UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

INTERLIGAÇÃO DA REDUNB AO BR6BONE

HENRIQUE DE OLIVEIRA ANDRADE
ROBSON LOPES DA GAMA JÚNIOR

ORIENTADORA: CLÁUDIA JACY BARENCO

PROJETO FINAL DE GRADUAÇÃO EM ENGENHARIA DE REDES DE COMUNICAÇÃO

PUBLICAÇÃO: UnB.LabRedes.PFG09/2002

BRASÍLIA / DF: AGOSTO/2002

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

INTERLIGAÇÃO DA REDUNB AO BR6BONE

HENRIQUE DE OLIVEIRA ANDRADE
ROBSON LOPES DA GAMA JÚNIOR

PROJETO FINAL DE GRADUAÇÃO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO.

APROVADA POR:

CLÁUDIA JACY BARENCO, Doutora, UnB
(ORIENTADORA)

RAFAEL TIMÓTEO DE SOUSA JUNIOR, Doutor, UnB
(EXAMINADOR INTERNO)

DATA: BRASÍLIA/DF, 30 DE AGOSTO DE 2002.

FICHA CATALOGRÁFICA

ANDRADE, HENRIQUE DE OLIVEIRA
LOPES, ROBSON DA GAMA JÚNIOR
Interligação da RedUnB ao Br6Bone [Distrito Federal] 2002.
xx, 113p., 297 mm (ENE/FT/UnB, ENGENHEIRO, Engenharia Elétrica, 2002).

Projeto Final de Graduação – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. IPv6 2. 6BONE
3. IPng

I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

ANDRADE, H. O. e LOPES, R. L. (2002) Interligação da RedUnB ao Br6Bone. Projeto Final de Graduação, Publicação 09/2002, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 113p.

CESSÃO DE DIREITOS

Henrique de Oliveira Andrade
SHIS QI 21 Conj. 5 Casa 14
CEP 71655-250 – Brasília – DF - Brasil

Robson Lopes da Gama Júnior
SQN 115 Bl. “C” Apto. 204
CEP 70772-030 – Brasília – DF - Brasil

A Deus que me deu o Dom da vida. A Luciana, reconhecendo os muitos momentos sacrificados do nosso convívio durante todo o curso. Aos meus pais e amigos que

sempre me apoiaram nessa etapa da caminhada.
Robson Lopes da Gama Jr

Aos meus pais, irmãos e amigos pelo constante apoio nas horas difíceis e compreensão nos momentos de ausência. Grande parte dessa vitória alcançada hoje é mérito deles.
Henrique de Oliveira Andrade

AGRADECIMENTOS

À Prof. Dra. Cláudia Jacy Barenco, do Curso de Engenharia de Redes de Comunicação - Departamento de Engenharia Elétrica, pelo empenho demonstrado no decorrer deste trabalho.
Ao Prof. Dr. Rafael Timóteo de Sousa Júnior, pelo apoio essencial no desenvolvimento deste projeto e durante todo o curso de graduação.
Aos bolsistas do Laboratório de Engenharia de Redes de Comunicação – LabRedes da Universidade de Brasília, em especial para o Robson Albuquerque, pelas conversas enriquecedoras, ajuda em diversos aspectos, colaboração e amizade.
A todos, os nossos sinceros agradecimentos.
O presente trabalho foi realizado com o apoio da RNP – Rede Nacional de Pesquisa, um Programa Prioritário do MCT - Ministério da Ciência e Tecnologia, apoiado e executado pelo CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, cuja missão principal é operar um serviço de backbone Internet voltado à comunidade de ensino e de pesquisa do Brasil. Foram utilizados recursos do Projeto REMAV Infovia de Brasília, que também é patrocinado pelo CNPq e a RNP.

RESUMO

O trabalho descrito nesta dissertação objetiva aprofundar o leitor no conhecimento do protocolo Internet de nova geração, o IPv6. Um estudo teórico, os passos para se montar um laboratório com suporte ao IPv6 e a configuração de um túnel com a Rede Nacional de Pesquisa – RNP para acesso ao 6Bone são temas deste trabalho.

ABSTRACT

The work described in this thesis aims a further study of the Internet next generation protocol, the IPv6. A theorical aproach, the steps needed to build a laboratory enviroment that supports the IPv6 and the setup of a tunnel linking the university to the National Research Network - RNP to access the 6Bone are topics of this paper.

ÍNDICE

ÍNDICE DE TABELAS

ÍNDICE DE FIGURAS

1. INTRODUÇÃO

A evolução da tecnologia de TCP/IP está integrada à evolução global da utilização das redes no mundo por várias razões. Primeiramente, sendo a Internet a maior integração em redes TCP/IP, muitos problemas relacionados ao crescimento surgem na Internet antes que venham à tona em outras interligações em redes TCP/IP. Em segundo lugar, os fundos para pesquisa e engenharia de TCP/IP provêm de companhias e entidades governamentais que usam a Internet operacional tendendo, portanto, a financiar projetos que causam impacto na Internet. Terceiro, a maioria dos pesquisadores engajados no trabalho de TCP/IP têm conexões com a Internet e usam-a diariamente. Desse modo, eles têm motivação imediata para solucionar problemas que vão melhorar seu serviço e ampliar sua funcionalidade.

A versão 4 do protocolo Internet (IPv4) fornece o mecanismo básico de comunicação da pilha TCP/IP e da Internet global. Essa versão permaneceu quase inalterada desde o seu início, no final da década de 70. Sua longevidade mostra que o projeto é robusto, facilmente implementado, interoperável e poderoso. Desde quando o IPv4 foi projetado, o desempenho do processador aumentou mais de duas ordens de magnitude, os tamanhos típicos de memória aumentaram 32 vezes, a largura de banda de rede do backbone da Internet cresceu 800 vezes, tecnologias de rede local afloraram e o número de hosts na internet cresceu em escala de 4 milhões.

Dessa forma surgiram alguns fatores que não foram antecipados por esse *design* inicial:

- *crescimento exponencial da Internet influindo em carência de endereços IPv4*

Os endereços IPv4 têm se tornado escassos, levando algumas organizações a usar o NAT (*Network Address Translator*) para mapear múltiplos endereços privativos para um simples endereço IP público. Apesar dos NATs promoverem o re-uso dos endereços privados, eles não suportam padronizações de segurança baseadas na camada de rede ou o mapeamento correto de todos os protocolos de alto nível, podendo também criar problemas quando conectam duas organizações que utilizam o espaço de endereçamentos privados. Além do mais, o crescimento de dispositivos conectados à Internet e suas novas aplicações, garantem que os endereços IPv4 públicos se esgotarão rapidamente.

- *crescimento da rede e a habilidade dos roteadores do backbone Internet em manter longas tabelas de rotas*

Devido à maneira como as IDs de rede IPv4 têm sido alocadas, hoje existem por volta de 70.000 rotas na tabela de roteamento dos roteadores do *backbone* Internet. A infraestrutura de roteamento IPv4 atual é uma combinação do que chamamos de roteamentos "*flat*" e "*hierarchical*". O último diz respeito ao uso do endereçamento seguindo algum tipo de hierarquia, por exemplo, os dois primeiros octetos para identificar sistemas autônomos e o restante um *site* local, ao contrário do esquema *flat* usado em redes locais e que não trazem este tipo de informação nos endereços.

- *necessidade de configuração simples*

A maior parte das implementações atuais IPv4 ainda devem ser realizadas

manualmente ou utilizar um protocolo de configuração de endereçamento dinâmico como o DHCP (*Dynamic Host Configuration Protocol*). Com um número grande de computadores e de equipamentos utilizando IP, surge a necessidade de uma otimização e simplicidade na configuração de endereços e de outros ajustes que não dão confiança à administração de uma infraestrutura DHCP.

- *a necessidade de melhor suporte para entrega em tempo real de dados — também conhecida como Qualidade de Serviço (QoS)*

O suporte de tráfego em tempo real se apóia no campo *Type of Service* (TOS) do cabeçalho IPv4 e na identificação do *payload*, utilizando tipicamente uma porta UDP ou TCP. Infelizmente, o campo TOS do IPv4 tem funcionalidade limitada e durante muito tempo houveram várias interpretações locais. Além disso, a identificação de *payload* utilizando portas TCP e UDP não é possível quando o *payload* do pacote está encriptado.

Direcionado a esses conceitos, a *Internet Engineering Task Force* (IETF) desenvolveu um conjunto de protocolos e de padrões conhecidos como IP versão 6 (IPv6). Essa nova versão, previamente chamada *IP-The Next Generation* (IPng), incorpora o conceitos de vários métodos propostos para atualização do protoloco IPv4. Seu *design* foi intencionalmente o alvo para minimizar o impacto nas camadas acima e abaixo dos protocolos, evitando a adição randômica de novas características.

1.1. CARACTERÍSTICAS DO IPV6

O protocolo IPv6 proposto mantém muitas das características que contribuíram para o sucesso do IPv4. Por exemplo, o IPv6 ainda aceita entrega sem conexão (isto é, permite que cada datagrama seja roteado independentemente), permite que o transmissor escolha o tamanho de um datagrama e requer que o transmissor especifique o número máximo de passos da rota que um datagrama pode fazer antes de ser concluído. O IPv6 também retém a maioria dos conceitos fornecidos pelas opções do IPv4, inclusive os recursos para a fragmentação e roteamento da origem.

A despeito de muitas semelhanças conceituais, o IPv6 muda a maioria dos detalhes do protocolo. Por exemplo, o IPv6 usa endereços maiores e acrescenta algumas características novas. Mais importante, revisa completamente o formato de datagrama, substituindo o campo de opções de comprimento variável do IPv4 por uma série de cabeçalhos de formato fixo.

As mudanças introduzidas pelo IPv6 podem ser agrupadas em cinco categorias:

Endereços maiores
O novo tamanho de endereço é a mudança mais visível. O IPv6 quadruplica o tamanho de um endereço de IPv4, de 32 para 128 bits. O espaço de endereço de IPv6 é tão grande que não pode ser consumido em futuro previsível.

Formato flexível de cabeçalho
O IPv6 usa um formato de datagrama inteiramente novo e incompatível. Ao contrário

do IPv4, que usa um cabeçalho de datagrama de formato fixo onde todos os campos, exceto o de opções, ocupam um número fixo de octetos com um deslocamento fixo, o IPv6 usa um conjunto de cabeçalhos opcionais.

Opções aprimoradas
Como o IPv4, o IPv6 permite que o datagrama inclua informações de controle opcionais. Ele inclui novas opções que oferecem recursos adicionais não disponíveis no IPv4.

Melhor suporte para QoS
Novos campos no cabeçalho IPv6 permitem que os roteadores identifiquem e ajam de maneira especial com pacotes que pertencem a um determinado fluxo. Como o tráfego é identificado no cabeçalho IPv6 o suporte ao QoS funciona mesmo quando o *payload* do pacote está encriptado pelo IPsec.

Provisão para extensão de protocolos
Talvez a mudança mais significativa no IPv6 seja uma transição de um protocolo que especifica inteiramente os detalhes, para um protocolo que pode permitir recursos adicionais. A capacidade de extensão tem o potencial para permitir que a IETF adapte o protocolo a mudanças no hardware de rede considerado ou a novos aplicativos.

1.2. DIFERENÇAS ENTRE IPV4 E IPV6:

A tabela 1 busca ilustrar mais detalhadamente as diferenças entre os protocolos IPv4 e IPv6:

Tabela 1 - Diferenças IPv4 e IPv6

IPv4	IPv6
Endereços de origem e destino de 32 bits (4 bytes)	Endereços de destino e origem com 128 bits (16 bytes)
Suporte IPsec opcional.	Suporte IPsec requerido.
Sem identificação de *payload* para "QoS handling by routers".	Identificação de *payload* para "QoS handling by routers" é incluída no cabeçalho IPv6 por meio do campo Flow Label.
A fragmentação é suportada por ambos roteadores e hosts.	A fragmentação não é recomendada para os roteadores e deve ser realizada nos *hosts* de envio.
Cabeçalho com *checksum*.	Cabeçalho não inclui um *checksum*.
Cabeçalho com campo para opções.	Todos os dados opcionais são movidos para os cabeçalhos de extensão IPv6.
O Address Resolution Protocol (ARP) utiliza *broadcast* de datagramas ARP Request para determinar um endereço IPv4 até o endereço da camada de enlace.	Os datagramas ARP Request são substituídos por mensagens multicast de Neighbor Solicitation.
O Internet Group Management Protocol (IGMP) é usado para gerenciar um membro do grupo multicast da subrede local.	O IGMP é substituído por mensagens Multicast Listener Discovery (MLD).

O ICMP Router Discovery é usado para determinar o endereço IPv4 do melhor *gateway default* e é opcional.	O ICMPv4 Router Discovery é substituído pelo ICMPv6 Router Solicitation e pelas mensagens Router Advertisement.
Endereços broadcast são usados para enviar tráfego a todos os nós da subrede.	Não existem endereços broadcast no IPv6. Ao invés, um tipo de multicast link-local é usado.
Deve ser configurado manualmente ou através de DHCP.	Não requer esse tipo de cuidado.
Utiliza recursos no DNS Domain Name System (DNS) para mapear nomes de host para os endereços IPv4.	Utiliza recursos no DNS Domain Name System (DNS) para mapear nomes de host para os endereços IPv6.

Cabeçalho IPv4:

A figura 1-1 mostra o cabeçalho IPv4:

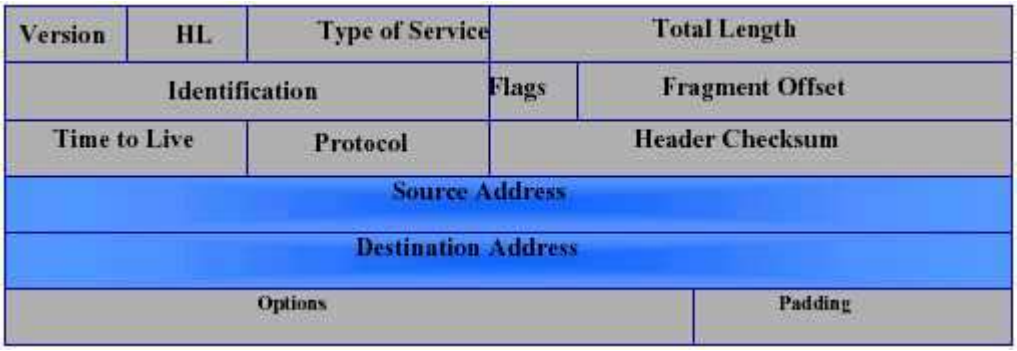


Figura 1-1 Cabeçalho IPv4

Os campos do cabeçalho IPv4 são os seguintes:
Version – Indica a versão do IP e é setado em 4. O tamanho desse campo é de 4 bits.
Internet Header Length – Indica o número de blocos de 4-bytes no cabeçalho IP. O comprimento do campo é de 4 bits. Como um cabeçalho IP possui um mínimo de 20 bytes em tamanho, o menor valor para o campo *Internet Header Length* (IHL) é 5. As opções IP podem estender o tamanho mínimo do cabeçalho IP em incrementos de 4 bytes. Caso uma opção IP não utilize todos os 4 bytes do campo de opções IP, os bits resultantes são completados com 0's, fazendo com que o cabeçalho IP completo tenha um tamanho integral de 32 bits (4 bytes). O tamanho máximo do cabeçalho IP incluindo opções é de 60 bytes (15*4).
Type of Service – Indica o serviço desejado para o pacote que vai trafegar na rede através dos roteadores. Seu tamanho é de 8 bits, que englobam bits de procedência, atraso, vazão, e características de confiabilidade.
Total Length – Indica o tamanho total do pacote IP (IP *header* + IP *payload*). Seu comprimento é de 16 bits, que indicam um pacote IP acima de 65,535 de tamanho.
Identification – Identifica o pacote IP específico. Possui um campo de 16 bits. O campo de identificação é selecionado pela fonte de origem do pacote IP. Quando o pacote é fragmentado, todos os fragmentos permanecem com um valor no campo de identificação fazendo com que a fonte de destino seja capaz de reconstruir o pacote completo.
Flags – Identifica *flags* para o processo de fragmentação. Seu campo é de 3 bits,

contudo, apenas 2 bits foram definidos para o uso. Existem dois *flags*—um indica se o pacote deve ser fragmentado e outro indica se mais fragmentos seguem o pacote recebido.
Fragment Offset – Indica a posição do fragmento quando relativo ao *payload* original. Esse campo possui um comprimento total de 13 bits.
Time to Live – Indica o número máximo de links que o pacote IP pode passar antes de ser descartado. Seu tamanho é de 8 bits. O campo *Time-to-Live* (TTL) foi originalmente utilizado como um contador de tempo com o qual o roteador determinava o comprimento do tempo requerido (em segundos) para encaminhar o pacote IP, decrementando o TTL. Os roteadores modernos quase todos, encaminham o pacote IP em menos de 1 segundo e são obrigados pela RFC 791 à decrementar o TTL em pelo menos 1. Então, o TTL se torna um contador máximo de links com o valor original setado pela origem. Quando o TTL se iguala a 0, o pacote é descartado e a mensagem "ICMP Time Expired" é enviada para o endereço IP que solicitou a entrega.
Protocol – Identifica o protocolo da camada superior. O campo tem comprimento igual a 8 bits.
Header Checksum – Provê um *checksum* apenas do cabeçalho IP. Possui um tamanho total de 16 bits. O *payload* IP não é incluído no cálculo do *checksum*, normalmente contém seu próprio *checksum*. Cada nó IP que recebe os pacotes IP verifica o *checksum* do cabeçalho IP e silenciosamente descarta o pacote IP quando verifica alguma falha. Quando um roteador encaminha um pacote IP deve decrementar o TTL. Ou seja, o *checksum* do cabeçalho é recomputado a cada *hop* entre a origem e destino.
Source Address – Armazena o endereço IP de origem. Esse campo possui um tamanho de 32 bits.
Destination Address – Armazena o endereço IP de destino. Possui um tamanho de 32 bits.
Options – Armazena uma ou mais opções IP. Seu campo é múltiplo de 32 bits. Caso esse campo não seja totalmente utilizado (todos os 32 bits), deve ocorrer um enchimento para que o cabeçalho IP seja um número inteiro de blocos de 4-bytes que pode ser indicado pelo campo *Internet Header Length*.

Estrutura de um pacote IPv6:

A figura 1-2 busca ilustrar a estrutura de um pacote IPv6 e a figura 1-3 ilustra o cabeçalho IPv6:

Figura 1-2 Estrutura de um pacote IPv6

Figura 1-3 O cabeçalho IPv6

Seguem os comentários sobre os campos.
Version – 4 bits são usados para indicar a versão. São setados em 6.
Traffic Class – Indica a classe ou prioridade do pacote IPv6. Possui o tamanho de 8 bits. Esse campo desenvolve funcionalidades similares ao ToS do IPv4. Na RFC 2460, os valores do campo *Traffic Class* não são definidos.
Flow Label – Indica se o pacote pertence a uma seqüência específica de pacotes entre o destinatário e a origem, merecendo especial cuidado pelos roteadores IPv6 intermediários. O tamanho total desse campo é de 20 bits. O *Flow Label* é utilizado em conexões de qualidade de serviço não-default, assim como para as de dados tempo real (voz e vídeo). Por tratamento padrão do roteador, o *Flow Label* é setado em zero. Podem existir múltiplos fluxos de dados entre a origem e o destino, distinguidos por *Flow Labels non-zero* separados.
Payload Length – Indica o comprimento total do *payload*. Seu tamanho máximo é de 16 bits. O campo *Payload Length* inclui as extensões de cabeçalhos e as PDUs de camada superior. Com 16 bits, o *payload* IPv6 acima de 65,535 bytes pode ser indicado. Para comprimentos de *payloads* acima de 65,535 bytes, o campo é setado em 0 e a opção "*Jumbo Payload*" é usada na opção *hop-by-hop* do cabeçalho de extensão.
Next Header – Indica tanto a primeira extensão de cabeçalho (caso presente) ou protocolo na camada superior PDU (assim como TCP, UDP, ou ICMPv6). Tamanho do campo = 8 bits. Quando indicam um protocolo de camada superior acima da camada Internet, os mesmos valores o protocolo IPv4 são usados.
Hop Limit – Indica o número máximo de saltos que o pacote pode efetuar antes de ser descartado. Tem um tamanho de 8 bits. Esse campo é similar ao TTL do IPv4 exceto pelo fato de não haver relação histórica ao montante de tempo (em segundos) que o pacote é enfileirado no roteador.
Source Address – Armazena o endereço IPv6 da origem. Contém 128 bits.
Destination Address – Armazena o endereço IPv6 do destinatário. Contém 128 bits. Em muitos casos o endereço de destino é setado para o endereço do destinatário final. Contudo, se um cabeçalho de roteamento de extensão está presente, o endereço do destino deve ser a próxima interface de rota na lista de rotas da origem.

Valores do campo *Next-Header* (próximo cabeçalho)

A tabela 2 mostra os valores típicos do campo *Next Header* para o cabeçalho IPv6 ou para um cabeçalho de extensão IPv6.

Tabela 2 - Valores do campo Next Header

Valor (decimal)	Cabeçalho (Header)
0	Opções do cabeçalho Hop-by-Hop
6	TCP
17	UDP
41	Cabeçalho encapsulado IPv6
43	Cabeçalho de Roteamento
44	Cabeçalho de fragmentação
46	Protocolo de reserva de recursos
50	Encapsulamento “Security Payload”
51	Cabeçalho de Autenticação
58	ICMPv6
59	Sem next header
60	Cabeçalho de opções de destino

A título de exemplo buscamos atestar a teoria com a prática utilizando um analisador de pacotes para capturar o que trafegava na rede. No nosso caso utilizamos o Ethereal. O pacote seguinte foi capturado durante uma conexão HTTP. Verifica-se a autenticidade da tabela ilustrada anteriormente quando o cabeçalho TCP é descrito com o valor 06.

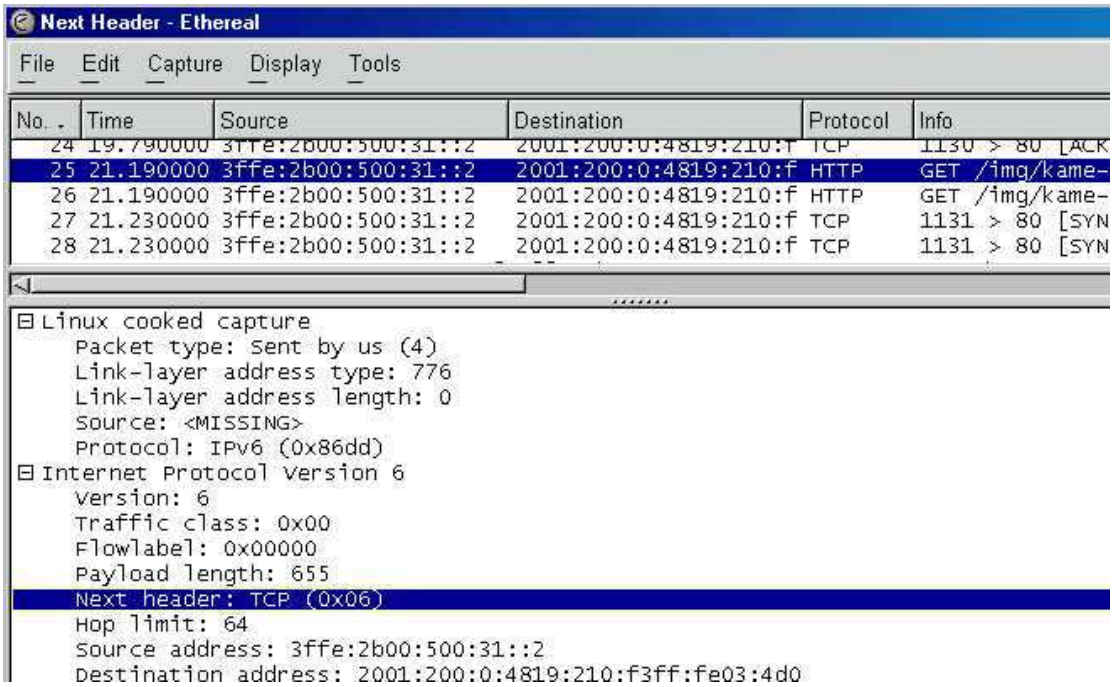


Figura 1-4 Campo Next Header do pacote http

1.2.1. Comparando os cabeçalhos IPv4 e IPv6

A tabela 3 faz uma ilustração comparativa entre os cabeçalhos das versões de IP analisadas.

Tabela 3 - Correspondentes cabeçalho IPv4 x IPv6

Cabeçalho IPv4	Cabeçalho IPv6

Version	Mesmo campo com números de versão diferentes.
Internet Header Length	Removido no IPv6. Isso ocorre porque o cabeçalho IPv6 é sempre fixo = 40 bytes. Cada cabeçalho de extensão é de tamanho fixo ou indica seu próprio tamanho.
Type of Service	Substituído no IPv6 pelo campo *Traffic Class*.
Total Length	Substituído no IPv6 pelo campo *Payload Length*, que indica apenas o comprimento do payload.
Identification	Removido no IPv6. Informações de fragmentação não estão incluídas no cabeçalho IPv6. Estão contidos no "Fragment extension header".
Fragmentation Flags	Removido no IPv6. Informações de fragmentação não estão incluídas no cabeçalho IPv6. Estão contidos no "Fragment extension header".
Fragment Offset	Removido no IPv6. Informações de fragmentação não estão incluídas no cabeçalho IPv6. Estão contidos no "Fragment extension header".
Time to Live	Substituído no IPv6 pelo campo *Hop Limit*.
Protocol	Substituído no IPv6 pelo campo *Next Header*.
Header Checksum	Removido no IPv6. No IPv6, a detecção de erros de bit, para todo o pacote IPv6 é realizada pela camada de comunicação.
Source Address	O mesmo exceto pelo fato de conter 128 bits.
Destination Address	O mesmo exceto pelo fato de conter 128 bits.
Options	Removido no IPv6. As opções do IPv4 são substituídas, no IPv6, pelos cabeçalhos de extensão.

1.3. CABEÇALHOS DE EXTENSÃO DO IPV6

O paradigma de um cabeçalho básico fixo, seguido de um conjunto de cabeçalhos de extensão opcionais, foi escolhido como uma acomodação entre a generalidade e a eficiência. Para ser totalmente geral, o IPv6 precisa incluir mecanismos a fim de aceitar funções como fragmentação, roteamento de origem e autenticação. Entretanto, a opção por alocar campos fixos no cabeçalho de datagrama para todos os mecanismos não é eficaz, porque a maioria dos datagramas não usa todos os mecanismos; o grande tamanho de endereços IPv6 causaria ineficiência. Por exemplo, ao enviar um datagrama através de uma única rede local, um cabeçalho que contenha campos de endereço vazios pode ocupar uma parcela substancial de cada quadro. Mais importante, os projetistas verificaram que ninguém pode prever quais recursos serão necessários.

O paradigma de cabeçalho de extensão do IPv6 funciona de forma semelhante às opções do IPv4 – um transmissor pode optar por escolher quais cabeçalhos de extensão incluir em determinado datagrama e quais omitir. Assim, os cabeçalhos de extensão fornecem flexibilidade máxima.

Resumindo:

Os cabeçalhos de extensão do IPv6 são semelhantes às opções do IPv4. Cada datagrama inclui cabeçalhos de extensão para aqueles recursos que utilizará.

1.3.1. Opções do IPv6

Talvez pareça que os cabeçalhos de extensão do IPv6 substituam completamente as opções do IPv4. Entretanto, os projetistas propõem dois cabeçalhos de extensão adicionais para conciliar qualquer informação variada não incluída em outros cabeçalhos de extensão. Os cabeçalhos adicionais consistem em um *cabeçalho de extensão entre passos da rota* e um *cabeçalho de extensão fim-a-fim*. Como indicam os nomes, os dois cabeçalhos de opção separam o conjunto de opções que deverão ser examinadas em cada passo da rota, do conjunto daquelas que somente são interpretadas no destino.

Embora cada um dos dois cabeçalhos de opção tenha um único código de tipo, ambos os cabeçalhos usam o formato ilustrado a seguir:

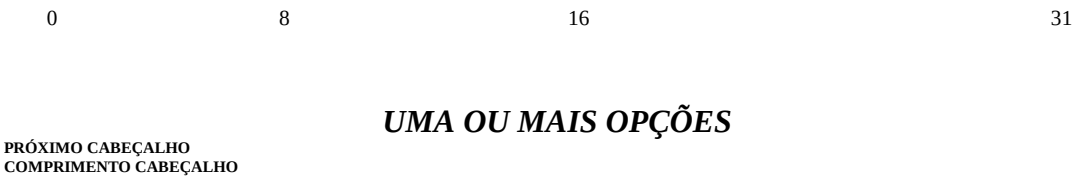


Figura 1-5 Formato geral do cabeçalho de opções

Temos então, representado o formato de um cabeçalho de extensão de opção do IPv6. Tanto o cabeçalho de opção entre passos da rota como o cabeçalho de opção fim-a-fim, usam o mesmo formato. O campo *PRÓXIMO CABEÇALHO*, do cabeçalho anterior, se distingue entre os dois tipos.

Como de costume, o campo *PRÓXIMO CABEÇALHO* fornece o tipo do cabeçalho seguinte. Já que um cabeçalho de opção não tem tamanho fixo, o campo denominado *COMPRIMENTO DO CABEÇALHO* especifica o comprimento total do cabeçalho. A área denominada *UMA OU MAIS OPÇÕES* representa uma sequência de opções individuais. A figura posterior ilustra como cada opção individual é codificada com tipo, comprimento e valor; as opções não são alinhadas ou preenchidas.

Figura 1-6 Formato do campo opções

Como a figura 1-6 indica, as opções do IPv6 seguem a mesma forma que as opções do IPv4. Cada opção começa por um campo *TIPO* de um octeto, seguido de um campo *COMPRIMENTO*. Se a opção requer dados adicionais, nos octetos que compreendem o *VALOR* seguem o campo *COMPRIMENTO*.

