



## **IMPLEMENTAÇÃO DO PROTOCOLO IPV6 NA REDERIO**



**ALEXANDRE URTADO DE ASSIS**

**NITERÓI**

**2<sup>o</sup> SEMESTRE DE 2003**

**UNIVERSIDADE FEDERAL FLUMINENSE - UFF**

**ENGENHARIA DE TELECOMUNICAÇÕES**

**IMPLEMENTAÇÃO DO PROTOCOLO IPV6 NA REDERIO**

**Trabalho Monográfico de Término de Curso,  
apresentado por Alexandre Urtado de Assis,  
em cumprimento às exigências para  
conclusão do Curso de Engenharia de  
Telecomunicações, sob a orientação do  
Professor Alexandre dos Santos De la Vega  
e do Professor Nilton Alves Jr.**

**Niterói**

**2<sup>o</sup> Semestre de 2003**

## Resumo

O objetivo deste trabalho é divulgar as principais características do protocolo IPv6 – *Internet Protocol version 6*, que é a nova versão do protocolo IP, responsável pelo endereçamento na Internet. O trabalho envolve: as características e benefícios do IPv6; a formatação do novo protocolo, descrevendo o formato do datagrama e o cabeçalho; a arquitetura de seu endereçamento, descrevendo sua hierarquia e estrutura; os tipos de endereços existentes no IPv6; e algumas operações básicas do novo protocolo.

Este trabalho ainda se destina a apresentar alguns experimentos feitos em laboratório com a utilização do protocolo IPv6 e a implementação do IPv6 no *Backbone* metropolitano da Rederio.

# Índice

|   |           |
|---|-----------|
| <b>Resumo.....</b>  | <b>3</b>  |
| <b>Índice .....</b>   | <b>4</b>  |
| <b>Índice de Figuras .....</b>                                  | <b>6</b>  |
| <b>Índice de Tabelas.....</b>                                   | <b>7</b>  |
| <b>1. Introdução.....</b>                                       | <b>8</b>  |
| <b>2. Características e benefícios para o uso do IPv6 .....</b> | <b>14</b> |
| <b>3. Formatação do Protocolo IPv6 .....</b>                    | <b>18</b> |
| 3.1. Formato do Datagrama .....                                 | 18        |
| 3.2. Formato do cabeçalho base.....                             | 19        |
| 3.3. Cabeçalhos de extensão.....                                | 21        |
| <b>4. Arquitetura dos endereços IPv6 .....</b>                  | <b>29</b> |
| 4.1. Hierarquia dos endereços IPv6.....                         | 34        |
| 4.2. Endereçamento no 6Bone.....                                | 38        |
| 4.3. Endereços IPv6 de Produção.....                            | 40        |
| <b>5. Tipos de Endereços IPv6 .....</b>                         | <b>43</b> |
| 5.1. Endereços <i>Unicast</i> .....                             | 43        |
| 5.2. Endereços <i>Anycast</i> .....                             | 49        |
| 5.3. Endereço <i>Multicast</i> .....                            | 49        |
| <b>6. Operações Básicas.....</b>                                | <b>52</b> |
| 6.1. <i>ICMPv6 Packet</i> .....                                 | 52        |
| 6.2. <i>Neighbor Discovery Protocol</i> .....                   | 53        |
| 6.3. <i>Router Discovery</i> .....                              | 54        |
| 6.4. Autoconfiguração.....                                      | 55        |
| <b>7. Experimentos.....</b>                                     | <b>57</b> |

|  |           |
|--|-----------|
| 7.1. Experimento 1: Configuração do protocolo IPv6 em um segmento de rede .....  | 58        |
| 7.2. Experimento 2: Configuração do protocolo IPv6 em um segmento de rede com utilização do protocolo de roteamento RIPv6..... | 68        |
| 7.3. Experimento 3: Configuração do protocolo IPv6 em um segmento de rede, utilizando interface <i>Tunnel</i> .....            | 78        |
| <b>8. Implementação na Rede Rio .....</b>  | <b>91</b> |
| 8.1 O Projeto IPv6 .....   | 94        |
| <b>9. Conclusão .....</b>  | <b>97</b> |
| <b>10. Bibliografia .....</b>  | <b>99</b> |

# Índice de Figuras

|     |   |    |
|-----|---|----|
| 1.  | Datagrama IPv6 – com cabeçalhos de extensão.....                                  | 18 |
| 2.  | Datagrama IPv6 – sem cabeçalhos de extensão.....                                  | 18 |
| 3.  | Formato do cabeçalho Base IPv6. ....  | 19 |
| 4.  | Formato dos cabeçalhos de Extensão do IPv6.....                                   | 21 |
| 5.  | Formato dos cabeçalhos de Extensão Hop by Hop Options e Destination Options ..... | 22 |
| 6.  | Opções dos cabeçalhos de Extensão Hop by Hop Options e Destination Options .....  | 23 |
| 7.  | Formato do cabeçalho de Extensão Routing Header.....                              | 24 |
| 8.  | Formato do cabeçalho de Extensão Routing Header tipo 0 .....                      | 25 |
| 9.  | Formato do cabeçalho de Extensão Fragment Header .....                            | 26 |
| 10. | Formato do cabeçalho de Extensão Authentication Header.....                       | 27 |
| 11. | Estrutura do endereço IPv6.....   | 29 |
| 12. | Estrutura do endereço Aggregatable Global Unicast Address.....                    | 45 |
| 13. | Estrutura do endereço Site Local Unicast Address.....                             | 47 |
| 14. | Estrutura do endereço Site Local Unicast Address.....                             | 48 |
| 15. | Estrutura do endereço IPv6 compatible IPv4 Address.....                           | 48 |
| 16. | Estrutura do endereço Anycast.....  | 49 |
| 17. | Estrutura do endereço Anycast.....  | 50 |
| 18. | Estrutura do endereço Anycast.....  | 51 |
| 19. | Estrutura do Pacote ICMPv6.....   | 53 |
| 20. | Processo de Neighbor Discovery .....  | 54 |
| 21. | Processo de Router Discovery.....   | 55 |
| 22. | Processo de Autoconfiguração.....   | 56 |
| 23. | Mapa de Distribuição da Redeiro .....   | 92 |
| 24. | Estrutura física da Redeiro .....   | 93 |
| 25. | Atual estágio do Backbone Ipv6 da Redeiro.....                                    | 95 |

# Índice de Tabelas

|      |  |    |
|------|--|----|
| I.   | Modelo de referência ISO/OSI.....                                      | 8  |
| II.  | Comparação entre o modelo TCP/IP e o modelo de referência ISO/OSI..... | 9  |
| III. | Tabela de Alocação de endereços IPv6 .....                             | 33 |
| IV.  | Tabela de distribuição dos endereços IPv6 de Produção .....            | 41 |

# 1. Introdução

O mundo das comunicações está em constante movimento. Novas tecnologias são introduzidas e as antigas devem se adaptar ou tornam-se obsoletas. Quando surgiu a rede mundial Internet, no final dos anos 70, cada fabricante utilizava sua própria estrutura de protocolos e equipamentos. Dentro deste cenário de grande variedade de sistemas, aparece a necessidade de interconexão entre os diversos sistemas computacionais. Desta forma, em 1977, a ISO – *International Organization for Standardization*, criou um sub-comitê para o desenvolvimento de padrões de comunicação para promover a interoperabilidade entre as diversas plataformas. Foi então desenvolvido o modelo de referência OSI – *Open Systems Interconnection*, que é constituído por sete camadas, descritas na tabela I:

## I. Modelo de referência ISO/OSI

|   |              |  |
|---|--------------|--|
| 7 | APLICAÇÃO    | Esta camada funciona como uma interface de ligação entre os processos de comunicação de rede e as aplicações utilizadas pelo usuário.  |
| 6 | APRESENTAÇÃO | Aqui os dados são convertidos e garantidos em um formato universal.  |
| 5 | SEÇÃO        | Estabelece e encerra os enlaces de comunicação.  |
| 4 | TRANSPORTE   | Efetua os processos de sequenciamento e, em alguns casos, confirmação de recebimento dos pacotes de dados.   |
| 3 | REDE         | O roteamento dos dados através da rede é implementado aqui.  |
| 2 | ENLACE       | Aqui a informação é formatada em quadros. Um quadro representa a exata estrutura dos dados fisicamente transmitidos através do fio ou outro meio.                                  |
| 1 | FÍSICA       | Define a conexão física entre o sistema computacional e a rede. Especifica o conector, a pinagem, níveis de tensão, dimensões físicas, características mecânicas e elétricas, etc. |

Cada camada se comunica com sua semelhante em outro computador. Quando a informação é passada de uma camada para outra inferior, um cabeçalho é adicionado aos dados para indicar de onde a informação vem e para onde vai. O bloco de cabeçalho+dados de uma camada é o campo dados da próxima camada.

O modelo de camadas ISO/OSI acabou se tornando apenas uma base para praticamente todos os protocolos desenvolvidos pela indústria. A partir do modelo ISO/OSI foram desenvolvidos outros modelos para serem efetivamente usados, dentre eles o modelo TCP/IP (*Transmission Control Protocol / Internet Protocol*). A aceitação mundial do conjunto de protocolos TCP/IP deveu-se principalmente a versão UNIX de Berkeley que além de incluir estes protocolos, colocava-os em uma situação de domínio público, onde qualquer organização, através de sua equipe técnica poderia modificá-los e assim garantir seu desenvolvimento.

Cada desenvolvedor tem uma arquitetura que difere em detalhes, as vezes fundamental no seu desenvolvimento. Sendo assim, é de se esperar uma variação nas descrições do conjunto de protocolos TCP/IP. Apresentaremos na tabela II abaixo a comparação entre o modelo TCI/IP e o modelo base ISO/OSI:

## II. Comparação entre o modelo TCP/IP e o modelo de referência ISO/OSI

|                       |                    |              |
|-----------------------|--------------------|--------------|
| TELNET<br>FTP<br>SMTP | NFS<br>SNMP<br>DNS | APLICAÇÃO    |
|                       |                    | APRESENTAÇÃO |
| TCP                   | UDP                | SESSÃO       |
|                       |                    | TRANSPORTE   |
| IP                    |                    | REDE         |
| ENLACE                |                    | ENLACE       |
| FÍSICA                |                    | FÍSICA       |

O protocolo IP – *Internet Protocol* é o responsável pela conexão entre os sistemas que estão se comunicando. Basicamente este protocolo se relaciona com a camada de rede (3) do modelo ISO/OSI. Este protocolo é o responsável principal do movimento da informação na rede. É nesta camada/protocolo que a informação é fragmentada no sistema fonte e reagrupada no sistema alvo. Cada um destes fragmentos pode ter caminhos diferentes pela rede de forma que os fragmentos podem chegar fora

de ordem. Se, por exemplo, o fragmento posterior chegar antes do anterior, o protocolo IP no sistema destino reagrupa os pacotes na seqüência correta.

Desde que a primeira versão do protocolo IP foi desenvolvida, o poder de processamento das máquinas cresceu muito e o número de máquinas conectadas à rede cresceu de algumas centenas a 4 milhões. A versão 4 do IP foi a que conseguiu acomodar todas as mudanças da Internet e vem se tornando cada vez mais um padrão para redes de computadores, embora não tenha sido originalmente projetado para dar suporte a uma rede de escala universal ou que permitisse aplicações multimídia.

A versão corrente do IP tem sido extremamente bem-sucedida. O IP possibilitou que a Internet tratasse de redes heterogêneas, mudanças drásticas na tecnologia de hardware e grande crescimento do número de usuários. Para tratar de heterogeneidade, o IP define também um formato de pacote uniforme (o datagrama IP) e um mecanismo de transferência de pacote. Os datagramas IP são a unidade fundamental de comunicação na Internet. Ele também define um conjunto de endereços que permitem a aplicativos e protocolos de camadas mais altas se comunicarem através de redes heterogêneas sem conhecer as diferenças entre seus endereços de camadas inferiores.

Essa versão conviveu com várias mudanças de tecnologias de hardware. Embora tenha sido definido antes mesmo da popularização das Redes Locais, seu projeto original funciona bem através de gerações de tecnologias de hardware. O IP pode funcionar sobre redes que operam várias ordens de grandeza mais rápido do que foi projetado e também com tamanhos de quadros muito maiores. Se o protocolo IP tem uma história de sucesso tão grande, qual seria a motivação para substituí-lo.

A maior motivação para a mudança seria o espaço de endereçamento limitado. Quando a atual versão foi definida, existiam poucas redes de computadores e os projetistas não imaginaram que tal tecnologia pudesse tornar-se o padrão que é hoje.

Desta forma, decidiram usar 32 bits, o que permitiria que a Internet possuísse mais de um milhão de redes. Porém o crescimento foi muito maior que o esperado e o planejamento de distribuição de endereços foi mal feito, fazendo com que já a algum tempo a Internet convivesse com problemas para inclusão de novas redes. A necessidade de se criar alternativas começa a aparecer aqui.

Em 1991, membros do IETF - *Internet Engineering Task Force* chegaram à conclusão de que o crescimento exponencial da rede levaria à exaustão dos endereços IP até o final do ano de 1994. Isso aconteceria se as tabelas de roteamento simplesmente não esgotassem toda a capacidade dos *hardwares* de roteamento da época.

Essa crise foi superada a curto prazo com a adoção do CIDR - *Classless Inter-Domain Routing*, que consistia resumidamente em dar blocos de endereços IP contíguos a regiões do planeta (Europa, Ásia, etc), e essas regiões dividiriam seus blocos em blocos menores, mas ainda contíguos, até que todas as redes tivessem seus endereços. Essa subdivisão das classes de endereços é feita através da máscara de rede, que é um identificador do endereço de Internet. Essa máscara é capaz de identificar a fração do endereço referente a rede a que pertence e a fração que indica o *host*. Desta forma a estrutura de classes que até então existia pôde ser subdividida e melhor aproveitada.

Mas o CIDR não seria uma solução duradoura, outra deveria ser projetada em longo prazo e que tivesse uma duração maior. Um novo protocolo precisava ser desenvolvido em substituição ao IPv4. Uma proposta foi a adoção do CLNP – *Connection Less Network Protocol*, protocolo que tem um espaço de 160 bits para endereçamento. Entretanto, além de não suportar serviços multimídia como desejado, por ser uma solução OSI não foi bem quista por alguns elementos.

Em 1993, o IESG - *Internet Engineering Steering Group* criou um grupo de trabalho para uma nova versão do protocolo IP, o IPngWG - *IP Next Generation*

*Working Group*, com base em alguns objetivos que deveriam ser alcançados. O grupo de trabalho, então, selecionou três protocolos para a camada de rede da arquitetura TCP/IP. O protocolo indicado pelo grupo foi o SIPP - *Simple Internet Protocol Plus*, por ser o que menos se diferenciava do IPv4, e por ter um plano de transição melhor. Mas uma combinação de aspectos positivos dos três protocolos foi feita e com isso gerou-se a recomendação para a versão 6 do IP em novembro de 1994.

Apesar de atualmente utilizarmos a versão 4, existe uma explicação do porque do desenvolvimento da versão 6, e não da versão 5. A explicação é porque já existe a versão 5, e é conhecida como protocolo ST2 – *Straems 2* que foi definida pela RFC 1819. O ST2 é um protocolo experimental projetado para reserva de recursos destinados a oferecer garantias fim-a-fim em tempo real na Internet, ou seja, foi projetado para aplicações multimídia. Ele permite que os aplicativos criem fluxos de dados simples com muitos destinos e com a qualidade de serviço desejada. Ele não é um substituto do IP, mas apenas um adjunto.

A base do IPv6 é o IPv4, isto é, foi criado sobre uma plataforma comprovadamente eficaz, o que é importante tanto para a transição entre a versão 4 e a 6, quanto para a excelência do IPv6. Porém a transição para o IPv6 não ocorrerá rapidamente. Inclusive essa é uma estratégia da nova versão do protocolo, onde se espera uma co-existência das duas versões por muitos anos.

As motivações para esse trabalho apareceram por que o IPv6 passa a ser de grande importância para empresas, organizações e instituições que trabalham com serviços de Internet. Dentro deste contexto se encaixa a Rederio de Computadores, como *backbone* acadêmico metropolitano, que precisa estar preparada para operar o protocolo IPv6 e estar pronta para as eminentes transformações.

Neste trabalho será abordado a teoria do protocolo IPv6, suas características, benefícios, funcionalidades, diferenças entre o IPv4 e a implementação do IPv6 no *Backbone* metropolitano da Rederio, experimentos, configuração de equipamentos e serviços a serem disponibilizados.

## 2. Características e benefícios para o uso do IPv6

O protocolo IPv6 foi criado não só para resolver problemas da quantidade de endereços disponíveis, mas também para oferecer novos serviços e benefícios que não existiam no IPv4 ou que não eram utilizados de forma otimizada. Dentre muitos benefícios, podemos destacar os seguintes:

- Largo espaço de endereçamento para alcance global e escalabilidade;
- Formato de cabeçalho simplificado para otimização de entrega de pacote;
- Arquitetura hierárquica de rede para um roteamento eficiente;
- Suporte aos atuais protocolos de roteamento;
- Serviços de autoconfiguração;
- Implementação de IPSec de forma nativa;
- Crescimento do número de endereços *multicast*;
- Implementações para qualidade de serviço.

A disponibilidade de um número quase ilimitado de endereços IP é um dos maiores benefícios da implementação de redes IPv6. Comparado ao IPv4, o IPv6 aumenta o número de bits do endereço por um fator 4. Desta forma, o endereço que na versão 4 era de 32 bits, passa a ter 128 bits. Assim, esses 128 bits fornecem aproximadamente  $3,4 \times 10^{38}$  possíveis endereços, o que seria suficiente para alocar nos dias de hoje cerca de 1030 endereços por pessoa existente neste planeta. É claro que esses números são apenas informativos, porque com o IPv6 os equipamentos possuem não mais 1 só endereço, mas vários endereços destinados a serviços diferenciados. Isso ficará mais claro nas seções a seguir.

Essa grande quantidade de endereços possibilita que todos os equipamentos, dentre eles computadores, telefones IP, televisores digitais, possam ter endereços únicos globais, o que possibilitaria a alcançabilidade fim a fim de tais equipamentos, sem a necessidade de processamentos especiais.

Esse crescimento do número de bits do endereço IP resulta no crescimento do seu cabeçalho, porém o cabeçalho IPv6 é mais simplificado comparando-o ao do IPv4. Este último possui pelo menos 20 octetos, além do comprimento variável do campo opções. Já o cabeçalho IPv6 possui um tamanho fixo de 40 octetos, graças ao aumento do tamanho dos endereços de origem e destino. No entanto, possui menos campos. Um dos campos retirados foi o de controle de erro, já que este controle é feito na camada de enlace e na camada de transporte, sendo considerado desnecessário neste nível. Estas remoções resultam num processamento mais rápido do cabeçalho, o que aumenta a eficiência de roteamento e a performance geral dos roteadores.

A disponibilidade de um espaço de endereços e prefixos de rede muito grande fornece uma flexibilidade na arquitetura de redes que permite uma organização hierárquica e possivelmente geográfica, onde um prefixo de rede pode ser usado para endereçar um país ou um continente inteiro subdividido em seus diversos níveis.

Essa alocação permite que grandes provedores agreguem a seu prefixo de rede todos os endereços de seus usuários e anunciem para outros provedores apenas uma rota. Da mesma forma, o uso de múltiplos níveis hierárquicos dentro de um mesmo prefixo permite uma grande flexibilidade e novas funcionalidades, tal como a utilização do escopo dos endereços. A hierarquização da estrutura do endereçamento IPv6 é destinada a reduzir o tamanho das tabelas de roteamento.

Para habilitar um roteamento escalado, o IPv6 suporta a existência de protocolos de roteamento internos e externos. O protocolo RIP recebeu uma nova versão, chamado

RIPng – *Routing Information Protocol next generation*. Como visto na RFC 2080, o RIPng é similar ao RIPv2 e oferece os mesmos benefícios, sendo que o RIPng inclui suporte para endereços e prefixos IPv6.

O OSPF – *Open Shortest Path First* também ganhou uma nova versão, o OSPFv3. Este novo protocolo possui algumas mudanças em relação à versão utilizada para IPv4, que era extremamente dependente de tais endereços. O OSPFv3, como visto na RFC 2740, inclui uma plataforma independente de implementação e um protocolo para processamento por enlace ao invés de processamento por nó. Ainda existem mudanças na autenticação e no formato do pacote.

O protocolo BGP – *Border Gateway Protocol* funciona em IPv6 da mesma forma e oferece os mesmos benefícios que o BGP IPv4, incluindo ainda suporte para endereços de família IPv6 e atributos do próximo *hop* (próximo nó por onde o pacote passará). Esses atributos usam endereços IPv6 e endereços de escopo.

