6 com um endereço loopback como destino não pode ser enviado por um roteador IPv6, e caso um pacote recebido em uma interface possua um endereço loopback como destino, este deve ser descartado;

- **Endereços IPv4-mapeado:** representado por **0:0:0:0:FFFF:wxyz** ou **::FFFF:wxyz**, é usado para mapear um endereço IPv4 em um endereço IPv6 de 128-bit, onde **wxyz** representa os 32 bits do endereço IPv4, utilizando dígitos decimais. É aplicado em técnicas de transição para que nós IPv6 e IPv4 se comuniquem. Ex. **::FFFF:192.168.100.1**.

Algumas faixas de endereços também são reservadas para uso específicos:

- **2002::/16:** prefixo utilizado no mecanismo de transição 6to4;
- **2001:0000::/32:** prefixo utilizado no mecanismo de transição TEREDO;
- **2001:db8::/32:** prefixo utilizado para representar endereços IPv6 em textos e documentações.

Outros endereços, utilizados no início do desenvolvimento do IPv6 tornaram-se obsoletos e não devem mais ser utilizados:

- **FEC0::/10:** prefixo utilizado pelos endereços do tipo site local, desenvolvidos para serem utilizados dentro de uma rede específica sem a necessidade de um prefixo global, equivalente aos endereços privados do IPv4. Sua utilização foi substituída pelos endereços ULA;
- **::wxyz:** utilizado para representar o endereço IPv4-compatível. Sua função é a mesma do endereço IPv4-mapeado, tornando-se obsoleto por desuso;
- **3FFE::/16:** prefixo utilizado para representar os endereços da rede de teste 6Bone. Criada para ajudar na implantação do IPv6, esta rede foi desativada em 6 de junho de 2006 (06/06/06).

2.2. Endereços Anycast

Um endereço IPv6 anycast é utilizado para identificar um grupo de interfaces, porém, com a propriedade de que um pacote enviado a um endereço anycast é encaminhado apenas a interface do grupo mais próxima da origem do pacote.

Os endereços anycast são atribuídos a partir da faixa de endereços unicast e não há diferenças sintáticas entre eles. Portanto, um endereço unicast atribuído a mais de uma interface transforma-se em um endereço anycast, devendo-se neste caso, configurar explicitamente os nós para que saibam que lhes foi atribuído um endereço anycast. Além disso, este endereço deve ser configurado nos roteadores como uma entrada separada (prefixo /128 – host route).

Este esquema de endereçamento pode ser utilizado para descobrir serviços na rede, como servidores DNS e proxies HTTP, garantindo a redundância desses serviços. Também pode-se utilizar para fazer balanceamento de carga em situações onde múltiplos hosts ou roteadores provêm o mesmo serviço, para localizar roteadores que fornecem acesso a uma determinada sub-rede ou para localizar os Agentes de Origem em redes com suporte a mobilidade IPv6.

Todos os roteadores devem ter suporte ao endereço anycast Subnet-Router. Este tipo de endereço é formado pelo prefixo da sub-rede e pelo IID preenchido com zeros (ex.: 2001:db8:cafe:dad0::/64). Um pacote enviado para o endereço Subnet-Router será entregue para o roteador mais próximo da origem dentro da mesma sub-rede.

Também foi definido um endereço anycast para ser utilizado no suporte a mobilidade IPv6. Este tipo de endereço é formado pelo prefixo da sub-rede seguido pelo IID **dfff:ffff:ffff:fffe** (ex.: **2001:db8::dfff:ffff:ffff:fffe**). Ele é utilizado pelo Nó Móvel, quando este precisar localizar um Agente Origem em sua Rede Original.

2.3 Endereços Multicast

Endereços multicast são utilizados para identificar grupos de interfaces, sendo que cada interface pode pertencer a mais de um grupo. Os pacotes enviados para esses endereços são entregues a todos as interfaces que compõe o grupo.

No IPv4, o suporte a multicast é opcional, já que foi introduzido apenas como uma extensão ao protocolo. Entretanto, no IPv6 é requerido que todos os nós suportem multicast, visto que muitas funcionalidades da nova versão do protocolo IP utilizam esse tipo de endereço.

Seu funcionamento é similar ao do broadcast, dado que um único pacote é enviado a vários hosts, diferenciando-se apenas pelo fato de que no broadcast o pacote é enviado a todos os hosts da rede, sem exceção, enquanto que no multicast apenas um grupo de hosts receberá esse pacote.

Deste modo, a possibilidade de transportar apenas uma cópia dos dados a todos os elementos do grupo, a partir de uma árvore de distribuição, pode reduzir a utilização de recurso de uma rede, bem como otimizar a entrega de dados aos hosts receptores. Aplicações como videoconferência, distribuição de vídeo sob demanda, atualizações de softwares e jogos on-line, são exemplos de serviços que vêm ganhando notoriedade e podem utilizar as vantagens apresentadas pelo multicast.

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

As flags são definidas da seguinte forma:

O primeiro bit mais a esquerda é reservado e deve ser marcado com 0;

- **Flag R:** Se o valor for 1, indica que o endereço multicast “carrega” o endereço de um Ponto de Encontro (Rendezvous Point). Se o valor for 0, indica que não há um endereço de Ponto de Encontro embutido;
- **Flag P:** Se o valor for 1, indica que o endereço multicast é baseado em um prefixo de rede. Se o valor for 0, indica que o endereço não é baseado em um prefixo de rede;
- **Flag T:** Se o valor for 0, indica que o endereço multicast é permanente, ou seja, é atribuído pela IANA. Se o valor for 1, indica que o endereço multicast não é permanente, ou seja, é atribuído dinamicamente.

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

- 1 – abrange apenas a interface local;
- 2 – abrange os nós de um enlace;
- 3 – abrange os nós de uma sub-rede
- 4 – abrange a menor área que pode ser configurada manualmente;
- 5 – abrange os nós de um site;
- 8 – abrange vários sites de uma mesma organização;
- E – abrange toda a Internet;
- 0, F – reservados;
- 6, 7, 9, A, B, C, D – não estão alocados.

Deste modo, um roteador ligado ao backbone da Internet não encaminhará pacotes com escopo menor do que 14 (E em hexa), por exemplo. No IPv4, o escopo de um grupo multicast é especificado através do campo TTL do cabeçalho. A lista abaixo apresenta alguns endereços multicast permanentes:

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

O endereço *multicast solicited-node* identifica um grupo multicast que todos os nós passam a fazer parte assim que um endereço unicast ou anycast lhes é atribuído. Um endereço solicited-node é formado agregando-se ao prefixo **FF02::1:FF00:0000/104** os 24 bits mais à direita do identificador da interface, e para cada endereço unicast ou anycast do nó, existe um endereço multicast solicited-node correspondente.

Em redes IPv6, o endereço solicited-node é utilizado pelo protocolo de Descoberta de Vizinhança para resolver o endereço MAC de uma interface. Para isso, envia-se uma mensagem Neighbor Solicitation para o endereço solicited-node. Com isso, apenas as interfaces registradas neste grupo examinam o pacote. Em uma rede IPv4, para se determinar o endereço MAC de uma interface, envia-se uma mensagem ARP Request para o endereço broadcast da camada de enlace, de modo que todas as interfaces do enlace examinam a mensagem.

Com o intuito de reduzir o número de protocolos necessários para a alocação de endereços multicast, foi definido um formato estendido de endereço multicast, que permite a alocação de endereços baseados em prefixos unicast e de endereços SSM (*source-specific multicast*).

Em endereços baseados no prefixo da rede, a *flag P* é marcada com o valor 1. Neste caso, o uso do campo escopo não altera, porém, o escopo deste endereço multicast não deve exceder o escopo do prefixo *unicast* “carregado” junto a ele. Os 8 bits após o campo escopo, são reservados e devem ser marcados com zeros. Na sequência, há 8 bits que especificam o tamanho do prefixo da rede indicado nos 64 bits que os seguem. Caso o prefixo da rede seja menor que 64 bits, os bits não utilizados no campo tamanho do prefixo, devem ser marcados com zeros. O campo identificador do grupo utiliza os 32 bits restantes. Note que, em um endereço onde a *flag P* é marcada com o valor 1, a *flag T* também deve ser marcada com o valor 1, pois este não representa um endereço definido pela IANA.

No modelo tradicional de *multicast*, chamado de *any-source multicast* (ASN), o participante de um grupo *multicast* não controla de que fonte deseja receber os dados. Com o SSM, uma interface pode

registrar-se em um grupo *multicast* e especificar as fontes de dados. O SSM pode ser implementado utilizando o protocolo MLDv2 (*Multicast Listener Discovery version 2*).

Para um endereço SSM, as *flags* P e T são marcadas com o valor 1. Os campos tamanho do prefixo e o prefixo da rede são marcados com zeros, chegando ao prefixo **FF3X::/32**, onde X é o valor do escopo. O campo Endereço de Origem do cabeçalho IPv6 identifica o dono do endereço multicast. Todo endereço SSM tem o formato **FF3X::/96**.

Os métodos de gerenciamento dos grupos multicast serão abordados no próximo módulo deste curso.

Também é importante destacar algumas características relacionadas ao endereço apresentadas pela nova arquitetura do protocolo IPv6. Assim como no IPv4, os endereços IPv6 são atribuídos às interfaces físicas, e não aos nós, de modo que cada interface precisa de pelo menos um endereço *unicast*. No entanto, é possível atribuir a uma única interface múltiplos endereços IPv6, independentemente do tipo (*unicast, multicast ou anycast*) ou sub-tipo (*loopback, link local, 6to4, etc.*). Deste modo um nó pode ser identificado através de qualquer endereço das suas interfaces, e com isso, torna-se necessário escolher entre seus múltiplos endereços qual utilizará como endereço de origem e destino ao estabelecer uma conexão.

Para resolver esta questão, foram definidos dois algoritmos, um para selecionar o endereço de origem e outro para o de destino. Esses algoritmos, que devem ser implementados por todos os nós IPv6, especificam o comportamento padrão desse nós, porém não substituem as escolhas feitas por aplicativos ou protocolos da camada superior.

Entre as regras mais importantes destacam-se:

- Pares de endereços do mesmo escopo ou tipo têm preferência;
- O menor escopo para endereço de destino tem preferência (utiliza-se o menor escopo possível);
- Endereços cujo tempo de vida não expirou tem preferência sobre endereços com tempo de vida expirado;
- Endereços de técnicas de transição (ISATAP, 6to4, etc.) não podem ser utilizados se um endereço IPv6 nativo estiver disponível;
- Se todos os critérios forem similares, pares de endereços com o maior prefixo comum terão preferência;
- Para endereços de origem, endereços globais terão preferência sobre endereços temporários;
- Em um Nô Móvel, o Endereço de Origem tem preferência sobre um Endereço Remoto.

Estas regras devem ser utilizadas quando não houver nenhuma outra especificação. As especificações também permitem a configuração de políticas que possam substituir esses padrões de preferências com combinações entre endereços de origem e destino.

3. Políticas de alocação e designação

Na hierarquia das políticas de atribuição, alocação e designação de endereços, cada RIR recebe da IANA um bloco /12 IPv6.

O bloco **2800::/12** corresponde ao espaço reservado para o LACNIC alocar na América Latina. O NIC.br por sua vez, trabalha com um /16 que faz parte deste /12.

A alocação mínima para ISPs é um bloco /32, no entanto, alocações maiores podem ser feitas mediante apresentação de justificativa de utilização. Um aspecto importante que merece destaque é que diferente do IPv4, com IPv6 a utilização é medida em relação ao número de designações de

blocos de endereços para usuários finais, e não em relação ao número de endereços designados aos usuários finais.

4. Recomendação do NIC.br

O NIC.br recomenda utilizar:

- **/64 a /56 para usuários domésticos:** Para usuários móveis pode-se utilizar /64, pois normalmente apenas uma rede é suficiente. Para usuários residenciais recomenda-se redes maiores. Se o provedor optar por, num primeiro momento, oferecer apenas /64 para usuários residenciais, ainda assim recomenda-se que no plano de numeração se reserve um /56.
- **/48 para usuários corporativos.** Empresas muito grandes podem receber mais de um bloco /48.

Para planejar a rede é preciso considerar que para cada rede física ou VLAN com IPv6 é preciso reservar um /64. Esse é o tamanho padrão e algumas funcionalidades, como a autoconfiguração dependem dele. É preciso considerar também a necessidade de expansão futura, assim como a necessidade de agregação nos protocolos de roteamento.

Capítulo 4: Funcionalidades Básicas do IPv6

Este capítulo tem como objetivo apresentar as funcionalidades básicas do IPv6, mostrando aspectos teóricos de implementação e exemplos. No decorrer do texto também serão feitas algumas comparações em relação ao IPv4, procurando realçar as principais diferenças e seus motivos.

O capítulo está estruturado em subtópicos dispostos na seguinte maneira:

1. Internet Control Message Protocol version 6 (ICMPv6): Descreve detalhes e características sobre o ICMPv6, protocolo fundamental para execução das principais funcionalidades básicas do IPv6.

2. Neighbor Discovery Protocol (NDP): Descreve detalhes e características do NDP, protocolo desenvolvido com base no ICMPv6.

3. Funcionalidades Básicas baseadas no NDP

3.1. Duplication Address Detection (DAD): Apresenta o funcionamento da detecção de endereços duplicados, mecanismo utilizado para evitar que dois dispositivos contenham o mesmo endereço IP. Situação que ocasionaria problema em roteamento dos pacotes.

3.2. Address Resolution: Apresenta o funcionamento da resolução de endereços de camada dois que permite o estabelecimento de uma comunicação entre dois nós.

3.3. Router Discovery: Apresenta o funcionamento da descoberta de roteadores numa rede. A partir dela, é possível tanto propagar características da rede para todos os dispositivos pertencentes como, também, informá-los de um gateway para a sua tabela de rotas.

3.4. Prefix Discovery: Apresenta o funcionamento da descoberta de prefixos numa rede. Dessa maneira, os dispositivos saberão se deverão encaminhar seus pacotes a um roteador ou ao enlace, dependendo do destino.

3.5 Parameter Discovery: Apresenta o funcionamento da descoberta de parâmetros de rede e de Internet para a autoconfiguração dos dispositivos, permitindo assim a comunicação com outros dispositivos.

3.6. Neighbor Unreachability Detection: Apresenta o funcionamento da detecção vizinhos inacessíveis numa rede, mecanismo que possibilita a criação de uma nova rota caso haja algum problema durante uma comunicação.

3.7. Redirect: Apresenta o funcionamento do redirecionamento de comunicação. Numa rede, caso existam duas rotas para o mesmo destino e a origem escolha a pior delas, o roteador pode avisar a origem que ela deve enviar os pacotes pelo outro caminho para que a distância até o destino seja reduzida.

4. Funcionalidades Básicas com foco no mecanismo de autoconfiguração

4.1. Autoconfiguração: Descreve os modos de operação e as informações que podem ser enviadas para autoconfiguração.

4.2. Autoconfiguração de endereços Stateless: interna do dispositivo: Explica o procedimento para criação e adição do endereço de link local utilizada por todos os nós de uma rede IPv6.

4.3. Autoconfiguração Stateless: Router Advertisement: Mostra a transmissão stateless(sem registro dos dados divulgados) de prefixos e características da rede via Router Advertisements para que os dispositivos de uma rede se autoconfigurem.

4.4. Autoconfiguração: DHCPv6

5. Estado dos endereços: Mostra o ciclo de vida de um endereço IPv6 focando nos estados e seus respectivos tempos.

6. Referências: Apresenta base de informações para o texto.

1. Internet Control Message Protocol version 6 (ICMPv6)

ICMPv6 é uma versão atualizada do ICMPv4 que foi desenvolvida para ser utilizada em conjunto com o IPv6 como parte substancial de sua arquitetura. Sua implementação, portanto, é obrigatória em todos os nós da rede com suporte à IPv6.

Embora esta nova versão possua as mesmas funcionalidades de sua predecessora, como reportar erros no processamento de pacotes, realizar diagnósticos e enviar mensagens sobre as características da rede, ambas não são compatíveis e possuem diferenças significativas.

Uma das principais diferenças é que o ICMPv6 assume funções de protocolo que existem isoladamente no IPv4. Tal mudança foi projetada com o intuito de reduzir a multiplicidade de protocolos, que é prejudicial, uma vez que piora a coerência e aumenta o tamanho das implementações nos diversos dispositivos. As função assumidas se referem às funcionalidades dos seguintes protocolos integrantes do IPv4:

- ARP (*Address Resolution Protocol*), cujo o objetivo é mapear os endereços físicos através de endereços lógicos.
- RARP (*Reverse Address Resolution Protocol*), que possui funcionalidade inversa a do ARP, ele é responsável por mapear os endereços lógicos em endereços físicos.
- IGMP (*Internet Group Management Protocol*), que atua com o gerenciamento de membros de grupos multicast.

É importante notar, que tanto o ARP quanto RARP são protocolos que operam entre as camadas 2 e 3 do modelo ISO/OSI, pois, eles não dependem de pacotes IP. Já o ICMPv6 é funciona inteiramente na camada 3, sendo encapsulado em pacotes IP. Isso significa que firewalls que operam na camada de rede exigem atenção extra com o IPv6, já que podem bloquear funções extremamente básicas como a descoberta de vizinhos e a autoconfiguração.

Uma outra diferença que se convém ressaltar é a utilização do ICMPv6 pelos subsequentes protocolos e funcionalidades:

- MLD (*Multicast Listener Discovery*), que opera com o gerenciamento de grupos multicast.
- NDP (*Neighbor Discovery Protocol*), que é responsável por identificar e aprender características de rede da vizinhança.
- *Path MTU(Maximum Transfer Unity) Discovery*, que trabalha no processo de descoberta do menor MTU no caminho de comunicação entre dois nós.
- *Mobility Support*, que cuida do gerenciamento de endereços de origem de host dinamicamente.
- Autoconfiguração Stateless, que permite a aquisição de endereços IP globais sem o uso de DHCP.

Deve-se ter em mente que, de forma geral, o ICMPv6 é muito mais importante para o funcionamento do IPv6, do que o ICMP é para o funcionamento do IPv4.

A seguir serão apresentadas os principais elementos sua especificação: código de identificação, localização, formato de seus pacotes e suas classes.

1.1. Código de Identificação e Localização

O pacote ICMPv6 é identificado no cabeçalho IPv6 pelo valor 58 no campo *Next Header*. Caso não existam cabeçalhos de extensão, ele se localiza logo após os cabeçalho base. A Figura 1 apresenta um esquema de como fica a cadeia de cabeçalhos.

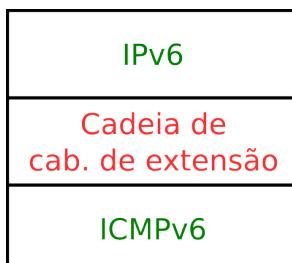


Figura 1: Localização do Protocolo ICMPv6

1.2. Formato do pacote

O ICMPv6 possui um cabeçalho de estrutura simples, baseado em quatro campos básicos:

- *Type* (8 bits): especifica o tipo da mensagem e, assim, determina o formato do corpo da mensagem (campo Data). Um exemplo é o valor 2, que indica a mensagem “Packet Too Big”.
- *Code* (8 bits): apresenta algumas informações adicionais sobre o motivo da mensagem. Um exemplo de seu uso seria a indicação motivo de uma falha de conexão entre dois dispositivos. Numa mensagem “Destination Unreachable”, o valor 0 representa a não existência de rota para o destino.
- *Checksum* (16 bits): é utilizado na detecção de dados corrompidos no cabeçalho ICMPv6 e em parte do cabeçalho IPv6.
- *Data*: mostra as informações relativas ao tipo da mensagem, podendo ser desde diagnósticos de rede até erros. Seu tamanho é variável de acordo com a mensagem, porém, não pode exceder o tamanho de MTU mínimo do IPv6 (1280 bits).

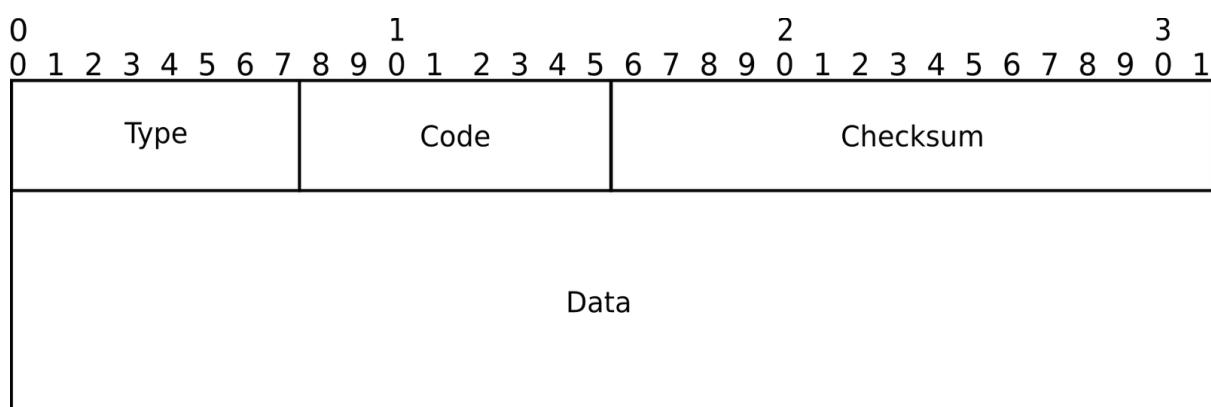


Figura 2: Formato do Pacote ICMPv6

1.3. Classes

Devido ao amplo conjunto de informações que podem ser transmitidas por meio dos pacotes ICMPv6, foram designadas duas classes para categorizar as mensagens: Erro e Informação. As duas tabelas a seguir mostram os tipos mensagens pertencentes a cada uma dessas classes.

Mensagens de Erro:

Type			Code		Extra
Valor	Nome	Descrição	Valor	Descrição	RFC
1	Destination Unreachable	Indica falhas na entrega do pacote	0	Sem rota para o destino	RFC2463, RFC4443
			1	Comunicação com o destino proibida administrativamente	
			2	Além do escopo do endereço de origem	
			3	Endereço não acessível	
			4	Porta não acessível	
			5	Endereço de origem falho na política de ingressão ou egressão	
			6	Rota para o destino rejeitada	
			7	Erro no cabeçalho de roteamento de origem	
2	Packet Too Big	Indica que o tamanho do pacote ultrapassou o limite do enlace.	0	-	RFC2463, RFC4443
3	Time Exceeded	Indica que o limite de encaminhamento ou o tempo de remontagem do pacote foi excedido.	0	Límite de encaminhamento excedido no tráfego	RFC2463, RFC4443
			1	Tempo de remontagem de fragmento excedido	
4	Parameter Problem	Indica erro em algum campo do cabeçalho IPv6 ou que o tipo indicado no próximo cabeçalho não foi reconhecido.	0	Campo errado do cabeçalho encontrado	RFC2463, RFC4443
			1	Encontrado um tipo do Next header não reconhecido	
			2	Encontrado um IPv6 option não reconhecido	
127	--	Reservado para expansão das mensagens de erro	--	--	RFC4443
255	--	Reservado para expansão das mensagens de erro	--	--	RFC4443

Mensagem de Informação:

Valor	Type	Descrição	Valor	Code	Extra
128	Echo Request	Utilizadas no comando ping	0	--	RFC2463,
129	Echo Reply		0	--	RFC4443
130	Multicast Listener Query	Utilizadas no gerenciamento de grupos multicast	0	--	
131	Multicast Listener Report		0	--	RFC2710
132	Multicast Listener Done		0	--	
133	Router Solicitation		0	--	
134	Router Advertisement	Utilizadas com o protocolo Descoberta de vizinhança	0	--	
135	Neighbor Solicitation		0	--	RFC2461
136	Neighbor Advertisement		0	--	
137	Redirect Message		0	--	
138	Router Renumbering	Utilizada no mecanismo de re-endereçamento de roteadores	0	Comando renumeração de roteadores	
			1	Resultado de renumeração de roteadores	RFC2894
			255	Reseta o numero de sequencia	
139	ICMP Node Information Query	Utilizadas para descobrir informações sobre nomes e endereços, são atualmente limitadas a ferramentas de diagnóstico, depuração e gestão de rede	0	Transmiti um campo que contém um endereço ipv6 que é o objetivo da query	
			1	Transmiti um campo que contém um nome que é o objetivo da query.	
			2	Transmiti um campo que contém um endereço ipv4 que é o objetivo da query	RFC4620
140	ICMP Node Information Reponse		0	Uma resposta realizada corretamente	
			1	O dispositivo responsável por fornecer a resposta se recusou a responder	
			2	O Qtype da query é desconectada pelo dispositivo que deve fornecer a resposta	
141	Inverse ND Solicitation Message	Utilizadas em uma extensão do protocolo de descoberta de vizinhança	0	--	RFC3122
142	Inverse ND Advertisement Message		0	--	
143	Version2 Multicast Listener Report	Utilizada no gerenciamento de grupos multicast	--	--	RFC3810
144	Ha Address Discovery Request Message		0	--	
145	HA Address Discovery Reply Message	Utilizadas no mecanismo de mobilidade IPv6	0	--	RFC3775
146	Mobile Prefix Solicitation		0	--	
147	Mobile Prefix Advertisement		0	--	
148	Certification Path Solicitation Message	Utilizadas pelo protocolo SEND	--	--	
149	Certification Path Advertisement Message		--	--	RFC3971
150	--	Utilizada experimentalmente com protocolos de mobilidade Seamoby	--	--	RFC4065
151	Multicast Router Advertisement	Utilizada pelo mecanismo multicast router discovery	--	--	
152	Multicast Router Scolitation		--	--	RFC4286
153	Multicast Router Termination		--	--	
154	FMIPv6	Utilizada pelo protocolo de mobilidade FAST Handovers	--	--	RFC5568

Além desses valores tabelados, os tipos 100, 101, 200 e 201 foram designados para uso experimental, enquanto os tipos do 102 até o 126 não são utilizados.

2. Neighbor Discovery Protocol (NDP)

O protocolo de descoberta de vizinhança (*Neighbor Discovery Protocol*) foi desenvolvido com a finalidade de resolver os problemas de interação entre os nós vizinhos de uma rede. De forma geral, ele é utilizado para que os dispositivos de uma rede consigam verificar a presença uns dos outros, determinar os endereços de seus vizinhos, encontrar roteadores e manter informações atualizadas sobre rotas a serem utilizadas na transmissão de pacotes. Para tanto, ele atua sobre dois aspectos primordiais da comunicação IPv6, a autoconfiguração de nós e a transmissão de pacotes.

No caso da autoconfiguração de nós, este protocolo fornece suporte para a realização de três funcionalidades:

- *Parameter Discovery*: atua na descoberta, por um nó, de informações sobre o enlace (como MTU) e a Internet (como hop limit).
- *Address Autoconfiguration*: trabalha com a autoconfiguração stateless de endereços na interfaces de um nó.
- *Duplicate Address Detection*: opera em um nó com a finalidade de descobrir se o endereço que se deseja configurar já está sendo utilizado por um outro nó na rede.

Já, no caso da transmissão de pacotes entre nós, ele contribui com o funcionamento de seis processos:

- *Router discovery*: trabalha com a descoberta dos roteadores pertencentes ao enlace.
- *Prefix discovery*: implementa a descoberta de prefixos de redes do enlace, cuja a finalidade é decidir para onde os pacotes serão direcionados numa comunicação (se para um roteador específico ou direto para um nó do enlace)

- *Address resolution*: descobre o endereço físico de uma interface de rede através de seu endereço lógico IPv6.
- *Neighbor Unreachability Detection*: permite que os nós descubram se um vizinho é ou continua sendo alcançável. Isso é necessário uma vez que diversos problemas podem acontecer tanto nos nós como na própria rede.
- *Redirect*: possibilita que o roteador informe a um nó uma rota melhor para ser utilizada no envio de pacotes a um determinado destino.
- *Next-hop Determination*: algoritmo para mapear endereços IP vizinhos para os quais pacotes devem ser enviados segundo seus endereços de destino.

2.1. Mensagens

O NDP foi construído com base nas seguintes mensagens do ICMPv6 para a realização de suas tarefas:

- Router Solicitation (RS), tipo 133.
- Router Advertisement (RA), tipo 134.
- Neighbor Solicitation (NS), tipo 135.
- Neighbor Advertisement (NA), tipo 136.
- Redirect, tipo 137.

Nas próximas subseções, cada uma dessas mensagens será apresentada de forma a mostrar sua importância para o protocolo, a forma como é utilizada e, a descrição de seus respectivos campos.

2.1.1. Router Solicitation (RS)

A mensagem *Router Solicitation* é enviada por um dispositivo para requisitar que roteadores da rede imediatamente se apresentem, através da resposta Router Advertisement.

Sua importância vem da necessidade da descoberta instantânea de informações (como rotas, MTU, hop limit e outras) que são disponibilizadas por roteadores. De tempos em tempos, roteadores enviam esses dados a todos os nós do enlace, contudo esse processo pode ser demasiadamente lento para um dispositivo que queira se comunicar logo. Portanto essa mensagem serve para solicitar ao roteador que o mesmo envie uma mensagem extra imediatamente. Geralmente essa situação acontece quando uma máquina tenta se conectar ou reconectar a uma rede (no momento em que ele habilita sua interface) e, por isso, ainda desconhece quaisquer detalhes das configurações do enlace e da Internet.

O pacote transmitido deve conter as seguintes características:

- O endereço de destino do cabeçalho IPv6 deve ser *All-Router (FF02::2) multicast Group*, uma vez que, o dispositivo não conhece os roteadores e suas informações.
- O endereço de origem do cabeçalho IPv6 deve ser o *unicast link local* da interface que está enviando a mensagem caso ele exista e, caso contrário, ele não deverá ser especificado, ou seja, utilizar-se o endereço (::).
- O campo *Hop Limit* do cabeçalho IPv6 deve ser marcado com o valor 255 para que o protocolo fique imune ao recebimento de mensagem vindas de equipamentos externos ao enlace.
- O campo *Type* do cabeçalho ICMPv6 deve ser padronizado com o valor 133.
- O campo *Code* do cabeçalho ICMPv6 deve ser padronizado com o valor 0, não apresentando outras opções.
- O campo *Data* do cabeçalho ICMPv6 deve ser dividido em dois subcampos:

- *Reserved* de 32 bits: não é utilizado e deve ser inicializado com 0 na transmissão. O receptor irá ignorar esse campo.
- *Options* de tamanho variável: pode conter dados extras para auxiliar nas funcionalidades básicas. No momento, apenas a opção *Source Link-Layer Address* está implementada, contudo no futuro é possível que existam outras. Caso seja transmitido uma opção não implementada para esse tipo de mensagem, ela será ignorada pelo receptor.

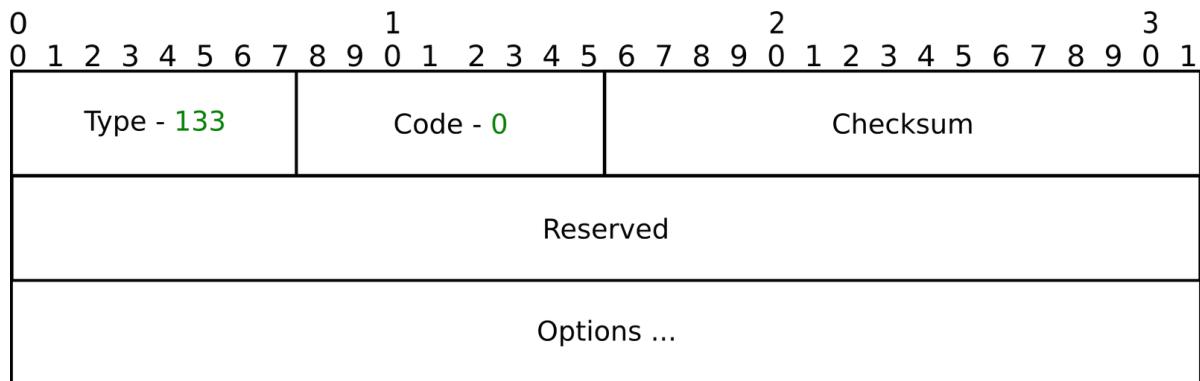


Figura 3: Formato do Pacote Router Solicitation

2.1.2. Router Advertisement (RA)

A mensagem *Router Advertisement* é enviada periodicamente ou em resposta a mensagem *Router Solicitation*, por um roteador para anunciar sua presença no enlace.

Sua importância provém do caráter informativo dessa mensagem. Além de anunciar o roteador como alternativa para rota de tráfego no enlace, ela também contém dados (como prefixos, MTU, DNS e outros) para que os nós realizem autoconfiguração. Nenhum outro dispositivo da rede possui tais dados de forma persistente, o que faz dos roteadores os únicos responsáveis por disseminá-los.

O pacote transmitido deve conter as seguintes características:

- O endereço de destino do cabeçalho IPv6 depende do motivo que originou a mensagem.
 - Caso seja periódico, ele deverá ser *All-Node (FF02::1) multicast address*.
 - Caso seja uma resposta, ele deverá ser um *unicast link local*, ou seja, direcionado para o dispositivo requisitante.
- O endereço de origem do cabeçalho IPv6 é o *unicast link local* da interface do roteador que está enviando a mensagem.
- O campo *Hop Limit* do cabeçalho IPv6 é marcado com o valor 255 para que o protocolo fique imune ao recebimento de mensagem vindas de equipamentos externos ao enlace.
- O campo *Type* do cabeçalho ICMPv6 é padronizado com o valor 134.
- O campo *Code* do cabeçalho ICMPv6 é padronizado com o valor 0, não apresentando outras opções.
- O campo *Data* do cabeçalho ICMPv6 é dividido em várias subcampos:
 - *Cur Hop Limit* de 8 bits: indica ao nó o valor que deve ser utilizado no campo *Hop limit* do cabeçalho IPv6 quando ele for enviar um pacote. O valor 0 indicará que o roteador não sugere nenhum valor default ao nó.

- *Managed Address Configuration Flag (M)* de 1 bit: indica se o nó deve utilizar a autoconfiguração de endereços stateful, DHCPv6, (1) ou não(0).
- *Other Stateful Configuration Flag (O)* de 1 bit: indica se o nó deve receber informações adicionais via DHCPv6, (1) ou não (0). Caso a Flag M esteja setada, a opção marcada nessa flag será ignorada.
- *Mobile IPv6 Home Agent Flag (H)* de 1 bit: indica se o roteador que está enviando essa mensagem também funciona como *Mobile IPv6 home agent no enlace* (1) ou não(0).
- *Router Selection Preferences (Prf)* de 2 bits: indica qual nível de preferencia deste roteador em relação aos outros roteadores da vizinhança. Os possíveis valores que esse campo pode assumir são:
 - 01: prioridade alta (High);
 - 00: prioridade média (medium) utilizada como default;
 - 11: prioridade baixa (low);
 - 10: reservado e não deve ser utilizada;
- *Neighbor Discovery Proxy Flag (P)* de 1 bit: determina se a mensagem enviada foi repassada por um proxy (1) ou não (0).
- *Reserved (Res)* de 2 bits: não é utilizado e deve ser inicializado com 0 na transmissão. O receptor irá ignorar esse campo.
- *Router Lifetime* de 16 bits: marca o tempo em segundos que o roteador deve ser considerado como roteador default. Caso o valor do campo seja 0, o roteador não pode ser adicionado à lista de roteamento default do nó.
- *Reachable Time* de 32 bits: determina o tempo máximo, em milisegundos, que um nó pode assumir que o roteador é alcançável depois de receber uma mensagem de confirmação de acessibilidade. Esse campo serve para auxiliar na funcionalidade *Neighbor Unreachability Detection*, que será explicada posteriormente.
- *Retrans Timer* de 32 bits: contém o intervalo de tempo, em milisegundos, que deve existir entre retransmissões de mensagens *Neighbor Solicitation*. Esse campo serve para auxiliar em duas funcionalidades a *Address Resolution* e a *Neighbor Unreachability Detection*, que serão explicadas posteriormente.
- *Options* de tamanho variável: pode conter dados extras para auxiliar nas funcionalidades básicas. No momento só estão implementados as opções *Source Link Layer Address*, *MTU*, *Prefix Information*, *Route Information* e *Recursive DNS Server*, contudo no futuro é possível que novas opções sejam criadas. Caso seja transmitido uma opção não implementada, o campo será ignorado.

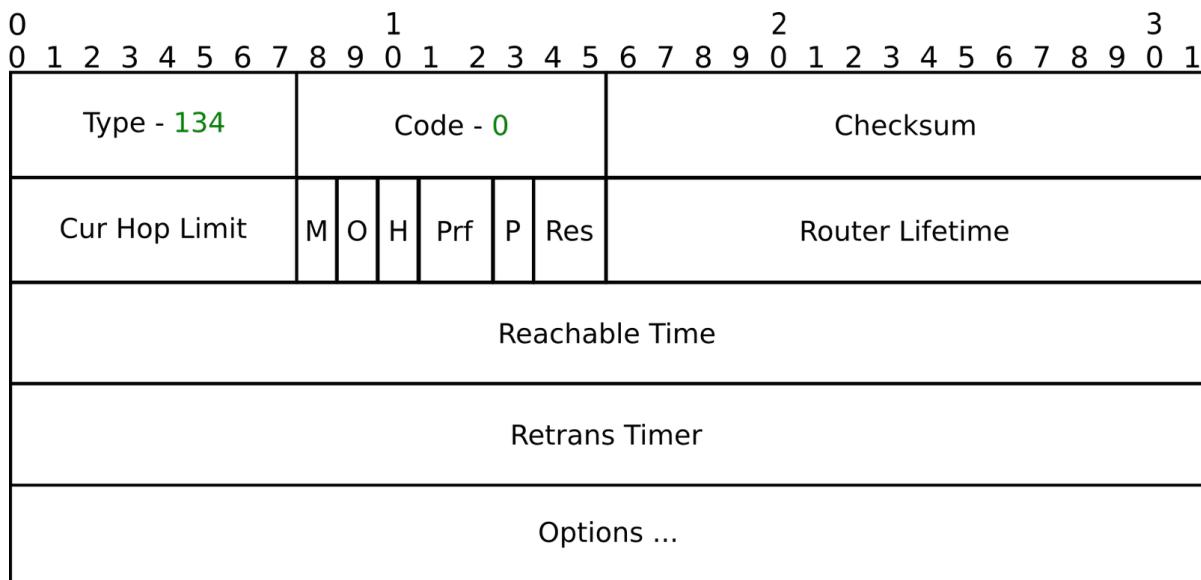


Figura 4: Formato do Pacote Router Advertisement

2.1.3. Neighbor Solicitation (NS)

A mensagem *Neighbor Solicitation* é enviada por um dispositivo para solicitar que um determinado vizinho se apresente imediatamente através da uma resposta do tipo *Neighbor Advertisement*. Por causa dessa característica, ela é utilizada para suprir três necessidades básicas da comunicação em redes IPv6.

A primeira consiste na descoberta de um endereço físico através de um endereço lógico. Nesse caso, a resposta ao *Neighbor Solicitation* conterá o endereço requisitado. No IPv4, o *Address Resolution Protocol* (ARP) realiza a mesma função.

A segunda consiste no teste de acessibilidade de nós vizinhos do enlace. Nesse caso, a mensagem pode ser enviada para verificar se determinado endereço lógico existe ou se ainda responde.

A terceira é a detecção de endereços IPv6 duplicados na vizinhança. A mensagem, nesse caso, serve para que um equipamento descubra se existe alguma interface de rede já configurada com o endereço lógico que se deseja assumir.

O pacote transmitido deve conter as seguintes características:

- O endereço de destino do cabeçalho IPv6 depende do propósito da origem da mensagem:
 - Caso seja para a descoberta de um endereço físico ou para a detecção de um endereço IPv6 duplicado, ele será preenchido com o *Solicited Node Multicast Address* ($FF02::1:FFXX:XXXX$), uma vez que não se conhece todas as informações do destino.
 - Caso seja para verificar a acessibilidade, ele será preenchido com um *unicast (link-local ou global)*, já que as informações sobre o destino são previamente conhecidas.
- O endereço de origem do cabeçalho IPv6 depende do motivo da origem da mensagem:
 - Para a descoberta de um endereço físico ou para a verificação de acessibilidade, ele será o *unicast (link-local ou global)* da interface do dispositivo requisitante.
 - Para detecção de endereços duplicados, ele não será especificado (::).
- O campo *Hop Limit* do cabeçalho IPv6 é marcado com o valor 255 para que o protocolo fique imune ao recebimento de mensagens vindas de equipamentos externos ao enlace.

- O campo *Type* do cabeçalho ICMPv6 é padronizado com o valor 135.
- O campo *Code* do cabeçalho ICMPv6 é padronizado com o valor 0, não apresentando outras opções.
- O campo *Data* do cabeçalho ICMPv6 é dividido em três subcampos:
 - *Reserved* de 32 bits: não é utilizado e deve ser inicializado com 0 na transmissão. O receptor irá ignorar esse campo.
 - *Target Address* de 128 bits: contém o endereço IPv6 do destino da mensagem *Neighbor Solicitation*. Ao contrário do campo de destino do cabeçalho ICMPv6, este campo não pode conter endereços multicast.
 - *Options* de tamanho variável: pode conter dados extras para auxiliar nas funcionalidades básicas. No momento só está implementado a opção *Source Link-Layer Address*, que serve para especificar o endereço MAC do solicitante, contudo no futuro existe a possibilidade de que outras sejam criadas. Caso seja transmitido uma opção não implementada, o campo será ignorado pelo receptor.

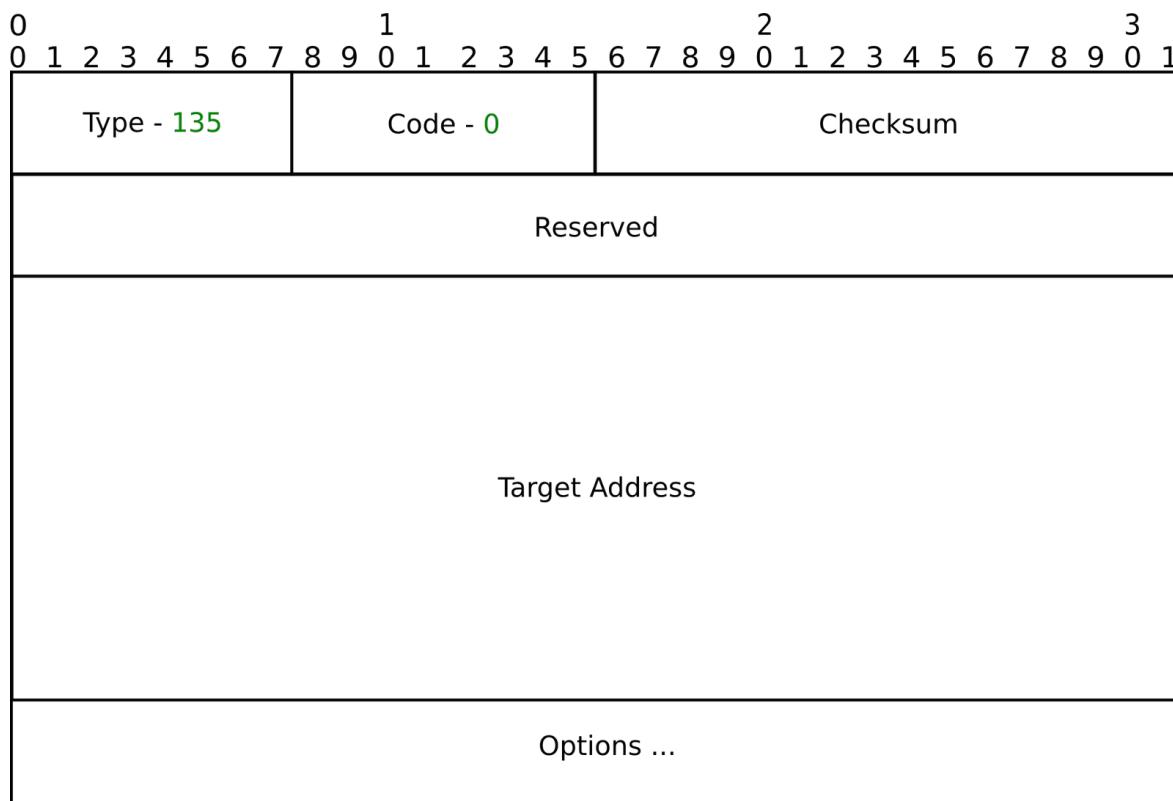


Figura 5: Formato do Pacote Neighbor Solicitation

2.1.4. *Neighbor Advertisement (NA)*

A mensagem *Neighbor Advertisement* é enviada tanto em resposta a uma mensagem *Neighbor Solicitation* quanto para anunciar, espontaneamente, a mudança de alguma característica de um dispositivo na rede.

Assim como a mensagem *Neighbor Solicitation*, essa mensagem também é utilizada para auxiliar nas funcionalidades de resolução de endereços físicos, no teste de acessibilidade de um nó vizinho e na detecção de endereços duplicados.

O pacote transmitido deve conter as seguintes características:

- O endereço de destino do cabeçalho IPv6 depende do propósito que originou a mensagem:
 - Para anunciar uma mudança ou para responder uma mensagem que possua o endereço de origem não especificado, o campo deverá ser o *All-Node (FF02::1) multicast address*, uma vez que ou pretende-se informar todos os nós vizinhos ou a origem é desconhecida.
 - Caso seja resposta a uma mensagem que possua endereço de origem, o campo deverá ser o endereço *unicast (link-local ou global)* especificado, uma vez que as informações sobre o destino foram previamente fornecidas pela mensagem *Neighbor Solicitation*.
- O endereço de origem do cabeçalho IPv6 deve ser o *unicast* da interface pela qual a mensagem será enviada.
- O campo *Hop Limit* do cabeçalho IPv6 deve ser marcado com o valor 255 para que o protocolo fique imune ao recebimento de mensagens vindas de equipamentos externos ao enlace.
- O campo *Type* do cabeçalho ICMPv6 deve ser padronizado com o valor 136.
- O campo *Code* do cabeçalho ICMPv6 deve ser padronizado com o valor 0, não apresentando outras opções.
- O campo *Data* do cabeçalho ICMPv6 deve ser dividido em seis subcampos:
 - *Router Flag (R)* de 1 bit: indica se origem é um roteador (1) ou não (0). Esse campo auxilia na funcionalidade *Neighbor Unreachability Detection*, que será explicada posteriormente.
 - *Solicited Flag (S)* de 1 bit: indica se a mensagem foi enviada por causa de uma mensagem *Neighbor Solicitation* com origem especificada (1) ou não (0). Esse campo auxilia na funcionalidade *Neighbor Unreachability Detection*, que será explicada posteriormente.
 - *Override Flag (O)* de 1 bit: indica se o destinatário deve sobrescrever o endereço físico armazenado anteriormente no cache de vizinhança (*Neighbor cache*) (1) ou não (0). Caso não haja nenhuma informação registrada, independente do valor da flag, o cache receberá uma entrada nova.
 - *Reserved de 29 bits*: não é utilizado e deve ser inicializado com 0 na transmissão. O receptor irá ignorar esse campo.
 - *Target Address* de 128 bits, depende de duas situações:
 - Caso seja uma resposta, o campo deverá conter o endereço IPv6 do nó que está respondendo.
 - Caso seja um anúncio de mudança, o campo deverá conter o endereço IPv6 que possuirá mudança no endereço físico.
 - *Options* de tamanho variável: pode conter dados extras para auxiliar nas funcionalidades básicas. No momento só está implementada a opção *Target link-layer address*, que pode especificar o endereço MAC para onde a mensagem será enviada, contudo no futuro outras podem ser criadas. Caso seja transmitido uma opção não implementada, o campo será ignorado.

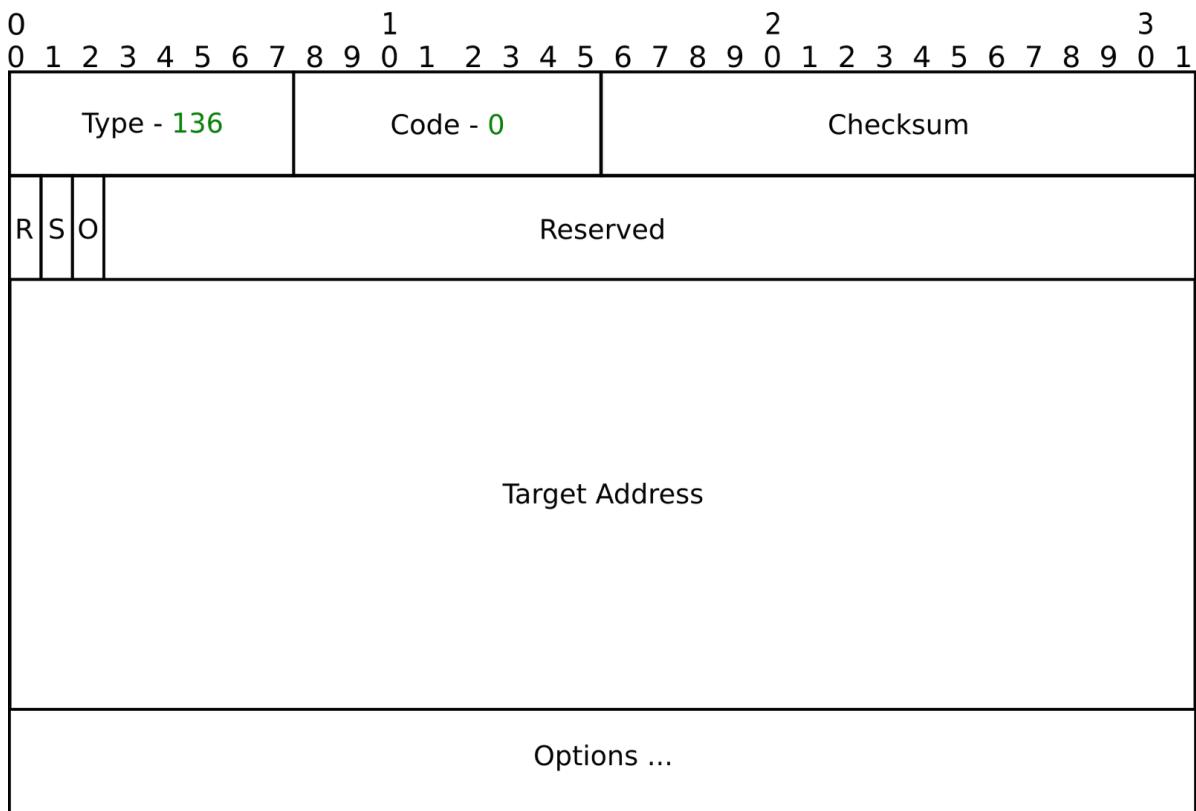


Figura 6: Formato do Pacote Neighbor Advertisement

2.1.5. Redirect

A mensagem Redirect é enviada por roteadores para informar, a um nó da rede, da existência de uma rota mais favorável à comunicação com determinado destino. Em outras palavras, ele envia o endereço do nó da rede que deve ser usado para a transmissão de pacotes ao destino em questão.