Os dois bits de ordem alta de cada campo *TIPO* de opção especificam como um host ou roteador poderá dispor do datagrama se ele não compreender a opção:

Tabela 4 - Campo Tipo de Opção

BITS NO TIPO	SIGNIFICADO
00	Pule esta opção
01	Descarte datagrama; não envie mensagem ICMP
10	Descarte datagrama; envie mensagem ICMP para origem
11	Descarte datagrama; envie ICMP para não-multicast

1.4. ANÁLISE DO DATAGRAMA DO IPV6

Cada cabeçalho básico e de extensão possui um campo PRÓXIMO CABEÇALHO. Os softwares dos roteadores intermediários e dos destinos finais que precisam processar o datagrama devem usar o valor no campo PRÓXIMO CABEÇALHO de cada cabeçalho, para analisar o datagrama. Para extrair todas as informações de cabeçalho de datagrama do IPv6, é necessária uma pesquisa seqüencial através dos cabeçalhos. Por exemplo, a figura 1-7 mostra os campos PRÓXIMO CABEÇALHO de três datagramas que contém nenhum (zero), um e dois cabeçalhos de extensão.

Naturalmente, analisar um datagrama do IPv6 que tem apenas um cabeçalho básico e dados é tão eficiente quanto analisar um datagrama do IPv4. Mais ainda, os roteadores intermediários raramente precisam processar todos os cabeçalhos de extensão.

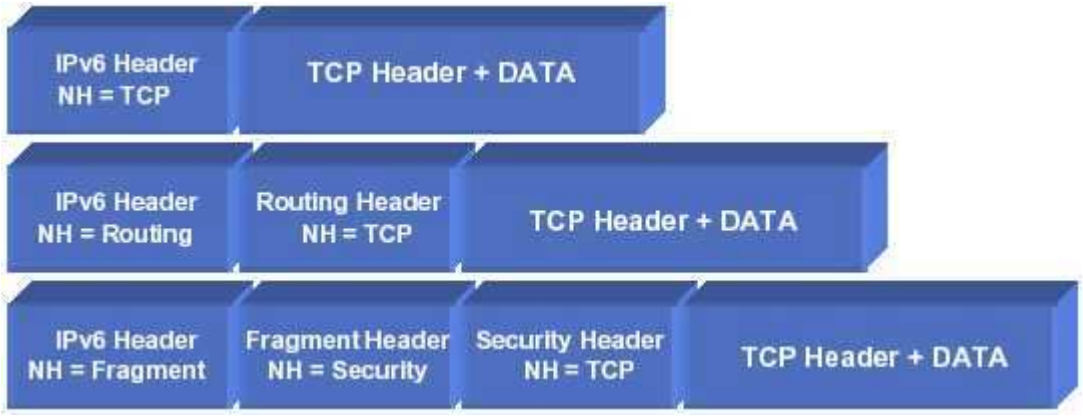


Figura 1-7 Exemplo de pacotes IPv6 com cabeçalhos de extensão

1.5. FRAGMENTAÇÃO E REMONTAGEM DO IPV6

Como o IPv4, o IPv6 planeja para que o destino final execute a remontagem do datagrama. Entretanto, os projetistas tomaram uma decisão inusitada sobre a fragmentação. Lembre-se de que o IPv4 requer que um roteador intermediário fragmente qualquer datagrama que seja grande demais para a MTU da rede sobre a qual precise viajar. No IPv6, a fragmentação está restrita a própria origem. Antes de enviar tráfego, uma origem precisa executar uma técnica de Descoberta de Caminho MTU para identificar a MTU mínima ao longo do caminho até o destino. Antes de

enviar um datagrama, a origem o fragmenta de tal modo que cada fragmento seja menor do que a MTU do caminho. Assim, a fragmentação é fim-a-fim. Nenhuma fragmentação necessita ocorrer em roteadores intermediários.

O cabeçalho básico do IPv6 não contém campos análogos aos campos usados para fragmentação em um cabeçalho do IPv4. Em vez disso, quando a fragmentação é necessária a origem insere um pequeno cabeçalho de extensão após o cabeçalho básico de cada fragmento. A figura abaixo mostra o conteúdo de um *cabeçalho de extensão de fragmento.*

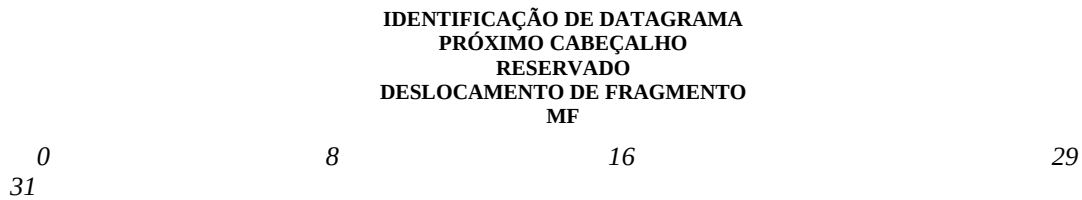


Figura 1-8 Cabeçalho de extensão de fragmento

O IPv6 retém grande parte da fragmentação do IPv4. Cada fragmento precisa ser um múltiplo de 8 octetos. Um bit no campo *MF* marca o último fragmento como o bit de *MAIS FRAGMENTOS* do IPv4; e o campo *IDENTIFICAÇÃO DE DATAGRAMA* transporta uma única ID que o receptor usa para agrupar fragmentos.

Conseqüência da Fragmentação fim-a-fim

A motivação para o uso da fragmentação fim-a-fim reside em sua habilidade para reduzir o *overhead* em roteadores e permitir que cada roteador lide com mais datagramas por unidade de tempo. Na realidade, o *overhead* de CPU, requerido para a fragmentação do IPv4, pode ser significativo – em um roteador convencional, a CPU pode chegar a 100% de utilização se o roteador fragmentar muitos ou todos os datagramas que recebe. Entretanto, a fragmentação fim-a-fim tem uma conseqüência importante: altera um pressuposto fundamental da Internet.

Para melhor se entender essa conseqüência, deve-se lembrar que o IPv4 é projetado para permitir que as rotas mudem a qualquer momento. Isso leva à vantagem da flexibilidade – o tráfego pode ser roteado ao longo de um caminho alternativo, sem interromper os serviços e sem informar a origem e o destino. No IPv6, entretanto, as rotas não podem ser mudadas tão facilmente, pois uma mudança em uma delas pode também mudar a MTU do caminho. Caso a MTU ao longo de uma nova rota, for menor do que a original, podem ocorrer duas situações: o roteador intermediário deve fragmentar o datagrama ou a origem deve ser informada.

Para sanar esse tipo de problema de mudanças de rota que acabam afetando a MTU do caminho, o IPv6 permite que os roteadores tomem certas atitudes. Quando um roteador intermediário precisa fragmentar um datagrama, ele não insere um cabeçalho de extensão de fragmento, nem muda campos do cabeçalho básico. Em vez disso, o roteador intermediário cria um datagrama inteiramente novo que encapsula o datagrama original como dados. O roteador divide o novo datagrama em fragmentos, repetindo o cabeçalho básico e inserindo um cabeçalho de extensão de fragmentos em cada um deles. Finalmente, o roteador envia cada fragmento ao destino final. Ali, o

datagrama original pode ser formado reunindo-se em um datagrama os fragmentos chegados e extraindo-se a parte dos dados. A figura seguinte ilustra o encapsulamento, o datagrama original é “quebrado” em três partes e novos cabeçalhos são gerados.

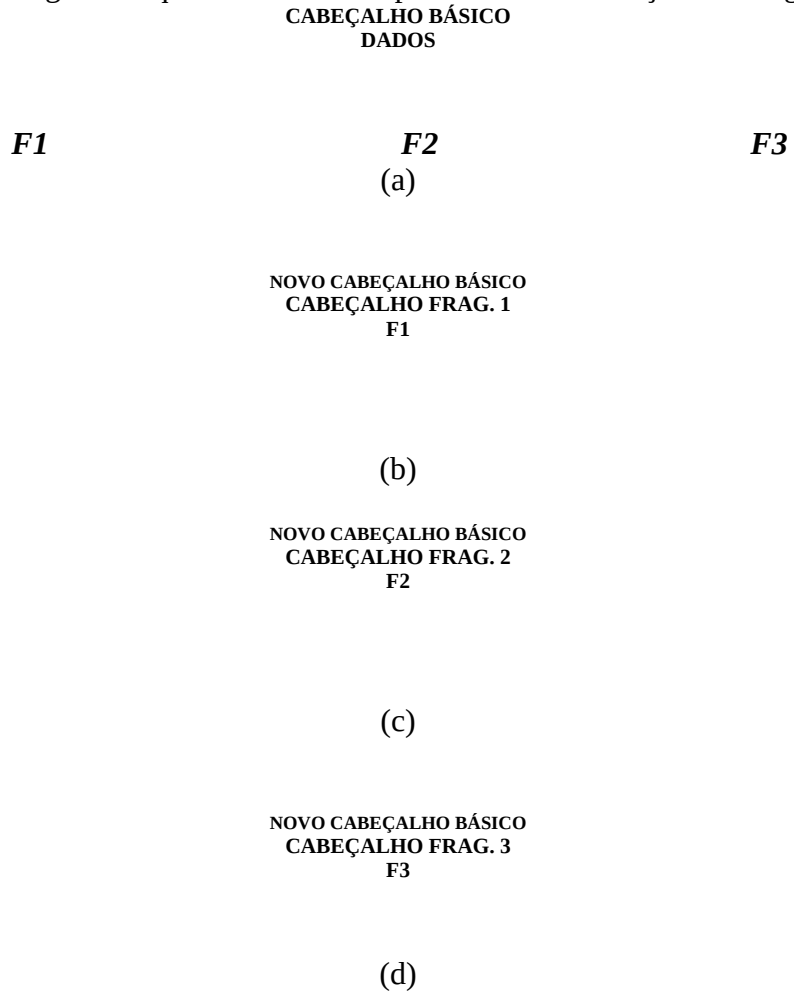


Figura 1-9 Estrutura da fragmentação

1.5.1. IPv6 MTU

O IPv6 requer que a camada de enlace tenha um “suporte mínimo” aos pacotes de tamanho igual a 1280 bytes. As camadas que não suportam devem prover um esquema de fragmentação e remontagem que seja transparente ao IPv6. Nos casos em que as camadas de enlace podem suportar um tamanho configurável MTU, é recomendável que sejam configurados com um tamanho de MTU de, pelo menos, 1500 bytes (o encapsulamento MTU IPv6 Ethernet II).

Um exemplo de MTU configurável é o Maximum Receive Unit (MRU) de um link Point-to-Point Protocol (PPP).

Como no caso do IPv4, o IPv6 provê, para o processo Path MTU Discovery o uso da mensagem ICMPv6 “Packet Too Big” descrita em “Path MTU Discovery”. O Path MTU Discovery permite a transmissão de pacotes IPv6 com tamanho maior que 1280 bytes.

Os hosts de origem IPv6 podem fragmentar *payloads* de camadas de protocolos acima que estejam com o comprimento superior ao MTU da rota a ser seguida. O processo utiliza o chamado “Fragment header” já foi comentado anteriormente. Contudo, o uso da fragmentação IPv6 é altamente desencorajado. Um

nó IPv6 deve ser capaz de remontar e fragmentar pacotes que tenham, pelo menos 1500 bytes.

1.6. ROTEAMENTO DE ORIGEM DO IPV6

O IPv6 retém a capacidade para que um transmissor especifique uma rota de origem livre. Diferentemente do IPv4, em que o roteamento de origem é fornecido por opções, o IPv6 usa um cabeçalho de extensão à parte. Os campos do cabeçalho de roteamento correspondem aos campos de uma opção de rota de origem do IPv4. O cabeçalho contém uma lista de endereços que especificam os roteadores intermediários através dos quais o datagrama deve trafegar. O campo *NÚMERO DE ENDEREÇOS* especifica o número total de endereços da lista e o campo *PRÓXIMO ENDEREÇO* especifica o próximo endereço para o qual o datagrama deverá ser enviado.

1.7. ESQUEMA DE ENDEREÇAMENTO NO IPV6

No IPv6, cada endereço ocupa 16 octetos, quatro vezes o tamanho de um endereço IPv4.

1.7.1. Três tipos básicos de endereço no IPv6

Como o IPv4, o IPv6 associa um endereço a uma conexão de rede específica, não a um computador específico. Assim, atribuições de endereços são semelhantes a IPv4: um roteador IPv6 tem dois ou mais endereços, e um host IPv6 com uma conexão de rede precisa de apenas um endereço. O IPv6 também retém (e estende) a hierarquia de endereço de IPv4 em que um prefixo é atribuído a uma rede física. Entretanto, para facilitar a atribuição e a modificação de endereço, o IPv6 permite que vários prefixos sejam atribuídos a determinada rede, e permite que um determinado computador tenha vários endereços simultâneos atribuídos a determinada interface.

Além de permitir vários endereços simultâneos por conexão de rede, o IPv6 expande e, em alguns casos, unifica endereços especiais do IPv4. Geralmente, um endereço de destino de um datagrama situa-se em uma das seguintes categorias:

Unicast — O endereço de destino especifica um único computador (host ou roteador); o datagrama deverá ser roteado para o destino ao longo do melhor caminho.

Anycast — O destino é um conjunto de computadores que juntos dividem um único prefixo de endereços (ex.: vinculam-se à mesma rede física). O datagrama deverá ser roteado para o grupo ao longo do melhor caminho possível e, então, entregue a exatamente um membro do grupo (ex.: membro mais próximo).

Multicast — O destino é um conjunto de computadores, possivelmente em diversos locais. Uma cópia do datagrama será entregue a cada membro do grupo usando *hardware multicast* ou *broadcast*, conforme o caso.

1.7.2. A dualidade de difusão e multicast

O IPv6 não usa os termos difusão (*broadcast*) ou difusão direta para definir a entrega a todos os computadores de uma rede física ou sub-rede lógica IP. Em vez disso, usa o termo *multicast* e trata difusão como uma forma especial de multicast.

1.7.3. Atribuição do espaço de endereço de IPv6 proposto

A questão de como se compartilhar o espaço de endereço tem gerado muita polêmica. Há dois pontos centrais: como gerenciar as atribuições de endereços e como mapear um endereço para uma rota. O primeiro ponto focaliza o problema prático de delinear uma hierarquia de autoridade. Ao contrário da Internet atual, que usa uma hierarquia de dois níveis de prefixo de rede (atribuído pela autoridade da Internet) e um sufixo de host (atribuídos pelas organizações), o grande espaço de endereço no IPv6 permite uma hierarquia de vários níveis ou várias hierarquias. O segundo ponto focaliza a eficácia computacional. Independente da hierarquia de autoridade que atribui endereços, um roteador deve examinar cada datagrama e escolher o caminho de destino. Para manter baixo o custo de roteadores de alta velocidade, o tempo de processamento exigido para escolher um caminho deve ser mantido curto.

Como mostra a tabela 5, os projetistas do IPv6 propõem a atribuição de classes de endereço de modo semelhante ao esquema usado para IPv4. Embora os primeiros oito bits de um endereço sejam suficientes para identificar seu tipo, o espaço de endereço não é partilhado em seções de igual tamanho.

Tabela 5 – Atribuição de Classes IPv6

Pref. binário	Tipo de endereço	Parte do espaço de endereço
0000 0000	Reservado (compatível com IPv4)	1/256
0000 0001	Reservado	1/256
0000 001	Reservado NSAP	1/128
0000 010	Reservado IPX	1/128
0000 011	Reservado	1/128
0000 100	Reservado	1/128
0000 101	Reservado	1/128
0000 110	Reservado	1/128
0000 111	Reservado	1/128
0001	Reservado	1/16
001	Reservado	1/8
010	Provedor – Unicast atribuído	1/8
011	Reservado	1/8
100	Reservado para uso geográfico	1/8
101	Reservado	1/8
110	Reservado	1/8
1110	Reservado	1/16
1111 0	Reservado	1/32
1111 10	Reservado	1/64
1111 110	Reservado	1/128
1111 1110	Disponível para uso local	1/128
1111 1111	Usado para Multicast	1/128

1.8. MODELO DE ENDEREÇAMENTO

Todos os tipos de endereçamento IPv6 são associados a interfaces, não a nós.

Um endereço unicast IPv6 refere-se a uma simples interface. Como cada interface pertence à um simples nó, qualquer endereço unicast dessas interfaces podem ser utilizadas como um identificador para o nó.

Todas as interfaces requerem, pelo menos, um endereço link-local unicast. Uma simples interface pode também ser associada a múltiplos endereços IPv6 de qualquer tipo (unicast, anycast, e multicast) ou escopo. Endereços unicast com escopo maior que o escopo de enlace não são necessários a interfaces que não são usadas como origem ou destino de qualquer pacote para, ou de, não-vizinhos. Isso algumas vezes é interessante para interfaces ponto-a-ponto.

Existe uma exceção nesse modelo de endereçamento:

Um endereço unicast ou um grupo de endereços unicast podem ser atribuídos à múltiplas interfaces físicas caso a implementação trate as múltiplas interfaces como apenas uma ao apresentá-las à camada de enlace. Isso é usual para load-sharing sobre múltiplas interfaces físicas.

Atualmente, o IPv6 mantém o modelo IPv4 onde um prefixo subrede é associado com um link. Prefixos de múltiplas subredes podem ser associados ao mesmo link.

1.8.1. Representação textual do endereçamento

Existem três formas convencionais de se representar os endereços IPv6 como *strings* de números hexadecimais:

1. A maneira preferida é **x:x:x:x:x:x:x:x**, onde os 'x's são os valores hexadecimais das oito partes de 16-bits do endereço.

Exemplos:

FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

1080:0:0:0:8:800:200C:417A

2. Devido a alguns métodos de alocação de certos estilos de endereços IPv6, é comum observar uma longa seqüência de zeros. Com o propósito de fazer com que a escrita/representação dos endereços fosse facilitada, uma sintaxe especial é utilizada para se comprimir os zeros. O uso de "::" indica múltiplos grupos de zeros de 16-bits. O "::" somente pode aparecer uma vez no endereço.

Como exemplo, observe os endereços seguintes:

1080:0:0:0:8:800:200C:417A	endereço unicast
FF01:0:0:0:0:0:0:101	endereço multicast
0:0:0:0:0:0:0:1	endereço de loopback
0:0:0:0:0:0:0:0	endereço não especificado

Que passam a ser representados assim:

1080::8:800:200C:417A	endereço unicast
FF01::101	endereço multicast

::1	endereço de loopback
::	endereço não especificado

3. Uma forma alternativa que é utilizada em alguns casos, onde existe um ambiente misto de nós IPv4 e IPv6, é o x:x:x:x:x:x:d.d.d.d, onde os 'x's são os valores hexadecimais das oito partições de 16-bits do endereço, e os 'd's são valores decimais das quatro partições de baixa ordem de 8-bits do endereço (representação padrão IPv4).

Como exemplo:

0:0:0:0:0:0:13.1.68.3
0:0:0:0:0:FFFF:129.144.52.38

ou na forma comprimida:
::13.1.68.3
::FFFF:129.144.52.38

1.8.2. Representação Textual dos prefixos de endereços

A representação textual dos prefixos de endereços IPv6 é similar à maneira como são feitas no IPv4 na notação CIDR. Um prefixo de endereço IPv6 é representado pela notação:

Endereço-IPv6/tamanho-prefixo

onde,

Endereço-IPv6	endereço IPv6 em qualquer das notações listadas anteriormente
Tamanho-prefixo	valor decimal especificando os bits contíguos mais à esquerda do endereço incluindo o prefixo

Por exemplo, a seguir estão representadas algumas formas legais de representações do prefixo 60-bits 12AB00000000CD3 (hexadecimal):

12AB:0000:0000:CD30:0000:0000:0000:0000/60
12AB::CD30:0:0:0:0/60
12AB:0:0:CD30::/60

As representações seguintes mostram maneiras incorretas de se escrever o mesmo prefixo:

12AB::CD30/60 o endereço a esquerda da "/" expande para

12AB:0000:0000:0000:0000:000:0000:CD30

12AB::CD3/60 o endereço a esquerda da "/" expande para

12AB:0000:0000:0000:0000:000:0000:0CD3

Quando se escreve ambos, o endereço do nó e o prefixo do endereço do nó

podem ser combinados da seguinte forma:

Endereço do nó	12AB:0:0:CD30:123:4567:89AB:CDEF
E seu número subrede	12AB:0:0:CD30::/60
São abreviados assim:	**12AB:0:0:CD30:123:4567:89AB:CDEF/60**

1.8.3. Identificadores de interfaces

Identificadores de interface em endereços unicast IPv6 são usados para identificar interfaces em um link. Solicita-se que sejam únicas no link. Elas podem também ser únicas em um escopo mais largo.

Em vários casos, os identificadores de interface serão os mesmos dos endereços de camada de enlace. O mesmo identificador de interface pode ser usado em múltiplas interfaces em um único nó.

Formatado de acordo com as regras estabelecidas para os prefixos, os identificadores requeridos são de 64 bits de extensão e construídos no formato EUI-64.

1.8.4. Endereço não-especificado e endereço de Loopback

O endereço 0:0:0:0:0:0:0:0 foi chamado de “endereço não-especificado”. Este endereço nunca deve ser destinado a qualquer nó. Indica ausência de endereço.

O endereço não-especificado não deve ser usado como endereço de destino de pacotes IPv6 ou em cabeçalhos de roteamento IPv6.

Ao endereço específico de unicast 0:0:0:0:0:0:0:1 denomina-se *loopback*.

Este pode ser usado quando um nó deseja enviar um pacote para ele mesmo (testes, por exemplo) e nunca como um endereço de interface física.

1.8.5. Endereços Unicast

Os nós IPv6 podem ter um pequeno ou considerável conhecimento da estrutura interna do endereçamento IPv6, dependendo do papel que o nó desempenha. No mínimo, um nó deve considerar que endereços unicast (incluindo o próprio) não contenha estrutura interna.

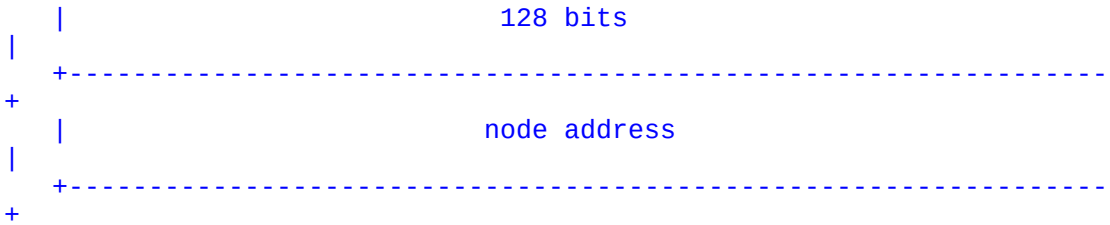
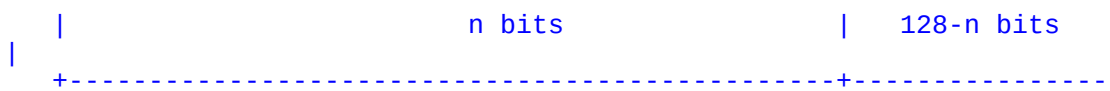


Figura 1-10 Estrutura de endereço unicast

Um host ligeiramente mais sofisticado (mas ainda assim simples), pode ainda estar atento no(s) prefixo(s) para o(s) link(s) em que está anexado, onde endereços diferentes podem ter valores diferentes para n:



```
+
   |                   subrede prefix               | interface ID
|
   +--------------------------------------------------+---------------
+
```

Figura 1-11 Uso de subnets em endereços unicast

Existem ainda hosts mais sofisticados que podem estar atentos a outros limites hierárquicos no endereçamento unicast. Apesar de um roteador simples poder não conhecer da estrutura interna do endereçamento unicast IPv6, geralmente estes terão ciência de um ou mais limites hierárquicos para a operação de roteamento de protocolos. Os limites conhecidos vão diferenciar roteadores entre si, dependendo de qual posição o roteador ocupa na hierarquia de roteamento.

1.8.6. Endereços unicast globais agregáveis

O endereço unicast global agregável é definido na [AGGR].

Esse formato de endereçamento é definido para suportar tanto a atual agregação baseada em provedores quanto o novo tipo de agregação denominado “exchanges”.

A combinação permitirá uma agregação de roteamento eficiente para ambos os sites que conectarem diretamente aos provedores e a quem conecte aos exchanges.

Os sites terão que escolher em conectarem cada tipo de pontos de agregação. Dessa forma, os endereços unicast globais de agregação possuem o seguinte formato:

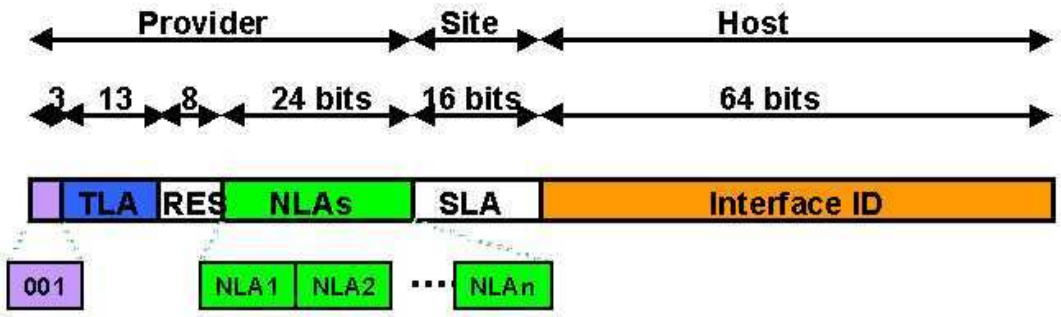


Figura 1-12 Estrutura de endereço unicast global agregável

Onde,

001 **Prefixo do Formato (3 bits) para os endereços unicast globais de agregação**
TLA ID **Identificador Top-Level Aggregation**
RES **Reservado para uso futuro**
NLA ID **Identificador Next-Level Aggregation**
SLA ID **Identificador Site-Level Aggregation**
INTERFACE ID **Identificador de Interface**

Os conteúdos, tamanho de campos e regras de assinaturas estão definidas na [AGGR].

Endereços Unicast IPv6 para uso local

Existem dois tipos de endereços unicast definidos para uso local. São eles: Link-Local e Site-Local.

O endereço local de enlace [link-local] é usado em enlaces únicos, e o site-local em *sites* únicos.

A seguir descreve-se o formato para o endereço Link-Local:

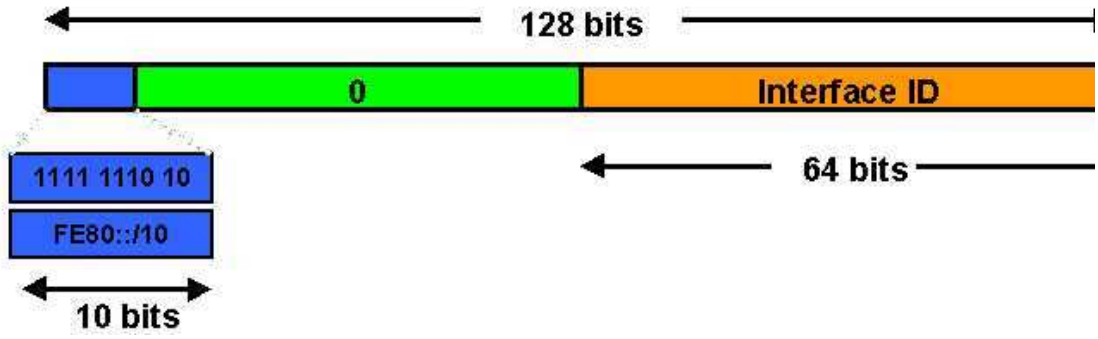


Figura 1-13 Estrutura de endereço Link-Local

Trata-se de endereços ***FE80::<ID da interface>/10.***

Os endereços locais de enlace [Link-Local] são usados em endereçamentos de enlaces únicos para propósitos de autoconfiguração de endereços, descobrimento de vizinhos e situações em que não há roteadores.

Portanto, os roteadores não devem encaminhar para outros links, qualquer pacote com origem ou destino link-local.

Nota: Para se "pingar" um endereço *link-local*, deve-se especificar uma interface que não necessariamente aquela à qual o endereço pertence, por exemplo:

ping6 –I eth0 <end link-local>

Os endereços de site-local possuem o formato seguinte:

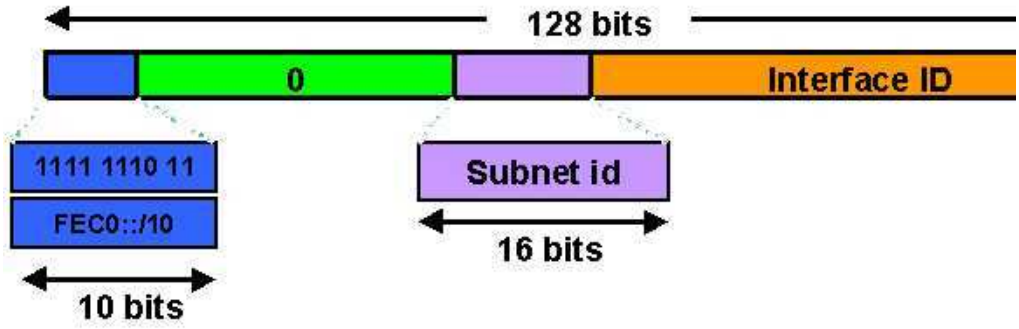


Figura 1-14 Estrutura de endereço Site-Local

Trata-se de endereços ***FEC0::<ID da sub rede>:<ID da interface>/10.***

Esse tipo de endereço permite direcionar mensagens, dentro de um site local ou organização, sem a necessidade de um prefixo global.

Os roteadores, de maneira análoga a anterior, não devem encaminhar para outros sites, qualquer pacote com origem ou destino site-local (seu âmbito está limitado a rede local da organização).