A característica de autoconfiguração de endereços existe no protocolo IPv6 para melhorar o gerenciamento de tais endereços e ainda facilitar a migração dos inúmeros equipamentos constituintes das redes do protocolo IPv4 para o protocolo IPv6. Essa característica habilita o desenvolvimento da Internet *plug-and-play* de novos dispositivos, tal como telefones celulares, dispositivos *wireless*, aparelhos domésticos e outros. Desta forma, os dispositivos conectados a rede não necessitariam de configuração manual ou de servidores de endereços. O funcionamento desta autoconfiguração será detalhado mais adiante.

Enquanto o uso de IPSec – *IP Security Protocol* é opcional em IPv4, no IPv6 ele torna-se obrigatório. Portanto, este serviço pode ser habilitado em todos os nós IPv6, o que potencialmente torna as redes mais seguras. A implementação de encriptação, autenticação e VPNs – *Virtual Private Networks* é feita de forma mais fácil, oferecendo

endereços globalmente únicos e seguros. O protocolo IPv6 pode ainda fornecer serviços de segurança fim-a-fim, tal como controle de acesso, confidencialidade, integridade de dados sem necessidade de *firewalls* adicionais, que podem provocar problemas de performance.

Uma característica muito importante do IPv6 é que ele não executa processos de *broadcast*. As funções em IPv4 que utilizavam processos *broadcast*, tais como descoberta de roteadores, descoberta de vizinhos, entre outros, em IPv6 são tratadas através de *multicast*.

O *multicast* permite que pacotes IP sejam enviados para múltiplos destinos ao mesmo tempo, sem afetar a performance da rede. O processo *multicast* melhora a eficiência de uma rede pela limitação de requisição *broadcast* para um menor número de nós, apenas aqueles interessados. O IPv6 utiliza grupos de endereços *multicast* específicos para várias funções, que em IPv4 eram feitas através de *broadcast*, evitando problemas causados por ele.

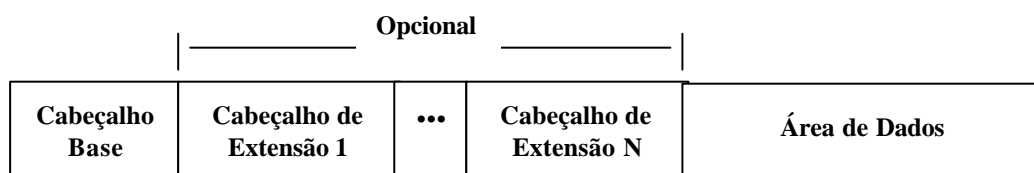
O serviço QoS – *Quality of Service* é tratado em IPv6 da mesma maneira que é tratado em IPv4, possuindo suporte por classe de serviço através do campo de tráfego e do modelo DiffServ - *Differentiated Services*. Entretanto, o cabeçalho IPv6 tem um novo campo chamado *flow label*, que pode conter um rótulo identificando um fluxo específico de dados. Desta forma, o nó fonte gera uma rota de fluxo com rótulo, disponibilizando QoS nesse caminho, onde cada roteador do caminho toma ações baseadas por esse rótulo.

### 3. Formatação do Protocolo IPv6

Nesta versão 6 do protocolo IP, foram feitas algumas mudanças no formato do Datagrama IPv6 e de seu cabeçalho. Também foram criados cabeçalhos de extensão independentes do cabeçalho base, característica que permite que o cabeçalho base possua tamanho fixo. Nesta seção veremos a estrutura do Datagrama IPv6, o formato de seu cabeçalho base e ainda as características de cada cabeçalho de extensão.

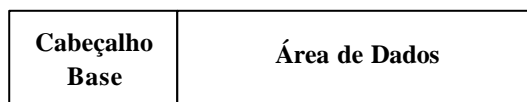
#### 3.1. Formato do Datagrama

A forma geral de um Datagrama IPv6 possui o cabeçalho base, seguido por zero ou mais cabeçalhos de extensão e por fim pelos dados.



1. Datagrama IPv6 – com cabeçalhos de extensão

Caso não haja nenhum cabeçalho de extensão, o cabeçalho base é diretamente seguido pela área de dados.

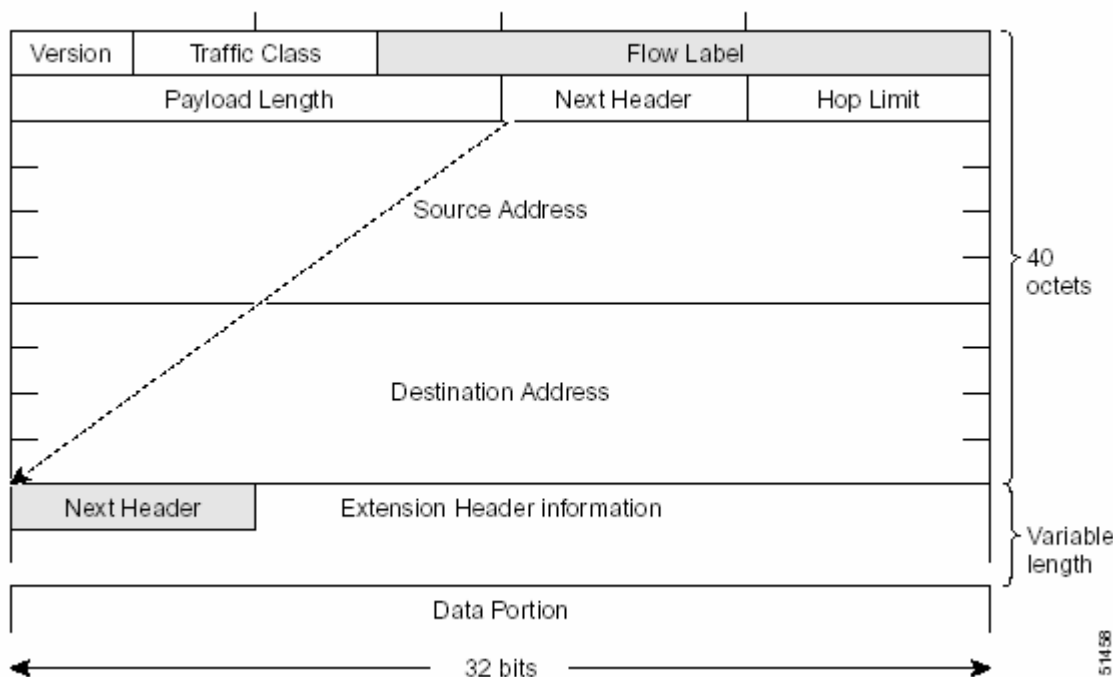


2. Datagrama IPv6 – sem cabeçalhos de extensão

As estruturas ilustradas acima não estão em escala. Em particular, os cabeçalhos de extensão não têm o mesmo tamanho e podem ser menores ou maiores que o cabeçalho base. Além disso, geralmente a área de dados é muito maior que o conjunto cabeçalho base e cabeçalhos de extensão.

### 3.2. Formato do cabeçalho base

O cabeçalho base do IPv6 é muito maior que o do IPv4, em decorrência do tamanho dos endereços de origem e de destino, porém, ele é mais simplificado que o cabeçalho IPv4, pois ele possui menos campos e informações. Esta simplificação o torna mais eficiente, pois ajuda a reduzir o processamento nos roteadores. Outra característica importante do cabeçalho base do IPv6 é que ele tem o tamanho fixo de 40 bytes, diferentemente do cabeçalho IPv4 onde seu tamanho não é fixo. Na figura 3 é mostrada a estrutura do cabeçalho Base IPv6.



3. Formato do cabeçalho Base IPv6.

O primeiro campo no Cabeçalho Base do IPv6 é o campo *VERSION* ou versão de protocolo. Este campo possui 4 bits assim como no IPv4 e contém o número 6 para indicar o protocolo IPv6, assim como o número 4 indica o IPv4. Existem outras opções para este campo que podem ser vistas na RFC 1770.

O segundo campo é o *TRAFFIC CLASS* ou prioridade. Este campo tem 8 bits e é similar ao campo *Type of Service* – ToS no IPv4. Esse campo classifica o pacote com

uma classe de serviço ou prioridade, que pode ser usado para diferenciar serviços. Sua funcionalidade é similar no IPv4 e no IPv6.

O campo *FLOW LABEL* não existia no IPv4. Este campo tem 20 bits e foi criado para marcar pacotes de um específico fluxo, com o objetivo de diferenciar esses pacotes na camada de rede. Portanto o campo *FLOW LABEL* habilita uma identificação de fluxo e um processo por fluxo em cada roteador no caminho do pacote. Com esse rótulo o roteador pode identificar o tipo de fluxo de cada pacote, sem que precise verificar sua aplicação. Esse campo permite que a diferenciação no tráfego seja feita na camada de rede, facilitando a prática de QoS – *Quality of Service*.

O campo *PAYLOAD LENGTH* é similar ao campo *TOTAL LENGTH* do IPv4. Ele tem 16 bits e indica o tamanho total da área de dados do pacote. No cabeçalho base IPv6 não existe o campo *HEADER LENGTH*, justamente por que ele tem tamanho fixo.

O campo *NEXT HEADER* é similar ao campo *PROTOCOL* do IPv4. Ele tem 8 bits e o valor deste campo indica o tipo de informação que segue o cabeçalho base do IPv6. Essa informação pode ser o protocolo usado na camada de transporte, UDP – *User Datagram Protocol* ou TCP – *Transmission Control Protocol*, ou um cabeçalho de extensão como mostra a figura 1.

O campo *HOP LIMIT* é similar ao campo TTL – *TIME TO LIVE* do IPv4. Ele tem 8 bits e o valor deste campo especifica o número máximo de roteadores (*hops*) que um pacote IPv6 pode passar antes de ser descartado. Em cada roteador esse valor é decrementado e por esta razão não existe *CHECKSUM* (código detector de erro do cabeçalho IP) no cabeçalho IPv6, para que não seja necessário que cada roteador recalcule o valor do *CHECKSUM* em cada pacote.

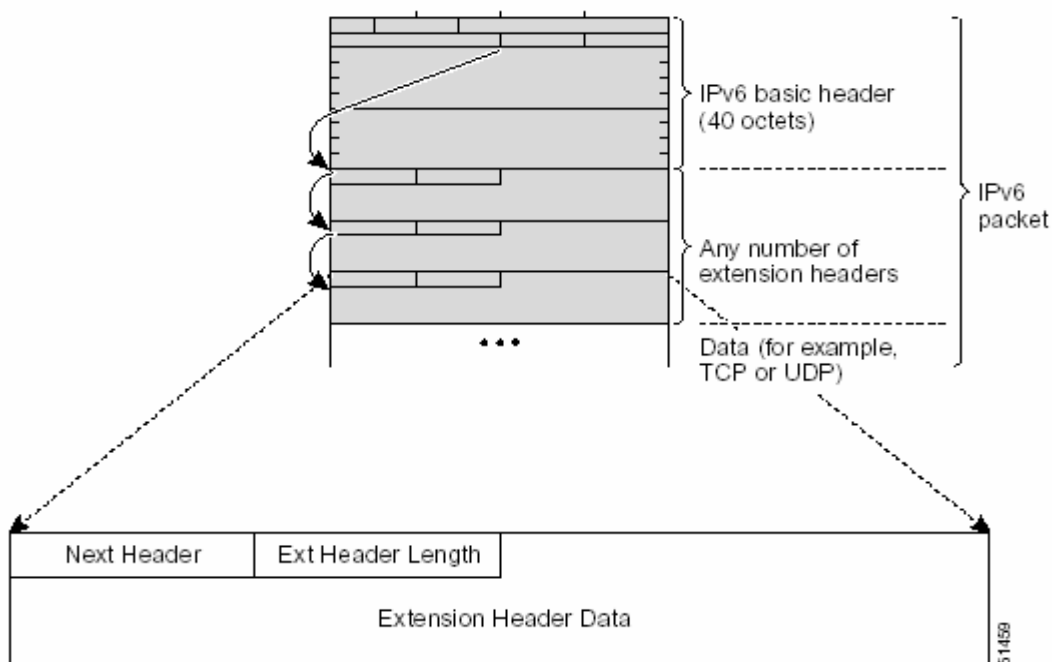
Os dois últimos campos *SOURCE ADDRESS* e *DESTINATION ADDRESS*, que indicam respectivamente endereços de origem e destino, são similares ao IPv4 a não ser

pelo tamanho destes campos. No IPv4 esses campos tinham 32 bits e agora devido ao aumento do tamanho dos endereços do Protocolo IP esses campos possuem 128 bits.

### 3.3. Cabeçalhos de extensão

Além do cabeçalho Base do IPv6, o datagrama IPv6 pode opcionalmente ser seguido por cabeçalhos de extensão, como mostra a figura 1. Não há um número fixo de cabeçalhos de extensão no datagrama IPv6. Diferentemente do cabeçalho base, os de extensão não tem tamanho fixo, podem variar de acordo com o tipo de cabeçalho de extensão ou um mesmo tipo de cabeçalho pode variar de tamanho. Para isso eles possuem o campo *EXTENSION HEADER LENGTH* que indica seu tamanho.

O outro campo fixo dos cabeçalhos de extensão é o *NEXT HEADER*, similar ao do cabeçalho base. Nele é identificado qual o tipo de cabeçalho que o seguirá, através de seu valor. Ao final do último cabeçalho de Extensão, o campo *NEXT HEADER* indica o protocolo da camada de transporte que é usado neste pacote. Na figura 4 é mostrada a estrutura dos cabeçalhos de extensão.



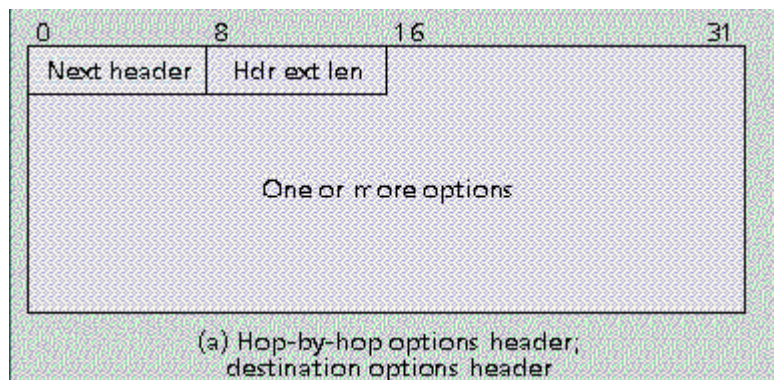
4. Formato dos cabeçalhos de Extensão do IPv6

Existem muitos tipos de cabeçalhos de extensão e cada um deles tem um valor associado. Quando são usados mais de um em um mesmo pacote, eles geralmente respeitam a ordem que segue, porém os nós estão preparados para receber em qualquer ordem:

1. *HOP BY HOP OPITIONS HEADER* (valor = 0): É usado para transportar informação opcional ou adicional que deve ser processada por todos os nós no caminho do pacote. Quando presente ele sempre vem em seguida do cabeçalho base do IPv6.

2. *DESTINATION OPTIONS HEADER* (valor = 60): Esse cabeçalho é usado para transportar informação opcional ou adicional que deve ser analisada somente pelo destino do pacote. Ele geralmente é usado seguindo o cabeçalho de extensão *HOP BY HOP OPITIONS HEADER*, sendo analisado somente pelo destino.

Os cabeçalhos de *HOP-BY-HOP OPTIONS* e *DESTINATION OPTIONS* têm o mesmo formato. Eles foram projetados para reunir várias informações isoladas e simples que não necessitam de mais um cabeçalho de extensão.



5. Formato dos cabeçalhos de Extensão Hop by Hop Options e Destination Options

A parte do cabeçalho que segue o campo *HEADER EXTENSION LENGHT* é dividida da seguinte forma, como mostra a figura 6:

|                    |                      |                     |
|--------------------|----------------------|---------------------|
| <b>8 bits</b>      | <b>8 bits</b>        | <b>n bits</b>       |
| <b><i>TYPE</i></b> | <b><i>LENGHT</i></b> | <b><i>VALUE</i></b> |

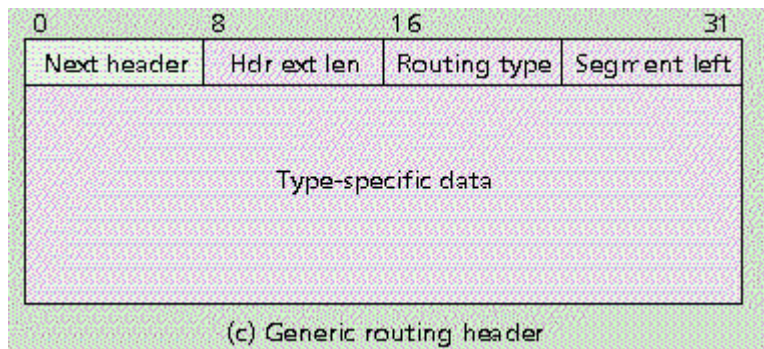
6. *Opções dos cabeçalhos de Extensão Hop by Hop Options e Destination Options*

O campo *TYPE* indica o tipo de opção. Caso essa opção contenha dados, o tamanho dos dados é indicado no campo *LENGHT* e os dados ficam no campo *VALUE*. Os 5 bits de mais baixa ordem em *TYPE* indicam a opção, enquanto o terceiro bit de mais alta ordem indica se os dados dessa opção podem mudar durante o trajeto do pacote. Caso essa opção não seja conhecida por algum nó durante o caminho do pacote, os dois bits de mais alta ordem indicam a ação a ser tomada, conforme é mostrado abaixo:

- 00 Ignore esta opção, continue o processamento dos cabeçalhos
- 01 Descarte o datagrama, mas não envie mensagem ICMP
- 10 Descarte o datagrama e envie mensagem ICMP para a origem
- 11 Descarte o datagrama e envie mensagem ICMP para a origem somente se o destino não for um endereço *multicast*

3. *ROUTING HEADER* (valor = 43): Esse cabeçalho é usado pela origem para listar um ou mais nós intermediários que devem ser visitados até o pacote chegar ao destino. Esse tipo de roteamento é utilizado quando existem mais de uma opção de caminho para os pacotes. Desta forma a origem pode traçar um caminho alternativo, mesmo que ele não seja o caminho indicado pelos protocolos de roteamento. Alternativamente esse cabeçalho de Extensão pode ser seguido por um outro do tipo *DESTINATION OPTIONS HEADER*. Neste caso esse cabeçalho de opções é processado por cada nó intermediário visitado.

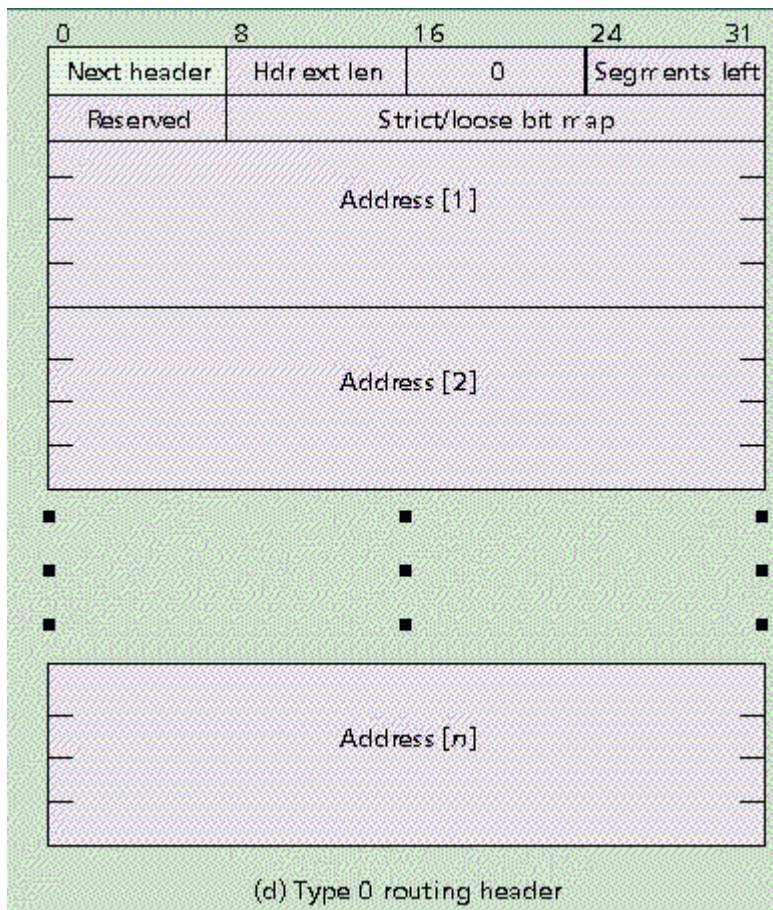
O cabeçalho de roteamento contém uma lista de um ou mais nós que devem ser visitados no caminho para o destino. Os cabeçalhos de roteamento sempre começam com um bloco de 32 bits divididos em 4 campos de 8 bits cada, como mostra a figura 7:



7. Formato do cabeçalho de Extensão Routing Header

O campo *NEXT HEADER* de 8 bits identifica o próximo cabeçalho. O campo *HEADER EXTENSION LENGTH* de 8 bits indica o tamanho do cabeçalho em unidades de 64 bits. O campo *ROUTING TYPE*, também de 8 bits, identifica um tipo de roteamento, caso esse tipo de roteamento não seja suportado por algum nó no caminho, o pacote deve ser descartado. *SEGMENTS LEFT* de 8 bits indica o número de nós intermediários, listados explicitamente, que devem ainda ser visitados antes da chegada do pacote ao destino.