O pacote transmitido deve conter as seguintes características:

- O endereço de destino do cabeçalho IPv6 deve ser o endereço *unicast (link local ou global)* do dispositivo que requisitou uma comunicação através deste roteador.
- O endereço de origem do cabeçalho IPv6 tem que ser o *link local address* da interface do roteador.
- O campo *Hop Limit* do cabeçalho IPv6 tem que ser marcado com o valor 255 para que o protocolo fique imune ao recebimento de mensagens vindas de equipamentos externos ao enlace.
- O campo *Type* do cabeçalho ICMPv6 deve ser padronizado com o valor 137.
- O campo *Code* do cabeçalho ICMPv6 deve ser padronizado com o valor 0, não apresentando outras opções.
- O campo *Data* do cabeçalho ICMPv6 deve ser dividido em quatro subcampos:
 - *Reserved* de 32 bits: não é utilizado e deve ser inicializado com 0 na transmissão. O receptor irá ignorar esse campo.
 - *Target Address* de 128 bits, depende de duas situações:

- Caso o destino da comunicação seja o mesmo que o redirecionado, ele deverá conter o endereço IPv6 do destino e deve repetir o campo *Destination Address* do cabeçalho ICMPv6.
- Caso contrário, ele deverá conter o endereço IPv6 de link local do roteador que será o primeiro passo na nova rota.
- *Destination Address* de 128 bits: contém o endereço IPv6 do destino da comunicação que será redirecionado para outro nó.
- *Options* de tamanho variável: pode conter dados extras para auxiliar nas funcionalidades básicas. No momento só estão implementado o Target link-layer address e o Redirect Header, contudo no futuro será possível que existam outras possibilidades. Caso seja transmitida uma opção não implementada para esse tipo de mensagem, ela será ignorada pelo receptor.

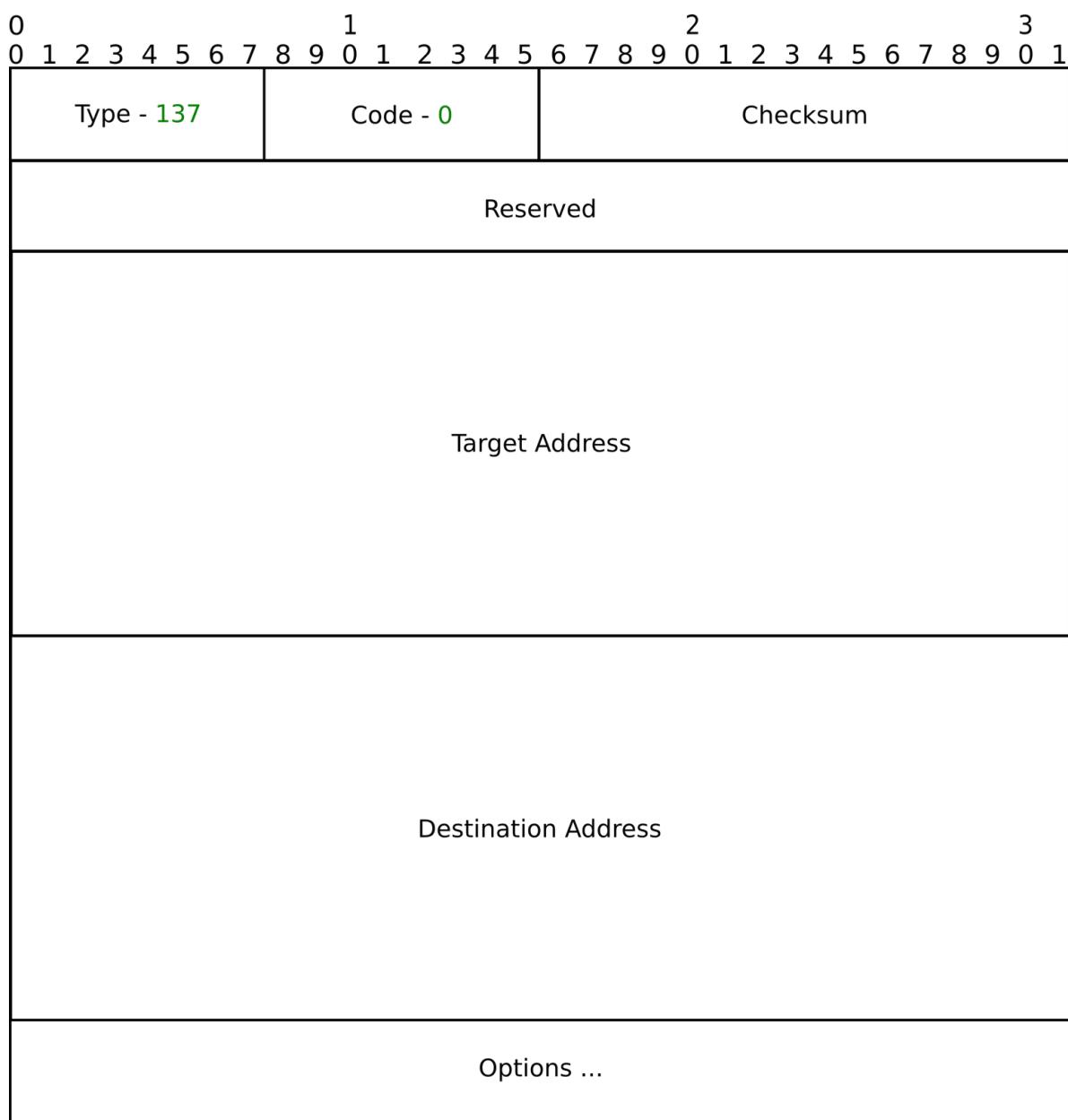


Figura 7: Formato do Pacote Redirect

2.2. Campo opções nas mensagens (*Options*)

As mensagens pertencentes ao protocolo Neighbor Discovery podem ou não incluir dados opcionais para agregar informações relevantes e assim auxiliar nas funcionalidades básicas.

Todas as opções possuem uma estrutura básica de dois campos de 8 bits, o *Type* e o *Length*. O primeiro serve para indicar qual é a opção que está sendo transmitida. Já o segundo serve para demarcar o tamanho, incluindo os campos *type* e *length*, do campo opcional.

Atualmente existem várias opções com distintas funções, porém só algumas serão detalhadas por serem mais utilizadas nas funcionalidades básicas do IPv6. Para se obter mais informações sobre esse campo deve pesquisar RFCs respecitivas sobre o assunto. No Apêndice estão listadas algumas que poderão ser úteis.

2.2.1. Source Link Layer Address

Sua função é carregar o endereço físico do nó de origem da mensagem. Ele é utilizado nas seguintes mensagens, *Neighbor Solicitation*, *Router Solicitation* e *Router Advertisement*.

O campo transmitido deve conter as seguintes características:

- *Type* de 8 bits: contém o valor 1 que é o identificador do campo opcional *Source Link Layer Address*.
- *Length* de 8 bits: indica o tamanho do campo opcional *Source Link Layer Address*.
- *Link-Layer Address* de tamanho variável: contém o endereço físico (*MAC address*) do nó de origem.

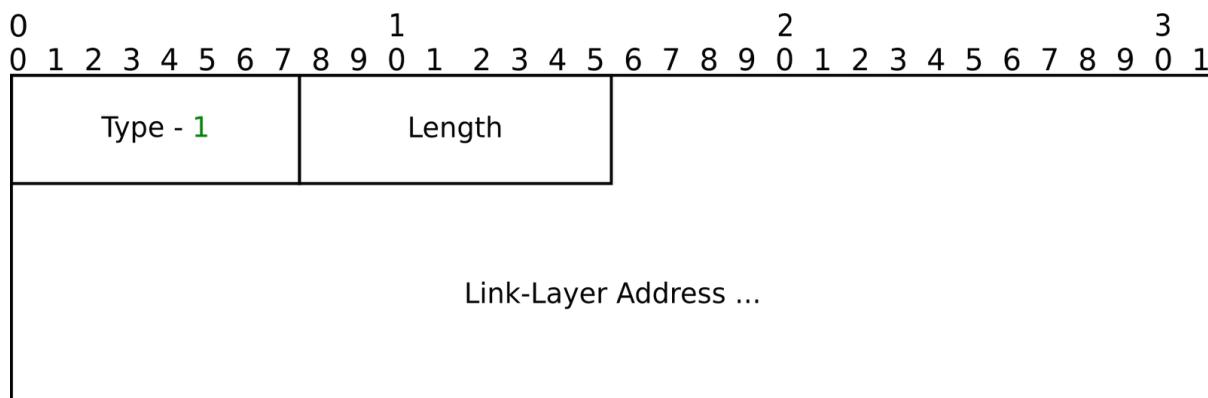


Figura 8: Formato do Pacote Source Link Layer Address

2.2.2. Target Link Layer Address

Sua função é carregar o endereço físico do nó de destino questionado por outra mensagem. Ele é utilizado nas seguintes mensagens, *Neighbor Advertisement* e *Redirect*.

O campo transmitido deve conter as seguintes características:

- *Type* de 8 bits: contém o valor 2 que é o identificador do campo opcional *Target Link Layer Address*.
- *Length* de 8 bits: indica o tamanho do campo opcional *Target Link Layer Address*.
- *Link-Layer Address* de tamanho variável: contém o endereço físico do nó de destino.

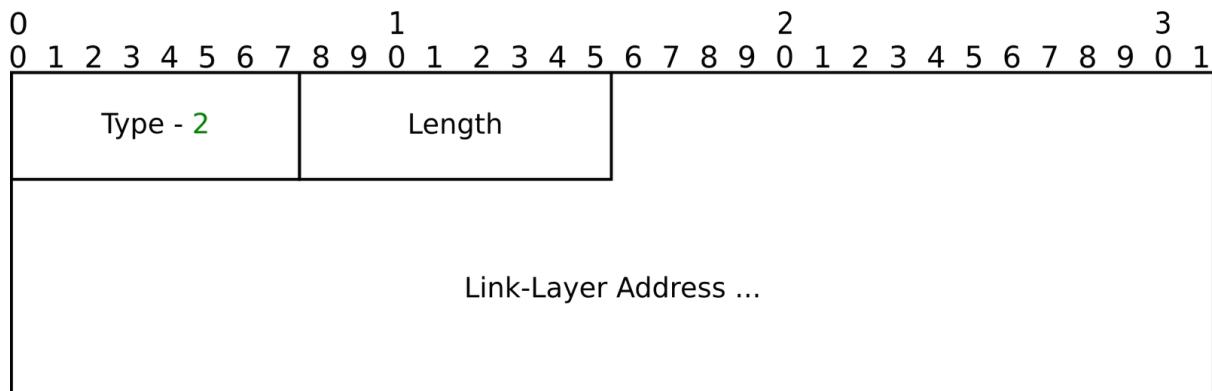


Figura 9: Formato do Pacote Target Link Layer Address

2.2.3. Prefix Information

Sua função é prover ao nó solicitante um prefixo de rede que pode ser utilizado tanto para que esse dispositivo se autoconfigure quanto para que ele identifique se um endereço de destino pertence ao enlace. Ele é utilizado nas mensagens Router Advertisement e deve ser ignorado em outras mensagens.

O campo transmitido deve conter as seguintes características:

- *Type* de 8 bits: contém o valor 3 que identificar o campo opcional *Prefix Information*.
- *Length* de 8 bits: contém o valor 4 que indica o tamanho dos campos fixos do campo opcional *Prefix Information*.
- *Prefix Length* de 8 bits: contém o tamanho do prefixo enviado.
- *On-Link Flag (L)* de 1 bit: indica se o prefixo enviado pode (1) ou não (0) ser utilizado para identificar endereços vizinhos do enlace. Caso o valor seja zero, o receptor não deve assumir que o prefixo não é do enlace e não pode utilizar essa informação para alterar sua tabela de prefixos do enlace (*Prefix List*).
- *Autonomous Address-Configuration Flag (A)* de 1 bit: indica se o prefixo pode ser utilizado para autoconfiguração stateless (1) ou não (0).
- *Reserved1* de 6 bits: não é utilizado e tem que ser inicializado com 0 na transmissão. O receptor tem que ignorar esse campo.
- *Valid Lifetime* de 32 bits: seu valor depende de duas situações,
 - Se a mensagem for utilizada para identificar um prefixo no enlace: esse campo marcará o tempo, em segundos, que o prefixo é considerado válido para se identificar os endereços lógicos pertencentes ao enlace. O valor (0xffffffff) indica infinito.
 - Se a mensagem for utilizada para a autoconfiguração de endereços Stateless: esse campo marcará o tempo, em segundos, que o endereço pode ser identificado como pertencente ao estado válido (olhar capítulo sobre estados dos endereços).
- *Preferred Lifetime* de 32 bits: marca o tempo, em segundos, que o endereço gerado via autoconfiguração stateless pode ser identificado como pertencente ao estado preferencial (olhar capítulo sobre estados dos endereços). O valor (0xffffffff) indica infinito.
- *Reserved2* de 32 bits: não é utilizado e tem que conter o valor zero na transmissão. O receptor tem que ignorar esse campo.

- *Prefix* de tamanho variável: contém o prefixo de rede a ser utilizado ou para a autoconfiguração ou para auxiliar na identificação de endereços do enlace.

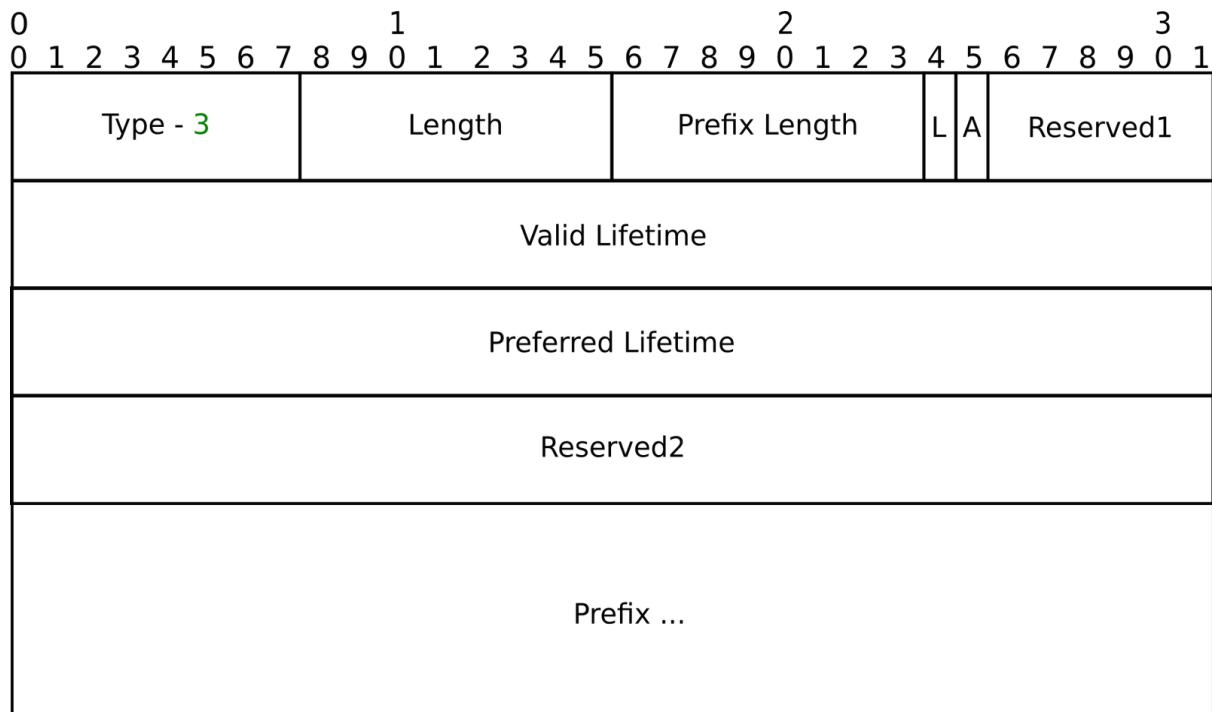


Figura 10: Formato do Pacote Prefix Information

2.2.4. Redirect Header

Sua função é enviar parte ou a totalidade da mensagem original que deverá ser redirecionada pelo nó de origem a outro nó. Ele é utilizado nas mensagens Redirect e deve ser ignorado em outras mensagens.

O campo transmitido deve conter as seguintes características:

- *Type* de 8 bits: contém o valor 4 que é o identificador do campo opcional *Redirect Header*.
- *Length* de 8 bits: indica o tamanho do campo opcional *Redirect Header*.
- *Reserved* de 48 bits: não é utilizado e tem que ser inicializado com 0 na transmissão. O receptor tem que ignorar esse campo.
- *IP Header + Data* de tamanho variável: contém parte ou a totalidade da mensagem original que deverá ser redirecionada. Esse campo não pode fazer com que o pacote Redirect exceda o tamanho mínimo de MTU para suportar o IPv6.

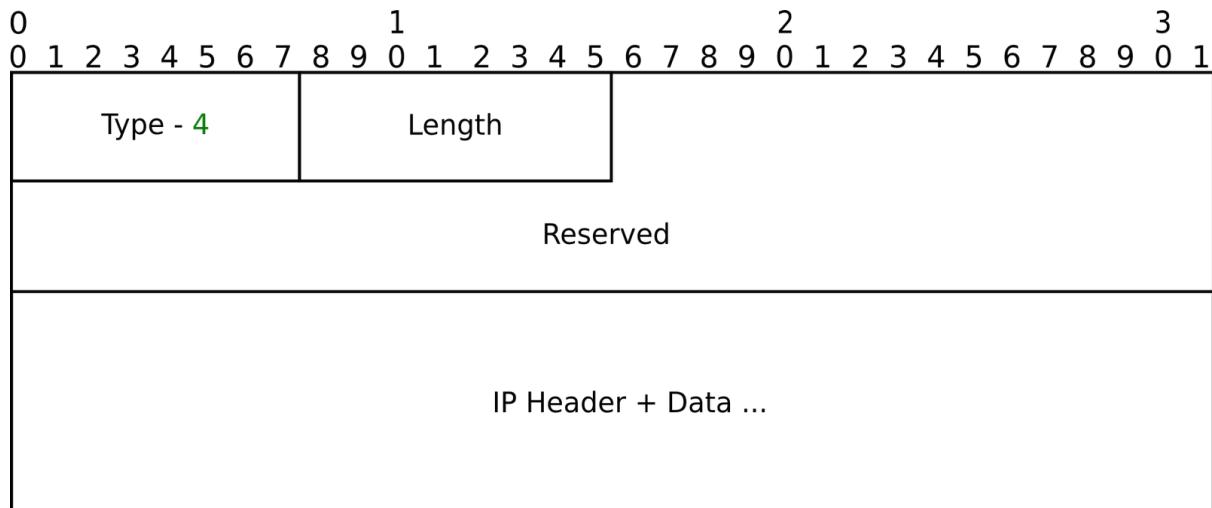


Figura 11: Formato do Pacote Redirect Header

2.2.5. MTU

Sua função é transmitir a todos os nós do enlace o mesmo tamanho de MTU válido. Ele é utilizado em casos que o MTU da camada de enlace não é bem conhecido.

Esse campo opcional é enviado nas mensagens *Router Advertisement* e deve ser ignorado em outras mensagens.

O campo transmitido deve conter as seguintes características:

- *Type* de 8 bits: contém o valor 5 que é o identificador do campo opcional *MTU*.
- *Length* de 8 bits: indica o tamanho do campo opcional *MTU*.
- *Reserved* de 16 bits: não é utilizado e tem que ser inicializado com 0 na transmissão. O receptor tem que ignorar esse campo.
- *MTU* de 32 bits: informa o tamanho máximo de *MTU* que deve ser utilizado em comunicações naquele enlace.

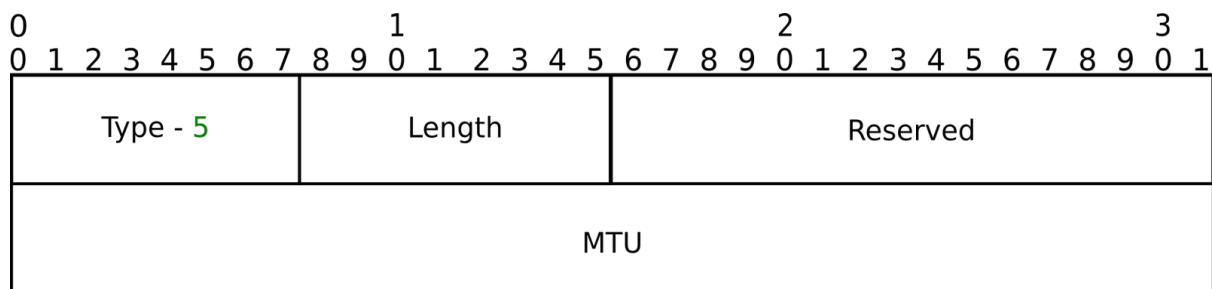


Figura 12: Formato do Pacote MTU

2.2.6. Recursive DNS Server Option (RDNSS)

Sua função é transmitir um ou mais endereços lógicos de servidores recursivos de DNS. Ele é utilizado em mensagens *Router Advertisement* e deve ser ignorado em outras mensagens.

O pacote transmitido deve conter as seguintes características:

- *Type* de 8 bits: contém o valor 25 que é o identificador do campo opcional *RDNSS*.

- *Length* de 8 bits: indica o tamanho do campo opcional *RDNSS*.
- *Reserved* de 16 bits: não é utilizado e tem que ser inicializado com 0 na transmissão. O receptor tem que ignorar esse campo.
- *Lifetime* de 32 bits: marca o tempo máximo, em segundos, que os endereços lógicos de RDNSS podem ser utilizados para resolução de endereços. O valor (0xffffffff) indica infinito enquanto que o valor 0 indica que o endereço não deve ser mais usado.
- *Address of IPv6 Recursive DNS Servers* de tamanho variável (mas múltiplo de 128 bits): contém os endereços lógicos IPv6 dos servidores que devem ser utilizado para a resolução de nomes.

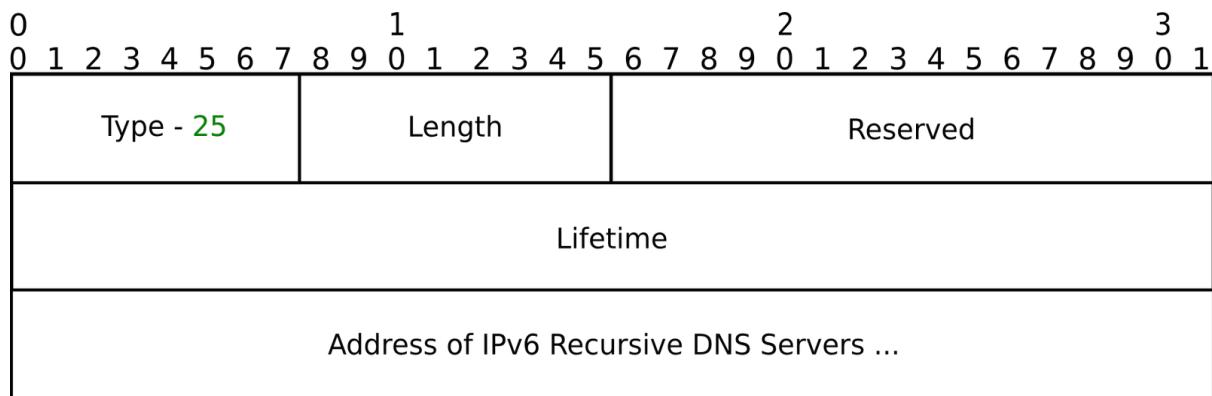


Figura 13: Formato do Pacote RDNSS

3. Funcionalidades Básicas baseadas no NDP

Neste tópico serão detalhadas e explicadas as funcionalidades básicas anteriormente citadas no tópico sobre o protocolo descoberta de vizinhança. Além disso, em cada funcionalidade será criada uma topologia exemplo para demonstrar a troca de mensagens e assim apresentar o resultado do procedimento.

3.1. Duplicate Address Detection (DAD)

A detecção de endereços duplicados é um procedimento realizado pelos nós para verificar a unicidade de um endereço *unicast* antes de atribuí-lo a uma de suas interfaces de rede. Independentemente da maneira como foi obtido o endereço, seja por métodos manuais, autoconfiguração stateless ou autoconfiguração stateful, endereços duplicados não podem ser aceitos em comunicações, uma vez que atrapalham o roteamento de pacotes.

O mecanismo DAD começa com a criação de uma mensagem *Neighbor Solicitation* pelo dispositivo que está tentando adicionar um endereço a sua interface. Essa mensagem é então enviada ao enlace a procura da existência de algum nó que esteja utilizando o mesmo endereço. Essa mensagem deve conter o campo *source* do IPv6 vazio (::), o campo *target* do ICMPv6 com o endereço a ser adicionado e, o campo *destination* do IPv6, com o endereço *Multicast Solicited Node*(FF02::1:FFXX:XXXX).

O processo de autoconfiguração é interrompido imediatamente, caso algum nó responda uma mensagem *Neighbor Advertisement* que contenha:

- o campo *source* do cabeçalho IPv6 com o mesmo endereço enviado no campo *target* da mensagem *Neighbor Solicitation*; e,
- o campo *destination* do cabeçalho IPv6 com o endereço Multicast All-nodes;

Para solucionar o conflito é preciso que um endereço distinto seja adicionado manualmente em um dos dispositivos ou que seja configurado algum procedimento para a adoção de um novo endereço, como por exemplo a escolha de um outro algoritmo que para criar identificador de interface.

Caso nenhuma resposta com as características descritas seja recebida em um tempo pré-determinado, o dispositivo poderá então finalizar sua configuração de interface e, assim, iniciar comunicação com qualquer outros nó. Normalmente, o tempo de espera pela mensagem *Neighbor Advertisement* é de 1 segundo, contudo esse valor pode ser alterado para cada interface de rede.

No IPv4 existe um processo semelhante, todavia a troca de mensagens é feita com outro protocolo, o ARP. Nele, essa funcionalidade é realizada com a mensagem *ARP Request* e o com o método *gratuitous ARP*.

A seguir está apresentado um exemplo de uma topologia na qual uma troca de mensagem foi simulada para representar tal mecanismos.

3.1.1. Exemplo

A topologia desse exemplo é constituída de três computadores interconectados por um switch, como indicado na Figura 14. O computador ‘Cópia’ irá trocar seu endereço lógico pelo o mesmo endereço do computador ‘Original’, porém não podem haver endereços repetidos no enlace.

Cada computador possui uma diferente descrição:

- Original: é o verdadeiro dono do endereço IPv6 *global* (2001:db8::10) e do *link local* (FE80::200:FF:FEAA:0). A partir do endereço *global*, a máquina pode se comunicar tanto na internet como na rede local;
- Cópia: é o dono do endereço IPv6 *link local* (FE80::200:FF:FEAA:2) e não possui endereço global. Contudo por algum motivo não especificado, ela tenta adicionar o mesmo endereço IPv6 global da máquina ‘Original’ (2001:db8::10) a sua interface;
- Cliente: é o dono do endereço IPv6 *global* (2001:db8::11) e do *link local* (FE80::200:FF:FEAA:1). Ele participa do exemplo como observador de quem é o verdadeiro dono do endereço IPv6 (2001:db8::10)

Rede Local

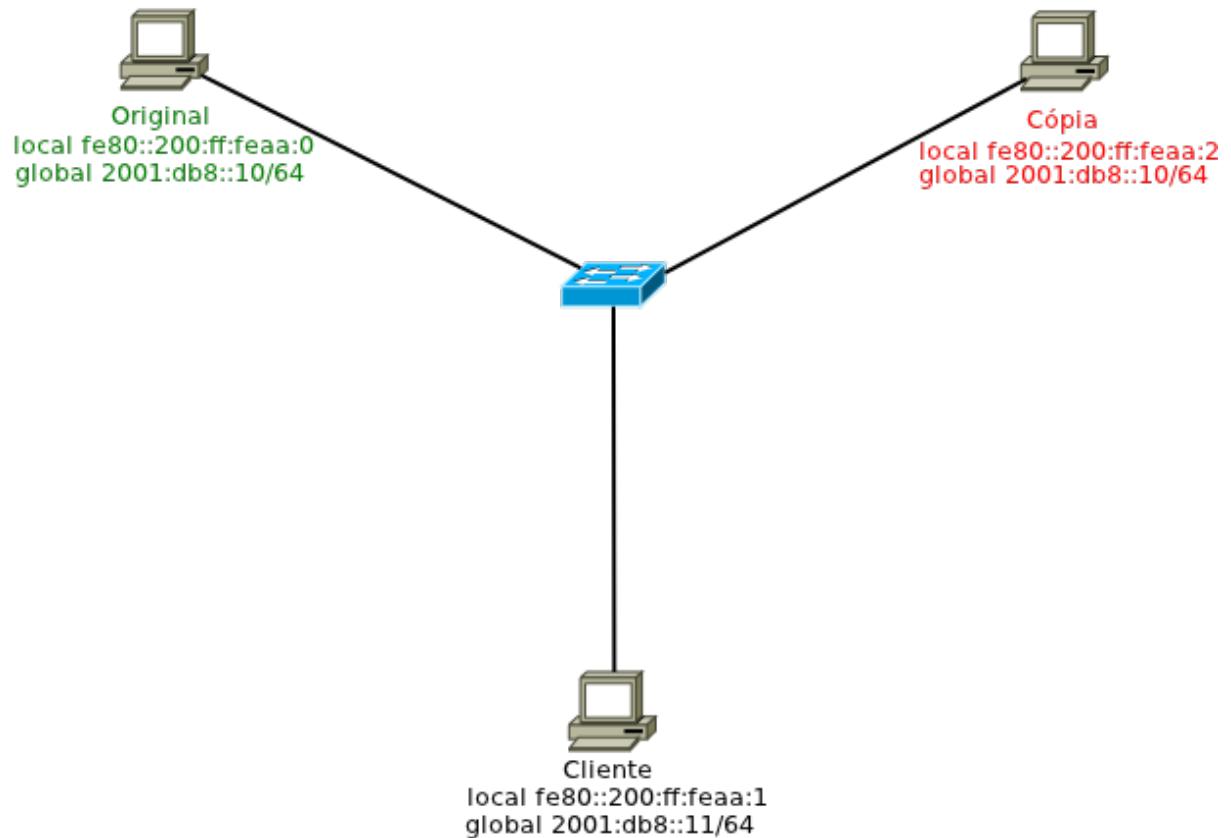


Figura 14: Topologia do exemplo da funcionalidade Detecção de Endereços Duplicados

No momento que a máquina Cópia tentar adicionar o endereço repetido a sua interface, a seguinte troca de mensagens acontece:

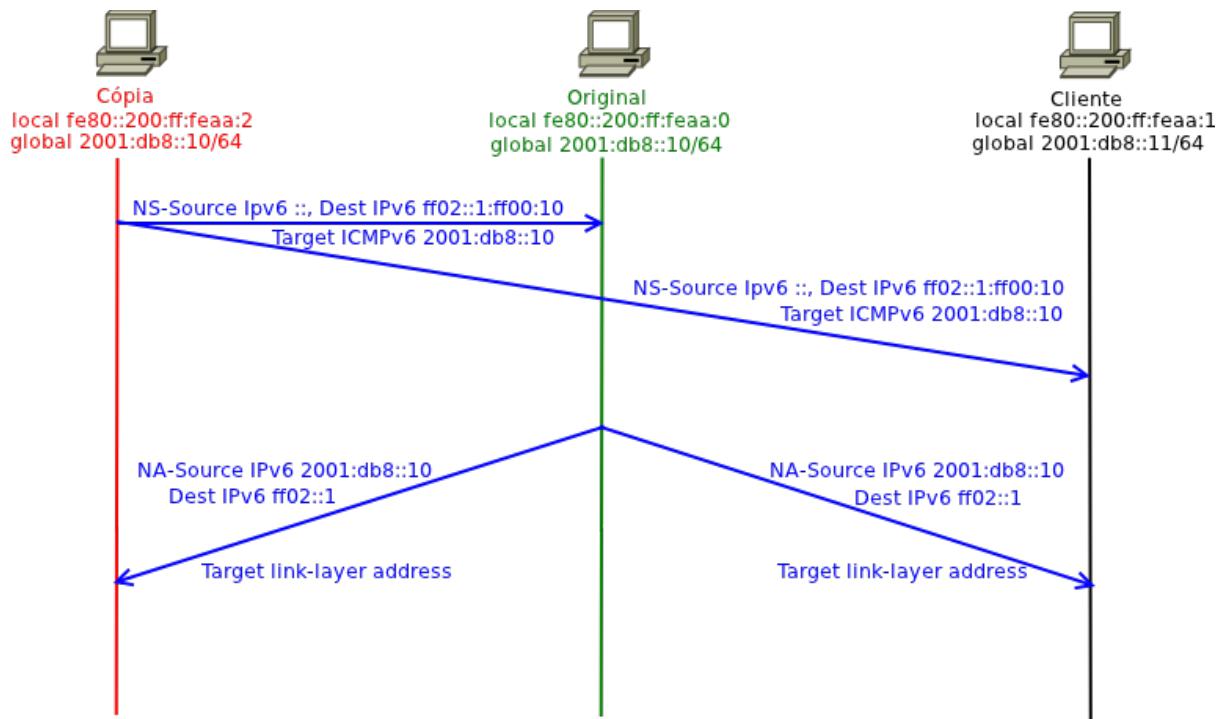


Figura 15: Troca de mensagens do exemplo da funcionalidade Detecção de Endereços Duplicados

No final do processo, a máquina ‘Cópia’ não consegue se autoconfigurar e a máquina ‘Cliente’ só reconhece a máquina ‘Original’ como o dono do endereço IPv6 (2001:db8::10).

3.2. Address Resolution

A resolução de endereços é um procedimento realizado para que um nó descubra endereços físicos de outros dispositivos quando ele só conhece seus endereços lógicos IPv6. Tal procedimento é aplicado em enlaces em que dois dispositivos desconhecem os endereços físicos uns dos outros, porém, desejam se comunicar.

O mecanismo inicia com o envio uma mensagem Neighbor Solicitation para todos os nós do enlace a procura do dono de determinado IPv6. O pacote deve ser preenchido com o endereço de destino *Multicast Solicited-Node* ($FF02::1:FFXX:XXXX$) no cabeçalho IPv6 e com o endereço a ser resolvido, campo *Target* do cabeçalho ICMPv6. O campo opcional *Source link-layer Address* é enviado com intuito de fornecer o endereço físico da origem ao destino e, assim, diminuir a trocas de mensagens entre os dispositivos ante do início de uma comunicação.

O nó destino, dono do endereço IPv6 enviado no campo *Target*, ao receber a mensagem, responde com uma mensagem *Neighbor Advertisement* enviada diretamente ao nó requisitante. O pacote enviado é preenchido com o endereço de destino IPv6 *unicast (local ou global)* do nó que enviou a mensagem *Neighbor Solicitation* e com seu próprio endereço físico no campo opcional *Target link-layer address*. A partir do recebimento dessas informações, o nó requisitante pode atualizar seu *Neighbor cache*, com o endereço físico e o status de alcance do outro nó para iniciar a transmissão de dados a seguir.

No IPv4 existe um processo semelhante, todavia a troca de mensagens é feita com outro protocolo, o ARP.

A seguir está apresentado uma topologia sobre a qual foi realizada uma troca de mensagem que representa o funcionamento desse mecanismo.

3.2.1. Exemplo

A topologia desse exemplo é constituída de dois computadores interconectados entre si. Nela, o ‘Cliente1’ inicia uma comunicação com o ‘Cliente2’ sem conhecer inicialmente seu endereço físico.

Tais computadores possuem as seguintes descrições:

- Cliente1: é o dono do endereço IPv6 (2001:db8::10) e deseja se comunicar com o ‘Cliente2’. Contudo, esse nó conhece apenas o endereço IPv6 do Cliente2 (2001:db8::11) e, por isso precisa realizar o procedimento *Address Resolution* para descobrir o endereço deste segundo computador;
- Cliente2: é o dono do endereço IPv6 (2001:db8::11) e deve receber um pedido de comunicação. Não é relevante o fato dele possuir ou não o endereço físico do ‘Cliente1’, já que esse dado será enviado através da mensagem *Neighbor Solicitation* no campo opcional *Source Link-Layer Address*.

Rede Local



Figura 16: Topologia do exemplo da funcionalidade Resolução de Endereços

No momento que a máquina ‘Cliente1’ tentar se comunicar com o máquina ‘Cliente2’, a seguinte troca de mensagens deverá acontecer:

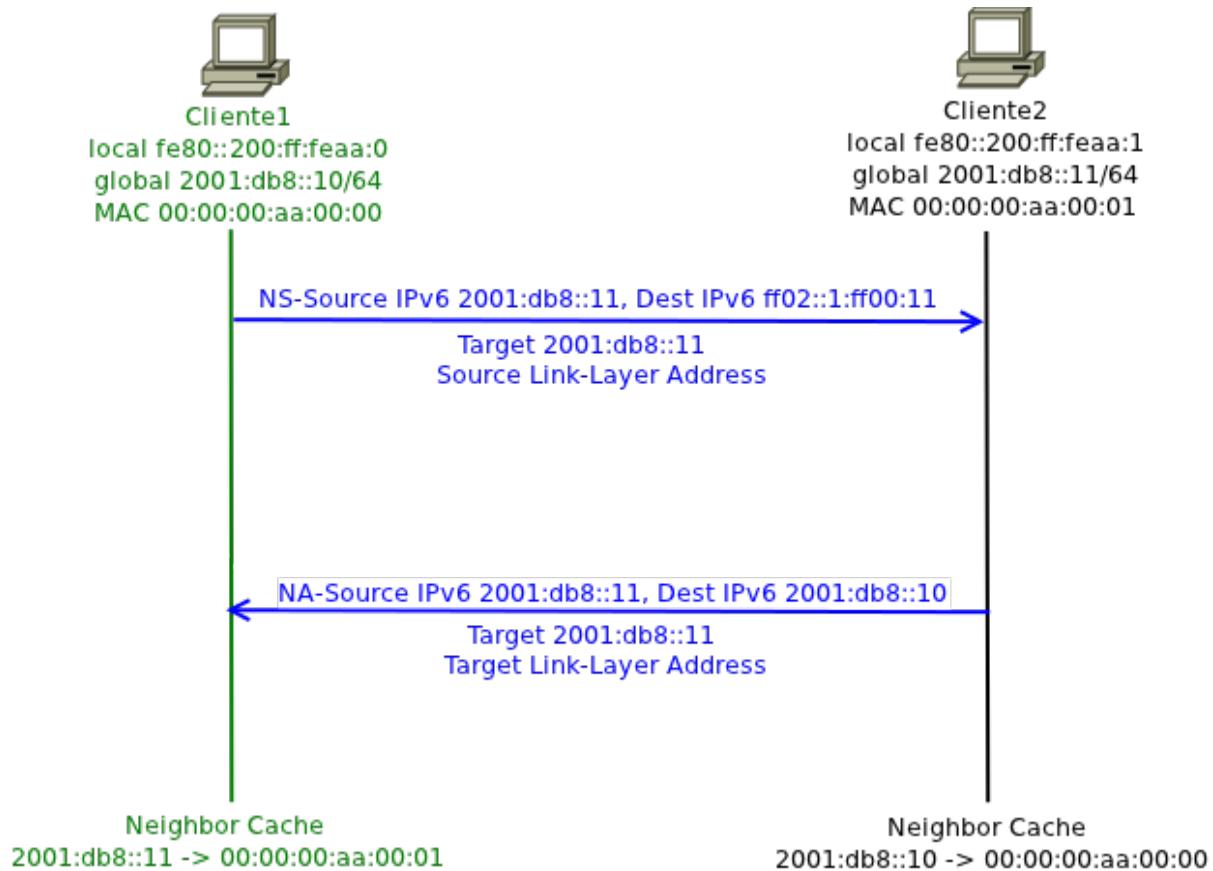


Figura 17: Troca de mensagens do exemplo da funcionalidade Resolução de Endereços

No final, ambas as máquinas sabem os endereços físicos uma da outra (os campos Source Link-layer Address e Target Link-Layer Address transportam os endereços MAC) e os têm armazenados em seus respectivos caches de vizinhanças (*Neighbor Cache*). A partir de então, transmissões de dados podem ser feitas entre esses dois dispositivos.

3.3. Router Discovery

A descoberta de roteadores é um procedimento realizado para que dispositivos encontrem os roteadores do enlace ao qual estão ligados. Existem dois momentos em que esse procedimento pode ser executado. O primeiro é quando um nó se conecta ou reconecta numa rede, ou seja, no momento em que ele configura sua interface. E, o segundo, é quando ocorre um evento periódico, protagonizado por roteadores, para o anunciar presença no enlace a todos os nós participantes.

Quando o mecanismo é originado por um nó da rede, ele é iniciado com o envio da mensagem *Router Solicitation* a todos os roteadores do enlace. Em termos técnicos, o pacote é montado com o valor *All-Router* (FF02::2) multicast Group no endereço de destino IPv6 e com o endereço de origem IPv6 *unicast (local ou global)* da interface do nó solicitante. Além de requisitar aos roteadores que se apresentem imediatamente, esse pacote pode também possuir a função de informar o endereço físico do dispositivo de origem para evitar o procedimento de resolução de endereços.

O roteador, ao receber a mensagem *Router Solicitation*, responde com uma mensagem *Router Advertisement*, enviada diretamente ao nó solicitante via destino IPv6 *unicast (local ou global)*. Porém, em alguns casos, é possível que essa resposta seja enviada para todos os nós do enlace através do endereço *Multicast All Node*. Essa mensagem notifica uma rota para ser adicionada ou atualizada nos caches (de vizinhança e de roteamento) dos dispositivos receptores que podem, a partir de então, transmitir dados através desse roteador.

Caso o roteador decida se anunciar periódicamente no enlace o procedimento é feito com o envio não solicitado da mensagem *Router Advertisement* com o endereço de destino IPv6 *Multicast All-Nodes* (*FF02::1*). Ao chegar num nó, essa mensagem recebe o mesmo tratamento anterior de adição ou de atualização dos caches apresentado para as mensagens RA geradas como resposta de mensagens RS.

A seguir estão apresentados dois exemplos com topologias diferentes, sobre quais serão montadas trocas de mensagens que representem o mecanismo desta funcionalidade.

3.3.1. Exemplo 1

O primeiro exemplo mostra uma requisição de um cliente para que um roteador se apresente. A topologia construída constitui-se apenas de um computador e um roteador conectados.

A descrição de cada um dos dispositivos é:

- Cliente: é a máquina dona dos endereços IPv6 *global* (2001:db8::10) e *link local* (FE80::200:FF:FEAA:0). Ela ao se conectar a rede irá procurar aprender rotas para se comunicar na internet. Para isso, ela precisa realizar procedimento *Router Discovery* para descobrir os roteadores presentes no enlace;
- Roteador: é a máquina que irá se apresentar como roteador do enlace através do endereço IPv6 *global* (2001:db8::11) e *link local* (FE80::200:FF:FEAA:1).

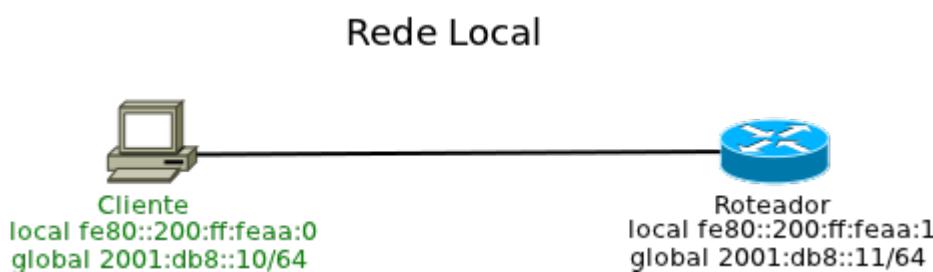


Figura 18: Topologia do exemplo 1 da funcionalidade Descoberta de Roteadores

No momento em que a máquina ‘Cliente’ tentar se conectar a rede, subindo sua interface, a seguinte troca de mensagens deverá acontecer:

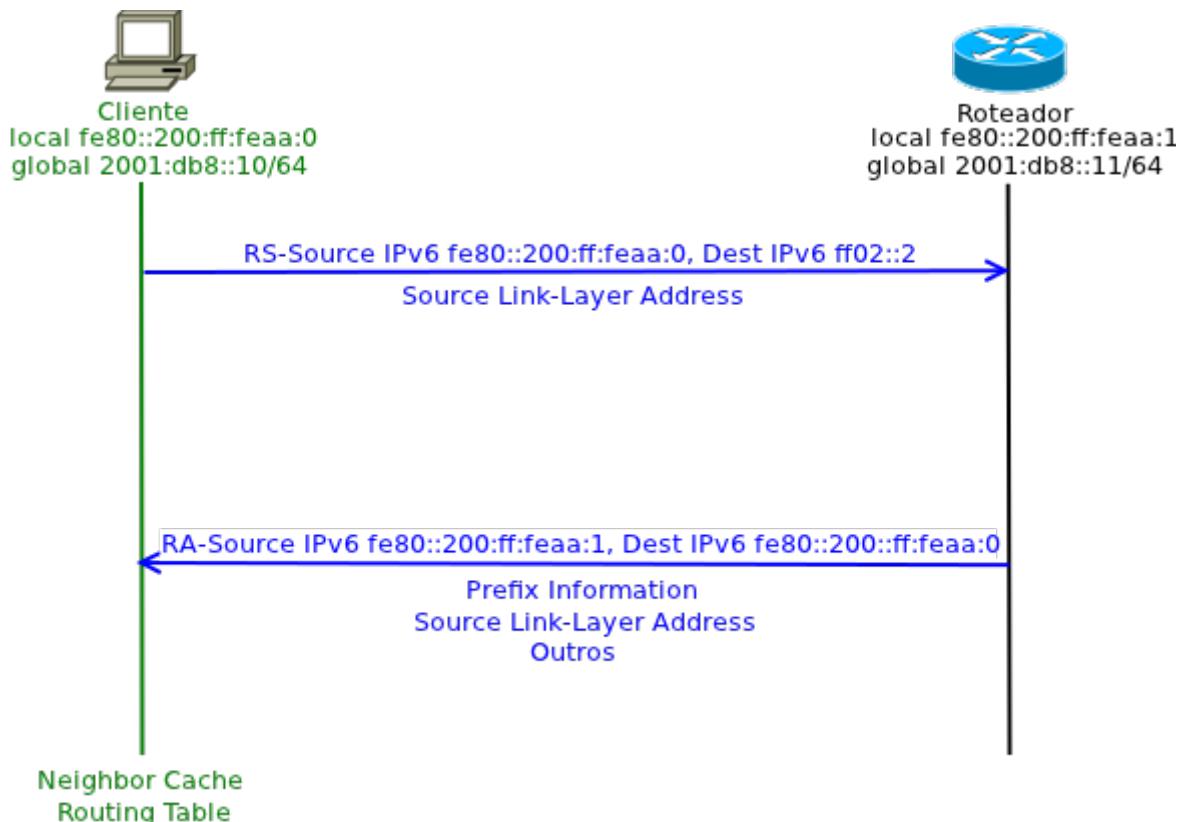


Figura 19: Troca de mensagens do exemplo1 da funcionalidade Descoberta de Roteadores

No final do procedimento, a máquina ‘Cliente’ deverá ter armazenado em seus caches, de vizinhança e de roteamento, as informações sobre o roteador. A partir desse momento, qualquer transmissão de dados poderá ser feita através do ‘Roteador’.

3.3.2. Exemplo 2

O segundo exemplo mostra o anúncio, periódico, da existência de um roteador no enlace. A topologia constitui-se de dois computadores e um roteador interconectados através de um switch.

Esta é a descrição desses componentes:

- Cliente: é a máquina dona dos endereços IPv6 *global* (2001:db8::10) e *link local* (FE80::200:FF:FEAA:0), que irá receber periodicamente o anúncio do roteador;
- Cliente2: é uma segunda máquina, dona os endereços IPv6 *global* (2001:db8::12) e *link local* (FE80::200:FF:FEAA:2), que irá receber periodicamente o anúncio do roteador;
- Roteador: é a máquina dona dos endereços IPv6 *global* (2001:db8::11) e *link local* (FE80::200:FF:FEAA:1), que irá se apresentar no enlace como roteador, sem ser solicitada, através da mensagem *Router Advertisement*.

Rede Local

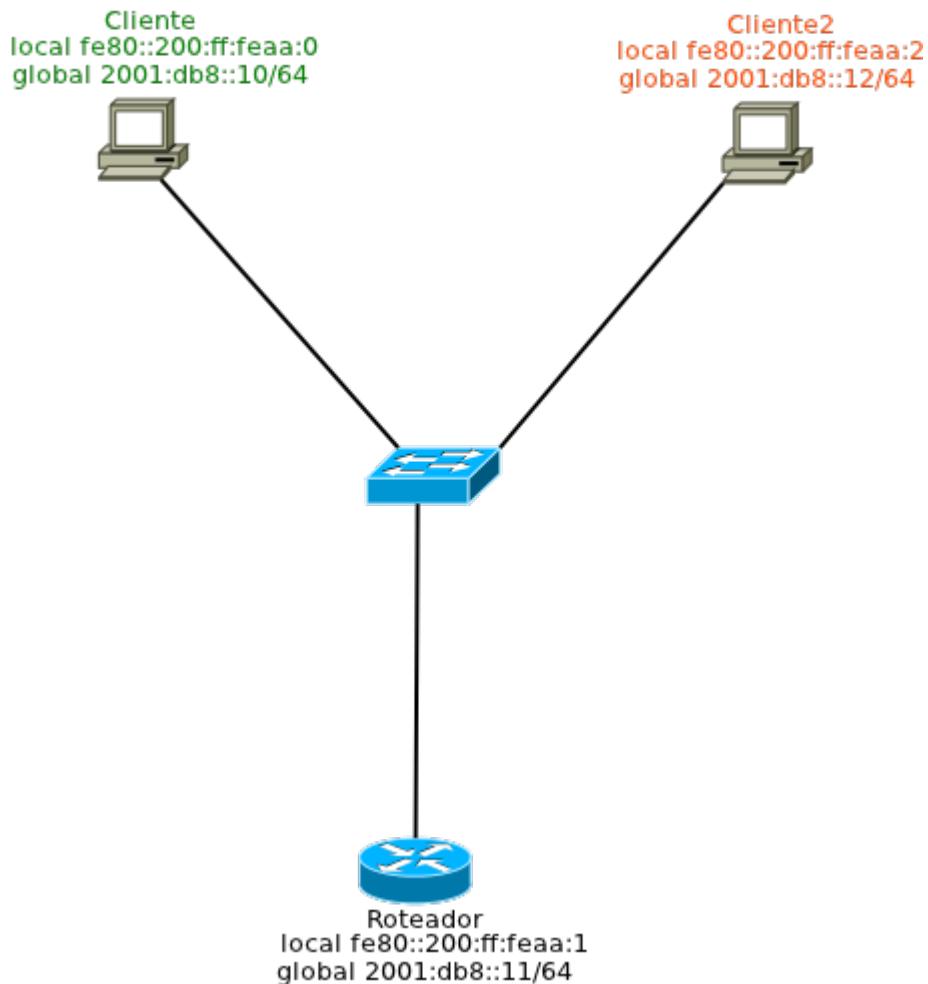


Figura 20: Topologia do exemplo2 da funcionalidade Descoberta de Roteadores

De tempos em tempos o roteador se anuncia no enlace e o seguinte envio de mensagens acontece:

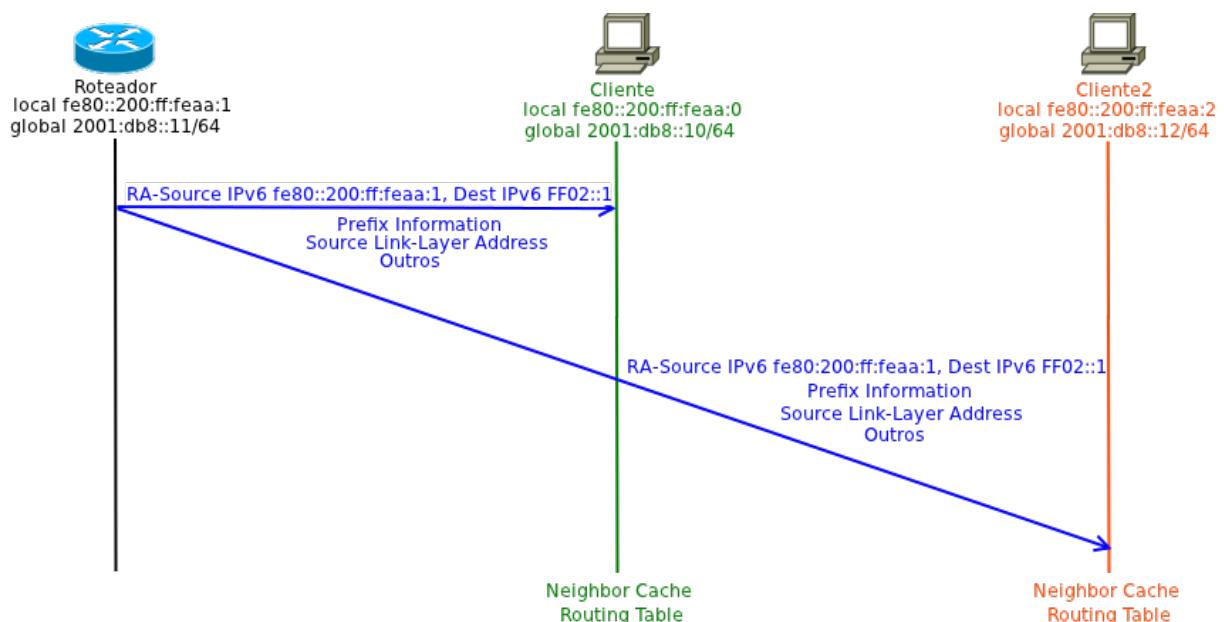


Figura 21: Troca de mensagens do exemplo2 da funcionalidade Descoberta de Roteadores

No final do procedimento, ambas as máquinas clientes terão armazenado em seus caches, tanto no de vizinhança quanto no de roteamento, as informações sobre o roteador. A partir desse momento, qualquer transmissão de dados poderá ser feita através do roteador.