1.8.7. Endereços Anycast

Um endereço anycast IPv6 é aquele designado a mais de uma interface (tipicamente pertencente a nós diferentes), com a propriedade de que um pacote enviado para um endereço anycast é roteado para a interface mais próxima que contenha aquele endereço, de acordo com a medida dos protocolos de roteamento.

Endereços Anycast são alocados de um espaço de endereço unicast, utilizando quaisquer formatos de endereço unicast definidos. Ou seja, os endereços anycast são sintaticamente indistingüíveis dos endereços unicast. Quando um endereço unicast é designado a mais de uma interface, convertendo-se em um endereço anycast, os nós com quais o endereço é designado devem ser explicitamente configurados para que reconheçam que se trata de um endereço anycast.

Para cada endereço anycast designado, existe um prefixo P mais extenso que identifica a região topológica em que todas as interfaces pertencentes àquele endereço residem. Com essa região identificada pelo P, cada membro do grupo anycast deve ser advertido com uma entrada separada no sistema de roteamento (normalmente referido a um "host route"); fora da região identificada pelo P, o endereço anycast pode ser agregado ao “routing advertisement” pelo prefixo P.

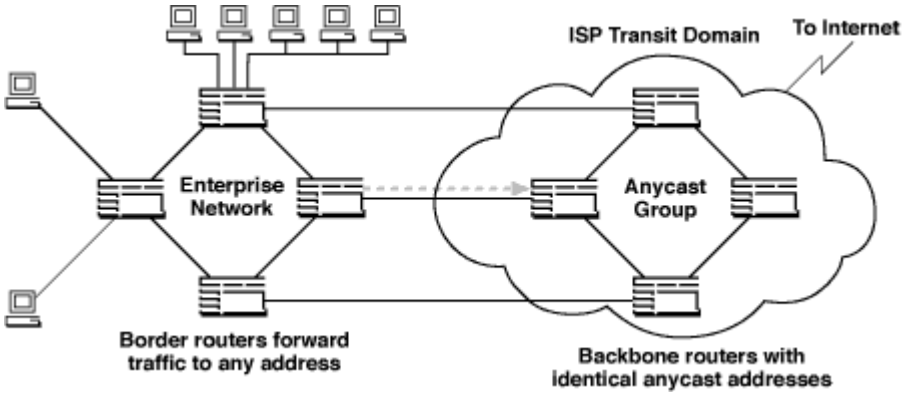


Figura 1-15 Modelo Anycast

Endereço Anycast requerido

O endereço anycast de roteamento de subredes é predefinido. Seu formato é conforme ilustra a figura seguinte:

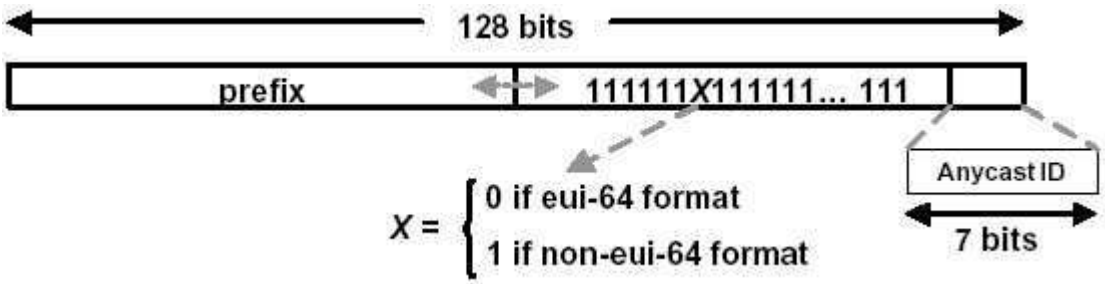


Figura 1-16 Modelo de endereçamento anycast

O prefixo de subrede no endereçamento anycast é aquele que identifica um enlace especial. Esse endereço anycast é sintaticamente idêntico ao endereço unicast para uma interface em um enlace com identificador de interface setado em zero.

Pacotes enviados a esse endereço serão entregues à um roteador na subrede. Todos os roteadores são obrigados a suportar esse estilo de endereçamento para as subredes em que têm interface.

O endereço anycast de roteamento de subredes é destinado à utilização em aplicações onde um nó necessite comunicar com um, dentre um grupo roteadores em uma subrede remota.

1.8.8. Endereços Multicast

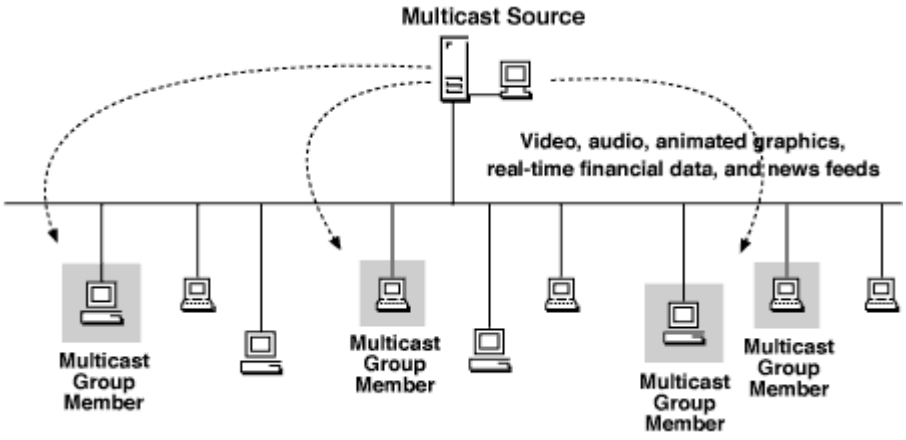


Figura 1-17 Modelo Multicast

Um endereço multicast é um identificador para um grupo de nós. Um nó pode pertencer à qualquer número de grupos de nós. Os endereços multicast possuem o seguinte formato:

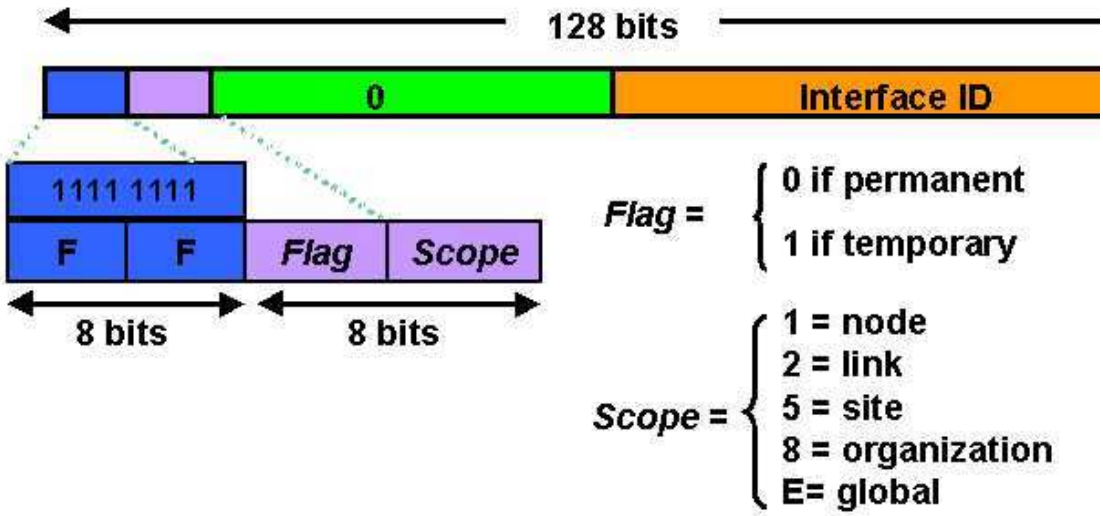


Figura 1-18 Estrutura de endereço multicast

11111111 [no início] - identifica o endereço como multicast.

flgs é um conjunto de 4 flags:

```
+-+-+-+-+
|0|0|0|T|
+-+-+-+-+
```

O três flags de maior ordem são reservados, e devem ser inicializados em 0.

T = 0 indica um endereço multicast permanentemente reservado, reserva feita pela "Global Internet numbering authority".

T = 1 indica um endereço multicast não reservado permanentemente ("transitório")

Scope é um valor multicast de 4-bits usados para limitar o escopo de um grupo multicast. Seus valores são:

0 reservado
1 escopo nó-local
2 escopo link-local
3 (não determinado)
4 (não determinado)
5 escopo site-local
6 (não determinado)
7 (não determinado)
8 escopo organization-local
9 (não determinado)
A (não determinado)
B (não determinado)
C (não determinado)
D (não determinado)
E escopo global
F reservado

O “group ID” identifica o grupo multicast, seja permanente ou transitório, com os escopos dados.

O significado de um endereço multicast permanente é independente do valor do escopo.

Por exemplo, se designarmos a um grupo de servidores de tempo (NTS) um endereço multicast permanente com um ID de grupo igual a 101 (hex), então:

FF01:0:0:0:0:0:0:101 significa todos servidores NTS do mesmo nó que a origem.

FF02:0:0:0:0:0:0:101 significa todos servidores NTS do mesmo enlace que a origem.

FF05:0:0:0:0:0:0:101 significa todos servidores NTS do mesmo site que a origem.

FF0E:0:0:0:0:0:0:101 significa todos servidores NTS na internet.

Os endereços multicast não permanentes só têm sentido em seu próprio âmbito (escopo).

Por exemplo, um grupo identificado com o endereço multicast site-local não permanente (também chamado temporal) FF15:0:0:0:0:0:0:101, não tem nenhuma relação com outro grupo usando o mesmo endereço em outro site, nem com outro grupo temporal usando o mesmo ID de grupo(em outro escopo), nem com outro grupo permanente com o mesmo ID de grupo.

Endereços Multicast não devem ser usados como endereços de origem em pacotes IPv6 ou em qualquer cabeçalho de roteamento.

Endereços Multicast pré-definidos

Alguns exemplos de endereços multicast pré-definidos se seguem:

FF00:0:0:0:0:0:0:0
FF01:0:0:0:0:0:0:0
FF02:0:0:0:0:0:0:0
FF03:0:0:0:0:0:0:0
FF04:0:0:0:0:0:0:0
FF05:0:0:0:0:0:0:0
FF06:0:0:0:0:0:0:0
FF07:0:0:0:0:0:0:0
FF08:0:0:0:0:0:0:0
FF09:0:0:0:0:0:0:0
FF0A:0:0:0:0:0:0:0
FF0B:0:0:0:0:0:0:0
FF0C:0:0:0:0:0:0:0
FF0D:0:0:0:0:0:0:0
FF0E:0:0:0:0:0:0:0
FF0F:0:0:0:0:0:0:0 Os endereços acima são reservados não devendo ser utilizados por qualquer grupo multicast.

Alguns exemplos úteis de endereços multicast, seguindo seus escopos, seriam:

FF01:0:0:0:0:0:0:1 - todos os nós (escopo local)

FF02:0:0:0:0:0:0:1 - todos os nós (escopo de enlace)
FF01:0:0:0:0:0:0:2 - todos os roteadores (escopo local)
FF02:0:0:0:0:0:0:2 - todos os roteadores (escopo de enlace)
FF05:0:0:0:0:0:0:2 - todos os roteadores (escopo de site)

O endereço FF02:0:0:0:0:1:FFXX:XXXX, denominado “Solicited-node address”, ou endereço de nó solicitado, permite calcular o endereço multicast a partir de um *unicast* ou anycast de um determinado nó. Para isso, se substituem os 24 bits de menor peso “x” pelos mesmos bits do endereço original.

Assim, o endereço 4037::01:800:200E:8C6C se converteria em FF02::1:FF0E:8C6C.

Endereços de Nó obrigatórios

Cada nó IPv6 deve reconhecer os seguintes endereços como identificação deles próprios:

- Os endereços *Link-local* de cada interface;
- Endereços determinados (manual ou automaticamente) Unicast/Anycast;
- O endereço *loopback*;
- Todos os endereços de nós multicasting;
- O endereço de nó solicitado multicast para cada endereço *unicast* e *anycast* determinado;
- Os endereços *multicast* dos grupos, os quais, o nó faz parte.

Endereços de Routers obrigatórios

No caso dos roteadores, estes devem reconhecer os seguintes endereços:

- Todos os endereços de nós determinados;
- Todos os endereços de roteadores multicast;
- Endereços multicast específicos para os protocolos de roteamento;
- Endereços Anycast de roteadores de subredes para as interfaces configuradas em agir como “forwarding interfaces”;
- Outros endereços anycast configurados.

1.9. COMPATIBILIDADE ENTRE REDES IPV6 E IPV4

Os mecanismos de transição IPv6 [TRAN] incluem uma técnica para *hosts* e roteadores para tunelamento dinâmico de pacotes IPv6 sobre IPv4.

Nós IPv6 que utilizam essa técnica são nomeados com endereços *unicast* IPv6 especiais que carregam um endereço IPv4 nos 32-bits de ordem mais baixa.

Esse tipo de endereço foi batizado como "***IPv4-compatível com endereço IPv6***" e possui o seguinte formato:

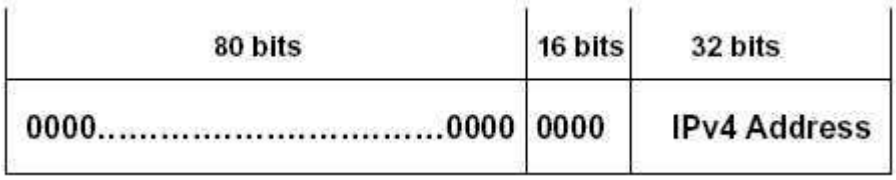


Figura 1-19 Modelo de endereço IPv4 compatível com endereço IPv6

O segundo tipo de endereço IPv6 que suporta um encapsulamento de endereço IPv4 também está definido. Esse endereço é usado para representar os endereços dos nós de “IPv4-only” (aqueles que não suportam IPv6) como endereços IPv6.

Esse tipo de endereço é denominado "***IPv4-mapeado endereço IPv6***" e contém o formato abaixo descrito:

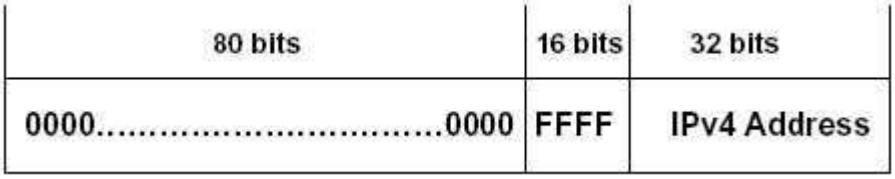


Figura 1-20 Modelo de endereço IPv4 mapeado endereço IPv6

Codificação e transição do endereço IPv4

De acordo com a tabela 5 apersentada anteriormente, mais de 72% do espaço de endereço foi reservado para uso futuro, sem incluir a seção reservada para endereços geográficos. Embora o prefixo *0000 0000* tenha na tabela anterior o rótulo *Reservado*, os projetistas planejaram usar nessa seção uma pequena fração de endereços para codificar endereços IPv4. Em particular, qualquer endereço que comece com 80 bits zero (0) seguidos de 16 um (1) ou 16 bits zero (0) contém um endereço de IPv4 nos 32 bits de ordem baixa. A codificação será necessária durante uma transição do IPv4 para IPv6, por duas razões. Primeira, um computador pode escolher uma atualização de software do IPv4 para IPv6 antes que lhe tenha sido atribuído um endereço válido do IPv6. Segunda, um computador que execute o software do IPv6 pode ter necessidade de se comunicar com outro que execute apenas o software do IPv4.

Dispor de uma forma de codificar um endereço de IPv4 em um endereço de IPv6 não soluciona o problema de tornar as duas versões interoperacionais. Além da codificação de endereços, a conversão é necessária. Para usar um *gateway*, um computador IPv6 gera um datagrama que contenha a codificação do IPv6 do endereço de destino do IPv4. O computador do IPv6 envia o datagrama para um *gateway* que usa IPv4 para se comunicar com o destino. Quando o *gateway* recebe uma resposta do destino, converte o datagrama do IPv4 para IPv6 e o devolve à origem IPv6.

1.10. ICMPv6

Como no IPv4, o IPv6 não provê facilidades para reportar erros. Entretanto, o IPv6 usa uma versão atualizada do Internet Control Message Protocol (ICMP) denominado ICMP versão 6 (ICMPv6). O ICMPv6 tem em comum funções IPv4 ICMP de “reporting delivery” ou “forwarding errors” e provê um simples *echo service* para *troubleshooting*.

O protocolo ICMPv6 disponibiliza também um “framework” para:

- Multicast Listener Discovery (MLD)

O MLD se caracteriza por uma série de três mensagens ICMP que substituem a versão 2 do Internet Group Management Protocol (IGMP) para IPv4 e gerenciam membros multicast de subredes. A figura 1-21 ilustra um pacote deste tipo.

- Neighbor Discovery (ND)

Já o Neighbor Discovery é uma série de cinco mensagens ICMPv6 que gerenciam a comunicação *node-to-node* em um determinado enlace. Nesse caso, o Neighbor Discovery substitue o Address Resolution Protocol (ARP), ICMPv4 Router Discovery e o ICMPv4 Redirect message.

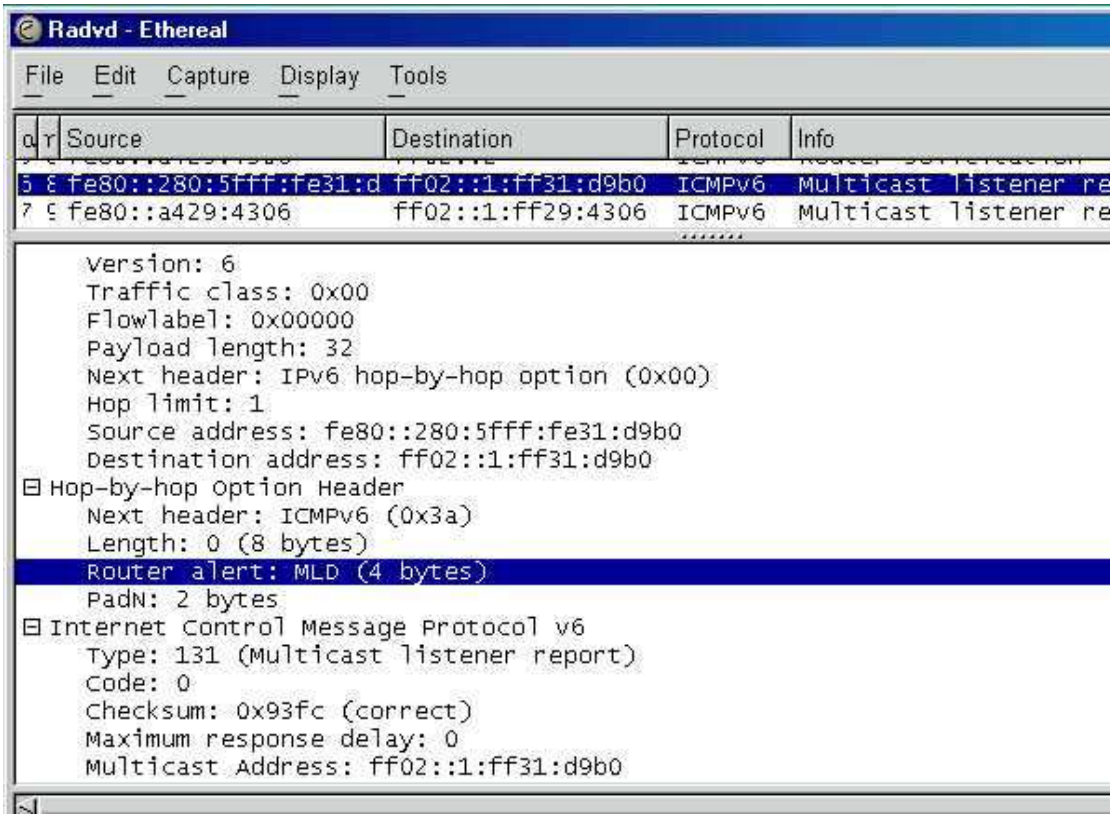


Figura 1-21 Pacote MLD

1.10.1. Tipos de mensagens ICMPv6

Existem dois tipos de mensagens ICMPv6:

1. Mensagens de Erro

Mensagens de erro são usadas para reportar erros na entrega e encaminhamento de pacotes IPv6 tanto para o nó destino, quanto para um roteador intermediário. O valor do campo type (8 bits) nas mensagens de erro ICMPv6 vai de 0 à 127. Dentre as opções existentes incluem: Destination Unreachable, Packet Too Big, Time Exceeded, e Parameter Problem.

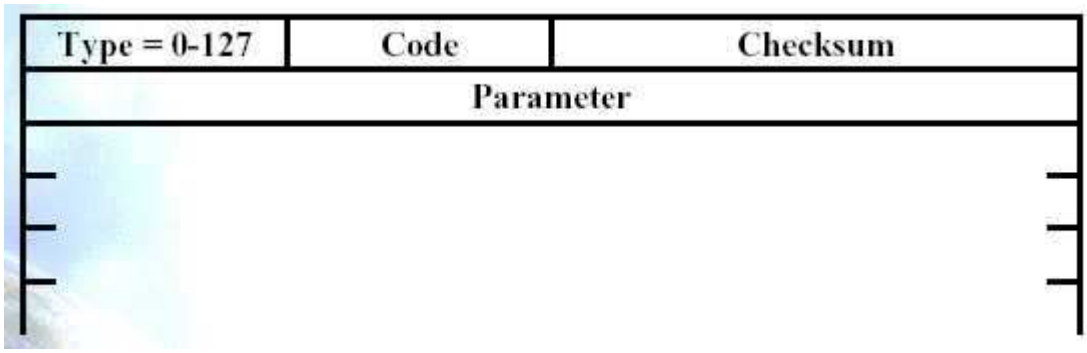


Figura 1-22 Estrutura de mensagem de erro ICMPv6

2. Mensagens Informacionais

As mensagens informacionais são usadas para promover funções de diagnóstico e funcionalidades adicionais de hosts assim como o MLD e Neighbor Discovery. O valor do campo type vai de 128 a 255. As mensagens informacionais ICMPv6 estão descritas na RFC 2463 e incluem Echo Request e Echo Reply.

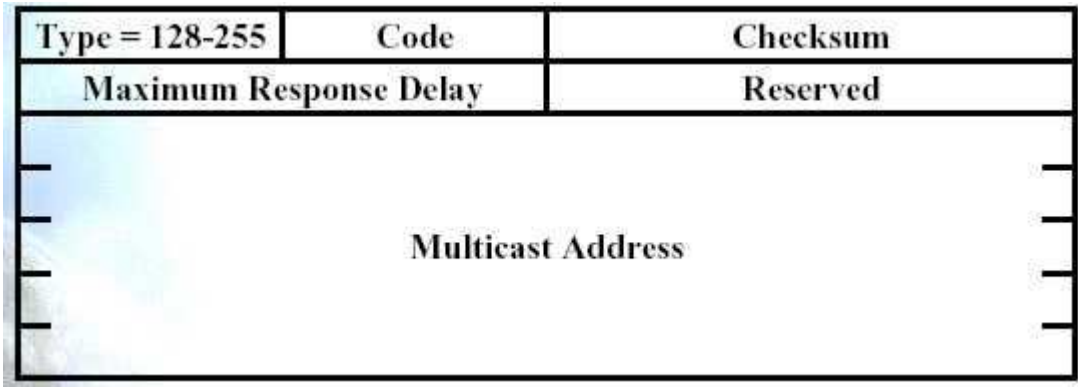


Figura 1-23 Estrutura de mensagem informacional ICMPv6

1.10.2. Cabeçalho ICMPv6

A figura 1-24 mostra a estrutura de todas as mensagens ICMPv6.

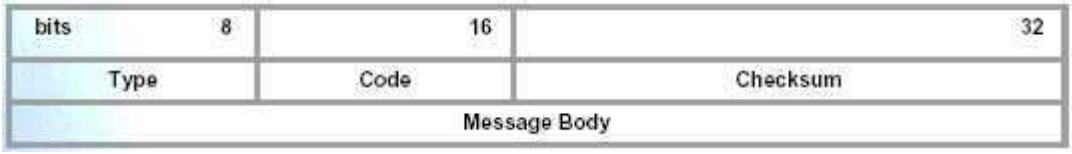


Figura 1-24 Estrutura mensagens ICMPv6

Os campos descritos acima são:

Type – Indica o tipo de mensagem ICMPv6. Seu tamanho é de 8 bits. Nos casos de mensagens de erro ICMPv6, o bit de maior ordem é setado em 0. Já no caso de mensagens informacionais, esse bit está setado em 1.

Code – Diferencia mensagens múltiplas com o mesmo campo tipo. Seu tamanho de campo é também igual a 8 bits. Se existir apenas uma mensagem para um campo tipo

qualquer, seu código será igual a 0.

Checksum – Armazena o checksum da mensagem ICMP. Seu tamanho é de 16 bits. O pseudo-cabeçalho IPv6 é adicionado à mensagem quando calculado o checksum.

Message body – Contém os dados específicos da mensagem ICMPv6.

1.10.3. Mensagens de erro ICMPv6

Essas mensagens são usadas para reportar erros de encaminhamento ou de entrega tanto pelos roteadores, quanto pelos hosts de destino.

Destination Unreachable

Uma mensagem classificada nesse tipo é enviada tanto pelos roteadores, quanto pelos hosts destino quando o pacote não pode ser encaminhado ao seu destino. A figura 1-25 ilustra justamente esse caso:

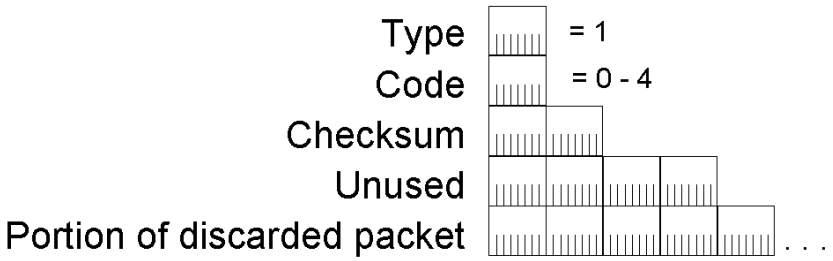


Figura 1-25 Mensagem Destination Unreachable

Nas mensagens Destination Unreachable, o campo Type é setado em 1 e o campo Code, em valores de 0 à 4. Após o campo de Checksum (32-bit), Unused field e da porção de pacote descartado, contendo a mensagem ICMPv6, o pacote não fica maior que 1280 bytes (o mínimo de MTU IPv6). O número de bytes descartados incluídos na mensagem varia de acordo com a quantidade de *extension headers* IPv6 presentes.

Para uma mensagem ICMPv6 sem *extension headers*, 1232 bytes do pacote descartado são incluídos (1280 menos um cabeçalho de 40-bytes e um cabeçalho de 8-bytes de ICMPv6 Destination Unreachable).

A tabela seguinte mostra os valores do campo field para as várias mensagens de Destination Unreachable.

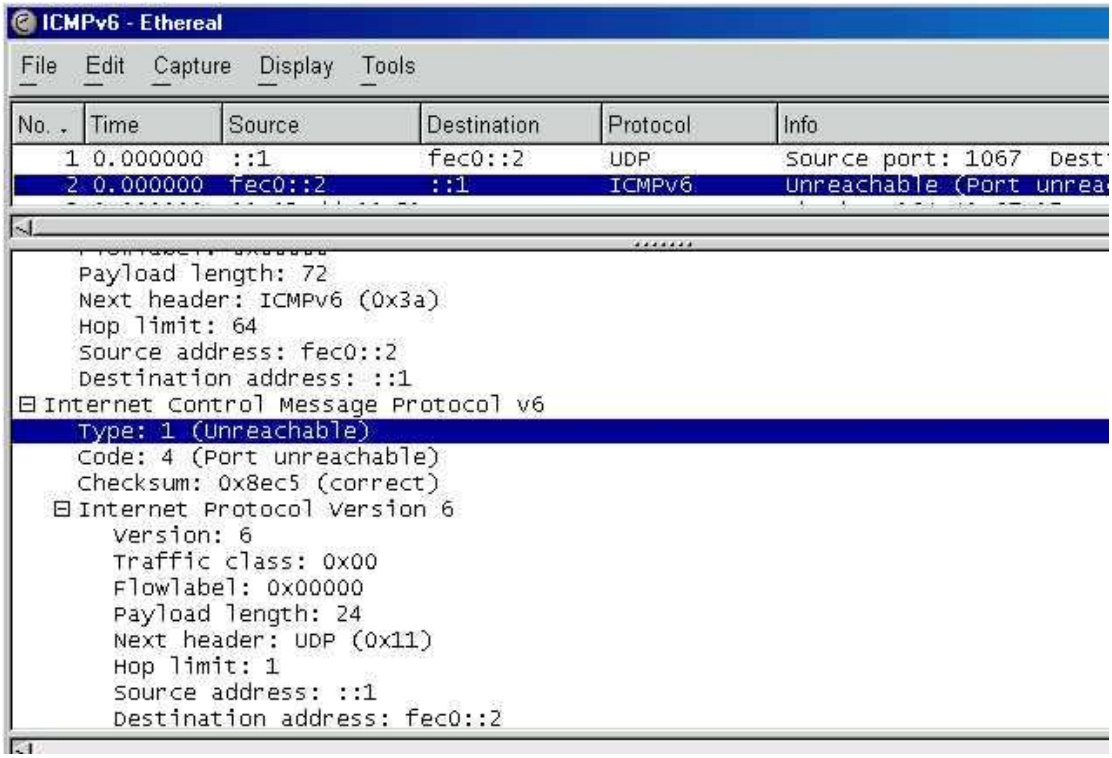
Tabela 6 Campo Field de mensagens ICMPv6 Destination Unreachable

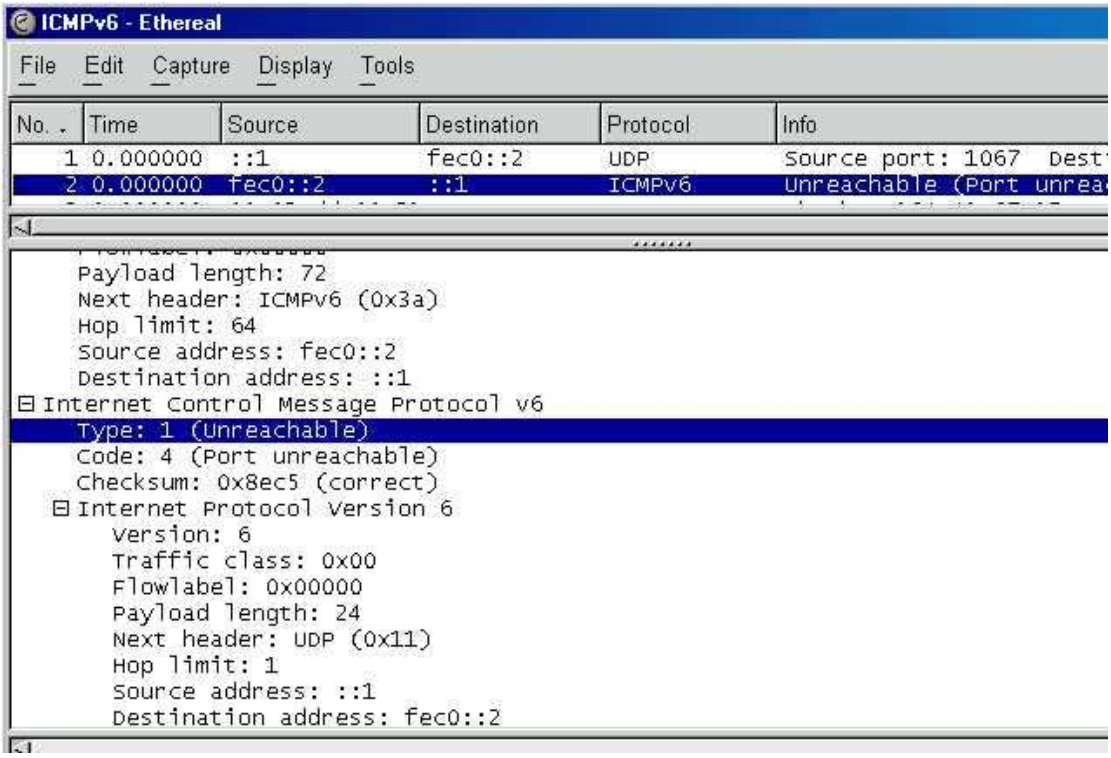
Código	Descrição
0	Caminho de destino não foi encontrado na tabela de roteamento.
1	A comunicação com o destino foi proibida por políticas administrativas. É tipicamente enviada quando um pacote é descartado por um firewall.
2	O endereço está aquém do escopo da origem.