O tipo mais comum de *ROUTING HEADER* é o zero, onde o campo *ROUTING TYPE* indica “0”. Neste caso além dos 32 bits do cabeçalho de roteamento, esse tipo 0 de cabeçalho de roteamento foi definido com mais 8 bits reservados e 24 bits de *STRICT/LOOSE BIT MAP*. Esses bits são numerados da esquerda para a direita, sendo que cada um corresponde a um *HOP*, indicando se o próximo destino deve ser um vizinho deste, 1 = *strict*, ou não, 0 = *loose*.



8. Formato do cabeçalho de Extensão Routing Header tipo 0

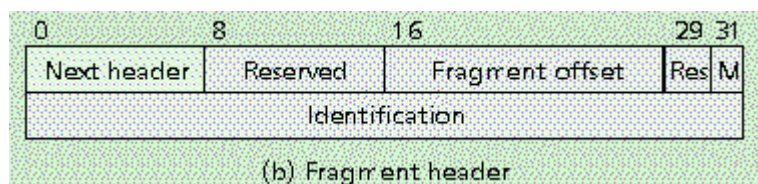
Quando se usa o roteamento de tipo 0, a origem não precisa informar separadamente o destino do datagrama, pois ele é considerado como sendo o último endereço listado no cabeçalho de roteamento, o campo *ADDRESS [N]* da figura 8, sendo que o cabeçalho base do IPv6 tem como destino o primeiro endereço listado no cabeçalho de roteamento. Até que esse nó seja atingido, o cabeçalho de roteamento não é examinado pelos roteadores do caminho. Quando o nó é alcançado o cabeçalho de roteamento é examinado e o próximo nó listado é colocado no cabeçalho base. O datagrama então é enviado com o campo *SEGMENTS LEFT* decrementado.

4. *FRAGMENT HEADER* (valor = 44): Esse cabeçalho é usado quando o pacote a ser enviado é maior que o MTU – *Maximum Transmission Unit* do caminho até

o destino. Neste caso é necessário que tal pacote seja fragmentado na origem, pois diferentemente do IPv4, nessa versão os roteadores não suportam fragmentação. Desta forma a origem divide o pacote em diversos fragmentos, sendo que cada fragmento possui um cabeçalho base seguido de pelo menos um de Extensão do tipo *FRAGMENT HEADER*. Esse cabeçalho de Extensão é somente processado no destino, onde eles são concatenados e transformados no pacote que os originaram.

Para esta operação, a origem realiza um *Path MTU discovery*, procedimento de descoberta do tamanho máximo de pacote que poderá trafegar entre a origem e o destino, a fim de identificá-lo. Assim, basta fragmentar o datagrama de tal modo que ele passe por todas as redes no caminho até seu destino.

Cada fragmento deve ser múltiplo de 8 octetos e cada *FRAGMENT HEADER* indica se existem outros fragmentos do mesmo dado ou não. A figura 9 mostra o esse cabeçalho.



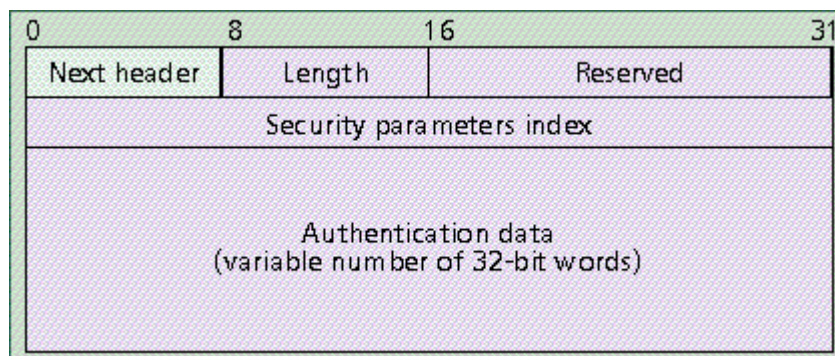
9. Formato do cabeçalho de Extensão Fragment Header

O campo *NEXT HEADER* de 8 bits indica o próximo cabeçalho. O campo *RESERVED* de 8 bits e o campo *RES* de 2 bits são reservados para o futuro. Já o campo *FRAGMENT OFFSET* de 13 bits indica a posição original deste fragmento no pacote original. O campo mais a direita é o *MFLAG* de 1 bit e indica se existem mais fragmentos, no caso afirmativo vale “1”, se é o último vale “0”. O campo *IDENTIFICATION* tem 32 bits e é a identificação do pacote original. Ela deve ser única em toda a Internet enquanto o pacote estiver trafegando.

Um problema gerado por esse tipo de fragmentação fim-a-fim, onde nós intermediários não podem fragmentar, é que se a rota mudar no meio da transmissão e o novo MTU for menor que aquele já descoberto, alguma coisa precisaria ser feita. O que acontece é que o datagrama IPv6 não é modificado, mas um datagrama novo é montado com o outro sendo encarado como dado. Assim, ele pode ser fragmentado e remontado fora da origem, isto é, em um nó entre a origem e o destino.

5. *AUTHENTICATION HEADER* (valor = 51): Esse cabeçalho é usado dentro do serviço IPsec - *IP Security Protocol* para prover autenticação e garantia de integridade aos pacotes IPv6. Esse cabeçalho é idêntico no IPv4 e no IPv6.

A autenticação é provida por um cabeçalho de extensão que suporta a integridade e autenticação dos dados de um pacote IP. Esse cabeçalho de extensão pode ser visto na figura 10.



10. Formato do cabeçalho de Extensão Authentication Header

O campo *NEXT HEADER* de 8 bits identifica o próximo cabeçalho. O campo *LENGTH* de 8 bits indica o tamanho do campo de dados em palavras de 32 bits. O campo *RESERVED* de 16 bits é reservado para uso futuro. O campo *SECURITY PARAMETERS INDEX* tem 32 bits e identifica uma associação de segurança. E o campo *AUTHENTICATION DATA* tem tamanho variável e contém os dados, em palavras de 32 bits.

O que o campo de dados representará vai depender do algoritmo de autenticação usado, mas no geral este campo é calculado com base em todo o datagrama, excluindo campos que mudem durante sua rota. No cálculo, esses campos são encarados como seqüências de bits “0”. Os cabeçalhos de fragmentação podem ser incluídos nesse cálculo.

6. *ENCAPSULATING SECURITY PAYLOAD HEADER* (valor = 50): Esse cabeçalho é também utilizado dentro do IPSec para providenciar autenticação e garantia de integridade aos pacotes IPv6. Da mesma forma esse cabeçalho é idêntico no IPv4 e no IPv6.

7. *IPV6 ENCRYPTION HEADER*: É usado para providenciar confidencialidade e integridade através da encriptação de dados.

8. *UPPER-LAYER HEADER*: Indica o protocolo da camada de transporte que será usado. Para um pacote com protocolo de camada de transporte TCP, o campo *NEXT HEADER* do último cabeçalho de extensão recebe o valor 6 e para o protocolo UDP recebe o valor 17.

## 4. Arquitetura dos endereços IPv6

Com a ampliação do tamanho do endereço de 32 bits para 128 bits foi resolvido um problema que é a exaustão dos endereços IP. Na versão 4 existem 4 Bilhões de endereços aproximadamente e na versão 6 temos aproximadamente  $3,4 \times 10^{38}$ , milhares de bilhões de endereços globais disponíveis.

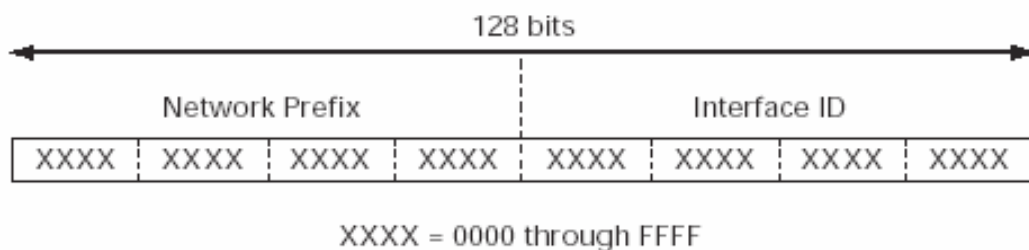
No IPv4 o endereço consiste de quatro grupos de 8 bits, sendo cada grupo representado por um número decimal que varia de 0 a 255. Esses números eram separados por pontos, como no exemplo:

**DECIMAL => 152.84.253.41**

**OU**

**BINÁRIO => 10011000.01010100.1111101.00101001**

Porém o espaço de endereço IPv6 é um número muito grande e difícil de se representar ( $2^{128}$ ). Desta forma, segundo a RFC 2373, o endereço IPv6 não é mais representado por números decimais. Nesta nova versão os 128 bits são divididos em 8 grupos de 16 bits representados por números hexadecimais, que variam de 0000 até FFFF e são separados por dois pontos “:”. Na figura 11 podemos visualizar a estrutura do endereço IPv6.



$3.4 \times 10^{38} = 340,282,366,920,938,463,374,607,432,768,211,456$  IPv6 Addresses

A representação do endereço IPv6 pode ser feita por três formas diferentes. A apresentada anteriormente é a mais usual. Abaixo um exemplo numérico desta forma:

**2001:0C00:0000:0000:5400:0000:0000:0009**

Os zeros à esquerda de cada grupo podem ser retirados, por exemplo, em **0C00** podemos simplificar para **C00**. Porém os zeros à direita não podem ser desprezados, pois, por exemplo, **5400** é diferente de **54**, porque **54** representa **0054**, logo ocasionaria um erro na representação deste endereço. Abaixo vemos a representação correta de nosso exemplo:

**2001:C00:0:0:5400:0:0:9**

Existe também uma forma abreviada de representar um endereço IPv6. Essa abreviação foi criada por que no início da utilização do IPv6 deveremos ter muitos campos apenas com zeros, desta forma podemos substituir essa seqüência de zeros por “::”, como no exemplo:

**2001:C00::5400:0:0:9**

**ou**

**2001:C00:0:0:5400::9**

Porém essa abreviação só pode ser feita uma vez. Caso contrário seria impossível diferenciar certos endereço. Por exemplo, no caso anterior poderíamos fazer o seguinte:

**2001:C00::5400::9**

Mas nesse caso seria impossível saber qual é o endereço, pois não saberemos quantos zeros existem na primeira abreviação ou na segunda. Observe que neste exemplo poderíamos representar o mesmo endereço de 3 formas diferentes:

**2001:C00:0:0:5400:0:0:9**

**ou**

**2001:C00:0:0:0:5400:0:9**

**ou**

**2001:C00:0:5400:0:0:0:9**

Por isso é importante que essa abreviação seja feita só uma vez em cada endereço. Essa abreviação pode ser feita não só no meio do endereço, mais também no início ou no fim, como mostra o exemplo com Tipos de endereços que serem abordados na próxima seção:

**Endereço *Loopback* => 0:0:0:0:0:0:0:1 Abreviando => ::1**  
**Endereço *Unspecifield* => 0:0:0:0:0:0:0:0 Abreviando => ::**

Existe ainda uma terceira forma para representar, que é mais conveniente para ambientes mistos com IPv4 e IPv6. Essa forma é a seguinte:

**X:X:X:X:X.X.d.d.d.d**

Onde os "X" são os grupos de 4 números hexadecimais e os "d" são valores decimais de 8 bits que variam de 0 a 255, como na notação do IPv4. Essa representação é usada em tipos de endereços que também serão abordados na próxima seção.

**0:0:0:0:0:0:152.84.253.35 abreviando => ::152.84.253.35**  
**0:0:0:0:0:FFFF:152.84.253.35 abreviando => ::FFFF:152.84.253.35**

Há também uma outra representação que se refere ao que conhecemos como endereços de rede. Essa representação é da forma endereço/prefixo. Onde endereço corresponde a sub-rede a qual o endereço pertence, completada por zeros, e o prefixo é a quantidade de bits deste endereço referente a sub-rede. Esta notação é similar à notação CIDR do IPv4.

Por exemplo, o endereço de sub-rede 200100000004CFE em hexadecimal, possui 60 bits de prefixo e pode ser representado das seguintes formas:

**2001:0000:0004:CFE0:0000:0000:0000/60**

Retirando os zeros a esquerda:

**2001:0:4:CFE0:0:0:0:0/60**

Abreviando:

**2001:0:4:CFE0::/60**

Essas notações e inclusive suas formas de abreviações são usadas tanto para indicar endereços IPv6 quanto para configurar equipamentos como roteadores e estações de trabalho.

De todo o espaço de endereçamento do IPv6,  $3,4 \times 10^{38}$  endereços, apenas 15% está previamente alocado para uso. Os outros 85% restantes foram reservados para o futuro.

Na tabela III os tipos de endereço do IPv6 são classificados através de um FP – *Format Prefix*. Esse prefixo é definido pelos primeiros bits de cada endereço, sendo que dependendo do endereço esse prefixo tem tamanho diferente. A tabela III também mostra a fração do endereço ocupada por seu respectivo prefixo.

Abaixo é apresentada a tabela III que relaciona o tipo de endereço com seu prefixo FP em binários.

### III. Tabela de Alocação de endereços IPv6

| <b>Alocação</b>                            | <b>Prefixo (binário)</b> | <b>Fração do Espaço de Endereçamento</b> |
|--|--------------------------|--|
| Reservado                                  | 0000 0000                | 1/256                                    |
| Não Alocado                                | 0000 0001                | 1/256                                    |
|  |                          |  |
| Reservado para Alocação NSAP               | 0000 001                 | 1/128                                    |
| Reservado para Alocação IPX                | 0000 010                 | 1/128                                    |
|  |                          |  |
| Não Alocado                                | 0000                     | 1/128                                    |
| Não Alocado                                | 0000                     | 1/32                                     |
| Não Alocado                                | 0001                     | 1/16                                     |
|  |                          |  |
| <i>Aggregatable Global Unicast Address</i> | 001                      | 1/8                                      |
| Não Alocado                                | 010                      | 1/8                                      |
| Não Alocado                                | 011                      | 1/8                                      |
| Não Alocado                                | 100                      | 1/8                                      |
| Não Alocado                                | 101                      | 1/8                                      |
| Não Alocado                                | 110                      | 1/8                                      |
|  |                          |  |
| Não Alocado                                | 1110                     | 1/16                                     |
| Não Alocado                                | 1111 0                   | 1/32                                     |
| Não Alocado                                | 1111 10                  | 1/64                                     |
| Não Alocado                                | 1111 110                 | 1/128                                    |
| Não Alocado                                | 1111 1110 0              | 1/512                                    |
|  |                          |  |
| <i>Site-local Unicast Address</i>          | 1111 1110 10             | 1/1024                                   |
| <i>Link-local Unicast Address</i>          | 1111 1110 11             | 1/1024                                   |
| <i>Multicast Address</i>                   | 1111 1111                | 1/256                                    |

## 4.1. Hierarquia dos endereços IPv6

Os endereços IPv6 denominados *Unicast*, que veremos com mais detalhes na próxima seção, foram projetados para sistemas de roteamento da Internet que repassam pacotes baseado num algoritmo de cálculo do prefixo mais longo, sem nenhum conhecimento da estrutura interna do endereço IPv6. Esse tipo específico de endereço IPv6 é indicado pelos primeiros bits do endereço, como vimos na tabela III - Alocações de endereços IPv6.

Dentre os tipos de endereços *Unicast* apresentados na tabela III, temos os endereços *Aggregatable Global Unicast Addresses* a serem globalmente utilizados na Internet e definidos pelo formato de prefixo FP = 001. Esses endereços foram criados para suportar a agregação *provider-based*, onde os endereços possuem uma hierarquia definida por seus provedores de acesso à rede, isto é, cada rede que possui um provedor de acesso terá seu endereço base idêntico ao de seu provedor, acrescido de mais um nível dentro da hierarquia.

Outro tipo de agregação é a denominada *exchange-based*, necessária para os pontos de troca de tráfego, conhecidos como *exchanges*, redes que não possuem provedores. São os grandes provedores de distribuição do mundo, que serão os pontos mais altos da hierarquia. Esta combinação permitirá uma agregação eficiente de rotas, tanto para sitios conectados a provedores, quanto para os pontos de troca de tráfego. Para esta estrutura hierárquica existem 4 níveis:

- TLA ID - Identificador *Top-Level Aggregation*;
- NLA ID - Identificador *Next-Level Aggregation*;
- SLA ID - Identificador *Site-Level Aggregation*;
- Interface ID - Identificador de Interface.

Em termos de topologia, essa estrutura permite uma organização em três níveis hierárquicos: pública, sítios e identificador de interface. A topologia pública abrange os campos TLA e NLA e corresponde ao conjunto de provedores de serviços Internet, provedores de trânsito e pontos de troca de tráfego. A topologia sítio, do campo SLA, tem abrangência local, ou seja, uma organização específica que não provê serviços de trânsito para outras organizações ou sítios, Já o identificador de interface, como o próprio nome indica, identifica a interface do nó indicada pelo campo *Interface ID*.

### ***Top-Level Aggregation ID***

Os identificadores TLA são o topo da hierarquia de roteamento. Este formato suporta 8.192 ou  $2^{13}$  identificadores TLA. Esse campo pode ser aumentado através do de um espaço previamente reservado contido em endereços do tipo *UNICAST*, ou utilizando um prefixo de formato adicional.

Os roteadores devem ter uma entrada na tabela de roteamento para cada TLA ID ativo, e podem ter entradas adicionais para otimizar o roteamento de suas topologias específicas. Mas, em todos os níveis, a topologia de roteamento deve ser projetada para minimizar a quantidade de entradas na tabela de roteamento.

### ***Next-Level Aggregation ID***

Os identificadores NLA são utilizados pelas organizações que possuam um TLA ID para criar uma estrutura de endereçamento hierárquica e identificar *sites*. Cada organização que recebe um TLA ID tem um espaço de endereçamento de 24 bits para o campo NLA, ou seja, 16.777.216 ou  $2^{24}$  endereços. O que significa que cada organização com de nível TLA possua aproximadamente a mesma quantidade de endereços que toda atual Internet - IPv4 pode suportar. Desta forma, uma distribuição plana de todo espaço NLA acarretaria uma tabela de rotas com aproximadamente 16

milhões de entradas. Daí a importância de se hierarquizar o endereçamento para minimizar a tabela de rotas e otimizar o roteamento.

As organizações possuidoras de um TLA ID podem suportar NLA IDs no seu próprio espaço Site ID, o que possibilita o provimento de serviços a outras organizações provedoras ou não de serviço público de trânsito. Por sua vez, as organizações possuidoras de um NLA ID podem usar o espaço Site ID para suportar outros NLA IDs, como mostrado abaixo:

|      |             |        |              |
|------|-------------|--------|--------------|
| n    | (24-n) bits | 16     | 64 bits      |
| NLA1 | Site ID     | SLA ID | Interface ID |

|       |               |        |              |
|-------|---------------|--------|--------------|
| m     | (24-n-m) bits | 16     | 64 bits      |
| NLA21 | Site ID       | SLA ID | Interface ID |

|      |                 |        |              |
|------|-----------------|--------|--------------|
| o    | (24-n-m-o) bits | 16     | 64 bits      |
| NLA3 | Site ID         | SLA ID | Interface ID |

O esquema acima leva a uma distribuição hierárquica, onde a responsabilidade para definição e alocação do espaço NLA é do possuidor do TLA, a responsabilidade do espaço NLA1 é do possuidor do NLA, a do NLA2 é do possuidor do NLA1 e assim por diante. Na alocação do espaço NLA há uma troca entre a eficiência da agregação do roteamento e a flexibilidade. Uma estrutura hierárquica permite uma maior agregação de rotas e, por conseguinte, uma diminuição das entradas das tabelas de rotas, com otimização do roteamento. Já uma estrutura plana de distribuição de endereços NLA facilita a alocação de endereços, mas resulta em grandes tabelas de rotas.