3.4. Prefix Discovery

A descoberta de prefixos é um procedimento realizado para que nós encontrem os prefixos de rede que pertencentes a seus enlace. Esse procedimento está atrelado com a funcionalidade Descoberta de Roteadores, uma vez que, utiliza a mesma troca de mensagens, com a adição de algumas informações no campo opcional. O opção *Prefix Information* possui essa função de informar prefixos através da flag *On-Link*, que deverá estar ativada. Sempre que ela estiver presente em mensagens *Router Advertisements* o mecanismo está operando.

Os exemplos do tópico *Router Discovery* mostram o envio da opção *Prefix Information* e podem ser utilizados também para representar o mecanismo do *Prefix Discovery*.

3.5. Parameter Discovery

A descoberta de parametros é um procedimento realizado para que nós encontrem informações adicionais, tanto do enlace quanto da Internet, para sua autoconfiguração. Esse procedimento também está atrelado à funcionalidade Descoberta de Roteadores (*Router Discovery*), uma vez que utiliza a mesma troca de mensagens, porém, com a adição informações no campo opcional. Existem vários tipos dados que podem ser transmitidos nesses campos, como por exemplo MTU, limite de encaminhamento, RDNSS e outros.

Os exemplos do tópico *Router Discovery* mostram o envio de campos opcionais, através da denominação “Outros”, e podem ser utilizados também para representar o mecanismo de *Parameter Discovery*.

3.6. Neighbor Unreachability Detection

A detecção de vizinhos inacessíveis é um procedimento utilizado pelos nós para evitar rotas estejam apresentando falhas. Contudo, esse procedimento só funciona se o problema estiver no meio de uma rota e não em seu destino final. Nesse último caso, nada poderá ser feito para se manter uma comunicação e, a transmissão não poderá ser executada até que alguma correção seja aplicada ao dispositivo problemático.

Existem várias razões para que um nó deixe de ser acessível como falhas no hardware, falta de energia ou problemas de conexão.

O mecanismo inicia-se quando, durante uma comunicação entre dois dispositivos, a comunicação com o primeiro dispositivo da rota apresenta sinais de falha. Nessa situação, caso protocolos de camadas de redes superiores consigam fazer a detecção do problema, este mecanismo não é obrigatório. Porém nem todos os equipamentos conseguem interpretar essas mensagens, como é o que acontece nos roteadores que só trabalham até a camada 3.

Para identificar se um vizinho continua acessível são enviados três (constante determinada pela variável MAX_UNICAST_SOLICIT da RFC 4861) mensagens *Neighbor Solicitation* com endereço de destino igual ao endereço *unicast* (*local* ou *global*) do dispositivo que está apresentando falha. Após o envio, o dispositivo original entra em modo de aguardo até que uma das duas situações ocorram: ele receba uma resposta mensagem *Neighbor Advertisement* ou o tempo de espera acabe.

Na primeira circunstância, o cache de vizinhança (*neighbor cache*) é atualizado e o direcionamento de pacotes para aquele nó continua a ser realizado normalmente.

Já na outra, o cache de vizinhança (*neighbor cache*) também é alterado, todavia retirando-se a informação sobre este destino. Caso o destino final seja um nó vizinho, o procedimento de resolução de endereços de camada dois é iniciado com o intuito de descobrir novamente o respectivo endereço físico. Caso contrário, os pacotes serão redirecionados a algum outro roteador escolhido randomicamente da lista de roteadores do enlace conhecidos. Em outras palavras, o roteador selecionado pode não ser a melhor rota no enlace até aquele determinado destino. Portanto se o procedimento de redirecionamento de comunicação estiver habilitado na rede, ele poderá acontecer caso exista uma rota melhor no enlace.

A seguir está apresentado um exemplo de uma topologia sobre a qual será mostrada uma troca de mensagem que represente o mecanismo desta funcionalidade.

3.6.1. Exemplo

A topologia deste exemplo é constituída de dois computadores interconectados por um roteador. Existe uma rota alternativa, porém ela possui mais saltos.

Cada dispositivo possui uma diferente descrição:

- Cliente1: é possui o endereço IPv6 *global* (2001:db8:cafe::10) e realiza uma comunicação unidirecional ativa com o ‘Cliente2’ através do ‘Roteador’.
- Cliente2: é o dono do endereço IPv6 *global* (2001:db8:dado::10) e está recebendo dados do ‘Cliente1’ através do roteador.
- Roteador: possui duas interfaces configuradas com os seguintes endereços IPv6 globais: (2001:db8:cafe::11) do lado do ‘Cliente1’ e (2001:db8:dado::11) do lado ‘Cliente2’. A comunicação entre os dois clientes é feita através deste roteador.
- RoteadorAlternativo1 e RoteadorAlternativo2: representam uma segunda rota para comunicar o ‘Cliente1’ ao ‘Cliente2’.

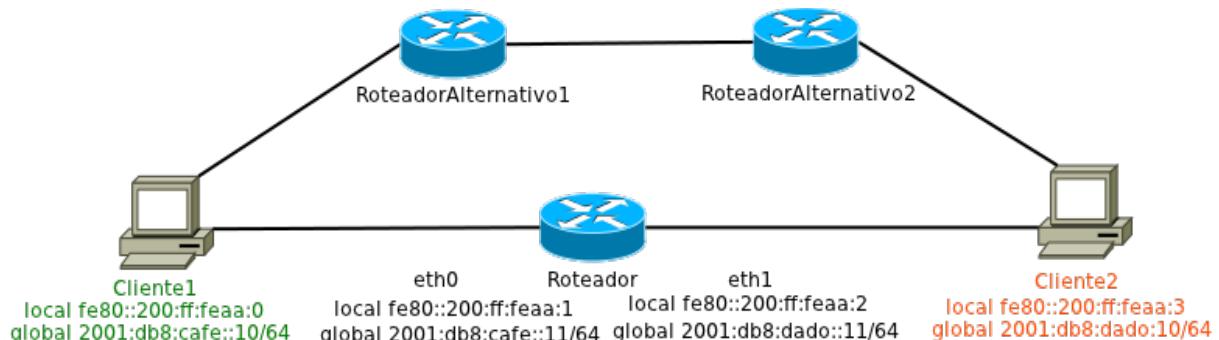


Figura 22: Topologia do exemplo da funcionalidade Detecção de vizinhos inacessíveis

Nessa situação, durante a comunicação entre os clientes, acontece um problema no ‘Roteador’. Passado certo tempo, a máquina ‘Cliente1’ começa a desconfiar que o ‘Roteador’ pode não estar operando devido a falta de mensagens dele provenientes. Isso acarreta na realização do mecanismo de detecção de vizinhos inacessíveis.

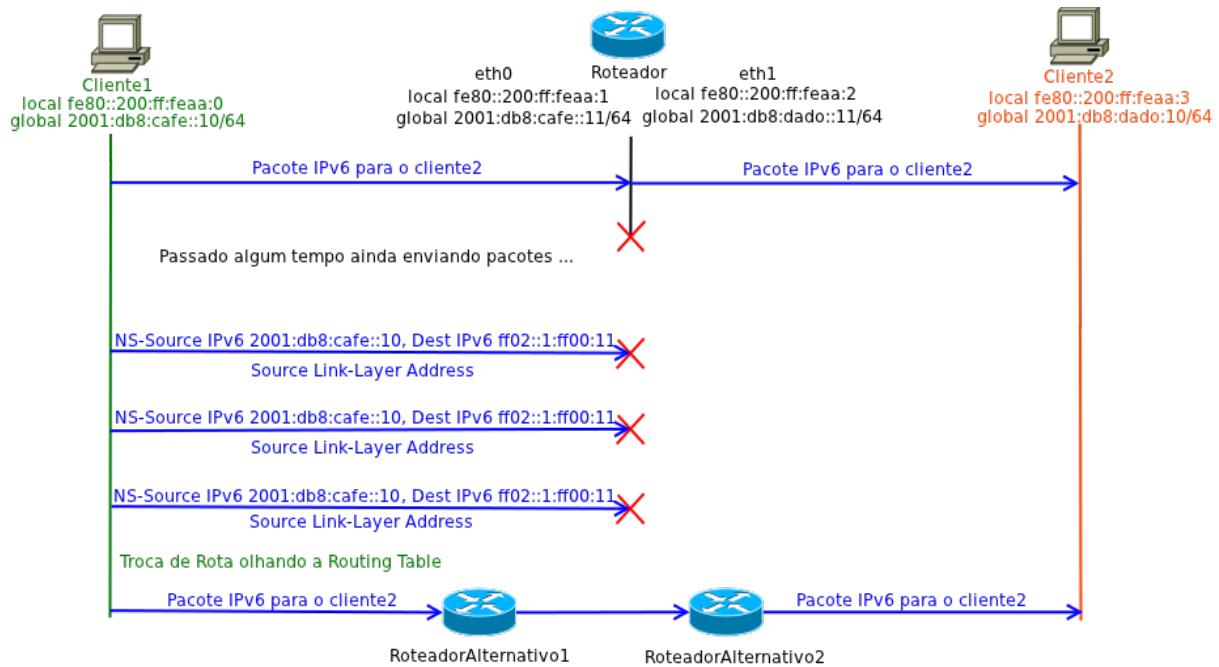


Figura 23: Troca de mensagens do exemplo da funcionalidade Detecção de vizinhos inacessíveis

No final do processo, como o roteador não responde as mensagens de neighbor solicitation, o ‘Cliente1’ por conhecer que o prefixo não pertence ao enlace, opta por procurar uma nova rota em sua tabela de roteamento continua sua comunicação com o cliente2.

3.7. Redirect

O redirecionamento é um procedimento utilizado por roteadores para informar a algum dispositivo a existência de uma rota melhor que pode ser utilizada em uma determinada comunicação. Essa nova rota pode ser a indicação tanto de um outro roteador de saída dos pacotes, quanto do destino final caso esse se encontre no mesmo enlace do dispositivo de origem.

O mecanismo é idêntico ao realizado no IPv4 ele é iniciado a partir da análise de uma comunicação que atravessa um determinado roteador. Quando um dispositivo envia pacotes através de um roteador, ele identifica que a opção de rota não foi a mais adequada e, numa tentativa de corrigir essa situação, envia uma mensagem do tipo *Redirect* ao dispositivo que remeteu a mensagem. Essa mensagem carrega informações sobre destino inicial da nova rota, que pode ser outro roteador ou o próprio destino final quando esse se encontra no enlace. Ao recebê-la, para dar continuidade à comunicação, o dispositivo remetente atualiza seu cache de destino (destination cache) e, se necessário, o cache de vizinhança (*neighbor cache*).

A mensagem *Redirect* também deverá conter o endereço físico (*MAC Address*) do novo destino. Assim, evita que o procedimento de resolução de endereços de camada dois (*Address Resolution*) aconteça, diminuindo o tráfego de mensagens no enlace.

Além disso, esse procedimento possui suporte a autentificação como forma de impedir que dispositivos não autorizados causem enviem *Redirects* inapropriados. Nessa situação os dispositivos então podem descartar mensagens não autenticadas.

A seguir está apresentado um exemplo de uma topologia sobre a qual será montada uma troca de mensagem que represente o mecanismo desta funcionalidade.

3.7.1. Exemplo

A topologia construída para este exemplo é constituída de um computador interconectado a dois roteadores como saídas para Internet. Nessa situação, um destino na Internet é acessado pelo ‘Cliente’ através do ‘Roteador1’. Contudo a rota para aquele destino que passa pelo Roteador2 é melhor.

Esses dispositivos possuem as seguintes descrições:

- Cliente: possui o endereço IPv6 global (2001:db8::10) e se comunica ativamente com servidores na Internet. Essa comunicação acontece através do ‘Roteador1’, a pesar da existência de uma rota melhor pelo Roteador2.
- Roteador1: possui o endereço IPv6 global (2001:db8::11) e pertence à rota da comunicação entre o ‘Cliente’ e um servidor na Internet.
- Roteador2: possui o endereço IPv6 global (2001:db8::12) e faz parte de uma rota melhor para a comunicação entre o ‘Cliente’ e o destino.

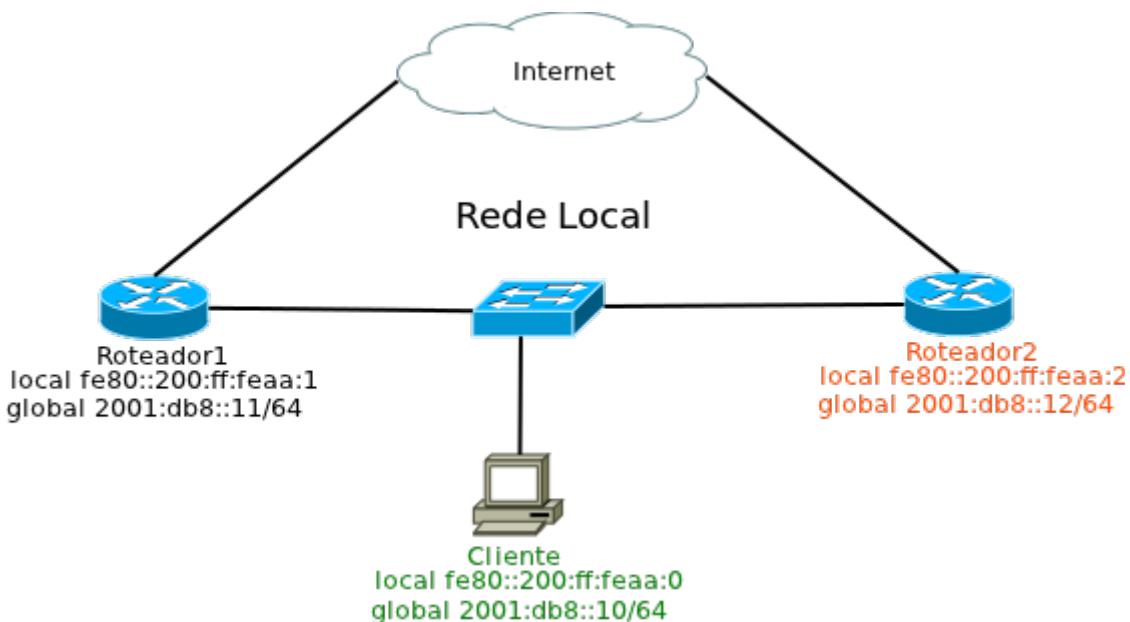


Figura 24: Topologia do exemplo da funcionalidade Redirecionamento

Nesse exemplo, durante uma comunicação entre o ‘Cliente’ e um destino na Internet, o ‘Roteador1’ irá sofrer um redirecionamento da rota que está sendo utilizada.

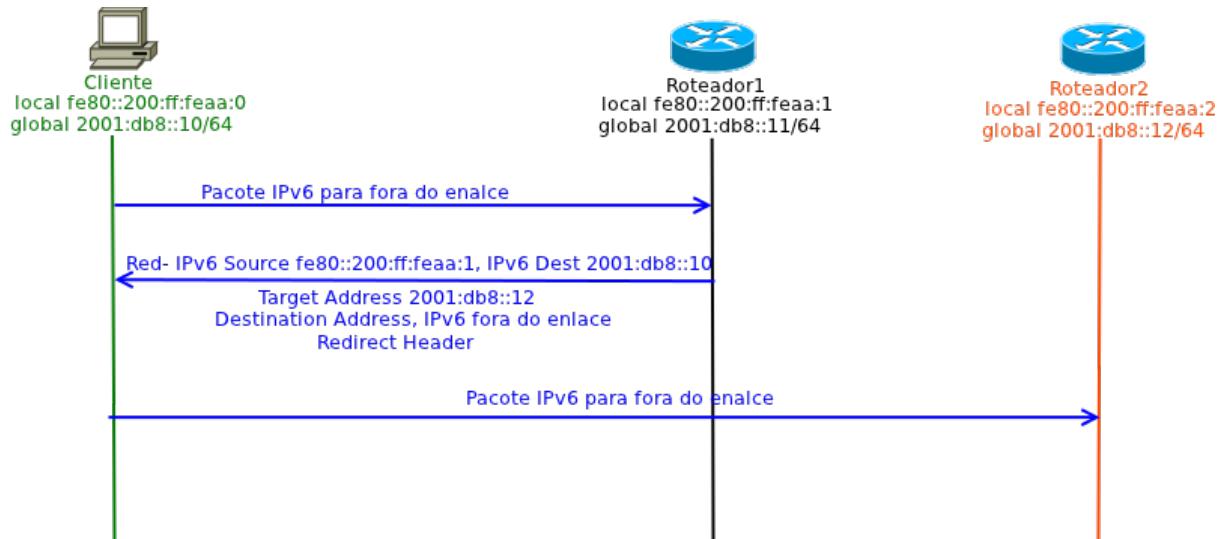


Figura 25: Troca de mensagens do exemplo da funcionalidade Redirecionamento

Isso acontece por que na tabela de rotas (*routing table*) do ‘Roteador1’ existe a informação de que o ‘Roteador2’ consegue enviar pacotes mais eficientemente para o destino daquela comunicação. Então, ele envia uma mensagem ao ‘Cliente’ para informar que o ‘Roteador2’ deve ser utilizado. O ‘Cliente’ ao receber a mensagem para de enviar pacotes para o ‘Roteador1’ e começa a mandá-los para o ‘Roteador2’, demonstrando a funcionalidade de Redirecionamento (*Redirect*).

4. Funcionalidades Básicas com foco no mecanismo de autoconfiguração

Neste tópico serão explicadas e detalhadas as funcionalidades básicas utilizadas para a autoconfiguração de dispositivos. Diferente do que acontecia tópico anterior, no qual todas as mensagens se baseavam no protocolo de descoberta de vizinhança (*neighbor discovery protocol*), neste, os mecanismos de autoconfiguração podem utilizar também o DHCP (Dynamic Host Configuration Protocol). O que define a escolha entre os protocolos é o modo de operação que se deseja utilizar, stateless ou stateful, e o objetivo de sua utilização.

4.1. Autoconfiguração

No IPv6, o processo de autoconfiguração é uma das principais características de seu funcionamento básico. É a partir dele que os diversos dispositivos podem adquirir informações da rede (e.g. servidores NTP), do enlace (e.g. prefixos pertencentes ao enlace) e de endereçamento. Isso faz com que a Internet ganhe dinamismo, visto que permite dispositivos se interconectarem sem a necessidade de configurações manuais.

Existem dois modos de divulgação de informações para a autoconfiguração dos dispositivos: o *stateless* e o *stateful*. No primeiro, o dispositivo que fornece informações de configuração não mantém o registro do estado e das características do nó destinatário. Ou seja, o nó destino se encarrega de se autonavar quanto o nó origem apenas informa as características da rede. As mensagens *Router Advertisement* são um exemplo desse modo de configuração.

Já no modo *stateful*, o dispositivo que fornece informações de configuração mantém o registro do estado e das características do nó destinatário. Ou seja, o nó destino se encarrega de se configurar e o nó origem passa informações específicas para a configuração de cada dispositivo e as matém gravadas em um log. DHCPv6 é um exemplo de mecanismo que atua com esse modo de operação.

4.2. Autoconfiguração Stateless de endereços: interna do dispositivo

A autoconfiguração stateless de endereços é realizada internamente em todos dispositivos que queiram se comunicar utilizando IPv6. Ela inicia-se com a criação de um endereço *unicast link-local* nas interfaces dos dispositivos assim que esses se conectam a rede. Esse endereço é formado com a junção do prefixo (FE80::/64) com o identificador da interface física da máquina. Existem diversas implementações para a geração desse identificador, a mais comum é baseada no *MAC address*.

Após concluir o processo de montagem do endereço, ele precisa ser identificado como único no enlace antes de ser adicionado a interface. Portanto, aplica-se o procedimento de detecção de endereços duplicados (*Duplication Address Detection*), explicado no item 3.1. A partir de então, é possível que o nó troque pacotes no enlace.

Para mais informações sobre esta montagem do endereço de *link local*, o leitor deverá ler o capítulo sobre endereçamento.

4.3. Autoconfiguração Stateless: Router Advertisement

Autoconfiguração Stateless através da mensagem Router Advertisement é um procedimento utilizado por roteadores para transmitir informações aos dispositivos, o que engloba desde características do enlace e da rede até prefixos utilizados pelos dispositivos na criação de endereços globais.

Existem duas maneiras para um nó adquirir esses dados. Uma delas provém do próprio dispositivo interessado na autoconfiguração, quando realiza o envio de uma requisição com a mensagem *Router Solicitation* aos roteadores da rede. Essa mensagem ocasiona a geração de uma resposta *Router Advertisement* com as informações para autoconfiguração do dispositivo. A outra maneira provém de iniciativa dos próprios roteadores que periodicamente enviam mensagens *Router Advertisement* para anunciar sua presença na rede e instruir os nós para autoconfiguração.

Independentemente do modo, após o recebimento da mensagem *Router Advertisement*, inicia-se o procedimento de autoconfiguração (contudo isso não ocorre caso ele já tenha passado pelo mesmo processo anteriormente). Se for enviado um prefixo em conjunto com uma flag *AdvAutonomous* setada, endereço global será criado com o auxílio do identificador de interface, cuja implementação mais comum é baseada no endereço MAC (leia mais no capítulo de endereçamento). Logo em seguida é feito o procedimento de detecção de endereços duplicados para evitar que o endereço criado seja aceito na interface caso já exista alguém o utilizando. As demais informações enviadas, como endereço de DNS, MTU e outras, são também processadas e adicionadas às configurações dos dispositivos. Nenhum outro procedimento precisa ser realizado nenhum outro procedimento extra para se configurar.

No sistema operacional Linux existem diversas implementações para se aplicar a autoconfiguração *stateless* com *Router Advertisement*. Selecionamos duas:

- **Radvd:** é uma ferramenta que envia *Router Advertisements* pré-configurados. A edição do seu arquivo de configuração permite discriminar interfaces e os dados que serão enviados por cada uma delas.
- **Quagga:** é um serviço de roteamento que suporta uma variedade de protocolos. Um deles é o *Neighbor Discovery*, que possibilita o envio mensagens *Router Advertisement* configurados.

No mercado, algumas empresas são responsáveis por desenvolver roteadores físicos com suporte a IPv6 que implementam esta funcionalidade como por exemplo, Cisco, Juniper e Mikrotik.

4.3.1. Exemplo

A topologia deste exemplo é constituída de um computador conectado a um roteador. Nessa situação o ‘Cliente’ requisita informações da rede ao roteador que responde via *Router Advertisement*.

Cada dispositivo possui uma diferente descrição:

- Cliente: possui o endereço IPv6 de *link local* (fe80::200:ff:fea:0) e ainda não adquiriu endereço global que permita conexão com a Internet. Então, ele requisita ao roteador informações para realizar sua autoconfiguração.
- Roteador: possui o endereço IPv6 (2001:db8::11) e conhece os dados da rede, do enlace e o prefixo, utilizados para que ‘Cliente’ se autoconfigure.

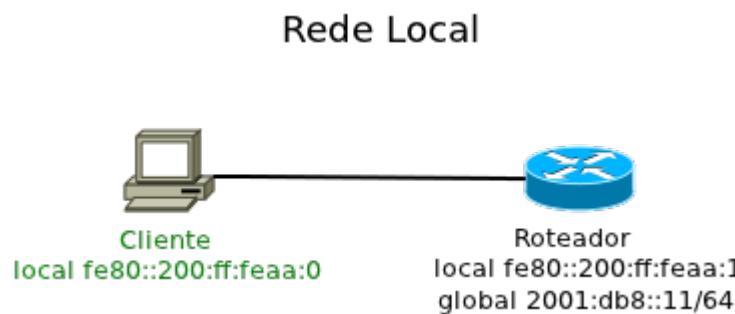


Figura 26: Topologia do exemplo da funcionalidade Autoconfiguração Stateless via router advertisement

Neste exemplo o cliente envia uma mensagem *Router Solicitation* para o roteador. Esse, responde com uma mensagem *Router Advertisement* que será processada e utilizada para autoconfiguração do dispositivo.

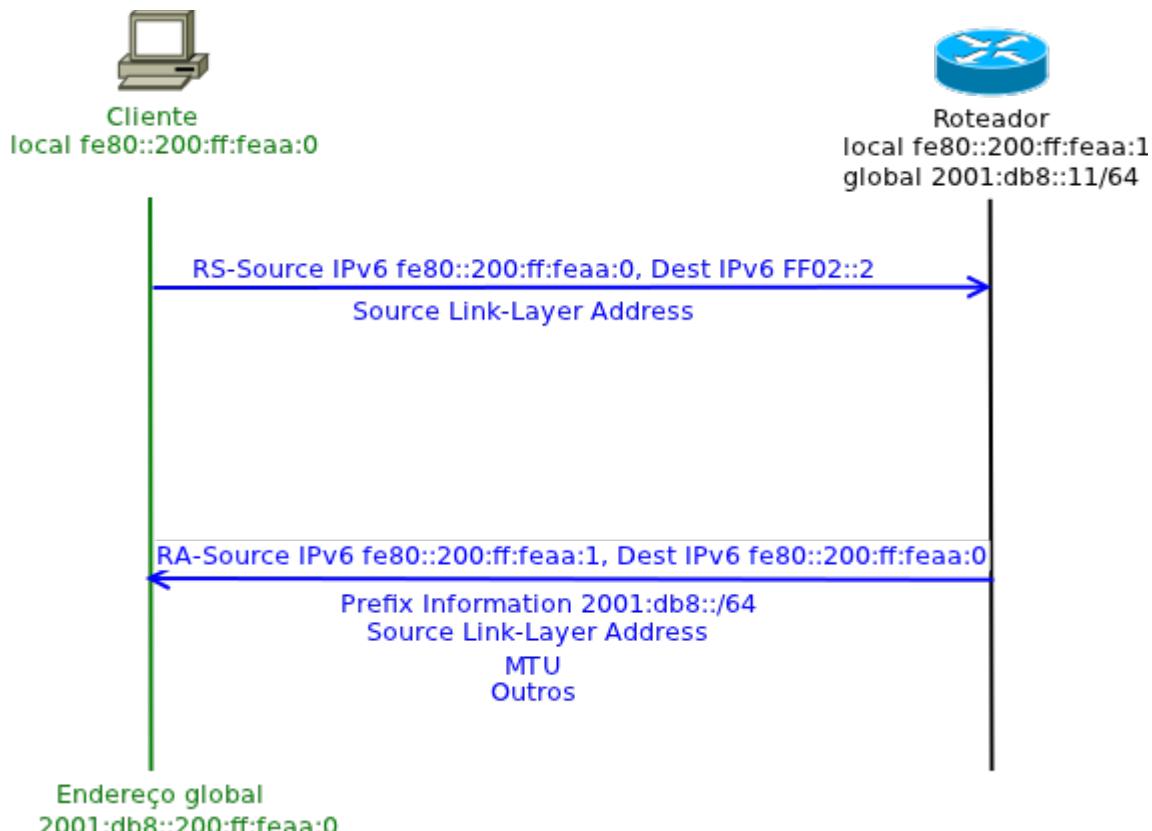


Figura 27: Troca de mensagens do exemplo da funcionalidade Autoconfiguração Stateless via Router Advertisement

Após a criação do endereço global, convém lembrar que antes do dispositivo adicioná-lo à interface, o procedimento de detecção de endereços duplicados (Duplicate Address Resolution) deve ser executado acontecer.

4.4. Autoconfiguração: DHCPv6

O Dynamic Host Configuration Protocol (DHCPv6) é um protocolo de configuração dinâmica de endereços stateful utilizado tanto para distribuir endereços IPv6 quanto para divulgar informações de rede.

Da mesma forma que sua versão predecessora, as mensagens pertencem a camada de transporte (camada 4). Entretanto, elas não são compatíveis entre si e atuam de forma independente. O DHCPv6 utiliza as portas UDP 546 para clientes e 547 para roteadores(relay agents) e servidores enquanto o DHCPv4 utiliza a UDP 68 e UDP 67.

A arquitetura é baseada no modelo cliente-servidor. Em cada rede deve haver ao menos um servidor DHCPv6 capaz de enviar dados para os clientes se configurarem. Normalmente, os clientes se comunicam com esses servidores utilizando seus endereços de *link local* como origem. Mas, outros endereços podem ser utilizados dependendo do servidor. O endereço de destino dessas mensagens é o All_DHCP_Relay_Agents_and_Servers, abaixo definido:

- **All_DHCP_Relay_Agents_and_Servers (FF02::1:2):** endereço multicast com escopo de enlace usado para que clientes enviem mensagens aos roteadores (*relay agent*) e aos servidores que se localizam na vizinhança.

Todavia, nem todos as comunicações DHCPv6 acontecem dentro do mesmo enlace. Nesse caso devem ser utilizados roteadores (*relay agent*) para retransmitir tanto as mensagens do cliente quanto as do servidor (leia mais no item influência dos roteadores no DHCPv6). Para isso, é utilizado o endereço de destino All_DHCP_Servers abaixo definido:

- **All_DHCP_Servers (FF05::1:3):** endereço multicast de escopo de site, usado pelos roteadores (*relay agent*) para se comunicarem com os servidores DHCPv6 ao retransmitirem as mensagens recebida dos clientes.

No protocolo DHCPv6 estão definidos 13 tipos de mensagens que podem ser utilizados para troca de informação entre clientes e servidores, com ou sem roteadores no meio do caminho. Abaixo está definido cada uma delas:

- **Solicit(1):** é enviada por um cliente no enlace com o intuito de encontrar servidores DHCPv6;
- **Advertise(2):** é enviada por um servidor DHCPv6 como resposta a mensagens *Solicit* de clientes, para indicar que ele está apto a fornecer as informações de configuração necessárias;
- **Request(3):** é enviada por um cliente a um servidor DHCPv6 específico para requisitar os dados de configuração;
- **Confirm(4):** é enviada por um cliente para qualquer servidor DHCPv6 disponível, para descobrir se o endereço autoconfigurado ainda é apropriado para uso;
- **Renew(5):** é enviada por um cliente para o servidor DHCPv6, que forneceu informações para autoconfiguração, com o intuito de atualizar os parâmetros configurados e de extender o tempo de vida do endereço recebido.
- **Rebind(6):** é enviada por um cliente para qualquer servidor DHCPv6 com o intuito de atualizar os parâmetros configurados e de extender o tempo de vida do endereço recebido. Essa mensagem é enviada depois do cliente receber uma resposta para a mensagem *Renew*.

- **Reply(7)**: é enviada por um servidor DHCPv6 a um cliente contendo o endereço e os parametros que devem ser utilizados para autoconfiguração. Ela serve como resposta à mensagem *Solicit*, *Request*, *Renew* e *Rebind*. Contudo, existem casos em que ela pode ser usada para outros objetivos. Um deles é como resposta à mensagem *Information-request* contendo os parametros de configuração. Outro, é em resposta à mensagem *Confirm* para confirmar ou renegar o endereço autoconfigurado no enlace. Por último, para informar que recebeu as mensagens *Release* ou *Decline*.
- **Release(8)**: é enviada por um cliente para o servidor DHCPv6, que forneceu informações para autoconfiguração, com o intuito de informar que não irá mais utilizar o(s) endereço(s) configurado(s).
- **Decline(9)**: é enviada por um cliente para o servidor DHCPv6 com o intuito de informar que o(s) endereço(s) que foi(ram) transmitido(s) para autoconfiguração já está(ão) sendo utilizado(s) no enlace do cliente.
- **Reconfigure(10)**: é enviada por um servidor DHCPv6 a um cliente já configurado com o intuito de informá-lo que existe algum parametro de configuração novo ou que precisa ser atualizado. A partir disso, inicia-se ou a transação *Renew/Reply* ou a transação *Information-request/Reply* para atualizar o cliente.
- **Information-Request(11)**: é enviada por um cliente para um servidor DHCPv6 com o intuito de solicitar a transmissão de parametros de configuração sem, necessariamente, o endereço IPv6 para a execução da autoconfiguração.
- **Relay-Forw(12)**: é enviada por um roteador (*relay agent*) para retransmitir as mensagens do protocolo recebidas aos servidores DHCPv6 ou a outro roteador (*relay agent*). As mensagens recebidas tanto de clientes ou quanto de um outros roteadores são encapsuladas nas opções das mensagens *Relay-Forw*.
- **Relay-Repl(13)**: é enviada por um servidor DHCPv6 para um roteador com a mensagem que deve ser transmitida ao cliente. Esta mensagem pode ser retransmitida entre roteadores até chegar no cliente, uma vez que, a mensagem do cliente se encontra encapsulada nas opções da mensagem *Relay-Repl*. O último roteador deve extraí-la e enviá-la ao cliente.

4.4.1. Formato do pacote DHCPv6

No DHCPv6, todas as mensagens trocadas entre clientes e servidores que se encontram em um mesmo enlace utilizam o seguinte modelo:

- O campo *Msg-type* de 8 bits: especifica o tipo da mensagem dentro do protocolo e assim limita as opções da mensagem (campo *Options*). Um exemplo de seu uso é o valor 1 que representa uma mensagem *Solicit*.
- O campo *Transaction-id* de 24 bits: apresenta o código de identificação da transação, assim num fluxo de mensagens é possível saber se a mensagem é uma resposta a uma determinada solicitação.
- O campo *Options* de tamanho variável: utilizado para transmitir informações extras que ajudarão no mecanismo de autoconfiguração.

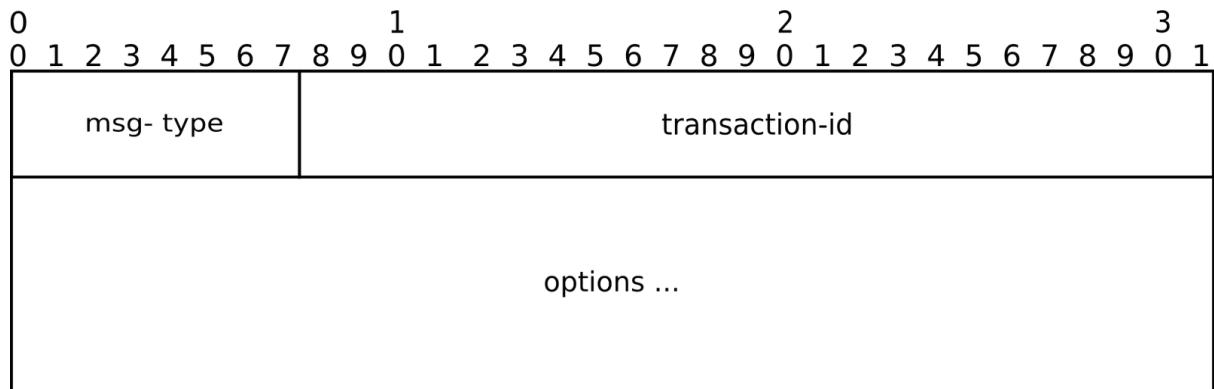


Figura 28: Formato do pacote DHCPv6

Contudo, quando existe um roteador (*relay agent*) no meio da comunicação, os pacotes tem de ser transformados antes de serem enviados pelo servidor. Como clientes enviam pacotes para um endereço *multicast*, eles não precisam modificar seu comportamento. Os pacotes alterados possuem o seguinte formato:

- O campo *Msg-type* de 8 bits: especifica o tipo da mensagem dentro do protocolo e assim limita as opções da mensagem (campo *Options*). Existem nesse caso apenas duas opções: o *Relay-Forw* e o *Relay- Repl.*
- O campo *Hop-count* de 8 bits: contabiliza o número de roteadores (*relay agents*) que foram atravessados antes da requisição até o servidor DHCPv6.
- O campo *Link-address* de 128 bits: contém o endereço global ou *site-local* que será utilizado para que o servidor identifique o link no qual o cliente está localizado.
- O campo *Peer-address* de 128 bits: contém o endereço do cliente ou do roteador (*relay agent*) que enviou a mensagem.
- O campo *Options* de tamanho variável: utilizado para transmitir informações extras que ajudam no mecanismo de autoconfiguração. É obrigatório a inclusão de pelo menos da opção *Relay Message Option*.

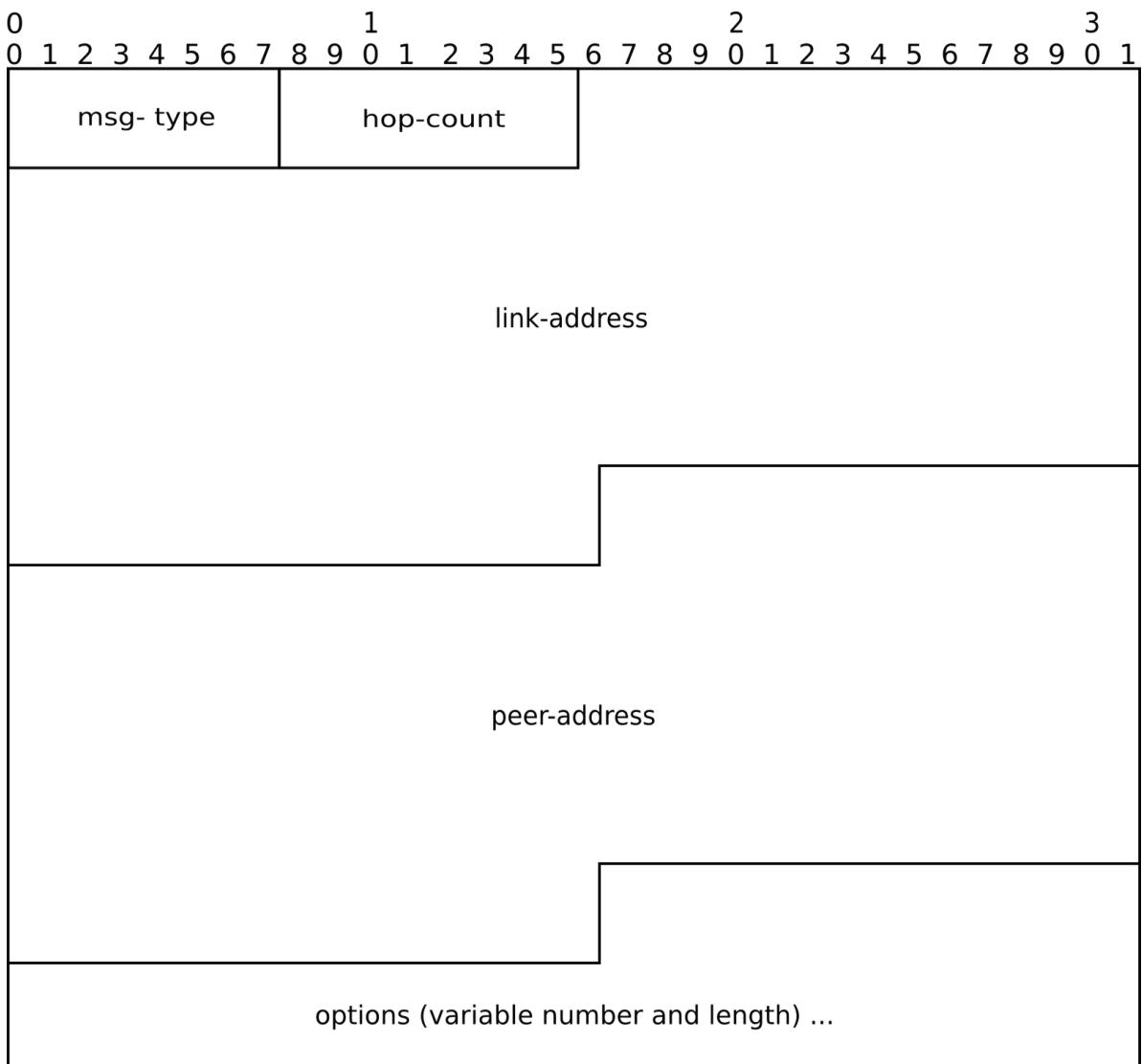


Figura 28: Formato do pacote DHCPv6 com influencia de um roteador (relay agent)

Há somente duas mensagens que utilizam este último tipo de pacote, a mensagem *Relay-Forw* e a mensagem *Relay-Repl*.

4.4.2. Formato do pacote DHCPv6 Options

Todos os pacotes opcionais do DHCPv6 possuem uma mesma estrutura padrão, a qual está apresentada a seguir:

- O campo *Option-code* de 16 bits: especifica o tipo do pacote opcional transmitido no campo *Options*.
- O campo *Option-len* de 16 bits: contém o tamanho utilizado no campo *option-data* nesse pacote.
- O campo *Option-data* de tamanho variável: contém os dados a serem transmitidos.

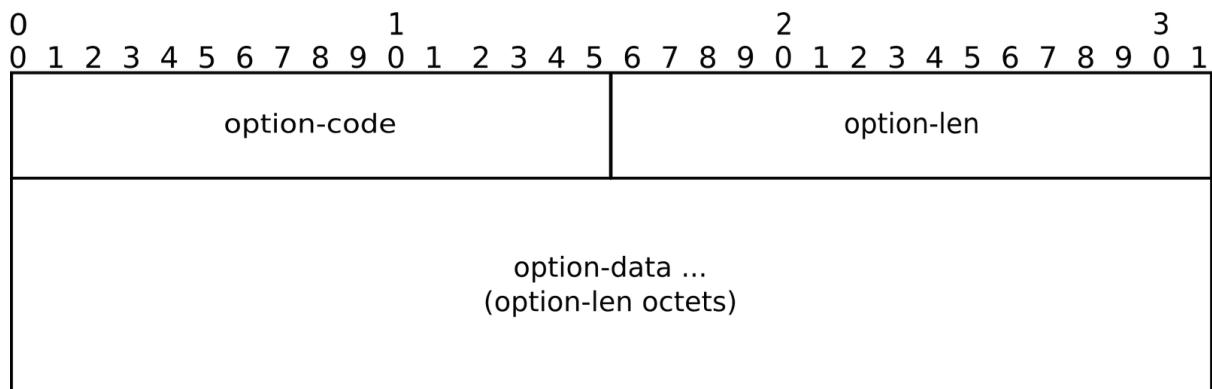


Figura 29: Formato genérico do pacote Options DHCPv6

Abaixo serão apresentado alguns exemplos de opções do DHCP que utilizam esta estrutura como base. Caso o leitor queira compreender mais sobre o assunto deverá ler a RFC3315.

4.4.2.1. Client Identifier Option

A opção Identificador de Cliente (*client identifier option*) é usada para transmitir o código DUID (*DHCP Unique Identifier*) que identifica o cliente na comunicação entre cliente e servidor.

O pacote transmitido deve ter as seguintes características:

- O campo *Option-code* com o valor 1 que representa o tipo *OPTION_CLIENTID*.
- O campo *Option-len* com o tamanho do DUID em octetos.
- O campo *Option-data* com as informações do DUID do cliente.

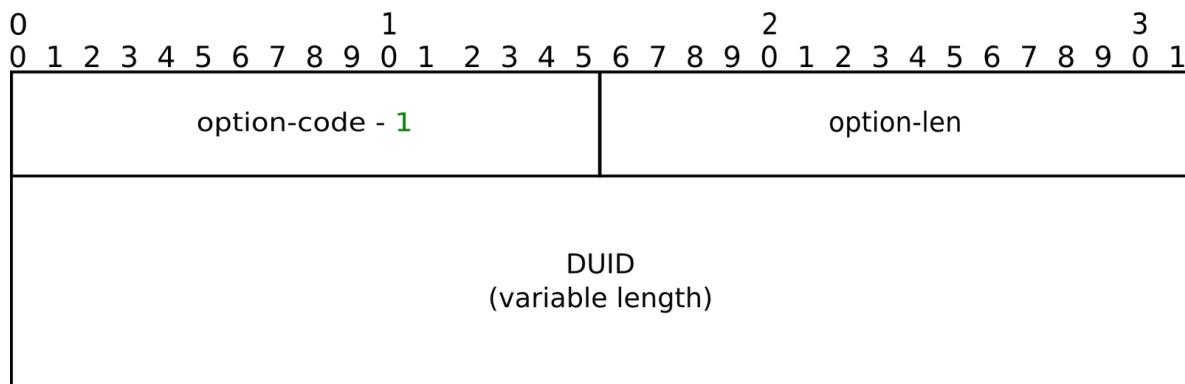


Figura 30: Formato do pacote Client Identifier Option DHCPv6

4.4.2.2. Server Identifier Option

A opção Identificador de Servidor (*server identifier option*) é usada para carregar o código DUID (*DHCP Unique Identifier*) que identifica o servidor durante comunicação entre cliente e servidor.

O pacote transmitido deve conter as seguintes características:

- O campo *Option-code* com o valor 2 que representa o tipo *OPTION_SERVERID*.
- O campo *Option-len* com o tamanho do DUID em octetos.
- O campo *Option-data* com as informações do DUID do servidor.

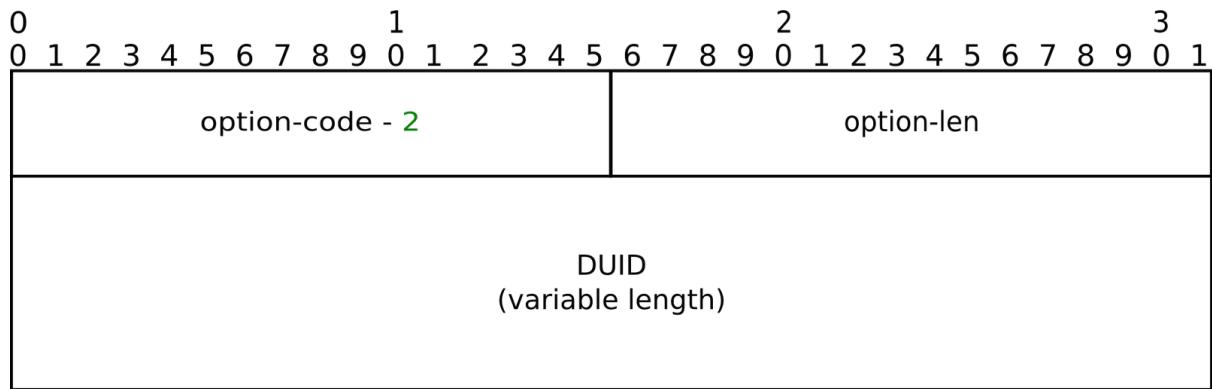


Figura 31: Formato do pacote Server Identifier Option DHCPv6

4.4.2.3. Identity Association for Non-temporary Addresses Option (OPTION_IA_NA)

A opção de Associação de Identidades para Endereços Não Temporários (Identity Association for Non-temporary Addresses option) é utilizada para transmitir uma associação de identidades (coleção de endereços delegados a um cliente), incluindo os endereços não temporários, em conjunto com seus parâmetros.

O pacote transmitido deve conter as seguintes características:

- O campo *Option-code* com o valor 3 que representa o tipo *OPTION_IA_NA*.
- O campo *Option-len* que contém o valor de 12 somado do tamanho do campo *IA-NA-options*.
- O campo *Option-data* é dividido em 4 sub-campos:
 - O campo IAID (Identity Association ID): contém o identificador único da delegação de endereço não temporário que se quer transmitir.
 - O campo T1 de 32 bits: contém o tempo em segundos em que o cliente deve recontactar o servidor, que delegou os endereços, para extender o tempo de vida dos endereços enviados.
 - O campo T2 de 32 bits: contém o tempo em segundos que o cliente deve contactar qualquer servidor disponível para extender o tempo de vida do(s) endereço(s) a ele atribuído(s).
 - O campo IA_NA-options de tamanho variável: contém informações específicas para associação que está sendo transmitida. Os endereços são exemplos de dados que são inseridos nesse campo.

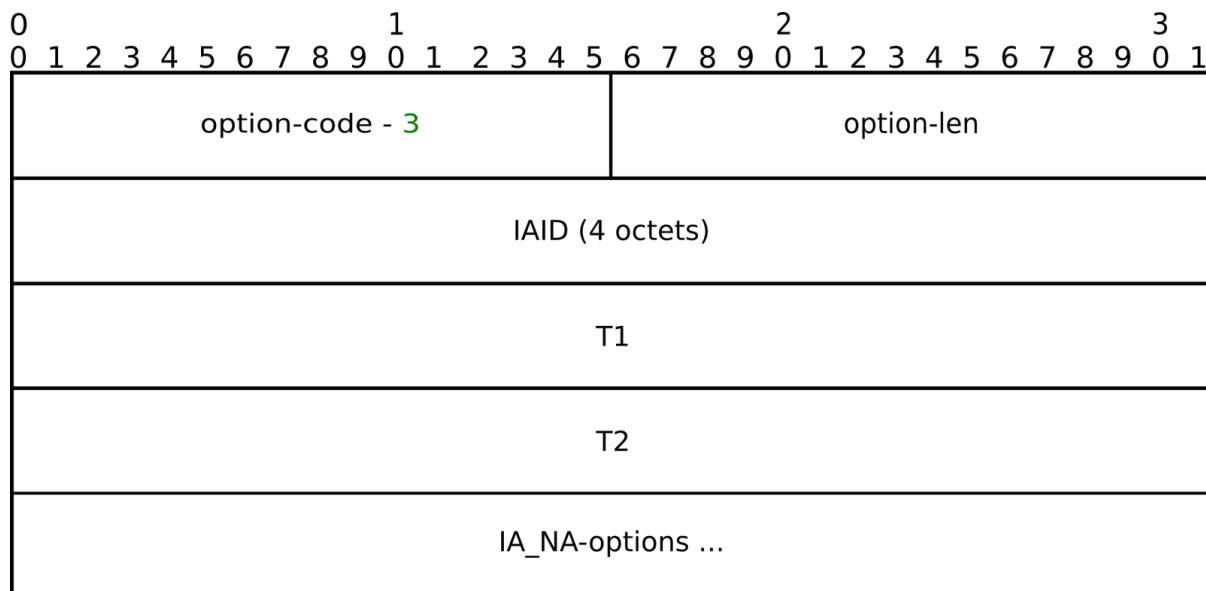


Figura 32: Formato do pacote Identity Association for Non-temporary Addresses Option DHCPv6

4.4.2.4. Identity Association for Prefix Delegation (OPTION_IA_PD)

A opção de Associação de Identidades para Delegação de Prefixos (Identity Association for prefix delegation) é utilizada para transmitir uma associação de identidades (coleção de prefixos delegados a um cliente), incluindo os prefixos, em conjunto com seus parâmetros.