3	O endereço de destino está inalcançável. Típica ocorrência quando há uma inabilidade de resolver endereço da camada de enlace do destino.
4	A porta de destino não foi alcançada. Tipicamente enviada quando um pacote contendo mensagem UDP chega no destino e lá não encontra nenhuma aplicação escutando a porta UDP.

O pacote ICMPv6 capturado abaixo mostra a mensagem de erro de porta inalcansável (código 4).

Figura 1-26 Pacote ICMPv6 Port Unreachable





Packet Too Big

Uma mensagem ICMPv6 “Packet Too Big” é enviada quando o pacote não pode ser encaminhado devido ao tamanho inferior do MTU do link. A figura 1-27 ilustra o pacote do caso descrito.

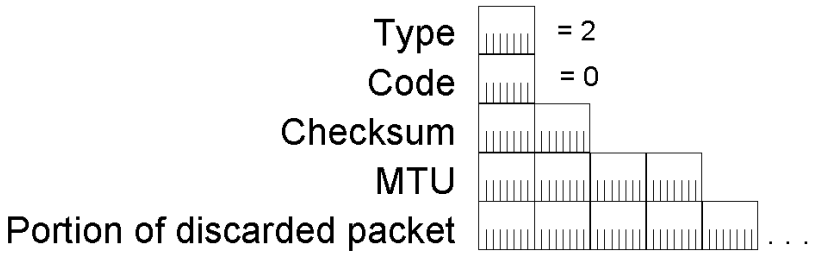


Figura 1-27 Mensagem ICMPv6 “Packet Too Big”

Nas mensagens “Packet Too Big”, o campo *Type* é setado em 2 e o *Code* em 0.

Time Exceeded

Uma mensagem ICMPv6 *Time Exceeded* é tipicamente enviada por um router quando o campo *Hop Limit* do cabeçalho IPv6 é zero, seja depois de receber ou decrementar o valor durante o processo de encaminhamento.

A figura 1-28 mostra a mensagem ICMPv6 *Time Exceeded*.

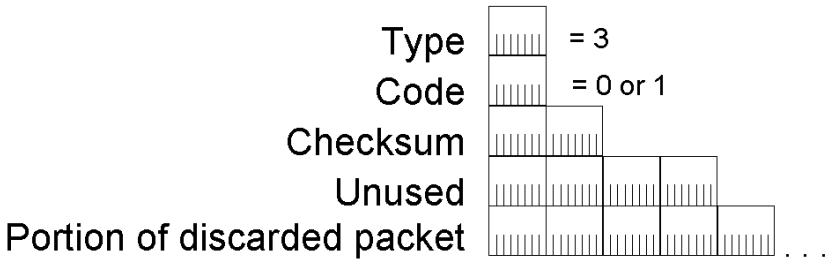


Figura 1-28 Mensagem ICMPv6 Time Exceeded

Nas mensagens *Time Exceeded*, o campo *Type* é setado em 3 e o campo *Code* em 0 (quando o campo *Hop Limit*, no cabeçalho IPv6 se torna 0) ou 1 (quando o tempo da remontagem da fragmentação do host destino excede). O recebimento de mensagens de code = 0 indica que, tanto *Hop Limit* dos pacotes de saída não são grandes o bastante para chegar até o destino, ou existe um loop no roteador.

Parameter Problem

As mensagens ICMPv6 *Parameter Problem* são tanto enviadas por um roteador, quanto pelo host de destino. Isso ocorre quando um erro é detectado seja no cabeçalho IPv6, seja no de extensão, prevenindo o IPv6 de futuros processos.
A figura seguinte relata esse caso - ICMPv6 *Parameter Problem*.

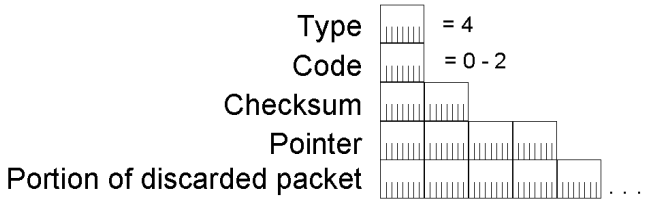


Figura 1-29 Mensagem ICMPv6 Parameter Problem

Nas mensagens “Parameter Problem”, o campo *Type* está setado em 4 e o valor do campo *Code* flutua de 0 a 2. A tabela 7 mostra os valores do campo Code para as mensagens - Parameter Problem.

Tabela 7 Campo Code de mensagens ICMPv6 Parameter Problem

Código	Descrição
0	Erro no campo do cabeçalho IPv6 ou no cabeçalho de extensão encontrado.
1	Um valor desconhecido de campo *Next Header* foi encontrado. É equivalente às mensagens “IPv4 Destination Unreachable-Protocol Unreachable”.
2	Uma opção IPv6 desconhecida foi encontrada.

1.10.4. Mensagens Informacionais ICMPv6

Esse tipo de mensagem, definida na RFC 2463, provê capacidades de diagnóstico.

Echo Request

Essa mensagem é enviada para o destino solicitando uma resposta imediata - *Echo Reply message*, gerando um simples diagnóstico na solução de problemas de conectividade e roteamento. A figura 1-30 ilustra a mensagem “ICMPv6 Echo Request”.

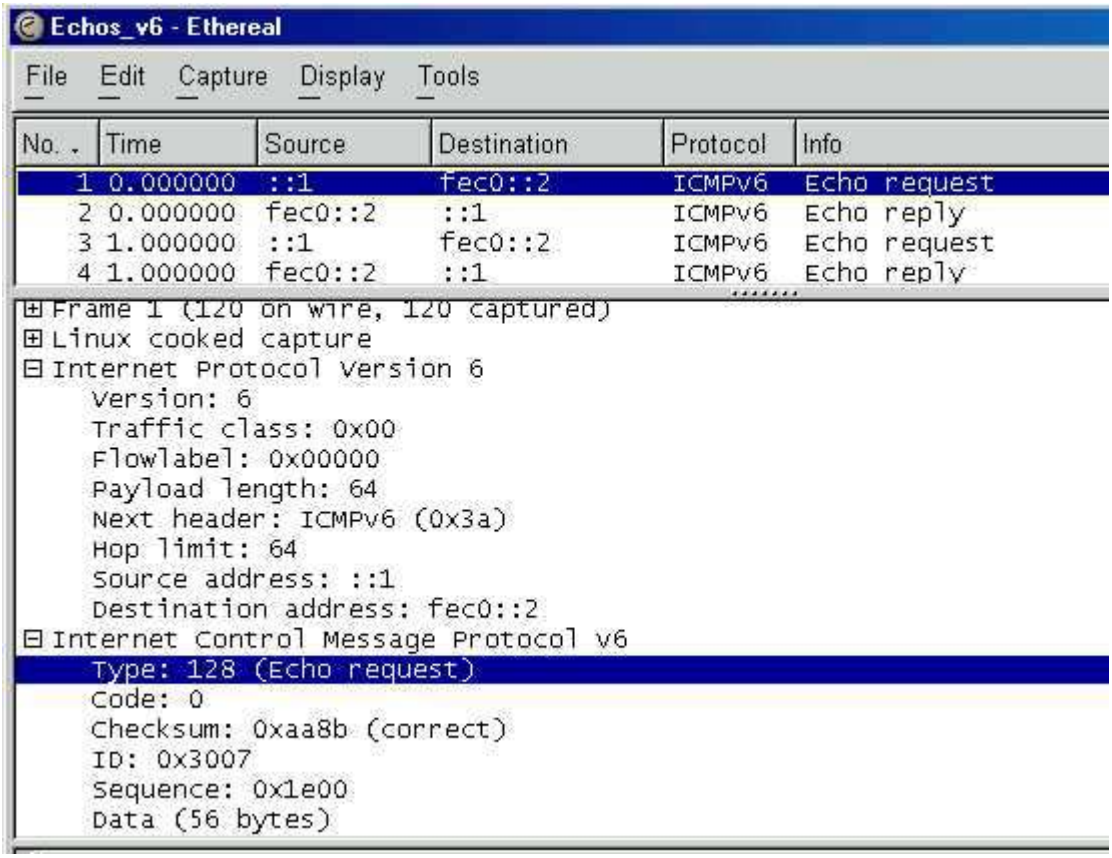


Figura 1-30 Mensagem ICMPv6 Echo Request

Na mensagem *Echo Request*, o campo *Type* é setado em 128 e o campo *Code* em 0. Seguindo o *Checksum* existem os campos de 16-bits: *Identifier* e *Sequence Number*.
Os campos *Identifier* e *Sequence Number* são setados pelo host de envio e usados para relacionar uma mensagem de chegada *Echo Reply* com sua correspondente *Echo Request*. O campo *Data* contém zero ou mais bytes de dados opcionais que também são enviados pelo host de origem.

Echo Reply

A mensagem “ICMPv6 Echo Reply” é enviada como uma resposta do ICMPv6 *Echo Request*. A figura 1-31 mostra a mensagem “ICMPv6 Echo Reply”.

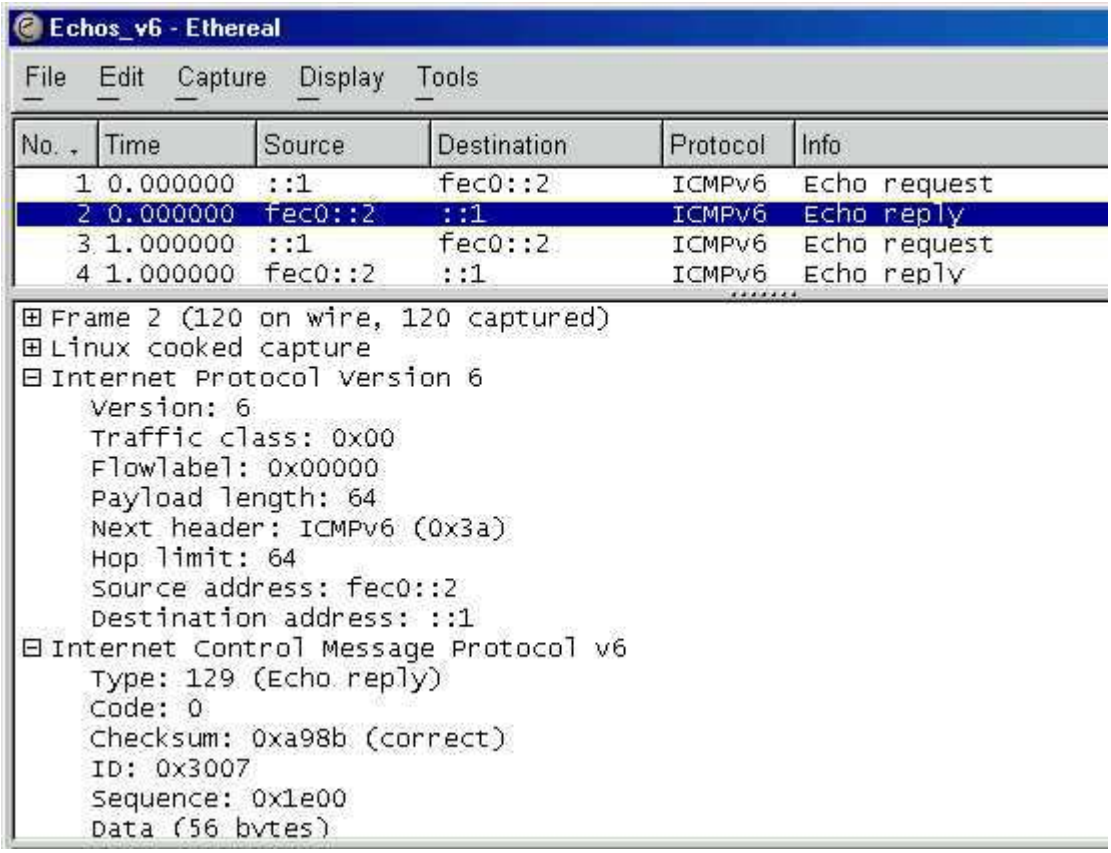


Figura 1-31 Mensagem ICMPv6 Echo Reply

Na mensagem *Echo Request*, o campo *Type* é setado em 129 e o campo *Code* em 0. Seguindo o *Checksum* existem os campos de 16-bits: *Identifier* e *Sequence Number*. Os campos *Identifier* e *Sequence Number* e dados são setados com os mesmos valores que no *Echo Request*.

1.10.5. Comparativo entre mensagens ICMPv4 x ICMPv6

A tabela seguinte lista e faz uma comparação entre o ICMPv4 e as mensagens ICMPv6.

Tabela 8 - Comparativo mensagens ICMPv4 x ICMPv6

Mensagem ICMPv4	Equivalente ICMPv6
Destination Unreachable-Network unreachable (Type 3, Code 1)	Destination Unreachable-No route to destination (Type 1, Code 0)
Destination Unreachable-Host unreachable (Type 3, Code 1)	Destination Unreachable-Address unreachable (Type 1, Code 3)
Destination Unreachable-Protocol unreachable (Type 3, Code 2)	Parameter Problem-Unrecognized Next Header field (Type 4, Code 1)
Destination Unreachable-Port unreachable (Type 3, Code 3)	Destination Unreachable-Port unreachable (Type 1, Code 4)
Destination Unreachable-Fragmentation necessita do DF setado (Type 3, Code 4)	Packet Too Big (Type 2, Code 0)
Destination Unreachable-Communication com host de destino administrativamente proibido (Type 3, Code 10)	Destination Unreachable-Communication com destino administrativamente proibido (Type 1, Code 1)

Time Exceeded-TTL expirado (Type 11, Code 0)	Time Exceeded-Hop Limit excedente (Type 3, Code 0)
Time Exceeded-Fragmentation timer expirado (Type 11, Code 1)	Time Exceeded-Fragmentation timer excedente (Type 3, Code 1)
Parameter Problem (Type 12, Code 0)	Parameter Problem (Type 4, Code 0 ou Code 2)
Source Quench (Type 4, Code 0)	Não é implementado no IPv6
Redirect (Type 5, Code 0)	Neighbor Discovery Redirect message (Type 137, Code 0).

1.11. NEIGHBOR DISCOVERY (ND)

O Neighbor Discovery (ND) do IPv6 é uma série de mensagens e processos que determinam relacionamentos entre nós vizinhos. O ND substitui o ARP, ICMP Router Discovery, e ICMP Redirect usados no IPv4 e, além disso, provê funcionalidades adicionais.
O ND é usado por:

- Hosts para descobrir roteadores vizinhos.
- Hosts para descobrir endereços, prefixos de endereços, e outros parâmetros de configuração.
- Nós tanto para resolver endereços da camada de enlace de nós vizinhos os quais recebem os pacotes encaminhados, quanto para determinar quando o endereço da camada de enlace modificou.
- Nós para determinar se o vizinho ainda está disponível.
- Roteadores para anunciar sua presença, parâmetros de configuração de host, e prefixos de enlace.
- Roteadores para informar aos hosts da presença de um endereço *next-hop* melhor quando se encaminhar pacotes a um destino específico.

A tabela 9 lista e descreve os processos do ND documentados na RFC 2461.

Tabela 9 - Processos IPv6 Neighbor Discovery

Processos	Descrição
Router discovery	Processo pelo qual um host descobre roteadores locais situados no mesmo link. Equivalente ao ICMPv4 Router Discovery.
Prefix discovery	Processo pelo qual um host descobre os prefixos de rede para um destino local link. Similar ao ICMPv4 Address Mask Request/Reply.
Parameter discovery	Processo pelo qual um host descobre parâmetros adicionais de operação, incluíndo o MTU de enlace e o default hop limit para pacotes de saída.
Address autoconfiguration	Processo para configuração de endereços IP para interfaces seja na presença ou ausência de um servidor de configuração de endereços stateful tal como o Dynamic Host Configuration Protocol version 6 (DHCPv6).
Address resolution	Processo equivalente ao ARP no IPv4.

Next-hop determination	Processo onde, baseado no endereço de destino, um nó determina o endereço do vizinho a ser enviado o pacote. O next-hop address ou forwarding address pode ser tanto o endereço de destino quanto um endereço de roteador default no mesmo link.
Neighbor unreachability detection	Processo pelo qual o nó detecta que a camada IPv6 do vizinho não está recebendo pacotes.
Duplicate address detection	Processo pelo qual o nó determina que um endereço considerado em uso não está sendo utilizado pelo nó vizinho. Equivalente ao "using gratuitous ARP frames" no IPv4.
Redirect function	Processo que informa a um host a existência de um melhor "first-hop" para alcançar o endereço IPv6 de destino. Equivalente ao IPv4 ICMP Redirect message.

1.11.1. Formato da mensagem Neighbor Discovery

Assim como as mensagens Multicast Listener Discovery (MLD), as mensagens ND usam a estrutura de mensagem do ICMPv6 e também os ICMPv6 types de 133 até 137. As mensagens ND consistem de um cabeçalho ND, composto de um cabeçalho ICMPv6, dos dados específicos da mensagem ND, e zero ou mais opções ND. A figura 1-32 descreve o que foi explicado.

Figura 1-32 Formato da mensagem Neighbor Discovery

Existem cinco mensagens ND diferentes:

- Router Solicitation
- Router Advertisement
- Neighbor Solicitation
- Neighbor Advertisement
- Redirect

As opções da mensagem ND provêm informações adicionais, tipicamente indicando MAC addresses, prefixos de rede do enlace, informação MTU do enlace e dados de redirecionamento.

Para assegurar o recebimento das mensagens originadas no enlace, todas elas são enviadas com o hop limit igual a 255. Quando uma mensagem é recebida, o campo Hop Limit no cabeçalho IPv6 é checado. Se o seu valor foi diferente de 255, a mensagem é silenciosamente descartada. A verificação desse campo dá a proteção de ataques enviados por nós de fora do enlace. Como o Hop limit é 255, um roteador não pode encaminhar os pacotes para fora do enlace.

1.11.2. As opções Neighbor Discovery

As opções ND estão ilustradas a seguir:

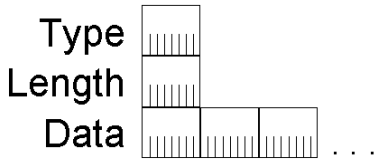


Figura 1-33 Formato do campo opções de Neighbor Discovery

O campo type (8-bits) indica o tipo de opção ND. A tabela 10 descreve os tipos definidos pela RFC 2461.

Tabela 10 – Opções campo Type de Neighbor Discovery

Tipo	Nome da opção
1	Endereço camada de enlace de origem
2	Endereço camada de enlace destino
3	Informação de prefixo
4	Cabeçalho redirected
5	MTU

O campo Length de 8 bits indica o comprimento de todo o campo opção em blocos de 8-bytes. Todas as opções ND devem situar-se entre os 8-bytes limites. O campo *data*, de comprimento variável contém os dados para a opção.

1.11.3. Opção de endereços camada de enlace Origem/Destino

A opção de endereço camada de enlace origem indica o endereço da entidade que enviou a mensagem. Essa opção é incluída na Neighbor Solicitation, Router Solicitation, e mensagens de Router Advertisement. A opção não é incluída quando o endereço da origem da mensagem ND é o endereço não-especificado - unspecified address (::).

A opção de endereço de camada de enlace de destino indica o endereço do nó vizinho que o pacote IPv6 deve ser direcionado. Essa opção é incluída no Neighbor Advertisement e Redirect messages.

Ambas as opções possuem formatos idênticos conforme ilustra a figura 1-34.

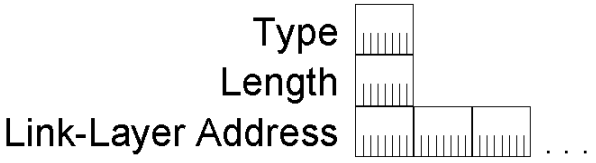


Figura 1-34 Formato opções de endereço da camada de enlace origem/destino

O campo type é setado em 1 no caso “origem” e 2 no “destino”. Já o campo Length é setado de acordo com o número de blocos de 8 bytes em todo “option”. O

terceiro campo é de comprimento variável e contém o endereço camada de enlace da origem/destino.

1.11.4. Opção informação de prefixo

A opção, informação de prefixo, é enviada nas mensagens de Router Advertisement para indicar tanto os prefixos quanto as informações a respeito dos endereços de autoconfiguração. Podem existir múltiplas opções de informação de prefixo nas mensagens Router Advertisement, indicando múltiplos prefixos de endereçamento.

A figura 1-35 ilustra o formato da opção.

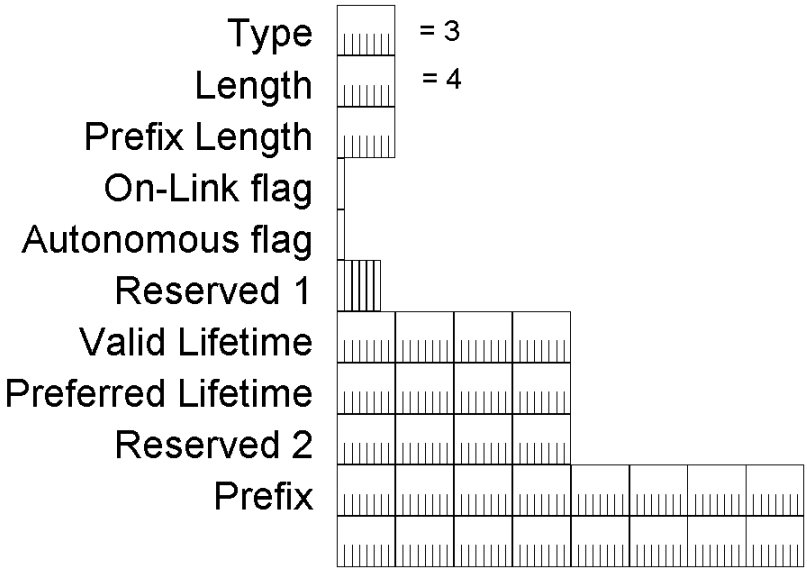


Figura 1-35 Formato mensagem opção de informação de prefixo

Algumas informações sobre essa figura:
Type – seu valor é igual a 3.
Length – o valor de seu campo é 4 (o pacote "opção" todo possui 32 bytes de tamanho).
Prefix Length – o tamanho de seu campo é de 8 bits. Seu valor, portanto, vai de 0 à 128. Indica o número dos bits principais no campo prefixo que abrangem o endereço de prefixo.
On-link flag – Indica, quando setado em 1, que o endereço relatado pelo prefixo incluso está disponível no mesmo enlace onde a mensagem Router Advertisement foi recebida. Quando seu valor é 0, essa premissa não é assumida. O tamanho desse campo é de 1 bit.
Autonomous flag – Indica, ao ser setado em 1, que o prefixo incluso é usado para gerar uma configuração de endereçamento tipo *stateless*. A afirmação é contrária quando o valor é 0. Esse campo também tem o tamanho igual a 1 bit.
Reserved 1 – Esse campo possui 6-bits e está reservado para uso futuro. Está setado em 0.
Valid Lifetime – Indica o número de segundos que o endereço, baseado no prefixo incluso e utilizando a configuração stateless, permanece válido. Com o tamanho de 32 bits, o campo também indica o número de segundos que o prefixo incluso é válido para uma determinação no próprio enlace. Para um valor infinito, o campo é setado

em 0xFFFFFFFF.
Preferred Lifetime – Indica o tempo, em segundos, que um endereço baseado no seu prefixo e usando configuração *stateless* permanece no estado *preferred*. O tamanho desse campo é igual a 32 bits. Endereços configurados automaticamente que continuam válidos podem estar em dois estados, *preferred* ou *deprecated*. No estado *preferred*, ele pode ser usado sem restrições em qualquer comunicação. Já no estado *deprecated*, o endereço não pode ser usado em novas comunicações. No entanto, conexões já estabelecidas podem continuar usando endereços no estado deprecated. Um endereço vai do estado preferred ao deprecated quando este timer expira. Para um tempo infinito no estado preferred esse campo é setado em 0xFFFFFFFF. **Reserved 2** – Um campo de 32-bits reservado para uso futuro. O campo é setado em 0.
Prefix – Indica o prefixo para o endereço IPv6 derivado da autoconfiguração stateless. O tamanho do campo é de 128.

1.11.5. Opção de Cabeçalho Redirected

Essa opção é enviada em mensagens tipo *redirect* para especificar o pacote IPv6 que leva o roteador a enviar uma mensagem *redirect*. Ela pode conter todo, ou parte o pacote *redirect* IPv6, dependendo do tamanho do pacote que foi enviado inicialmente.
O formato dessa opção está descrito abaixo:

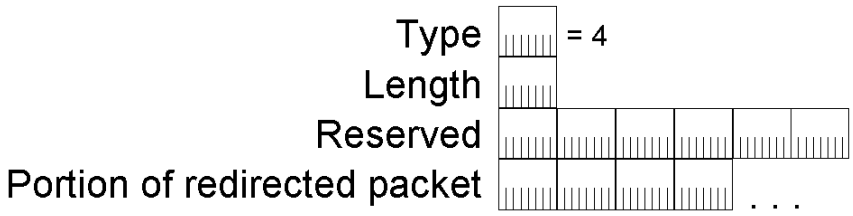


Figura 1-36 Formato da opção Redirected Header

O detalhamento dos campos é feito a seguir:
Type – O valor é igual a 4.
Length – O valor desse campo é o número de blocos de 8 bytes em todo option.
Reserved – Um campo de 48-bits reservados para o futuro e setados em 0.
Portion of redirected packet – Contém ou o pacote IPv6, ou uma porção do pacote IPv6 que causou o envio da mensagem redirect.

1.11.6. Opção MTU

A opção MTU é enviada nas mensagens Router Advertisement para indicar o MTU do enlace. Essa opção é tipicamente usada apenas quando o MTU IPv6 para um enlace não é bem conhecido

Em ambientes “bridged” ou “switched” camada 2, é possível ter tecnologias de camadas de enlace diferentes com diferentes MTUs no mesmo segmento de rede. Nesse caso, diferenças em tamanhos de MTUs IPv6 situados na mesma rede não são descobertas por meio do Path MTU Discovery. A opção MTU é usada para indicar o maior MTU IPv6 suportado nas tecnologias do segmento enlace.

Considere a configuração “switched” mostrada na figura 1-37.

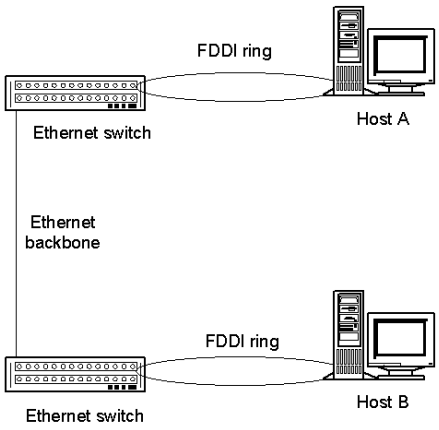


Figura 1-37 Exemplo opção MTU

Os hosts IPv6, A e B, são conectados a dois *switches* Ethernet (camada 2) diferentes usando portas Fiber Distributed Data Interface (FDDI). Os dois switches são conectados por um backbone Ethernet. Quando os Hosts A e B negociam uma conexão TCP, cada um reporta o tamanho o segmento TCP máximo igual a 4312 (MTU FDDI [4352] – [40 bytes] de cabeçalho IPv6). Quando os dados TCP começam, os switches silenciosamente descartam os pacotes IPv6 maiores que 1500 bytes entre o enlace.