Este exemplo deixa claro que os espaços previamente definidos para cada nível de hierarquização, são passíveis de alterações. Porém a estrutura de hierarquização deve ser sempre respeitada para a otimização do roteamento.

## Site-Level Aggregation ID

O identificador SLA é utilizado por uma organização individual, que é responsável por definir a estrutura de endereços do espaço SLA. Dentro deste espaço, a organização pode criar localmente sua própria estrutura de endereçamento hierárquica, num procedimento similar a divisão em sub-redes do IPv4, só que com um número muito maior de sub-redes.

A exemplo do esquema apresentado no NLA, a organização possuidora do SLA pode decidir utilizar uma estrutura plana, aumentando as tabelas de rotas, ou definir uma estrutura hierárquica que seria da forma:

|      |             |        |              |
|------|-------------|--------|--------------|
| n    | (16-n) bits | 16     | 64 bits      |
| SLA1 | Subnet      | SLA ID | Interface ID |

|      |               |        |              |
|------|---------------|--------|--------------|
| m    | (16-n-m) bits | 16     | 64 bits      |
| SLA2 | Subnet        | SLA ID | Interface ID |

## Interface ID

Os identificadores de interface ou *Interface ID*, como o próprio nome indica, são utilizados para identificar interfaces de um enlace específico e devem ser únicos nesse enlace. Também devem ser únicos num escopo mais abrangente. Em muitos casos, o identificador de interface será o endereço de interface da camada de enlace (*MAC Address*) ou obtido a partir deste.

Para os endereços *AGGREGATABLE GLOBAL UNICAST*, os identificadores de interface de 64 bits devem ser construídos no formato IEEE EUI-64. Estes identificadores podem ter um escopo global quando formados a partir de registros de escopo global, como é o caso dos endereços MAC de 48 bits definidos pelo IEEE; ou

um escopo local quando não existirem tais registros. É o caso das conexões seriais ponto-a-ponto. Para cada RFC que define o protocolo IPv6 sobre algum enlace específico, como IPv6 sobre *Ethernet* ou IPv6 sobre FDDI, há procedimentos para formação do *Interface ID*. Na prática para IPv6 sobre *Ethernet*, que são os casos mais comuns na Internet, existe uma combinação entre o endereço MAC e alguns algoritmos característicos. Abaixo temos um exemplo:

Para um endereço MAC:

**00:A0:C9:C8:E0:C2**

E um prefixo de rede :

**2001::/16**

Seria obtida a seguinte *Interface ID*:

**02A0:C9FF:FEC8:E0C2**

E o endereço completo seria:

**2001:: 02A0:C9FF:FEC8:E0C2/128**

## **4.2. Endereçamento no 6Bone**

Para que fossem realizados testes e também para o desenvolvimento do protocolo IPv6 o IETF – *Internet Engineering Task Force*, criou um *Backbone* de teste chamado 6Bone. Esse projeto inicialmente operava como uma rede virtual interligada por túneis IPv6 sobre IPv4. Hoje com o desenvolvimento e crescimento do projeto, o 6Bone está migrando para um *Backbone* com IPv6 nativo.

Com o propósito de utilizar o mínimo do espaço de endereçamento de produção IPv6, o IANA – *Internet Assigned Numbers Authority* alocou ao Projeto 6Bone o prefixo TLA 3FFE::/16. Esse endereço é especificado na RFC 2471 - *IPv6 Testing Address Allocation*.

Sob esse prefixo, o 6Bone criou um formato próprio para o particionamento de seu espaço de endereços. Esse formato é baseado no utilizado em endereços de produção, especificado no RFC 2374 - *An IPv6 Aggregatable Global Unicast Address Format*.

Os prefixos TLA e NLA do formato de produção são simulados dentro da faixa de endereços do 6Bone. Eles são chamados *pseudo Top-Level Aggregation Identifier* e *pseudo Next-Level Aggregation Identifier*, pTLA e pNLA respectivamente.

O formato de endereços utilizado é representado abaixo:

| FP  | TLA    | 8 bits | 24 bits | 16 bits | 64 bits      |
|-----|--------|--------|---------|---------|--------------|
| 001 | 0x1FFE | pTLA   | pNLA    | SLA ID  | Interface ID |

- FP            **FORMAT PREFIX**  
Identifica o tipo de endereço IPv6. Os bits 001 identificam endereços *unicast Aggregatable Global*.
- TLA ID        **TOP-LEVEL AGGREGATION IDENTIFIER**  
Prefixo do topo da hierarquia de roteamento. O TLA ID 0x1FFE é o identificador atribuído pela IANA ao 6Bone.
- pTLA ID      **PSEUDO TOP-LEVEL AGGREGATION IDENTIFIER**  
Prefixo alocado pelo 6BONE aos participantes do projeto. Define o nível máximo de agregação dentro do 6Bone. Equivale a um identificador de *Backbone*.
- pNLA ID      **PSEUDO NEXT-LEVEL AGGREGATION IDENTIFIER**  
Utilizado por organizações detentoras de um pTLA ID para criar uma hierarquia de endereçamento e identificar *sites*.
- SLA ID        **SITE-LEVEL AGGREGATION IDENTIFIER**  
Utilizado por organizações individuais para criar sua própria hierarquia de endereçamento e para identificar sub-redes.

O tamanho do campo pTLA ID original permitia ao 6Bone atribuir identificadores para até 256 *Backbones*. No início de 1999, devido a grande expansão do 6Bone, decidiu-se aumentar o campo pTLA para que este pudesse acomodar um maior número de redes. Existem hoje, dois formatos para os campos pTLA e pNLA utilizados no 6BONE. A única diferença entre esses dois formatos é a quantidade de bits utilizados por cada um desses dois campos.

No formato original, o campo pTLA possui 8 bits e o campo pNLA 24 bits, como ilustrado no diagrama acima. A notação dos prefixos pTLA fica, então, 3FFE:nn00::/24, onde "nn" representa o campo pTLA.

O novo formato utiliza os primeiros 4 bits do campo pNLA, que diminui para 20 bits, e os acrescenta ao campo pTLA, que aumenta para 12 bits. A nova notação dos prefixos pTLA fica, então, 3FFE:nnn0::/28, onde "nnn" representa o novo campo pTLA. Para evitar conflitos com os pTLAs já atribuídos, o valor "nnn" começa a partir de 0x800.

### **4.3. Endereços IPv6 de Produção**

Atualmente, já estão sendo oferecidos endereços IPv6 de produção pelos quatro *Regional Internet Registries* (RIR): ARIN – *American Registry for Internet Numbers*, responsável pela América do Norte e África sub-Saara; RIPE NCC – *Réseaux IP Européens*, responsável pela Europa, Oriente Médio, Ásia Central e Norte da África; LACNIC - *Latin American and Caribbean IP Address Registry*, responsável pela América Latina e Caribe; e APNIC – *Asia Pacific Network Information Centre*, responsável pela Ásia.

Inicialmente, foi reservado pelo IANA o prefixo TLA 2001::/16 para endereçamento de produção. Sendo distribuído para os RIR's os seguintes prefixos, apresentados na tabela IV:

#### IV. Tabela de distribuição dos endereços IPv6 de Produção

| Prefixo IPv6   | Sub-TLA - Valores Binários | Alocado por         | Data   |
|----------------|----------------------------|---------------------|--------|
| 2001:0000::/23 | 0000 000X XXXX X           | IANA                | jul/99 |
| 2001:0200::/23 | 0000 001X XXXX X           | APNIC               | jul/99 |
| 2001:0400::/23 | 0000 010X XXXX X           | ARIN                | jul/99 |
| 2001:0600::/23 | 0000 011X XXXX X           | RIPE NCC            | jul/99 |
| 2001:0800::/23 | 0000 100X XXXX X           | RIPE NCC            | May 02 |
| 2001:0A00::/23 | 0000 101X XXXX X           | RIPE NCC            | nov/02 |
| 2001:0C00::/23 | 0000 110X XXXX X           | APNIC               | May 02 |
| 2001:0E00::/23 | 0000 111X XXXX X           | APNIC               | jan/03 |
| 2001:1000::/23 | 0001 000X XXXX X           | (future assignment) |        |
| 2001:1200::/23 | 0001 001X XXXX X           | LACNIC              | nov/02 |
| 2001:1400::/23 | 0001 010X XXXX X           | RIPE NCC            | Feb 03 |
| 2001:1600::/23 | 0001 011X XXXX X           | RIPE NCC            | jul/03 |
| 2001:1800::/23 | 0001 100X XXXX X           | ARIN                | Apr 03 |

Através desses prefixos, são alocados os identificadores para *Backbones*, utilizando para isso o campo Sub-TLA. O tamanho do prefixo mínimo alocado é de 32 bits.

O processo de alocação utiliza um procedimento chamado *slow start*. Ao se obter um identificador Sub-TLA, os 6 bits seguintes são reservados pelo RIR que fez a alocação. O RIR só fará abcações subseqüentes desse espaço reservado quando a organização tiver utilizado pelo menos 80% do espaço previamente alocado.

Cada organização que recebe um prefixo Sub-TLA é responsável pelas alocações dentro de seus clientes ou associados. Para uma organização obter um prefixo Sub-TLA é necessário que lá seja um AS – *Autonomo System*, ou se torne um AS. Seguindo então um processo de hierarquização, todos os clientes desta organização

deverão receber um prefixo do tipo SLA de 48 bits, agregado ao prefixo Sub-TLA de seu provedor.

No caso da América Latina e por sua vez do Brasil, o órgão de registro responsável pelas alocações dos prefixos Sub-TLA é o LACNIC. Sendo então todos os endereços IPv6 da América Latina agregados ao um mesmo prefixo de rede.

## 5. Tipos de Endereços IPv6

Segundo a RFC 2374, uma mesma interface, que utiliza o protocolo IPv6, pode utilizar mais de um endereço, diferentemente do IPv4, onde tal característica só era possível em roteadores. Essa característica é importante porque na versão 6 algumas aplicações, em geral de controle, utilizam-se de endereços especiais que veremos adiante. Para o endereçamento das interfaces existem então 3 tipos de endereços:

- *Unicast*;
- *Anycast*;
- *Multicast*.

Outra característica marcante do IPv6 é que não existem mais os endereços *broadcast*, que endereçavam todos os *hosts* de um mesmo domínio de colisão, isto é, uma pacote com endereço de destino do tipo *broadcast* era enviado para todos os *hosts* de seu domínio de colisão. Com a abolição desse tipo endereço, outro protocolo muito comum no IPv4 também ficou em desuso, o ARP – *Address Resolution Protocol*, que usava endereços *broadcast* para descoberta do endereço MAC da interface referente ao endereço de destino do pacote.

### 5.1. Endereços *Unicast*

Esse tipo de endereço é comumente usado em IPv4, que identifica apenas uma única interface. Desta forma um pacote destinado a um endereço do tipo *Unicast* é enviado diretamente para a interface associada a esse endereço.

Foram definidos pela RFC 2374 vários tipos de endereços *Unicast*:

- *Agregatable Global Unicast Address*

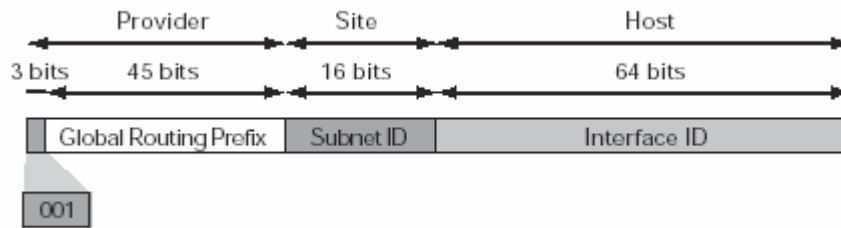
- *Loopback Address*
- *Unspecified Address*
- *NSAP Address*
- *IPX Address*
- *Site-local Unicast Address*
- *Link-local Unicast Address*
- *IPv4-compatible IPv6 Address*

## **Agregatable Global Unicast Address**

Esse tipo de endereço *unicast* é equivalente ao endereço global *unicast* usado em IPv4. Sendo assim é o endereço que será usado globalmente na Internet. Essa estrutura de endereços globais permite uma agregação de prefixos de roteamento que limitam o número de entradas nas tabelas de rotas.

A estrutura deste tipo de endereço é dividida em 4 níveis, o primeiro é o FP – *Format Prefix*, que indica justamente que se trata de um endereço do tipo *Global Unicast*, esse FP deve ser sempre 001, como vimos na tabela III - Alocação de endereços IPv6, na seção anterior.

O segundo campo é chamado *Global Routing Prefix*, e é destinado a identificação dos ISP's – *Internet Service Provider*, ele possui vários níveis e seguem a estrutura apresentada na seção anterior. O terceiro campo *Subnet ID* também foi apresentado anteriormente como sendo o campo *Site ID* da estrutura de hierarquização do endereço IPv6, o último nível é o *Interface ID*, que também já foi abordado e possui 64 bits. Abaixo, vemos na figura 12 a estrutura desse tipo de endereço:



12. Estrutura do endereço Aggregatable Global Unicast Address

## Loopback Address

Esse tipo de endereço, como o próprio nome já diz, é o endereço da própria interface. Porém ele só pode ser usado quando um nó envia um pacote para ele mesmo. No IPv4 esse tipo de endereço era geralmente o 127.0.0.1, em IPv6 é indicado por:

**0:0:0:0:0:0:0:1**

ou simplesmente:

**::1**

Esse endereço não pode ser associado a nenhuma interface física, nem como endereço de fonte, nem como endereço de destino, mas pode ser imaginado como sendo de uma interface virtual, a interface *loopback*. Um pacote IPv6 com endereço destino do tipo *loopback address* também não deve deixar o próprio *host*, sendo que esse endereço nunca será repassado por um roteador IPv6.

## Unspecified Address

Esse tipo de endereço indica exatamente a ausência de um endereço. Ele nunca deverá ser utilizado como um endereço válido para nenhum *host*. A sua utilidade é para que estações que ainda não foram inicializadas, sejam identificadas com endereços deste tipo, ou seja, *hosts* que ainda não tenham aprendido seus próprios endereços globais, utilizem tais endereços para se autoconfigurar. Além disso, esse tipo de endereço não

deve ser utilizado como endereço de destino ou em cabeçalho de roteamento de pacotes IPv6. Seu formato é o seguinte:

**0:0:0:0:0:0:0**

ou simplesmente:

::

## **NSAP Address**

Esse tipo de endereço é identificado pelo prefixo FP - 0000001, já visto na tabela III. Ele foi definido pela RFC 1888 - *OSI NSAPs and IPv6* como mecanismo de suporte para endereçamento OSI NSAP - *Network Service Access Point* em redes IPv6. Possui além do FP de 7 bits, que o indica, 121 bits para constituição de seu endereço.

## **IPX Address**

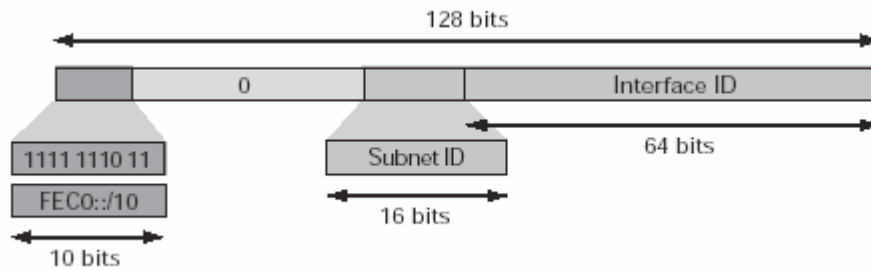
É também um endereço especial para compatibilidade de endereços existentes. É identificado pelo prefixo 0000010, incluído para prover mecanismo de mapeamento de endereços IPX - *Internal Packet eXchange* em endereços IPv6. Os endereços IPX são utilizados em redes *Netware*, de propriedade da Novell. Da mesma forma que o *NSAP Address* possui 7 bits de FP e 121 bits para constituição do endereço.

## **Site Local Unicast Address**

O endereço do tipo *Site Local* é similar aos endereços privados usados em IPv4, como as redes 10.0.0.0 /8, 172.16.0.0/16 e 198.168.0.0/16. Esses endereços podem ser usados para uma comunicação restrita dentro de um domínio específico.

Este tipo de endereço é identificado pelo prefixo **FEC0::/10** ou **1111111011** em binário. Ele pode ser definido para uso interno numa organização através da concatenação do campo de SLA (16 bits) com a identificação da interface (64 bits). Este

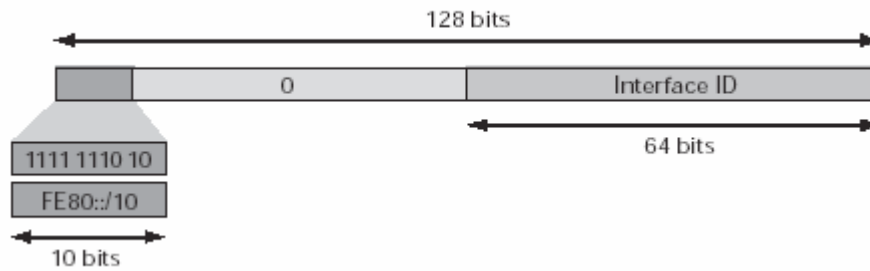
tipo de endereçamento pode ser considerado como privado, visto que ele está restrito a um domínio sem ligação à Internet. Desta forma ele não pode ser anunciado externamente por roteadores. Abaixo podemos visualizar a estrutura deste tipo de endereço na figura 13.



13. Estrutura do endereço Site Local Unicast Address

## Link Local Unicast Address

Este tipo de endereço é automaticamente configurado em qualquer *host* IPv6, através da conjugação do seu prefixo **FE80::/10** ou **1111111010** em binário, como pode ser visto na tabela III, e a identificação da interface no formato EUI-64, mostrado anteriormente. Estes endereços são utilizados nos processos de configuração dinâmica automática (autoconfiguração) e no processo de descoberta de elementos na hierarquia de roteamento (*Neighbor Discovery Protocol*). Estes procedimentos serão vistos com detalhes na próxima seção. Este endereçamento permite também a comunicação entre nós pertencentes ao mesmo enlace. Como nos endereços *Site Local*, esse tipo de endereço não deve ser enviado como endereço de origem ou destino em pacotes. Além disso esses endereços não são repassados pelos roteadores. Abaixo podemos visualizar a estrutura deste tipo de endereço na figura 14.



14. Estrutura do endereço Site Local Unicast Address

## IPv4-compatible IPv6 Address

Esse tipo de endereço é usado em IPv6 como um mecanismo de transição entre IPv6 e IPv4. É utilizado como endereços de destino e origem em *tunnel* (encapsulamento de um protocolo sobre outro) IPv6 sobre IPv4.

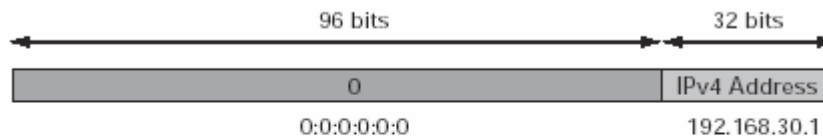
É representado por um endereço IPv6 cujos últimos 32 bits são um endereço IPv4. Desta forma, anexando-se um prefixo nulo (96 bits de zeros) a um endereço IPv4 (32 bits) obtém-se o seguinte formato:

**0:0:0:0:0:192.168.30.1**

ou no seu formato abreviado

**::192.168.30.1**

Abaixo é mostrada a estrutura deste endereço na figura 15:



IPv4-Compatible Address = 0:0:0:0:0:192.168.30.1  
 = ::192.168.30.1  
 = ::C0A8:1E01

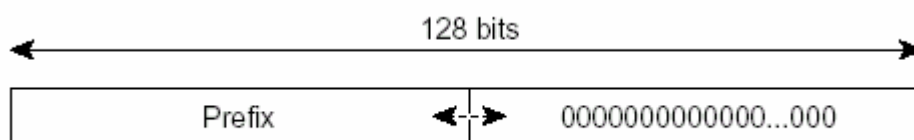
15. Estrutura do endereço IPv6 compatible IPv4 Address

## 5.2. Endereços *Anycast*

Esse tipo de endereço é utilizado para identificar um grupo de interfaces pertencentes a *hosts* diferentes. Um pacote destinado a um endereço *Anycast* é enviado para um das interfaces identificadas pelo endereço. Especificamente, o pacote é enviado para a interface mais próxima, de acordo com o protocolo de roteamento.