O pacote transmitido deve conter as seguintes características:

- O campo *Option-code* com o valor 25 que representa o tipo *OPTION_IA_PD*.
- O campo *Option-len* que contém o valor de 12 somado do tamanho do campo *IA-PD-options*.
- O campo *Option-data* é dividido em 4 sub-campos:
 - O campo IAID (Identity Association ID): contém o identificador único da delegação de prefixos que se quer transmitir.
 - O campo T1 de 32 bits: contém o tempo em segundos em que o cliente deve recontactar o servidor, que delegou o(s) prefixo(s), para extender o tempo de vida do(s) prefixo(s) enviado(s).
 - O campo T2 de 32 bits: contém o tempo em segundos que o cliente deve contactar qualquer servidor disponível para extender o tempo de vida do(s) prefixo(s) a ele atribuído(s).
 - O campo IA_PD-options de tamanho variável: contém informações específicas para associação que está sendo transmitida. Os prefixos são exemplos de dados que são inseridos nesse campo.

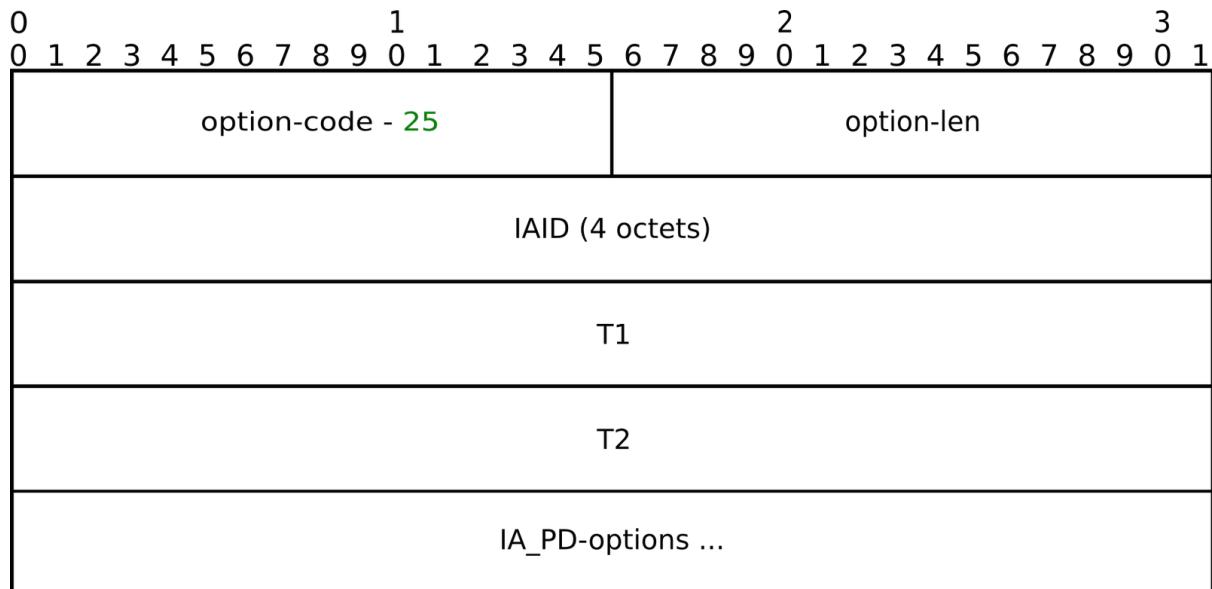


Figura 33: Formato do pacote Identity Association for Prefix Delegation Option DHCPv6

4.4.2.5. Option Request Option

A opção de Requisição de Opções (Option Request Option) é utilizada para que um cliente requisite uma lista de opções a um servidor.

Essa opção pode ser incluída em mensagens do tipo *Solicit*, *Request*, *Renew*, *Rebind*, *Confirm* ou *Information-request* para informar quais informações o cliente quer o servidor inclua em sua resposta. E, ela pode ser incluída em uma mensagem do tipo *Reconfigure* para que o servidor informe ao cliente quais informações ele deveria requisitar.

O pacote transmitido deve conter as seguintes características:

- O campo *Option-code* com o valor 6 que representa o tipo *OPTION_ORO*.
- O campo *Option-len* que contém o valor de 2 multiplicado pelo número de opções requisitadas.
- O campo *Request-option-code-n* que contém o código da opção requisitada.

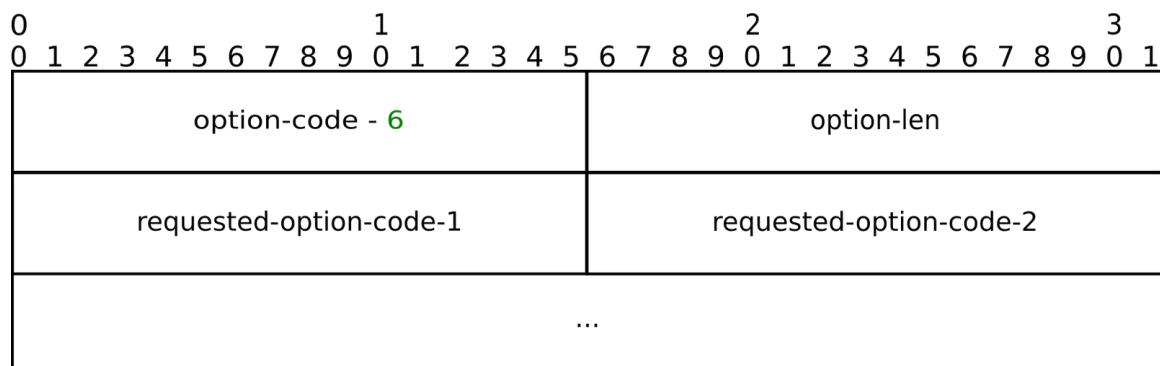


Figura 34: Formato do pacote Option Request Option DHCPv6

4.4.2.6. Elapsed Time Option

A opção Tempo Decorrido (*Elapsed Time Option*) serve para que o cliente indique o tempo que ele passou tentando completar uma troca de mensagens DHCP. O tempo é medido em relação a primeira troca de mensagem, que será enviada com o valor 0.

Essa mensagem tem obrigatoriamente que ser enviada pelo cliente e é utilizada por roteadores (*relay agents*) e servidores para controlar política de resposta à clientes. Por exemplo, um segundo servidor DHCP pode responder no lugar de um outro não tenha respondido em tempo rasoável.

O pacote transmitido deve conter as seguintes características:

- O campo *Option-code* com o valor 8 que representa o tipo *OPTION_ELAPSED_TIME*.
- O campo *Option-len* que contém o valor 2.
- O campo *Elapsed-time* de 16 bits: contém a quantidade de tempo passado desde o momento que o cliente começou a troca de mensagens DHCP. O tempo neste campo é medido em centésimo de segundo (10^{-2} segundos).

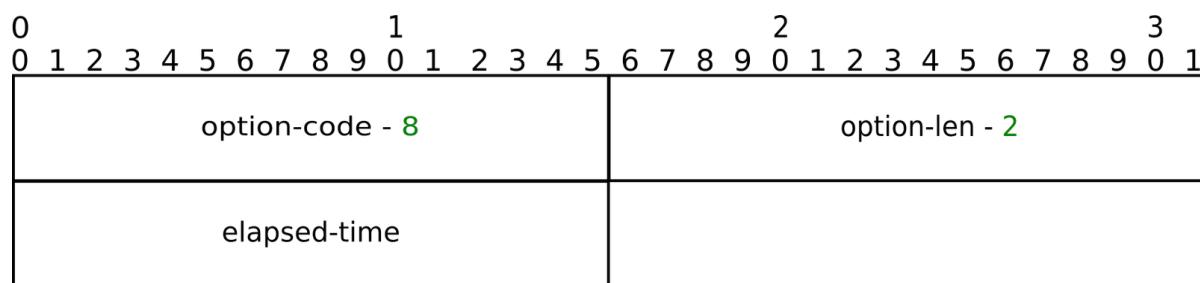


Figura 35: Formato do pacote Elapsed Time Option DHCPv6

4.4.3. DHCPv6 em modo Stateful

Conforme explicado no item 4.1, o modo de operação *Stateful* indica que as informações transmitidas foram armazenadas de forma centralizada. No caso DHCPv6 ocorre apenas o registro da entrega de dados relacionados a atribuição de endereços IPv6. As demais, não são registradas.

O procedimento é iniciado com uma requisição de um cliente aos servidores da rede por um endereço global IPv6. Esse pedido é feito com uma mensagem *Solicit* enviada pelo cliente, utilizando seu endereço de *link local* como origem, para o destino *multicast All_DHCP_Relay_Agents_and_Servers* (FF02::1:2). Os servidores DHCPv6, que receberem e estiverem com seus serviços habilitados, respondem diretamente ao cliente (destino *link local*) com a mensagem *Advertise*. Essa última mensagem carrega o número identificador do servidor, o identificador do cliente e o endereço global requisitado, que servem para que o cliente possa escolher o servidor com o qual quer se comunicar. Após a escolha, o cliente inicia o procedimento de negociação do endereço. Para isso, envia uma mensagem *Request*, com seu *link local* como endereço de origem, para o destino *multicast All_DHCP_Relay_Agents_and_Servers*. Apesar dessa mensagem alcançar todos os servidores da rede, somente o dono do número identificador do servidor irá responder. Ao recebê-la, o servidor armazena as informações do cliente e o endereço atribuído em registro e envia uma mensagem *Reply* diretamente ao cliente. Ao chegar no cliente, essa resposta dispara o processo de adição do endereço a interface, a qual engloba a funcionalidade de detecção de endereços duplicados.

Existe uma configuração para o cliente, chamada *rapid commit*, que permite a troca de informações com apenas duas mensagens ao invés das quatro utilizadas acima. Contudo, ela só é aconselhável quando a rede possui apenas um servidor ou, quando existem muitos endereços a serem resolvidos.

4.4.3.1. Exemplo

A topologia deste exemplo é constituída por um computador conectado a um servidor DHCPv6. Nessa situação o cliente requisita um endereço global ao servidor DHCPv6. A partir disso, uma negociação acontece entre as duas máquinas até que um endereço seja reservado para o cliente.

A descrição dos dispositivos que compõe tal topologia é a seguinte:

- Cliente: possui o endereço IPv6 de *link local* (FE80::200:FF:FEAA:0) e ainda não adquiriu endereço global que permita sua conexão com a Internet. Ele requisita ao servidor um endereço para realizar sua autoconfiguração.
- ServidorDHCPv6: possui o endereço global IPv6 (2001:db8::11) e administra um conjunto (*pool*) de endereços IPv6 do qual disponibilizará um para que a máquina ‘Cliente’ se autoconfigure.



Figura 36: Topologia do exemplo da funcionalidade Autoconfiguração Stateful DHCPv6

Neste exemplo, o ‘Cliente’ envia uma mensagem *Solicit* para a rede procurando por servidores habilitados. O ‘ServidorDHCPv6’ responde com uma mensagem *Advertise* se anunciando para fornecer informações. Essa mensagem carrega o endereço solicitado para auxiliar o ‘Cliente’ na escolha entre os servidores que responderam. Então, ele elege um servidor (no exemplo só há um servidor) e envia uma mensagem *Request* para requisitar permissão de uso do endereço global passado. O ‘ServidorDHCPv6’ armazena, num registro, o endereço passado ao ‘Cliente’ e manda uma mensagem *Reply* como confirmação.

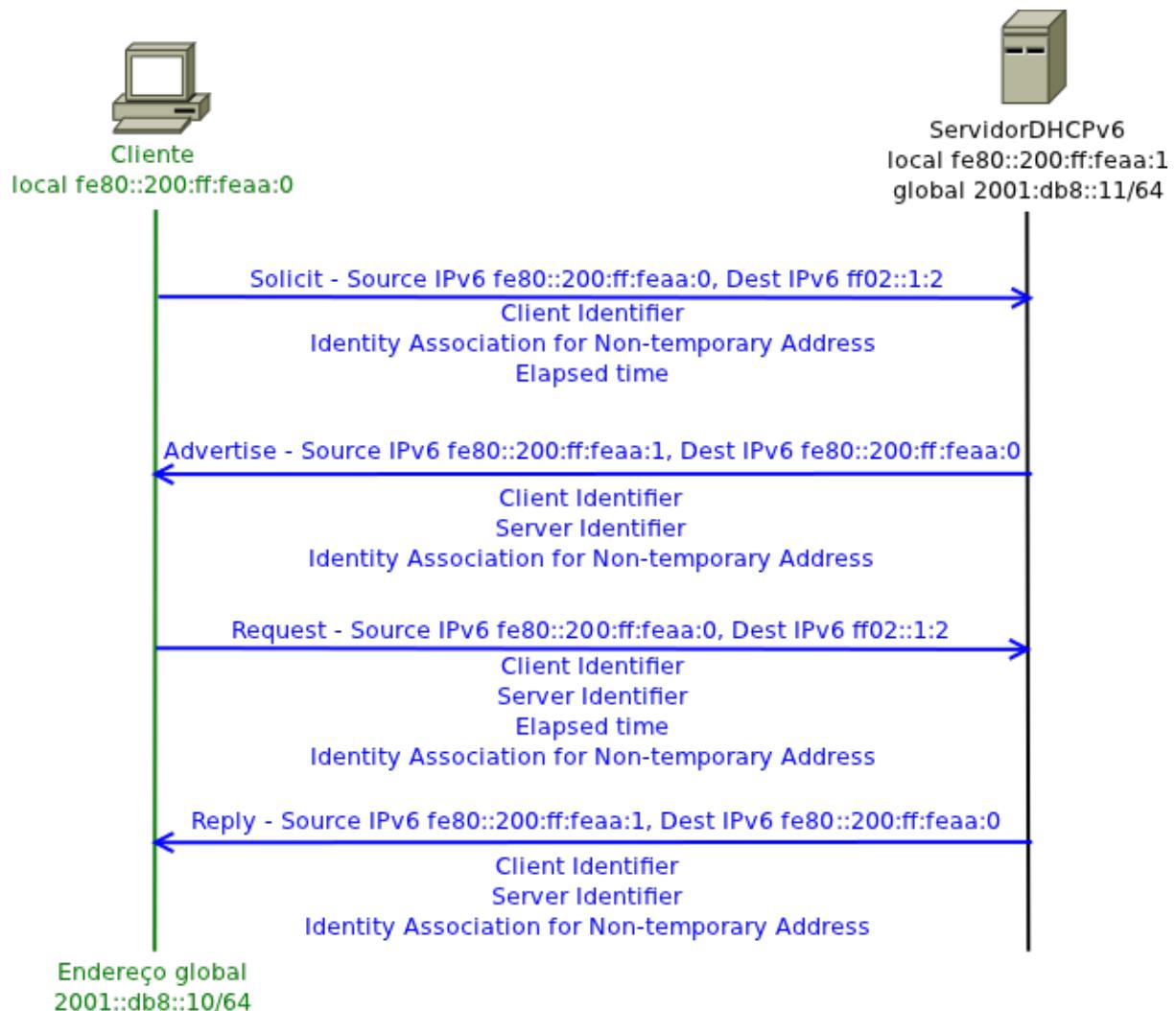


Figura 37: Troca de mensagens do exemplo da funcionalidade Autoconfiguração Stateful via DHCPv6

Após a última mensagem, a *Reply*, o ‘Cliente’ iniciar o procedimento de detecção de endereços duplicados no enlace. E, só a partir de então utiliza o endereço em suas comunicações.

4.4.4. DHCPv6 em modo stateless

Outro modo de operação do DHCPv6 é o *stateless*, no qual não há registro das informações trocadas entre o cliente e o servidor. Neste modo também não ocorre delegação de endereços IPv6, somente informações adicionais como DNS, NTP e outras são comunicadas.

O procedimento é iniciado pelo cliente através do envio de uma mensagem *Information-request* para o destino *All_DHCP_Relay_Agents_and_Servers* (FF02::1:2). Esta mensagem informa aos servidores DHCPv6 quais os dados que estão sendo requisitados pelos clientes. Logo, todos os servidores habilitados respondem diretamente ao cliente com uma mensagem *Reply* contendo os itens pedidos. Dessa maneira o cliente pode se autoconfigurar utilizando o DHCPv6 em modo stateless.

4.4.4.1. Exemplo

A topologia deste exemplo é constituída de um computador conectado a um servidor DHCPv6. Nessa situação o cliente requisita somente informações adicionais (DNS) ao servidor DHCPv6.

A descrição dos dispositivos que compõe tal topologia é a seguinte:

- Cliente: possui os endereços IPv6 *global* (2001:DB8::10) e *link local* (FE80::200:FF:FEAA:0) e ainda não adquiriu informações adicionais da rede (DNS) que permitam sua conexão com a Internet. Então, ele requisita ao ‘ServidorDHCPv6’ estes dados.
- ServidorDHCPv6: possui os endereços IPv6 *global* (2001:db8::11) e *link local* (FE80::200:FF:FEAA:1). Possui, também, conhecimento das informações adicionais da rede (DNS) que o ‘Cliente’ precisa para se autoconfigurar.



Figura 38: Topologia do exemplo da funcionalidade Autoconfiguração Stateless DHCPv6

Neste exemplo, o ‘Cliente’ envia uma mensagem *Information-request* para a rede procurando por servidores habilitados que possam mandar informações adicionais (DNS). O ‘ServidorDHCPv6’ responde com uma mensagem *Reply* fornecendo os dados pedidos.

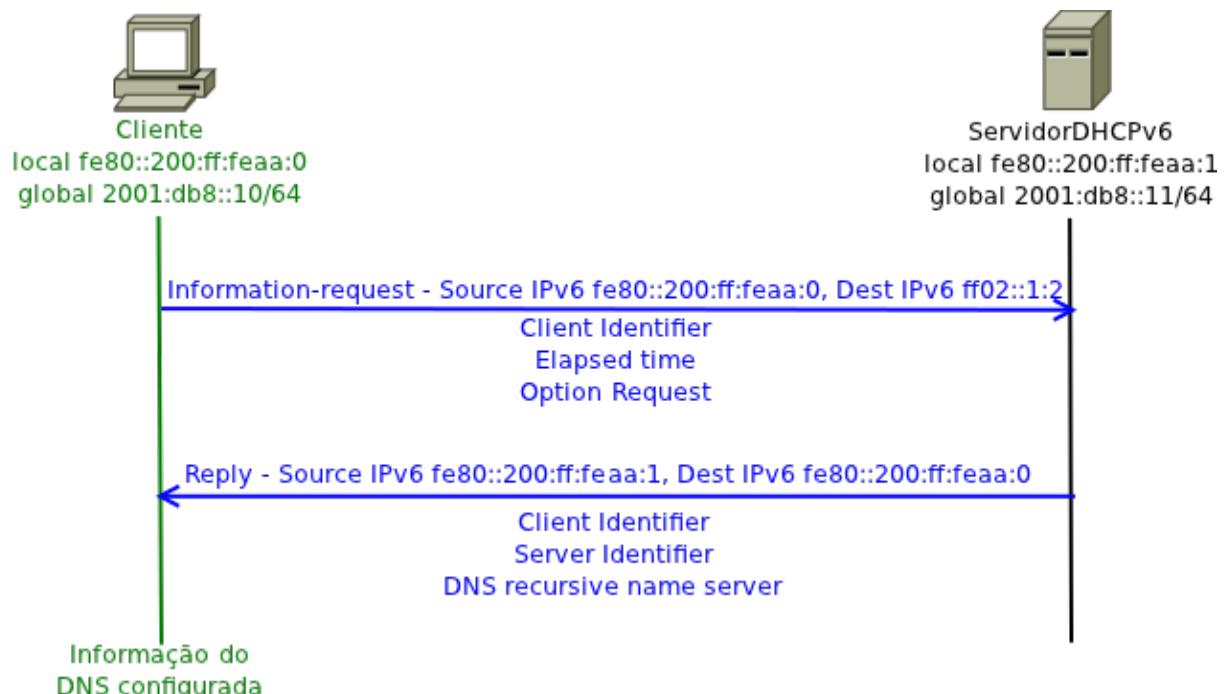


Figura 39: Troca de mensagens do exemplo da funcionalidade Autoconfiguração Stateless via DHCPv6

Nenhum registro é criado no servidor pois estes dados são distribuídos em modo stateless.

4.4.5. DHCPv6 prefix delegation

Uma funcionalidade desenvolvida para o DHCPv6 e que não existe no DHCPv4 é o *prefix delegation*. Ela serve para distribuir prefixos de rede para roteadores. Esses prefixos ao serem recebidos são repartidos em prefixos menores e redistribuídos para a reliazação da autoconfiguração entre os dispositivos de um enlace. Do lado do servidor DHCPv6, os dados trafegados são mantidos em logs, o que caracteriza o modo *stateful*. Já, do lado do roteador, as trocas de mensagem para autoconfiguração dos nós operam em modo *steless*, sem log.

O procedimento é iniciado com um pedido a partir de um roteador (cliente) aos servidores da rede por um prefixo IPv6. Da mesma forma que a funcionalidade de autoconfiguração de endereços *stateful* via DHCPv6, a troca de informações acontece com as mesmas 4 mensagens. A primeira é a mensagem *Solicit* enviada pelo roteador com o endereço de *link local* como origem, um identificador do cliente e com o endereço *All_DHCP_Relay_Agents_and_Servers* (FF02::1:2) como destino para localizar os servidores DHCPv6 habilitados na rede. Esses, ao receberem essa mensagem, respondem com uma mensagem *Advertise* enviada diretamente ao cliente (origem e destino *link local*) com o prefixo pedido e um identificador de servidor. O prefixo transmitido é retirado de um conjunto de prefixos pré-determinados (pool). Ele precisa ser maior que /64 para que o roteador consiga subdividi-lo para a utilização no processo de autoconfiguração via *Router Advertisement*. O roteador, então, escolhe um servidor para se comunicar e faz um pedido para efetivar a delegação do prefixo. Para isso, é enviado uma mensagem *Request* com o endereço *link local* como origem e o endereço *All_DHCP_Relay_Agents_and_Servers* (FF02::1:2) como destino. Apesar dessa mensagem ser direcionada para um grupo *multicast*, só o servidor escolhido responderá, devido ao identificador presente nos dados transmitidos. Esse servidor armazena em log, o prefixo e os dados que o cliente irá receber, e envia uma mensagem *Reply* diretamente ao cliente (origem e destino *link local*) informando-o que o prefixo já pode ser utilizado.

Após o recebimento do prefixo, o roteador pode subdividí-lo em prefixos de tamanho /64 que serão distribuídos entre todas as suas interfaces habilitadas, com exceção da que efetuou a comunicação DHCPv6. Em seguida é realizada a autoconfiguração de endereços dos dispositivos conectados à esse roteador via *Router Advertisement*, conforme apresentado nos capítulos anteriores.

Esta funcionalidade é utilizada em situações em que o servidor DHCPv6 não possui nenhum conhecimento sobre a topologia de rede a qual o roteador requisitante está conectado e, também, quando desconhece outras informações além da identidade do roteador requisitante para escolher o prefixo.

Outras informações, como por exemplo DNS, também podem ser enviadas nesse procedimento de delegação de prefixos para ser repassados aos Clientes. Contudo esses dados não serão armazenados, caracterizando o modo *Stateless* de distribuição de informação.

4.4.5.1. Exemplo

A topologia deste exemplo é constituída de dois computadores conectados a um roteador, em interfaces distintas. O roteador também se encontra conectado a um servidor DHCPv6, porém em outra interface. Nessa situação o roteador requisita um prefixo ao servidor DHCPv6 para reparti-lo em prefixos /64 e, em seguida, enviá-los a seus clientes para que realizem a autoconfiguração de endereços *stateless*.

A descrição dos dispositivos que compõe tal topologia é a seguinte:

- ServidorDHCPv6: possui os endereços IPv6 *global* (2001:db8::11) e *link local* (FE80::200:FF:FEAA:1). Possui, também, um conjunto (pool) do qual prefixos podem ser requisitados por roteadores, para serem distribuídos entre a seus respectivos clientes.

- Roteador: possui três interfaces conectadas a diferentes dispositivos. A eth0 possui os endereços IPv6 *global* (2001:db8::10) e *link local* (FE80::200:FF:FEAA:0) e está ligada ao ‘ServidorDHCPv6’. A partir dessa interface é feita a requisição de prefixo para a redistribuição pelo roteador.
- A eth1 possui o endereço IPv6 *link local* (FE80::200:FF:FEAA:2) e está ligada ao ‘Cliente1’.
- A eth2 possui o endereço IPv6 *link local* (FE80::200:FF:FEAA:3) e está ligada ao ‘Cliente2’. Ambas as interfaces, eth1 e eth2, realizam a funcionalidade básica de autoconfiguração de endereços via *Router Advertisement*.
- Cliente1: possui apenas endereço IPv6 *link local* (FE80::200:FF:FEAA:4). Ainda não adquiriu um endereço global que permita sua conexão com a Internet. Realiza a autoconfiguração de endereço via *Router Advertisement*.
- Cliente2: possui apenas endereço IPv6 *link local* (FE80::200:FF:FEAA:5). Ainda não adquiriu um endereço global que permita sua conexão com a Internet. Realiza a autoconfiguração de endereço via *Router Advertisement*.

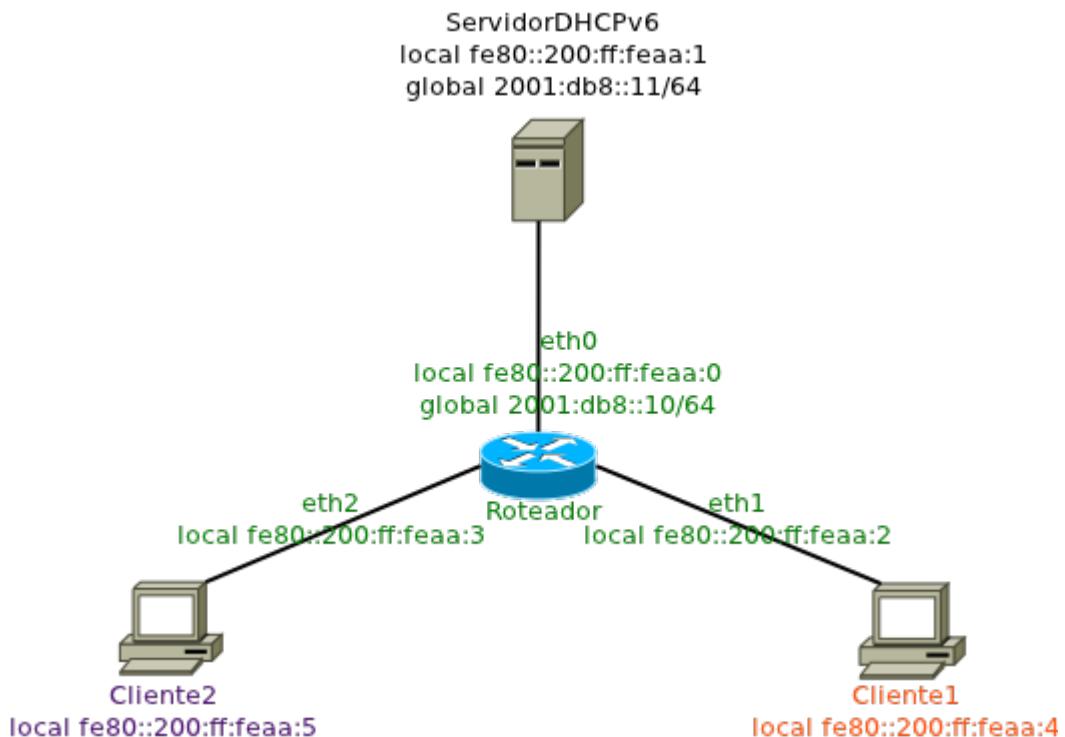


Figura 40: Topologia do exemplo da funcionalidade delegação de prefixes DHCPv6

Neste exemplo, o ‘Roteador’ envia uma mensagem *Solicit* para a rede a procura de servidores DHCPv6 habilitados que lhe possam fornecer um prefixo IPv6. O ‘ServidorDHCPv6’ responde com uma mensagem *Advertise* anunciando que pode enviar as informações pedidas. Esta mensagem carrega o prefixo solicitado para auxiliar o ‘Roteador’ na escolha de um servidor dentre todos os que responderam. Então, ele elege um (no exemplo só há um servidor) e envia uma mensagem *Request* para requisitar permissão de uso do prefixo passado. O ‘ServidorDHCPv6’ armazena, num registro, o prefixo passado e manda uma mensagem *Reply* como confirmação.

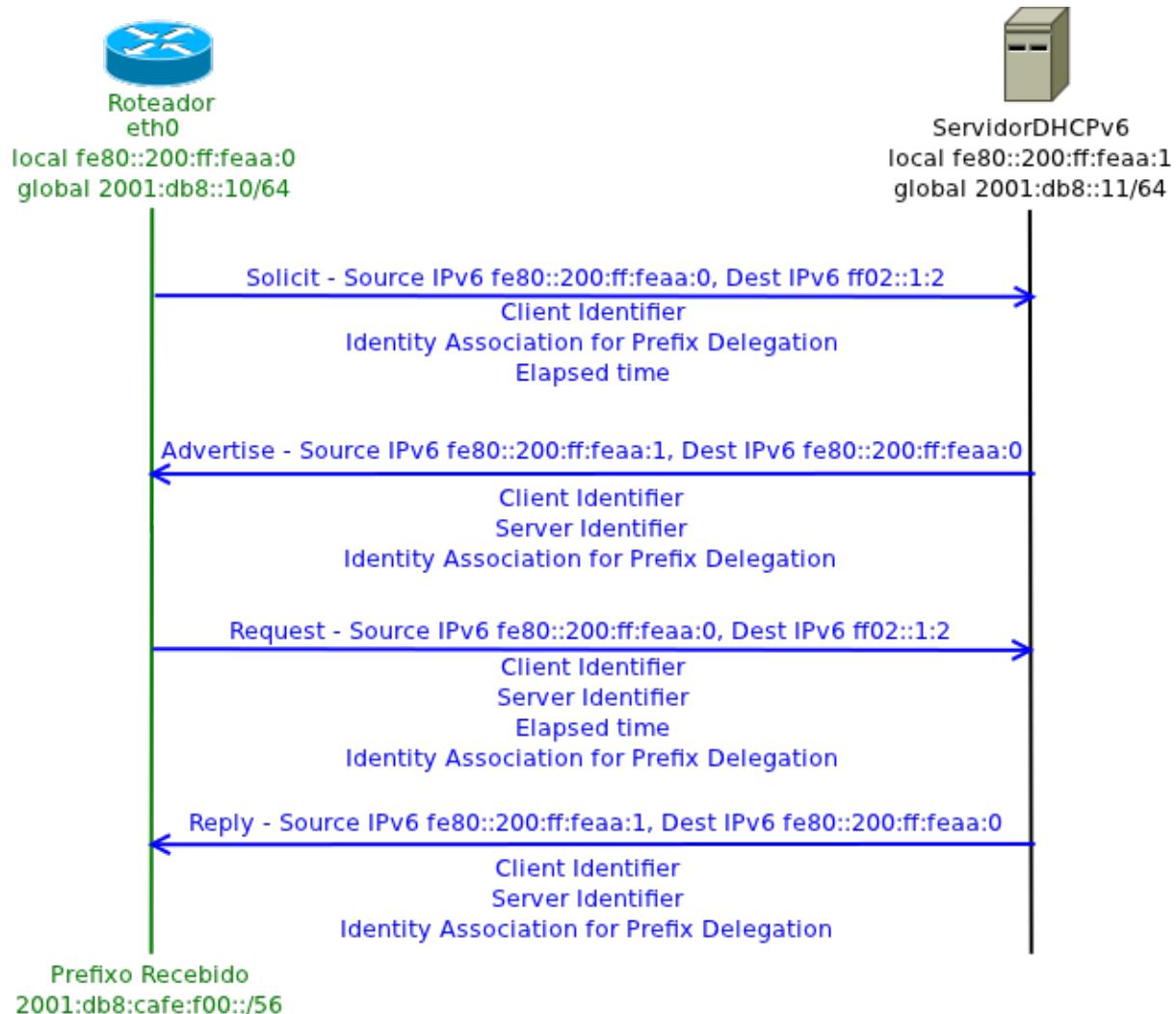


Figura 41: Troca de mensagens do exemplo da funcionalidade delegação de prefixes DHCPv6

Após a última mensagem, a *Reply*, o servidor mantém em registro a informação cedida (stateful) e o ‘Roteador’ inicia o procedimento de divisão do prefixo /56 recebido em prefixos /64 para cada uma de suas interfaces habilitadas. A partir disso é iniciada, de maneira automática, a funcionalidade de autoconfiguração de endereços stateless via Router Advertisement.

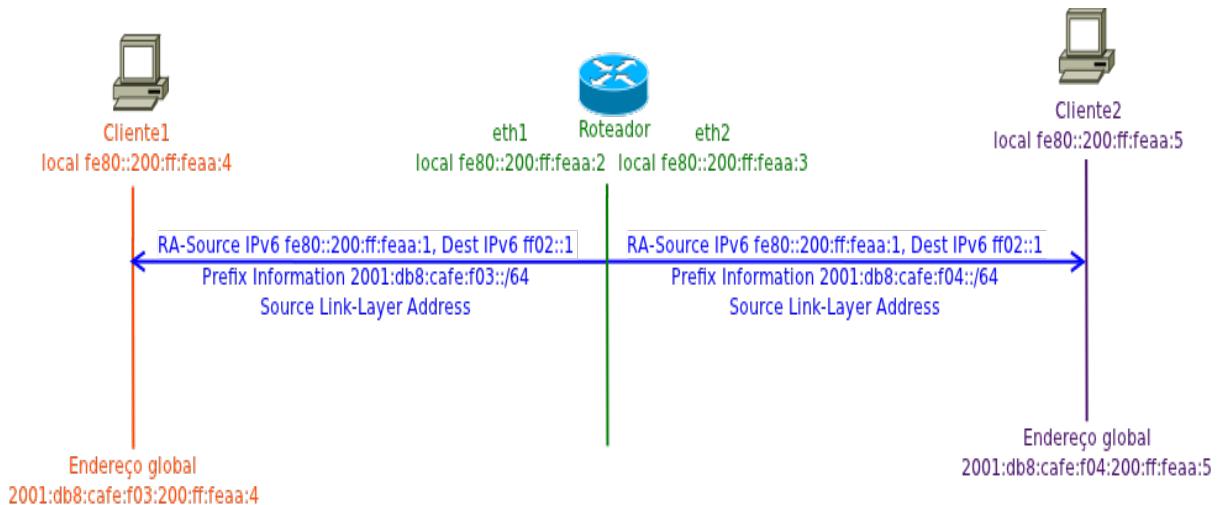


Figura 42: Troca de mensagens do exemplo da funcionalidade delegação de prefixes Router Advertisement

O cliente, ao final do processo, recebe um endereço IPv6 global que deve passar pela funcionalidade de detecção de endereços duplicados antes de ser adicionado a sua interface.

4.4.6. Influência dos roteadores no DHCPv6

Roteadores podem influenciar de duas maneiras os procedimentos do DHCPv6. Uma delas é quando eles se encontram no meio da comunicação entre um cliente e um servidor DHCPv6. Conforme exemplificado a seguir na Figura 43. Nesse caso, eles atuam como retransmissores das mensagens DHCPv6 trafegadas entre o cliente e o servidor.



Figura 43: Influencia do roteador no meio de uma comunicação DHCPv6

Do lado cliente, a comunicação é transparente, ou seja, as mensagens são trocadas como se o cliente estivesse em contato direto com o servidor.

Já do lado do servidor, as mensagens originais são encapsuladas nas opções de mensagens Relay-Forw (caso gerada num roteador) e Relay-Repl (caso gerada num servidor). Além disso, o endereço Multicast All_DHCP_Servers (FF05::1:3) é utilizado pelos roteadores na comunicação com os servidores.

Roteadores podem, também, estar dispostos entre um cliente e um servidor e impactarem na funcionalidade DHCPv6. Para tanto a comunicação deverá acontecer entre *agent relays* com as mesmas mensagens explicadas anteriormente, *Relay-Forw* e *Relay-Repl*.

A segunda maneira que um roteador pode alterar o comportamento do DHCPv6 é através da mensagem *Router Advertisement* do protocolo *Neighbor Discovery*. Nessa mensagem existem duas flags que, ao serem ativadas, modificam o modo do cliente operar a captura informações via DHCPv6. As flags são:

- *AdvManagedFlag(M)*: define que o endereço IPv6 se encontra disponível para autoconfiguração via DHCPv6.
- *AdvOtherConfigFlag(O)*: define que outras informações (exemplo: DNS) se encontram disponíveis para autoconfiguração via DHCPv6.

Caso nenhuma das *flags* esteja setada, o cliente não deverá buscar nenhuma informação no servidor DHCPv6.

5. Estado dos Endereços

Todo endereço IPv6, desde a sua criação, possui um estado de operação atrelado que indica como ele deve ser utilizado na rede. No total, são cinco esses estados:

- **Tentativa**: o endereço neste estado ainda não foi atribuído a uma interface principalmente porque o processo de detecção de endereços ainda não foi concluído. Todos os pacotes que forem direcionados a um endereço neste estado serão descartados.
- **Preferencial**: o endereço neste estado já foi atribuído a uma interface e, pode ser utilizado indistintamente para comunicação na rede. Ou seja, ele pode ser utilizado tanto como origem quanto como destino dos pacotes IPv6.
- **Depreciado**: o endereço neste estado já foi atribuído a uma interface, porém não pode ser utilizado como origem em novas comunicações. Contudo, os pacotes de comunicações previamente estabelecidas trafegam normalmente.
- **Válido**: o endereço neste estado já foi atribuído a uma interface e serve para designar os endereços tanto no estado preferencial quanto no depreciado.
- **Inválido**: o endereço neste estado não pode ser atribuído a uma interface e não deve ser utilizado para comunicações na rede. Ou seja, ele não pode ser utilizado nem como origem nem no destino de pacotes IPv6.

Existe uma ordem para esses estados serem designados aos endereços, o que remete a ideia de ciclo de vida. O endereço ao ser criado recebe o estado de tentativa. Com ele permanece até que o processo de detecção de endereços duplicados indique sua unicidade no enlace. A seguir, ele é adicionado a uma interface e recebe o estado preferencial, o que lhe permite ser utilizado em comunicações com a Internet. Após o término do tempo chamado *Preferred lifetime*, o endereço passa do estado preferencial para o depreciado, no qual ele serve apenas para manter as comunicações já em andamento. Passado o tempo designado como *Valid lifetime*, o endereço entra no estado Inválido e não pode mais ser utilizado.

Uma consideração que pode ser feita sobre o estado dos endereços, é a de que nem todo endereço passa por todos estados. Isso pode acontecer por dois motivos. O primeiro, provém da habilidade dos dispositivos de conseguirem reiniciar os contadores de tempo relacionados ao *Preferred lifetime* e ao *Valid Lifetime*. O segundo, é fato de que esses tempos não serem necessariamente limitados. Algumas implementações permitem que eles sejam infinitos.

Estados

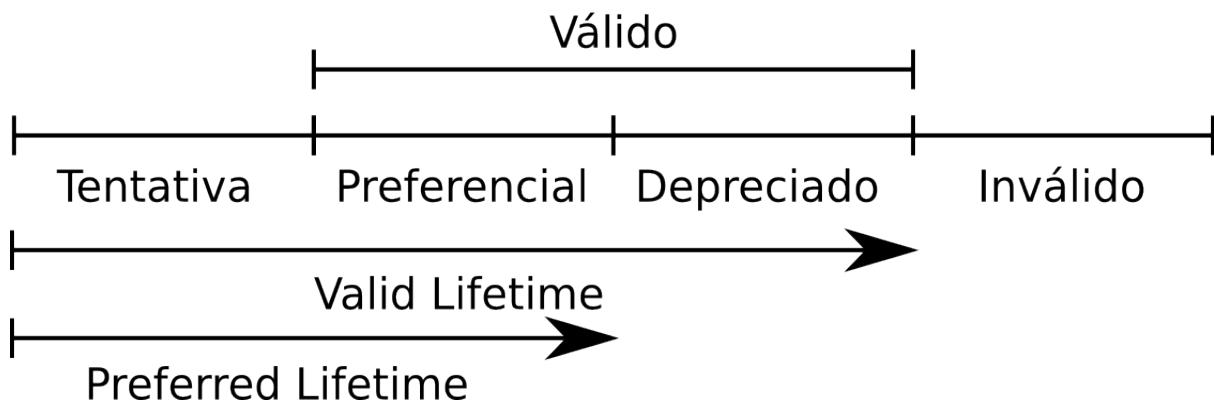


Figura 44: Estados dos endereços IPv6

6. Referencias

- RFC3315 - Dynamic Host Configuration Protocol for IPv6 (DHCPv6) - <http://tools.ietf.org/html/rfc3315>
- RFC3633 - IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6 - <http://tools.ietf.org/html/rfc3633>
- RFC 3775 - Mobility Support in IPv6 - <http://tools.ietf.org/html/rfc3775>
- RFC 4389 - Neighbor Discovery Proxies (ND Proxy) - <http://tools.ietf.org/html/rfc4389>
- RFC 4861 - Neighbor Discovery for IP version 6 (IPv6) - <http://tools.ietf.org/html/rfc4861>
- RFC4862 - IPv6 Stateless Address Autoconfiguration - <http://tools.ietf.org/html/rfc4862>
- RFC 5175 - IPv6 Router Advertisement Flags Option - <http://tools.ietf.org/html/rfc5175>
- RFC6106 - IPv6 Router Advertisement Options for DNS Configuration - <http://tools.ietf.org/html/rfc6106>
- Silvano Gai. Internetworking IPv6 with Cisco Routers. McGraw-Hill Computer Communications Series, first edition, 12/12/1997.

Andrew Hines. Neighbour Discovery in IPv6 Topic 17. University of Paderborn, 04/08/2004.

Capítulo 5: Segurança em IPv6

Este capítulo tem como objetivo explorar questões de segurança relacionadas ao protocolo IPv6. Os pontos aqui cobertos serão explorados, sempre que possível, em três aspectos:

- motivo ou causa da falha de segurança
- ataque realizado que explora esta falha
- como defender a rede deste tipo de ataque, que pode ser através de correção da falha ou utilização de ferramentas específicas

Apesar do protocolo IPv6 ser da camada 3, um ponto relevante para segurança é que sua implantação pode levar a falhas em outras camadas. O endereço IPv6 possui 128 bits enquanto o IPv4 possui 32 bits, por isso o IPv4 podia ser armazenado como um inteiro sem sinal de 32 bits, mas isto é insuficiente para tratar o IPv6 que necessitaria no mínimo 4 palavras de 32 bits. Para armazenar um IPv4 como caracteres são necessários de 4 a 15 caracteres, já no IPv6 o tamanho pode variar de 3 a 39 caracteres, por exemplo, ::1, 2001:: ou 2001:0db8:1406:b0ca:2012:0704:baba:bebe. Estas mudanças de tamanho podem gerar problemas na camada de aplicação ou mesmo em software ou hardware de outras camadas cujos campos de armazenamento de endereços IP possuem tamanho exato para o IPv4.

Um exemplo desta falha na camada de aplicação ocorre em um banco de dados que, por questões de segurança, faz log de conexões recebidas e cujo o campo endereço de origem possui o tamanho exato para armazenar endereços IPv4. Ao receber uma conexão IPv6 a aplicação irá tentar salvar as informações da conexão no banco de dados, mas o valor a ser escrito é maior que o tamanho máximo do campo, por isto o dado não é salvo. Isto significa que caso um atacante utilize IPv6 para atacar este servidor, os logs com o IP deste ataque não existirão e não será possível recuperar as informações relativas ao atacante.

Entretanto, antes de tratar especificamente de falhas, ataques e defesas na camada 3, é preciso abordar algumas lendas com relação à segurança no contexto do IPv6:

Lenda 1 - “IPv6 é mais seguro que IPv4” ou “IPv4 é mais seguro que IPv6”: estes mitos em geral são usados para se argumentar em favor de uma versão ou de outra do protocolo e usam-se os mais diversos argumentos na tentativa de defender um dos dois lados. Podem acontecer cenários em que um protocolo possua uma falha que a outra versão não possui, mas estes cenários são geralmente bastante particulares. Na prática as duas versões possuem o mesmo nível de segurança e falhas similares. O IPv6 pode ter corrigido alguns problemas conhecidos do IPv4 mas, por ter menos utilização e tempo de debug, pode possuir novas falhas que poderão ser exploradas.

Lenda 2 - “IPsec é mandatório no IPv6, por isso, ele é mais seguro que o IPv4”: a especificação do IPv6 diz que a **inclusão** do IPsec é mandatória em toda implementação do protocolo. Isto gerou o mito que a **utilização** do IPsec é mandatória, o que não é verdade. Para utilizar o IPsec é necessário configurá-lo explicitamente. Discussões recentes sobre a especificação do IPv6 estão tendendo para que a inclusão do IPsec passe a ser opcional como era no IPv4, fato que também ajuda a desmentir este mito. Esta mudança tem foco principalmente em dispositivos portáteis com processamento e memórias limitados, para que estes possam utilizar IPv6 sem desrespeitar a especificação oficial.

Lenda 3 - “Se o IPv6 não for implementado na minha rede, posso ignorá-lo”: seguir o raciocínio de que se você não possui um equipamento, não precisa se preocupar com ele, pode gerar sérios problemas para a sua rede. Mesmo que a decisão da sua empresa seja esperar um pouco mais para implementar IPv6 é necessário preocupar-se com segurança IPv6 desde de já pelos seguintes motivos:

- Os sistemas operacionais atuais possuem suporte nativo a IPv6, muitos vêm com esse suporte ativo por padrão e alguns possuem preferência pela utilização de IPv6.
- Usuários com pouco conhecimento técnico conseguem configurar túneis automáticos de IPv6 em IPv4, passando este tráfego por sua rede segura sem ser analisado.

Estes pontos mostram que o IPv6 pode ser usado mesmo que não haja implementação oficial na sua rede e existem ataques que exploram o fato do IPv6 ser ignorado. Na próxima sessão serão cobertos ataques que exploram redes que ignoram o IPv6.

Lenda 4 - “IPv6 garante comunicação fim a fim”: a especificação original do IPv6 prevê a comunicação fim a fim, assim como acontecia com a especificação original do IPv4. Entretanto mecanismos como firewalls e sistemas de detecção de intrusão controlam a comunicação fim a fim.

1. Falhas de segurança, ataques e defesas para redes IPv6

Conforme mencionado na introdução, este capítulo possui a seguinte dinâmica:

- apresenta-se uma falha de segurança e a razão dela existir
- apresenta-se o ataque que explora esta falha
- apresenta-se o modo de defender a rede deste tipo de ataque, que pode ser através de correção da falha ou utilização de ferramentas específicas

Com o intuito de facilitar a leitura, segue abaixo uma tabela resumo mencionando as falhas e seus respectivos ataques e defesas:

Falha	Ataque	Defesa
Possibilidade de falsificação do Neighbor Discovery	Negação de serviço impedindo obtenção de endereço IPv6 válido	SEND, NDPmon
Falha	Ataque	Defesa
Possibilidade de falsificação do Router Advertisement	Man-in-the-middle ou negação de serviço por configuração inválida	SEND, RA Guard, NDPmon
Conteúdo exposto e falta de autenticação	Man-in-the-middle ou falsificação de pacotes	IPsec
	Varredura de Rede	Crypto-generated Address
	Varredura de Rede	Unique Local Addresses
	Varredura de Rede	Privacy Addresses
	Varredura de Rede	Grande quantidade de endereços
Utilizar MAC na definição do IP	Rastreabilidade de Dispositivos	RFC4941 (random address) e hash por prefixo de rede
Ignorar ou mal implementar o IPv6	Novidade / Complexidade	Treinamento de equipes

Ignorar ou mal implementar o IPv6	Falta de políticas, treinamentos e ferramentas	Treinamento de equipes
Ignorar ou mal implementar o IPv6	Túnel automático	
Túnel automático	Contornar segurança IPv4	Firewall, desabilitar túneis automáticos
6to4, Teredo	Fake relay, man in the middle	Firewall, Tunnel Broker, Túnel Manual
Falta de familiaridade com o modelo fim a fim	Exploração de falhas em serviços que aceitam conexões entrantes	Firewall, IDS

2. Falhas na descoberta de vizinhos e autoconfiguração stateless (SEND e RA Guard)

O protocolo ICMPv6 é uma versão atualizada do protocolo ICMPv4 para ser utilizado em conjunto com o IPv6, sendo parte substancial de sua arquitetura. Sua implementação é obrigatória em todos os nós da rede que utilizam IPv6.

Embora esta versão possua as mesmas funcionalidades do que sua predecessora, como reportar erros no processamento de pacotes, realizar diagnósticos e enviar mensagens sobre as características da rede, ambas não são compatíveis e possuem diferenças significativas. Uma delas é que o ICMPv6 executa também a função de outros protocolos, antes separados no IPv4. Essa mudança possui como principal objetivo evitar a multiplicidade de protocolos e assim aumentar a coerência e a diminuir o tamanho das implementações.

O ICMPv6 agrupa as funcionalidades dos protocolos ARP (Address Resolution Protocol), RARP (Reverse Address Resolution Protocol) e IGMP (Internet Group Management Protocol). Mais detalhes sobre o funcionamento do ICMPv6 estão no capítulo de funcionalidades básicas desta apostila.

Com esta mudança os protocolos ARP, RARP e IGMP deixaram de existir no IPv6 e todas as suas funções foram integradas ao ICMPv6. Isto significa que ele não pode ser completamente bloqueado no firewall como podia ser feito no ICMPv4, mas isto será tratado posteriormente. O objetivo desta sessão é mostrar como os problemas de segurança do ARP possuem equivalentes no ICMPv6 e o que foi criado para que o IPv6 pudesse resolver estes problemas.

Como explicado nas funcionalidades básicas o ICMPv6 é usado para fazer a descoberta de vizinhança através do Neighbor Solicitation e do Neighbor Advertisement. Quando um novo dispositivo entra na rede ele envia um Neighbor Solicitation para saber se o IPv6 obtido na autoconfiguração ou se um novo IPv6 atribuído manualmente já está sendo utilizado na rede. Se este IPv6 já estiver em utilização o dispositivo que o está utilizando deve mandar uma mensagem de Neighbor Advertisement e o dispositivo originador fica impedido de utilizar este IPv6.

Este mecanismo de validar o IPv6 a ser utilizado permite um ataque de negação de serviço se o dispositivo atacante estiver na mesma rede. O ataque consiste em enviar uma resposta de Neighbor Advertisement para todos os pacotes de Neighbor Solicitation recebidos, fazendo com que os novos dispositivos fiquem impedidos de utilizar a rede pois não irão conseguir validar os IPv6 que desejam utilizar.

Outros ataques ao ICMPv6 utilizam o pacote de Router Advertisement que é gerado pelos roteadores da rede com o objetivo de anunciar rotas e eventualmente outros parâmetros de rede. Um dos parâmetros do pacote de Router Advertisement é a validade das informações contidas no mesmo. Este parâmetro de validade pode ser utilizado para gerar um ataque de negação de serviço que consiste em clonar um pacote de Router Advertisement válido recebido, mudar o parâmetro de validade para zero e reenviar o pacote para a rede. Isto fará com que todos os dispositivos da rede esqueçam as informações anunciadas pelo roteador, pois elas não seriam mais válidas e consequentemente fiquem sem rota para envio de pacotes para fora da rede.