Com a opção MTU, o roteador do segmento de rede (não mostrado) reporta à todos os hosts, por meio da mensagem Router Advertisement, um MTU IPv6 de 1500 bytes. Assim que os hosts ajustam seus MTUs de 4312 para 1500, os dados da conexão TCP de tamanho máximo deixam de ser descartados pelos switches intermediários.
Esse formato é relatado a seguir:

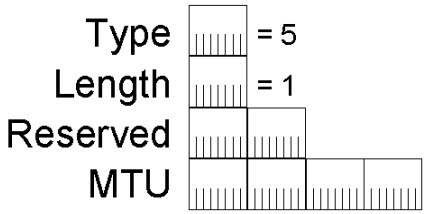


Figura 1-38 Formato opção MTU

Os campos dessa opção têm as seguintes funcionalidades:
Type – Possui o valor igual a 5.
Length – O campo todo tem 8 bytes. O valor setado é 1.
Reserved – Um campo de 16-bits reservado para o uso futuro. É preenchido com o valor igual a 0.
MTU – Indica o MTU IPv6 que deve ser usado pelo host em um dado link onde a mensagem Router Advertisement foi recebida. O tamanho do campo é de 32 bits. Um detalhe importante é o fato do valor contido no campo MTU ser ignorado caso seja maior que o MTU do enlace.

1.11.7. Router Solicitation

É enviada por hosts IPv6 para descobrir a presença de roteadores IPv6 no

enlace. O host envia uma mensagem tipo “multicast Router Solicitation” a fim de que os roteadores IPv6 respondam imediatamente, ou esperando por uma mensagem periódica Router Advertisement. Um pacote de mensagem Router Solicitation é ilustrado a seguir:

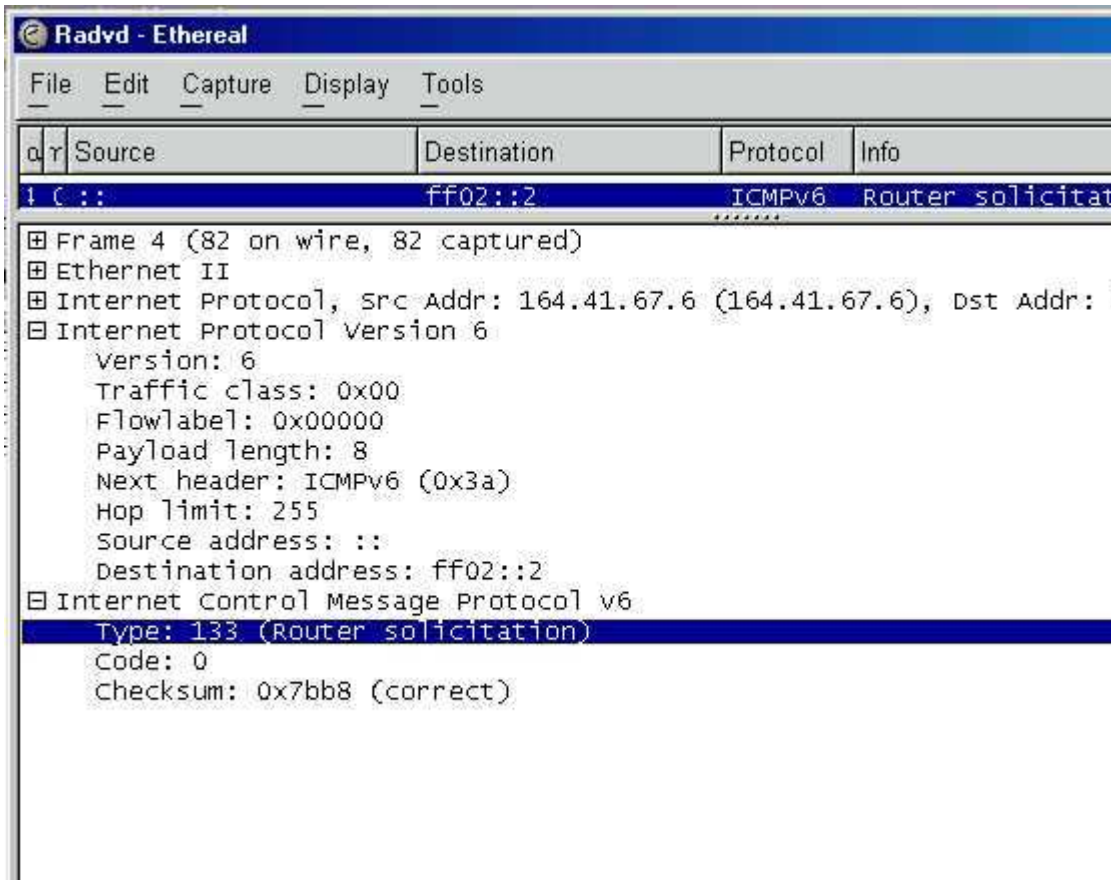


Figura 1-39 Formato da Mensagem Router Solicitation

Alguns comentários sobre os campos seguintes:
Type – Setado em 133.
Code – Valor igual a 0.
Checksum – O valor desse campo é o ICMPv6 checksum.
Reserved – 32-bits reservados para utilização futura que são setados em 0.
Source Link-Layer Address option – Esse campo contém o endereço de camada de enlace da entidade de envio.

1.11.8. Router Advertisement

Roteadores IPv6 enviam mensagens Router Advertisement seja periodicamente ou em resposta ao recebimento de uma Router Solicitation. Esta contém a informação requerida pelo host para determinar: o prefixo de enlace, o MTU do enlace, se utilizar ou não a autoconfiguration de endereço dentre outras.

Seu formato é ilustrado na figura 1-40.

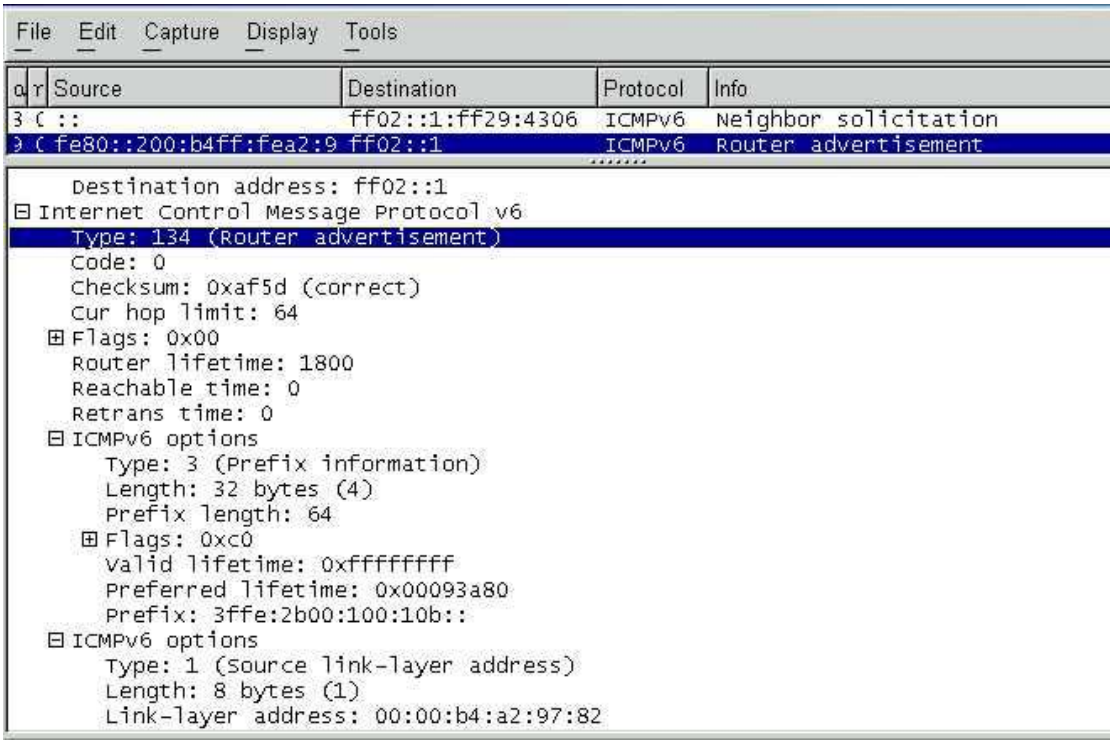


Figura 1-40 Formato da mensagem Router Advertisement

Onde os campos representam:
Type – possui o valor igual a 134.
Code – setado em 0.
Checksum – O valor desse campo é ICMPv6 checksum.
Cur Hop Limit – Indica o valor default do campo Hop Count no cabeçalho IPv6 para pacotes enviados por hosts que recebem essa mensagem Router Advertisement. O valor desse campo é de 8 bits. Um Cur Hop Limit de 0 indica que o valor default do campo Hop Count não é especificado pelo roteador.
Managed Address Configuration flag – Indica, quando setado em 1, que hosts que recebam mensagem Router Advertisement devem usar um protocolo de configuração de endereço stateful (exemplo, DHCPv6) para obter endereços ao invés de endereços derivados da autoconfiguração stateless. O tamanho desse campo é de 1 bit.
Other Stateful Configuration flag – Indica, quando setado em 1, que os hosts ao receberem essa mensagem Router Advertisement devem utilizar um protocolo de configuração de endereços stateful (exemplo, DHCPv6) para obter informação de configuração “non-address”. O tamanho desse campo é de 1 bit.
Reserved – Um campo de 6-bits setados em zero e reservados para o futuro.
Router Lifetime – Indica o tempo de vida (em segundos) do roteador como default. Seu tamanho é de 16 bits. Com isso, o máximo tempo de vida de um roteador se torna 65.535 segundos (por volta de 18,2 horas). Quando um roteador possui o valor desse campo igual a zero indica que ele não pode ser considerado como um roteador. Ou seja, qualquer outra informação é válida.
Reachable Time – Indica a quantidade, em milisegundos, que um nó pode considerar um outro vizinho como sendo alcançável após o recebimento da confirmação de alcançabilidade. O valor desse campo é de 32 bits.
Retrans Timer – Com o tamanho de 32 bits, indica o tempo (dado em milisegundos) entre retransmissões de mensagens de Neighbor Solicitation.

Source Link-Layer Address option – Contém o endereço (camada de enlace) da interface a qual a mensagem de Neighbor Solicitation foi enviada.
MTU option – Contém o MTU do enlace. Deve ser enviado apenas em enlaces de MTU variável ou em ambientes comutados com tecnologias de múltiplas camadas de enlace presentes no mesmo segmento.
Prefix Information options – Contém os prefixos do enlace que são usados para a autoconfiguração de endereços. O prefixo local-link nunca é enviado como esse tipo de opção.

1.11.9. Neighbor Solicitation

A mensagem Neighbor Solicitation é enviada por hosts IPv6 para se descobrir o endereço camada de enlace de um nó IPv6 "on-link". Típicas mensagens Neighbor Solicitation são multicast para resolução de endereços, e unicast quando a alcançabilidade de um nó vizinho é verificada.

O formato da mensagem Neighbor Solicitation é descrito a seguir:

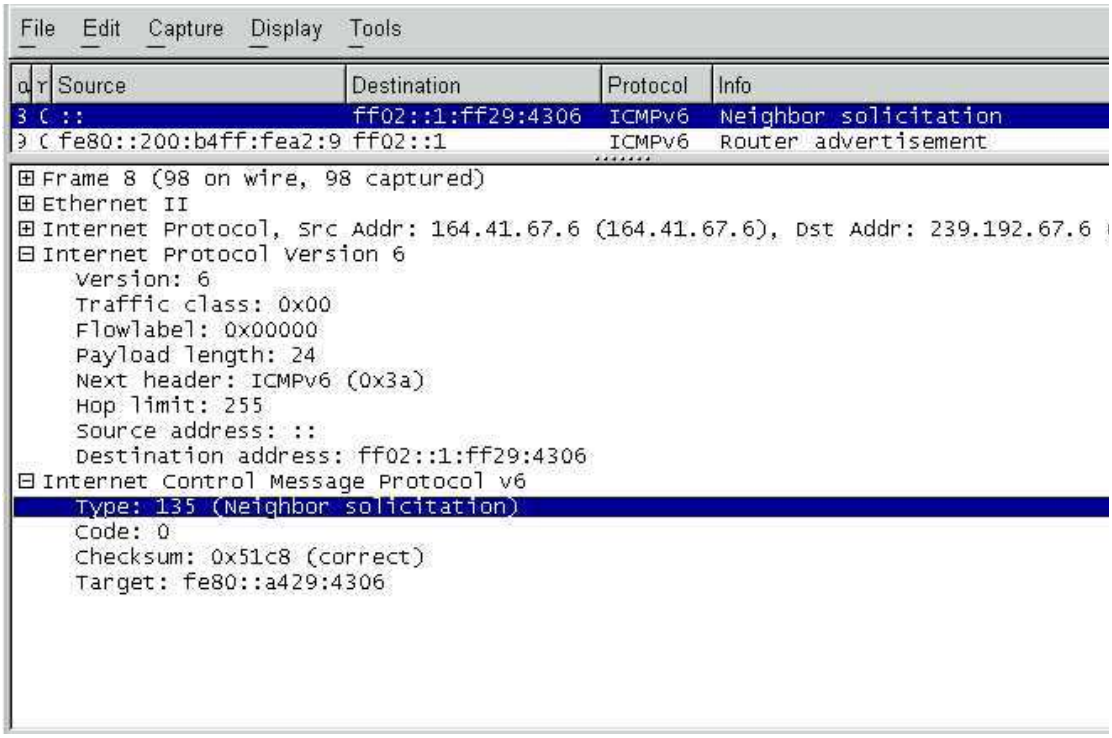


Figura 1-41 Formato da mensagem Neighbor Solicitation

Os campos dessa mensagem têm a função de:
Type – Valor igual a 135.
Code – Valor igual a 0.
Checksum – o valor desse campo é o ICMPv6 checksum.
Reserved – 32-bits reservados para o futuro. São setados em zero.
Target Address – Indica o endereço IP do destino. O tamanho desse campo é igual a 128 bits.
Source Link-Layer Address option – Contém o endereço camada de enlace da origem. Para um nó Ethernet, esse campo conterá o endereço MAC Ethernet do nó que enviou. O endereço nesse campo é usado pelo nó receptor para determinar o endereço MAC unicast do nó o qual o Neighbor Advertisement correspondente é

enviado. Durante a detecção de endereço duplicado, quando o endereço IPv6 origem é o “unspecified address” (::), esse campo não é incluído.

1.11.10. Neighbor Advertisement

Essa mensagem é enviada pelo nó IPV6 em resposta à mensagem de Neighbor Solicitation recebida. Um nó IPv6 também envia um tipo de mensagem chamada “unsolicited Neighbor Advertisements” a fim de informar aos nós vizinhos as mudanças nos endereçamentos do enlace. O Neighbor Advertisement contém informações requeridas pelos nós a fim de determinar o tipo de mensagem, o endereço da origem, e o papel da origem na rede.

O formato da mensagem Neighbor Advertisement está abaixo relacionado:

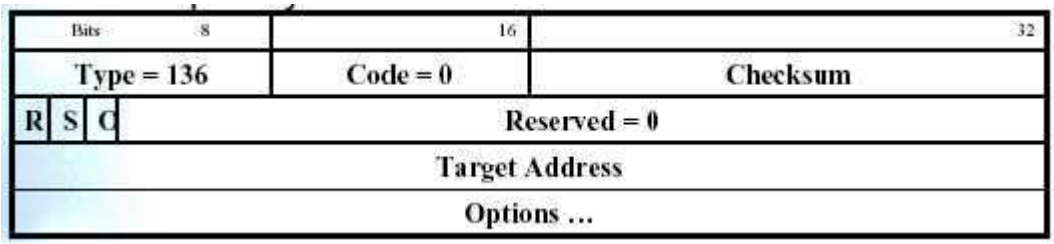


Figura 1-42 Formato da mensagem Neighbor Advertisement

Seus campos são os seguintes:
Type – Seu valor é setado em 136.
Code – Valor do campo igual a 0.
Checksum – O valor do campo é o ICMPv6 checksum.
Router flag – Indica o papel da origem da mensagem Router Advertisement. O tamanho do campo é de apenas 1 bit. Esse campo é setado em 1 quando a origem é um roteador e zero para as demais possibilidades. Esse campo é usado pela Neighbor Unreachability Detection para determinar alterações roteadores x hosts.
Solicited flag – Quando setado em 1, indica a mensagem Neighbor Advertisement é enviada como resposta de uma Neighbor Solicitation. O campo possui um tamanho de 1 bit.
Override flag – Indica, quando setado em 1, que o endereço destino incluso nessa opção deve sobrepor aquele existente na entrada de vizinhos cache. O tamanho do campo é de 1 bit.
Reserved – 29-bits reservados para uso futuro e setados em 0.
Target Address – Indica o endereço destino. O campo possui 128 bits.
Target Link-Layer Address option – Contém o endereço camada de enlace do destino, que é o transmissor do Neighbor Advertisement.

1.11.11. Redirect

Essa mensagem é usada por um roteador para informar a um host um “first-hop” melhor para um destino específico. Essas mensagens são enviadas apenas por roteadores para tráfegos unicast, unicast para hosts, e processadas apenas por hosts.

O formato desse tipo de mensagem é detalhado a seguir:

Bits 8	16	32
Type = 137	Code = 0	Checksum
Reserved = 0		
Target Address		
Destination Address		
Options …		

Figura 1-43 Formato da mensagem Redirect

Os campos possuem os seguintes significados:
Type – O valor desse campo é 137.
Code – Tem valor igual a 0.
Checksum – O valor desse campo é o ICMPv6 checksum.
Reserved – Um campo de 32-bits reservados para o futuro, e setados em 0.
Target Address – Indica o melhor "next-hop" para os pacotes enviados ao destino. O tamanho desse campo é igual a 128 bits. Para um tráfego off-link, esse campo é setado com o endereço local-link do roteador local. No caso do tráfego on-link, esse campo é setado com o campo de endereço de destino na mensagem Redirect.
Destination Address – Contém o endereço de destino do pacote que levou o roteador a enviar a mensagem Redirect. O campo também 128 bits.
Target Link-Layer Address option – Contém o endereço camada de enlace do destino (o nó para onde o pacote deve ser entregue).
Redirected Header option – Inclui uma porção do pacote original que levou o envio da mensagem redirect.

1.12. MECANISMOS DE TRANSIÇÃO

Uma série de técnicas têm sido desenvolvidas e implementadas para efetuar a transição entre ambos os protocolos. Basicamente elas são agrupadas em três categorias.

1.12.1. Perspectiva Dual Stack

Essa técnica permite a coexistência entre IPv4 e IPv6 nos mesmos equipamentos e redes. Os nós e aplicações funcionam fazendo uso do transporte IPv4 e IPv6 assegurando assim a funcionalidade e acessibilidades de serviços.

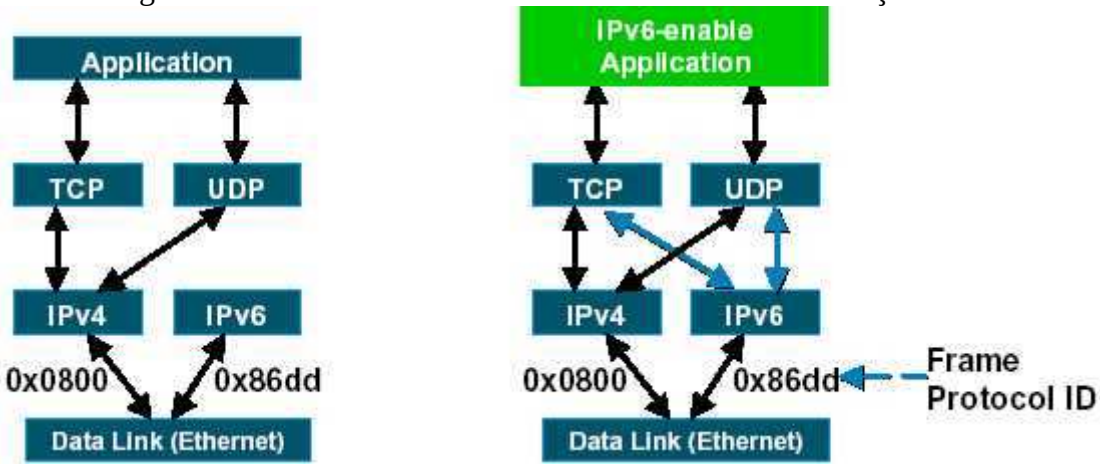


Figura 1-44 Perspectiva Dual Stack

Em outras palavras, um nó Dual Stack:

- Possui ambas as pilhas habilitadas
- Possui aplicações que falam com ambas
- Escolhe a versão do protocolo com base no *name lookup* e na preferência da aplicação.

A acessibilidade a esses serviços pode ser efetuada usando mapeamento a nível de DNS, identificando o nó onde se encontra esse serviço com registo A para IPv4 ou AAAA ou A6 para IPv6.

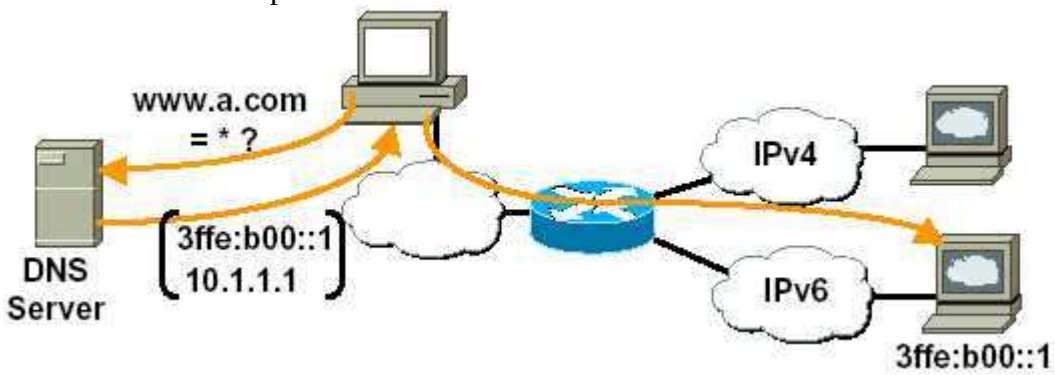


Figura 1-45 Perspectiva lógica Dual Stack

1.12.2. Perspectiva de Tunneling

Esta perspectiva permite encapsular pacotes IPv6 sobre o atual transporte IPv4 permitindo a acessibilidade a nós e serviços IPv6 e é atualmente utilizada no 6Bone, podendo também ser utilizada por prestadores de serviços de telefonia móvel numa primeira fase de disponibilização de serviços de 3ª geração.

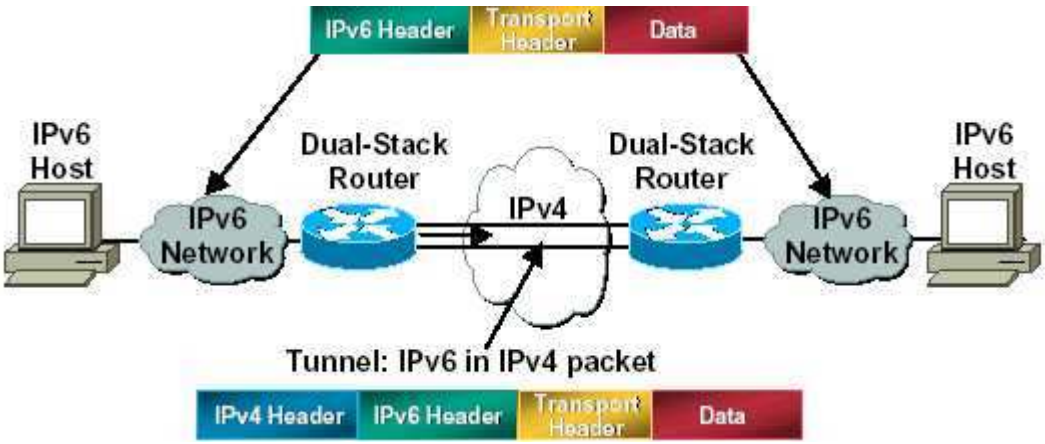


Figura 1-46 Perspectiva Tunneling

- ***Túneis Configurados manualmente***

Neste caso o endereço IPv6 é configurado manualmente numa interface de *tunneling* e endereços IPv4 são também configurados manualmente nas extremidades desse túnel. As extremidades devem suportar transporte IPv4 e IPv6 e podem ser roteadores ou hosts.

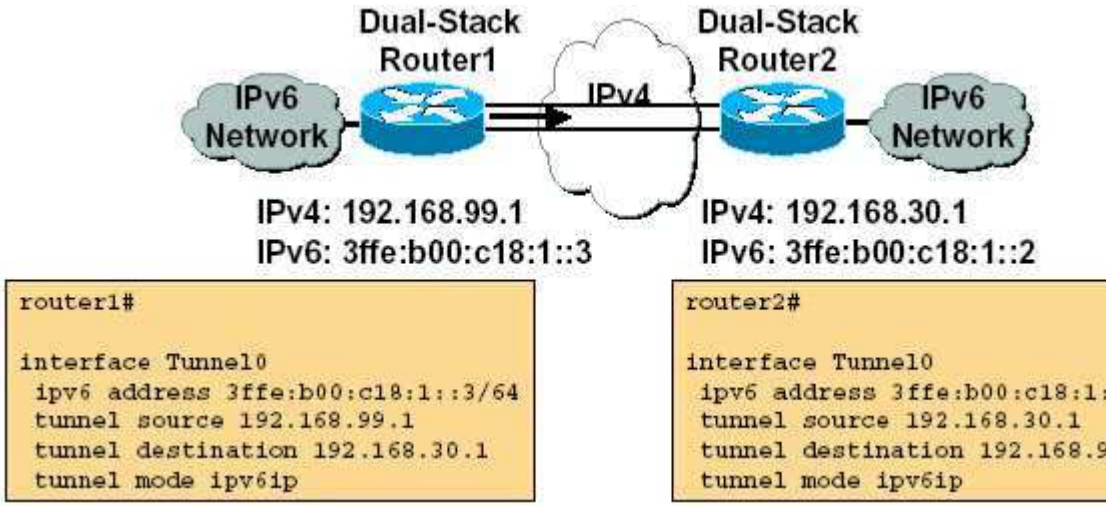


Figura 1-47 Túneis Configuráveis

- ***Túneis Automáticos***

Os endereços IPv6 de origem e destino do túnel são determinados automaticamente usando os 32 bits do endereço IPv4, formando um endereço do tipo IPv6 – IPv4 compatível. Ex: ::194.65.3.21 As extremidades devem suportar transporte IPv4 e IPv6 e podem também ser roteadores ou hosts.

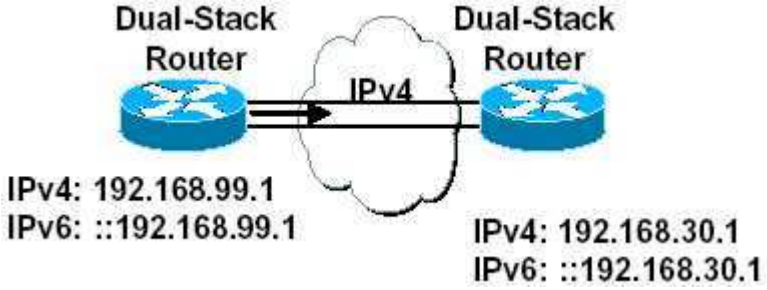


Figura 1-48 Túneis Compatíveis

- ***Túneis 6 to4***

Este tipo de túnel é estabelecido entre *routers* IPv6 sobre uma infra-estrutura IPv4 e é ilustrado na figura 1-49. O endereço IPv6 de origem e destino são determinados pela concatenação do endereço IPv4 com o prefixo 2002::/16 formando um endereço do tipo 2002:194.65.3.20::/48. As extremidades devem suportar transporte IPv4 e IPv6 e podem ser apenas roteadores.

- ***Túnel Broker***

Nesse tipo de túnel a informação é enviada via Http-IPv4. O processo de estabelecimento é dividido em quatro etapas:

- Cliente envia solicitação via IPv4 a entidade Túnel Broker
- A entidade responde via IPv4
- A entidade configura o túnel no servidor de túnel ou no roteador
- O cliente passa a ter um túnel ligado ao servidor ou roteador

Realizamos um teste que será mostrado mais adiante com este tipo de

configuração utilizando uma ferramenta chamada Freenet6.

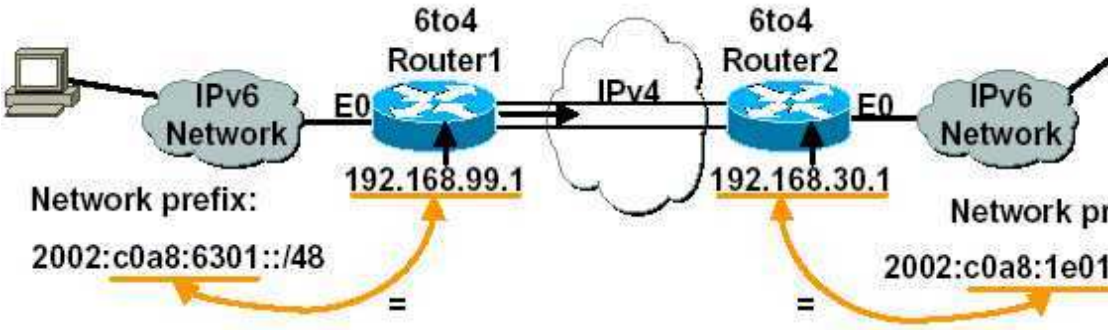


Figura 1-49 Túneis 6to4

1.12.3. Tradução

Por meio dessa técnica equipamentos puramente IPv6 passam a se comunicar com aqueles equipamentos puramente IPv4.