Um endereço do tipo *Anycast* não pode ser utilizado como endereço de origem de um pacote IPv6. Este tipo de endereçamento será útil na detecção rápida de um determinado servidor ou serviço. Por exemplo, poderá ser definido um grupo de servidores de DNS configurados com endereçamento *Anycast*, assim um *host* irá alcançar o servidor mais próximo utilizando este tipo de endereço.

Existe um prefixo mais longo desse mesmo endereço para cada endereço *Anycast* atribuído que identifica a região ao qual todas as interfaces pertencem. Abaixo é mostrada a estrutura básica deste tipo de endereço na figura 16.



16. Estrutura do endereço *Anycast*

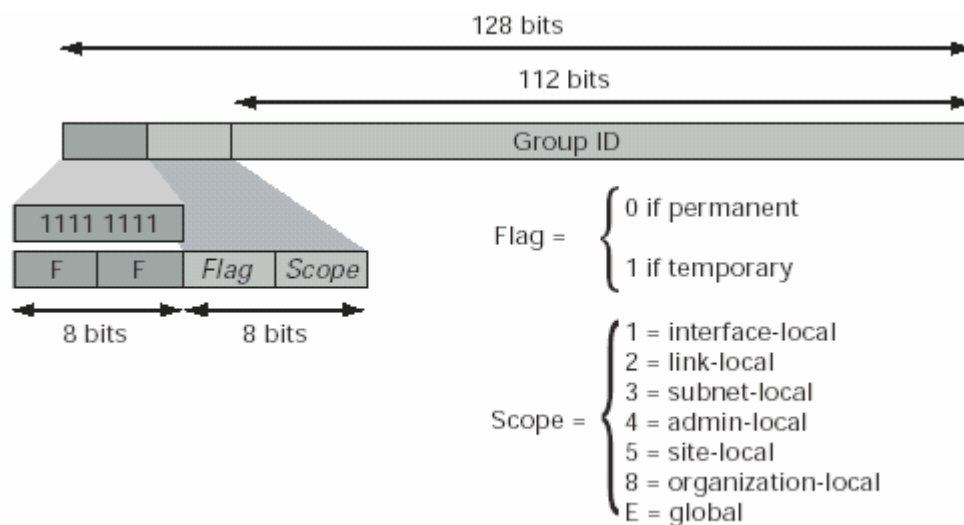
## 5.3 Endereços *Multicast*

Da mesma forma que o endereço *Anycast*, este endereço identifica um grupo de interfaces pertencente a diferentes *hosts* mas um pacote destinado a um endereço *Multicast* é enviado para todas as interfaces que fazem parte deste grupo.

Um endereço do tipo *Multicast Address* é um endereço IPv6, que é indicado pelo prefixo FP, como visto na tabela III, **FF00::/8** ou **11111111** em binário. O segundo

octeto que se segue ao prefixo (FP = FF) define o tempo de vida (*lifetime*), os 4 primeiros bits e o escopo do endereço *Multicast*, os últimos 4 bits deste octeto. Um endereço com *lifetime* permanente tem um parâmetro de tempo de vida igual a "0", enquanto um endereço temporário tem o mesmo parâmetro igual a "1". O escopo para este tipo de endereço apresenta os valores já definidos de 1, 2, 3, 4, 5, 8 e "E" (os outros estão reservados para o futuro, sendo que o escopo F já está reservado para ser usado para âmbito galáctico) para identificar um *host*, enlace, *site*, organização ou um escopo global, respectivamente. Os outros 112 *bits* são utilizados para identificar o grupo *Multicast*.

Abaixo, visualizamos a estrutura de um endereço do tipo *Multicast* na figura 17:



17. Estrutura do endereço Anycast

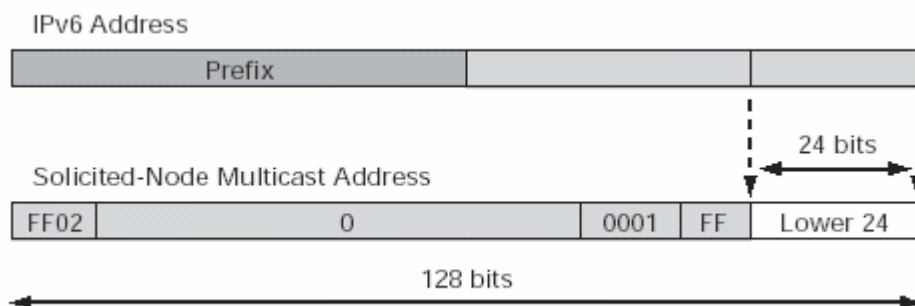
Dentro dos endereços *Multicast* já reservados, podemos identificar alguns endereços especiais utilizados para funções específicas (todos de *lifetime* permanente):

- FF01::1 – Indica todas as interfaces de escopo local, isto é, somente as interfaces de um mesmo *host*.

- FF02::1 – Indica todas as interfaces de um escopo de enlace local, isto é, todos os *hosts* de um mesmo domínio de colisão.
- FF01::2 – Indica todos os roteadores dentro de um escopo local, isto é, todas as interfaces de um mesmo roteador.
- FF02::2 – Indica todos os roteadores dentro de um escopo de enlace local, isto é, todos os roteadores interligados por um mesmo enlace.
- FF05::2 – Indica todos os roteadores dentro de um escopo *site* local, isto é, todos os roteadores que possuem um mesmo *site ID*.
- FF02::1:FFxx:xxxx – Endereço especial chamado de *Solicited-Node Multicast Address*, onde xx:xxxx representam os últimos 24 *bits* do endereço IPv6 *Unicast* do *host*.

### **Solicited-Node Multicast Address**

Esse tipo de endereço *Multicast* especial é usado para mensagens de solicitação de vizinho que auxilia o *Neighbor Discovery Protocol* e que será visto com mais detalhes na próxima seção. Esse endereço é um grupo *Multicast* que corresponde a um endereço IPv6 *Unicast*. A figura 18 abaixo apresenta a estrutura desse endereço.



18. Estrutura do endereço Anycast

## 6. Operações Básicas

Nesta seção serão abordadas algumas características importantes do IPv6. Estaremos interessados em verificar as operações básicas do novo protocolo IPv6, as suas mudanças em relação ao protocolo IPv4 e algumas funcionalidades. Essas funcionalidades do IPv6 utilizam principalmente o protocolo ICMP – *Internet Control Message Protocol*. Desta forma, primeiramente analisaremos o novo protocolo ICMPv6.

### 6.1 ICMPv6 Packet

As funcionalidades do ICMP em IPv6, definido pela RFC 1885, são similares as do ICMP em IPv4, mensagens de erros gerais como *destination unreachable messages*, *echo request messages* e *echo reply messages*. Além disso, o pacote ICMP em IPv6 é usado para novas funcionalidades como *neighbor discovery process* e *path MTU Discovery*.

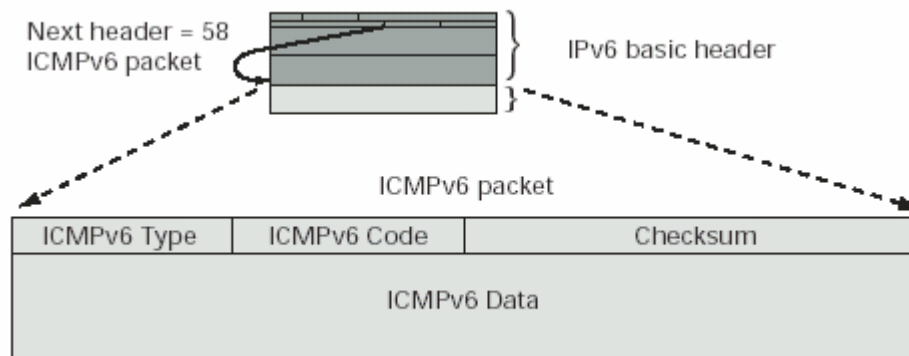
O pacote ICMP é na verdade um cabeçalho de extensão do cabeçalho básico IPv6. Quando o campo *next header* está com valor 58, indica que o próximo cabeçalho de extensão será o pacote ICMP.

Dentro do pacote ICMPv6, existem os campos *type* e *code*, que identificam o tipo de mensagem que será enviada. O campo *checksum* possui um valor que é derivado dos outros campos, sendo utilizado para controle de erro. Por fim, o campo *data* contém o erro ou a informação de diagnóstico relevante ao processo.

Assim com em IPv4, o ICMPv6 também é freqüentemente bloqueado por políticas de segurança através de *firewalls*, porque muitas vezes os ataques às redes são baseados em ICMP. Entretanto, o ICMPv6 pode utilizar IPSec, autenticação e

criptação. Desta forma, ataques baseados em ICMPv6 passam a ser menos frequentes.

Abaixo, é apresentada a estrutura do pacote ICMPv6. na figura 19



19. Estrutura do Pacote ICMPv6

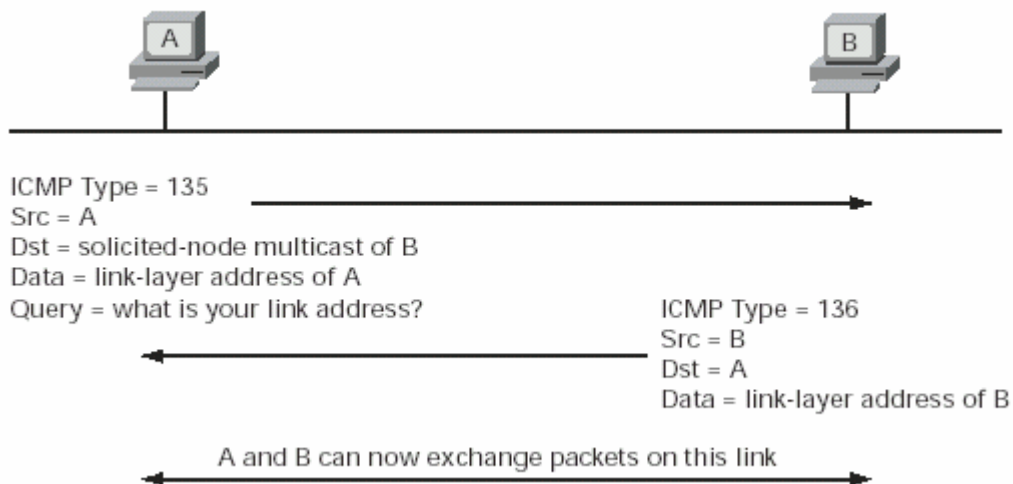
## 6.2 Neighbor Discovery Protocol

O Protocolo *Neighbor Discovery*, definido na RFC 1970, é uma das novas funcionalidades do IPv6, sendo importante para dinamizar alguns de seus processos. Ele habilita roteadores e *hosts* IPv6 a determinar o *MAC Address* de seus vizinhos de mesmo enlace, encontrar roteadores vizinhos e manter uma tabela de vizinhos. O processo *Neighbor Discovery* utiliza mensagens ICMPv6, endereços *Multicast* do tipo *Solicited-Node*, vistos anteriormente, para determinar os endereços *MAC* e verificar alcançabilidade de algum vizinho. Desta forma, é importante que todos os *hosts* IPv6 estejam associados a alguns grupos *Multicast* específicos vistos na seção anterior.

O processo *Neighbor Discovery* utiliza 2 mecanismos de operação: *Neighbor Solicitation* e *Neighbor Advertisement* que serão apresentados a seguir. O *Neighbor Solicitation* é usado quando um *host* precisa determinar o endereço *MAC* de um vizinho do mesmo enlace. Esta função substitui o protocolo ARP no IPv6 sem utilizar endereços *broadcast*. O processo funciona da seguinte forma: o *host* origem envia uma mensagem para os seus vizinhos com valor 135, no campo *type* do protocolo ICMP. Ele utiliza

como endereço de origem um do tipo *Multicast Solicited Node*, relativo ao endereço IPv6 do vizinho a ser alcançado. Na área de dados, ele envia o seu endereço MAC e também uma solicitação do endereço MAC do vizinho.

Quando o vizinho recebe a mensagem para o grupo *Multicast*, identifica como endereçada a ele e responde o pedido utilizando o *Neighbor Advertisement Message*, que nada mais é que uma mensagem de resposta à solicitação feita. Para isso, ele envia uma mensagem com o valor 136 no campo *type* do ICMP e no campo *data* transmite o seu endereço MAC. Abaixo podemos verificar um esquema deste processo.



20. Processo de Neighbor Discovery

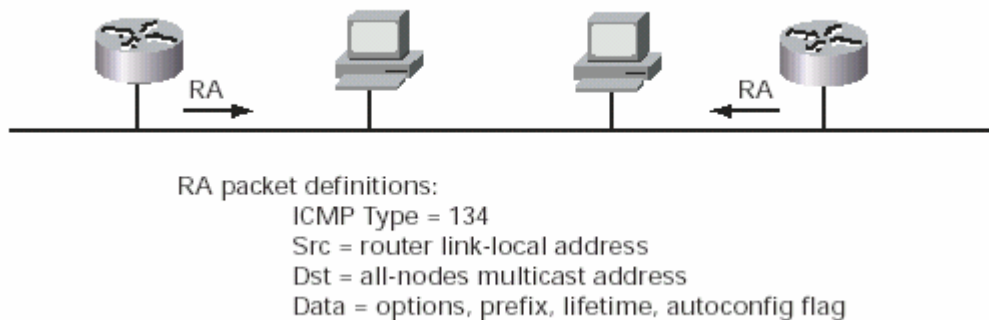
### 6.3 Router Discovery

O *Router Discovery* é um processo usado para que os *hosts* possam descobrir os roteadores existentes em seu enlace. Para isso, ele utiliza as mensagens de *Router Advertisement* e *Router Solicitation*.

As mensagens de *Router Advertisement* são periodicamente enviadas por cada interface de um roteador IPv6 e também é enviada como resposta de uma mensagem de *Router Solicitation*. Essas mensagens são enviadas para todos os nós de um mesmo

enlace através do endereço *Multicast* FF02::1 ou para o endereço *Unicast* específico recebido através de uma mensagem de *Router Solicitation*.

Para isso, ele utiliza o valor 134 no campo *type* do ICMPv6 e tem na área de dados as seguintes informações: prefixo de rede para autoconfiguração, tipo de autoconfiguração, o tempo de vida do prefixo informado e outras informações adicionais, como a MTU. Abaixo mostramos um esquema do *Router Advertisement*:



21. Processo de Router Discovery

A *Router Solicitation* utiliza o valor 133 no campo *type* do ICMPv6 e é utilizada pelos *hosts* que ainda não têm endereços IPv6 configurados. Desta forma, eles utilizam como endereço de origem um do tipo *Unspecified* (::), e como endereço de destino FF02::2, grupo *Multicast*, que alcança todos os roteadores do enlace.

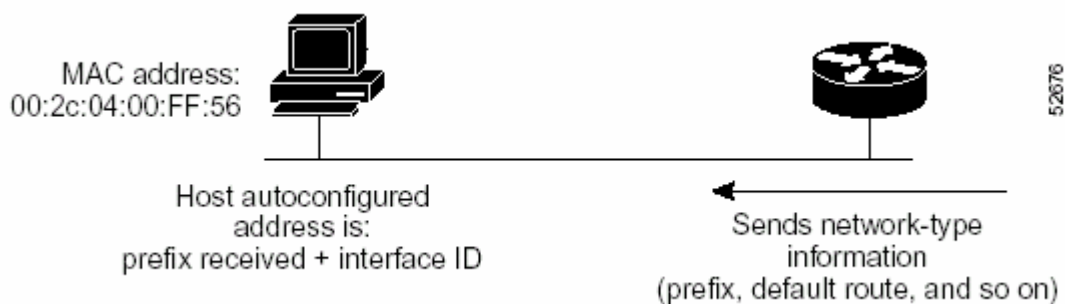
## 6.4 Autoconfiguração

O protocolo IPv6 define uma nova característica que não existia em IPv4, a autoconfiguração de endereços sem a necessidade de servidores DHCP – *Dinamic Host Configuration Protocol*. Para tal função todos os *hosts* IPv6 devem ter endereços do tipo *link-local*, que são automaticamente configurados no momento que o *hosts* é inicializado, como visto anteriormente (prefixo FE80::/10 + endereço MAC). Esse tipo

de endereço habilita o *host* a se comunicar com vizinhos do mesmo enlace e também a configurar-se.

Desta forma, ao receber uma mensagem de *Router advertisement*, vindo do roteador de seu enlace, ele pode automaticamente configurar-se com um endereço *Unicast Global*. Como visto acima, a mensagem de *Router advertisement* informa o prefixo da rede e o endereço do roteador, que serão usados na configuração da rota *default*. Porém, para que o *host* seja capaz de utilizar as informações do *Router advertisement* para autoconfigurar-se é necessário que o prefixo de rede informado seja de 64 bits, caso contrário o *host* não poderá executar tal operação.

Outra forma de receber as informações necessárias para a autoconfiguração é o envio, por parte do *host* de uma mensagem de *Router solicitation* como visto acima. Abaixo vemos um esquema exemplificando esse processo.



22. Processo de Autoconfiguração

## 7. Experimentos

Neste capítulo serão apresentados 3 experimentos feitos no Laboratório de Redes da RedeRio/SECTI que está localizado nas dependências do Centro Brasileiro de Pesquisas Físicas – CBPF/MCT. Tais experimentos tiveram objetivo geral de entender melhor o funcionamento do protocolo IPv6, além de testar suas funcionalidades nos equipamentos existentes em nosso *Backbone*.

Nestes experimentos foram utilizados os seguintes equipamentos:

- 2 Roteadores Cisco 2501, com versão de IOS 12.2(4)T1 – com suporte à IPv6;
- 1 Roteador Cisco 4500, com versão de IOS 12.1 – sem suporte à IPv6;
- 1 PC com sistema operacional Linux – *Mandrake 9.11*;
- 1 PC com sistema operacional Linux – *Red Hat 9.1*;

Após cada um desses experimentos, foi preparado um relatório que descrevem os seguintes itens:

- **Objetivo** - descreve as motivações e os propósitos para cada experimento;
- **Arquitetura utilizada na rede** - mostra a estrutura de rede utilizada em cada experimento, observando os equipamentos envolvidos, os enlaces, os endereços das interfaces, utilização de interface *tunnel* e o protocolo de roteamento;
- **Configuração de equipamentos** - descreve os comandos utilizados para configuração dos equipamentos envolvidos em cada experimento e mostra a configuração dos roteadores envolvidos em cada experimento, através do comando `SHOW RUNNING-CONFIG`;
- **Conclusões** - apresenta alguns comandos específicos para verificar a conectividade e funcionamento do experimento;

## 7.1. Experimento 1: Configuração do protocolo IPv6 em um segmento de rede

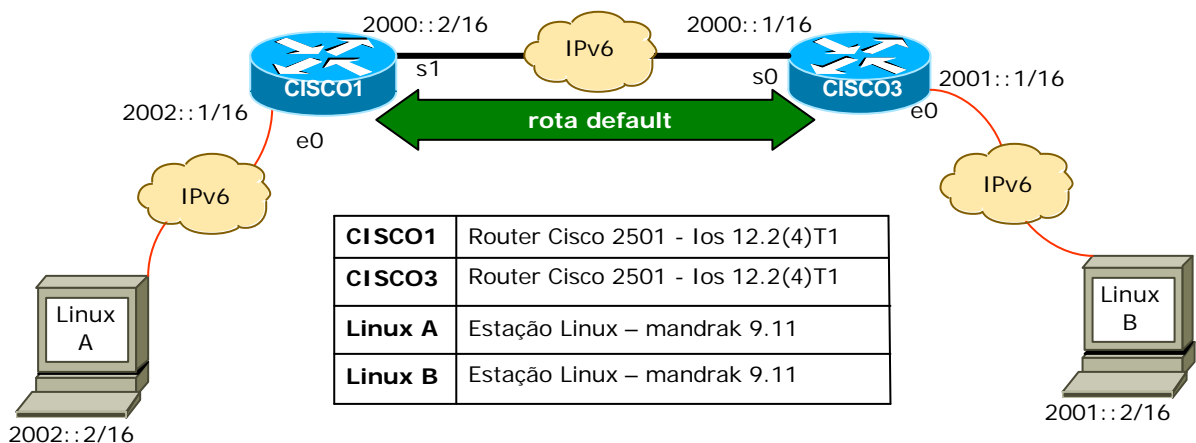
### Objetivos

Para este experimento utilizamos somente equipamentos (roteadores e estações) com suporte ao protocolo IPv6. Configuramos todas as interfaces dos roteadores e das estações Linux com endereços de rede IPv6. As estações também possuem endereços IPv4, pois é uma limitação do sistema operacional onde só é possível configurar endereços IPv6 se também existir endereços IPv4.