Outro ataque que utiliza o Router Advertisement consiste em um dispositivo que não é roteador enviar pacotes de Router Advertisement. Este envio pode ser feito com dois objetivos:

- Anunciar-se como um roteador de alta prioridade fazendo com que todos os pacotes de rede sejam enviados para o dispositivo atacante, permitindo a este dispositivo agir como man-in-the-middle analisando o tráfego da rede antes de encaminhar para um roteador válido.
- Anunciar um endereço inválido como um roteador de alta prioridade, isto fará com que os pacotes sejam enviados para um endereço não existente e consequentemente nunca cheguem ao destino, gerando um “buraco negro” na rede.

Podemos notar que estes ataques consistem em falsificação de pacotes, algo que já existia no IPv4 e que a especificação do IPv6 criou um mecanismo opcional para resolver este problema, algo sem solução no IPv4. O mecanismo criado para o IPv6 é o SEND (Secure Neighbor Discovery) que utiliza autenticação dos dispositivos para evitar que pacotes falsificados e cuja especificação está na **RFC3971**.

O funcionamento do SEND depende dos seguintes componentes:

- Caminho de Certificação responsável por garantir a autoridade dos roteadores. Os dispositivos devem possuir este Caminho de Certificação para garantir a autenticidade do roteador antes de aceitar um roteador como roteador padrão
- CGA (Cryptographically Generated Addresses), isto é, Endereços Criptograficamente Gerados. Eles são usados para garantir que o originador de uma mensagem de Neighbor Discovery é o dono do endereço contido na mensagem. Existe uma opção de utilizar endereços não CGA, mas os detalhes ainda dependem de especificação futura. Também é necessário que todos os dispositivos gerem um par de chaves pública-privada antes de solicitar a posse de um endereço CGA. O campo CGA é utilizado para o envio da chave pública dos dispositivos
- Adição da opção de Assinatura RSA ao protocolo de Neighbor Discovery que garante a integridade da mensagem e autentica o originador da mensagem. É mandatório utilizar RSA por questões de compatibilidade
- Adição dos campos Timestamp e Nonce. O objetivo é evitar ataques do tipo replay, onde a máquina atacante reenvia pacotes válidos. O Timestamp é utilizado para proteção quando uma conexão ainda não está estabelecida e o Nonce é utilizado para mensagens pareadas do tipo Solicitation-Advertisement

A implementação do SEND pode ser feita através de um repositório central global ou local ou de um modelo mais descentralizado, similar ao que já é feito nos servidores DNS na Internet hoje.

O modelo centralizado assume a existência de uma entidade global que valida e autoriza os roteadores e para a utilização do SEND bastaria configurar em todos os dispositivos com a chave da entidade global. Esta entidade global poderia estar sob controle da IANA ou de maneira cooperativa entre os RIRs (Registros Regionais), entretanto não existe tal entidade atualmente.

No modelo descentralizado os dispositivos devem ser configurados com uma coleção de chaves públicas confiáveis. Esta coleção pode ser gerada dentro da própria organização, pelo provedor de Internet ou por uma entidade terceirizada. No caso deste modelo descentralizado pode ser necessário que um dispositivo comece a operar sem o SEND se um novo roteador na rede não puder trabalhar com SEND, mas pelo menos o SEND é utilizado na comunicação com os roteadores que estão na coleção de chaves.

Apesar de ser muito poderoso, há poucas implementações funcionais do SEND, o que impede que seja utilizado, na prática, atualmente.

Uma solução mais simples que o SEND, mas sem o objetivo de substitui-lo, foi desenvolvida para tratar somente do problema de falsificação do pacote Router Advertisement e não é capaz de resolver o ataque ao Neighbor Solicitation, esta solução é o RA-Guard ([RFC6105](#)).

O RA-Guard é aplicável a redes IPv6 conectadas através de switchs de camada 2 capazes de trabalhar com este protocolo. O RA-Guard não pode ser utilizado em meios compartilhados, em conexões diretas ou com switchs que não implementam o RA-Guard.

O funcionamento do RA-Guard consiste em, sabendo as portas que os roteadores estão conectados, somente permitir pacotes de Router Advertisement vindos destas portas. Os pacotes de Router Advertisement vindos de outras portas são descartados pelo switch, isto impede que equipamentos não roteadores se anunciem na rede ou falsifiquem anúncios.

Abaixo uma topologia exemplo onde o RA-Guard é utilizado:

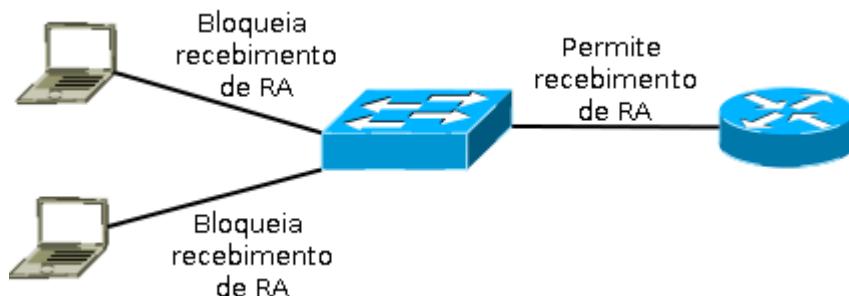


Figura 1: Funcionamento do RA-Guard

O RA-Guard é uma solução mais simples de ser configurada que o SEND pois não depende de criptografia ou autenticação. Entretanto depende do suporte dos equipamentos de camada 2 para que possa ser utilizada.

De forma geral o RA-Guard funciona como um filtro stateless bloqueando todos os pacotes ICMP tipo 134. Alguns ataques tentam furar esse bloqueio gerando pacotes fragmentados, em que o segundo fragmento é que contém o cabeçalho ICMP. Para evitar esse tipo de ataque é recomendado bloquear todos os fragmentos enviados para o endereço multicast all nodes: ff02::1.

Uma terceira possibilidade de proteção é o uso do conceito de Host Isolation, em que para cada host é utilizada uma VLAN privada, de forma que o mesmo pode apenas trocar pacotes com o roteador oficial, e não consegue comunicar-se diretamente com outros computadores na rede.

3. IPsec

Quando o protocolo IPv4 foi concebido definiu-se que os dados enviados em um determinado pacote IP não receberiam, nesta camada, qualquer tipo de ofuscamento ou criptografia e, caso esta proteção fosse necessária, caberia à camada de aplicação esta responsabilidade. Outro ponto, também não previsto na concepção do protocolo IP, é a autenticidade do pacote, por exemplo, o endereço IP de origem contido no pacote pode ser alterado ou falsificado e o dispositivo destino não terá como validar sua autenticidade.

IPsec é uma suite de protocolos, uma extensão do protocolo IP, que visa prover serviços de segurança como autenticação, integridade e confidencialidade. Os serviços são provados na camada IP e oferecem, por conseguinte, proteção às camadas superiores. A arquitetura do IPsec foi originalmente especificada na **RFC2401** em 1998 e posteriormente atualizada pela **RFC4301** em 2005.

Ele foi desenvolvido originalmente para o IPv6, mas adaptado para funcionar também com o IPv4. É pouco utilizado, contudo, com o IPv4, porque não funciona em conjunto com o NAT, este amplamente difundido em redes desse tipo.

O IPsec possui dois modos de operação, o Modo Túnel e o Modo Transporte. O IPsec possui dois protocolos AH (*Authentication Header* - Cabeçalho de Autenticação) e ESP (*Encapsulated Security Payload* - Dados Encapsulados com Segurança).

O Modo Transporte tem o objetivo de realizar IPsec somente entre dois pontos. A configuração do IPsec precisa ser feita em cada um dos dois dispositivos, e para cada nova comunicação IPsec a ser criada um novo par de configurações deve ser realizado. Observe que, apesar desta comunicação ser feita entre dois pontos, ela pode passar por outros dispositivos de rede, como switchs ou roteadores, até chegar à máquina de destino.



Figura 2: Topologia de Modo Transporte entre dois roteadores

O Modo Túnel tem o objetivo de utilizar IPsec para todo o tráfego que irá sair da rede local e passar por locais onde pode ser alterado ou falsificado. A diferença prática em relação ao Modo Transporte é que ao invés de configurar todos os dispositivos para utilizar IPsec, esta configuração é feita somente nos roteadores de borda da rede. Os roteadores encapsulam o pacote original, que é desencapsulado ao chegar ao roteador de borda do destino e encaminhado para a máquina de destino. Assim, todo o tráfego inter-redes é protegido.



Figura 3: Topologia de Modo Túnel entre as redes 2001:db8:baba:: e 2001:db8:bebe::

A formação dos pacotes no modo Transporte e Túnel é demonstrada nas figuras abaixo:

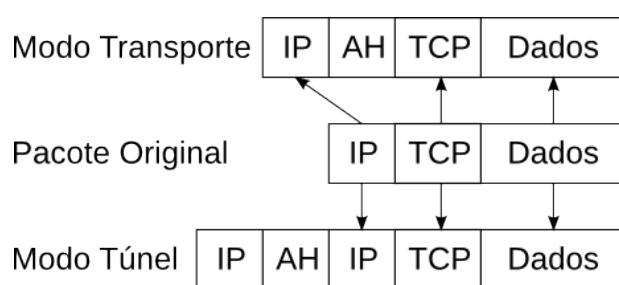


Figura 4a: Somente Autenticação

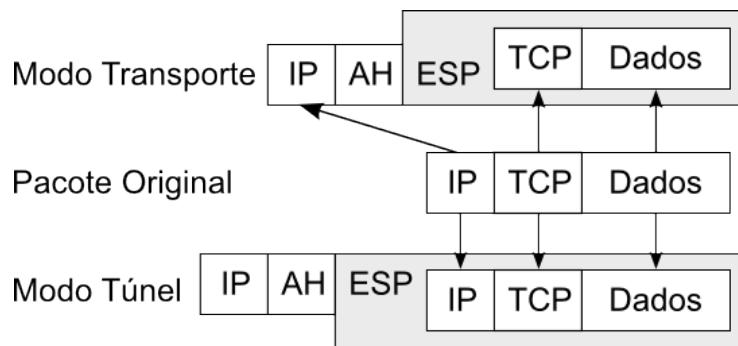


Figura 4b: Autenticação separada da criptografia

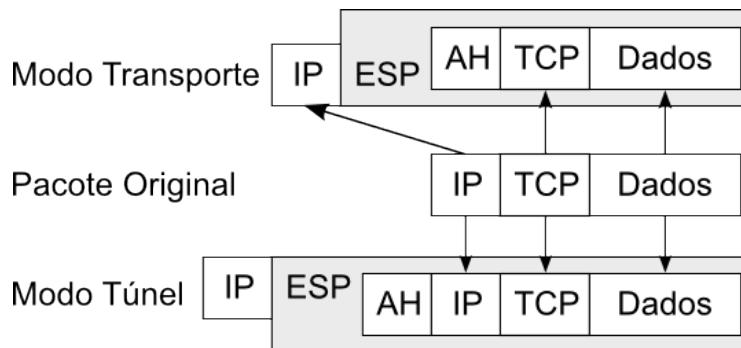


Figura 4c: Autenticação inclusa na critografia

O *Authentication Header* tem a função de proteger a integridade do pacote enviado. Isto é, se qualquer conteúdo do pacote original for alterado, por exemplo endereço de origem, dados ou outro campo, o receptor será capaz de identificar que o pacote foi alterado e por isto descartar o pacote. Se a comunicação entre dois dispositivos estiver configurada para utilizar o AH e o pacote recebido não possuir este cabeçalho a mensagem será tratada como sendo falsa e será descartada. Para fazer esta autenticação é calculado o valor do campo HMAC (*Hash Message Authentication Code* - Código de Autenticação da Mensagem calculado via função de Hash) através de uma senha secreta, dos dados contidos no pacote e de todas as partes que devem chegar inalteradas ao destino, por exemplo, os endereços IP. Campos que sofrem alteração durante o envio, por exemplo, TTL (*Time to Live*) não são utilizados para o cálculo do HMAC.

Abaixo figura representativa do cabeçalho AH:

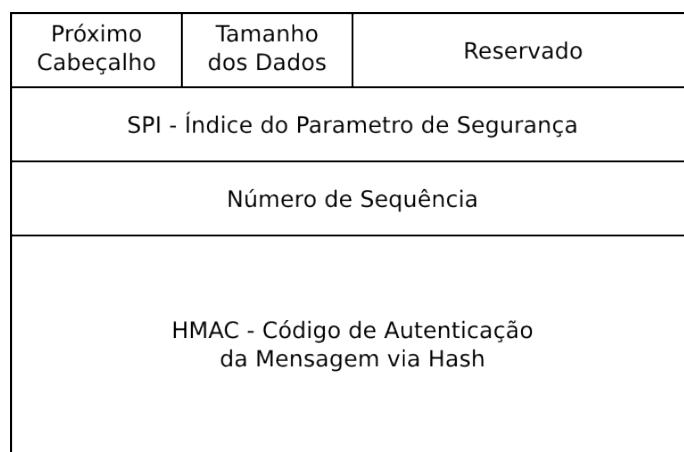


Figura 5: Cabeçalho AH

Os campos podem ser descritos da seguinte maneira:

- Próximo cabeçalho (1 byte): o cabeçalho AH é colocado entre outros cabeçalhos, assim este com indica o valor relativo ao cabeçalho que virá na sequência

- Tamanho dos dados (1 byte): tamanho da parte de dados
- Reservado (2 bytes): reservado para uso futuro
- SPI (*Security Parameter Index*) (4 bytes): código de identificação para que o destino sabia qual a chave deve ser usada para autenticação
- Número de sequência (*Sequence Number*) (4 bytes): utilizado para evitar ataques que reenviam pacotes que já foram recebidos
- HMAC (12 bytes): código de autenticação da mensagem via hash

O *Encapsulated Security Payload* adiciona confidencialidade através de criptografia dos dados a serem enviados. Note que ao utilizar o ESP os campos do AH são inclusos gerando um único cabeçalho que mantém as características de integridade e autenticação do AH adicionando a confidencialidade.

Abaixo figura representativa do cabeçalho ESP:

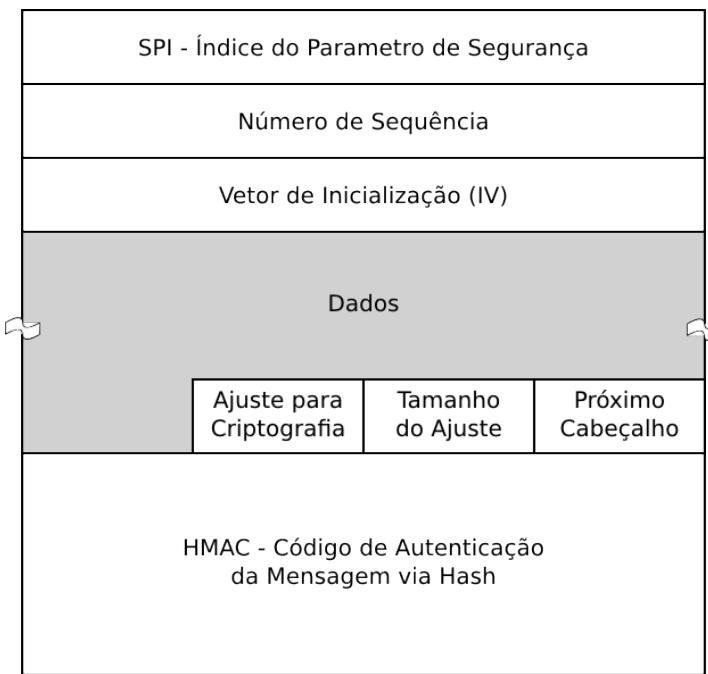


Figura 6: Cabeçalho ESP

Os campos podem ser descritos da seguinte maneira (os campos presentes no AH possuem mesmas funções explicadas anteriormente):

- Vetor de Inicialização (*Initialization Vector - IV*) (4 bytes) - usado para evitar que criptografias simétricas fiquem expostas a ataques que fazem análise de frequência de caracteres
- Dados (tamanho variável): dados protegidos por criptografia
- Ajuste para Criptografia (*Padding*): como a criptografia é feita através da cifra de blocos pode ser necessário adicionar alguns bytes para que o tamanho total dos dados a serem criados seja múltiplo do tamanho do bloco de cifra
- Tamanho do Ajuste (*Padding Length*) (1 bytes): utilizado para evitar ataques que reenviam pacotes que já foram recebidos

Como mencionado anteriormente o IPsec foi desenvolvido para ser utilizado junto ao IPv6 cuja a especificação original não considera a utilização de NAT e como o NAT altera o endereço de origem do pacote antes de encaminha-lo, isto invalida o pacote com IPsec. Para que o IPsec funcione através

de NAT algumas adaptações são necessárias, por exemplo, pode ser utilizado NAT que ao invés de alterar o endereço de origem no próprio pacote encapsula o pacote original que será desencapsulado no destino.

Um ponto fundamental para o funcionamento do IPsec são as chaves a serem usadas para autenticação, integridade e criptografia. É necessário que os dois lados saibam as chaves que devem ser usadas. Um ponto recorrente quando se fala de criptografia é como trocar as chaves por um meio que ainda não está seguro. As ideias básicas de utilizar outro meio como telefone ou email criptografado são válidas, mas necessitam de intervenção humana. O IPsec sugere a utilização do protocolo IKE que resolve a maior parte dos possíveis ataques, mas possui algumas vulnerabilidades. O protocolo IKE pode trabalhar de dois modos:

- chaves pré-compartilhadas
- certificados X.509

O protocolo IKE trabalha em duas fases:

- Fase 1: a autenticidade dos dispositivos é verificada, através de uma série de mensagens trocadas, e uma chave ISAKMP SA (*Internet Security Association Key Management Security Association*) é gerada
- Fase 2: a partir da ISAKMP SA as chaves para o AH e ESP para esta comunicação são geradas e o IPsec começa a ser utilizado

O funcionamento com chaves pré-compartilhadas é simples, cada dispositivo que utilizará IPsec deve ter uma cópia do arquivo de chaves pré-compartilhadas com uma única chave utilizada para identificar este dispositivo. Baseada nestas chaves a autenticidade do outro lado é verificada e a ISAKMP SA é gerada.

O funcionamento com certificados X.509 considera que cada dispositivo deve ter um certificado que o identifica unicamente e este certificado X.509 deve ter sua validade garantida e verificável por uma entidade certificadora confiável e conhecida. O primeiro passo do processo é o envio dos certificados e a validação dos mesmos junto a entidade certificadora. Com a validação dos certificados o processo de autenticação e geração da ISAKMP SA é realizado. Esta troca de chaves já é realizada por meio de software como, por exemplo, o racoon que está disponível para Linux e BSD.

Uma última consideração com relação ao IPsec é que apesar de não ter sido projetado como técnica de transição de IPv4 para IPv6 ele pode ser utilizado com esta finalidade, tratando todo o pacote IPv6 como dado e encapsulando este em um pacote ESP IPv4. IPsec também pode ser utilizado para encapsular um pacote IPv4 em IPv6.

4. Proteções contra varreduras de rede

Um ataque muito comum em redes IPv4 é a varredura de redes para determinar quais são os IPs válidos na rede, para posteriormente atacar os dispositivos. Como as redes IPv4 possuem uma pequena quantidade de endereços possíveis é fácil verificar se um IPv4 está em uso ou não. Se todos os endereços IPv4 fossem verificados sequencialmente a uma taxa de 1000 endereços por segundo seriam necessários 50 dias para varrer os 4.294.967.296 endereços IPv4.

A primeira forma de defesa do IPv6 é a grande quantidade de endereços possíveis. Se considerarmos somente os 64 bits menos significativos do endereço, utilizados na autoconfiguração, uma varredura destes endereços com a mesma taxa de 1000 endereços por segundo seriam necessários 257.698.037.760 dias ou 705 milhões de anos. Se considerarmos todos os 128 bits disponíveis do endereçamento seriam necessários $10,8 \times 10^{30}$ (10,8 noniliões) de anos.

Entretanto, não é possível afirmar que esta quantidade de endereços é suficiente para proteger redes de ataques de varredura. Pesquisas nas redes atuais mostram que apesar da grande quantidade de

endereços possíveis alguns padrões de endereçamento existem. Existindo um padrão, passa a ser possível fazer uma varredura desde que se utilize alguma inteligência, buscando por endereços dentro dos padrões e descobrindo dispositivos em tempo suficiente para a realização de ataques.

De acordo com pesquisas os principais padrões de endereços IPv6 são:

- SLAAC (Baseado no endereço MAC)
- IPv4-based (por exemplo, 2001:db8::192.168.10.1)
- “Low byte” (por exemplo, 2001:db8::1, 2001:db8::2, 2001:db8::3, etc.)
- Privacy Addresses (Interface de identificação aleatória)
- “Wordy” (por exemplo, 2001:db8::cafe:faca, 2001:db8::baba:bebe)
- Endereços baseados em técnicas de transição (por exemplo, 6to4, Teredo etc.)

As porcentagens de participação dos tipos de endereço é diferente nos dispositivos e nos roteadores e de acordo com a pesquisa de Malone, D. (2008. Observations of IPv6 Addresses. Passive and Active Measurement Conference (PAM 2008, LNCS 4979), 29–30 April 2008) as porcentagens são:

Dispositivos		Roteadores	
Tipo de endereço	Porcentagem	Tipo de endereço	Porcentagem
SLAAC	50%	Low-byte	70%
IPv4-based	20%	IPv4-based	5%
Teredo	10%	SLAAC	1%
Low-byte	8%	Wordy	<1%
Privacy	6%	Privacy	<1%
Wordy	<1%	Teredo	<1%
Outros	<1%	Outros	<1%

Figura 7: Porcentagem dos tipos de endereço

Com base nestas informações pode-se notar que apesar da grande quantidade de endereços possíveis a maioria utiliza padrões bem conhecidos o que facilita o desenvolvimento de programas de varredura inteligentes.

Existem algumas opções para ocultar o endereço e que são especificadas nas **RFCs 3972, 3315, 3971, 5157 e 4941**. Em geral, o ocultamento de endereços baseia-se em endereços aleatórios ou em endereços criptografados e a utilização destes endereços é recomendada para evitar ataques de varredura.

Uma outra falha relacionada a definição de endereços reside no fato de que endereços gerados automaticamente através do endereço MAC possuirão os mesmos 64 bits menos significativos, independentemente da rede que ele esteja conectado. Sendo estes 64 bits constantes é possível criar uma espécie de super cookie, rastreando, detectando a mobilidade e identificando o dispositivo, mesmo que isto não seja desejado pelo dispositivo. Para evitar esta rastreabilidade pode-se utilizar as extensões de privacidade definidas na **RFC 4941**. A rastreabilidade de host não é uma exclusividade do protocolo IPv6, existem técnicas baseadas nas informações disponibilizadas pelos navegadores com a mesma finalidade.

5. Firewall

No IPv4 o ARP era um protocolo independente do IP, que funcionava entre as camadas 2 e 3. Normalmente o mapeamento entre os IPs e os endereços físicos não eram uma preocupação ao tratar-se de firewalls. No IPv6, como já visto, essa função é realizada por novos protocolos, que utilizam mensagens ICMPv6. Outras mudanças, como a nova autoconfiguração stateless e a não fragmentação dos pacotes por roteadores, também dependem de troca de informações via ICMPv6. Logo, há aqui uma grande mudança de paradigma: se os pacotes ICMPv6 forem arbitrariamente bloqueados, a rede simplesmente para de funcionar.

Os componentes de segurança de redes, tais como listas de controle de acesso (ACLs) de roteadores e firewalls, devem ser geridos cuidadosamente para preservar a funcionalidade ICMPv6. Quaisquer medidas de segurança em um segmento de rede devem permitir os nós IPv6 a usarem ICMPv6 para realizar as tarefas essenciais como Neighbor Discovery, Path MTU Discovery, entre outras. Se um roteador IPv6 não for capaz de receber mensagens de Router Solicitation nem de respondê-las, os nós que enviaram as mensagens podem se encontrar na situação de negação de serviço (denial of service). As mensagens de Path MTU Discovery devem poder atravessar os perímetros da rede, de modo a permitir o funcionamento apropriado de comunicações IPv6 fim a fim.

Quaisquer redes IP, independente de serem somente IPv4 ou serem pilha dupla, devem possuir a capacidade de detectar e verificar pacotes ICMPv6 e IPv6. A ausência dessa capacidade permite que nós IPv6 mal-intencionados atuem em redes que deveriam operar somente em IPv4. ICMPv6 é fundamental para a operação de redes IPv6, e mesmo nós IPv6 maliciosos dependem de ICMPv6 para atuarem. Administradores e gestores de redes devem verificar se suas ferramentas de monitoração conseguem analisar o tráfego ICMPv6 de forma similar ao que já é feito com ICMP no IPv4..

É importante estabelecer políticas de filtragem para a rede local de modo a limitar mensagens ICMPv6 que possam trafegar entre a rede local e a Internet. O tratamento de regras de firewall para o tráfego de ICMPv6 requer uma precisão maior do que o equivalente para IPv4. Determinados aspectos do IPv6, tal como o Path MTU Discovery, requerem a transmissão das mensagens ICMPv6 associadas (e.g., Packet Too Big) para o funcionamento adequado. Por outro lado, mensagens associadas a outros aspectos, como o Neighbor Discovery, não devem ser transmitidas além do link local.

Consequentemente, a opção de permitir tudo ou bloquear tudo para gerência de tráfego ICMP, a qual poderia ser aplicada em IPv4, não é apropriada para IPv6. Firewalls deveriam ser capazes de bloquear ou permitir tráfego ICMPv6 seletivamente com base no tipo e no código.

Firewalls mais sofisticados podem fazer melhor ao associar tipos específicos de mensagens ICMPv6 orientados à conexão (tais como Packet Too Big ou, em algumas situações, Destination Unreachable) aos registros de estados correspondentes para conexões existentes (fluxos TCP, UDP ou SCTP), de modo que esses firewalls stateful possam ser mais restritivos quanto aos tipos permitidos para os nós que legitimamente possam ser parte dos fluxos de tráfego permitidos.

As seguintes tabelas apresentam recomendações de firewalls para os diversos tipos de tráfego. As seções "Manutenção das Comunicações" e "Mensagens de Erros" das tabelas supõem as capacidades sofisticadas de conexão-associação mencionadas anteriormente, enquanto que firewalls menos sofisticados deveriam permitir tais mensagens sempre que a conexão correspondente possa ocorrer, baseado em outras regras (e.g., endereços de origem e destino permitidos).

As recomendações para firewall ICMPv6 da tabela são baseadas na **RFC 4890**. Tais recomendações permitem a propagação de mensagens ICMPv6 necessárias para o bom funcionamento da rede, mas descartam mensagens que possam oferecer potenciais riscos de segurança. Várias mensagens ICMPv6 devem ser utilizadas somente no contexto de link local, ao invés de fim a fim, e os filtros devem considerar os tipos de endereços nos pacotes ICMPv6 bem como os endereços específicos de origem,

endereços de destino e o tipo ICMPv6. A **RFC 4890** classifica as mensagens ICMPv6 de acordo com a finalidade de uso ser para comunicações fim a fim (tráfego para atravessar um firewall) ou comunicações locais no mesmo link (tráfego local endereçado para uma interface de um firewall). Todas as mensagens experimentais e indefinidas devem ser descartadas. ACLs devem permitir somente as mensagens ICMPv6 necessárias, baseadas nas necessidades locais específicas, enquanto que todo o restante deve ser descartado.

Recomendado não descartar		
Mensagem (Tipo)	Trânsito	Local
Mensagens de Erro:		Permite não local quando associado a conexões permitidas
Time Exceeded (3) – Código 1	✓	✓
Parameter Problem (4) – Código 0	✓	✓
IPv6 Móvel:		Permite não local para dispositivos terminais permitidos
Home Agent Address Discovery Request (144)	✓	
Home Agent Address Discovery Reply (145)	✓	
Mobile Prefix Solicitation (146)	✓	
Mobile Prefix Advertisement (147)	✓	
Obrigatório não descartar		
Mensagem (Tipo)	Trânsito	Local
Manutenção da Comunicação:		Permite não local quando associado a conexões permitidas
Destination Unreachable (1) – Todos os códigos	✓	✓
Packet Too Big (2)	✓	✓
Time Exceeded (3) – Somente código 0	✓	✓
Parameter Problem (4) – Somente códigos 1 e 2	✓	✓
Verificação de Conectividade:		Permite/Nega de acordo com a política de segurança da topologia
Echo Request (128)	✓	✓
Echo Response (129)	✓	✓
Configuração de Endereços e Seleção de Roteadores:		Permitido somente em tráfego link-local
Router Solicitation (133)		✓
Router Advertisement (134)		✓
Neighbor Solicitation (135)		✓
Neighbor Advertisement (136)		✓
Inverse Neighbor Discovery Solicitation (141)		✓
Inverse Neighbor Discovery Advertisement (142)		✓
Notificação de Recebedores de Multicast Link-Local:		Permitido somente em tráfego link-local
Listener Query (130)		✓
Listener Report (131)		✓
Listener Done (132)		✓
Listener Report v2 (143)		✓
Notificação do Caminho de Certificação SEND:		Permitido somente em tráfego link-local
Certification Path Solicitation (148)		✓
Certification Path Advertisement (149)		✓
Multicast Router Discovery:		Permitido somente em tráfego link-local
Multicast Router Advertisement (151)		✓
Multicast Router Solicitation (152)		✓
Multicast Router Termination (153)		✓

Figura 8: Tabelas de recomendações ICMPv6 com base na RFC 4890

Conforme apontado pela **RFC 5095**, a funcionalidade provida pela mensagem *Routing Header* de tipo 0 pode ser utilizada para alcançar amplificação de tráfego através de um caminho remoto com a finalidade de gerar tráfego de negação de serviço (denial of service). Desse modo, é recomendado descartar esse tipo de mensagem.

6. Técnicas de Transição de IPv4 para IPv6

O IPv6 foi concebido para inicialmente trabalhar simultaneamente com o IPv4 em pilha dupla e gradualmente substituir o IPv4, mas foram também desenvolvidas técnicas de transição auxiliares, como túneis e tradução. Elas são tratadas em detalhes no capítulo específico, mas aqui cabe analisar algumas implicações do ponto de vista de segurança.

A transição de IPv4 para IPv6 abre algumas brechas de segurança. Uma delas, já comentada anteriormente, ocorre quando se ignora a existência de IPv6 numa rede que supostamente usa apenas o IPv4. Computadores e equipamentos que suportam IPv6 nativamente podem se comunicar utilizando o protocolo, evitando a segurança implementada para IPv4.

É importante considerar também que existem técnicas, muitas vezes habilitadas por padrão nos sistemas operacionais, que criam túneis automáticos, encapsulando pacotes IPv6 em pacotes IPv4. Se a rede IPv4 não tratar disso adequadamente um atacante pode acessá-la, evitando a segurança IPv4, ou um usuário dentro da rede pode acessar conteúdo ou redes que seriam normalmente bloqueadas se o acesso fosse via IPv4.

As técnicas de transição também possuem vulnerabilidades específicas e podem ser alvo de ataques. As técnicas 6to4 e Teredo, por exemplo, dependem de servidores públicos para que o túnel que transporta IPv6 dentro de IPv4 seja estabelecido. Estes servidores públicos não possuem garantia de qualidade e de confiabilidade e podem agir de maneira maliciosa, por exemplo, analisando, armazenando e até alterando dados, fazendo assim ataques do tipo man-in-the-middle.

A **RFC 4942** explora os detalhes de segurança com relação as técnicas de transição e as questões de segurança neste ponto podem ser resumidas aos seguintes pontos:

- mesmo que sua rede não tenha IPv6, não o ignore
- se você não deseja utilizar técnicas de tunelamento automático na sua rede, elas devem ser bloqueadas no firewall
- técnicas de transição podem depender de servidores públicos não confiáveis

A tabela abaixo demonstra as regras necessárias para o bloqueio de várias técnicas de transição quando não se deseja que elas sejam utilizadas em uma determinada rede.

Técnica de Transição	Regra de filtragem
Túnel manual 6over4	IPv4 Protocol == 41
Túnel manual GRE	IPv4.Protocol == 47
Túneis automáticos 6to4	IPv4.Protocol == 41 IPv4.{src,dst} == 192.88.99.0/24
Túneis automáticos Teredo	IPv4.dst == servidores_teredo UDP.DstPort == 3544

Figura 9: Tabelas de regras de filtragem para técnicas de transição

7. Considerações Finais

IPv6 é uma ainda é uma novidade para muitas das pessoas envolvidas no desenvolvimento e administração de redes e alguns novos comportamentos ou funcionalidades do IPv6 são complexos, fazendo que este seja ignorado ou mal implantado em redes. Aliado a isto soma-se a falta de políticas, boas práticas, treinamentos e ferramentas.

Este cenário leva a brechas de segurança, algumas já conhecidas e outras que somente serão descoberta com uma utilização em mais larga escala do IPv6. A única maneira de se proteger destes problemas é a capacitar as pessoas que administram ou desenvolvem redes e programas a trabalhar com IPv6. Esta capacitação pode ser obtida através de treinamentos, criação de redes de teste e desenvolvimento de software com IPv6.

8. Referências

- NIST SP 800-119, Guidelines for the Secure Deployment of IPv6, December 2010.
<http://csrc.nist.gov/publications/PubsSPs.html>
- <http://thc.org/thc-ipv6/>
- <http://www.si6networks.com/presentations/deepsec2011/fgont-deepsec2011-hacking-ipv6-networks.pdf>
- Malone, D. 2008. Observations of IPv6 Addresses. Passive and Active Measurement Conference (PAM 2008, LNCS 4979), 29–30 April 2008.
- RFC 2401 - Security Architecture for the Internet Protocol
- RFC 3315 - Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- RFC 3971 - SEcure Neighbor Discovery (SEND)
- RFC 3972 - Cryptographically Generated Addresses (CGA)
- RFC 4301 - Security Architecture for the Internet Protocol
- RFC 4890 - Recommendations for Filtering ICMPv6 Messages in Firewalls
- RFC 4941 - Privacy Extensions for Stateless Address Autoconfiguration in IPv6
- RFC 5095 - Deprecation of Type 0 Routing Headers in IPv6
- RFC 5157 - IPv6 Implications for Network Scanning
- RFC 6105 - IPv6 Router Advertisement Guard

Capítulo 6: Técnicas de Transição para o IPv6

Este texto tem como objetivo apresentar as técnicas de transição do IPv4 para o IPv6, mostrando alguns aspectos práticos e exemplos.

A importância deste tópico vem do fato de o IPv4 e o IPv6 não serem diretamente compatíveis entre si. O IPv6 não foi projetado para ser uma extensão, ou complemento, do IPv4, mas sim um substituto que resolve o problema do esgotamento de endereços.

Embora não interoperem, ambos os protocolos podem funcionar simultaneamente nos mesmos equipamentos e com base nisto a transição foi pensada para ser feita de forma gradual. No projeto inicial do IPv6, uma vez que o protocolo estivesse pronto, sua implantação começaria a ser feita gradualmente na Internet, de forma que funcionasse simultaneamente ao IPv4. A isso chamamos de pilha dupla, ou dual stack. Quando o IPv6 estivesse implantado em todos os dispositivos, o IPv4 deixaria de ser realmente útil e poderia ser abandonado paulatinamente.

No período de implantação do IPv6 haveria necessidade de técnicas auxiliares de transição, inicialmente para interconectar ilhas IPv6 em uma Internet majoritariamente IPv4 e, depois de algum tempo, para fazer o contrário.

A transição feita desta forma seria muito simples de ser executada tecnicamente. Contudo, por diversas razões, não foi o que aconteceu. Atualmente o IPv6 ainda não está sendo amplamente utilizado na Internet e o esgotamento do IPv4 já se tornou uma realidade. Hoje existe a necessidade de se implantar o IPv6 numa Internet sempre crescente, onde os novos usuários ainda precisam de conectividade IPv4, mas não há mais endereços IPv4 livres para atendê-los. Assim, novas técnicas auxiliares foram e continuam sendo desenvolvidas para essa nova realidade.

Esse texto está organizado da seguinte forma: inicialmente apresentam-se os cenários possíveis para a coexistência de redes IPv6 e IPv4, esta apresentação é seguida por uma introdução às técnicas de coexistência e transição, classificando-as segundo suas funcionalidades. Após isso, aborda-se algumas delas em detalhes, incluindo as mais amplamente utilizadas e aquelas cujo desenvolvimento recente parece promissor. Encaixam-se aí a própria pilha dupla, os túneis ponto a ponto 6over4 e GRE, os Tunnel Brokers, o DS-Lite, o NAT64, o IVI, o 464XLAT, o 6PE, o 6VPE, o 6rd e o 4rd. São apresentadas também algumas técnicas já em desuso, como 6to4, Teredo e ISATAP, mas com as quais ainda se convive no ambiente da Internet ou outras redes, principalmente por serem usadas de forma automática por alguns Sistemas Operacionais e equipamentos, como CPEs (*Customer-premises equipment*). Por fim, são apresentados alguns mecanismos para estender a vida do IPv4, que não são exatamente técnicas de transição mas podem ser utilizados como tais, em conjunto com a implantação do IPv6, como NAT444 e A+P.

Note-se que o estudo das técnicas de transição é importante mesmo para aqueles que não pretendem fazer uso das mesmas, ou para os que administraram redes em que a implantação do IPv6 ainda não foi iniciada. Algumas dessas técnicas, como afirmado anteriormente, são utilizadas de forma automática por computadores e equipamentos de rede, permitindo o uso do IPv6 por equipamentos numa rede IPv4 e eventualmente contornando mecanismos de segurança e controle. É o caso, por exemplo, de redes com computadores Windows Vista ou Windows 7.

O primeiro módulo tratará, então, de como se pode organizar e classificar os diferentes casos onde as técnicas de transição podem ser necessárias.

1. Cenários de coexistência de IPv6 e IPv4

Na transição do IPv4 para o IPv6 é necessária a coexistência e interoperabilidade entre ambos os protocolos e para isso é necessário o uso de tecnologias auxiliares, conhecidas como técnicas de transição. A necessidade de coexistência ocorre em diferentes cenários, cada qual com características e demandas singulares. Uma técnica de transição usada isoladamente normalmente não é capaz de atender simultaneamente a todos. Assim, o primeiro passo para entender as técnicas de transição é entender os cenários existentes, as necessidades apresentadas e as dificuldades envolvidas.

A enumeração dos cenários a seguir é uma generalização e extensão da enumeração feita na [RFC 6144](#). Contudo, enquanto esta RFC trata apenas de cenários utilizados com soluções de tradução, os mesmos são aqui usados para descrever também situações onde soluções de tunelamento podem ser aplicadas.

Cenário 1: Rede IPv6 para Internet IPv4 (R6-I4)

Devido a falta de endereços IPv4 ou outras limitações técnicas ou econômicas a rede cliente possui somente IPv6, mas necessita conectar-se à Internet IPv4.

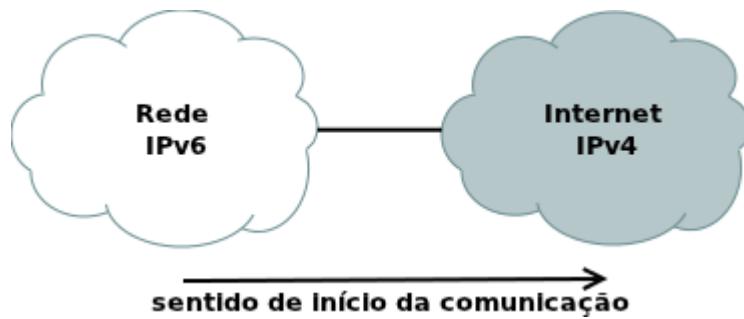


Figura 1: cenário 1

Este cenário também pode ocorrer em *greenfield networks*, expressão em inglês usada para se referir a projetos totalmente novos, aos quais não se aplicam as restrições normalmente encontradas em tecnologias já em uso. Algumas empresas têm se decidido por criar redes somente IPv6 nesse caso, por motivos de simplicidade, facilidade de gerência e outros, mas necessitam ainda acessar servidores de clientes e fornecedores que estão na Internet IPv4.

Este cenário possui uma complexidade simples e é de fácil solução, sendo suportado por tanto por técnicas stateless quanto stateful, que serão explicadas mais adiante.

Cenário 2: Internet IPv4 para Rede IPv6 (I4-R6)

Mesma rede existente no cenário 1, mas que necessita receber conexões da Internet IPv4, para o caso, por exemplo, de haver servidores IPv6 na rede, que devem atender solicitações de clientes na Internet IPv4.

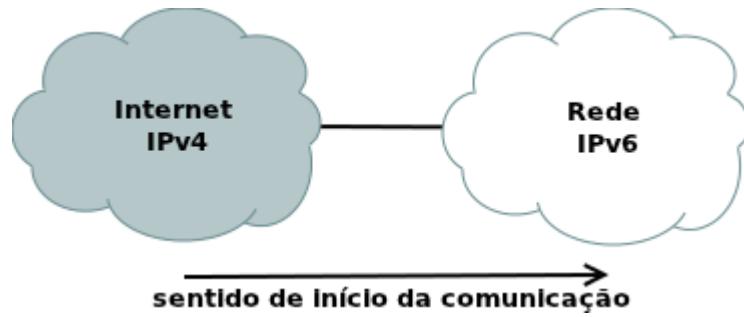


Figura 2: cenário 2

A inversão no sentido de origem da comunicação torna este cenário muito mais complexo que o cenário 1, pois normalmente não se consegue fazer um mapeamento 1:1 de todos endereços IPv6

existentes na rede para endereços IPv4 válidos. Esse caso normalmente exige soluções stateful, mas pode ser também atendido por soluções stateless, desde que suportem conexões iniciadas via IPv4 para um subconjunto dos endereços IPv6 na rede.

Cenário 3: Internet IPv6 para Rede IPv4 (I6-R4)

Este é um típico cenário onde uma rede legada, onde não é possível fazer a atualização para IPv6, necessita continuar em uso e responder requisições da Internet IPv6.

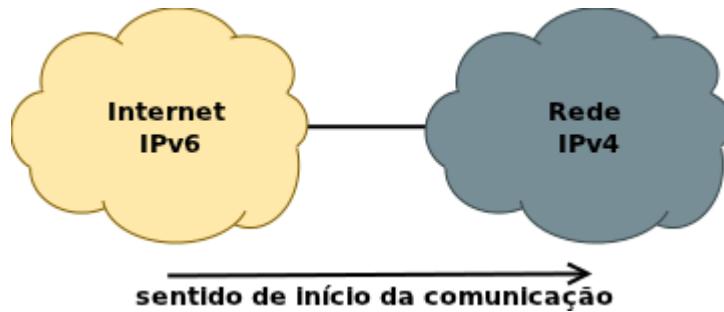


Figura 3: cenário 3

Para este cenário só cabem soluções stateful, já que a rede IPv4 deve comunicar-se com toda a Internet IPv6.

Cenário 4: Rede IPv4 para Internet IPv6 (R4-I6)

Este cenário só deve ser encontrado em estágios bem avançados da implementação do IPv6, quando a maior parte dos serviços na Internet já tiverem migrado para o novo protocolo. Técnicas de tradução na própria rede provavelmente não conseguirão solucionar esse problema.

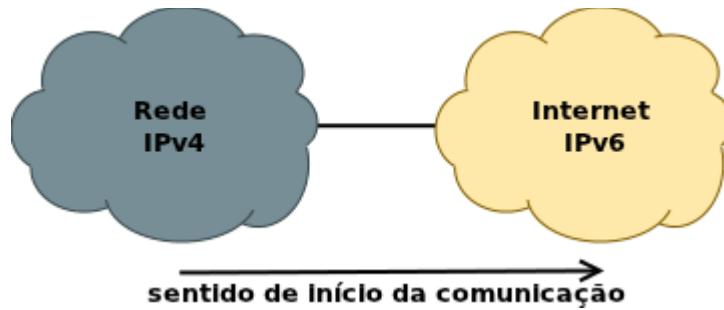


Figura 4: cenário 4

Cenário 5: Rede IPv6 para Rede IPv4 (R6-R4)

Ambas as redes deste cenário estão na mesma organização e os endereços IPv6 e IPv4 podem ser públicos e válidos na Internet ou privados e válidos somente dentro da organização. Este cenário é bastante similar ao cenário 1 e os mesmos tipos de técnicas aplicadas a ele podem ser aplicadas a este.

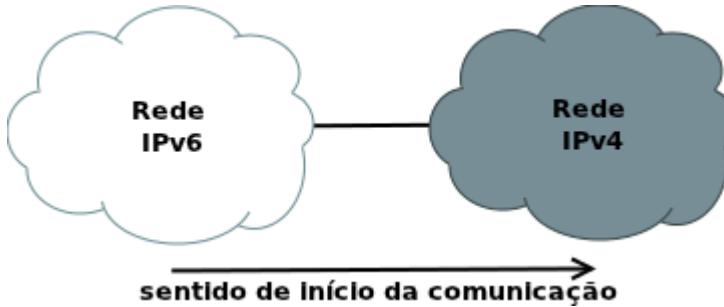


Figura 5: cenário 5

Cenário 6: Rede IPv4 para Rede IPv6 (R4-R6)

De forma análoga ao cenário anterior, essa é uma situação semelhante ao cenário 2, mas com ambas as redes dentro da mesma organização. Os endereços IPv6 e IPv4 podem ser públicos e válidos na Internet ou privados e válidos somente dentro da organização. Os mesmos tipos de técnicas aplicadas ao cenário 2 podem ser aplicadas a este.

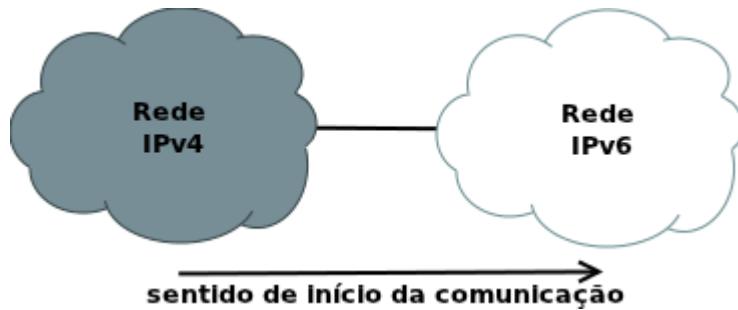


Figura 6: cenário 6

Cenário 7: Internet IPv6 para Internet IPv4 (I6-I4)

Este cenário, onde qualquer dispositivo na Internet IPv6 pode iniciar uma conexão com um dispositivo na Internet IPv4, necessita da técnica de transição perfeita, que também seria capaz de resolver todos os cenários anteriores, mas infelizmente ela não existe. A grande diferença na quantidade de endereços torna, até este momento, uma solução para este cenário tecnicamente improvável.

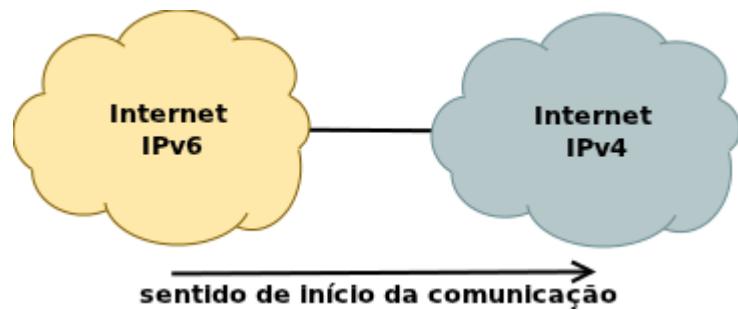


Figura 7: cenário 7

Cenário 8: Internet IPv4 para Internet IPv6 (I4-I6)

Similar ao cenário 7 e com mesma dificuldade técnica de implementação.

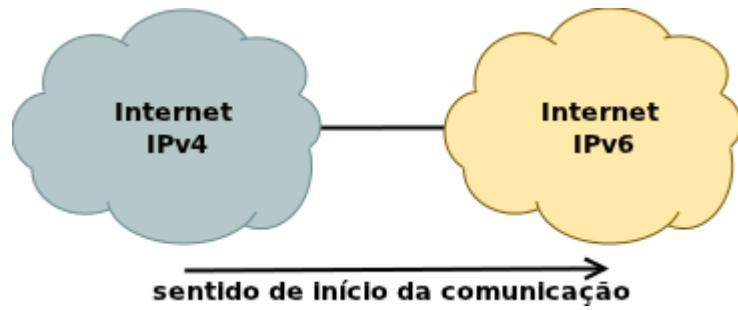


Figura 8: cenário 8

Cenário 9: Rede IPv6 para Rede IPv6 bidirecional via Internet IPv4 (R6-I4-R6)

Este cenário apresenta o caso em que a comunicação entre duas redes com IPv6 necessita ser feita através da Internet IPv4 ou de Rede IPv4. A comunicação pode ser iniciada por ambas as Redes IPv6.



Figura 9: cenário 9

Cenário 10: Rede IPv4 para Rede IPv4 bidirecional via Internet IPv6 (R4-I6-R4)

Este cenário apresenta o caso em que a comunicação entre duas redes com IPv4 necessita ser transmitida através da Internet IPv6 ou de Rede IPv6. A comunicação pode ser iniciada por ambas as Redes IPv4.



Figura 10: cenário 10

No início da explicação de cada técnica de transição, no decorrer deste texto, haverá um quadro informativo para identificar quais destes cenários são suportados pela mesma. A seguinte legenda será utilizada na tabela: Rede IPv4 (R4), Rede IPv6 (R6), Internet IPv4 (I4), Internet IPv6 (I6).

2. Classificação das técnicas de transição

Como já foi visto, desde 1983 a estrutura da Internet é baseada no IPv4. Uma troca completa e imediata do protocolo seria inviável devido ao tamanho e à proporção desta rede. Por isso, o IPv6 foi projetado para ser implantado gradualmente.

O período de transição e de coexistência dos dois protocolos exigiu o desenvolvimento de técnicas auxiliares. O primeiro problema que elas procuravam resolver era como conectar redes IPv6 a outras redes IPv6 por meio de equipamentos ou de uma Internet que só suportasse IPv4. Surgiram então diversos tipos de **túneis** IPv6 sobre IPv4 para atender tal necessidade, usando diferentes técnicas, estabelecidos manualmente ou automaticamente. Foram criadas também técnicas para permitir que redes IPv6 e IPv4 interoperassem, por meio da **tradução** dos pacotes.

Mais recentemente, o problema principal a ser resolvido pelas técnicas de transição passou a ser a implantação do IPv6 num ambiente em que o IPv4 não está mais disponível, mas ainda é necessário para os novos usuários da Internet. Foram, e continuam sendo, desenvolvidos então diversos tipos de túneis IPv4 sobre IPv6 para, aliados a técnicas de tradução, solucionar esse problema.

Pode-se, então, classificar as técnicas de transição segundo 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.

Deve-se notar que tanto os túneis quanto as técnicas de tradução podem ser **stateful** ou **stateless**. Técnicas stateful são aquelas em que é necessário manter tabelas de estado com informações sobre os endereços ou pacotes para processá-los. Nas técnicas stateless não é necessário guardar informações, cada pacote é tratado de forma independente. De forma geral técnicas stateful são mais caras: gastam mais CPU e memória, por isso não escalam bem. Sempre que possível deve-se dar preferência a técnicas **stateless**.

Há casos em que é necessária a comunicação entre IPv4 e IPv6 para apenas um, ou poucos tipos de aplicações. Ou ainda, quando é usada uma técnica de tradução e ela funciona para quase todas as aplicações, mas falha para algumas poucas, especificamente aquelas que carregam endereços IP literais no protocolo, na camada de aplicação. Para esses casos podem ser usados gateways específicos, na camada de aplicação. São chamados de Application Level Gateways, ou **ALGs**.