Dentre os mecanismos que utilizam essa técnica encontram-se:

- Translation
  - NAT -PT [RFC 2766]
  - TCP-UDP Relay [RFC 3142]
  - DSTM (Dual Stack Transition Mechanism)
- API
  - BIS (Bump-In -the-Stack) [RFC 2767]
  - BIA (Bump-In-the-API)
- ALG
  - SOCKS-based Gateway [RFC 3089]
  - NAT -PT [RFC 2766]

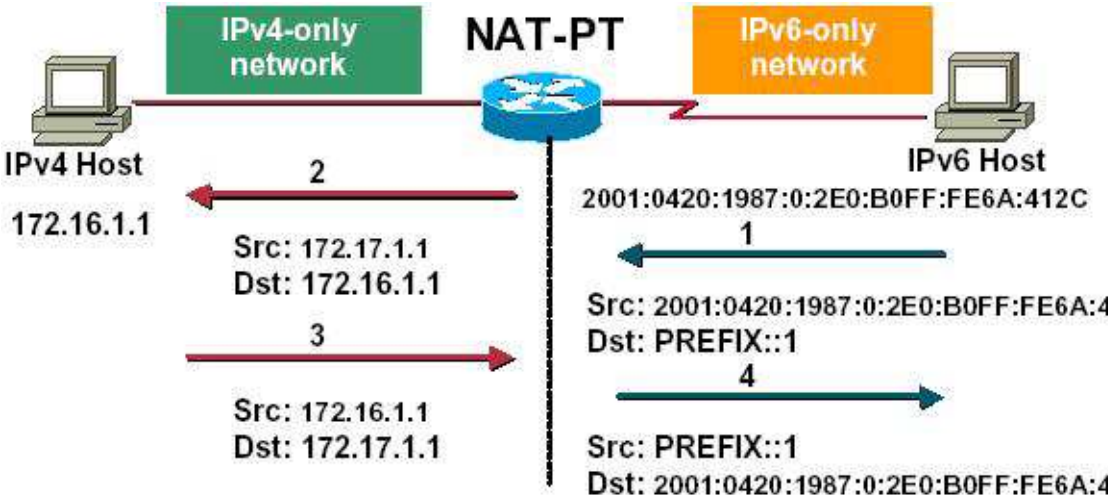


Figura 1-50 Processo de tradução NAT-PT

2. O DOMAIN NAME SYSTEM - DNS

Durante a década de 70, a ARPAnet era uma pequena comunidade de umas

poucas centenas de hosts. Um único arquivo, HOSTS.TXT, continha toda a informação que era necessária para se saber a localização destes hosts. Este arquivo continha um mapeamento nome-endereço para todos os hosts conectados a ARPAnet e mantido pela Network Information Center sendo distribuído de um único host.

Quando a ARPAnet passou a utilizar o TCP/IP, a "população" da rede explodiu. Assim, começaram a ocorrer alguns problemas com o arquivo HOSTS.TXT:

Tráfego:

O pedágio, em termos de tráfego de rede, e uso de processador envolvido na distribuição do arquivo HOSTS.TXT estava ficando insustentável.

Colisões de nome:

Não poderiam haver dois hosts com o mesmo nome dentro do arquivo. Enquanto o NIC (network information center) podia distribuir endereços de uma maneira que eles fossem únicos, ele não tinha autoridade sobre os nomes de host. Não havia nada que evitasse que alguém adicionasse um host com um nome conflitante (que já existisse)

Consistência:

Manter a consistência do arquivo para uma rede em crescente expansão se tornava mais e mais difícil. Quando o arquivo HOSTS.TXT conseguia englobar os pontos mais distantes da rede, um host já tinha mudado de nome ou de endereço, ou ainda, um novo host havia surgido em outro lugar.

Assim, começou-se a estudar um sucessor do HOSTS.TXT, procurando-se criar um sistema que fosse a solução de todos esses problemas. O novo sistema deveria possibilitar o uso de um nome hierárquico para os hosts. Isso iria garantir uma unicidade nos nomes dos hosts.

Paul Mockpetris foi o responsável por desenvolver a arquitetura deste novo sistema. Em 1984, ele escreveu as RFCs 882 e 883, que descrevem o Domain Name System (DNS). Depois, essas RFCs foram sobrepostas pelas RFCs 1034 e 1035, que são as atuais especificações do DNS. Hoje em dia, estas RFCs são adicionadas de varias outras, que descrevem mecanismos de segurança, problemas de implementação, mecanismos dinâmicos de atualização de nomes entre muitas outras coisas.

O DNS é uma base de dados distribuída. Isso possibilita controle local dos segmentos da base de dados como um todo, e ainda sim os dados de cada segmento estão disponíveis em toda a rede, através de um esquema cliente-servidor. Robustez e performance adequados são atingidos. Seu principal objetivo é a tradução de nomes de hosts para seus respectivos endereços IP.

Programas chamados *name servers* constituem a parte servidora do mecanismo cliente-servidor do DNS. Os name servers contém informações sobre alguns segmentos da base de dados, mantendo estes sempre disponíveis para os clientes, que são chamados *resolvers*. Os resolvers são freqüentemente somente rotinas "library" que criam queries e as mandam através da rede para o *name server*.

A estrutura básica do DNS é bastante similar à estrutura de um sistema de arquivos UNIX. Esta estrutura é mostrada na figura 2-1:

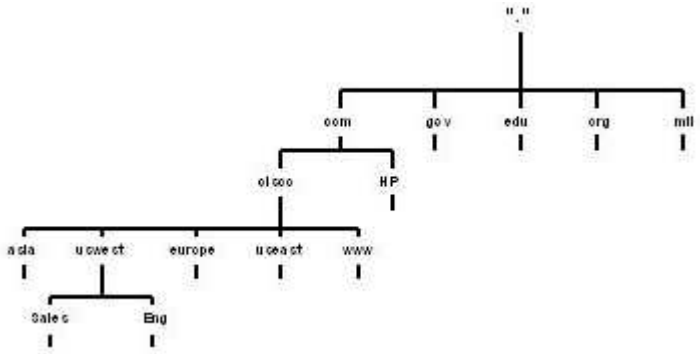


Figura 2-1 Modelo da Hierarquia DNS

Toda a base de dados é feita na forma de uma árvore invertida, com o nó raiz (*root)* no topo, como pode ser visto pela figura acima. Cada nó tem um rótulo texto, que identifica o nó relativo ao seu "pai". Cada nó é também raiz de uma nova sub-árvore, uma espécie de "tronco" que cresce da árvore "principal". Cada uma dessas sub-árvores representa uma parte da base de dados, ou seja, um domínio do DNS. Subdomínios são os "filhos" dos domínios pais.

Cada domínio tem um nome único. O *domain name* em um domínio identifica sua posição na base de dados. No DNS, o domain name é uma seqüência de rótulos do nó da raiz do domínio até o nó da raiz de toda árvore.

No DNS, cada domínio pode ser administrado por uma organização diferente. Cada organização pode "quebrar" seu domínio em um número de sub-domínios e então distribuir a responsabilidade por esses sub-domínios para outras organizações. Um exemplo disso é mostrado na figura 2-2.

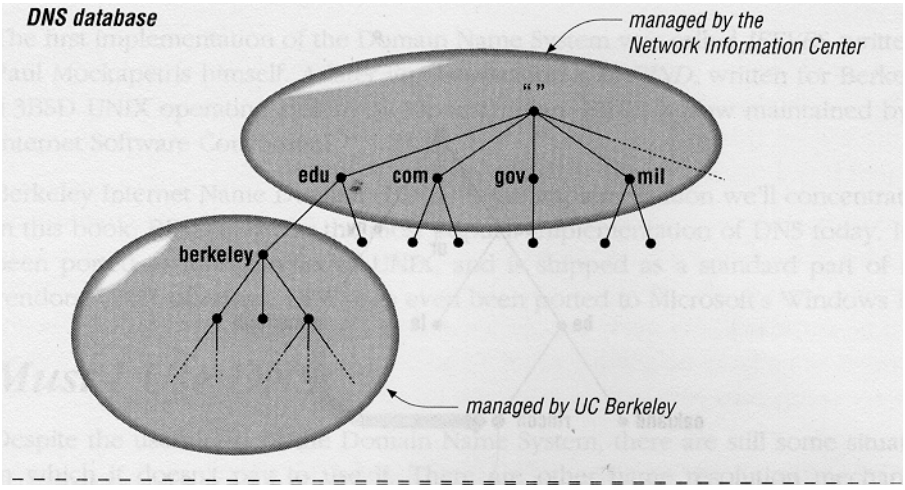


Figura 2-2 Banco de Dados DNS

Por que uma estrutura tão complicada? Para resolver os problemas que o HOSTS.TXT tinha. Por exemplo, fazendo com que os nomes de domínio fossem hierárquicos eliminam-se as colisões de nomes. Cada domínio tem um domain name único, então a organização que dirige este domínio tem a liberdade de nomear os hosts e subdomínios livremente. Qualquer que seja o nome por ela escolhido, este não será conflitante com o nome de hosts de outras organizações, devido ao fato de que eles

terminarão em seus nomes de domínios únicos.

Hoje não contamos com nenhum servidor de DNS no topo da hierarquia com suporte nativo IPv6, assim sendo, toda consulta é realizada usando o atual protocolo, o IPv4.

Domain Names

Cada nó tem um “texto rótulo” que pode ter o comprimento de 63 caracteres. Um rótulo *null* (de comprimento zero) é reservado para o *root*. O domain name completo de qualquer nó na arvore é a seqüência de rótulos no caminho, daquele nó até a raiz. Os nomes de domínio (*domain names)* são sempre lidos do nó em direção à raiz (.) e com pontos separando os nomes no caminho.

O DNS requer que nós que são filhos do mesmo pai tenham nomes diferentes. Esta restrição garante que um nome de domínio identifique um único nó na árvore. Esta restrição não é uma limitação, pois os rótulos somente necessitam ser diferentes dentre “filhos”, não dentre todos os nós da árvore. Por exemplo, se o domínio *fulano.com.br* possui um host chamado *beltrano*, podemos ter o host *beltrano.ciclano.fulano.com.br* e também o host *beltrano.joaoninguem.fulano.com.br*. Porém, nunca poderemos ter dois hosts chamados *beltrano.ciclano.fulano.com.br*.

Domains e sub-domains

Um domínio é simplesmente uma sub-árvore do espaço do domain name. O *domain name* de um domínio é o mesmo que o domain name no topo do domínio. Um exemplo é mostrado na figura abaixo.

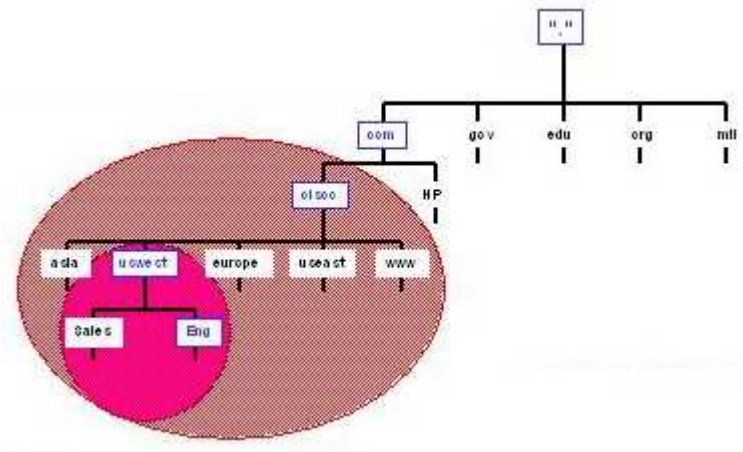


Figura 2-3 Modelo Domínios e Sub-domínios

Note que o domínio é *cisco.com*, dentro do círculo marrom e o subdomínio é chamado *uswest.cisco.com*, dentro do círculo rosa.

Todo domain name de uma sub-árvore é considerado parte de um domínio. Devido ao fato de que um domain name pode estar em muitas sub-árvores, um domain name também pode estar em muitos domínios. Assim, temos que um domínio é apenas uma sub-árvore do espaço de nomes de domínio. Porém, se um domínio é feito apenas de nomes de domínio e outros domínios, onde estão todos os outros

hosts? Domínios são grupos de hosts?

Os hosts são representados por nomes de domínio, que por sua vez são somente “catálogos” dentro da base de dados do DNS. Os hosts são nomes de domínio que apontam para a informação sobre hosts individuais. Um domínio contém todos os hosts que possuem seus nomes de domínio dentro do domínio. Estes hosts estão relacionados logicamente, muitas vezes geograficamente ou por filiação organizacional, e não necessariamente por endereço de rede ou por tipo de *hardware*. Muitas vezes, você poderá encontrar dez hosts diferentes, cada um deles em redes diferentes, muitas vezes até mesmo em países diferentes, porém todos dentro do mesmo domínio.

Uma maneira simples de dizer se um domínio é um sub-domínio de outro domínio é simplesmente comparar seus nomes de domínio. O domain name de um subdomínio termina com o domain name do domínio de seu pai. Por exemplo, o domínio *redes.unb.br* deve ser subdomínio de *unb.br*, pois *redes.unb.br* termina com *unb.br*. Similarmente, *unb.br* é um subdomínio de *.br*.

Tipos de name servers

O DNS especifica dois tipos de name servers: *primários* e *secundários*.

Um name server primário de uma zona lê todos os dados para esta zona de um arquivo em um host.

Um name server secundário de uma zona pega todos os dados da zona de outro name server, que é autoritário para esta zona, chamado servidor mestre (master server). Freqüentemente, o master server de uma zona é o servidor primário desta zona, porém isto não é uma regra, ele pode pegar seus dados de outro secundário. Quando um secundário inicia, ele contata seu master name server e, quando necessário, puxa seus dados de zona. Isso é referido como transferência de zona. Ambos, name server primário e secundário para uma zona são autoritários para aquela zona.

Após criar os dados para uma zona, e colocar um DNS primário para funcionar, não existe a necessidade de copiar os dados host a host para criar novos name servers para a zona. Simplesmente colocamos um DNS secundário para funcionar e carregamos todos os dados do primário no secundário. Assim, o DNS secundário irá transmitir novos dados de zona quando necessário.

Outra importância dos DNS secundários é o aumento de redundância que se tem ao se implementar mais de um DNS em uma zona.

Resolvers

Resolvers são clientes que acessam os name servers. Programas rodando em um host que precisam de informações do espaço de domain name usam o resolver. O resolver faz:

- perguntas ao name server
- interpreta as respostas (que podem ser erros ou registros de recursos)
- retorna a informação para os programas que a requisitaram

No BIND, o resolver é somente um grupo de tabelas que são ligadas a programas

como *telnet* e *ftp*. Nem mesmo um processo separado ele é. Simplesmente gera uma pergunta (query), que é enviada, e espera uma resposta, além de reenviar a query, caso esta não seja respondida. A maioria do trabalho é feito pelo name server.

Tipos de respostas

Existem dois tipos de respostas para servidores DNS:

A resposta authoritative, quando eles têm autoridade sobre a zona (domínio). Pode ser dada por um DNS primário ou um secundário. As respostas são sempre acuradas.

A resposta não authoritative é dada quando o servidor DNS não possui autoridade sobre o domínio. As informações podem ser corretas, porém nem sempre as são. As informações são retiradas do cache da máquina, que pode estar desatualizado.

Processamento de requisições

Vamos utilizar um exemplo para explicar o processamento de requisições. Um host faz uma requisição, procurando o endereço IP de *www.abc.com*, a *root.servers.net*, que não sabe o endereço. Ele retorna o endereço de *gtld-servers.net*, o qual também não sabe o endereço. Por isso, ele retorna o endereço de *nameserver.abc.com.* Assim uma requisição é feita a *nameserver.abc.com*, que possui o endereço IP desejado em sua tabela, e retorna a resposta adequada. A figura abaixo ilustra o processo:

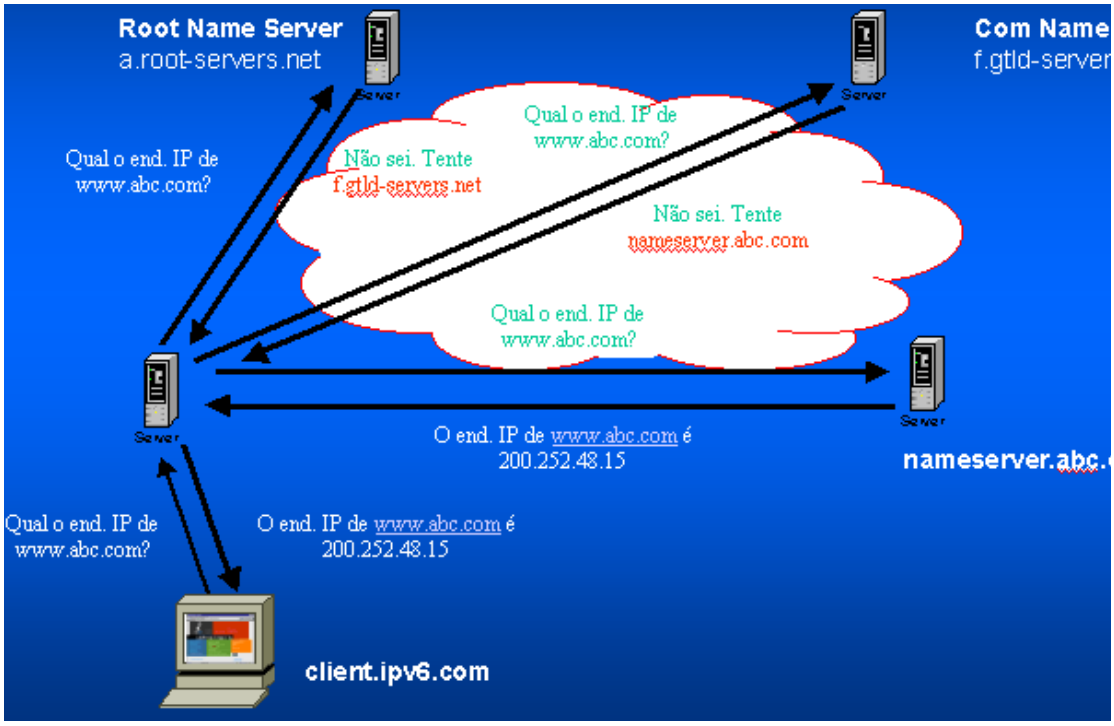


Figura 2-4 Requisições DNS

Tipos de Requisições

Existem dois tipos de requisições que são utilizadas pelo DNS:

Recursivas:

- São normalmente enviadas por clientes (resolvers);
- Perguntam ao NS (name server) por uma resposta e não por uma referência;
- Movem a responsabilidade para o NS de achar o domínio, caso não possua autoridade para isso;
- Assim, o NS requisita a outro NS, caso não tenha a resposta.

Não Recursivas (Interativas)

- Ocorrem principalmente entre name servers;
- São utilizadas quando o NS não sabe um endereço (não é autoritativo);
- O NS irá retornar a sua melhor resposta, ou seja, a que mais se aproxima da solicitação;
- Isso possibilita ao cliente que fez a requisição que continue sua busca.

Mapeamento de endereços-a-nomes

O mapeamento de endereços para nomes é utilizado para produzir uma saída que seja mais fácil para o entendimento, interpretação e memorização do ser humano.

Quando utiliza tabelas, o mapeamento de endereço para nomes é trivial. Ele necessita de uma procura seqüencial direta através da tabela do host para achar um endereço. A procura retorna o nome oficial listado. No DNS, entretanto, mapeamento de endereços para nomes não é tão simples. Dados, incluindo endereços, no *domain name space* são indexados por nome. Dado um domain name, o processo de se achar um endereço é relativamente simples. Entretanto, achar o domain name que mapeia o dado endereço parece exigir uma exaustiva procura de dados anexados a cada domain name na árvore.

Para isso, existe uma solução melhor. Devido ao fato de ser fácil de encontrar dados a partir do momento no qual você tem o domain name que indexa os dados, por que não criar parte do domain name space que usa endereços como rótulo? No domain name space da Internet, esta parte do name space está dentro do domínio *in-addr.arpa.domain*.

Nós no *in-addr.arpa.domain* são rotulados de acordo com os números do endereço IP (números de 32 bits, divididos em quatro grupos de 8 bits por pontos). Assim, poderíamos ter 256 subdomínios, cada um correspondendo por um valor no primeiro octeto do endereço IP. Cada um desses 256 subdomínios poderia ter mais 256 subdomínios, que teriam por si 256 subdomínios cada, e assim por diante, até que tivéssemos quatro “camadas”. A figura 2-5 ilustra exatamente isso.

Podemos ver que quando lido em um domain name, o endereço IP aparece de trás para frente, pois o nome é lido da folha para a raiz da árvore. Por exemplo, se o endereço IP de *winnie.corp.hp.com* é 15.16.192.152, o subdomínio *in-addr.arpa*

correspondente é 152.192.16.15.*in-addr.arpa*, o qual mapeia de volta ao domain name *winnie.corp.hp.com.*

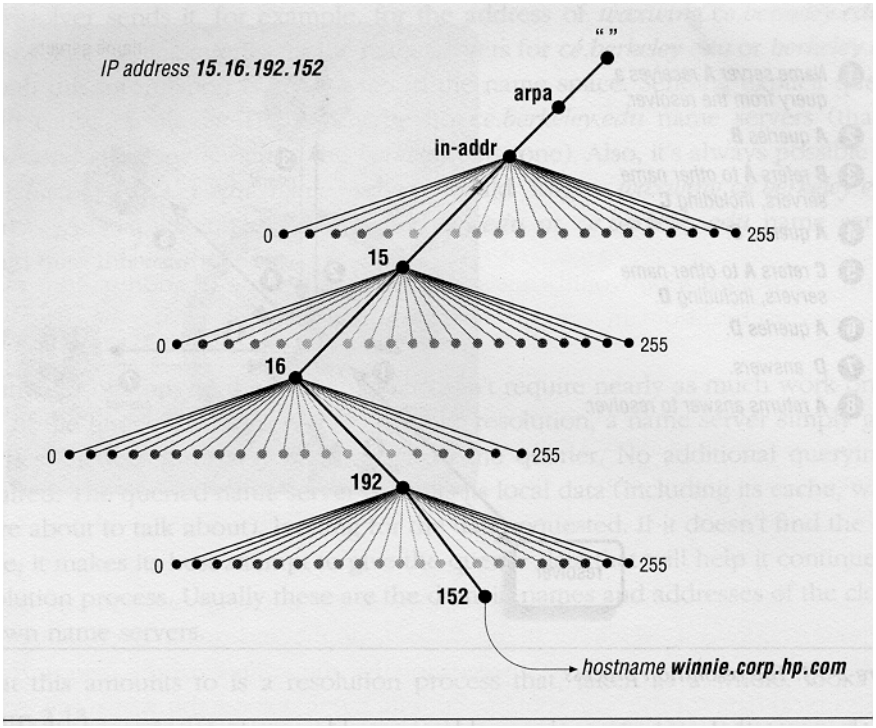


Figura 2-5 Mapeamento Reverso

Mapeamento Reverso

O mapeamento reverso é uma procura por um domain name que indexa dados específicos. Ele é processado somente pelo name server que recebe a query. Este name server procura em todo o seu banco de dados pelo item, e retorna o domain name que o indexa, se possível. Caso ele não consiga encontrar o dado, ele desiste. Não acontecem tentativas de se enviar uma query para outro name server.

Devido ao fato de que qualquer name server somente sabe sobre parte de todo o domain name space, um query reverso nunca garante resposta. Por exemplo, se um mesmo name server recebe um query reverso para um endereço IP o qual ele nada sabe a respeito, ele não pode retornar uma resposta, porém ele também não sabe que aquele endereço IP não existe, porque ele possui apenas parte da tabela de dados DNS.

2.1. BIND

A implementação do DNS na maioria dos sistemas Unix é feita através do software Berkley Internet Name Domain (BIND). O BIND é uma implementação de DNS muito utilizada e, além disso, sua distribuição é gratuita, feita pelo Internet Software Consortium (ISC). Em sua distribuição possui um servidor, uma biblioteca cliente, e alguns programas utilitários. O BIND poderá operar em duas configurações: Slave Name Server (Secondary DNS Server) ou Master Name Server (Primary DNS

Server).

Do ponto de vista conceitual o software de implementação do DNS é dividido em duas componentes - um resolver e um servidor de nomes. O resolver não existe como um processo distinto que ocorre num dado host. Pelo contrário, o resolver é um conjunto de rotinas de software que é “linkado” a qualquer programa que necessite de acesso a endereços.

A base do BIND revela que todos os computadores usam o *resolver code*, mas nem todos os computadores possuem o *name server*. Um computador que não possui um servidor de nomes local e dependa de outros sistemas para a resolução de nomes é denominado por “resolver-only system”. Na maioria dos sistemas Unix existe um servidor local de nomes.

O servidor de nomes BIND existe como um processo distinto, conhecido por *named* (pronunciado por "neime" "d"). Os servidores de nomes são diferentemente classificados, de acordo com o modo em que são configurados. As três principais categorias de servidores de nomes são:

Primary

O servidor primário é aquele servidor a partir do qual todos os dados acerca de um dado domínio são conhecidos. O servidor primário carrega a informação do domínio diretamente a partir de um arquivo no disco local, criado pelo administrador do sistema. Os servidores primários são authoritative, isto é, possuem informação completa e precisa acerca dos seus domínios. Deverá haver apenas um servidor primário por domínio.

Secondary

Os servidores secundários transferem (obtêm) a base de dados completa do domínio a partir do servidor primário. Um arquivo base de dados particular é denominado por arquivo de zona (zone file); à operação de transferência deste arquivo para um servidor secundário, chama-se transferência de um arquivo de zona (zone file transfer). Um servidor secundário assegura que possui informação atualizada de um dado domínio, através da transferência periódica do arquivo de zona. Os servidores secundários são igualmente authoritatives para os seus domínios.

Caching-only

Os servidores caching-only obtêm as respostas de todos os serviços de resolução de nome a partir de outros servidores. Isto é o que caracteriza estes tipos de servidores. São "non-authoritative" o que significa que a sua informação pode não ser precisa, isto é, completa, pois se trata de informação de 2ª mão.

No DNS, existe apenas um server primário por domínio, sendo da responsabilidade do administrador do sistema o preenchimento do arquivo base de dados com a informação relativa ao domínio.

Por questões de segurança e operabilidade, o BIND utilizado será a versão 9.2.1, que não apresenta os *bug*s encontrados nas versões anteriores. Por realizar grandes e complexas funções, o BIND possui um alto potencial para bugs que podem afetar a segurança de um sistema, e, de fato, muitos hackers já exploraram bugs no passado que permitiram ataques remotos para obter acesso como usuário root nos hosts rodando BIND. Para minimizar estes riscos, o BIND pode ser rodado como um usuário não-root, que irá limitar qualquer dano que possa ser feito como um usuário

simples em shell local. Além disso, o BIND pode rodar “enjaulado”.

O principal benefício da “jaula” é que esta irá limitar a porção do sistema em que o daemon do DNS roda, restrito ao diretório raiz da “jaula”. Além disso, já que a “jaula” somente precisa suportar o DNS, os programas relacionados ao BIND na “jaula” podem ser extremamente limitados. Mais importante ainda, é que não há necessidade de programas para mudança de ID, que podem ser usados para se obter acesso como super-usuário root.

2.1.1. Configuração do BIND

O BIND pode ser configurado de modo a existir em modos diferentes. As configurações comuns são as dos sistemas resolver-only, servidores caching-olny, servidores primários e servidores secundários.

Os sistemas resolver-only apenas utilizam o resolver, isto é não necessitam de qualquer execução de daemon. Para configurar um sistema deste tipo apenas é necessário programar o arquivo /etc/resolv.conf.

As três outras configurações, em termos de opções para o BIND, referem-se ao software do servidor de nomes named.

Caching-only: Apenas é necessário um arquivo de cache para a configuração, mas a configuração mais comum também inclui um arquivo de resolução do domínio de loopback.

Primário: A configuração de um servidor primário requer um conjunto de arquivos de configuração; arquivos de zona para o domínio regular (named.hosts) e do domínio inverso (named.rev), o arquivo de boot (named.boot) o arquivo de cache (named.ca) e o arquivo de loopback (named.local). Nenhuma outra configuração necessita deste conjunto de arquivos.

Secundário: Configurar um servidor secundário não requer a criação de arquivos de zona já que os mesmos são transferidos a partir do servidor primário. Contudo os outros arquivos (boot, cache, loopback) são necessários.

Configuração do RESOLVER

A configuração do resolver pode ser feita por 2 processos: configuração *default* e configuração específica.

Esta última é feita através do arquivo /etc/resolv.conf. A configuração por *default* envolve um pouco menos de trabalho, porque o resolver não necessita de ler um arquivo de configuração. O outro processo, através do arquivo /etc/resolv.conf fornece um controle adicional, já que os comandos são colocados no arquivo diretamente pelo administrador do sistema.

A configuração default

Utiliza o host local como name server padrão. Deriva o nome do domínio da string obtida através do comando hostname. Fá-lo removendo a parte da string que precede o primeiro ponto (.) , utilizando o restante como nome de domínio.

Para que a configuração *default* trabalhe, o host local deverá correr o daemon named e o host name deverá estar devidamente definido.

A configuração específica

Utiliza o arquivo de configuração /etc/resolv.conf e possui algumas vantagens sobre a configuração padrão. Para além de definir claramente a configuração do sistema, permite a definição de servers de backup que poderão ser utilizados sempre que o name server *default* não responda.

Embora possam existir algumas pequenas variações nos comandos (específicas dos sistemas), existem duas entradas que são universalmente suportadas:

nameserver address

As entradas nameserver identificam, através dos endereços IPs os servidores que o resolver deverá interrogar para obtenção de informação de domínio. Os name servers são interrogados pela ordem listada no arquivo de configuração e o número máximo de servers varia de sistema para sistema. Se não existirem entradas nameserver ou se o arquivo resolv.conf não existir todas as queries são enviados ao local host.

domain name

A entrada domain define o nome do domínio *default*. O resolver faz o append do nome do domínio a qualquer host name cujo nome não termine por um ponto.