Verificaremos os seguintes itens:

- configuração das interfaces de todos os roteadores;
- configuração de uma rota *default* em cada roteador para substituição dos protocolos de roteamento;
- conectividade entre os roteadores e estações;
- resultados das tabelas de rotas IPv6 nos roteadores.

### Arquitetura utilizada na rede



## Configuração dos equipamentos

### Comandos para os roteadores CISCO1 e CISCO3

#### Habilitando o roteamento IPv6:

```
CISCO1#  
CISCO1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO1(config)#ipv6 unicast-routing  
CISCO1(config)#
```

#### Configuração de endereço IPv6 global-unicast numa interface serial:

```
CISCO3#  
CISCO3#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO3(config)#interface serial [nº da interface]  
CISCO3(config-if)#ipv6 enable  
CISCO3(config-if)#ipv6 address [endereço IPv6*] [prefixo de rede**]  
CISCO3(config-if)#clock rate [Banda***]  
CISCO3(config-if)#
```

Obs:

\*Refere-se a notação do endereço a ser configurado –  
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx – onde “x” são algarismos hexadecimais  
divididos em 8 grupos de 4 cada separado por “:” – Ex: 2000::1 (note que os grupos  
somente formados por zeros podem ser simplificados).

\*\*Refere-se ao no de bits que fazem parte do prefixo de rede – Ex: 2000::1/16  
(os 16 primeiros bits deste endereço “2000” referem-se a rede e o restante “112 bits”  
indicam a interface “::1”).

\*\*\*Refere-se a taxa de transmissão de bits utilizada nesta interface (geralmente  
entre 16000 bits/s até 2000000 bits/s).

### **Configuração de endereço IPv6 *global-unicast* numa interface ethernet:**

```
CISCO3#  
CISCO3#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO3(config)#interface ethernet [n° da interface]  
CISCO3(config)#ipv6 enable  
CISCO3(config-if)#ipv6 address [endereço IPv6] [prefixo de rede]  
CISCO3(config-if)#
```

### **Configuração de uma rota *default*:**

```
CISCO1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO1(config)#ipv6 route ::/0 [interface de saída*]  
CISCO1(config-if)#
```

Obs:

\* - refere-se a interface pela qual o roteador comunica-se com seu ISP – *Internet*

*Service Provider* , pode ser colocado a interface, Ex: serial 1, ou o endereço da interface

do próximo roteador, Ex: 2000::1.

## **SHOW RUNNING-CONFIG**

O comando *show running-config* é nativo do sistema operacional dos roteadores Cisco, que apresenta as atuais configurações do equipamento. Este comando é importante para verificarmos se as configurações feitas através de linhas de comando foram executadas.

Podemos verificar vários itens no *show running-config*, onde se destacam:

- o protocolo de endereçamento utilizado em cada interface;
- os endereços dados a cada interface;
- se a interface está ativada ou não;
- a taxa de transmissão de bits nas interfaces seriais;
- protocolos de roteamento utilizados
- redes anunciadas por cada protocolo de roteamento;
- rotas estáticas configuradas;
- filtros existentes, etc.

Abaixo são mostrados os resultados do comando *show running-config* para os roteadores utilizados neste experimento:

## Resultados CISCO1

```
CISCO1#show running-config
Building configuration...

Current configuration : 587 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname CISCO1
!
enable secret 5 $1$53N9$VIUVRghlRRihc/lyd2Q9r1
enable password cisco
!
ip subnet-zero
!
ipv6 unicast-routing
!
!
!
interface Ethernet0
  no ip address
  ipv6 address 2002::1/16
  ipv6 enable
!
interface Serial0
  no ip address
  shutdown
  no fair-queue
!
interface Serial1
  no ip address
  ipv6 address 2000::2/16
  ipv6 enable
!
ip classless
ip http server
ip pim bidir-enable
!
ipv6 route ::/0 2000::1
!
!
line con 0
  password cisco
line aux 0
line vty 0 4
!
end
```

## Resultados CISCO3

CISCO3#**show running-config**

Building configuration...

Current configuration : 598 bytes

```
!  
version 12.2  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname CISCO3  
!  
enable secret 5 $1$KNqz$FJlrixXvAm.bMEfz5y0m7.  
enable password cisco  
!  
ip subnet-zero  
!  
ipv6 unicast-routing  
!  
!  
!  
interface Ethernet0  
no ip address  
ipv6 address 2001::1/16  
ipv6 enable  
!  
interface Serial0  
no ip address  
ipv6 address 2000::1/16  
ipv6 enable  
clockrate 2000000  
!  
interface Serial1  
no ip address  
shutdown  
!  
ip classless  
ip http server  
ip pim bidir-enable  
!  
ipv6 route ::/0 2000::2  
!  
!  
line con 0  
password cisco  
line aux 0  
line vty 0 4  
login  
!  
end
```

## Conclusões

Neste item serão apresentados três comandos fundamentais para que possamos tirar as conclusões a respeito do correto funcionamento de nossa rede de testes. São eles os comandos **ping6** (que verifica a conectividade da rede IPv6), **traceroute6** (que mostra por quais roteadores um pacote enviado passa até alcançar seu destino) e o **show ipv6 route** (que apresenta a tabela de rotas de cada roteador).

### PING6

Verifica a conectividade entre a estação **Linux A** (de onde é executado o programa ping6) e a estação **Linux B**. O programa ping6 envia um pacote ICMP com 64 bytes de dados para o endereço indicado (neste caso 2002::2 – endereço IPv6 do Linux A) e aguarda o retorno do mesmo, calculando o tempo gasto para execução desta tarefa.

```
[root@linuxA bruno]# ping6 -c 5 2001::2
PING 2001::2(2001::2) 56 data bytes
64 bytes from 2001::2: icmp_seq=1 ttl=62 time=7.88 ms
64 bytes from 2001::2: icmp_seq=2 ttl=62 time=7.15 ms
64 bytes from 2001::2: icmp_seq=3 ttl=62 time=7.25 ms
64 bytes from 2001::2: icmp_seq=4 ttl=62 time=7.09 ms
64 bytes from 2001::2: icmp_seq=5 ttl=62 time=6.98 ms

--- 2001::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4044ms
rtt min/avg/max/mdev = 6.986/7.274/7.880/0.323 ms
```

### **Verifica a conectividade entre a estação Linux B e a estação Linux A.**

```
[root@linuxB aua]# ping6 -c 5 2002::2
PING 2002::2(2002::2) 56 data bytes
64 bytes from 2002::2: icmp_seq=1 ttl=62 time=8.08 ms
64 bytes from 2002::2: icmp_seq=2 ttl=62 time=6.94 ms
64 bytes from 2002::2: icmp_seq=3 ttl=62 time=7.05 ms
64 bytes from 2002::2: icmp_seq=4 ttl=62 time=7.04 ms
64 bytes from 2002::2: icmp_seq=5 ttl=62 time=6.92 ms

--- 2002::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4042ms
rtt min/avg/max/mdev = 6.924/7.208/8.082/0.452 ms
```

## TRACEROUTE6

O exemplo abaixo mostra o caminho percorrido da estação **Linux A** até alcançar o destino, a estação **Linux B**. Podemos verificar que o pacote enviado do **Linux A** para o **Linux B** passa por 2002::1 - roteador **CISCO1**, interface ethernet 0, passa por 2000::1 - roteador **CISCO3**, interface serial 0, e alcança seu destino ao chegar em 2001::2 – estação **Linux B**.

```
[root@linuxA bruno]# traceroute6 2001::2
traceroute to 2001::2 (2001::2) from 2002::2, 30 hops max, 16 byte
packets
 1  2002::1 (2002::1)  2.983 ms  2.537 ms  *
 2  2000::1 (2000::1)  6.019 ms  27.437 ms  *
 3  2001::2 (2001::2)  7.284 ms  7 ms  6.963 ms
```

O exemplo abaixo mostra o caminho percorrido da estação **Linux B** até alcançar o destino, a estação **Linux A**. Podemos verificar que o pacote enviado do **Linux B** para o **Linux A** passa por 2001::1 - roteador **CISCO3**, interface ethernet 0, passa por 2000::2 - roteador **CISCO1**, interface serial 1, e alcança seu destino ao chegar em 2002::2 – estação **Linux A**.

```
[root@linuxB aua]# traceroute6 2002::2
traceroute to 2002::2 (2002::2) from 2001::2, 30 hops max, 16 byte
packets
 1  2001::1 (2001::1)  2.748 ms  2.63 ms  *
 2  2000::2 (2000::2)  7.256 ms  6.29 ms  *
 3  2002::2 (2002::2)  7.696 ms  7.061 ms  6.914 ms
```

## SHOW IPV6 ROUTE

**CISCO1** - Podemos verificar na tabela de rotas IPv6 do roteador CISCO1, que existem 3 tipos de rotas L, C, S. As rotas “L” (locais) são as rotas configuradas manualmente nas interfaces ou aquelas autoconfiguradas pelo protocolo IPv6. Por exemplo, as rotas para 2000::2/128 (1)-(endereço da interface serial 1) e 2002::1/128 (3)-(endereço da interface ethernet 0) são rotas do tipo “L” configuradas manualmente e são aprendidas pelas próprias interfaces. Já as rotas FE80::/10 (5)-(prefixo de endereço *link local*) e FF00::/8 (6)-(prefixo de endereço *multicast*) são rotas do tipo “L” configuradas automaticamente pelo protocolo. Essas rotas são necessárias para configurar equipamentos que não possuem endereços, para reconhecimento de vizinhos e reconhecimento de grupos multicast. As rotas do tipo “C” para 2000::/16 (2) e 2002::/16 (4) são de redes diretamente conectadas e aprendidas através das interfaces serial 1 e ethernet 0 respectivamente. A rota do tipo “S” para ::/0 (7) é uma rota *default*, isto é, indica o roteador para qual todos os pacotes enviados para redes que ele não conhece devem ser enviadas, neste caso para 2000::1 (endereço da rede 2000::/16 – diretamente conectada), entrada (2) da tabela de rotas.

```
CISCO1#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
Timers: Uptime/Expires

L   2000::2/128 [0/0]
    via ::, Serial1, 00:22:07/never      (1)
C   2000::/16 [0/0]
    via ::, Serial1, 00:22:10/never     (2)
L   2002::1/128 [0/0]
    via ::, Ethernet0, 00:15:02/never   (3)
C   2002::/16 [0/0]
    via ::, Ethernet0, 00:15:05/never   (4)
L   FE80::/10 [0/0]
    via ::, Null0, 00:35:09/never       (5)
L   FF00::/8 [0/0]
    via ::, Null0, 00:35:09/never       (6)
S   ::/0 [1/0]
    via 2000::1, Null, 00:22:10/never    (7)
```

**CISCO3** – Verificamos que a tabela de rotas é bem semelhante a tabela do CISCO1, as diferenças se devem somente aos endereços das interfaces e a rota *default*.

```
CISCO3#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
Timers: Uptime/Expires

L   2000::1/128 [0/0]
    via ::, Serial0, 00:26:38/never
C   2000::/16 [0/0]
    via ::, Serial0, 00:26:41/never
L   2001::1/128 [0/0]
    via ::, Ethernet0, 00:22:22/never
C   2001::/16 [0/0]
    via ::, Ethernet0, 00:22:25/never
L   FE80::/10 [0/0]
    via ::, Null0, 00:53:33/never
L   FF00::/8 [0/0]
    via ::, Null0, 00:53:33/never
S   ::/0 [1/0]
    via 2000::2, Null, 00:26:41/never
```

## 7.2. Experimento 2: Configuração do protocolo IPv6 em um segmento de rede com utilização do protocolo de roteamento RIPv6

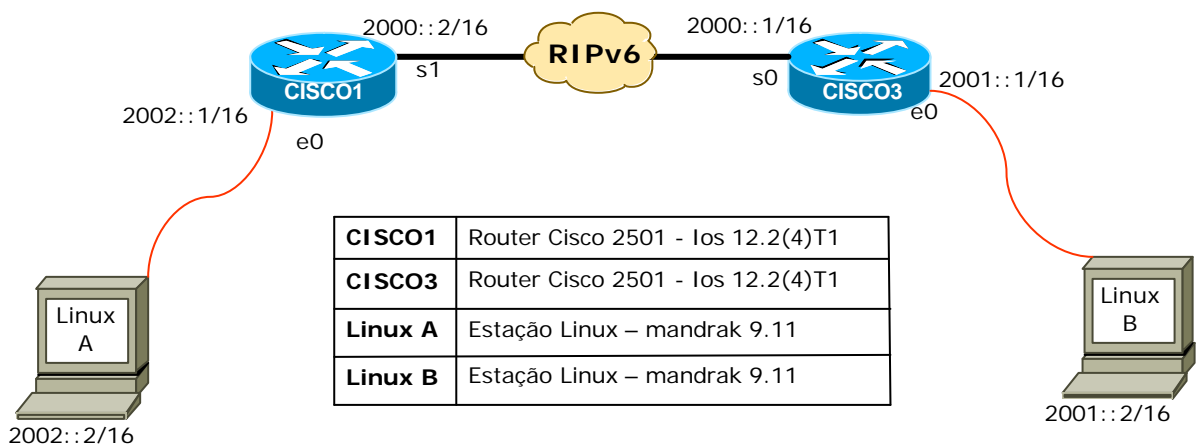
### Objetivos

Para este experimento utilizamos somente equipamentos (roteadores e estações) com suporte ao protocolo IPv6. Configuramos todas as interfaces dos roteadores e das estações Linux com endereços de rede IPv6. As estações também possuem endereços IPv4, pois é uma limitação do sistema operacional onde só é possível configurar endereços IPv6 se também existir endereços IPv4.

Verificaremos os seguintes itens:

- configuração das interfaces de todos os equipamentos;
- configuração do protocolo de roteamento RIPv6 e anúncio de suas redes;
- conectividade entre os roteadores e estações;
- resultados das tabelas de rotas IPv6 nos roteadores.

### Arquitetura utilizada na rede



## Configuração dos equipamentos

### Comandos para os roteadores CISCO1 e CISCO3

#### Habilitando o roteamento IPv6:

```
CISCO1#  
CISCO1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO1(config)#ipv6 unicast-routing  
CISCO1(config)#
```

#### Configuração de endereço IPv6 global-unicast numa interface serial:

```
CISCO3#  
CISCO3#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO3(config)#interface serial [nº da interface]  
CISCO3(config-if)#ipv6 enable  
CISCO3(config-if)#ipv6 address [endereço IPv6*] [prefixo de rede**]  
CISCO3(config-if)#clock rate [Banda***]  
CISCO3(config-if)#
```

Obs:

\*Refere-se a notação do endereço a ser configurado –  
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx – onde “x” são algarismos hexadecimais  
divididos em 8 grupos de 4 cada separado por “:” – Ex: 2000::1 (note que os grupos  
somente formados por zeros podem ser simplificados).

\*\*Refere-se ao no de bits que fazem parte do prefixo de rede – Ex: 2000::1/16  
(os 16 primeiros bits deste endereço “2000” referem-se a rede e o restante “112 bits”  
indicam a interface “::1”).

\*\*\*Refere-se a taxa de transmissão de bits utilizada nesta interface (geralmente  
entre 16000 bits/s até 2000000 bits/s).

## Configuração de endereço IPv6 *global-unicast* numa interface ethernet:

```
CISCO3#
CISCO3#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
CISCO3(config)#interface ethernet [n° da interface]
CISCO3(config)#ipv6 enable
CISCO3(config-if)#ipv6 address [endereço IPv6] [prefixo de rede]
CISCO3(config-if)#
```

## Criação de um processo do protocolo RIPv6:

```
CISCO1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
CISCO1(config)#ipv6 router rip [nome*]
CISCO1(config-if)#
```

Obs:

\* - refere-se ao nome dado ao processo criado para o funcionamento do protocolo RIPv6, Ex: ipv6 router rip “teste2”.

## Habilitação do protocolo RIPv6 numa interface qualquer:

```
CISCO1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
CISCO1(config)#interface [tipo da interface] [n° da interface]
CISCO1(config-if)#ipv6 rip [nome] enable
```

## Anuncio das redes conectadas e habilitadas no processo do RIPv6:

```
CISCO1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
CISCO1(config)#ipv6 router rip [nome]
CISCO1(config-router)#redistribute connected
CISCO1(config-router)#
```

## **SHOW RUNNING-CONFIG**

O *show running-config* é um comando existente nos roteadores Cisco, que apresenta as atuais configurações do equipamento. Este comando é importante para verificarmos se as configurações feitas através de linhas de comando foram executadas.

Abaixo são mostrados os resultados do comando *show running-config* para os roteadores utilizados neste experimento, é importante observar as interfaces que estão habilitadas e anunciadas no processo RIPv6:

## Resultados CISCO1

```
CISCO1#show running-config
Building configuration...

Current configuration : 748 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname CISCO1
!
enable secret 5 $1$53N9$VIUVRghlRRihc/lyd2Q9r1
enable password cisco
!
ip subnet-zero
!
ipv6 unicast-routing
!
!
!
interface Ethernet0
 ip address 152.84.50.1 255.255.255.0
 ipv6 address 2002::1/16
 ipv6 enable
 ipv6 rip teste2 enable
!
interface Serial0
 no ip address
 shutdown
 ipv6 rip teste2 enable
 no fair-queue
!
interface Serial1
 no ip address
 ipv6 address 2000::2/16
 ipv6 enable
 ipv6 rip teste2 enable
!
ip classless
ip http server
ip pim bidir-enable
!
ipv6 router rip teste2
 redistribute connected
!
!
!
line con 0
 password cisco
line aux 0
line vty 0
 password cisco
 login
line vty 1 4
 login
!
end
```

## Resultados CISCO3

CISCO3#**show running-config**

Building configuration...

Current configuration : 700 bytes

```
!  
version 12.2  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname CISCO3  
!  
enable secret 5 $1$KNqz$FJlrixXvAm.bMEfz5y0m7.  
enable password cisco  
!  
ip subnet-zero  
!  
ipv6 unicast-routing  
!  
!  
!  
interface Ethernet0  
no ip address  
ipv6 address 2001::1/16  
ipv6 enable  
ipv6 nd managed-config-flag  
ipv6 rip teste2 enable  
!  
interface Serial0  
no ip address  
ipv6 address 2000::1/16  
ipv6 enable  
ipv6 rip teste2 enable  
clockrate 2000000  
!  
interface Serial1  
no ip address  
shutdown  
!  
ip classless  
ip http server  
ip pim bidir-enable  
!  
ipv6 router rip teste2  
redistribute connected  
!  
!  
!  
line con 0  
password cisco  
line aux 0  
line vty 0 4  
login  
!  
end
```

## Conclusões

Neste item serão apresentados três comandos fundamentais para que possamos tirar as conclusões a respeito do correto funcionamento de nossa rede de testes. São eles os comandos **ping6** (que verifica a conectividade da rede IPv6), **traceroute6** (que mostra por quais roteadores um pacote enviado passa até alcançar seu destino) e o **show ipv6 route** (que apresenta a tabela de rotas de cada roteador).