Uma grande dificuldade no processo de implantação do IPv6 é o desenvolvimento de uma variedade enorme de técnicas de transição, o que dificulta a escolha do que efetivamente utilizar. A **figura 11** ilustra essa variedade, nomeando diversos tipos de técnicas para túneis hoje padronizadas, ou em discussão na IETF, e organizando-as segundo sua funcionalidade e método de funcionamento. Nem todas serão abordadas neste texto.

De forma geral, os critérios que devem ser utilizados na escolha da técnica a ser utilizada, são:

- deve-se preferir técnicas que impliquem na utilização de IPv6 nativo pelos usuários finais, de forma que túneis IPv4 dentro de IPv6 devem ser preferidos em detrimento de túneis IPv6 sobre IPv4;
- deve-se preferir técnicas stateless em detrimento de técnicas statefull;
- deve-se evitar técnicas para prolongar o uso do protocolo IPv4, sem a adoção concomitante do IPv6;
- deve-se analisar a adequação da técnica à topologia da rede onde será aplicada e
- deve-se analisar a maturidade da técnica e as opções de implantação, como por exemplo suporte à mesma nos equipamentos de rede e em softwares.

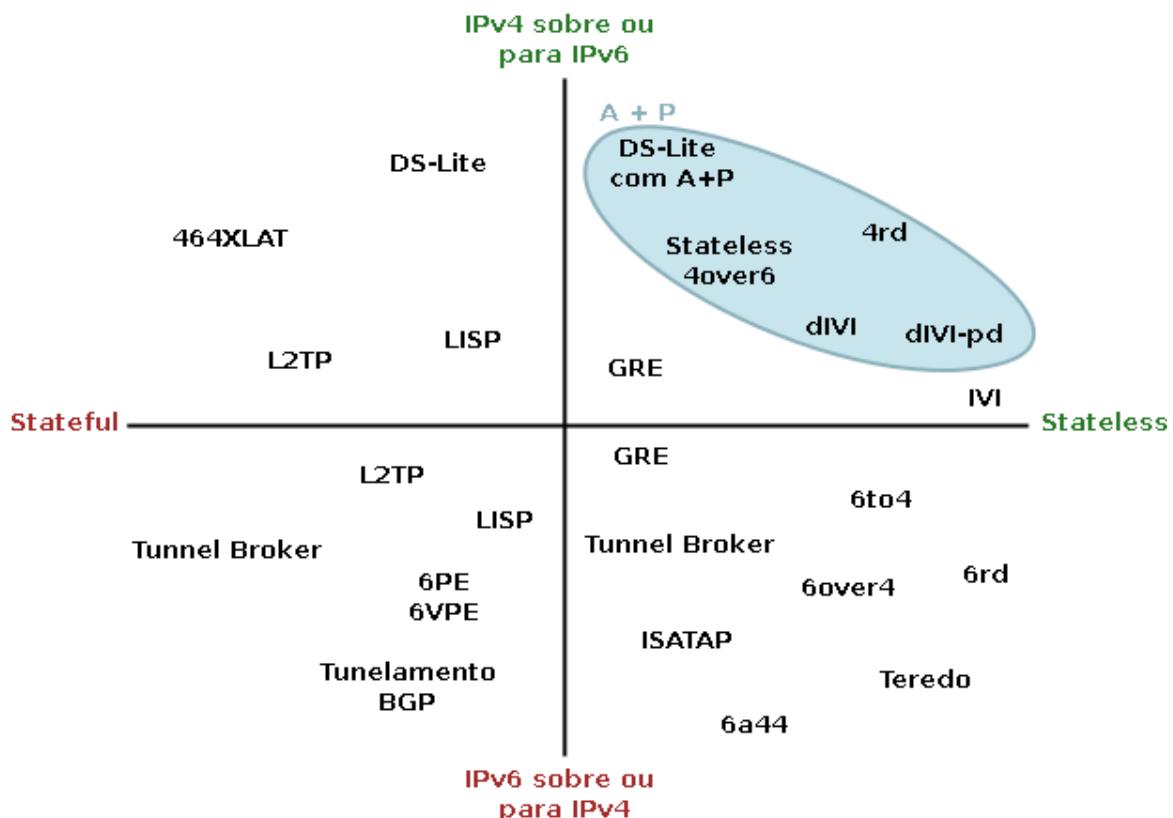


Figura 11: classificação das técnicas de transição

Como uma terceira possibilidade de classificação, pode-se dividir as técnicas conforme seus casos de uso:

- Fornecer IPv6 e IPv4 para todos os dispositivos: pilha dupla.
- Oferecer conectividade IPv6 nativa em conjunto com conectividade IPv4 com compartilhamento e preservação de endereços: DS-Lite, DS-Lite com A+P, 4rd, NAT64, IVI e 464XLAT.
- Transportar IPv6 em uma rede MPLS IPv4: 6PE e 6VPE.
- Obter conectividade IPv6, quando o provedor Internet não a oferecer: tunnel broker e túneis estáticos 6over4 ou GRE.
- Oferecer conectividade IPv6 para os usuários sobre uma rede de transporte IPv4: 6rd (normalmente usado em provedores) e ISATAP (para redes internas).
- Mecanismos para compartilhar endereços IPv4, estendendo sua vida: A+P e NAT444.

3. Pilha Dupla: IPv6 e IPv4 em todos os dispositivos

Na atual fase de implantação do IPv6, não é aconselhável ter nós com suporte apenas a esta versão do protocolo IP, visto que muitos serviços e dispositivos na Internet ainda trabalham somente com IPv4. Como citado anteriormente, manter o IPv4 já existente funcionando de forma estável e implantar o IPv6 nativamente, para que coexistam nos mesmos equipamentos, é a forma básica escolhida para a transição na Internet. Esta técnica é conhecida como pilha dupla (Dual Stack ou DS) e deve ser usada sempre que possível.

A utilização deste método permite que dispositivos e roteadores estejam equipados com pilhas para ambos os protocolos, tendo a capacidade de enviar e receber os dois tipos de pacotes, IPv4 e IPv6.

Com isso, um nó Pilha Dupla, ou nó IPv6/IPv4, 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.

Cada nó IPv6/IPv4 é configurado com ambos endereços, utilizando mecanismos IPv4 (ex. DHCP) para adquirir seu endereço IPv4 e mecanismos IPv6 (ex. configuração manual, autoconfiguração stateless e/ou DHCPv6) para adquirir seu endereço IPv6.

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

O funcionamento da pilha dupla está ilustrado na **figura 12**.

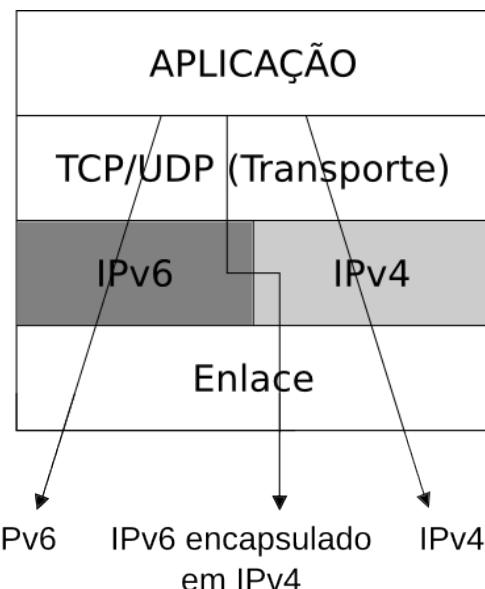


Figura 12: funcionamento da pilha dupla

Alguns aspectos referentes à infra-estrutura da rede devem ser considerados ao se implementar a técnica de pilha dupla: a estruturação do serviço de DNS e a configuração dos protocolos de roteamento e de firewalls.

Em relação ao DNS, é preciso configurar os novos endereços IPv6, usando registros do tipo AAAA (quad-A), que armazenam seus endereços. Para mais detalhes sobre o suporte do DNS ao IPv6, consulte a [RFC 3596](#). Responder os endereços IPv6 (registros AAAA) quando disponíveis para um determinado nome de domínio é o comportamento padrão do servidor DNS, mesmo que ele opere apenas com IPv4. O protocolo por meio do qual é feita a consulta não interfere na resposta. Ao receber endereços IPv6 e IPv4 como resposta a uma consulta no DNS a aplicação decide qual protocolo usar. Normalmente a preferência é pelo protocolo IPv6 e, em caso de falha, tenta-se o IPv4. Mais recentemente têm sido experimentadas técnicas que implicam em tentativas simultâneas de conexão IPv6 e IPv4 e optam pela que for mais rápida ([draft-ietf-v6ops-happy-eyeballs-07](#)). Em uma rede com pilha dupla, a configuração do roteamento IPv6 normalmente é independente da configuração do roteamento IPv4. Isto implica no fato de que, se antes de implementar-se o IPv6 a rede utilizava apenas o protocolo de roteamento interno OSPFv2 (com suporte apenas ao IPv4), será necessário migrar para um protocolo de roteamento que suporte tanto IPv6 quanto IPv4 (como ISIS por exemplo) ou forçar a execução do OSPFv3 paralelamente ao OSPFv2.

A forma como é feita a filtragem dos pacotes que trafegam na rede pode depender da plataforma que se estiver utilizando. Em um ambiente Linux, por exemplo, os filtros de pacotes são totalmente independentes uns dos outros, de modo que o iptables filtra apenas pacotes IPv4 e o ip6tables apenas IPv6, não compartilhando nenhuma configuração. No FreeBSD, as regras são aplicadas a ambos os protocolos no mesmo arquivo de configuração. Entretanto a regra pode ser aplicada simultaneamente aos dois protocolos ou a somente um. Para aplicar a somente um deles basta utilizar inet ou inet6

dependendo do protocolo à qual as regras devem ser aplicadas. De uma forma ou de outra, novas regras terão de ser configuradas no firewall ao implantar-se o IPv6.

É importante reforçar que configurações independentes para IPv4 e IPv6 são necessárias para diversos aspectos da rede, entre eles:

- Informações nos servidores DNS autoritativos;
- Protocolos de roteamento;
- Firewalls;
- Gerenciamento das redes.

Utilizar pilha dupla pode não ser possível em todas as ocasiões. Por exemplo, quando não há mais IPv4 disponíveis e o provedor precisa atender a usuários novos com IPv6 e IPv4. Para redes corporativas que já utilizam NAT isso não é um impedimento: o IPv6 nativo pode ser utilizado em conjunto com o IPv4 compartilhado. Outra situação que dificulta a implantação do IPv6 usando pilha dupla é a existência de equipamentos que não o suportam e que não podem ser facilmente substituídos. Para contornar essas situações existem diversas técnicas disponíveis, algumas das quais serão abordadas nas próximas sessões.

4. Túneis 6over4 (IPv6-over-IPv4)

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Sim	Não							

Quando a utilização de pilha dupla não é possível, uma das alternativas a ser considerada é a utilização de túneis. As técnicas de tunelamento fazem o encapsulamento de pacotes IPv6 em pacotes IPv4. Este encapsulamento é conhecido como 6in4 ou IPv6-in-IPv4 ([RFC 4213](#)). Ele consiste em colocar o pacote IPv6 dentro de um pacote IPv4, adequar os endereços de origem e destino para o IPv4 e colocar no cabeçalho o tipo 41 (29 em hexadecimal). Esse tipo de encapsulamento é conhecido por 6in4, ou como “protocolo 41”. Quando o destino receber o pacote com tipo 41 ele irá remover o cabeçalho IPv4 e tratar o pacote como IPv6. A figura 13 ilustra esse comportamento.

Também é possível, de forma análoga, encapsular pacotes IPv4 em pacotes IPv6, técnica conhecida como 4in6. Algumas das técnicas de transição estudadas mais à frente fazem isso.

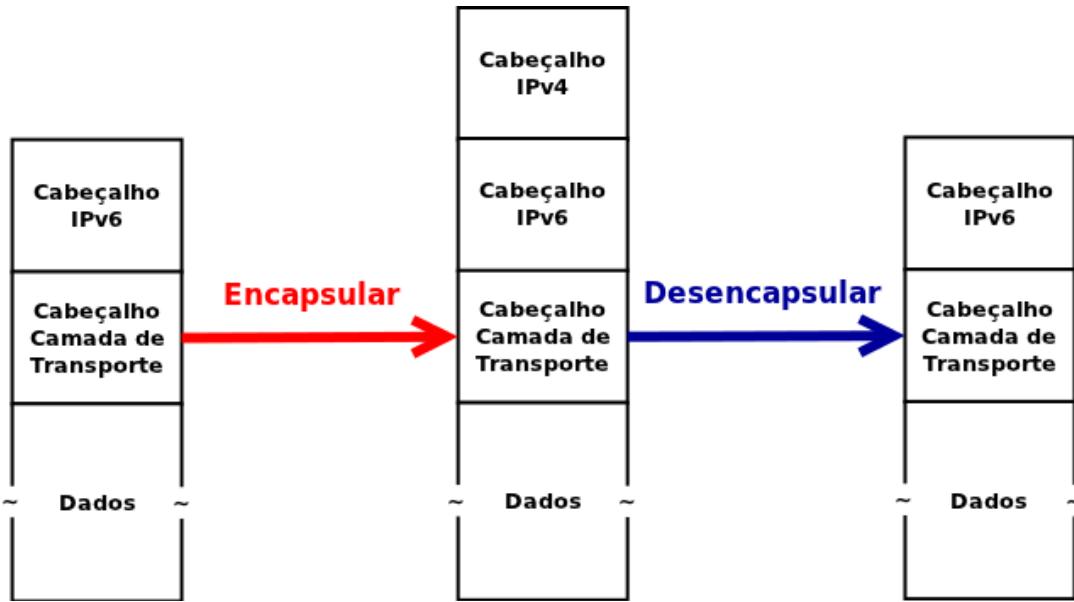


Figura 13: funcionamento 6in4

Uma das formas de utilizar-se túneis é criando-os manualmente. A técnica 6over4 ([RFC 4213](#)) utiliza um túnel manual estabelecido entre dois nós IPv4 para enviar o tráfego IPv6. Todo o tráfego IPv6 a ser enviado é encapsulado em IPv4 usando 6in4, explicado anteriormente. A configuração manual consiste em definir quais serão os IPs v4 de origem e destino que serão utilizados em cada ponta do túnel. Ao ser recebido pelo nó destino, o pacote IPv6 é desencapsulado e tratado adequadamente.

Esse tipo de túnel pode ser utilizado para contornar um equipamento ou enlace sem suporte a IPv6 numa rede, ou para criar túneis estáticos entre duas redes IPv6 através da Internet IPv4.

É importante entender a diferença entre 6over4 e 6in4. O túnel 6over4 é um túnel estabelecido manualmente que tem o objetivo de permitir conexão IPv6 entre dois nós de rede conectados por uma rede via IPv4. Ele usa o encapsulamento 6in4. Já o encapsulamento 6in4, com a utilização do tipo 41, pode ser utilizado também em outras técnicas de transição que transportam pacotes IPv6 em redes IPv4, como poderá ser observado ao longo deste texto.

Criar um túnel 6over4 é bastante fácil. A seguir serão mostrados exemplos de como fazer esta implementação no Linux e com roteadores Cisco. A topologia da implementação em Linux é:

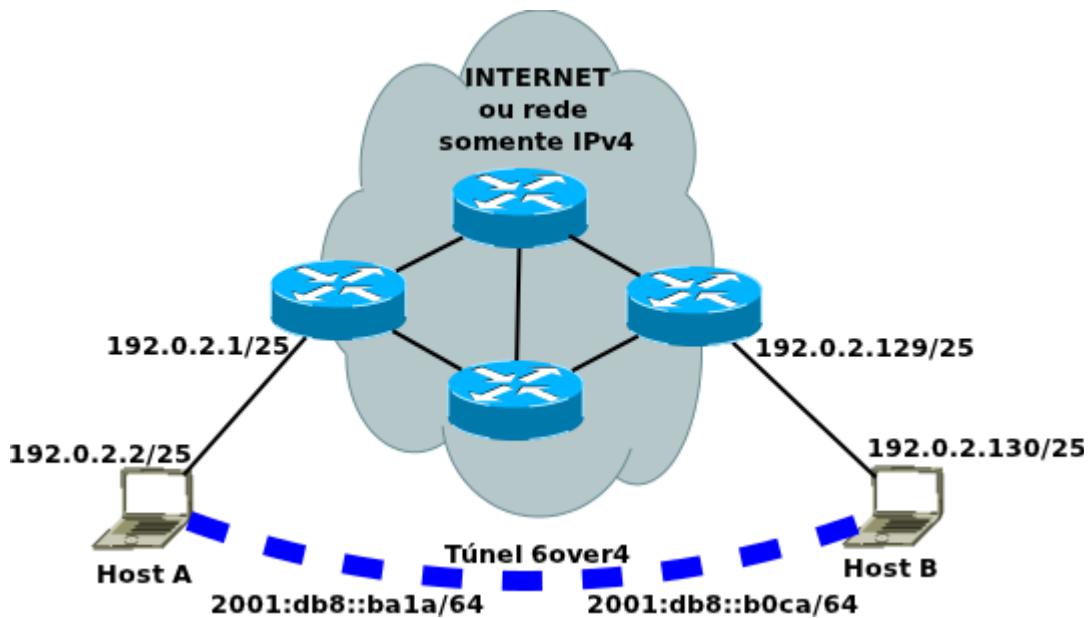


Figura 14: túnel manual 6over4 entre dois dispositivos

Os computadores *Host A* e *Host B* são computadores Linux e os roteadores simplesmente representam uma rede somente IPv4 ou a Internet IPv4. Os computadores devem ser configurados com os seguintes passos:

No *Host A*, basta digitar os comandos:

```
ip tunnel add toHostB mode sit ttl 64 remote 192.0.2.130 local 192.0.2.2
ip link set dev toHostB up
ip -6 route add 2001:db8::b0ca dev toHostB
```

De forma análoga, no *Host B*:

```
ip tunnel add toHostA mode sit ttl 64 remote 192.0.2.2 local 192.0.2.130
ip link set dev toHostA up
ip -6 route add 2001:db8::bala dev toHostA
```

Para verificar o correto funcionamento pode-se utilizar o comando ping6 antes e depois de fazer as configurações. Será possível notar que as duas máquinas passaram a comunicar-se via IPv6.

Para o exemplo de configuração em roteadores Cisco de túneis 6over4 a topologia será:

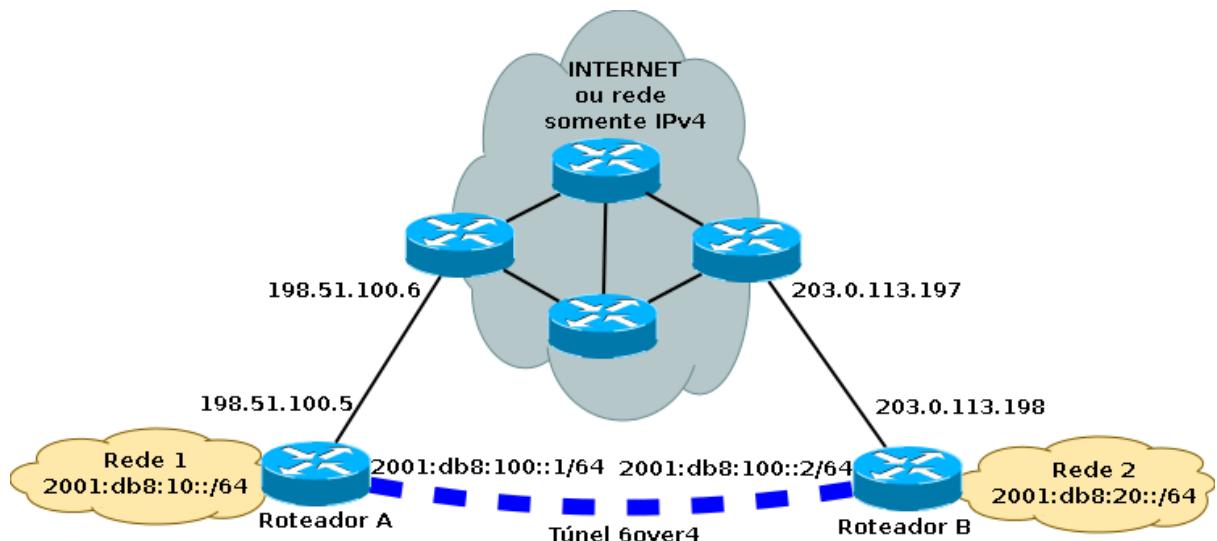


Figura 15: túnel manual 6over4 entre dois roteadores

Para a configuração do túnel somente é necessária a configuração do Roteador A e do Roteador B.

No Roteador A:

```

configure terminal
interface Tunnel10
  ipv6 address 2001:db8:100::1/64
  tunnel source 198.51.100.5
  tunnel destination 203.0.113.198
  tunnel mode ipv6ip
end

```

Ainda no Roteador A, é necessário ativar o roteamento IPv6 e criar uma rota para a rede do Roteador B, apontando para o Túnel 6over4:

```

ipv6 unicast-routing
ipv6 route 2001:db8:20::/64 2001:db8:100::2

```

De forma análoga, no Roteador B:

```

configure terminal
interface Tunnel20
  ipv6 address 2001:db8:100::2/64
  tunnel source 203.0.113.198
  tunnel destination 198.51.100.5
  tunnel mode ipv6ip
end
ipv6 unicast-routing
ipv6 route 2001:db8:10::/64 2001:db8:100::1

```

Mais uma vez, é possível testar a configuração utilizando o ping para IPv6.

5. Túneis GRE

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Sim	Não							

Outra opção de túnel estático para o transporte de IPv6 em redes IPv4 é o GRE (Generic Routing Encapsulation - **RFC 2784**). Ele é um túnel estático entre dois nós originalmente desenvolvido pela Cisco com a finalidade de encapsular vários tipos diferentes de protocolos, como por exemplo IPv6 e ISIS (a lista completa dos protocolos suportados pode ser encontrada em <http://www.iana.org/assignments/ethernet-numbers>). Este tipo de encapsulamento é suportado na maioria dos sistemas operacionais e roteadores e possibilita a criação de um link ponto a ponto. Assim como o 6over4 sua configuração é manual, de modo que pode gerar um esforço na sua manutenção e gerenciamento proporcional à quantidade de túneis.

O pacote com cabeçalho é explicado na figura a seguir:

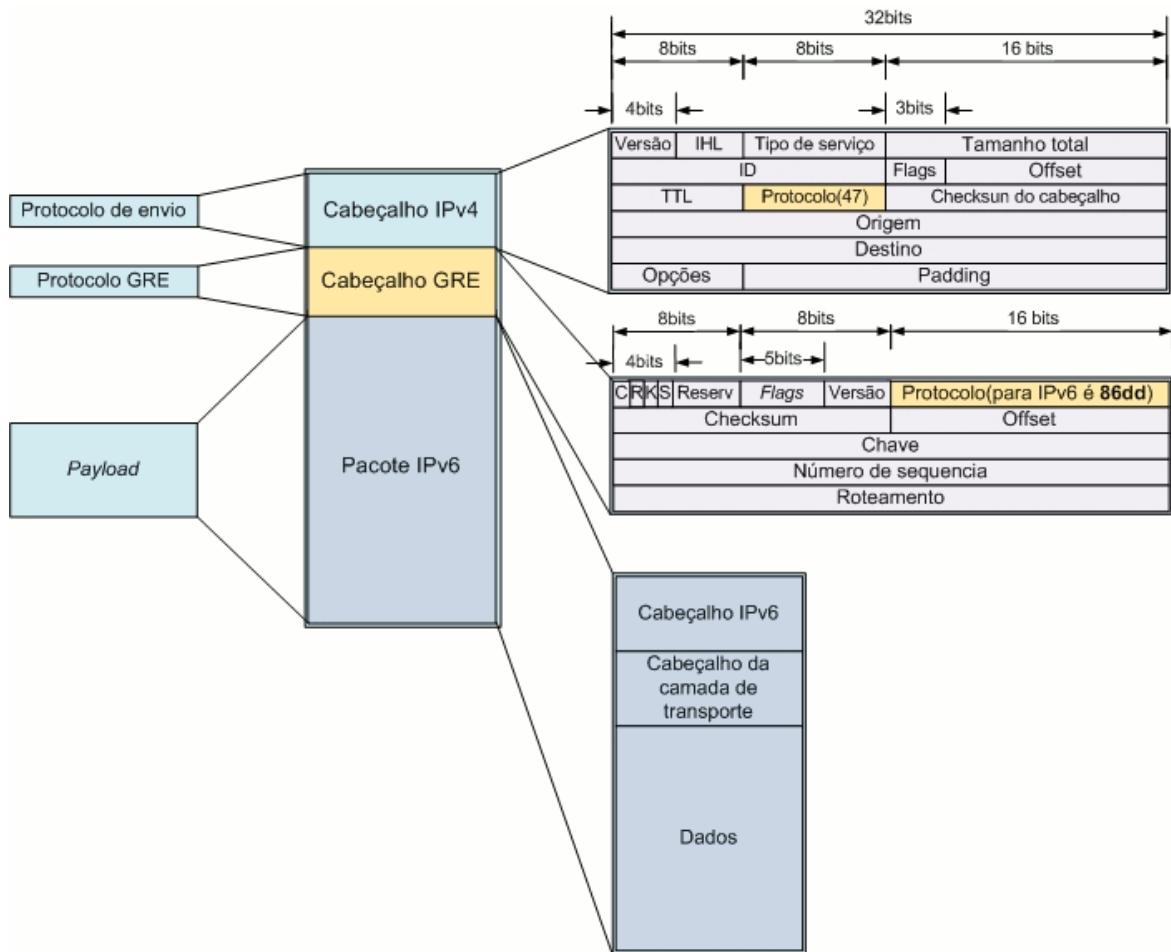


Figura 16: pacote com cabeçalho GRE

O funcionamento deste tipo de túnel é muito simples: consiste em pegar os pacotes originais, adicionar o cabeçalho GRE e o cabeçalho IPv4 e enviar ao IP de destino. Quando o pacote encapsulado chegar na outra ponta do túnel (IP de destino) remove-se dele os cabeçalhos IPv4 e GRE, restando apenas o pacote original, que é encaminhado normalmente ao destinatário.

A configuração dos túneis GRE é muito semelhante àquela feita para o 6over4. No exemplo dado para roteadores Cisco, no item 4, basta trocar:

```
tunnel mode ipv6ip  
por
```

```
tunnel mode gre
```

6. Tunnel Broker

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Sim	Não							

Descrita na [RFC 3053](#), essa técnica permite que dispositivos isolados, ou toda uma rede IPv4, obtenham conectividade IPv6 por meio do estabelecimento de um túnel com um provedor, tornando-se, na prática, dispositivos, ou uma rede, pilha dupla.

Seu funcionamento é bastante simples: primeiramente é necessário realizar um cadastro, normalmente via Web, em um provedor que ofereça esse serviço, chamado, neste contexto, de Tunnel Broker. O provedor realizará de forma automática, ou semi automática, a configuração do seu lado do túnel e

permitirá o download de instruções, ou de um software ou script de configuração, para configurar o lado do usuário. Os Tunnel Brokers normalmente oferecem blocos fixos IPv6 que variam de /64 a /48.

Dentre as opções existentes, recomenda-se:

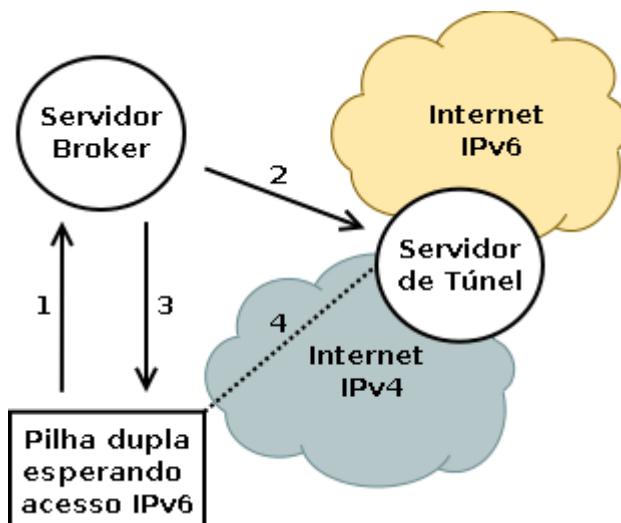
- <http://tunnelbroker.net/> - serviço oferecido pela Hurricane Electric, que provê túneis para usuários domésticos ou corporativos, inclusive com a possibilidade de se fechar sessões BGP para provimento de trânsito IPv6 via túnel.
- <http://www.sixxs.net/main/> - serviço oferecido de forma colaborativa por um grande número de organizações. Não é possível fechar sessões BGP, mas é possível obter redes fixas de tamanho /48 roteadas através do túnel. A Algar Telecom/CTBC é responsável por um dos POPs em que são configurados os túneis, no Brasil, de forma que para usuários em redes brasileiras os túneis funcionam com qualidade e velocidade próximas às de conexões nativas.

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 ([draft-massar-v6ops-ayiya-02](#)), ou TSP (Tunnel Setup Protocol), definido na [RFC 5572](#).

A utilização de Tunnel Brokers é recomendada para usuários domésticos e corporativos que querem testar o IPv6, ou começar um processo de implantação em suas redes, mas cujos provedores de acesso ainda não oferecem suporte ao protocolo. Muitos Sistemas Autônomos brasileiros têm utilizado com sucesso túneis com a Hurricane Electric para anunciar seus blocos em caráter de teste e muitas empresas e usuários domésticos têm utilizado túneis SixXS para familiarizar-se com o IPv6.

A implantação de um serviço de Tunnel Broker em um provedor Internet não é trivial, pois não há softwares abertos disponíveis para a funcionalidade de Servidor Broker.

As figuras abaixo mostram a topologia lógica e física do Tunnel Broker.



1 - Cliente pilha dupla solicita túnel (pode ser solicitada autenticação) via IPv4

2 - Broker cadastra usuário no Servidor de túnel

3 - Broker informa cliente parâmetros para criação do túnel

4 - Túnel estabelecido

Figura 17: Topologia lógica do Tunnel Broker

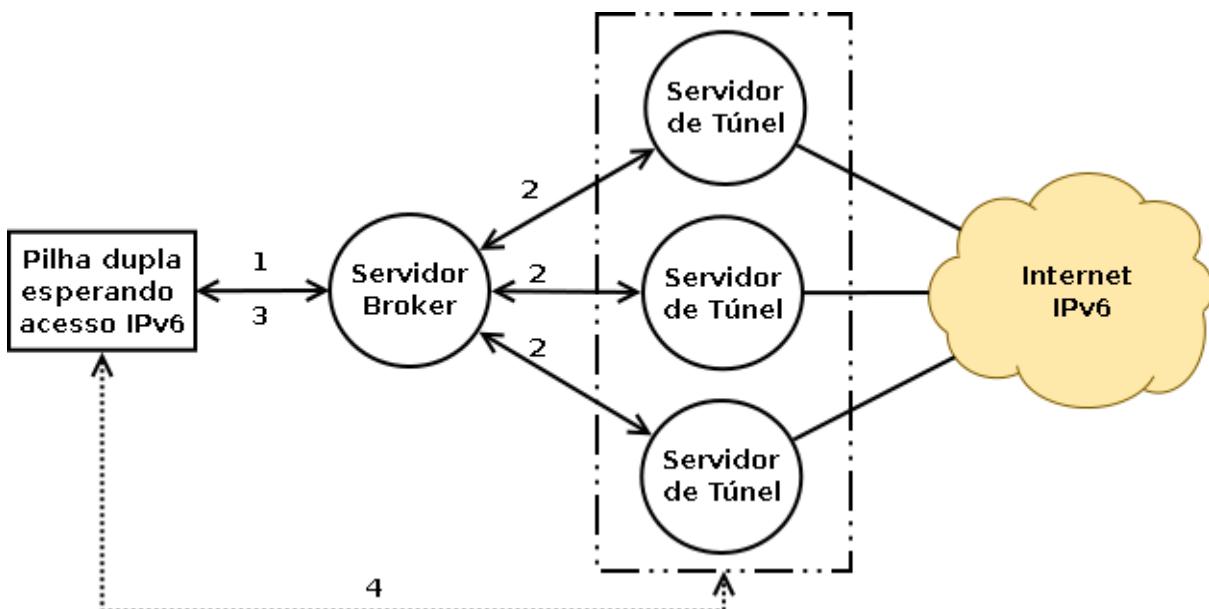


Figura 18: Topologia física do Tunnel Broker

O exemplo de implementação de Tunnel Broker será baseado no OpenWRT (openwrt.org). Ele é um firmware opensource para roteadores sem fio SOHO (small office / home office). Como provedor do túnel será utilizada a solução da Hurricane Eletric (tunnelbroker.com). Abaixo o passo a passo da instalação:

1. Criar um usuário em tunnelbroker.com e solicitar um túnel

2. Instalar os pacotes necessários no OpenWRT:

```
opkg install ip ip6tables kmod-sit kmod-iptunnel6 radvd
```

3. Criar o arquivo /etc/hotplug.d/iface/15-ipv6 com o seguinte código (ele considera que a conexão com o provedor utiliza PPP, se for outro tipo de conexão o código necessita pequenas alterações):

```
. /etc/functions.sh
NAME=ipv6
COMMAND=/usr/sbin/ip
[ "$ACTION" = "ifup" -a "$INTERFACE" = "wan" -a "$DEVICE" = "ppp0" ] && {
[ -x $COMMAND ] && {
    # setup tunnel
    logger "HE-IPv6: starting tunnel..."
    IPADDR=$(ip -4 addr show dev $DEVICE |
    awk '/inet / {print $2}' |
    cut -d/ -f1)
    username="abcdef1234567890abcdef1234567890" # MD5 of your username
    password="abcdef1234567890abcdef1234567890" # MD5 of your password
    tunnelid="69999" # global tunnel-ID
    # update tunnel to use dynamic ipv4
    wget -q -O /dev/null "http://ipv4.tunnelbroker.net/ipv4_end.php?ipv4b=$IPADDR&pass=$password&user_id=$username&tunnel_id=$tunnelid"
    SERVER_IPv4_ENDPOINT=216.66.80.30 # change this IP to your option
    CLIENT_IPv6_ENDPOINT=2001:470:1f0a:9999::2/64 # change this, too
    # setup tunnel
    ip tunnel add he-ipv6 mode sit remote $SERVER_IPv4_ENDPOINT local $IPADDR ttl
255
    ip link set he-ipv6 up
    ip addr add $CLIENT_IPv6_ENDPOINT dev he-ipv6
    ip route add ::/0 dev he-ipv6
}
}
[ "$ACTION" = "ifdown" -a "$INTERFACE" = "wan" -a "$DEVICE" = "ppp0" ] && {
[ -x $COMMAND ] && {
    # destroy tunnel
}}
```

```

    logger "HE-IPv6: destroying tunnel..."
    ip route del ::/0 dev he-ipv6
    ip tunnel del he-ipv6
}
}

# You got a routed /64

```

4. Adicionar um IP para a interface do túnel:

```

uci set network.lan.ip6addr=2001:470:1f0b:9999::1/64
uci commit

```

5. Configurar o firewall do OpenWRT para aceitar pacotes com protocolo 41 vindos da interface WAN

6. Configurar o anúncio da rede IPv6 na LAN, editando o arquivo /etc/config/radvd :

```

config interface
    option interface      'lan'
    option AdvSendAdvert   1
    option AdvManagedFlag  0
    option AdvOtherConfigFlag 0
    option ignore          0
config prefix
    option interface      'lan'
    # If not specified, a non-link-local prefix of the interface is used
    option prefix          '2001:470:1f0b:9999::/64'
    option AdvOnLink        1
    option AdvAutonomous   1
    option AdvRouterAddr   0
    option ignore          0
config rdNSS
    option interface      'lan'
    # If not specified, the link-local address of the interface is used
    option addr            '2001:470:1f0b:9999::/64'
    option ignore          1

```

Alterar o endereço “:9999” para a rota que você utilizou. Salvar o arquivo e executar os comandos para que as alterações sejam aplicadas:

```

/etc/init.d/radvd enable
/etc/init.d/radvd start

```

7. A configuração está completa. Reiniciar o roteador e testar o túnel. Pode-se executar o ping6 diretamente no roteador e funcionando corretamente executá-lo a partir de um computador na LAN:

```
ping6 ipv6.google.com
```

Em caso de dúvidas, os tutoriais da Hurricane Eletric ou do OpenWRT podem ser consultados em:

<http://www.tunnelbroker.net/forums/index.php?topic=1016.0>

<http://wiki.openwrt.org/doc/uci/network#dynamic.ipv6-in-ipv4.tunnel.he.net.only>

Outro exemplo de configuração é a utilização de Tunnel Broker no Windows. É possível utilizá-lo em diversas versões do Windows (2000, XP, 2008, Vista e 7) desde que o suporte IPv6 seja instalado nas versões que não o suportam nativamente. A configuração deve ser feita através do console usando um usuário com permissões administrativas. As configurações para estas versões do Windows são:

Explicação das variáveis usadas:

```

$ipv4a = IPv4 do servidor do túnel
$ipv4b = IPv4 do usuário do túnel
$ipv6a = rede /64 alocada ao lado do servidor do túnel
$ipv6b = rede /64 alocada ao lado do usuário do túnel

```

Windows 2000/XP:

```

ip6v install
ip6v rtu ::/0 2::$ipv4a pub
ip6v adu 2/$ipv6b

```

Windows 2008/Vista/7

```
netsh interface ip6v add v6v4tunnel interface=IP6Tunnel $ipv4b $ipv4a
```

```
netsh interface ipv6 add address IP6Tunnel $ipv6b
netsh interface ipv6 add route ::/0 IP6Tunnel $ipv6a
```

7. Dual Stack Lite (DS-Lite)

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Sim	Não	Não	Não	Sim	Não	Não	Não	Não	Não

Serão analisadas agora algumas técnicas bastante pertinentes ao momento atual da transição para o IPv6, num cenário em que não há mais IPv4 disponíveis, mas a base de usuários do provedor continua a crescer e ainda há muitos serviços exclusivamente disponíveis em IPv4 na Internet. Desta forma, o provedor não pode oferecer exclusivamente conectividade IPv6 ao usuário final, sendo forçado a oferecer também conectividade IPv4, mas com IPs de alguma forma compartilhados.

A primeira técnica a ser analisada será o Dual Stack Lite (Pilha dupla simplificada), padronizada na [RFC 6333](#). Ela pode ser aplicada em situações em que o provedor já oferece IPv6 nativo para seus usuários. Sua implementação necessita de um equipamento denominado AFTR (Address Family Transition Router), que implementa um CGN (Carrier Grade NAT), que é um NAT de grande porte, na rede do provedor. Entre o AFTR e o CPE do usuário utiliza-se um túnel IPv4 sobre IPv6 para transportar o tráfego IPv4. No contexto do DS-Lite, o CPE do usuário é chamado de B4, abreviação para DS-Lite Basic Bridging BroadBand. Nas extremidades desses túneis são usados endereços da faixa 192.0.0.0/29, especialmente reservada para este fim. Para o CPE do usuário e os demais equipamentos da rede do usuário são utilizados IPs da [RFC 1918](#) e não há problema se diferentes usuários utilizarem faixas de IPs repetidas, dado que o AFTR identifica os diferentes túneis com base no IPv6 de origem dos pacotes encapsulados. Na CPE do usuário deve existir um DHCP v4 para a distribuição dos endereços na rede interna. Deve existir também um proxy DNS, que permita consultas via IPv4, mas faça essas consultas ao DNS recursivo do provedor via IPv6, evitando traduções desnecessárias no AFTR.

É importante frisar alguns pontos:

- O AFTR usa CGN, mas não força o usuário a utilizar duplo NAT. Ou seja, AFTR realiza a função de NAT, de forma concentrada, para cada um dos dispositivos de cada usuário.
- O DS-Lite utiliza endereços privados na faixa 192.0.0.0/29 para as extremidades dos túneis v4 sobre v6, evitando a utilização desnecessária de endereços IPv4 na infraestrutura do provedor.

A figura 19 mostra um exemplo de topologia.

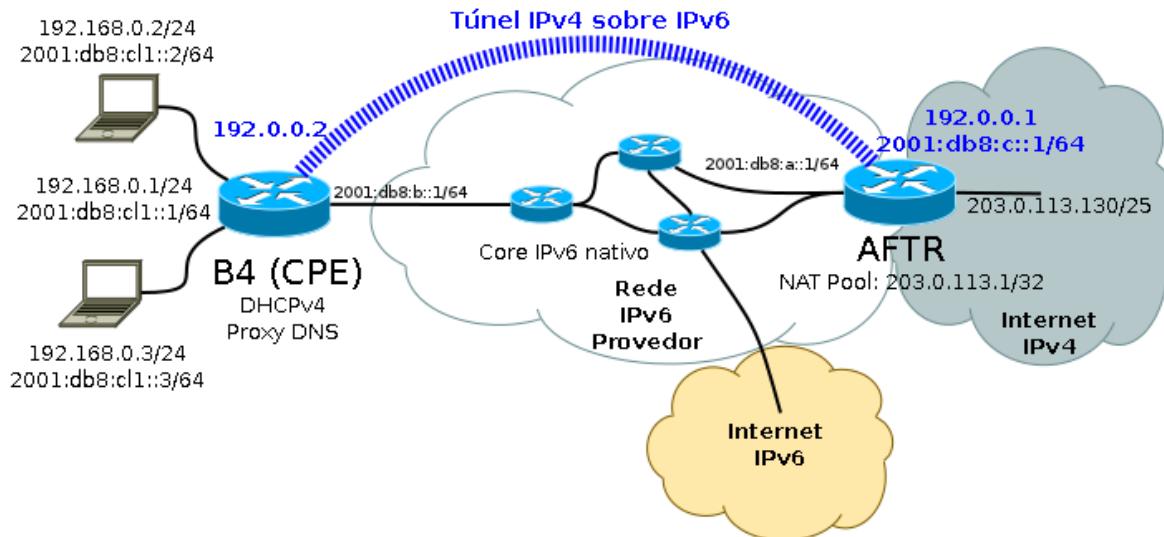


Figura 19: Exemplo topologia DS-Lite

Uma alternativa para implantar o DS-Lite é a utilização do software AFTR desenvolvido pelo ISC (Internet Systems Consortium), inicialmente por solicitação e com financiamento da Comcast, um grande provedor que opera com cabo nos Estados Unidos. O software está disponível no URL <http://www.isc.org/software/aftr> e pode ser utilizado em servidores GNU/Linux no provedor, permitindo uma implementação de baixo custo, robusta e escalável. Para o B4 (CPE) podem ser utilizados também dispositivos rodando Linux. Em especial, é possível utilizar roteadores Linksys WRT54GL e outros compatíveis com o firmware OpenWRT disponível no URL:

<http://www.kangaroo.comcast.net/wiki/doku.php?id=wrt54gl:wrt54gl>.

A configuração desta topologia é bastante simples. Para configurar o AFTR, basta criar um arquivo chamado aftr-script, contendo:

```
aftr_start() {
    set -x
    ip link set tun0 up
    ip addr add 192.0.0.1 peer 192.0.0.2 dev tun0
    ip route add 203.0.113.1/32 dev tun0
    ip -6 addr add fe80::1 dev tun0
    ip -6 route add 2001:db8:c::1/64 dev tun0
    arp -i eth0 -s 203.0.113.131 0a:0b:0c:0d:0e:f0 pub
}
aftr_stop() {
    set -x
    ip link set tun0 down
}
case "$1" in
start)
    aftr_start
;;
stop)
    aftr_stop
;;
*)
    echo "Usage: $0 start|stop"
    exit 1
;;
esac
exit 0
```

E um arquivo de configuração chamado aftr.conf, contendo:

```
default tunnel mss on
defmtu 1450
address endpoint 2001:db8:c::1
address icmp 203.0.113.1
pool 203.0.113.1
acl6 ::0/0
```

E então iniciar o serviço.

Para o B4 (CPE), basta criar o túnel IPv4 sobre IPv6:

```
modprobe ip6_tunnel
ip -6 tunnel add dsltun mode ipip6 remote 2001:db8:c::1 local 2001:db8:0:b::1 dev eth1
ip addr add 192.0.0.2 peer 192.0.0.1 dev dsltun
ip link set dev dsltun up
ip route add default dev dsltun
```

Além disso, deve-se configurar o DHCPv4 e o proxy DNS no B4.

Esse exemplo de configuração usa apenas um endereço para o pool de NAT, mas poderiam ser utilizados mais. Note que o endereço IPv4 da interface física do servidor AFTR não está na mesma rede dos endereços usados no pool. O endereço IPv6 da extremidade AFTR do túnel não é o endereço físico da interface, mas outro, numa rede diferente. Os pacotes direcionados para os endereços do Pool IPv4 e para o endereço IPv6 da extremidade do túnel são roteados para a interface de túnel e tratados pelo software AFTR. Por fim, é importante notar que os mesmos endereços 192.0.0.1 e 192.0.0.2 são usados para múltiplos clientes e que a detecção de novos túneis de clientes é feita automaticamente pelo AFTR, com base no endereço IPv6 de origem dos mesmos.

Uma variação desta técnica, que tenta resolver o mesmo problema, é a combinação do DS Lite com o Address Plus Port (A+P) e é conhecida como DS Lite A+P. O A+P será apresentado com mais detalhes no item 17 deste texto. O funcionamento do DS Lite A+P é similar ao DS Lite, mas ao invés de ser um endereço IPv4 privado, o CPE do usuário recebe um endereço IPv4 público. As portas disponíveis para utilização, contudo, são limitadas, pois este IP público é compartilhado com outros nós. O CPE deve então realizar a função de tradução de endereços (NAT), oferecendo IPs privados (RFC 1918) para os demais nós na rede, mas obedecendo à restrição das portas imposta pelo A+P na tradução.

Com a utilização do DS-Lite com A+P, a escalabilidade é melhor, dado que o NAT é feito de forma distribuída, nos CPEs. O usuário pode também realizar o mapeamento de portas no NAT e receber conexões entrantes, numa situação muito próxima a que existiria sem o compartilhamento de endereços.

O DS Lite com A+P é ilustrado na figura a seguir. Na figura, o CPE recebe o endereço IPv4 restrito das portas 1024 a 2047, à guisa de exemplo, mas tanto as portas disponíveis, quanto a quantidade das mesmas, poderiam ser outras.

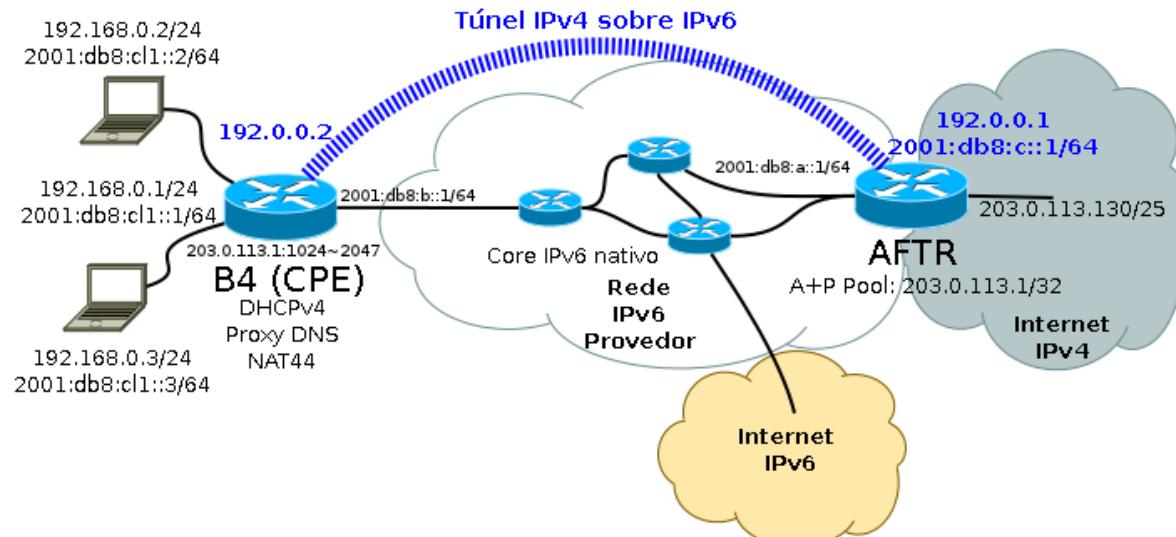


Figura 20: Exemplo topologia DS-Lite com A+P

Deve-se notar que o DS-Lite e o DS-Lite com A+P usam IPv4 sobre IPv6, mas utilizam-se de técnicas stateful para o compartilhamento dos endereços IPv4.

8. IVI, dIVI e dIVI-pd

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Sim	Sim	Não	Não	Sim	Sim	Não	Não	Não	Não

No item anterior foi apresentado o DS-Lite, que era útil para utilização no provedor, num cenário em que não há mais endereços IPv4 disponíveis, mas onde sua base de usuários continua a crescer. Desta forma, o provedor não pode oferecer exclusivamente conectividade IPv6 ao usuário final, sendo forçado a oferecer também conectividade IPv4, mas com IPs de alguma forma compartilhados.

O dIVI (**draft-xli-behave-divi-04**) e o dIVI-pd (**draft-xli-behave-divi-pd-01**) são alternativas de solução para o mesmo problema, mas têm a vantagem de usar técnicas stateless baseadas numa dupla tradução de pacotes, diferentemente do DS-Lite, que é stateful e baseado em tunelamento. Estes métodos de tradução stateless são capazes de manter a transparência fim a fim do endereço IP, não necessitando de técnicas auxiliares como DNS64 (tradução de DNS) ou ALG (gateways para aplicações específicas). Ambos os protocolos usam compartilhamento de IPs v4 com restrição de portas, de forma análoga ao A+P, discutido no item 17.

Ambas as soluções são extensões do IVI (**RFC6219**), cujo nome vem da concatenação dos numerais romanos IV (4) e VI (6). O IVI é um mecanismo de tradução stateless 1:1, desenvolvido por pesquisadores da CERNET2, a rede acadêmica chinesa, que é somente IPv6. A China optou por criar uma rede acadêmica IPv6 pura, totalmente nova, no lugar de implantar o IPv6 em pilha dupla na rede já existente. Essa estratégia pode parecer estranha atualmente, mas permitiu o desenvolvimento da indústria nacional de equipamentos de rede e, em conjunto com incentivos econômicos para uso da nova rede, alavancou a implantação do IPv6 nas universidades e o desenvolvimento de diversas aplicações. Em muitas universidades chinesas, na atualidade, o tráfego IPv6 é maior do que o IPv4.

O IVI foi criado inicialmente para permitir que servidores IPv6, ligados à CERNET2, pudessem comunicar-se com a Internet IPv4. Para isso um endereço IPv4 é atribuído virtualmente ao dispositivo, utilizando-se um mecanismo de tradução de pacotes stateless.

As três soluções, IVI, dIVI e dIVI-pd são experimentais. Existem relatos de implantação e teste realizados na CERNET/CERNET2 e pela China Telecom. Para o IVI há código disponível publicamente, na forma de um patch para o kernel do Linux, para o dIVI e o dIVI-pd não há implementações públicas disponíveis.

Em primeiro lugar, será apresentado o funcionamento do **IVI**.

Pode-se entender o conceito do funcionamento do IVI imaginando-se que ele cria um nó IPv6 espelho para o IPv4 e um nó IPv4 espelho para o IPv6, sendo que um nó espelho é um endereço que simula a presença do dispositivo na rede, mas que na verdade encaminha os pacotes enviados a ele para o dispositivo real através da tradução stateless. O servidor ou usuário IPv6 nativo na rede atendida pelo IVI, embora não tenha um endereço IPv4 atribuído a si, é visto por um nó IPv4 na Internet por meio de seu “endereço espelho” e, de forma análoga, enxerga um nó IPv4 qualquer na Internet por meio de seu “endereço IPv6 espelho”.