A configuração mais comum para o arquivo /etc/resolv.conf define o nome do domínio padrão, o host local como o primeiro name server e dois backup name servers. Exemplo:

```
domain nuts.com
#try yourself first
nameserver 127.0.0.1
#try almost next
nameserver 128.66.12.1
#finally try filbert
nameserver 128.66.1.2
```

Configuração do NAMED

Enquanto que para configurar o resolver apenas foi necessário um único arquivo, para configurar o daemon named são necessários vários arquivos, a saber:

named.boot: Faz o set de parâmetros gerais do named e aponta para as fontes de informação dos domínios servidos pelo server.

named.ca: Aponta para os root domain servers.

named.local: Utilizado para resolver localmente o endereço de loopback.

named.hosts: Arquivo zona que mapeia os hosts names em endereços IP.

named.ver: Arquivo zona para o mapeamento inverso, isto é, endereços IP em host names.

O arquivo named.boot

O arquivo named.boot aponta o daemon named para as fontes de informação DNS. Algumas destas fontes são arquivos locais; outras se encontram em servidores remotos. Apenas é necessário criar os arquivos referenciados nos comandos primary e cache. A tabela 11 sumariza os comandos de configuração utilizados nos arquivos named.boot

Tabela 11 - Comandos named.boot

Comando	Função
directory	Define o diretório para todos as subseqüentes referências a arquivos
primary	Declara este server como primário para a zona especificada
secondary	Declara este server como secundário para a zona especificada
cache	Aponta para o arquivo cache
forwards	Lista os servidores para os quais os queries são encaminhados
slave	Força o server a usar unicamente os forwarders

Dependendo do modo como o arquivo named.boot é configurado, o name server deve atuar como um primary server, secondary server ou caching-only server.

Configuração de um caching-only server

A configuração do arquivo named.boot, named.local e named.ca é tudo o que é necessário para configurar um caching-only server. O arquivo mais comum para a configuração do named.boot é:

```
primary   0.0.127.IN-ADDR.ARPA    /etc/named.local
cache     .                        /etc/named.ca
```

A única linha mesmo indispensável para a configuração deste arquivo é a segunda, com a indicação de que o arquivo de inicialização da cache é o /etc/named.ca (embora possa ser utilizado qualquer outro nome para este arquivo). No entanto é comum ter uma linha com um comando primary. O objetivo é definir o local server como o primary server para o seu próprio domínio de loopback e dizer que a informação do domínio se encontre armazenada no arquivo named.local. O domínio loopback é um domínio in-addr.arpa que mapeia o endereço 127.0.0.1 no nome local.host.

Configuração de um primary & secondary servers

Nos exemplos que se seguem é utilizado o domínio imaginário nuts.com como base de trabalho para a configuração de um primary e secondary server. Vejamos o arquivo named.boot que define *almond* como primary server para o domínio nuts.com.

```
nuts.com     primary name server boot file.
directory    /etc
primary      nuts.com                    named.hosts
```

```
primary     66.128.IN-ADDR.ARPA     named.rev
primary     0.0.127.IN-ADDR.ARPA    named.local
cache           .                   named.ca
```

O primeiro comando (primary) declara que este servidor é o primary server para o domínio nuts.com e que os dados deste domínio são obtidos a partir do arquivo local named.hosts. Embora se utilize o nome named.hosts, outro nome poderia ter sido utilizado, por ex. nuts.com.hosts.

O segundo comando (primary) aponta para o arquivo que mapeia endereços IP da rede 128.66.0.0 em host names. Este comando indica que o local server é o primary server para o domínio inverso 66.128.in-addr.arpa e que os dados desse domínio são carregados a partir do arquivo named.rev. Novamente outro nome qualquer poderia ter sido utilizado em vez de named.rev.

Os dois últimos comandos no exemplo de configuração são o comando primary para o domínio de loopback e o comando de cache.

A configuração de um servidor secundário difere daquela realizada no primário, substituindo os comandos primary por secondary. Comandos secondary apontam para remote servers como sendo a fonte de informação do domínio em vez de arquivos locais. Começam com a palavra chave secondary, seguida pelo nome do domínio, o endereço do primary server para aquele domínio e finalmente o nome de um arquivo local aonde a informação recebida do primary server será armazenada. O exemplo seguinte mostra a configuração do arquivo named.boot de modo a configurar filbert como secondary server para o domínio nuts.com:

```
 nuts.com                    secondary name server boot file.
directory                    /etc
secondary       nuts.com                    128.66.12.1
        nuts.com.hosts
secondary       66.128.IN-ADDR.ARPA     128.66.12.1             128.66.rev
primary                      0.0.127.IN-ADDR.ARPA               named.local
cache                        .                                  named.ca
```

O primeiro comando secondary define este host como um secondary server para o domínio nuts.com. O comando diz a named para efetuar o download dos dados referentes ao domínio nuts.com a partir do server 128.66.12.1 e para os armazenar no arquivo /etc/nuts.com.hosts.

A linha seguinte nesta configuração indica que este local server é também um secondary server para o domínio inverso 66.128.in-addr.arpa, e que a informação respeitante a este domínio deve ser recolhida de 128.66.12.1., e os dados armazenados no arquivo 128.66.rev.

2.1.2. Procedimentos para a instalação do BIND

Muitas foram as melhorias e mudanças na última versão do BIND, entre elas o suporte ao IPv6. O software BIND se encontra na versão 9.2.1. Fizemos o *download* gratuitamente no endereço ftp://ftp.isc.org/isc/bind9/9.1.2/bind-9.2.1.tar.gz .

O código fonte do software já inclui os fontes do pacote de bibliotecas criptográficas OpenSSL necessários para as funções de segurança disponíveis (DNSSEC e TSIG), mas para uma melhor performance das operações de criptografia é indicado que o pacote seja instalado separadamente e configurado através da opção do autoconf. Adotamos o seguinte procedimento para instalação:

```
$./configure --sysconfdir=/etc
$ make
$ make install
```

Na documentação encontramos que em alguns sistemas, os sockets IPv6 e IPv4 interagem em situações inesperadas. Para reduzir o impacto desses problemas, o servidor DNS não escuta mais requisições de endereços IPv6 por default. Como precisávamos aceitar queries IPv6, especificamos "listen-on-v6 { any; };" nas opções do arquivo named.conf.

Arquivos de Dados de Domínios (Resource records)

Os arquivos de dados do domínio possuem um formato bastante semelhante. Aqui, iremos apenas dar uma breve explicação sobre eles e exemplificá-los. Os arquivos de dados (resource records) devem começar na primeira coluna. Nas RFCs do DNS, os exemplos são apresentados em uma certa ordem, que assim se segue:

SOA (start of authority)

O SOA é a primeira entrada dos resource files, existindo em cada domínio da rede, indicando a autoridade (authority) da zona (domínio). Assim, o SOA indica que este name server é a melhor fonte de informação para dados dentro desta zona.

Abaixo, mostraremos a tabela SOA utilizada no nosso servidor DNS.

```
$TTL 86400
dns.IPv6.br.  IN SOA      projeto.dns.IPv6.br. postmaster@dns.IPv6.br. (
                          2002080203   ;serial
                          3H           ;refresh
                          15M          ;retry
                          1W           ;expiry
                          1D     )     ;minimum TTL
```

O campo serial é referente à data na qual o arquivo foi criado no formato *ano*/*mês*/*dia*/*quantas vezes modificado*.

O campo refresh é referente à freqüência que o DNS secundário checa o DNS primário para uma mudança no valor do campo “serial”.

O campo retry é referente há quanto tempo o DNS secundário deve esperar antes de reconectar ao primário, caso tenha ocorrido uma conexão sem sucesso.

O campo expiry é referente há quanto tempo o servidor secundário deve utilizar sua tabela de endereços atual, caso ele esteja impossibilitado de se atualizar com o servidor primário.

O campo minimum TTL é referente há quanto tempo outros name servers devem fazer o cache, ou salvar esta tabela.

NS (name server)

O NS define a hierarquia de domínios. Assim adicionamos um NS Record para cada name server em nosso domínio ou zona. Os dados armazenados nas tabelas NS são enviados como respostas de name servers referindo-se a outros name servers, quando o endereço IP desejado não pode ser encontrado em sua própria tabela. Abaixo mostramos um exemplo:

www.redes.unb.br.	IN	NS	projeto.dns.IPv6.br.
www.unb.br.	IN	NS	projeto.dns.IPv6.br.

A (address)

O Address é um *resource record* que faz o mapeamento de um nome de host para um endereço IP. Um mesmo host pode ter mais de um endereço IP a ele associado. Esta é a tabela que um name server enviaria a outro name server para atender a uma requisição de endereços. A tabela abaixo foi utilizada pelo DNS por nós configurado.

projeto.dns.IPv6.br.	IN	A	164.41.67.5
projeto.dns.IPv6.br.	IN	AAAA	3ffe:2b00:100:10b::10
maquina.dns.IPv6.br.	IN	AAAA	3ffe:2b00:100:10b::3

Note que os endereços em hexadecimal (AAAA) são endereços IPv6.As figuras 2-6 e 2-7 ilustram o processo de requisição e resposta tipo IPv6. Por meio desses arquivos pode-se analisar melhor os campos e o formato dos pacotes.

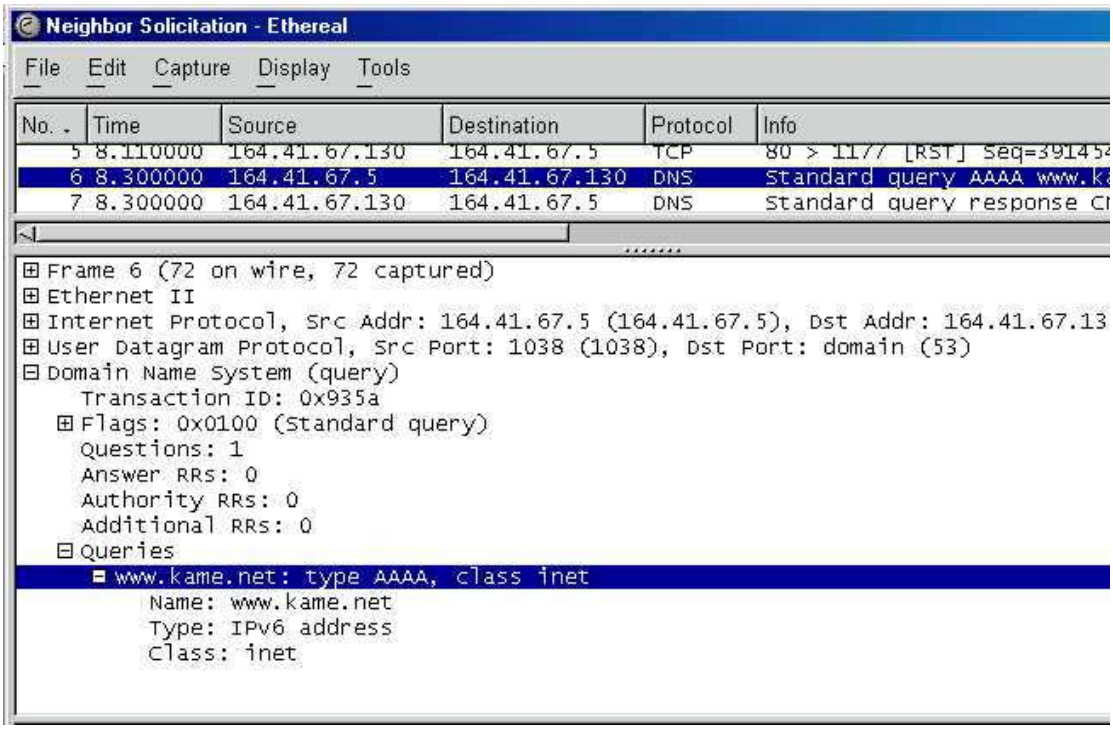


Figura 2-6 Query DNS

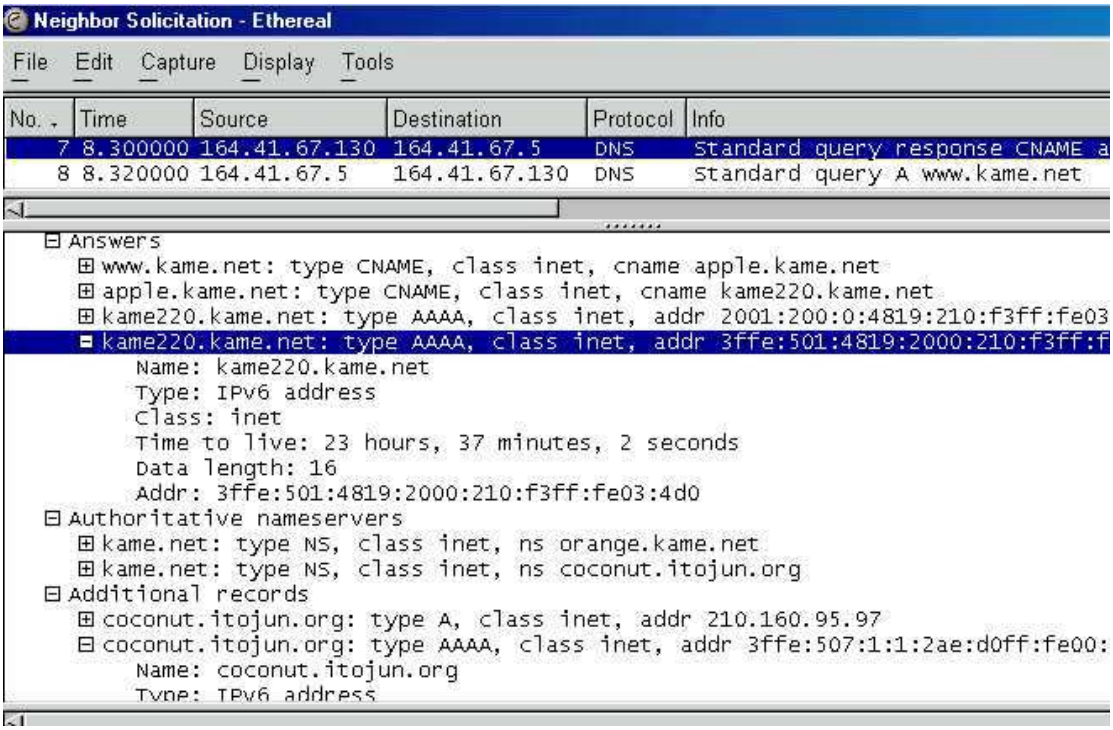


Figura 2-7 Resposta DNS

CNAME (canonical name)

Este é o *resource record* que define um *alias* (apelido) para um host, ou seja, mapeia um apelido (alias) ao seu nome canônico. Assim, ele apenas possibilita que uma máquina seja conhecida por mais de um nome de host. Um host pode ter um número ilimitado de aliases. Deve haver sempre um endereço IP (Address - A) listado para este host antes que um alias possa ser adicionado. A tabela abaixo ilustra o que foi feito por nós no DNS IPv6.

ns.dns.IPv6.br.	IN	CNAME	projeto.dns.IPv6.br.
www.dns.IPv6.br.	IN	CNAME	projeto.dns.IPv6.br.
ftp.dns.IPv6.br.	IN	CNAME	projeto.dns.IPv6.br.
mail.dns.IPv6.br.	IN	CNAME	projeto.dns.IPv6.br.

Assim vemos que os serviços www, ftp, mail e ns estão todos localizados no mesmo host, projeto.dns.IPv6.br, cuja tabela de endereços foi mostrada acima.

PTR (Pointer Record)

O PTR é utilizado exclusivamente em DNS's reversos. O método mais freqüentemente utilizado é chamado "in-addr.arpa". O arquivo de dado de domínio (resource Record) 'PTR' "in-addr.arpa" é exatamente o inverso do resource record 'A'. Isso possibilita que o mapeamento reverso seja feito, ou seja, que o host seja conhecido por seu endereço IP. Por exemplo,

2.67.41.164.in-addr.arpa	IN	PTR	www.redes.unb.br

Isso mostra que o endereço 164.41.67.2 se refere ao host www.redes.unb.br.

Um novo domínio foi criado para a resolução de reverso no IPv6, o **ip6.int**. Esse domínio está em fase de transição para um outro domínio de reverso, o **ip6.arpa**, seguindo as orientações da RFC 3152. A forma de atribuição e delegação de reverso permanece a mesma. Para efeitos de compatibilidade um serviço de DNS deve suportar ambos.

MX (Mail Exchange)

O *resource record* 'MX', possibilita que todo o correio destinado a um certo domínio seja roteado em direção a um host especifico. A tabela abaixo mostra um exemplo:

dogs.com.	IN	MX	10	mail.dogs.com
projeto.dns.IPv6.br.	IN	MX	20	mail.dns.IPv6.br

3. A REDE NACIONAL DE ENSINO E PESQUISA - RNP

A RNP atua desde 1991 no desenvolvimento da tecnologia Internet no país,

consolidando um *backbone* nacional interligando a comunidade acadêmica e atuando em nível nacional a partir de uma Coordenação Nacional distribuída em NA's - Núcleos de Apoio, localizados no Rio de Janeiro, São Paulo, Pernambuco e Distrito Federal.

A RNP é um Programa Prioritário do MCT - Ministério da Ciência e Tecnologia, apoiado e executado pelo CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico. Em 1991, a RNP deu início à introdução da tecnologia Internet no país e vem desempenhando, desde então, um papel de destaque na consolidação do backbone nacional para a comunidade acadêmica, na disseminação de serviços e aplicações de rede Internet e na capacitação de recursos humanos.

Atualmente, a RNP conecta os 27 estados brasileiros, interligando dezenas de milhares de computadores em mais de 800 instituições em todo o país. Diversos centros de pesquisa e instituições de ensino superior fazem uso intensivo da Internet através dos serviços da RNP. O modelo de prestação de serviços de redes operado pela RNP foi concebido para dar suporte à introdução da tecnologia de redes Internet no país, bem como sua difusão e sua capilarização através do apoio à implantação de redes estaduais no país. Esses objetivos foram plenamente atingidos ao longo dos últimos quatro anos.

Deste modo, a Rede Nacional para Ensino e Pesquisa tem como objetivo principal a implantação de um serviço de redes Internet para a comunidade de ensino e pesquisa que atenda aos seguintes requisitos:

. alta qualidade para o tráfego de produção Internet;
. suporte a aplicações de educação superior, em especial, Bibliotecas Digitais;
. interligação das redes metropolitanas de alta velocidade (ReMAVs) para experimentos de novas aplicações em longa distância.

A adequação tecnológica do novo backbone, RNP2, permitirá atender as aplicações previstas pelo MEC e pelo MCT de forma eficiente, pois integrará na mesma infra-estrutura dois tipos de serviços de rede:

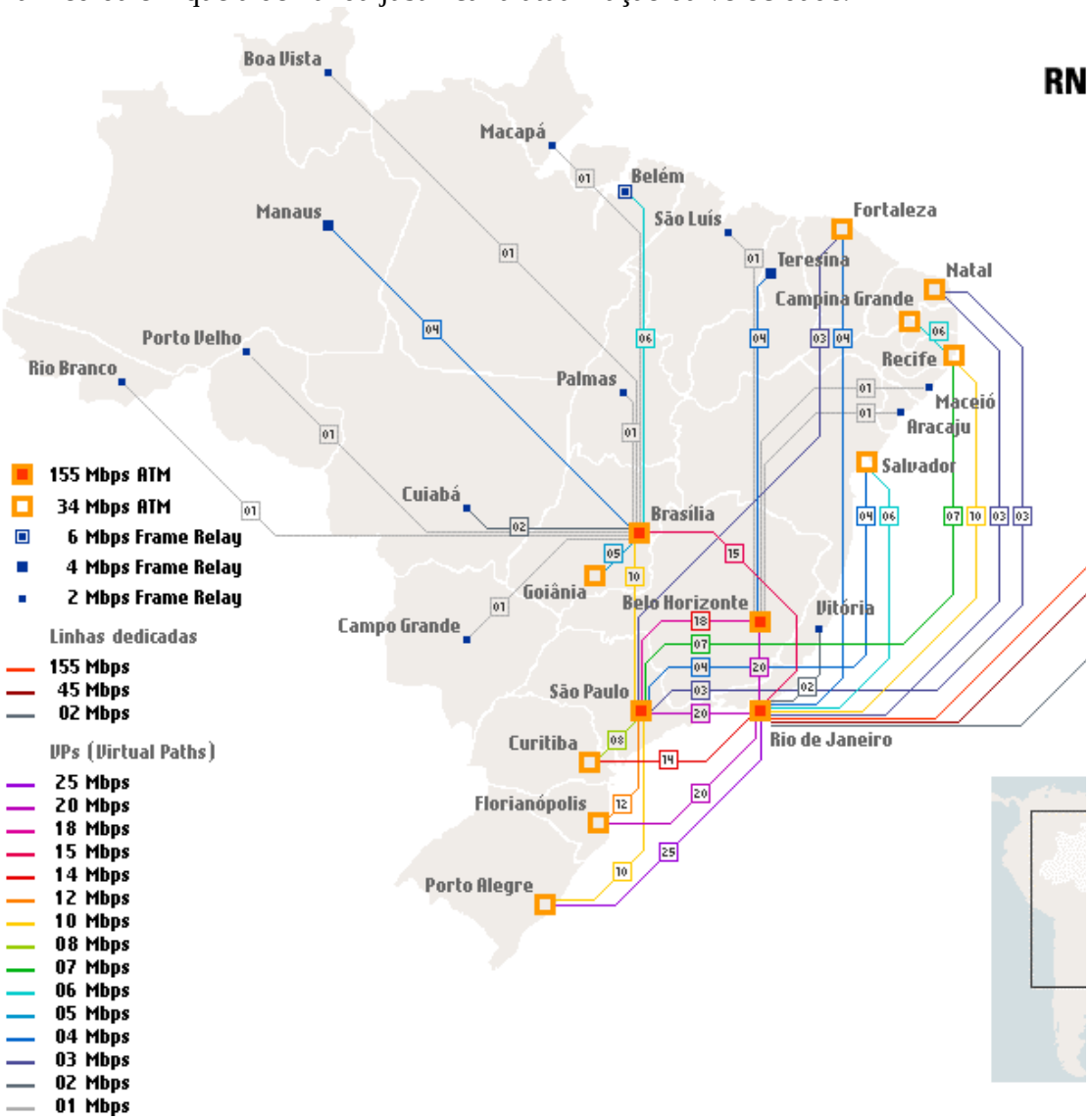
Backbone para produção em que estarão conectadas todas as IFES (Instituições Federais de Ensino Superior) indicadas pela SESu (Secretaria de Educação Superior do MEC) e os Institutos de Pesquisa do MCT/CNPq, além das agências e órgãos dos dois ministérios.

Backbone para experimentação com tecnologia ATM e capacidade para dar suporte às aplicações avançadas Internet2, incluindo bibliotecas digitais, ensino e conferências à distância, entre outras. Interligar as principais Redes Metropolitanas de Alta Velocidade (REMAVs), atualmente em fase de implantação, e estabelecer enlaces de alta velocidade para os Estados Unidos, a fim de integrar o Brasil ao projeto Internet2 e outras iniciativas da Europa, Ásia e Mercosul.

A disponibilidade desta infra-estrutura é um marco estratégico para o suporte ao Programa da Sociedade da Informação, iniciativa do MCT que tem como foco a utilização das tecnologias da informação para o desenvolvimento. A proposta deste programa multi-institucional, ainda em preparação, é impulsionar a utilização das tecnologias da informação no país, com forte articulação entre os setores público, privado e acadêmico.

O Backbone da RNP

O backbone RNP2 foi projetado para atender a requisitos técnicos de aplicações avançadas e começou a ser implementado em julho de 2000. Foi utilizada tecnologia ATM para os Pontos de Presença (PoPs) que concentram maior fluxo de tráfego de dados e *Frame Relay* para os PoPs com menor tráfego. Há 27 PoPs instalados nas principais cidades e capitais do país. A velocidade das Portas de Acesso dos PoPs, de até 155 Mbps, garante o atendimento da soma das diversas conexões virtuais estabelecidas (VP) e permite a elevação da largura de banda dessas conexões na medida em que a demanda justificar a atualização da velocidade.



A RNP2 possui três conexões internacionais próprias. Uma, de 155 Mbps, é usada para tráfego Internet de produção e será ligada à rede Internet2 através do STAR TAP de Chicago. Outra, de 45 Mbps, está ligada à Internet2 através do GigaPoP da Flórida e destina-se exclusivamente a interconexão e colaboração entre redes acadêmicas dentro do projeto Americas Path (AmPath). A conexão mais recente, com capacidade de 2 Mbps, liga o RNP2 à rede acadêmica portuguesa RCTS, da Fundação para a Computação Científica Nacional. O backbone interliga todas as ReMAVs, Instituições Federais de Ensino Superior (IFES) e Unidades de Pesquisa do

MCT.

Figura 3-1 Backbone RNP

3.1. O 6BONE

O 6Bone, *backbone* IPv6 coordenado pelo NGTrans (Next Generation Transition Working Group), é um teste de campo para auxiliar na evolução, no desenvolvimento e no aperfeiçoamento do novo protocolo. Atualmente integra pelo menos 58 países, dentre os quais também o Brasil desde janeiro de 1998.

Operacional desde junho de 1996, este *backbone* é implementado através de uma rede virtual sobre a rede física IPv4 da atual Internet. A rede virtual é composta de redes locais IPv6 ligadas entre si por túneis ponto-a-ponto IPv6 sobre IPv4. Os túneis são realizados por roteadores com pilha dupla (IPv6 e IPv4) com suporte para roteamento estático e dinâmico (RIPng e BGP4+), e as redes locais IPv6 são compostas por estações com sistemas operacionais com suporte a IPv6 ou com pilha dupla (IPv4 e v6).

ENDEREÇAMENTO NO 6BONE E NO BR-6BONE

Como apresentado na Internet *Draft* IPv6 Testing Address Allocation, o formato de um endereço *unicast* global é o seguinte:

FP	TLA ID	NLA ID	SLA ID	Interface ID

onde,
FP: *Format Prefix* (3 bits) - prefixo utilizado para identificar endereços *unicast* globais, com FP = 001 em binário;
TLA ID: *Top-Level Aggregation Identifier* (13 bits);
NLA ID: *Next-Level Aggregation Identifier* (32 bits), designado para identificar redes de trânsito e *sites* de usuários finais, de acordo com a arquitetura do 6Bone;
SLA ID: *Site-Level Aggregation Identifier* (16 bits), identifica a rede do usuário final;
Interface ID (64 bits) - identifica a interface de cada sistema (roteador, servidor, estação, etc.).

Para as atividades do 6Bone, a IANA (*Internet Assigned Numbers Authority*) alocou o prefixo TLA como sendo 0x1FFE. A partir dessa alocação, o 6Bone definiu seu esquema de endereçamento com a estrutura de endereços especificada no documento Test Address Usage, da forma:

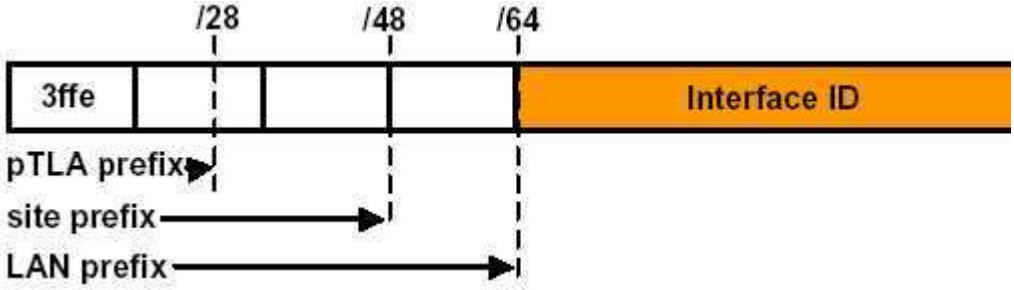


Figura 3-2 Formato Endereço 6Bone

onde,
0x3FFE (16 bits): é o prefixo formado pelo FP e pelo TLA já definidos anteriormente;

NLA1 (8 bits): primeira parte da divisão do NLA original, também chamado de pTLA (pseudo *Top-Level Aggregation*). É o prefixo alocado para cada *backbone* ou instituição participante do 6Bone;
NLA2 (24 bits): segunda parte da divisão do NLA original;
SLA ID: *Site-Level Aggregation Identifier* (16 bits), identifica a rede do usuário final;
Interface ID (64 bits) - identifica a interface de cada sistema (roteador, servidor, estação, etc.).

Assim, os endereços do 6Bone apresentam o prefixo 3FFE:nn00/24, onde "nn" representa o pTLA alocado e 24 é o tamanho do prefixo em bits.

Em janeiro/98 foi alocado para o Brasil o prefixo 3FFE:2B00/24. Para o Backbone IPv6 brasileiro, chamado de Br6Bone, o esquema de endereçamento foi definido no seguinte formato:

3FFE:2B	NLA3	NLA4	SLA ID	Interface ID

onde,
3FFE:2B00 (24 bits) é o prefixo pTLA alocado para o Brasil;
NLA3 (8 bits): para trânsito ou redes com ISPs intermediários, utilizamos mais 8 bits (primeira parte do NLA2). Quando em redes sem ISPs intermediários, o NLA3 é 00 (3FFE:2B00). Este esquema está em acordo com as recomendações do documento Test Address Usage. Denominaremos este campo de pNLA1 (pseudo NLA1), para prefixos a serem alocados a grandes backbones no Brasil;
NLA4 (16 bits): segunda parte da divisão do NLA2 do esquema de endereçamento do 6Bone;
SLA ID: *Site-Level Aggregation Identifier* (16 bits). Recomendamos sua divisão em duas partes (SLA1 e SLA2) de 8 bits cada, como forma de utilização de subredes e melhor administração do espaço de endereços;
Interface ID (64 bits) - identifica a interface de cada sistema (roteador, servidor, estação, etc.).