### PING6

O exemplo abaixo verifica a conectividade entre a estação **Linux A** (de onde é executado o programa ping6) e a estação **Linux B**. O programa ping6 envia um pacote ICMP com 64 bytes de dados para o endereço indicado (neste caso 2001::2 – endereço IPv6 do Linux B) e aguarda o retorno do mesmo, calculando o tempo gasto para execução desta tarefa.

```
[root@linuxA bruno]# ping6 -c 5 2001::2
PING 2001::2(2001::2) 56 data bytes
64 bytes from 2001::2: icmp_seq=1 ttl=62 time=6.77 ms
64 bytes from 2001::2: icmp_seq=2 ttl=62 time=6.69 ms
64 bytes from 2001::2: icmp_seq=3 ttl=62 time=6.91 ms
64 bytes from 2001::2: icmp_seq=4 ttl=62 time=6.79 ms
64 bytes from 2001::2: icmp_seq=5 ttl=62 time=6.75 ms

--- 2001::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4042ms
rtt min/avg/max/mdev = 6.690/6.785/6.915/0.104 ms
```

### Verifica a conectividade entre a estação Linux B e a estação Linux A.

```
[root@linuxB aua]# ping6 -c 5 2002::2
PING 2002::2(2002::2) 56 data bytes
64 bytes from 2002::2: icmp_seq=1 ttl=62 time=6.87 ms
64 bytes from 2002::2: icmp_seq=2 ttl=62 time=6.69 ms
64 bytes from 2002::2: icmp_seq=3 ttl=62 time=6.65 ms
64 bytes from 2002::2: icmp_seq=4 ttl=62 time=7.08 ms
64 bytes from 2002::2: icmp_seq=5 ttl=62 time=6.81 ms

--- 2002::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4037ms
rtt min/avg/max/mdev = 6.657/6.824/7.085/0.151 ms
```

## TRACEROUTE6

O exemplo abaixo mostra o caminho percorrido da estação **Linux A** até alcançar o destino, a estação **Linux B**. Podemos verificar que o pacote enviado do **Linux A** para o **Linux B** passa por 2002::1 - roteador **CISCO1**, interface ethernet 0, passa por 2000::1 - roteador **CISCO3**, interface serial 0, e alcança seu destino ao chegar em 2001::2 – estação **Linux B**.

```
[root@linuxA bruno]# traceroute6 2001::2
traceroute to 2001::2 (2001::2) from 2002::2, 30 hops max, 16 byte
packets
 1  2002::1 (2002::1)  8.617 ms  2.403 ms  *
 2  2000::1 (2000::1)  5.78 ms  6.277 ms  *
 3  2001::2 (2001::2)  8.948 ms  6.787 ms  6.608 ms
```

O exemplo abaixo mostra o caminho percorrido da estação **Linux B** até alcançar o destino, a estação **Linux A**. Podemos verificar que o pacote enviado do **Linux B** para o **Linux A** passa por 2001::1 - roteador **CISCO3**, interface ethernet 0, passa por 2000::2 - roteador **CISCO1**, interface serial 1, e alcança seu destino ao chegar em 2002::2 – estação **Linux A**.

```
[root@linuxB aua]# traceroute6 2002::2
traceroute to 2002::2 (2002::2) from 2001::2, 30 hops max, 16 byte
packets
 1  2001::1 (2001::1)  2.748 ms  2.63 ms  *
 2  2000::2 (2000::2)  7.256 ms  6.29 ms  *
 3  2002::2 (2002::2)  7.696 ms  7.061 ms  6.914 ms
```

## SHOW IPV6 ROUTE

CISCO1 - Podemos verificar na tabela de rotas IPv6 do roteador CISCO1, que existem 3 tipos de rotas L, C, R. As rotas do tipo “L” são as rotas configuradas manualmente nas interfaces ou aquelas autoconfiguradas pelo protocolo IPv6. Por exemplo, as rotas para 2000::2/128 (1)-(endereço da interface serial 1) e 2002::1/128 (4)-(endereço da interface ethernet 0) são rotas do tipo “L” configuradas manualmente e são aprendidas pelas próprias interfaces. Já as rotas FE80::/10 (6)-(prefixo de endereço *link local*) e FF00::/8 (7)-(prefixo de endereço *multicast*) são rotas do tipo “L” configuradas automaticamente pelo protocolo. Essas rotas são necessárias para configurar equipamentos que não possuem endereços, para reconhecimento de vizinhos e reconhecimento de grupos multicast. As rotas do tipo “C” para 2000::/16 (2) e 2002::/16 (5) são de redes diretamente conectadas e aprendidas através das interfaces serial 1 e ethernet 0 respectivamente. A rota do tipo “R” para 2001::/16 (7) é uma rota aprendida pelo protocolo RIPv6, através de sua interface serial 1(interface de comunicação com o CISCO3), note que ele aprende essa rota através de um endereço *link local* (\*), que o endereço deste tipo para a interface serial 1 (cada interface é automaticamente configurada com um endereço desse tipo quando o protocolo IPv6 é habilitado na mesma).

```

CISCO1#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
Timers: Uptime/Expires

L   2000::2/128 [0/0]
    via ::, Serial1, 01:23:19/never           (1)
C   2000::/16 [0/0]
    via ::, Serial1, 01:23:22/never         (2)
R   2001::/16 [120/2]
    via FE80::200:CFF:FE46:DEBC*, Serial1, 00:10:21/00:02:48 (3)
L   2002::1/128 [0/0]
    via ::, Ethernet0, 01:16:13/never       (4)
C   2002::/16 [0/0]
    via ::, Ethernet0, 01:16:17/never       (5)
L   FE80::/10 [0/0]
    via ::, Null0, 01:36:21/never           (6)
L   FF00::/8 [0/0]
    via ::, Null0, 01:36:21/never           (7)

```

**CISCO3** – Verificamos que a tabela de rotas é bem semelhante a tabela do CISCO1, as diferenças se devem somente aos endereços das interfaces e a rota aprendida por RIPv6.

```

CISCO3#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
Timers: Uptime/Expires

L   2000::1/128 [0/0]
    via ::, Serial0, 01:22:30/never
C   2000::/16 [0/0]
    via ::, Serial0, 01:22:33/never
L   2001::1/128 [0/0]
    via ::, Ethernet0, 00:28:11/never
C   2001::/16 [0/0]
    via ::, Ethernet0, 00:28:14/never
R   2002::/16 [120/2]
    via FE80::200:CFF:FE46:DE08, Serial0, 00:09:48/00:02:53
L   FE80::/10 [0/0]
    via ::, Null0, 01:49:24/never
L   FF00::/8 [0/0]
    via ::, Null0, 01:49:24/never

```

### 7.3. Experimento 3: Configuração do protocolo IPv6 em um segmento de rede, utilizando interface *Tunnel*

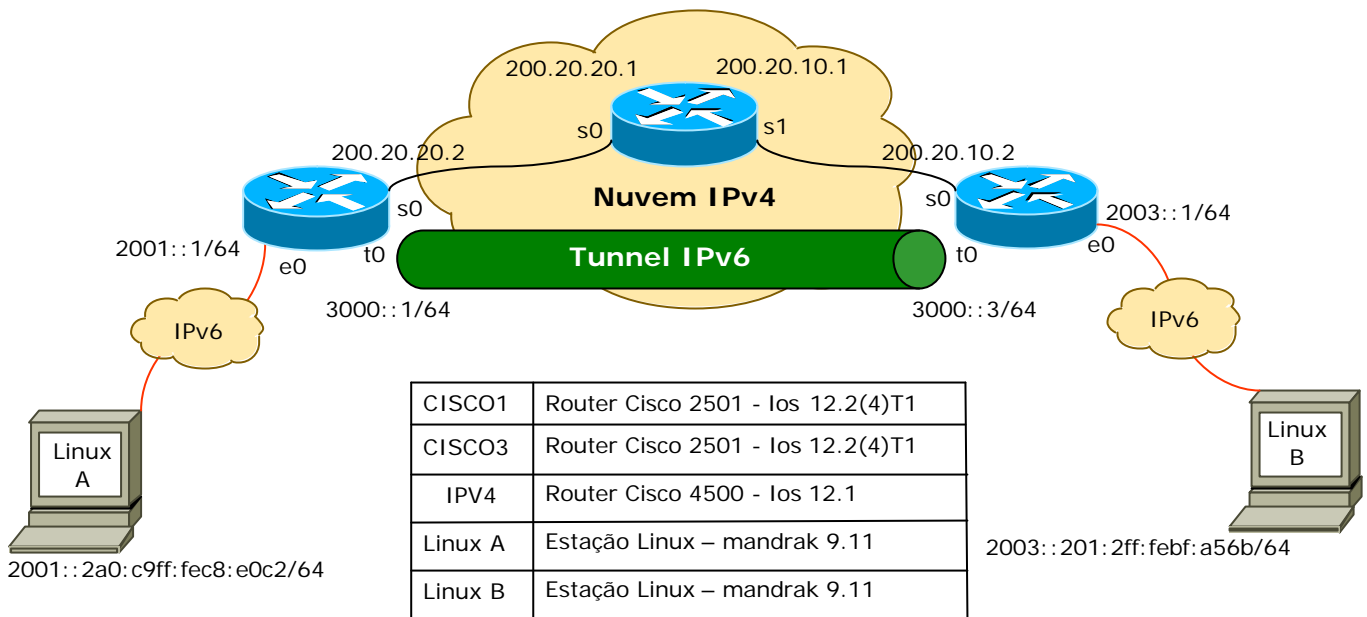
#### Objetivos

Para este experimento utilizamos equipamentos (roteadores e estações) com suporte ao protocolo IPv6 e um roteador com IOS sem suporte IPv6. Configuramos as interfaces *Ethernet* dos roteadores Cisco1 e Cisco3 e as estações Linux com endereços de rede IPv6 e também endereços IPv4. As interfaces Seriais dos roteadores Cisco1, Cisco3 e IPv4 possuem apenas endereços IPv4. Desta forma, podemos verificar no diagrama abaixo a formação de duas "ilhas" IPv6. Para uni-las utilizaremos uma conexão através de um *Tunnel* entre os roteadores Cisco1 e Cisco3.

Verificaremos os seguintes itens:

- configuração das interfaces *Ethernet* e *Tunnel* dos roteadores com suporte IPv6;
- configuração do protocolo de roteamento RIPv6 e anuncio de suas redes em cada roteador com suporte IPv6;
- conectividade entre os roteadores e estações;
- autoconfiguração das estações Linux. (Verificamos que a estação concatena o prefixo de rede do *gateway* com o próprio MAC e alguns caracteres de controle);
- resultados das tabelas de rotas IPv6 nos roteadores.

## Arquitetura utilizada na rede



## Configuração dos equipamentos

### Comandos para os roteadores CISCO1 e CISCO3

#### Habilitando o roteamento IPv6:

```
CISCO1#  
CISCO1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO1(config)#ipv6 unicast-routing  
CISCO1(config)#
```

#### Configuração de endereço IPv6 global-unicast numa interface ethernet:

```
CISCO3#  
CISCO3#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO3(config)#interface ethernet [n° da interface]  
CISCO3(config)#ipv6 enable  
CISCO3(config-if)#ipv6 address [endereço IPv6*] [prefixo de rede**]  
CISCO3(config-if)#
```

Obs:

\*Refere-se a notação do endereço a ser configurado –  
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx – onde “x” são algarismos hexadecimais  
divididos em 8 grupos de 4 cada separado por “:” – Ex: 2000::1 (note que os grupos  
somente formados por zeros podem ser simplificados).

\*\*Refere-se ao no de bits que fazem parte do prefixo de rede – Ex: 2000::1/16  
(os 16 primeiros bits deste endereço “2000” referem-se a rede e o restante (112 bits)  
indicam a interface “::1”).

#### Configuração de endereço IPv6 global-unicast numa interface tunnel:

```
CISCO3#  
CISCO3#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO3(config)#interface tunnel [n° da interface]  
CISCO3(config-if)#ipv6 enable  
CISCO3(config-if)#ipv6 address [endereço IPv6] [prefixo de rede]  
CISCO3(config-if)#
```

## Configuração de Tunnel entre dois roteadores para encapsulamento IPv6 sobre

### IPv4:

```
CISCO3#  
CISCO3#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO3(config)#interface tunnel [n° da interface]  
CISCO3(config-if)#tunnel source [tipo da interface de saída] [n° da  
interface]  
CISCO3(config-if)#tunnel destination [endereço de destino**]  
CISCO3(config-if)# tunnel mode ipv6ip***
```

Obs:

\*Refere-se a interface usada como fonte do *tunnel* Ex: Serial.

\*\*Refere-se ao endereço IPv4 da interface de destino do *tunnel* Ex: a interface serial 0 do roteador Cisco1, 200.20.20.2.

\*\*\*Refere-se ao modo de encapsulamento usado pelo *tunnel* Ex: neste caso usaremos IPv6 sobre IPv4.

### Criação de um processo do protocolo RIPv6:

```
CISCO1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO1(config)#ipv6 router rip [nome*]  
CISCO1(config-router)#
```

Obs:

\*Refere-se ao nome dado ao processo criado para o funcionamento do protocolo

RIPv6, Ex: ipv6 router rip “teste3”.

### Habilitação do protocolo RIPv6 numa interface qualquer:

```
CISCO1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO1(config)#interface [tipo da interface] [n° da interface]  
CISCO1(config-if)#ipv6 rip [nome] enable
```

## **Anuncio das redes conectadas e habilitadas no processo do RIPv6:**

```
CISCO1#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
CISCO1(config)#ipv6 router rip [nome]  
CISCO1(config-router)#redistribute connected  
CISCO1(config-router)#
```

## **SHOW RUNNING-CONFIG**

O *show running-config* é um comando existente nos roteadores Cisco, que apresenta as atuais configurações do equipamento. Este comando é importante para verificarmos se as configurações feitas através de linhas de comando foram executadas.

Abaixo são mostrados os resultados do comando *show running-config* para os roteadores utilizados neste experimento, é importante observar a configuração da interface *Tunnel*:

## Resultados CISCO1

CISCO1#**show running-config**

Building configuration...

Current configuration : 598 bytes

```
!  
version 12.2  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname CISCO1  
!  
!  
ip subnet-zero  
!  
ipv6 unicast-routing  
!  
!  
interface Tunnel0  
no ip address  
ipv6 address 3000::1/64  
ipv6 enable  
ipv6 rip 1 enable  
tunnel source Serial0  
tunnel destination 200.20.10.2  
tunnel mode ipv6ip  
!  
interface Ethernet0  
ip address 152.84.50.1 255.255.255.0  
ipv6 address 2001::1/64  
ipv6 enable  
ipv6 rip 1 enable  
!  
interface Serial0  
ip address 200.20.20.2 255.255.255.0  
no fair-queue  
!  
interface Serial1  
no ip address  
shutdown  
!  
router rip  
network 152.84.0.0  
network 200.20.20.0  
!  
ip classless  
ip http server  
ip pim bidir-enable  
!  
ipv6 router rip 1  
redistribute connected  
!  
!  
line con 0  
line aux 0  
line vty 0 4  
!  
end
```

## Resultados CISCO3

CISCO3#**show running-config**

Building configuration...

Current configuration : 598 bytes

```
!  
version 12.2  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname CISCO3  
!  
!  
ip subnet-zero  
!  
ipv6 unicast-routing  
!  
!  
interface Tunnel0  
no ip address  
ipv6 address 3000::3/64  
ipv6 enable  
ipv6 rip 1 enable  
tunnel source Serial0  
tunnel destination 200.20.20.2  
tunnel mode ipv6ip  
!  
interface Ethernet0  
ip address 200.20.30.1 255.255.255.0  
ipv6 address 2003::1/64  
ipv6 enable  
ipv6 rip 1 enable  
!  
interface Serial0  
ip address 200.20.10.2 255.255.255.0  
no fair-queue  
!  
interface Serial1  
no ip address  
shutdown  
!  
router rip  
network 200.20.10.0  
network 200.20.30.0  
!  
ip classless  
ip http server  
ip pim bidir-enable  
!  
ipv6 router rip 1  
redistribute connected  
!  
!  
line con 0  
line aux 0  
line vty 0 4  
!  
end
```

## Resultados IPV4

IPV4#**show running-config**  
Building configuration...

```
Current configuration : 598 bytes
!
version 12.1
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname IPv4
!
!
!
ip subnet-zero
!
!
!
interface Ethernet0
  no ip address
  shutdown
  media-type 10BaseT
!
interface Ethernet1
  no ip address
  shutdown
  media-type 10BaseT
!
interface Serial0
  ip address 200.20.10.1 255.255.255.0
  no fair-queue
  clockrate 2000000
!
interface Serial1
  ip address 200.20.20.1 255.255.255.0
  clockrate 2000000
!
interface Serial2
  no ip address
  shutdown
!
interface Serial3
  no ip address
  shutdown
!
router rip
  network 200.20.10.0
  network 200.20.20.0
!
ip classless
no ip http server
!
!
line con 0
line aux 0
line vty 0 4
!
end
```

## Conclusões

Neste item serão apresentados três comandos fundamentais para que possamos tirar as conclusões a respeito do correto funcionamento de nossa rede de testes. São eles os comandos **ping6** (que verifica a conectividade da rede IPv6), **traceroute6** (que mostra por quais roteadores um pacote enviado passa até alcançar seu destino) e o **show ipv6 route** (que apresenta a tabela de rotas de cada roteador).

### PING6

Verifica a conectividade entre a estação **Linux A** (de onde é executado o programa ping6) e a estação **Linux B**. O programa ping6 envia um pacote ICMP com 64 bytes de dados para o endereço indicado (neste caso 2003::201:2ff:febf:a56b – endereço IPv6 do Linux B) e aguarda o retorno do mesmo, calculando o tempo gasto para execução desta tarefa.

```
[root@linuxA raphagg]# ping6 2003::201:2ff:febf:a56b
PING 2003::201:2ff:febf:a56b(2003::201:2ff:febf:a56b) from
2001::2a0:c9ff:fec8:e0c2 : 56 data bytes
64 bytes from 2003::201:2ff:febf:a56b: icmp_seq=1 ttl=62 time=12.8 ms
64 bytes from 2003::201:2ff:febf:a56b: icmp_seq=2 ttl=62 time=11.9 ms
64 bytes from 2003::201:2ff:febf:a56b: icmp_seq=3 ttl=62 time=11.9 ms
64 bytes from 2003::201:2ff:febf:a56b: icmp_seq=4 ttl=62 time=11.9 ms
64 bytes from 2003::201:2ff:febf:a56b: icmp_seq=5 ttl=62 time=11.9 ms
64 bytes from 2003::201:2ff:febf:a56b: icmp_seq=6 ttl=62 time=12.6 ms

--- 2003::201:2ff:febf:a56b ping statistics ---
6 packets transmitted, 6 received, 0% loss, time 5010ms
rtt min/avg/max/mdev = 11.908/12.225/12.874/0.410 ms
```

## Verifica a conectividade entre a estação **Linux B** e a estação **Linux A**.

```
[root@linuxB raphagg]# ping6 2001::2a0:c9ff:fec8:e0c2
PING 2001::2a0:c9ff:fec8:e0c2(2001::2a0:c9ff:fec8:e0c2) 56 data bytes
64 bytes from 2001::2a0:c9ff:fec8:e0c2: icmp_seq=1 ttl=62 time=12.7 ms
64 bytes from 2001::2a0:c9ff:fec8:e0c2: icmp_seq=2 ttl=62 time=12.3 ms
64 bytes from 2001::2a0:c9ff:fec8:e0c2: icmp_seq=3 ttl=62 time=12.9 ms
64 bytes from 2001::2a0:c9ff:fec8:e0c2: icmp_seq=4 ttl=62 time=12.2 ms
64 bytes from 2001::2a0:c9ff:fec8:e0c2: icmp_seq=5 ttl=62 time=12.0 ms
64 bytes from 2001::2a0:c9ff:fec8:e0c2: icmp_seq=6 ttl=62 time=12.6 ms
64 bytes from 2001::2a0:c9ff:fec8:e0c2: icmp_seq=7 ttl=62 time=12.0 ms

--- 2001::2a0:c9ff:fec8:e0c2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6056ms
rtt min/avg/max/mdev = 12.073/12.434/12.929/0.325 ms
```

## TRACEROUTE6

Mostra o caminho percorrido da estação **Linux A** até alcançar o destino, a estação **Linux B**. Podemos verificar que o pacote enviado do **Linux A** para o **Linux B** passa por 2001::1 - roteador **CISCO1**, interface ethernet 0, passa por 3000::3 - roteador **CISCO3**, interface tunnel 0, e alcança seu destino ao chegar em 2003::2201:2ff:febf:a56b – estação **Linux B**. Podemos verificar que os pacotes passam pela nuvem IPv4, sem serem percebidos, isto é, utiliza Tunnel com encapsulamento IPv6 sobre IPv4.

```
[root@linuxA raphagg]# traceroute6 2003::201:2ff:febf:a56b
traceroute to 2003::201:2ff:febf:a56b (2003::201:2ff:febf:a56b) from
2001::2a0:c9ff:fec8:e0c2, 30 hops max, 16 byte packets
 1  2001::1 (2001::1)  2.476 ms *  2.396 ms
 2  3000::3 (3000::3)  10.843 ms *  11.007 ms
 3  2003::201:2ff:febf:a56b (2003::201:2ff:febf:a56b)  12.227 ms
11.705 ms  12.058 ms
```

Mostra o caminho percorrido da estação **Linux B** até alcançar o destino, a estação **Linux A**. Podemos verificar que o pacote enviado do **Linux B** para o **Linux A** passa por 2003::1 - roteador **CISCO3**, interface ethernet 0, passa por 3000::1 - roteador **CISCO1**, interface tunnel 0, e alcança seu destino ao chegar em 2001::2a0:c9ff:fec8:e0c2 – estação **Linux A**. Podemos verificar que os pacotes passam pela nuvem IPv4, sem serem percebidos, isto é, utiliza Tunnel com encapsulamento IPv6 sobre IPv4.

```
root@multicast raphagg]# traceroute6 2001::2a0:c9ff:fec8:e0c2
traceroute to 2001::2a0:c9ff:fec8:e0c2 (2001::2a0:c9ff:fec8:e0c2) from
2003::201:2ff:febf:a56b, 30 hops max, 16 byte packets
 1  2003::1 (2003::1)  2.975 ms  2.617 ms *
 2  3000::1 (3000::1)  10.953 ms  11.091 ms *
 3  2001::2a0:c9ff:fec8:e0c2 (2001::2a0:c9ff:fec8:e0c2)  13.746 ms
12.076 ms  11.593 ms
```