A figura abaixo demonstra o conceito.

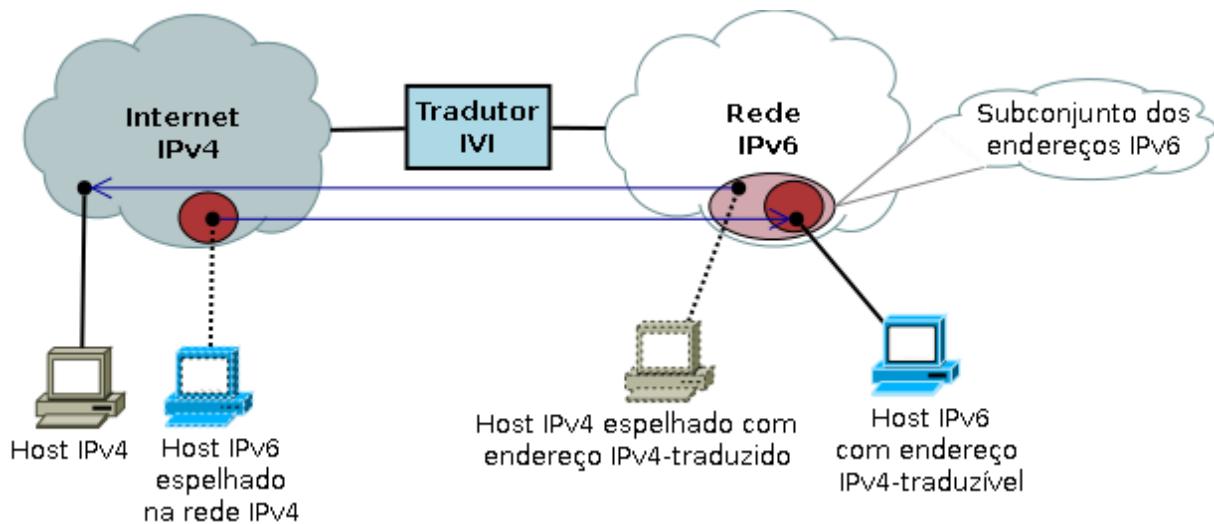


Figura 21: Explicação conceitual do IVI

O provedor, para implantar o IVI, escolhe um subconjunto de seu bloco IPv4, que será usado para formar endereços IPv6, a fim de atender os servidores, ou usuários somente IPv6, que se comunicarão com a Internet IPv4 por meio do IVI. Os endereços são mapeados conforme a figura a seguir.

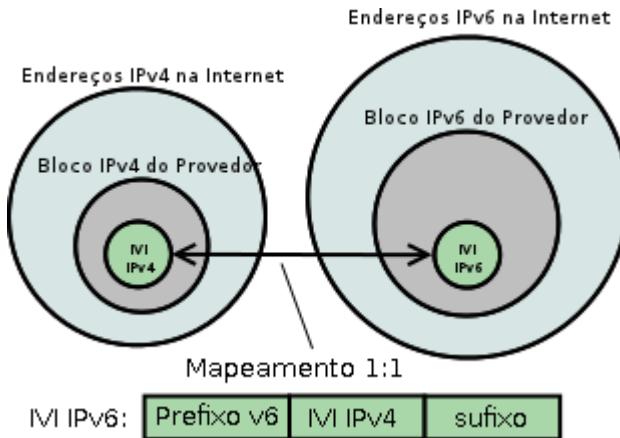


Figura 22: Mapeamento de endereços IPv6 em IPv4 e vice-versa

Por exemplo, um provedor cujo bloco IPv6 é 2001:db8::/32 e cujo bloco IPv4 é 192.61.100.0/24 pode escolher o prefixo IPv6 2001:db8:ff00::/40 e o bloco IPv4 192.51.100.0/26 para formar os endereços IVI. Os endereços IPv4 são mapeados para IPv6, então, da seguinte forma:

192.51.100.1 - 2001:db8:ffc0:3364:0100::0

192.51.100.2 - 2001:db8:ffc0:3364:0200::0

(...)

192.51.100.62 - 2001:db8:ffc0:3364:3e00::0

Na implementação feita pela CERNET o prefixo é um /40 e os bits de 32 a 39 são todos 1, para identificar o endereço IVI. Os bits 40 a 71 abrigam o endereço válido IPv4, representado no formato hexadecimal. Nesse formato, um IPv4 /24 é mapeado para um IPv6 /64 e um IPv4 /32 é mapeado para um IPv6 /72, mas o sufixo usado é sempre composto por zeros, de forma que apenas um endereço IPv6 é, na prática, mapeado para um endereço IPv4.

Note que o IVI não é uma solução para o esgotamento do IPv4. Para seu funcionamento é necessário separar um conjunto de endereços IPv4 válidos, do bloco disponível no provedor, e mapeá-lo para um conjunto de endereços IPv6 globais. A tradução no IVI é feita de forma stateless, o que é simples de implementar e escala bem. Note também que o IVI precisa funcionar em conjunto com o DNS64,

caso seja necessário resolver os nomes de dispositivos IPv4. Veja ainda que os métodos de tradução como IVI e NAT64 não funcionam com aplicações que carregam os endereços IP na camada de aplicação, sendo necessária a utilização de ALGs (Application Layer Gateways) nesses casos. Por fim, é importante observar que o IVI oferece conectividade fim a fim, seja IPv4 ou IPv6, para os dispositivos atendidos.

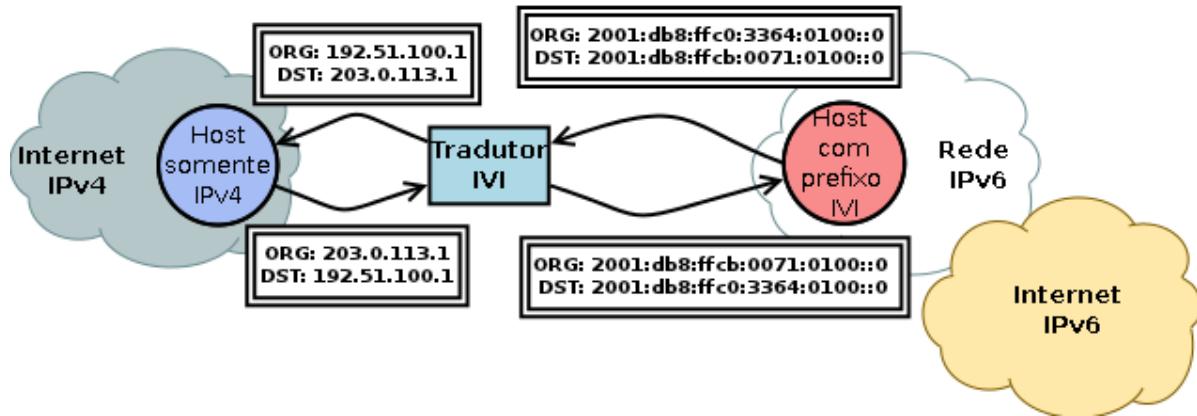


Figura 23: Tradução de endereços no IVI

As regras aplicadas na tradução do cabeçalho dos pacotes estão especificadas na **RFC 6145** e são mostradas a seguir:

Campo IPv4	Tradução para IPv6
Versão (0x4)	Versão (0x6)
IHL	(descartado)
Tipo de Serviço	Classe de Tráfego
Tamanho Total	Tamanho do Payload = Tamanho Total - IHL * 4
Identificação	(descartado)
Flags	(descartado)
Offset	(descartado)
Tempo de vida	Límite de Nós
Protocolo	Próximo Cabeçalho
CRC do Cabeçalho	(descartado)
Endereço de Origem	Aplicar mapeamento stateless IVI
Endereço de Destino	Aplicar mapeamento stateless IVI
Opções	(descartado)

Figura 24: Conversão de cabeçalhos IPv4 para IPv6

Campo IPv6	Cabeçalho IPv4 Traduzido
Versão (0x6)	Versão (0x4)
Classe de Tráfego	Tipo de Serviço
Etiqueta de Fluxo	(descartado)
Tamanho do Payload	Tamanho Total = Tamanho do Payload + 20
Próximo Cabeçalho	Protocolo
Limite de Nós	Tempo de Vida
Endereço de Origem	Aplicar mapeamento stateless IVI
Endereço de Destino	Aplicar mapeamento stateless IVI
---	IHL = 5
---	CRC do Cabeçalho Recalculado

Figura 25: Conversão de cabeçalhos IPv6 para IPv4

O caso de aplicação mais comum para o IVI é oferecer visibilidade IPv4 para servidores somente IPv6 dentro de uma determinada rede, mas ele poderia ser utilizado também para usuários, com a mesma finalidade, desde que haja uma quantidade suficiente de endereços IPv4 disponíveis. Os dispositivos que utilizarem o IVI devem usar endereçamento manual ou DHCPv6, pois o endereço precisa seguir um padrão específico que não pode ser obtido pela autoconfiguração IPv6.

O IVI ainda não está largamente implementado e atualmente a única maneira de testá-lo é através do Linux. Para usar o IVI é necessário aplicar um patch ao kernel do Linux, este patch está disponível em: <http://www.ivi2.org/IVI/>.

Após aplicar o patch é necessário habilitar a opção IVI e o protocolo IPv6 deve ser adicionado no modo “built in” e não como módulo. No menuconfig do kernel estas opções estão em:
Networking →

```
Networking Options →
    [*] IVI(test only)
    <*> The IPv6 protocol
```

Deve-se então compilar e instalar o kernel, e executar o script de configuração abaixo, fazendo as modificações de endereço necessárias para a rede onde a técnica será aplicada:

```
#!/bin/bash
# habilite o redirecionamento (forwarding)
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
echo 1 > /proc/sys/net/ipv4/conf/all/forwarding

# configure rota para IVI6 = 2001:0db8:ffc0:3364::/70,
#                           IVI4 = 192.51.100.0/26

# configure rota IPv6, sempre configurar rotas explicitas para as
# redes mapeadas, nao usar rotas default
route add -A inet6 2001:0db8:ffc0:3364::/70 gw 2001:db8::1 dev eth0

# configure mapeamento para      source-PF = 2001:db8::/48
# configure mapeamento para destination-PF = 2001:db8::/48

# para cada mapeamento, um pseudo-endereço único (10.0.0.x/8) deve ser configurado
ip addr add 10.0.0.1/8 dev eth0

# Mapeamento IPv4-to-IPv6, múltiplos mapeamentos podem ser feitos via múltiplos
# comandos:
```

```
# mroute IVI4-network IVI4-mask pseudo-address interface-PF destination-PF
mroute 192.51.100.0 255.255.255.192 10.0.0.1 eth0 2001:db8:: 2001:db8::

# Mapeamento IPv6-to-IPv4
# mroute6 destination-PF destination-PF-pref-len
mroute6 2001:db8:ff00:: 40
```

Uma vez apresentado o IVI, pode-se entender o funcionamento do dIVI e do dIVI-pd. Ambas as técnicas utilizam tradução dupla e compartilhamento de endereços IPv4, com restrição de portas. Dessa forma, os nós atendidos por elas usam pilha dupla, tendo IPv6 nativo para a comunicação com a Internet IPv6 e um IPv4 compartilhado. Com o dIVI e o dIVI-pd a comunicação fim a fim é possível, tanto com o uso de IPv4, como com o uso de IPv6, guardando-se a restrição de que apenas um subconjunto do total de portas está disponível para um determinado dispositivo no IPv4. Não é necessário o uso de NAT64 e nem de ALG.

A figura 26 representa tanto o funcionamento do dIVI, quanto do dIVI-pd.

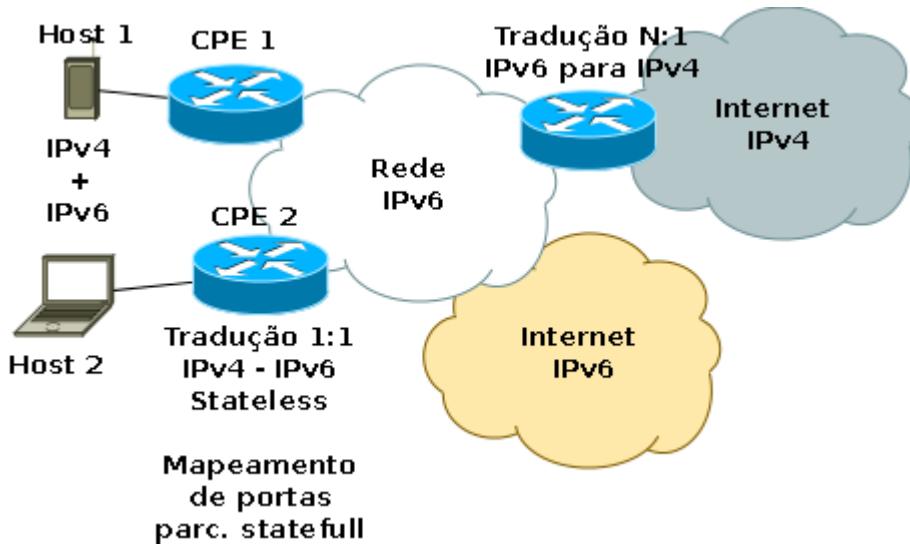


Figura 26: Topologia da rede com a utilização do dIVI e dIVI-pd

A seguir o funcionamento do **dIVI** será apresentado com maior nível de detalhamento.

No dIVI os endereços IPv4 são mapeados em IPv6 usando um formato definido no **draft-bcx-behave-address-fmt-extension**, que é uma extensão da **RFC 6052**. Nesse formato, que pode usar prefixos IPv6 com diferentes tamanhos, é possível carregar o endereço IPv4 que está sendo compartilhado e também informações que identificam quais são as faixas de portas que podem ser utilizadas pelo nó: o PSID que é uma espécie de identificador da CPE e o Q, que indica qual a taxa de compartilhamento de endereços. Com essas informações um algoritmo muito simples na CPE identifica quais portas podem e quais não podem ser usadas. De forma análoga, o tradutor IVI N:1 na rede do provedor traduz o IPv4 de destino para o endereço IPv6 correspondente, baseando-se tanto no endereço, quanto na porta. Os bits de 64 a 71, representados por “u”, devem possuir valor zero. Eles são reservados para compatibilidade com o formato de identificação de dispositivo na arquitetura de endereçamento IPv6, de acordo com a **RFC 4291**.

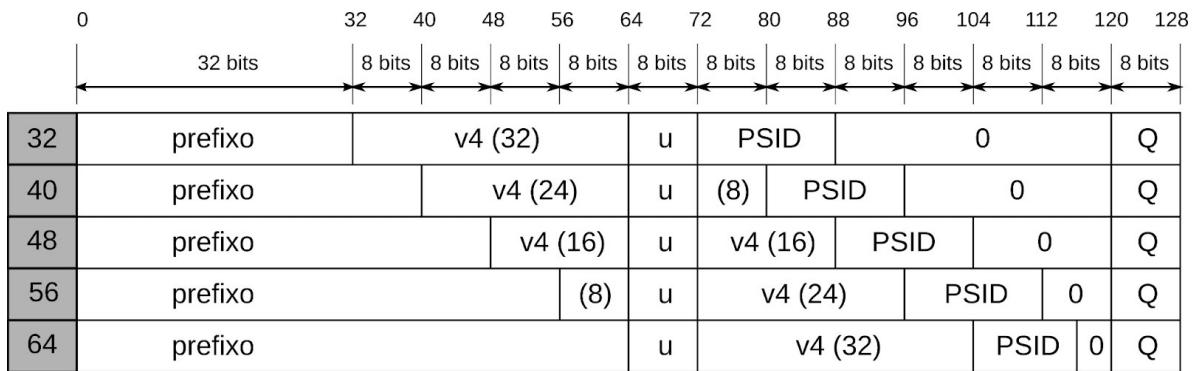


Figura 27: Endereçamento IPv6 traduzido do IPv4 pelo dIVI

Na CPE pode haver dois tipos de tradução. Pode ser feita uma tradução totalmente stateless, mas, nesse caso, o nó deve conhecer a priori a restrição das portas e enviar os pacotes já obedecendo a isso. Outra possibilidade é a tradução na CPE ser parcialmente statefull, de forma que o nó não precise obedecer a qualquer restrição quanto à porta de origem. Nesse segundo caso, a CPE deve fazer a tradução das portas e manter o estado dessa tradução.

Note-se que no dIVI apenas um endereço IPv4 e um endereço IPv6 são atribuídos ao dispositivo. Um caso de uso esperado para essa técnica é o uso para dispositivos móveis, em redes 3G ou 4G. O sistema operacional do dispositivo poderia suportar as funções da CPE dIVI, ou seja, tradução 1:1 de IPv4 para IPv6 e mapeamento de portas, de forma que, funcionando como um nó somente IPv6 na realidade, ainda assim pudesse oferecer uma API pilha dupla para as aplicações.

Uma vez detalhado o funcionamento do dIVI, pode-se agora apresentar com mais detalhes o funcionamento do **dIVI-pd**.

O dIVI e o dIVI-pd são, de fato, muito parecidos. O dIVI-pd permite que seja designado um prefixo IPv6 para o dispositivo, no lugar de um único endereço. Dessa forma é possível utilizar a autoconfiguração stateless, ou ainda endereçar toda uma rede com diversos nós. É importante observar que apenas um endereço IPv4 continua sendo atribuído para cada CPE no dIVI-pd, de forma que se for necessário atribuir endereços IPv4 para mais de um nó, a CPE deverá fazê-lo utilizando NAT44 e endereços privados, da **RFC 1918**.

O dIVI-pd também utiliza os endereços no formato definido pela **RFC 6052**, com o prefixo de comprimento /64 e dois tipos de extensões, o CPE index, como parte do prefixo e o sufixo no mesmo formato utilizado pelo dIVI.

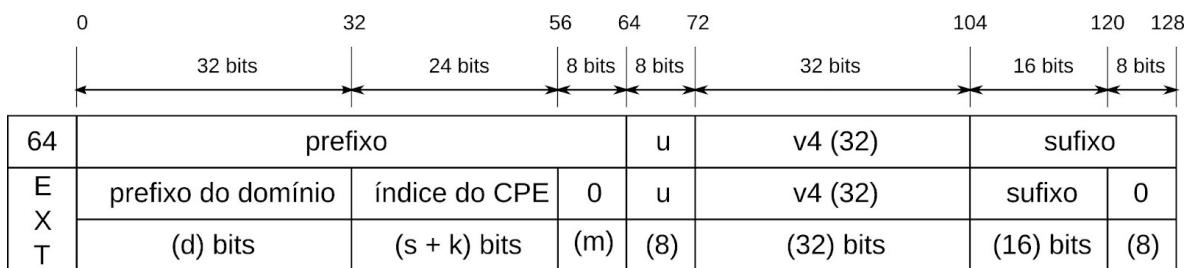


Figura 28: Endereçamento IPv6 traduzido do IPv4 pelo dIVI-pd

Note que para os usuários é possível atribuir prefixos /64, ou mais curtos, como /60 ou /56. Isso determina o valor de m (preenchimento com zeros). Note ainda que o fato do formato de sufixo ser o mesmo utilizado pelo dIVI permite a construção de CPEs compatíveis com as duas técnicas.

O **dIVI** e o **dIVI-pd** são realmente ótimas soluções, tendo características praticamente ideais para uso por provedores de acesso, por isso seu desenvolvimento e padronização devem ser acompanhados com muita atenção:

- operam com base em redes somente IPv6, que é para onde caminha a Internet;

- utilizam traduções stateless, que são simples de implementar e baratas do ponto de vista computacional, permitindo boa escalabilidade;
- permitem conexões em ambos os sentidos, mantendo a conectividade de fim a fim;
- não necessitam de ALG ou DNS64;
- quando necessário o uso de técnicas statefull, para tradução de portas, isso é feito no lado do usuário, mantendo o princípio de que a complexidade na Internet deve estar na extremidades e não próxima ao core da rede;
- a tradução implementada no provedor, de IPv6 para IPv4, N:1, pode ser usada de forma isolada, em conjunto com DNS64 e eventualmente ALGs, para clientes somente IPv6; CPEs com suporte à tradução no sentido inverso poderiam ser implementados apenas para os clientes para os quais esse método não for suficiente e que realmente necessitarem de endereços IPv4 nos dispositivos.

9. NAT64 e DNS64

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Sim	Não	Não	Não	Sim	Não	Não	Não	Não	Não

Uma outra técnica de tradução aplicável em situações similares as do IVI, dIVI e dIVI-pd, ou seja, para nós somente IPv6 acessarem a Internet IPv4 é o NAT64 (**RFC 6146**). O NAT64 é uma técnica stateful de tradução de pacotes IPv6 em IPv4. Ele necessita de uma técnica auxiliar para a conversão do DNS, chamada de DNS64 (**RFC 6147**). São sistemas distintos, mas que trabalham em conjunto para permitir a comunicação entre as redes IPv6 e IPv4.

O NAT64 necessita fazer a tradução de endereços IPv4 em IPv6, esta tradução é feita conforme ilustrado na figura 27. O processo é definido em detalhes na **RFC 6052**:

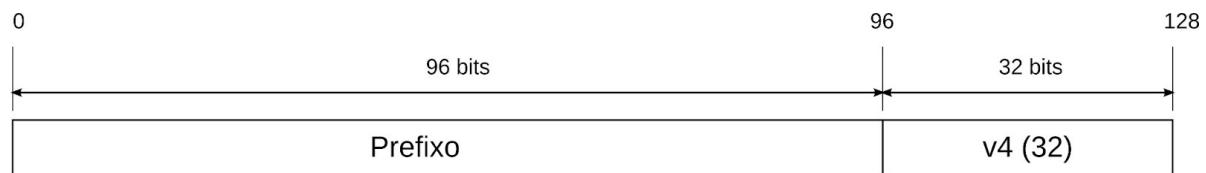


Figura 29: Endereçamento IPv6 traduzido do IPv4 pelo NAT64

Os bits 64 a 71 são reservados para a compatibilidade de identificação de *host* conforme a **RFC 4291** e devem ser zeros.

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.

Já a tradução do cabeçalho IPv6 em cabeçalho IPv4 e vice-versa é feita da mesma maneira que no IVI, já estudado anteriormente. O processo está ilustrado nas figuras 24 e 25 e é especificado em detalhes na **RFC 6145**.

O funcionamento do NAT64 é ilustrado no diagrama de sequência e na topologia das figuras 28 e 29, a seguir:

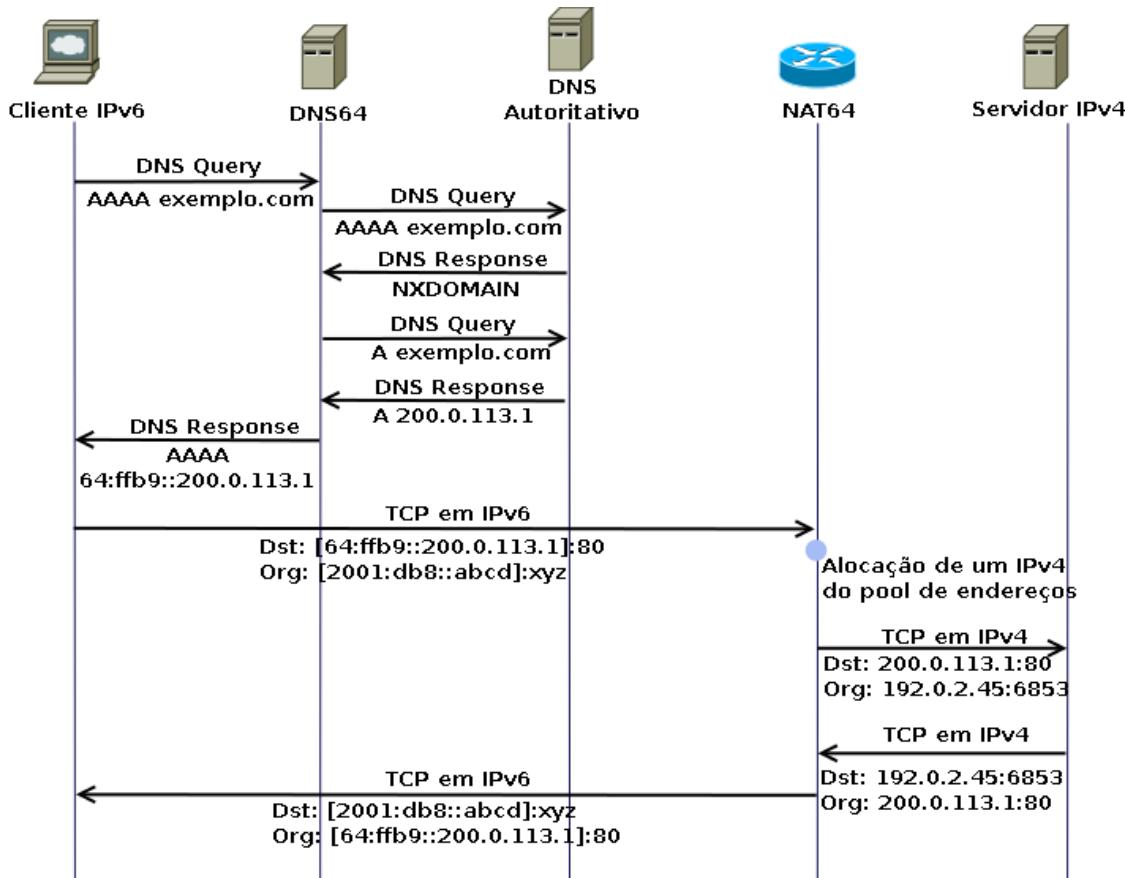


Figura 30: Diagrama de sequência do NAT64 / DNS64

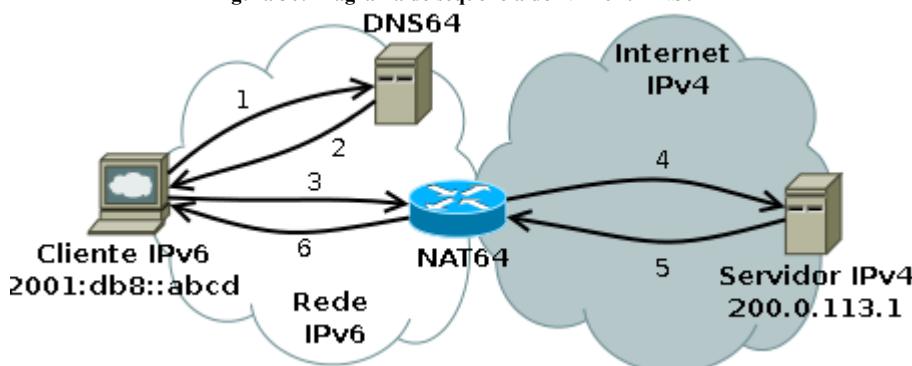


Figura 31: Topologia de rede do NAT64 / DNS64

O NAT64 possui implementações para Linux, Windows, grandes roteadores (Cisco e Juniper) e roteadores domésticos baseados em Linux. Como exemplo serão apresentadas a seguir configurações para Linux e Cisco.

Há várias opções de implementação do NAT64 no Linux, uma delas é a desenvolvida pelo projeto Ecdysis (<http://ecdysis.viagenie.ca>), que também pode ser utilizada em sistemas operacionais *BSD. Apesar da necessidade de instalar um módulo ao kernel do Linux, sua instalação é bastante simples. O arquivo fonte deve ser baixado em <http://ecdysis.viagenie.ca/download/ecdysis-nf-nat64-20101117.tar.gz> e descompactado em uma pasta adequada, na sequência os seguintes comandos devem ser executados como root:

1. Após o download, compile o módulo do kernel:

```
make && make install
```

2. No arquivo nat64-config.sh comente as seguintes linhas:

```
# 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
```

3. Habilite o módulo do kernel:

```
insmod      nf_nat64.ko      nat64_ipv4_addr=$IPV4_ADDR      nat64_prefix_addr=$PREFIX_ADDR
  nat64_prefix_len=$PREFIX_LEN
```

Os parâmetros usados acima são os seguintes:

- \$IPV4_ADDR = Endereço IPv4 da interface conectada à Internet
- \$PREFIX_ADDR = 64:ff9b::
- \$PREFIX_LEN = 96

4. Verifique, através do comando `lsmod`, se o módulo foi lido corretamente. A saída do comanda deve ser algo parecido com:

Module	Size	Used	by
nf_nat64	14542	0	

5. Rode o arquivo de configuração:

```
./nat64-config.sh $IPV4_ADDR
```

6. Verifique se a interface NAT64 está “up”, através do comando `ip link`.

7. Pode-se testar a conectividade via NAT64 através do comando

```
ping6 64:ff9b::200.160.4.22
```

Para fazer a configuração do NAT64 em roteadores Cisco os comandos são:

```
Router> enable
Router# configure terminal
Router(config)# ipv6 unicast-routing
Router(config)# interface giabitethernet0/0/0
Router(config-if)# description interface towards ipv4 side
Router(config-if)# ipv6 enable
Router(config-if)# ipv6 address 64:ff9b::/96
Router(config-if)# nat64 enable
Router(config-if)# exit
Router(config)# interface giabitethernet1/2/0
Router(config-if)# description interface towards ipv6 side
Router(config-if)# ip address 192.0.2.0 255.255.255.0
Router(config-if)# nat64 enable
Router(config-if)# exit
Router(config)# nat64 prefix stateless 64:ff9b::/96
Router(config)# nat64 route 192.0.2.0/24 gigabitethernet0/0/1
Router# end
```

Outra opção para Linux é o projeto Linux NAT64, disponível em:

<http://sourceforge.net/projects/linuxnat64/>.

Um exemplo de configuração do NAT64 para roteadores Juniper está disponível em:

http://www.juniper.net/techpubs/en_US/junos10.4/information-products/topic-collections/nce/nat64-ipv6-ipv4-depletion/configuring-nat64-ipv6-ipv4-depletion.pdf

Para o DNS64 as principais opções são o Bind (<http://www.isc.org/software/bind>), que possui versões para Linux e Windows, ou o Totd (<http://www.dillema.net/software/totd.html>), com versões para Linux e FreeBSD. Por ser mais atual e amplamente usado, o exemplo de configuração será baseado

no Bind. Após a instalação do Bind, as seguintes linhas devem ser adicionadas ao arquivo de configuração :

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

Depois, basta reiniciar o Bind para que a alteração tenha efeito.

É interessante notar que o DNS64 pode apresentar problemas em sua interação com o DNSSEC. Um validador DNSSEC que não saiba lidar com o DNS64 pode rejeitar todos os dados que vêm deste, como se não fossem válidos. A [RFC 6147](#), onde o DNS64 é definido, especifica formas de contornar o problema.

10. 464XLAT

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Sim	Não	Não	Não	Sim	Não	Não	Não	Não	Não

O 464XLAT ([draft-ietf-v6ops-464xlat-01](#)) é uma solução similar ao dIVI e ao dIVI-pd, que utiliza dupla tradução de IPv4 para IPv6, a fim de oferecer um IPv4 compartilhado para usuários IPv6 nativos. Esta técnica usa uma tradução stateless e outra stateful. O tradutor stateless é chamado de CLAT (*customer side translator*) e faz uma tradução 1:1, ou seja, cada IPv4 possui um IPv6 correspondente. O tradutor stateful é o PLAT (*provider side translator*) e faz uma tradução 1:N, onde vários IPv6 globais são representados por um IPv4 global para falar com a Internet IPv4.

O funcionamento do 464XLAT é ilustrado nas figuras a seguir:

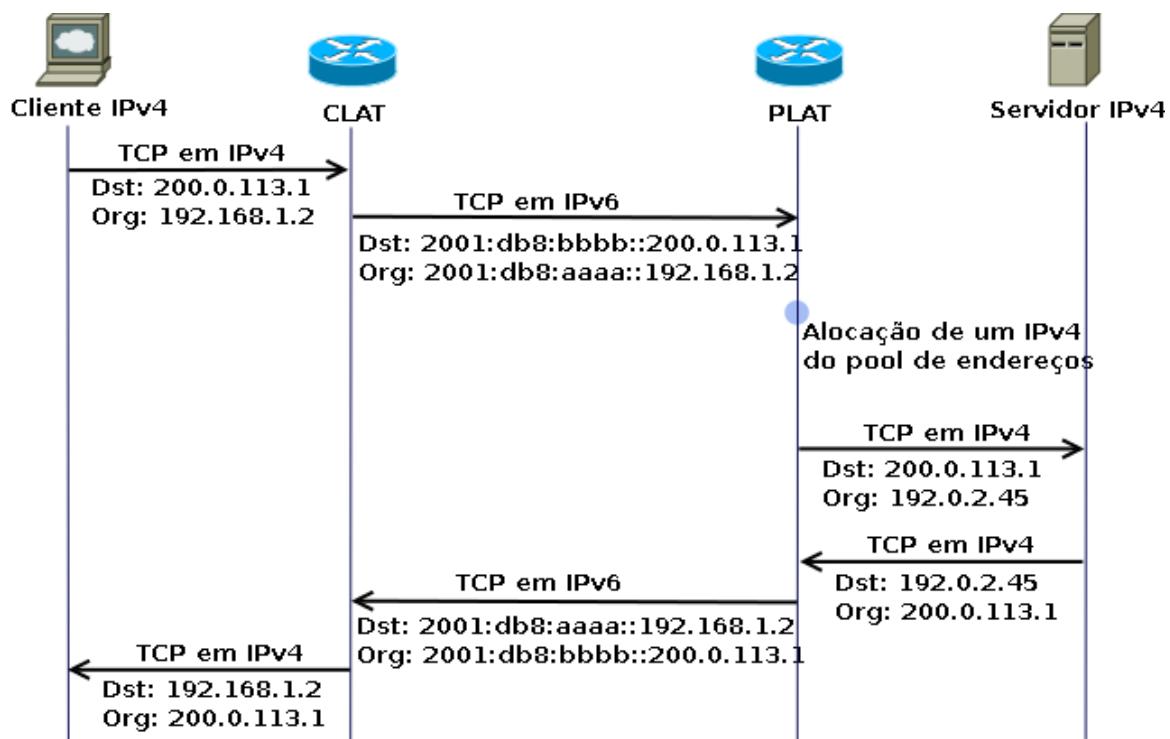


Figura 32: Diagrama de sequência do 464XLAT

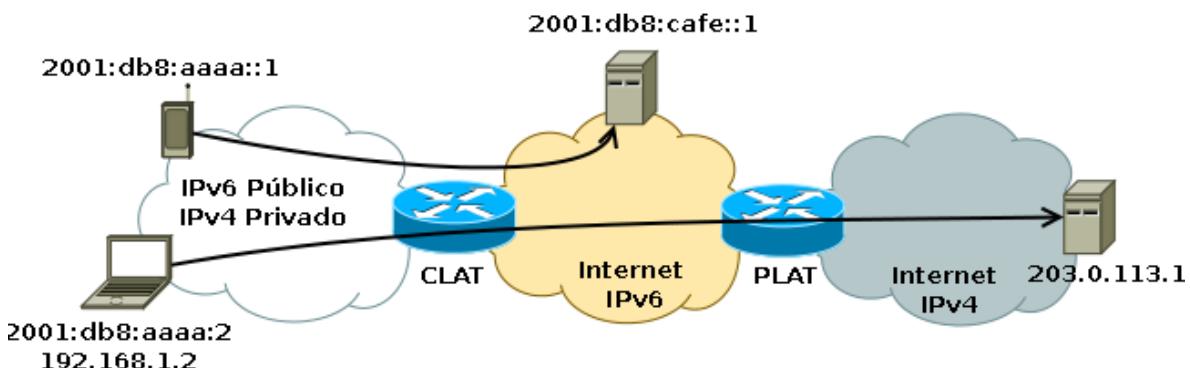


Figura 33: Topologia de rede do 464XLAT

É recomendado que haja um cache DNS implementado no CLAT, capaz de responder as solicitações dos clientes IPv4, fazendo as perguntas ao servidor DNS recursivo do provedor por meio do protocolo IPv6, evitando assim traduções desnecessárias para esse fim.

O uso da tradução stateless na extremidade do usuário e stateful no provedor não é a melhor escolha, se levarmos em consideração os princípios básicos de projeto da Internet. O inverso, como feito no dIVI, ou dIVI-pd, é mais recomendável. Contudo, o 464XLAT não é realmente uma técnica nova, projetada do zero, mas sim uma aplicação inovadora de duas técnicas já padronizadas e relativamente maduras.

O CLAT é uma tradução stateless baseada na **RFC 6145** e funciona de maneira semelhante ao IVI, mas utiliza IPv4 privado e não público. A forma como o endereço IPv4 é incluído no endereço IPv6 também é um pouco diferente. A regra de tradução é:

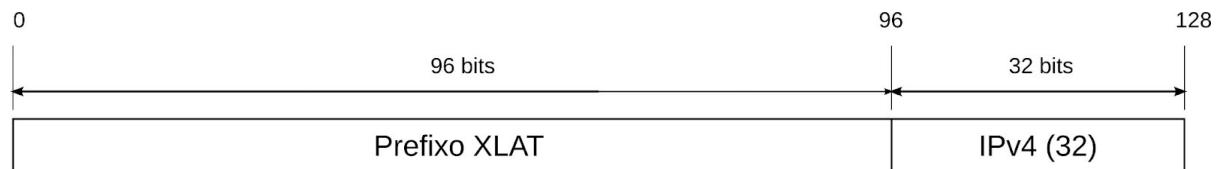


Figura 34: Endereço IPv6 traduzido do IPv4 pelo 464XLAT

Este prefixo XLAT de 96 bits é único por cliente e é atribuído a este pelo provedor do serviço. Como o prefixo utilizado é um /96 a autoconfiguração stateless não é possível e é necessário a utilização de DHCPv6 para a atribuição de endereços. O CLAT pode ser implementado no CPE ou em celulares. Para a implementação em celulares existe um projeto disponível para Android em <http://code.google.com/p/android-clat/>. Para a implementação em CPE pode-se utilizar o IVI (<http://www.ivi2.org/IVI/>), com as configurações adequadas.

Já o PLAT é um NAT64 (**RFC 6146**) que converte o endereço IPv6 em um dos endereços IPv4 disponíveis no banco de endereços do provedor, para fazer a sua implementação, basta seguir as recomendações feitas na seção anterior.

Testes foram realizados por pelo provedor estadunidense T-Mobile e o pelo Ponto de Troca de Tráfego japonês JPIX, em conjunto com seus participantes. Sua implementação em larga escala está sendo considerada. Os softwares ou implementações do CLAT e XLAT não são vinculadas e diferentes versões de um ou do outro podem ser utilizadas para obter o sistema desejado.

Embora não seja a solução ideal, do ponto de vista técnico, é uma solução que poderá ser usada em larga escala em breve, pois já existem implementações funcionais de seus componentes básicos. Outro ponto a considerar é que o 464XLAT pode funcionar em conjunto com NAT64, já que o PLAT é um NAT64. Basta acrescentar o DNS64 ao sistema. Usuários podem ser somente IPv6 e usar o NAT64/DNS64 e migrar para a utilização do 464XLAT, acrescentando um CPE que execute a função de CLAT, caso haja, por exemplo, aplicações que não funcionem com o NAT64.

11. 4rd

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Sim	Sim	Não	Sim						

O 4rd é uma solução similar ao DS-Lite, no sentido de que usa túneis 4in6 para fornecer IPs versão 4 compartilhados para usuários que já têm IPv6 nativo. Mas, de forma análoga às técnicas de tradução dIVI e 464XLAT, o 4rd é stateless e usa compartilhamento de IPs com restrição de portas.

A técnica foi desenvolvida com base no 6rd, que será estudado mais à frente, e está em processo de padronização, definida atualmente no [draft-despres-intarea-4rd-01](#). Existe uma implementação pública que funciona no vyatta e pode ser encontrada no seguinte URL: <http://bougaidenpa.org/masakazu/archives/176>. Produtos da linha SEIL do provedor japonês IIJ também implementam o protocolo.

A figura a seguir ilustra uma situação típica de uso do 4rd.

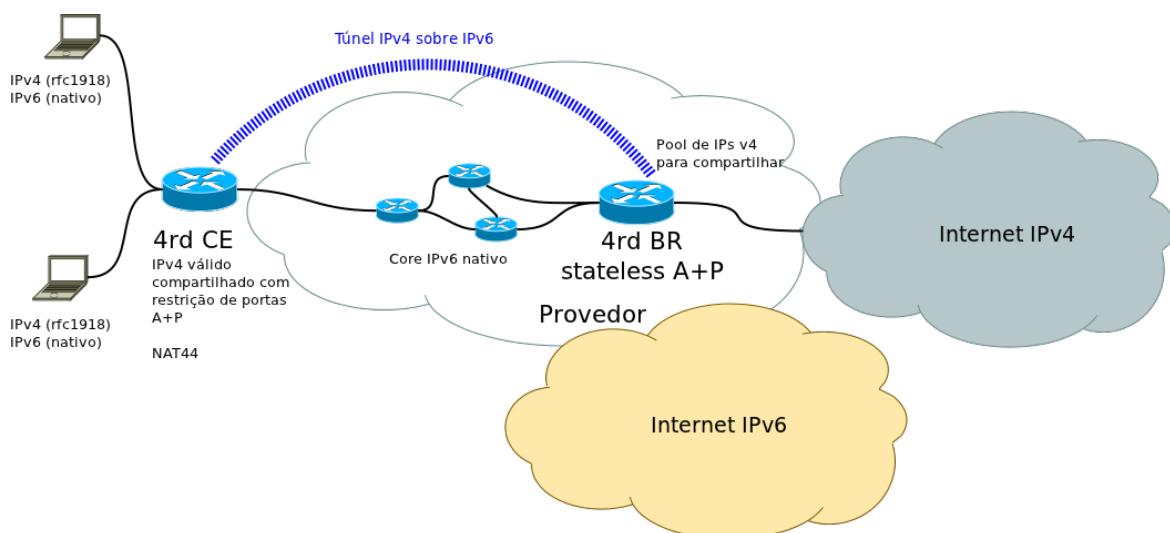


Figura 35: 4rd

É importante considerar que o 4rd, além de distribuir IPs versão 4 compartilhados com A+P, pode também distribuir IPs válidos sem restrição de portas, para cada CPE, além de subredes IPv4.

A figura abaixo representa a forma como os endereços IPv4 e IPv6 são mapeados 1:1, para o caso em que os endereços IPv4 são compartilhados com A+P. Perceba que o mapeamento é stateless:

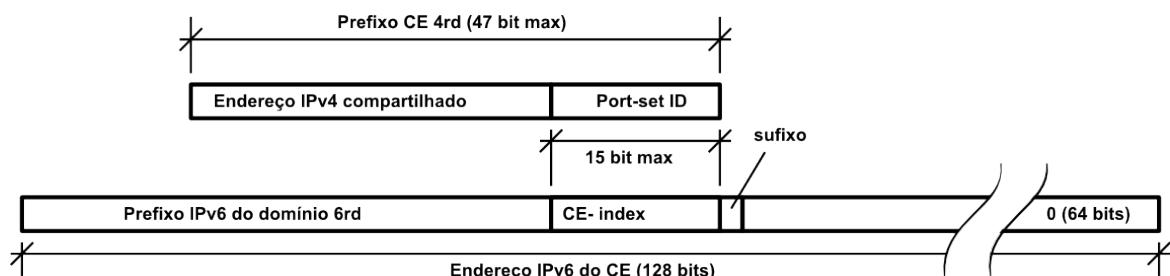


Figura 36: Tradução de endereços feita pelo 4rd

Para o mapeamento das portas, regras simples foram estabelecidas. As portas baixas, de 0 a 4096, nunca são designadas a um cliente. O tamanho do port-set-ID pode ir de 1 a 15 bits. Os clientes podem receber de 1 a 4 blocos diferentes de portas, de tamanhos variados, conforme o tamanho do port-set-ID. Os 16 bits de cada porta disponível são definidos da seguinte forma:

- **1º. bloco:** 0001 + port-set-ID (n=1 a 12 bits) + sufixo (que varia de 0 a 12-n)
- **2º. bloco:** 001 + port-set-ID (n=1 a 13 bits) + sufixo (que varia de 0 a 13-n)
- **3º. bloco:** 01 + port-set-ID (n=1 a 14 bits) + sufixo (que varia de 0 a 14-n)

- **4o. bloco:** 1 + port-set-ID (n=1 a 15 bits) + sufixo (que varia de 0 a 15-n)

Note que se o port-set-ID tiver 15 bits, apenas um bloco estará disponível, se tiver 14 bits, 2 blocos, se tiver 13 bits, 3 blocos, e se tiver de 1 a 12 bits, 4 blocos de portas estarão disponíveis.

A tabela a seguir mostra a quantidade de CPEs possível para cada escolha, assim como a quantidade de portas disponíveis para cada uma, para um mesmo IPv4 compartilhado.

tamanho do port set (bits)	qtd de ID's (CPEs)	Head=0001 1o. bloco	Head=001 2o. bloco	Head=01 3o. bloco	Head=1 4o. bloco	total de portas	portas não utilizadas
1	2	2048	4096	8192	16384	30720	4096
2	4	1024	2048	4096	8192	15360	4096
3	8	512	1024	2048	4096	7680	4096
4	16	256	512	1024	2048	3840	4096
5	32	128	256	512	1024	1920	4096
6	64	64	128	256	512	960	4096
7	128	32	64	128	256	480	4096
8	256	16	32	64	128	240	4096
9	512	8	16	32	64	120	4096
10	1024	4	8	16	32	60	4096
11	2048	2	4	8	16	30	4096
12	4096	1	2	4	8	15	4096
13	8192	-	1	2	4	7	8192
14	16384	-	-	1	2	3	16384
15	32768	-	-	-	1	1	32768

Figura 37: Modos de compartilhamento de portas

A configuração dos CPEs pode ser feita manualmente, mas a proposta também define uma opção de configuração via DHCPv6, que pode ser utilizada.

Assim como as técnicas dIVI e dIVI-pd, a técnica 4rd tem características praticamente ideais para uso por provedores de acesso, por isso seu desenvolvimento e padronização devem ser acompanhados com muita atenção:

- opera com base em redes somente IPv6, que é para onde caminha a Internet;
- utiliza traduções stateless, que são simples de implementar e baratas do ponto de vista computacional, permitindo boa escalabilidade;
- permitem conexões em ambos os sentidos, mantendo a conectividade de fim a fim;
- quando necessário o uso de técnicas statefull, para restrição de portas e compartilhamento via NAT44, isso é feito no lado do usuário, mantendo o princípio de que a complexidade na Internet deve estar na extremidades e não próxima ao core da rede;

12. 6PE e 6VPE

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Sim	Não							

Roteamento através de MPLS tem sido largamente utilizado nas redes dos grandes provedores de conectividade Internet. Entretanto, grande parte destes equipamentos já instalados não possuem suporte a IPv6. Dado o alto custo destes equipamentos, pode existir a necessidade de mantê-los em operação. No intuito de resolver este problema pode-se utilizar as técnicas apresentadas neste tópico.

As técnicas em questão são o 6PE e o 6VPE, definidas, respectivamente, nas **RFCs 4798 e 4659**, que permitem que redes IPv6 estabeleçam a comunicação por meio de um core MPLS IPv4, usando LSPs (Label Switch Paths). Sua implementação utiliza MBGP (Multiprotocol BGP) sobre IPv4 para se trocar rotas IPv6 e necessita que os PEs (Rot. Borda) sejam Pilha Dupla. Através do MBGP os roteadores de borda recebem as rotas IPv6 mas aplicam MPLS IPv4 sobre os pacotes para realizar o roteamento. Quando o pacote chegar à rede IPv6 de destino, o cabeçalho MPLS é removido e o pacote é encaminhado normalmente através do IPv6.

A diferença entre o 6PE e o 6VPE é que no primeiro caso, os roteadores mantém apenas uma tabela global de roteamento, de forma que o 6PE é mais indicado para provimento de conectividade Internet. Já os roteadores 6VPE são capazes de manter várias tabelas de roteamento separadas logicamente, de forma que a técnica é apropriada para prover serviços de redes privadas (VPNs).

A seguir o diagrama que explica o funcionamento do 6PE.

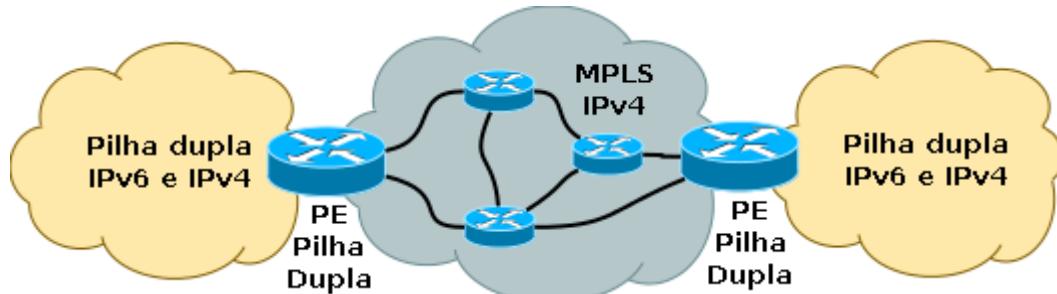


Figura 38: Topologia de rede 6PE

13. 6rd

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Sim	Não							

O 6rd tem o objetivo de permitir ao usuário final ter conexão com as redes IPv6 apesar da rede da operadora continuar funcionando em IPv4. Este tipo de técnica, assim como o 6PE/6VPE, permite que os provedores utilizem a infraestrutura IPv4 já existente para fazer uma implantação rápida do IPv6.

O 6rd ([RFC5569](#)) é uma extensão da técnica 6to4, que está em desuso e será melhor explicada no item seguinte. O 6rd resolve algumas das limitações técnicas do 6to4, como por exemplo sua assimetria e a falta de controle sobre os relays utilizados, permitindo sua utilização em larga escala.

Para entender o funcionamento do 6rd pode-se observar a figura 39, que ilustra a topologia típica de uso.

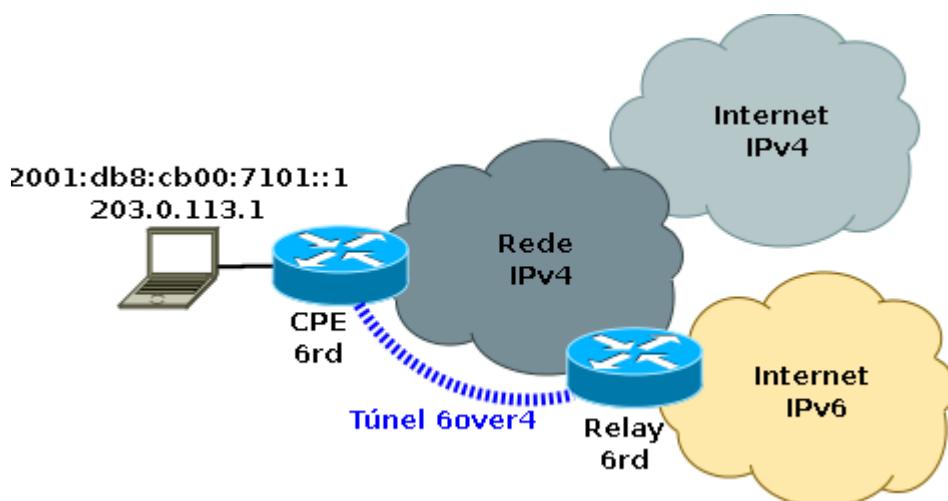


Figura 39: Topologia de rede 6rd

Analizando a figura, é possível notar que o 6rd depende basicamente de dois componentes:

- **CPE 6rd:** instalado como interface entre a rede da operadora e do usuário;
- **Relay 6rd:** instalado na interface entre a rede IPv4 da operadora e a Internet IPv6.