## SHOW IPV6 ROUTE

CISCO1 - Podemos verificar na tabela de rotas IPv6 do roteador CISCO1, que existem 3 tipos de rotas L, C, S. As rotas “L” (locais) são as rotas configuradas manualmente nas interfaces ou aquelas autoconfiguradas pelo protocolo IPv6. Por exemplo, as rotas para 2000::2/128 (1)-(endereço da interface serial 1) e 2002::1/128 (3)-(endereço da interface ethernet 0) são rotas do tipo “L” configuradas manualmente e são aprendidas pelas próprias interfaces. Já as rotas FE80::/10 (5)-(prefixo de endereço *link local*) e FF00::/8 (6)-(prefixo de endereço *multicast*) são rotas do tipo “L” configuradas automaticamente pelo protocolo. Essas rotas são necessárias para configurar equipamentos que não possuem endereços, para reconhecimento de vizinhos e reconhecimento de grupos multicast. As rotas do tipo “C” para 2000::/16 (2) e 2002::/16 (4) são de redes diretamente conectadas e aprendidas através das interfaces serial 1 e ethernet 0 respectivamente. A rota do tipo “S” para ::/0 (7) é uma rota *default*, isto é, indica o roteador para qual todos os pacotes enviados para redes que ele não conhece devem ser enviadas, neste caso para 2000::1 (endereço da rede 2000::/16 – diretamente conectada), entrada (2) da tabela de rotas.

```
CISCO1#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
Timers: Uptime/Expires

L   2000::2/128 [0/0]
    via ::, Serial1, 00:22:07/never (1)
C   2000::/16 [0/0]
    via ::, Serial1, 00:22:10/never (2)
L   2002::1/128 [0/0]
    via ::, Ethernet0, 00:15:02/never (3)
C   2002::/16 [0/0]
    via ::, Ethernet0, 00:15:05/never (4)
L   FE80::/10 [0/0]
    via ::, Null0, 00:35:09/never (5)
L   FF00::/8 [0/0]
    via ::, Null0, 00:35:09/never (6)
S   ::/0 [1/0]
    via 2000::1, Null, 00:22:10/never (7)
```

CISCO3 – Verificamos que a tabela de rotas é bem semelhante a tabela do CISCO1, as diferenças se devem somente aos endereços das interfaces e a rota *default*.

```
CISCO3#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
Timers: Uptime/Expires

L   2000::1/128 [0/0]
    via ::, Serial0, 00:26:38/never
C   2000::/16 [0/0]
    via ::, Serial0, 00:26:41/never
L   2001::1/128 [0/0]
    via ::, Ethernet0, 00:22:22/never
C   2001::/16 [0/0]
    via ::, Ethernet0, 00:22:25/never
L   FE80::/10 [0/0]
    via ::, Null0, 00:53:33/never
L   FF00::/8 [0/0]
    via ::, Null0, 00:53:33/never
S   ::/0 [1/0]
    via 2000::2, Null, 00:26:41/never
```

## 8. Implementação na Rede Rio

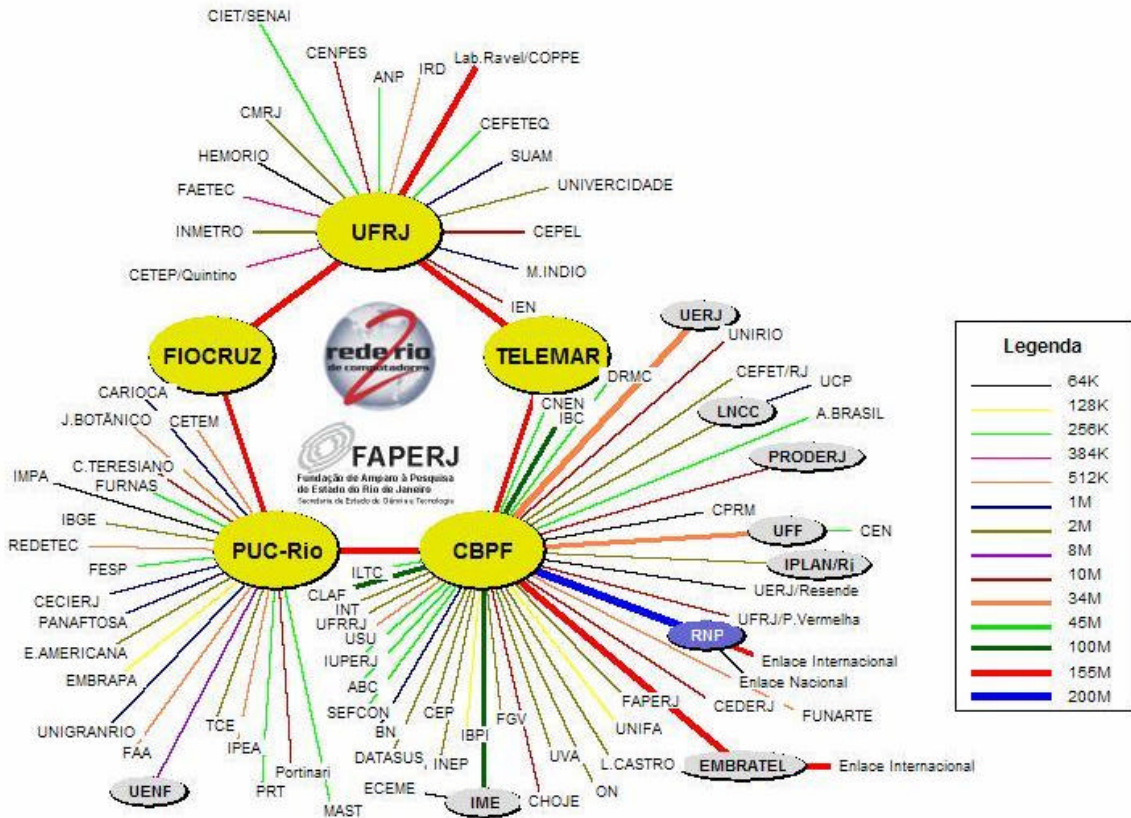
O Rio de Janeiro ocupa, hoje, um lugar de destaque no país como produtor de tecnologia e conhecimento. Nos últimos anos, os recursos investidos nestas áreas vêm crescendo significativamente.

A Rederio de computadores, inaugurada em 1992, é um dos principais instrumentos de desenvolvimento científico do estado do Rio de Janeiro interconectando os mais avançados centros de pesquisa do país, sediados nas universidades e nas empresas públicas e privadas do Estado.

Inicialmente, a Rederio interligava três instituições principais: a UFRJ - Universidade Federal do Rio de Janeiro, o LNCC – Laboratório Nacional de Computação Científica e a PUC-RJ - Pontifícia Universidade Católica do Rio de Janeiro, que funcionavam como ponto de troca de tráfego, através de um backbone de 256Kbps. Além dessas três principais, reunia apenas 7 outras instituições localizadas na região metropolitana do Rio de Janeiro, ou seja, o alcance era bastante limitado.

Oito anos depois, algumas mudanças se fazem notar: o aumento da velocidade de transmissão do backbone, 600 vezes superior a que era utilizada quando da inauguração da rede; a substituição de uma das instituições principais, o LNCC pelo CBPF – Centro Brasileiro de Pesquisas Físicas, por motivo de transferência de cidade da primeira; a inclusão de duas novas organizações: FIOCRUZ - Fundação Oswaldo Cruz e TELEMAR e a ampliação do alcance da rede, agora beneficiando mais de noventa instituições, incluindo aquelas que constituem a Rede Governo do estado. Abaixo é apresentado um mapa da distribuição dos pontos de presença da Rederio na região metropolitana do Rio de Janeiro, mostrando suas ligações através de fibra ótica.





24. Estrutura física da Rederío

A maioria das conexões com os associados é estabelecida a 256 Kbps, através de circuitos digitais dedicados providos pela TELEMAR. Algumas ligações usam velocidades e meios diferentes. É o caso do LNCC e do Campus da Praia Vermelha da UFRJ, que acessam o PoP CBPF a 10 Mbps, e da UENF – Universidade Estadual do Norte Fluminense, cujo enlace com o Teleporto é 512 Kbps. Já a UFRRJ - Universidade Federal Rural do Rio de Janeiro e a UFF - Universidade Federal Fluminense se conectam à UFRJ via rádio. As demais ligações são do tipo LPCD – linha privada comutada digital.

## CEO

A Coordenação de Engenharia Operacional - CEO é responsável pelo gerenciamento do Backbone, de todas conexões e dos serviços oferecidos pela Rede-

Rio. A CEO monitora a rede, emite relatórios periódicos com estatísticas apontando a taxa de utilização de cada enlace além de oferecer serviço de suporte. Para isso, utilizam-se estações de trabalho e equipamentos de rede, mantidos no ar 24 horas por dia, operando programas de gerenciamento capazes de detectar os mais variados tipos de problemas, através de um monitoramento contínuo das linhas dedicadas.

Ao pessoal técnico do CEO cabe gerenciar estes equipamentos, colocar em funcionamento novas tecnologias e demandas da comunidade, garantir a segurança dos equipamentos da rede e acionar as operadoras de comunicação em eventuais problemas com as conexões as diversas instituições consorciadas.

A CEO fica localizada na sede do CBPF - Centro Brasileiro de Pesquisas Físicas e sua equipe é formada por técnicos da CAT - Coordenação de Atividades Técnicas. A FAPERJ – Fundação de Amparo a Pesquisa do Estado do Rio de Janeiro, é o órgão responsável pelo financiamento da Rede Rio e a SECTI-RJ - Secretaria de Estado de Ciência, Tecnologia e Inovação pela sua coordenação.

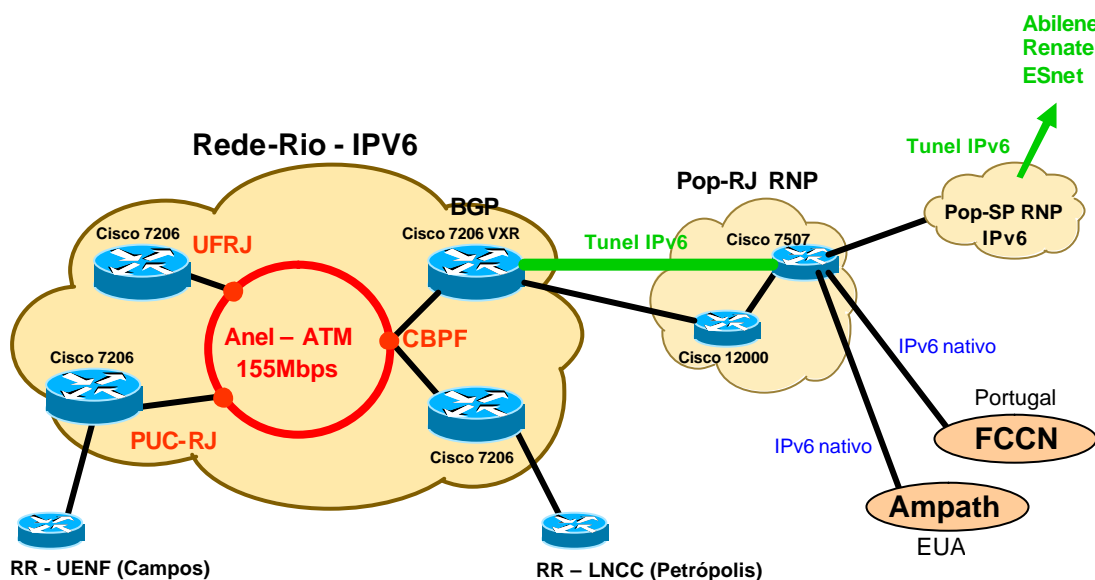
## **8.1 O Projeto IPv6**

O Projeto IPv6 da Rederio iniciou em julho de 2003, com objetivo de participar do projeto da RNP – Rede Nacional de Pesquisa, chamado Br6Bone – <http://www.6bone.rnp.br>. A partir deste momento foi iniciado um estudo sobre as características do protocolo IPv6, as necessidades de um *backbone* metropolitano como a Rederio, investir no estudo e capacitação de profissionais para tal objetivo e a viabilidade de tal protocolo ser efetivamente ser implementado no *backbone*.

Após esta etapa de estudos foi iniciado uma outra relativa a execução de experimentos capazes de simular a utilização deste protocolo em nosso *backbone*. Estes experimentos foram vistos em detalhes na seção anterior. Ao fim da etapa de

experimentação a Rederio entrou em contato com a RNP, para em fim fazer parte do projeto Br6Bone.

Em Setembro de 2003 a Rederio recebeu de forma provisória um bloco de endereços IPv6 cedidos pela RNP – (2001:12F0:04C0::/42). Hoje a Rederio possui todo seu *backbone* metropolitano (pontos de presença – UFRJ, PUC-RJ, CBPF) configurados com o Protocolo IPv6 e tendo como saída nacional e internacional um enlace com a RNP, através de *tunnel* Ipv6 sobre IPv4. Abaixo é apresentado um esquema do atual estágio do *backbone* IPv6 da Rederio.



25. Atual estágio do Backbone Ipv6 da Rederio

A Rederio oferece a qualquer de seus afiliados, conectados à rede IPv4, a possibilidade conexão ao *backbone* IPv6, através da adoção de blocos de endereços com prefixos /48, tendo já conquistado duas instituições interessadas e que já estão conectadas à rede IPv6:

- CBPF – Centro Brasileiro de Pesquisas Físicas – 2001:12F0:04C0::/48
- UFRJ – Universidade Federal do Rio de Janeiro – 2001:12F0:4C1::/48

Porém como a Rederio é um AS – *Autonomus System*, é necessário que possua seu próprio bloco de endereços. Desta forma existem algumas etapas a serem conquistadas:

1. obtenção do Bloco de endereços IPv6 junto ao órgão de registro responsável, que no caso da América Latina é o LACNIC - *Latin American and Caribbean Internet Address Registry*;
2. migração de todo o Backbone metropolitano da Rederio para os novos endereços IPv6, fornecidos pelo LACNIC;
3. configuração do protocolo BGP – *Border Gateway Protocol* no roteador de borda da Rederio para troca de Tráfego com a RNP;
4. organização do Plano de endereçamento e regras de utilização do Bloco de endereços IPv6 da Rederio.

Após estas etapas iniciais o projeto IPv6 tem como objetivo buscar novas instituições interessadas a participar, além de oferecer novos serviços como:

- Serviço de QoS – *Quality of Server* para tráfego multimídia;
- Planejamento para otimização e economia de tráfego com a utilização de endereços *Multicast*;
- Sistema Nativo de Autoconfiguração de equipamento para usuários finais;
- Sistemas de vídeo-conferência multiponto sobre IPv6 com utilização do software Isabel

## 9. Conclusão

Esse trabalho buscou primeiramente evidenciar a importância do protocolo IPv6 e a sua adoção por parte dos usuários da Internet, mostrando os principais problemas existentes no protocolo IPv4, como a limitação do número de endereços, e as possíveis soluções advindas através do IPv6.

Outra preocupação deste trabalho foi descrever os passos executados quando da implementação do protocolo IPv6 no *backbone* da Rederio, através da apresentação de diversos experimentos relatados na seção 7 e o andamento do Projeto IPv6, mostrando o atual estágio do *backbone* IPv6 da Rederio e os passos a serem seguidos para a continuação deste Projeto, que foram apresentados na seção 8.

Como foi visto ao longo deste trabalho, começa a existir uma necessidade de que os usuários da Internet adotem o IPv6, pois num futuro próximo o IPv4 não será capaz de cobrir com eficiência as dimensões alcançadas pela Internet, nem garantir novos serviços que utilizem transmissão multimídia, como vídeo sob demanda, vídeo-conferência, telefonia IP e transmissões de TV.

A segurança também é uma preocupação do IPv6, não só através de autenticação de pacotes ou uso de artifícios como IPSec, mas também através de um sistema de endereçamento Global, onde cada usuário terá um endereço único, que poderá ser usado como uma identificação dentro da rede. Para isto, os provedores e empresas poderão cadastrar seus usuários e vincula-los à seus endereços IPv6, buscando transparecer as operações feitas através da rede.

Porém a utilização definitiva do IPv6, com a substituição total do IPv4, ainda pode demorar alguns anos. Estudos apontam que esta etapa só será atingida no final da próxima década, com prazo previsto para o ano de 2019. Da mesma forma, a utilização

do protocolo IPv6 já passa a ser meta conquistada por grandes empresas, como a Cisco Systems, que definiu o ano de 2005 como prazo final para que todo o seu *backbone*, espalhado pelo mundo, utilize IPv6.

Apesar destes prazos distantes é muito importante que empresas e instituições que trabalham com Internet, comecem estudos para utilização do IPv6, não só para se preparar para o futuro, mas também para buscar novas características que devam ser incorporadas a este protocolo. Desde sua criação, já houveram inúmeras modificações no IPv6, que fizeram dele um protocolo mais robusto e confiável, sem perder a compatibilidade com os outras camadas da pilha TCP/IP, o que é importante para a sua afirmação na comunidade da Internet.

Ainda existem pontos de dificuldade, como compatibilidade com equipamentos antigos e aplicações que utilizem, de forma simples, o IPv6. Desde o início dos testes, existiram problemas desta ordem, pois os sistemas operacionais não tratam ainda o IPv6 com as mesmas facilidades que tratam o IPv4, sendo que em alguns casos esses sistemas nem funcionam de forma eficiente. Isso talvez seja o ponto mais importante a ser tratado neste momento, para que o IPv6 seja adotado por cada vez mais usuários.

## 10. Bibliografia

### 1. Livros e textos:

- COMER, Douglas E., *Redes de Computadores e Internet*, editora Prentice Hall, New Jersey 1999.
- HUITEMA, Christian, *IPv6: The New Internet Protocol*, 1ª edição, editora Prentice Hall, New Jersey 1996.
- NAUGLE, Matthew, *Guia Ilustrado do TCP/IP*, editora Berkeley, São Paulo 2001.
- TANEMBAUM, Andrew, *Computer Networks*, 3ª edição, editora Prentice Hall, New Jersey 1996.

### 3. RFC's:

- **RFC 3513:** HINDEN, Robert M. e DEERING, Stephen E., *IP Version 6 Addressing Architecture*, 2003.
- **RFC 2374:** HINDEN, Robert M., O'DELL, Mike e DEERING, Stephen E., *An IPv6 Aggregatable Global Unicast Address Format*, 1998.
- **RFC 2462:** THOMSON, Susan e NARTEN, Thomas, *IPv6 Stateless Address Autoconfiguration*, 1998.
- **RFC 2461:** NARTEN, Thomas, NORDMARK, Erik e SIMPSON, William Allen, *Neighbor Discovery for IP Version 6 (IPv6)*, 1998.
- **RFC 2460:** HINDEN, Robert M. e DEERING, Stephen E., *Internet Protocol, Version 6 (IPv6) Specification*, 1998.

## 2. Web Pages:

- CEO/RNP - Projeto Brasileiro de IPv6, <http://www.6bone.rnp.br/>.
- CISCO SYSTEMS, *Implementing Basic Connectivity for IPv6*, [http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6\\_c/sa\\_bconn.pdf](http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/sa_bconn.pdf), 2004.
- NED, Frank, *A Nova Geração de Protocolos IP*, RNP-News Generation Vol.2/No.8, <http://www.rnp.br/newsgen/9811/intr-ipv6.html>, 1998.
- PEREIRA, Luiz Gustavo, *Tutorial IPv6*, <http://penta2.ufrgs.br/redes296/ipv6/>
- SILVA, Adailton e FARIA, Marcel. *Hierarquia de Endereços IPv6*. RNP-News Generation Vol5/No2, [http://www.rnp.br/newsgen/0103/end\\_ipv6.html?ipv6](http://www.rnp.br/newsgen/0103/end_ipv6.html?ipv6), 2001.
- SILVA, Adailton, *O IPV6 na RNP e no Brasil*. RNP-News Generaton Vol.2/No.7, <http://www.rnp.br/newsgen/9809/exp-ipv6.html?ipv6>, 1998.
- SAMMANASU, Casimir, *The ABCs of IP Version 6*, <http://www.cisco.com/go/abc> 2002.
- VIEIRA, Sandro e CARDADOR, Wellington, *Nova Geração de Protocolos IP*, <http://proenca.uel.br/curso-redes-especializacao/2001-uel/trab-03/equipe-10/index.html>, 2001.