O CPE 6rd é um CPE tradicional (xDSL modem, cable modem, 3G modem etc), cujo software foi modificado para permitir o uso do 6rd. A necessidade dessa modificação dificulta a implementação da técnica, uma vez que requer a substituição, lógica ou física, de equipamentos em campo. Tal modificação nos CPEs normalmente é viável nos casos em que o provedor gerencia remotamente o equipamento, sendo capaz de fazer upgrades em seu firmware.

O 6rd relay é um equipamento que vai encapsular e desencapsular pacotes para trafegarem corretamente nas redes IPv4 e IPv6.

O CPE 6rd atribui ao usuário um endereço IPv4, como um CPE normal. Entretanto um endereço IPv6 também é atribuído ao usuário. Este endereço IPv6 é um endereço IPv6 público válido, mas é construído de maneira específica para que o relay 6rd identifique-o como um endereço 6rd. O endereço IPv6 atribuído é constituído da seguinte forma:

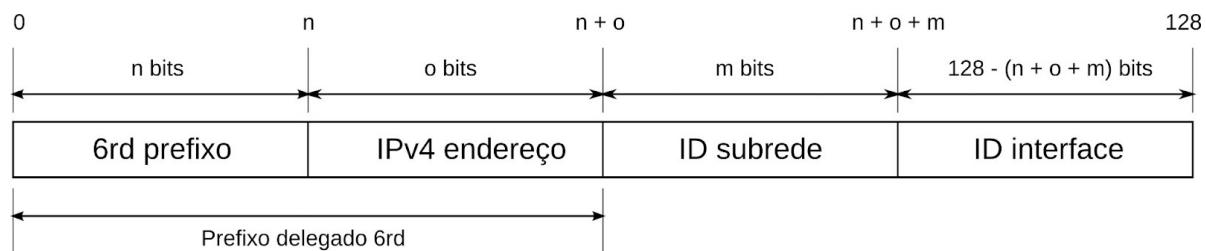


Figura 40: Tradução de endereço IPv4 para IPv6 no 6rd

No 6rd o tamanho **n** do prefixo e o tamanho **o** do endereço IPv4, que formam o prefixo delegado 6rd, são escolhas do provedor de acesso. Para permitir que a autoconfiguração de endereço stateless funcione é necessário que o tamanho deste prefixo **n + o** seja menor que 64 bits. O ID subrede de tamanho **m** pode ser definido pela operadora, mas é mais provável que a operadora deixe a definição do valor e tamanho do campo para o usuário final adequar às necessidades de sua rede.

Normalmente utiliza-se $n=32$, $o=32$ e $m=0$. Pode-se, contudo, aumentar o número de bits utilizados por **n** para além de 32, forçando o endereço IPv4 a utilizar parte dos 64 bits menos significativos, o que impede o funcionamento da autoconfiguração stateless. Para evitar que isto ocorra, o endereço IPv4 pode ocupar menos de 32 bits. Tal configuração é possível se os endereços IPv4 fizerem parte de uma mesma rede, pois pode-se omitir o prefixo da mesma. Por exemplo, se todos os endereços IPv4 forem da rede 198.51.0.0/16, os 16 bits que representam os números 198 e 51 podem ser omitidos e a representação do endereço IPv4 necessitará somente de 16 bits, ao invés dos 32 bits necessários para representar o endereço completo.

O 6rd é uma técnica funcional cuja implementação em massa foi testada com sucesso no provedor francês Free. Entretanto, a técnica não tem sido adotada por outros, principalmente pela necessidade de atualização do software ou de substituição dos CPEs.

Para configurar o CPE e o roteador de borda com Linux é necessário no mínimo o kernel mínimo 2.6.33 e as configurações para isto são:

Roteador relay 6rd:

```
ip tunnel add paraDentro mode sit local 203.0.113.1 ttl 64
ip tunnel 6rd dev paraDentro 6rd-prefix 2001:db8::/32
ip link set paraDentro up
ip -6 route add 2001:db8:cb00:7101::/64 dev paraDentro
ip -6 route add 2001:db8::/32 dev paraDentro
ip -6 route add 2000::/3 dev eth1
```

Roteador CPE 6rd:

```
ip -6 addr add 2001:db8:cb00:7182::/64 dev eth0
ip tunnel add paraFora mode sit local 203.0.113.130 ttl 64
ip tunnel 6rd dev paraFora 6rd-prefix 2001:db8::/32
ip link set paraFora up
ip -6 addr add 2001:db8:cb00:7182::1/128 dev paraFora
```

```
ip -6 route add ::/96 dev paraFora
ip -6 route add 2000::/3 via ::203.0.113.1
```

Configurando o 6rd com equipamentos Cisco os comandos seriam:

Roteador relay 6rd:

```
ipv6 general-prefix DELEGATED_PREFIX 6rd Tunnel0
interface Loopback0
  ip address 203.0.113.1 255.255.255.0
!
interface Tunnel0
  tunnel source Loopback0
  tunnel mode ipv6ip 6rd
  tunnel 6rd ipv4 prefix-len 8
  tunnel 6rd prefix 2001:db8::/32
  ipv6 address DELEGATED_PREFIX::/128 anycast
!
ipv6 route 2001:db8::/32 Tunnel0
ipv6 route 2001:db8:cb00:7101::/64 Null0
```

Roteador CPE 6rd:

```
ipv6 general-prefix DELEGATED_PREFIX 6rd Tunnel0
interface Dialer0
  ip address dhcp ! (203.0.113.130)
!
interface Tunnel0
  tunnel source Dialer0
  tunnel mode ipv6ip 6rd
  tunnel 6rd ipv4 prefix-len 8
  tunnel 6rd prefix 2001:db8::/32
  tunnel 6rd br 203.0.113.1
  ipv6 address DELEGATED_PREFIX ::/128 anycast
!
interface Ethernet0
  ipv6 address DELEGATED_PREFIX ::/64 eui-64
!
ipv6 route 2001:db8::/32 Tunnel0
ipv6 route ::/0 Tunnel0 2001:db8:cb00:7101::
ipv6 route 2001:db8:cb00:7182::/64 Null0
```

Já para fazer esta configuração em roteadores Juniper é possível encontrar um exemplo em:

<http://www.juniper.net/us/en/local/pdf/implementation-guides/8010078-en.pdf>

É importante deixar claro que o 6rd não é uma técnica para ser usada em novos usuários Internet, mas sim para os usuários já existentes, de forma a conseguir uma implantação muito rápida do IPv6. O 6rd funciona com base numa rede IPv4 e não resolve o problema da escassez de endereços. Técnicas escolhidas para novos usuários Internet devem preferencialmente basear-se em redes IPv6 e, quando necessário, preservar endereços IPv4, compartilhando-os.

14. 6to4

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Sim	Não							

O 6to4 (**RFC 3056**) é umas das técnicas de transição mais antigas em uso e é a técnica que inspirou a criação do 6rd. Sua concepção era simples e muito interessante: com ajuda de relays pilha dupla distribuídos na Internet, abertos, instalado de forma colaborativa por diversas redes, qualquer rede IPv4 poderia obter conectividade IPv6, através de túneis 6in4 automáticos.

Por meio do 6to4 qualquer computador com um IPv4 válido poderia funcionar como uma extremidade de um conjunto de túneis automáticos e prover todo um bloco IPv6 /48 para ser distribuído e usado em uma rede.

A técnica funcionou parcialmente e ainda é usada na Internet, mas apresenta diversos problemas. De fato, talvez tenha trazido mais problemas para a implantação do IPv6 de forma geral, do que ajudado.

O 6to4 é composto dos seguintes elementos:

- **Relay 6to4:** roteador com suporte ao 6to4 e que possui conexão nativa IPv4 e IPv6. Ele funciona como a extremidade dos túneis automáticos para os Roteadores 6to4 que precisam se comunicar com a Internet IPv6. Os relays 6to4 usam o endereço anycast IPv4 192.88.99.1 e anunciam rotas para 2002::/16 através deles, para a Internet.
- **Roteador 6to4:** roteador que suporta 6to4 que fica na extremidade de uma rede IPv4 e é responsável por trazer conectividade IPv6 para esta rede, por meio dos túneis 6to4. No caso dos acessos à Internet IPv6, ele direcionará o tráfego até o Relay Router mais próximo, que encaminhará o pacote ao seu destino. Para acesso a outras redes 6to4, os túneis são fechados diretamente com outros Roteadores 6to4.
- **Cliente 6to4:** equipamento de rede ou computador que usa endereços IPv6 fornecidos pelo túnel 6to4. O cliente 6to4 é um cliente pilha dupla convencional, normalmente numa rede doméstica ou corporativa, que pode usar IPv4 nativo ou compartilhado. O cliente não diferencia um endereço IPv6 obtido via 6to4, de um endereço IPv6 nativo.

As funções de Roteador e Cliente 6to4 podem estar presentes no mesmo equipamento. Um desktop convencional, por exemplo, usando Windows Vista, atua de forma automática como Roteador 6to4, desde que tenha um endereço IPv4 válido disponível.

O endereçamento 6to4, conforme definição da IANA, utiliza o prefixo de endereço global **2002:wwxx:yyzz::/48**, onde **wwxx:yyzz** é o endereço IPv4 público do cliente convertido para hexadecimal. O exemplo a seguir mostra como fazer a conversão de endereços:

Endereço IPv4: 200.192.180.002.

200=C8

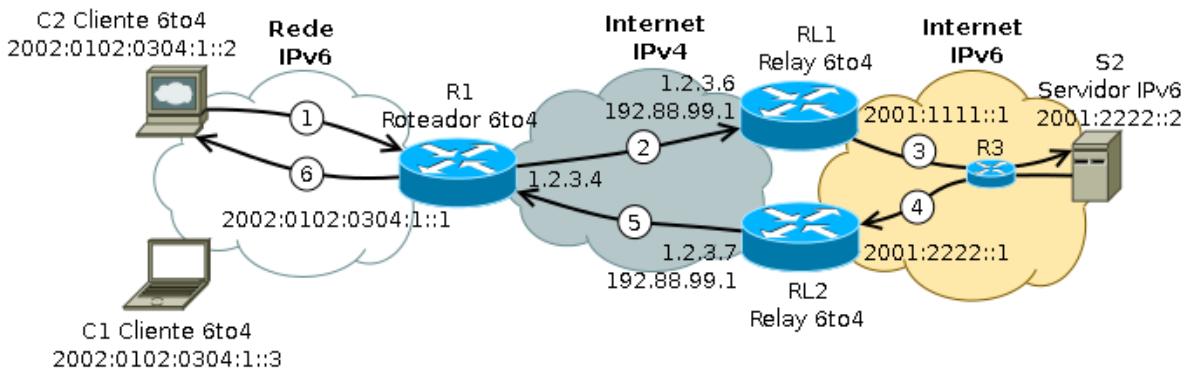
192=C0

180=B4

002=02

Com isso, o bloco IPv6 correspondente, via 6to4, é 2002:C8C0:B402::/48.

A figura e tabela abaixo demonstram o fluxo dos pacotes em uma rede 6to4. É importante notar que não existe a necessidade de os pacotes irem e voltarem pelo mesmo relay 6to4. As etapas 1, 3, 4 e 6 utilizam pacotes IPv6 e as etapas 2 e 5 utilizam pacotes IPv6 encapsulados em IPv4 através do protocolo 41.



Equipamento	Rota
RL1	::/0 rede IPv6 através da interface LAN / 2002::/16 através da interface virtual 6to4
RL2	::/0 rede IPv6 através da interface LAN / 2002::/16 através da interface virtual 6to4
S2	Rota padrão através de R3
R3	2002::/16 através do Relay RL2 (rota descoberta através da divulgação via BGP)
R1	::/0 através do Relay 6to4 RL1 ou RL2 utilizando a interface virtual 6to4 2002::/16 através da interface virtual 6to4 2002:0102:0304:1::64 para a rede local através da interface LAN
C1	::/0 através de R1 / 2002:0102:0304:1::64 através da interface LAN
C2	::/0 através de R1 / 2002:0102:0304:1::64 através da interface LAN

1 De acordo com a tabela de roteamento, o pacote é enviado através da rede local IPv6 para o roteador R1 utilizando a rota ::/0;

2 O pacote IPv6 é recebido por R1 através da interface LAN, que verifica sua tabela de roteamento e descobre que deve enviar o pacote para a interface virtual 6to4 (rota para a rede 2002::/16). Nesta interface o pacote IPv6 é encapsulado em um pacote IPv4 (protocolo tipo 41) e enviado ao Relay RL1 ou RL2 (O Relay 6to4 pode ser definido manualmente no roteador 6to4 ou então automaticamente através da utilização do endereço anycast 192.88.99.1). Vamos supor que o pacote foi enviado para o Relay RL1;

3 RL1 recebe o pacote 6to4 através de sua interface IPv4, vê que o pacote utiliza o protocolo 41 e o encaminha para a interface virtual. Esta desencapsula o pacote IPv6 e verifica na sua tabela de roteamento que deve enviá-lo pela interface LAN através do roteador R3, que simplesmente repassa o pacote IPv6 ao servidor S2;

4 S2 responde com o envio de outro pacote IPv6 com destino ao Cliente C2 utilizando a sua rota padrão, que aponta para o roteador R3. R3 recebe o pacote e, através da rota recebida via BGP, sabe que deve enviá-lo para o relay mais próximo, que é RL2;

5 RL2 recebe o pacote IPv6 e verifica que o destino é a rede 6to4 (2002::/16). Deste modo, de acordo com sua tabela de roteamento, ele encaminha o pacote para a interface virtual 6to4, que o empacota em um pacote IPv4 (protocolo 41) e o envia ao endereço IPv4 implícito no endereço IPv6 do destinatário do pacote;

6 O roteador R1 recebe o pacote através de seu endereço IPv4, verifica que o pacote está utilizando o protocolo 41 e o encaminha à interface virtual 6to4. Esta o desencapsula e verifica o endereço de destino. De acordo com sua tabela de roteamento e o endereço de destino, o pacote IPv6 é enviado através da sua interface LAN para o Cliente 6to4 C2.

Figura 41: Topologia e funcionamento do túnel 6to4

Dentre os problemas que afetam o 6to4, pode-se citar problemas de qualidade com relays públicos e problemas de segurança. Alie-se a isso o fato de que diversos sistemas operacionais suportam túneis 6to4 de forma automática, entre eles o Windows XP, o Windows Vista e o Windows 7. O fato dos sistemas operacionais ativarem os túneis 6to4 sem intervenção ou conhecimento dos usuários traz algumas consequências sérias. Uma delas é que firewalls ou outras medidas de segurança em redes corporativas podem ser inadvertidamente contornadas. Outra, é que, via túnel, os pacotes podem seguir caminhos mais longos, trazendo uma experiência pior para o usuário, em comparação àquela que ele teria se estivesse simplesmente usando IPv4. Um agravante é que não há relays públicos 6to4 no Brasil, ocasionando a ida do pacote para localidades distantes como América do Norte ou Europa, mesmo que a origem e o destino estejam no país.

Provedores de conteúdos e serviços na Internet podem sofrer com a questão, pois ao implantar o IPv6 em um servidor Web, por exemplo, usuários que antes acessavam-no bem via IPv4, podem passar a fazê-lo de forma lenta e instável, via IPv6 obtido automaticamente por meio de um túnel automático

6to4. Isso já foi motivo para adiamento da implantação do IPv6, mas atualizações dos sistemas operacionais têm mudado seu comportamento e o número de usuários potencialmente afetados diminuiu para patamares muito pequenos e aceitáveis.

Ainda assim, é recomendável agir para mitigar esse problema, principalmente porque existe uma medida bastante simples e efetiva, que pode ser utilizada. Deve-se lembrar que o caminho de ida do 6to4 pode ser diferente do caminho de volta. Isto permite que um relay 6to4 seja criado em um servidor Web, ou em uma rede, com o objetivo de responder às requisições dirigidas ao serviço advindas de clientes 6to4. A implementação deste relay não irá reduzir o tempo gasto para receber pacotes 6to4, mas garante que os pacotes 6to4 de resposta saiam da rede com destino ao originador da requisição, já encapsulados em IPv4 e isto dará a vantagem do tempo de resposta ser consideravelmente reduzido, já que não será necessário o pacote ir até o relay localizado no exterior. Esta redução pode melhorar bastante a experiência de acesso de um usuário que utilize 6to4 para acessar um serviço qualquer.

Para implementar este relay é necessário que os roteadores de borda da rede permitam a saída de pacotes com IP de origem 192.88.99.1. Provavelmente isto estará bloqueado por padrão na rede já que este IP não faz parte do bloco a ela designado. É preciso verificar também se o provedor de upstream não está filtrando também esse endereço. Normalmente se a rede em questão for um AS, com bloco próprio, o upstream não terá filtros antispoofing. Caso contrário, terá. Com a liberação do endereço, basta configurar o próprio servidor Web, ou um outro elemento na rede, para fazer o encapsulamento das respostas usando 6to4. No Linux a configuração para isto é:

```
ip tunnel add tunel6to4 mode sit ttl 64 remote any local 192.88.99.1
ip link set dev tunel6to4 up
ip addr add 192.88.99.1/24 dev lo
ip -6 addr add 2002:c058:6301::/16 dev tunel6to4
ip link set lo up
```

Para redes corporativas, é recomendável bloquear o protocolo 41 para evitar a utilização de túneis automáticos IPv4 pelos usuários. É possível também desabilitar essa função no Windows. Para isso deve ser criada e configurada uma chave de registro, do tipo DWORD:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tcpip6\Parameters\DisabledComponents

Ao fazer isso, a chave será criada com o valor 0x00. O bit 1 (do menos significativo para o mais significativo) deve ser mudado para 1, para desabilitar o 6to4. Ou seja, o valor da chave deve passar a ser 0x01. A função de todos os bits é descrita a seguir. Note que 0 é o valor padrão e significa que a função está ativa, 1 desativa a função:

bit 0: todos os túneis IPv6, incluindo ISATAP, 6to4 e Teredo;

bit 1: 6to4

bit 2: ISATAP

bit 3: Teredo

bit 4: interfaces IPv6 reais

bit 5: preferência por IPv4 e não IPv6

O 6to4 é, então, um protocolo com histórico importante, mas cujo uso deve ser evitado atualmente. Deve-se desativá-lo em redes corporativas e bloqueá-lo nos firewalls. Contudo, para redes pilha dupla que têm serviços IPv6 públicos na Internet, principalmente servidores Web, é recomendada a instalação de um relay 6to4 para responder a solicitações de usuários externos usando essa tecnologia, mitigando parte dos problemas trazidos pela mesma.

15. Teredo

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Sim	Não							

A técnica de tunelamento automática Teredo, criada pela Microsoft e definida na [RFC 4380](#), permite que nós localizados atrás de Network Address Translations (NAT), obtenham conectividade IPv6 utilizando tunelamento em IPv4, usando o protocolo UDP.

Sua utilização não é recomendada, dado que não é muito eficiente, tem alta taxa de falhas e algumas considerações de segurança. Contudo, é importante conhecê-la bem, já que está implementada e é utilizada de forma automática em algumas versões do Windows. A utilização de túneis automáticos implica que, mesmo a rede não tendo IPv6 implantado, usuários podem ter endereços IPv6 em seus dispositivos, inclusive com capacidade para receber conexões entrantes, contornando mecanismos e regras de segurança existentes no ambiente.

Existem dois elementos importantes no Teredo, o Servidor Teredo e o Relay Teredo. A conexão é realizada através de um Servidor Teredo, que a inicia após determinar o tipo de NAT usado na rede do cliente. Em seguida, caso o nó destino possua IPv6 nativo, um Relay Teredo é utilizado para criar uma interface entre o cliente e o nó destino. O Relay utilizado será sempre o que estiver mais próximo do nó destino e não o mais próximo do cliente.

Os Servidores Teredo utilizam a porta **UDP 3544** para comunicar-se com os dispositivos. Bloquear pacotes IPv4 enviados de uma rede para a Internet nessa porta e na direção inversa, é uma forma efetiva para evitar a utilização indesejada, muitas vezes involuntária, desse tipo de túneis.

Por padrão, os Windows 7 e Vista já trazem o Teredo instalado e ativado por padrão, enquanto que no Windows XP, 2003 e 2008, ele vem apenas instalado. Quanto ao FreeBSD e ao Linux, ele não vem instalado. Caso seu uso seja desejado é possível utilizar um software chamado miredo.

Para iniciar o túnel, existe uma comunicação entre o cliente e o Servidor Teredo com a finalidade de identificar o tipo de NAT usado na rede. Para isso são usados dois IPs versão 4 no servidor. O Teredo é capaz de funcionar com NAT do tipo Cone, também chamado de NAT Estático e NAT Cone Restrito, também chamado de NAT Dinâmico, mas não funciona com NAT Simétrico. Foge ao escopo deste texto explicar o funcionamento de cada tipo de NAT.

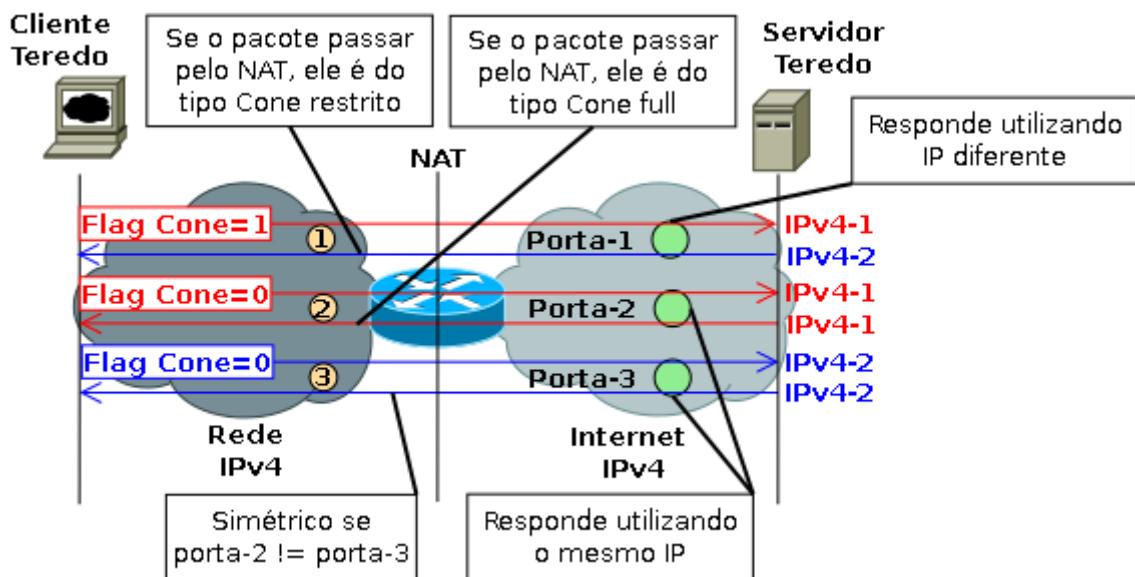


Figura 42: Definição do tipo de túnel Teredo

Uma vez identificado o tipo de NAT, o cliente constrói seu endereço IPv6, conforme a figura a seguir:

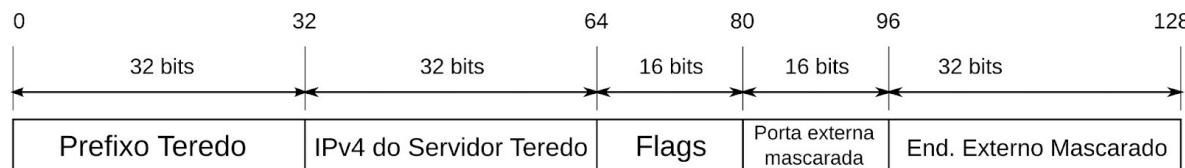


Figura 43: Tradução de endereço IPv4 para IPv6 no Teredo

O prefixo é sempre **2001:0000::/32**. As Flags servem para identificar o tipo de NAT.

A figura a seguir representa o estabelecimento do túnel Teredo na situação mais complexa possível, quando o cliente está numa rede com NAT Cone Restrito. Note que no estabelecimento do túnel, os primeiros pacotes fluem através do Servidor Teredo. Uma vez estabelecido o túnel, toda a comunicação é feita através do Relay, bidirecionalmente.

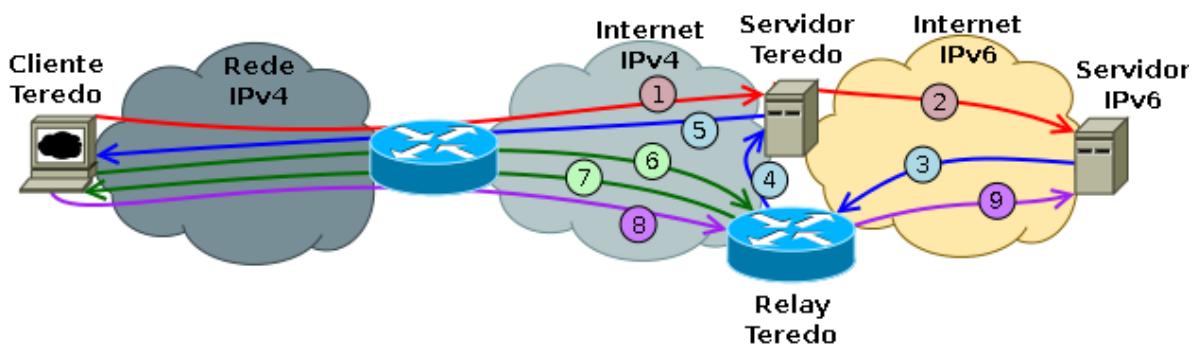


Figura 44: Estabelecimento de túnel Teredo

Além de bloquear a porta UDP 3544, para evitar a criação de túneis Teredo, estes podem ser desabilitados no próprio Windows. Para isso deve ser criada e configurada uma chave de registro, do tipo DWORD:

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tcpip6\Parameters\DisabledComponents`

Ao fazer isso, a chave será criada com o valor 0x00. O bit 2 (do menos significativo para o mais significativo) deve ser mudado para 1, para desabilitar o Teredo. Ou seja, o valor da chave deve passar a ser 0x02. A função de todos os bits é descrita a seguir. Note que 0 é o valor padrão e significa que a função está ativa, 1 desativa a função:

- bit 0:** todos os túneis IPv6, incluindo ISATAP, 6to4 e Teredo;
- bit 1:** 6to4
- bit 2:** ISATAP
- bit 3:** Teredo
- bit 4:** interfaces IPv6 reais
- bit 5:** preferência por IPv4 e não IPv6

Assim como o 6to4, o Teredo possui questões de segurança. Através do encapsulamento ele pode permitir que tráfego que seria bloqueado em IPv4 consiga chegar ao destino. Ele vem instalado e habilitado por padrão no Windows. Recomenda-se que seja desabilitado em redes corporativas

16. ISATAP

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Sim	Não							

ISATAP (sigla para Intra-Site Automatic Tunnel Addressing Protocol) é uma técnica de tunelamento que liga dispositivos a roteadores. Sua utilização ocorre dentro das organizações, pois não há um serviço público de ISATAP. É possível utilizar a técnica quando a organização tem IPv6 na extremidade de sua rede, fornecido por seu provedor, mas sua infraestrutura interna, ou parte dela, não suporta o protocolo.

A figura abaixo demonstra o conceito do ISATAP.

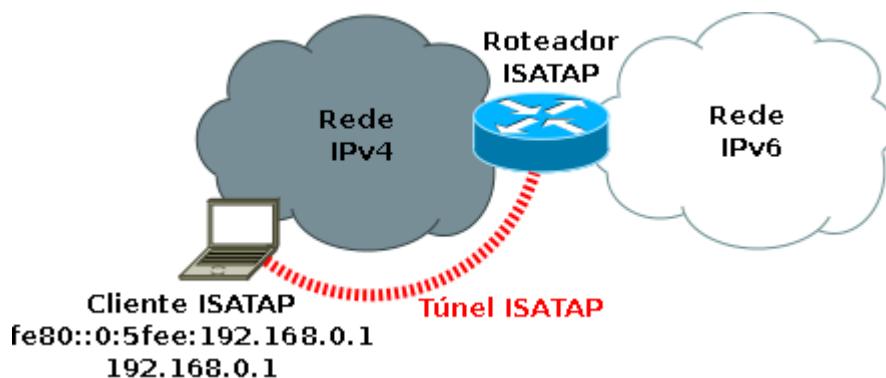


Figura 45: Topologia de rede ISATAP

Esta técnica, definida na **RFC 5214**, é baseada em túneis IPv6 criados automaticamente dentro da rede IPv4 e em endereços IPv6 associados aos clientes de acordo com o prefixo especificado no roteador ISATAP e no IPv4 do cliente. Para a criação destes túneis, são utilizadas as especificações da seção 3 da **RFC 4213**, que trata do tunelamento através do protocolo IPv4 tipo 41 ou 6in4.

Os endereços IPv4 dos clientes e roteadores são utilizados como parte dos endereços ISATAP, permitindo a um nó determinar facilmente os pontos de entrada e saída dos túneis IPv6, sem utilizar nenhum protocolo ou recurso auxiliar.

O formato do endereço ISATAP segue o seguinte padrão:

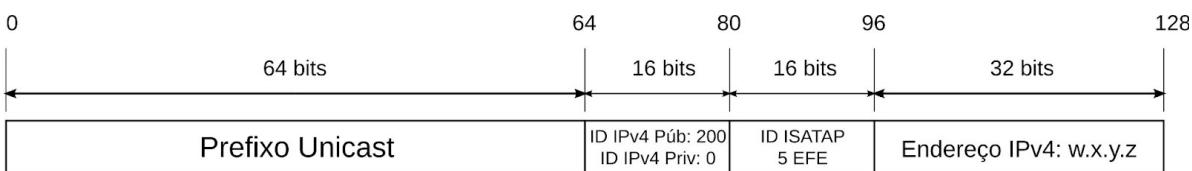


Figura 46: Tradução de endereço IPv4 para IPv6 no ISATAP

- Prefixo unicast : É qualquer prefixo unicast válido em IPv6, que pode ser link-local (FE80::/64) ou global. Normalmente utiliza-se uma rede /64 obtida a partir do prefixo global fornecido pelo provedor Internet para uso na rede.
- ID IPv4 público ou privado: Se o endereço IPv4 for público, este campo deve ter o valor "200". Se for privado (192.168.0.0/16, 172.16.0.0/12 e 10.0.0.0/8), o valor do campo será zero;
- ID ISATAP: Sempre tem o valor 5EFE;
- Endereço IPv4: É o IPv4 do cliente ou roteador em formato IPv4;

O ISATAP é suportado pela maior parte dos sistemas operacionais e roteadores e é de fácil implantação.

17. A+P

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Não								

Os mecanismos A+P e NAT444, que serão explicados a seguir, não ajudam diretamente na transição de IPv4 para IPv6, mas têm sido usados na tentativa de prolongar a vida útil do IPv4.

Esses mecanismos podem ser usados, contudo, em conjunto com a implantação nativa do IPv6, a fim de garantir conectividade IPv4 para os usuários, numa Internet em transição, onde muitos dos serviços disponíveis ainda são somente IPv4.

A técnica Address Plus Port (A+P), que significa endereço mais porta, está definida na **RFC 6346** e compartilha o mesmo endereço público com mais de um usuário, simultaneamente. Para isto ser possível é necessária uma limitação das portas que estarão disponíveis para cada um. É possível fazer a atribuição dos endereços e portas para os diferentes usuários de forma stateless.

O mesmo IPv4 válido é compartilhado entre diversos usuários diferentes. A CPE é responsável por fazer um NAT na rede do usuário, de forma a atender os dispositivos nela presentes com IPs privados, da **RFC 1918**, e obedecer a restrição de portas ao fazer a tradução. No caso da implantação em redes com dispositivos móveis, como smartphones, estes devem ser atualizados e estar cientes da restrição de portas.

Esta técnica provavelmente não seria notada por usuários domésticos comuns, pois estes continuariam com IPs válidos e técnicas atuais como STUN ou uPnP para permitir conectividade fim a fim, em conjunto com o NAT na rede do usuário, continuariam funcionando.

A restrição de portas não é, contudo, adequada para serviços corporativos, pois não permitira o uso de servidores em portas padronizadas. O problema pode ser agravado se as portas forem atribuídas de forma dinâmica.

A figura abaixo demonstra o funcionamento do A+P.

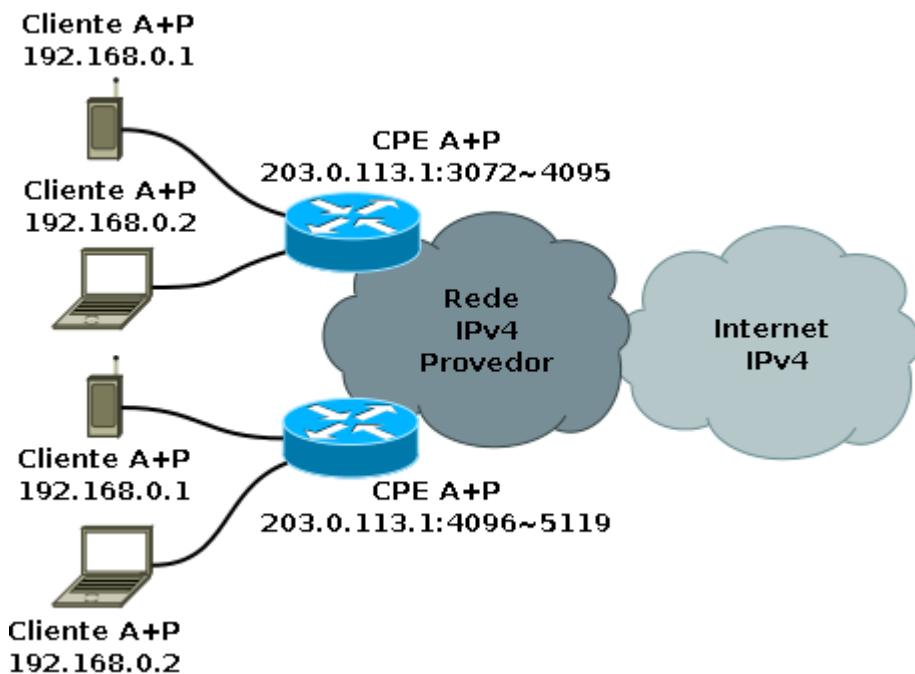


Figura 47: Topologia de rede A+P

Exemplos práticos da utilização do A+P já foram dados, nos itens que trataram do DS-Lite com A+P, dIVI e dIVI-pd.

A utilização de A+P, se possível, geralmente é preferível à utilização do NAT444, estudado no item seguinte. É importante lembrar que a utilização dessas técnicas deve ser sempre acompanhada da implantação do IPv6.

18. NAT444

Cenário	R6-I4	I4-R6	I6-R4	R4-I6	R6-R4	R4-R6	I6-I4	I4-I6	R6-I4-R6	R4-I6-R4
Suporta	Não	Não								

Assim como o A+P, o NAT444 tem sido usado na tentativa de prolongar a vida útil do IPv4 na Internet. Este mecanismo fere o princípio de comunicação fim a fim da Internet e seu uso deve ser evitado ao máximo. Alternativas que levem as redes na direção de redes somente IPv6 são preferíveis, assim como alternativas que usem métodos stateless e que mantenham a complexidade nas extremidades da rede.

Se usado, o NAT444 deve acompanhar a implantação do IPv6 nativo para os usuários. Não deve ser usado isoladamente.

O NAT444 é descrito no em **draft-shirasaki-nat444-05** e também é conhecido como LSN (Large Scale NAT) ou CGN (Carrier Grade NAT). Este mecanismo atribui um IPv4 privado para cada um dos usuários de um ISP, de forma semelhante ao que já é normalmente feito em redes domésticas e em diversas redes corporativas. Ou seja, os usuários conviverão, nesse caso, com duas camadas de NAT.

A utilização desta técnica resolveria, de forma provisória, o problema da falta de endereços IPv4, já que eles seriam largamente reutilizados, mas o custo seria comprometer as conexões fim a fim e possivelmente a “quebra” de diversas aplicações hoje existentes.

Pode-se argumentar que o NAT já é usado normalmente e que não há prejuízo na utilização da Internet por conta disso. Isso não é verdade. O NAT, na rede dos usuários, por si só, já é prejudicial, embora tenha desempenhado um importante papel nos últimos anos para a conservação dos endereços IPv4 na Internet. Técnicas como servidores STUN, uPnP e outras foram desenvolvidas para restaurar, parcialmente, a comunicação fim a fim perdida com uma camada apenas de NAT. Com o uso de NAT444 elas deixarão de funcionar.

Outro ponto a considerar é que essa técnica é cara, exigindo equipamentos com grande poder de processamento. Investimentos altos tendem a ser politicamente conservados dentro de grandes corporações, o que pode levar a um atraso na adoção do IPv6.

Um ponto a considerar, do ponto de vista estritamente técnico, é a escolha do bloco de IPs a ser usado no NAT. Como o uso dos blocos da RFC1918 é comum nas redes dos usuários, qualquer bloco escolhido dentre os disponíveis pelo provedor fatalmente colidirá com o bloco de algum de seus clientes. Existe uma proposta em estudo para a reserva de um novo bloco, exclusivo para a utilização em situações onde houver duplo NAT. O ARIN prontificou-se a ceder o bloco em questão e a proposta está sendo analisada pelo IETF: **draft-weil-shared-transition-space-request-15**.

Devido ao rápido esgotamento do IPv4, podem existir situações em que essa técnica terá de ser utilizada. Seu uso muitas vezes é incentivado por fabricantes de equipamentos, talvez devido ao alto custo dos equipamentos necessários para sua implementação.

A figura abaixo exemplifica o funcionamento das redes hoje e como ficará o funcionamento da rede com a utilização do NAT444.

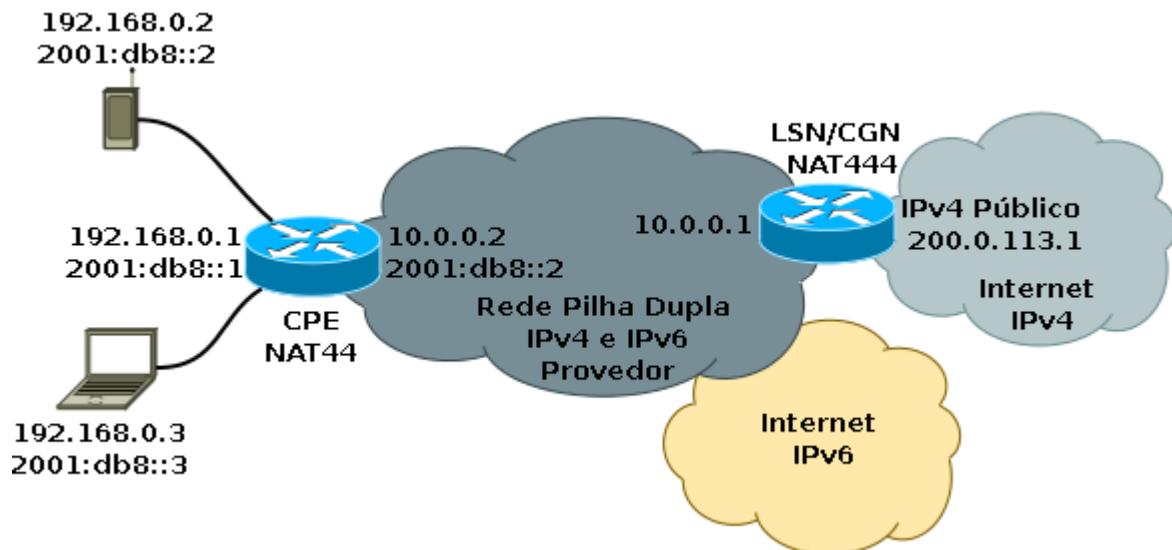


Figura 48: Topologia de rede NAT44

19. Considerações Finais

A utilização de pilha dupla IPv4 e IPv6 na Internet foi imaginada com a técnica padrão para uma transição sem solução de continuidade na migração para o IPv6. A pilha dupla parece ser, ainda hoje, a melhor escolha para provedores e redes corporativas, desde que não haja falta de endereços IPv4 válidos e, consequentemente, for possível utilizá-la.

O rápido esgotamento dos endereços IPv4, a existência de equipamentos legados onde não é possível utilizar IPv6 e a presença de equipamentos somente IPv6, por falta de IPs v4 livres, criaram a demanda por outras técnicas de transição. As principais foram apresentadas ao longo deste texto.

Deve-se considerar, ao escolher a técnica a ser usada em uma rede específica, que a Internet caminha para utilizar o IPv6. Não há dúvida de que no futuro a Internet utilizará majoritariamente IPv6 e, possivelmente, somente IPv6. Aqueles que trabalham com redes há pelo menos pouco mais de uma década viveram outras transições similares, como por exemplo, quando, em redes corporativas era comum o uso de IPX/SPX, Netbios, Appletalk e IP concomitantemente. A convergência tecnológica é um processo natural. Ela facilita o gerenciamento das redes, a interoperabilidade, o desenvolvimento de novas aplicações e serviços, reduzindo custos, de forma geral.

Com isso em mente, os provedores devem planejar-se para atender, daqui a pouco tempo, seus novos usuários exclusivamente com redes IPv6, de forma nativa. Devem oferecer paliativamente a eles um IPv4 válido, se houver disponível, ou compartilhado, se necessário. Isso deve ser feito enquanto houver uma quantidade relevante de dispositivos na Internet que não tenham implantado IPv6. Pode-se fazer isso com auxílio de técnicas de transição baseadas em túneis, ou tradução, usando a rede que será exclusivamente IPv6.

Há muitas técnicas disponíveis. Algumas delas já relativamente maduras e outras em processo de desenvolvimento e padronização. A IETF é uma organização aberta, na qual colaboram provedores Internet, fabricantes de equipamentos, universidades e outros interessados. Ela tem sido muito ágil nesse processo, dada a urgência criada pela necessidade. É importante avaliar cuidadosamente, então, se é preciso investir agora em equipamentos que suportam uma determinada técnica, para serem usados daqui a um ou dois anos, ou se é melhor esperar algum tempo até que tecnologias melhores e mais baratas, do ponto de vista financeiro e computacional, estejam mais maduras. Não é um problema para o qual se possa recomendar soluções em uma direção, ou outra, genericamente. Cada operador ou usuário na Internet deve analisar sua situação específica e escolher dentre as várias opções possíveis, a melhor para seu caso.

Um dos pontos a considerar na escolha das técnicas de transição a serem utilizadas é se elas são stateless ou stateful. Técnicas stateless são preferíveis, dado que escalam melhor e são mais baratas. Se necessário usar técnicas stateful, é preferível que estejam implantadas nos equipamentos dos usuários e não no provedor.

De forma geral, tanto as técnicas de tradução quanto as de túneis forçam a redução do MTU no escopo em que são usados na rede. Embora o presente texto não tenha abordado essa questão, todas elas apresentam mecanismos para contornar esse problema. Contudo, as técnicas baseadas em tradução aparentemente oferecem alguma vantagem, pois não encapsulam o pacote novamente, apenas traduzem e trocam os cabeçalhos na camada IP, o que poderia ser interpretado, grosso modo, como um túnel com compactação do cabeçalho.

O NAT444 é uma técnica a ser evitada. Seu uso não leva a rede do provedor em direção ao IPv6, acarreta altos custos financeiros e dificuldades técnicas para os usuários da Internet. O NAT444 fere o princípio da conexão fim a fim, que foi essencial para o desenvolvimento da Internet nos moldes em que a conhecemos atualmente e é uma das bases que a fazem ser um ambiente propício à inovação, novas idéias, aplicações, serviços e negócios. O compartilhamento de IPs com restrição de portas (A+P) e NAT44 são soluções preferíveis, adotadas em conjunto com túneis ou dupla tradução sobre redes nativas IPv6. NAT 64 em conjunto com DNS 64 e possivelmente ALGs também é uma solução preferível ao NAT444.

Para redes corporativas já existentes, o caminho mais indicado hoje é a implantação do IPv6 de forma gradual, em pilha dupla. Isso deve ser feito de forma urgente nos servidores expostos na Internet, como os servidores Web e de emails e de forma paulatina para os usuários e outros serviços. Ainda há problemas com aplicações utilizadas no dia a dia em relação ao suporte ao IPv6, por isso redes somente IPv6 ainda não podem ser recomendadas de forma genérica. Essa situação, no entanto, vem avançando rapidamente. É possível vislumbrar, já hoje, situações em que os benefícios trazidos pela facilidade de gerenciar uma rede com apenas um protocolo e endereços suficientes para todos os dispositivos, sem a necessidade de traduções, suplantem os problemas advindos da falta de suporte ao IPv6 em algumas aplicações. Pode ser que o futuro em que será vantajoso para as redes corporativas trabalhar apenas com IPv6, com auxílio de técnicas de transição para a comunicação com a Internet IPv4, não esteja muito distante.

Finalmente, deve-se considerar que todas as formas de compartilhamento do IPv4 para usuários geram dificuldades no processo de sua identificação à partir de ações executadas na Internet, caso isso seja necessário. Hoje os provedores Internet normalmente guardam registros, logs, que associam, univocamente, um usuário a um IPv4, dentro de um determinado período de tempo. É comum que em investigações criminais esses logs sejam requisitados, por meio de ordens judiciais, que trazem como dados o IP do usuário e o momento de acesso a um determinado site ou serviço na rede. Com o compartilhamento de IPs não haverá apenas um usuário associado a um IP num dado momento, mas sim diversos. Podem ser alguns poucos, dezenas, ou mesmo centenas. Esse será um problema temporário, resolvido facilmente com a migração dos principais serviços na Internet, como portais de conteúdo, de comércio eletrônico, serviços bancários ou de governo, para IPv6. Paliativamente, no contexto do compartilhamento do IPv4, pode-se propiciar uma possibilidade melhor de identificação, dependendo da técnica utilizada, se como informação for obtido não apenas o IP, mas também a porta de origem, à partir da qual a conexão foi realizada. Dessa forma, para aqueles serviços na Internet que hoje já guardam logs dos IPs de origem dos usuários, como bancos e serviços de comércio eletrônico, é recomendado que passem também a guardar as portas de origem, além de implantarem de forma urgente o IPv6.

20. Referências

- **RFC 1918** - Address Allocation for Private Internets - <http://tools.ietf.org/html/rfc1918>
- **RFC 2784** - Generic Routing Encapsulation (GRE) - <http://tools.ietf.org/html/rfc2784>
- **RFC 3053** - IPv6 Tunnel Broker - <http://tools.ietf.org/html/rfc3053>
- **RFC 3056** - Connection of IPv6 Domains via IPv4 Clouds - <http://tools.ietf.org/html/rfc3056>
- **RFC 3596** - DNS Extensions to Support IP Version 6 - <http://tools.ietf.org/html/rfc3596>
- **RFC 4213** - Basic Transition Mechanisms for IPv6 Hosts and Routers - <http://tools.ietf.org/html/rfc4213>
- **RFC 4291** - IP Version 6 Addressing Architecture - <http://tools.ietf.org/html/rfc4291>
- **RFC 4659** - BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN - <http://tools.ietf.org/html/rfc4659>
- **RFC 4798** - Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE) - <http://tools.ietf.org/html/rfc4798>
- **RFC 5211** - An Internet Transition Plan - <http://tools.ietf.org/html/rfc5211>
- **RFC 5214** - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) - <http://tools.ietf.org/html/rfc5214>
- **RFC 5572** - IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP) - <http://tools.ietf.org/html/rfc5572>
- **RFC 5569** - IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) - <http://tools.ietf.org/html/rfc5569>
- **RFC 6052** - IPv6 Addressing of IPv4/IPv6 Translators - <http://tools.ietf.org/html/rfc6052>
- **RFC 6144** - Framework for IPv4/IPv6 Translation - <http://tools.ietf.org/html/rfc6144>
- **RFC 6145** - IP/ICMP Translation Algorithm - <http://tools.ietf.org/html/rfc6145>
- **RFC 6146** - Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers - <http://tools.ietf.org/html/rfc6146>
- **RFC 6147** - DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers - <http://tools.ietf.org/html/rfc6147>
- **RFC 6219** - The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition - <http://tools.ietf.org/html/rfc6219>
- **RFC 6333** - Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion - <http://tools.ietf.org/html/rfc6333>
- **RFC 6346** - The Address plus Port (A+P) Approach to the IPv4 Address Shortage - <http://tools.ietf.org/html/rfc6346>
- **draft-bcx-behave-address-fmt-extension** - Extended IPv6 Addressing for Encoding Port Range - <http://tools.ietf.org/html/draft-bcx-behave-address-fmt-extension-01>
- **draft-despres-intarea-4rd-01** - IPv4 Residual Deployment across IPv6-Service networks (4rd) ISP-NAT's made optional - <http://tools.ietf.org/html/draft-despres-intarea-4rd-01>
- **draft-ietf-v6ops-464xlat-01** - 464XLAT: Combination of Stateful and Stateless Translation - <http://tools.ietf.org/html/draft-ietf-v6ops-464xlat-01>

- **[draft-ietf-v6ops-happy-eyeballs-07](http://tools.ietf.org/html/draft-ietf-v6ops-happy-eyeballs-07)** - Happy Eyeballs: Success with Dual-Stack Hosts - <http://tools.ietf.org/html/draft-ietf-v6ops-happy-eyeballs-07>
- **[draft-massar-v6ops-ayiya-02](http://tools.ietf.org/html/draft-massar-v6ops-ayiya-02)** - AYIYA: Anything In Anything - <http://tools.ietf.org/html/draft-massar-v6ops-ayiya-02>
- **[draft-shirasaki-nat44-05](http://tools.ietf.org/html/draft-shirasaki-nat44-05)** - NAT44 - <http://tools.ietf.org/html/draft-shirasaki-nat44-05>
- **[draft-weil-shared-transition-space-request-15](http://tools.ietf.org/html/draft-weil-shared-transition-space-request-15)** - IANA Reserved IPv4 Prefix for Shared Address Space - <http://tools.ietf.org/html/draft-weil-shared-transition-space-request-15>
- **[draft-xli-behave-divi-04](http://tools.ietf.org/html/draft-xli-behave-divi-04)** - dIVI: Dual-Stateless IPv4/IPv6 Translation - <http://tools.ietf.org/html/draft-xli-behave-divi-04>
- **[draft-xli-behave-divi-pd-01](http://tools.ietf.org/html/draft-xli-behave-divi-pd-01)** - dIVI-pd: Dual-Stateless IPv4/IPv6 Translation with Prefix Delegation - <http://tools.ietf.org/html/draft-xli-behave-divi-pd-01>
- <http://www.iana.org/assignments/ethernet-numbers>
- <http://tunnelbroker.net/>
- <http://www.sixxs.net/main/>
- <http://www.tunnelbroker.net/forums/index.php?topic=1016.0>
- <http://wiki.openwrt.org/doc/uci/network#dynamic.ipv6-in-ipv4.tunnel.he.net.only>
- <http://www.isc.org/software/aftr>
- <http://www.kangaroo.comcast.net/wiki/doku.php?id=wrt54gl:wrt54gl>
- <http://www.litech.org/tayga/>
- <http://sourceforge.net/projects/linuxnat64/>
- http://www.juniper.net/techpubs/en_US/junos10.4/information-products/topic-collections/nce/nat64-ipv6-ipv4-depletion/configuring-nat64-ipv6-ipv4-depletion.pdf
- <http://www.isc.org/software/bind>
- <http://www.dillema.net/software/totd.html>
- <http://code.google.com/p/android-clat/>
- <http://www.ivi2.org/IVI/>
- <http://bougaidenpa.org/masakazu/archives/176>
- <http://www.juniper.net/us/en/local/pdf/implementation-guides/8010078-en.pdf>
- <http://ecdysis.viagenie.ca/>