3.2. O BACKBONE IPV6 BRASILEIRO

Para o Br6Bone, a RNP dispõe de um túnel IPv6 sobre IPv4 implementado com a Cisco/USA, em pleno funcionamento desde abril/98. Através desse túnel, tem-se conectividade com o 6Bone. Internamente, o backbone IPv6 da RNP já possui 6 PoP's operando nativamente em SP, RJ , BA, RN, MG e RS. A figura 3-3 foi montada de acordo com informações conseguidas no site do BR6bone e com o pessoal da RNP, e mostra como está estruturado o *backbone* IPv6 brasileiro.

Além de testar as diversas aplicações desenvolvidas ou ainda em desenvolvimento, o Br6Bone serve também para fazer testes em várias áreas como: conexões *multihomed*, roteamento com BGP4+, RIPng e IGRPng, aplicações *multicasting*, servidores de nomes IPv4 e IPv6, conexões IPv6 sobre IPv4 e IPv4 sobre IPv6, NAT (*Network Address Translation*) de IPv6 para IPv4 e vice-versa, DHCPv6, auto-configuração, "tunelamento", IPSec, etc.

Para adesão de uma instituição ao Br6Bone, tivemos que seguir alguns pré-requisitos que são os mesmos utilizados no 6Bone mundial; basicamente no que se refere às recomendações e especificações da RFC 1933 - Transition Mechanisms for IPv6 Hosts and Routers e da Internet *Draft* 6Bone Routing Practice, além, é claro, do comprometimento de disponibilizar e publicar informações de pesquisas e de testes relevantes ao projeto. A documentação enviada está anexada no final do trabalho.

O IETF NGTRANS WG e o 6BONE

A IETF é uma sociedade aberta da qual participam pesquisadores, projetistas, operadores de telecomunicações e de provedores de serviços Internet, bem como fabricantes de equipamentos. Todos são voluntários e estão, direta ou indiretamente, relacionados com a arquitetura da Internet, com a especificação e o desenvolvimento de protocolos de comunicação e aplicações, ou com a operação, a segurança e o gerenciamento desta rede.

O NGTrans, já citado anteriormente, é um grupo de trabalho da IETF que visa estudar e definir os mecanismos e procedimentos para suportar a transição da Internet do IPv4 para o IPv6. Sua estratégia se baseia em:

. produzir um documento detalhando a infra-estrutura, como será e o que será necessário para a transição;

. definir e especificar os mecanismos obrigatórios e opcionais a serem implementados pelos fabricantes nos *hosts*, roteadores e outros equipamentos de rede, a fim de suportar o período de transição;

. articular um plano operacional concreto a ser executado pelos ISPs (*Internet Service Providers*) quando da transição entre o IPv4 e o IPv6.

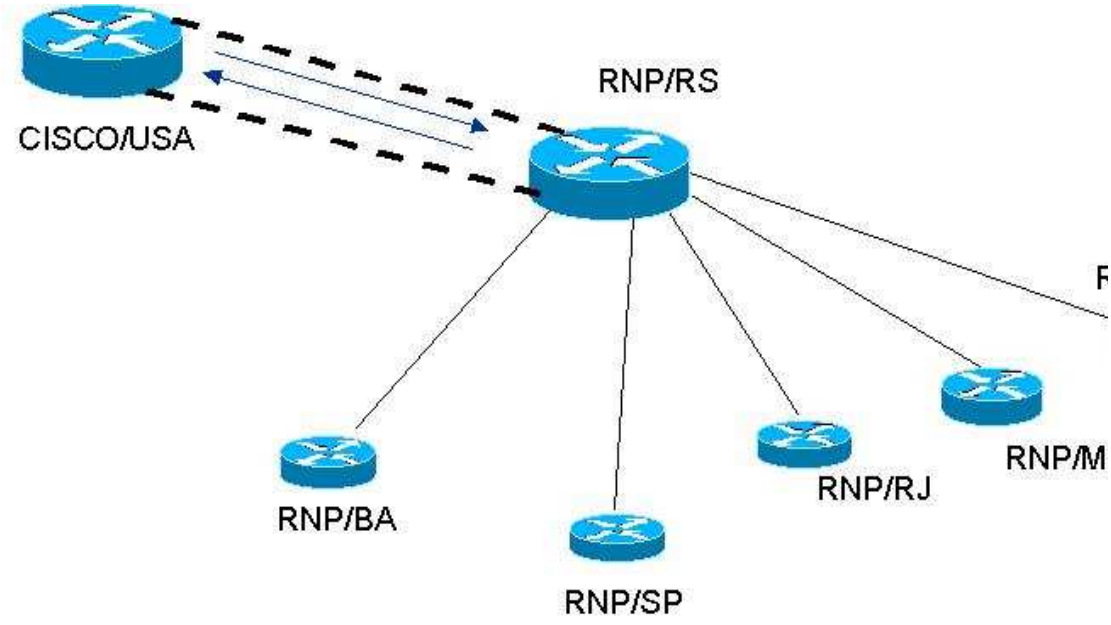


Figura 3-3 Backbone IPv6 brasileiro

4. IPV6 E O LINUX

O Linux é um sistema operacional livre. Foi escrito por Linus Torvalds com a assistência de um grupo técnico altamente capacitado através da Internet. O Linux possui todas as características presentes nos mais modernos sistemas operacionais, incluindo multi-tarefa real, memória virtual, *shared libraries* (bibliotecas de "linkagem" dinâmica), carregamento de drivers sob demanda, suporte nativo a redes TCP/IP, fácil integração com outros sistemas operacionais e padrões de rede, nomes longos de arquivos, proteção de acesso a recursos compartilhados, suporte a vários idiomas e conformância com os mais respeitados padrões internacionais.

O Linux tem suporte ao IPv6 desde a sua versão de kernel 2.1.8 quando foi lançado um *patch* em novembro de 1996 por Pedro Roque.

Procedimentos de Instalação

Para ativar o suporte ao IPv6 tivemos que recompilar o kernel. Em nosso caso utilizamos a distribuição Red Hat 7.1 e fizemos o *download* do kernel 2.4.9. Seguimos os seguintes passos durante a instalação:

```
# tar xvzf linux-2.4.9.tar.gz
# mv linux/ /usr/src/linux-2.4.9
# cd /usr/src
# ln -s linux-2.4.9 linux
# cd /usr/src/linux
# make menuconfig
```

Neste ponto, selecionamos as opções que atenderiam às nossas necessidades. NOTA: As opções podem variar conforme a versão do kernel.

Em "Code maturity level options":

[*] Prompt for development and/or incomplete code/drivers

As demais opções para o suporte ao IPv6 se encontram em "Networking Options":

<*> The IPv6 Protocol
[*] enable EUI-64 token format (de acordo com o novo padrão de formato de endereçamento usado no 6Bone o EUI-64)
<*> IP: tunneling (suporte ao tunelamento)
<*> IP: GRE tunnels over IP

E sendo essa máquina o gateway e firewall de nossa rede IPv6, selecionamos as

opções:

[*] IP: optimize as router not host
[*] IP: advanced router
[*] IP: policy routing
[*] Network firewalls
[*] IP: firewalling
[*] IP: transparent proxy support
[*] IP: masquerading
[*] IP: ICMP masquerading
[*] IP: fast network address translation

Uma vez reconfigurado o sistema, salvamos as definições e continuamos a compilação do kernel:

```
# make dep
# make clean
# make bzImage
# make modules
# make modules_install
```

Para instalar o novo kernel:

```
# cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-2.4.9-IPv6
# vi /etc/lilo.conf
      default=linux-IPv6
      image=/boot/vmlinuz-2.4.9-IPv6
      label=linux-IPv6
      read-only
      root=/dev/hda1
# lilo -v
```

Reiniciamos a máquina e agora seguimos utilizando o novo kernel com suporte ao IPv6. Como nossa intenção era de que a máquina atuasse como roteador de pacotes IPv6 interligando uma rede a um túnel, foi necessário ativar o *forwarding* de pacotes:

```
# echo "1" > /proc/sys/net/IPv6/conf/all/forwarding
```

Para o suporte a rede IPv6, modificamos os arquivos de configuração de hosts, /etc/hosts, acrescentando:

```
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
ff02::3     ip6-allhosts
```

Não foi necessário modificar o arquivo de configuração de protocolos, /etc/protocols, que já mostrava o suporte ao IPv6:

```
IPv6            41      IPv6        # IPv6
```

```
IPv6-route      43      IPv6-Route  # Routing Header for IPv6
IPv6-frag       44      IPv6-Frag   # Fragment Header for IPv6
IPv6-crypt      50      IPv6-Crypt  # Encryption Header for
IPv6
IPv6-auth       51      IPv6-Auth   # Authentcation Header for
IPv6
icmpv6          58      IPv6-ICMP   # ICMP for IPv6
IPv6-nonxt      59      IPv6-NoNxt  # No Next Header for IPv6
IPv6-opts       60      IPv6-Opts   # Destination Options for
IPv6
```

4.1. PROCEDIMENTOS DE CONFIGURAÇÃO DO TÚNEL

Basicamente, a configuração se resume ao uso correto dos aplicativos **ifconfig** e **route**, bastando usá-los da mesma forma que se faria em uma máquina IPv4, só que agora utilizando-se endereços IPv6.

Para iniciarmos a rede, colocamos os comandos apropriados em um script próprio para a inicialização das interfaces a ser executado no startup do sistema, no nosso caso, no arquivo rc.local em /etc/rc.d.

O endereço alocado para nossa rede foi 3FFE:2B00:0100:010B::/64 correspondendo a um Leaf Site, possuindo uma capacidade de endereçamento de 64 bits para os hosts da rede (elevando o número 2 a 64 vê-se a quantidade de dispositivos que podem possuir endereçamento nessa rede IPv6). A máquina que estabelecerá o túnel IPv6-IPv4 deve possuir um endereço IPv4 configurado, rodando e válido na Internet. Este endereço é conhecido pela outra ponta do túnel para efeito de configuração, no nosso caso o pessoal da RNP, bem como nós devemos saber qual endereço IPv4 a outra ponta do túnel possui. Em nosso caso estamos estabelecendo um túnel com o CEO/RNP e o endereço da outra ponta é 200.136.100.141; a máquina que deve realizar o roteamento possuirá o IP 3FFE:2B00:0100:010B::1.

Assumindo que o endereço IPv4, 164.41.67.5, já está configurando e rodando na máquina router-tunnel da rede, devemos então ativar o endereço IPv6 na interface de rede, então ativar o dispositivo (*device*) de tunelamento sit0, criando um novo dispositivo de túnel sit1, o qual realizará o túnel para o endereço IPv4 da outra ponta. O nome *sit* é uma sigla para *Simple Internet Transition*. Esses dispositivos tem a capacidade de encapsular pacotes IPv6 dentro de pacotes IPv4 para serem transportados em um túnel para a outra ponta. *Sit0* é um dispositivo especial e não pode ser usado para túneis dedicados.

Ativamos em seguida esse túnel e uma rota para o 6Bone (3FFE:/16), através do túnel estabelecido pelo dispositivo sit1, observe o formato IPv4 compatível do *gateway* para essa rota nos comandos que implementam abaixo os passos da configuração e que foram colocados no script de inicialização da rede:

```
# Adicionando Configuracao IPv6
ifconfig eth0 add 3ffe:2b00:0100:010b::1/64
ifconfig eth1 add 3ffe:2b00:0100:010b::3/64

# Ativar roteamento de IPv6 e Ipv4
echo "1" > /proc/sys/net/IPv6/conf/all/forwarding
echo "1" > /proc/sys/net/IPv4/ip_forward

# Criar Interface de Tunel
/sbin/ifconfig sit0 up

# Configurar endereço IPv4 da ponta remota do tunel
/sbin/ifconfig sit0 tunnel ::200.136.100.141

# Ativar tunel com RNP
/sbin/ifconfig sit1 up

# Configurar endereço IPv6 da ponta local do tunel
ifconfig sit1 inet6 add 3ffe:2b00:500:31::2/126

# Configurar rota default
/sbin/route -A inet6 add ::0/0 gw 3ffe:2b00:500:31::1 dev sit1


# Configurar rotas do tunel
/sbin/route -A inet6 add 3ffe:2b00:500:31::0/64 dev sit1
/sbin/route -A inet6 add 3ffe::/16 dev sit1
/sbin/route -A inet6 add 2000::/3 dev sit1
/sbin/route -A inet6 add 3ffe:2b00:100:10b:280::0/64 dev eth1

#Script RADVD
/usr/local/sbin/radvd &
```

A última linha ativa um utilitário de *Router Advertisement - RADVD,* o qual ouve as solicitações de router e envia anúncios de routers, permitindo a autoconfiguração de hosts e a escolha do *gateway default* pelos mesmos.

No arquivo radvd.conf localizado em /usr/local/, configuramos a interface ligada à rede IPv6 e também o escopo de endereçamento que deve ser usado:

```
# interface eth1
{
      AdvSendAdvert on;
      prefix 3ffe:2b00:0100:010b::0/64
      {
            AdvOnLink on;
            AdvAutonomous on;
      };
};
```

Essas opções configuradas basicamente querem dizer que o radvd deve anunciar/publicar (AdvSendAdvert on) o prefixo 3ffe:2b00:0100:010b::0/64 através da interface eth1. O prefixo é setado também como autônomo (AdvAutonomous on) e como on-link (AdvOnLink on). Todas as outras opções foram deixadas com valores default.

A opção Autônomo significa que o prefixo pode ser usado em configurações de endereçamento automáticas. Já o outro valor (on-link) significa que todos os hosts podem ser alcançados pela interface que o host (router) recebe a Router Advertisements (RA).

Para fins de teste, utilizamos o ping6, netstat, traceroute6, tracepath6 e o próprio route, nas suas novas versões com o suporte ao IPv6:

ping6 ::1
ping6 3ffe:2b00:010B:0100::1
route –A inet6
netstat –nlptu
traceroute6 3ffe::1
tracepath6 3ffe::1

A figura 4-1 ilustra como ficou a nossa configuração, sendo que o nosso Router1 é uma máquina Linux com duas interfaces de rede. No meio do caminho os pacotes passam por diversos roteadores, entre eles podemos citar:

1. 164.41.67.1 [swicth laborat=F3rio redes]
2. * * * [identificação não disponível]
3. 164.41.2.3 [roteador UnB]
4. 200.130.103.9 [rt.pop-df.rnp.br] A partir daqui já estamos na RNP
5. 200.19.119.65 [bb3.pop-df.rnp.br]
6. 200.143.254.138 [rj.bb3.rnp.br]
7. 200.143.254.93 [rj7507-fast6_1.bb3.rnp.br]
8. 200.143.255.9 [ambrosia.nc-rj.rnp.br]
9. 200.17.63.190 [janeway_e.nc-rj.rnp.br]
10. 200.17.63.148 [lago.nc-rj.rnp.br]

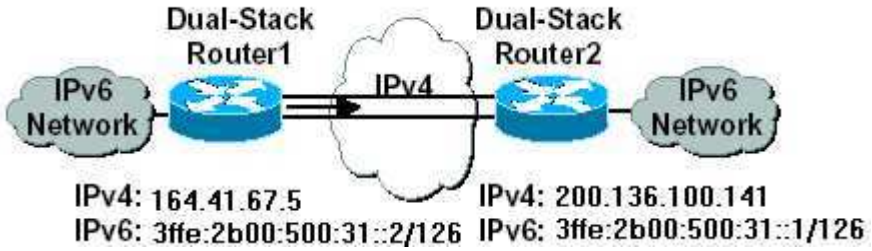


Figura 4-1 Túnel RedUnB

Túnel Freenet6

O Freenet6 TSP (*Tunnel Setup Protocol*) é uma iniciativa realizada pela Viagénie, uma companhia privada do Canadá envolvida com o IPv6 desde 1996, a fim de acelerar a transição para o IPv6. Sendo a Internet um “mundo sobre IPv4”, esse projeto tem como fundamento promover o crescimento do IPv6 utilizando túneis configuráveis.

Dessa maneira qualquer host conectado à Internet por meio do IPv4 contendo uma pilha IPv6 pode estabelecer um link à Internet IPv6.

O protocolo é usado para requisitar um único endereço IPv6 através do cliente que utiliza um servidor de túneis conforme o modelo *tunnel-broker* do IPv6.

Afim de utilizar o Freenet6, existem alguns requisitos básicos que devem ser verificados pelos hosts e redes:

Requesitos de Host:

- Pilha IPv6 instalada
- Endereço IPv4 válido
- Privilégios de usuário *root*

Requisitos de Rede:

- Mecanismo de túnel configurado
- Protocolo TSP, que usa TCP na porta de número 4343 (não definida pelo IANA)
- Firewall – deve ter regras especiais que permitam o protocolo IPv6 (número 41) e acesso à porta TCP 4343
- Roteador – se estiver usando listas de acesso também deve permitir o protocolo IPv6 e a porta 4343.
- Network Address Translation (NAT) – Caso um nó final esteja atrás de um gateway NAT, não será possível o tráfego IPv6 sobre IPv4 de qualquer servidor de túnel, exceto em 2 situações: 1. Se o gateway NAT manipular endereçamento estático e o administrador da rede mapear um endereço IPv4 válido para o usuário final. 2. Se o gateway NAT estiver rodando sobre qualquer plataforma BSD e o usuário final é quem o gerencia, é possível configurar IPfilter especiais para redirecionar pacotes IPv6 sobre IPv4 a determinado host atrás do NAT.

O uso de contas é obrigatório no Freenet6 em dois casos:

Túnel Autenticado

Provê um endereço IPv6 único e permanente ao nó. Os usuários sempre terão o mesmo endereço IPv6 ainda que seus endereços IPv4 mudem. Qualquer mudança no endereço IPv4 será tratada pelo protocolo TSP com uma conta válida.

Delegação do prefixo /48 IPv6

O Freenet6 tem a capacidade de delegar prefixos /48 a qualquer usuário final para um endereçamento de site.

Procedimentos de Instalação e Compilação do código

Descompactação do pacote em um diretório temporário:

```
#cd /tmp
#tar zxf freenet6-0.xx.tgz

As etapas posteriores a descompactação são:
# cd freenet6-0.xx
# make
# make all target= sistema_operacional

Instalação do Pacote (opcional)
#make install target=sistema_operacional installdir=/seu/diretório/destino

Usado no cliente. (requer privilégios de root/administrador)
# cd /seu/diretório/destino/bin (ou diretório temporário)
#./tspc -vf ./tspc.conf
```

No arquivo TSPC.CONF tivemos que mudar os campos *userid* e *passwd* com um login que a gente requisitou junto ao suporte do Freenet6.

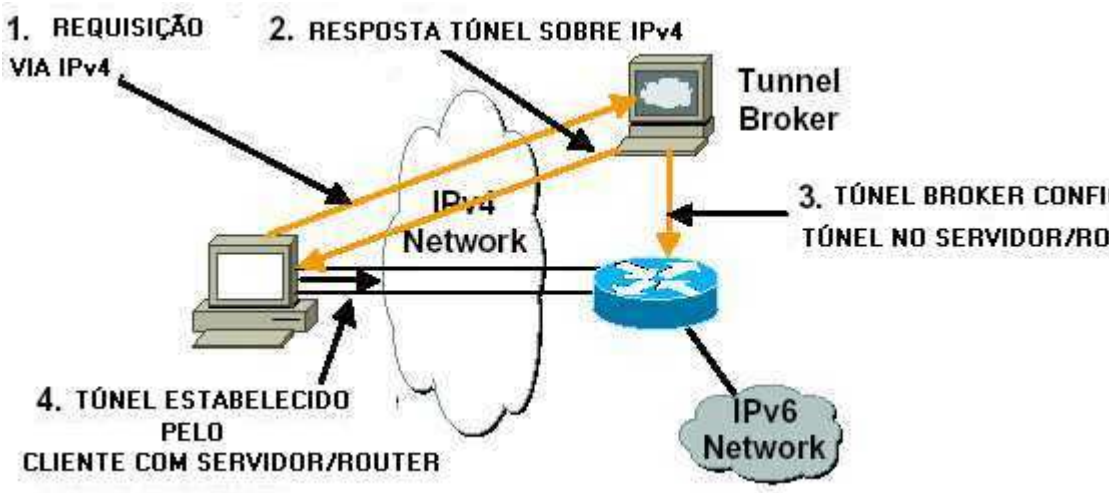


Figura 4-2 Exemplo de configuração utililando *Tunnel Broker*

Para realizarmos um teste de acessibilidade somente-IPv6, instalamos o Apache 2.0.40, fizemos as mudanças pertinentes em algumas diretivas para o suporte ao IPv6 e disponibilizamos uma página de teste através do endereço http://www.6bone.unb.br . Nessa página o usuário navegará em ambiente puramente IPv6. Essa iniciativa foi pioneira no que diz respeito a uma página de teste puramente IPv6 em idioma português.

5. IPV6 E O WINDOWS

A Microsoft também está desenvolvendo suporte ao IPv6 em seus sistemas operacionais Windows NT, 2000 e XP, mas por enquanto o próprio usuário deve baixar um *driver* com as atualizações. A solução Microsoft é chamada MSR IPv6. Fazendo, de forma suscinta uma descrição do suporte dessa implementação podemos relatar suporte a:

1. Processamento básico do cabeçalho IPv6;
2. Cabeçalhos Hop-By-Hop e Destination Options;
3. Cabeçalho de Fragmentação;
4. Cabeçalho de Roteamento;
5. Neighbor Discovery;
6. Autoconfiguração de endereços Stateless;
7. ICMPv6;
8. Multicast Listener Discovery;
9. Meios Ethernet e FDDI;
10. Túneis Automáticos e configurados;
11. IPv6 over IPv4;
12. 6to4;
13. UDP e TCP over IPv6;

14. Funcionalidade de Host e Router;
15. Autenticação IPsec.

Para o laboratório NT do LabRedes fizemos o *download* do programa e prosseguimos com a instalação da seguinte forma.

Para fazer funcionar o IPv6 basta acrescentar o novo protocolo na guia "Protocolos" dentro das propriedades de Rede, como ilustra a figura 5-1.

Nem tudo é simples como no IPv4, para o IPv6 as configurações devem ser todas passadas por linha de comando. Como habilitamos no nosso roteador o *daemon* RADVD que faz anúncios das rotas, todos os hosts da rede se autoconfiguraram o que facilitou muito o nosso trabalho. A tabela seguinte descreve os endereços obtidos por meio do RADVD.

Tabela 12 – Mapa de endereçamento Lab_NT

Nome da Máquina	Endereço IPv6 gerado
LAB_NT 01	3ffe:2b00:100:10b:280:5fff:fe31:9939
LAB_NT 02	3ffe:2b00:100:10b:280:5fff:fe31:992d
LAB_NT 03	3ffe:2b00:100:10b:280:5fff:fe31:9954
LAB_NT 04	3ffe:2b00:100:10b:280:5fff:fe31:9958
LAB_NT 05	3ffe:2b00:100:10b:280:5fff:fe31:99d7
LAB_NT 06	3ffe:2b00:100:10b:280:5fff:fe31:9932
LAB_NT 07	com defeito
LAB_NT 08	3ffe:2b00:100:10b:280:5fff:fe31:9979
LAB_NT 09	3ffe:2b00:100:10b:280:5fff:fe31:d9b0
LAB_NT 10	3ffe:2b00:100:10b:280:5fff:fe31:198d
LAB_NT 11	com defeito
LAB_NT 12	3ffe:2b00:100:10b:280:5fff:fe31:5920
LAB_NT 13	3ffe:2b00:100:10b:280:5fff:fe31:d974
LAB_NT 14	3ffe:2b00:100:10b:280:5fff:fe31:9977
LAB_NT 15	3ffe:2b00:100:10b:280:5fff:fe31:d9a4
LAB_NT 16	3ffe:2b00:100:10b:280:5fff:fe31:592f
LAB_NT 17	3ffe:2b00:100:10b:280:5fff:fe31:990f
LAB_NT 18	3ffe:2b00:100:10b:280:5fff:fe31:9933
LAB_NT 19	3ffe:2b00:100:10b:280:5fff:fe31:1943
LAB_NT 20	3ffe:2b00:100:10b:280:5fff:fe31:d9b4

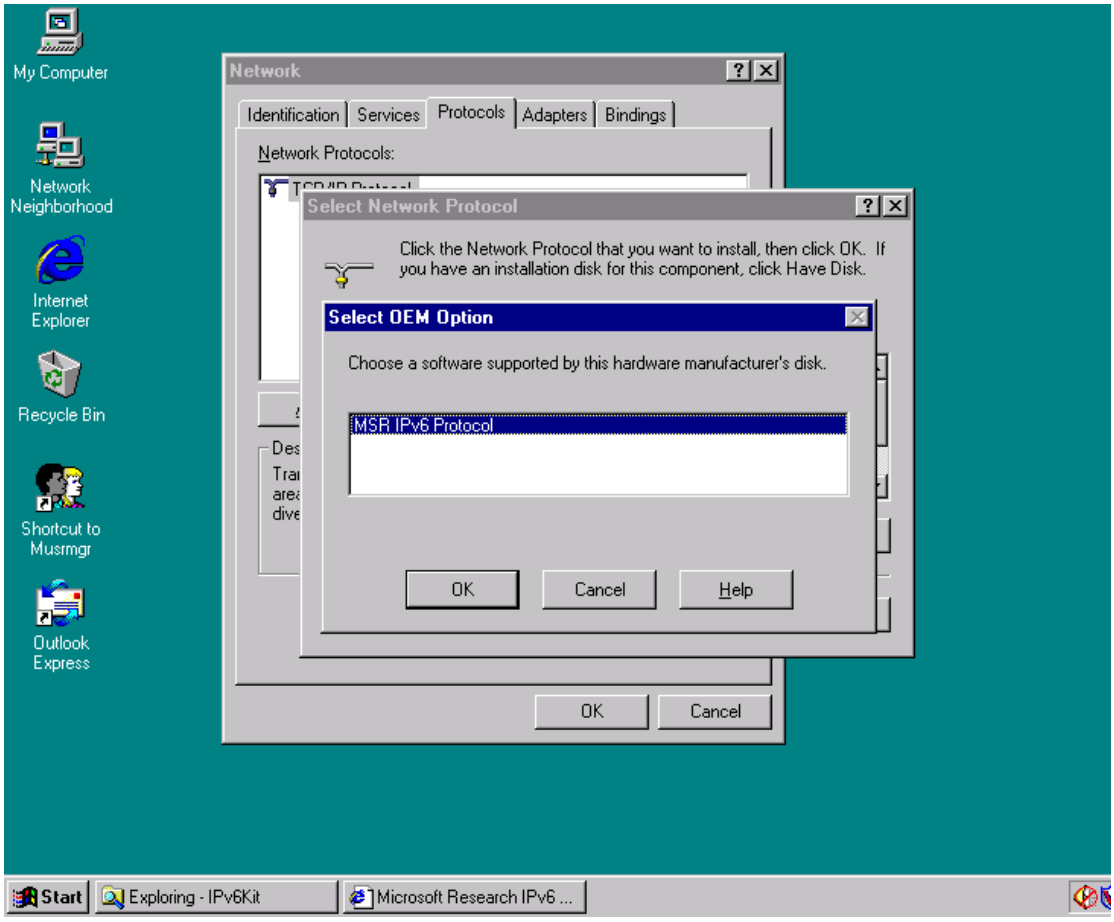


Figura 5-1 Instalação do MSR IPv6

A título de exemplo seguem abaixo alguns dos comandos básicos da configuração IPv6 no Windows:

net start tcpip6

No Windows as mudanças feitas na configuração do IPv6 não são permanentes e são perdidas quando a máquina é desligada ou quando a própria pilha IPv6 é reiniciada. Esse comando para inicializar a pilha IPv6 e a opção *stop* também é utilizada.

ipv6 if

Comando que mostra informações sobre as interfaces.

ipv6 adu if#/address [lifetime VL[/PL]] [anycast] [unicast]

Comando para atribuir ou apagar endereços IPv6 das respectivas interfaces.

Ex.: ipv6 adu 3/3ffe:2b00:0100:010b::11

ipv6 rt

Mostra as configurações da tabela de roteamento.

Os utilitários ping6, tracert6,... podem ser utilizados na solução de problemas e testes.

Com relação à navegação, constatamos que o Internet Explorer 6, utilizado em todo o laboratório NT, não possui suporte ao IPv6. Outros browser tais como o Mozilla 1.0, e o Netscape 6.2 para Windows foram utilizados na tentativa de encontrar um navegador com suporte HTTP IPv6. Sendo assim fizemos um *downgrade* do software IE 6 para a versão 5.5 e modificamos o arquivo Wininetd.dll

pelo gerado no “MSR IPv6 protocol”. Essa alteração funcionou perfeitamente.

Toolnet6

Uma outra implementação para Windows diz respeito ao Toolnet6. O Toolnet6 é uma implementação de DNS-NAT (Network Address Translator) que provê conectividade IPv6 no Windows. Essa implementação permite acesso a redes tanto IPv6 quanto IPv4 em ambientes Microsoft Windows NT, fazendo com que a transição para o novo protocolo seja feita suavemente.

Fizemos um teste com esse programa, mas enfrentamos dificuldades em colocá-lo funcionando corretamente. Consideramos uma implementação muito específica e que funciona com algumas placas de rede da 3COM.

Procedimentos de instalação

Dividimos o processo de instalação realizado a fim de facilitar e detalhar cada procedimento realizado nesse teste.

. criação de disquete com drivers de instalação;

. instação de placa e driver ethernet;. configuração dos parâmetros toolnet6;. configuração tabela NAT;. configuração rede TCP/IP;. restart da máquina;

Criação de disquete com drivers de instalação O download do arquivo referente ao Windows NT (nt-e.exe) pode ser feito na página da Hitachi. Existem vários tipos de arquivos cabendo à cada um a análise do sistema a ser utilizado. Após a descompactação do mesmo recomenda-se a gravação dos arquivos gerados em um disco flexível para uso durante a instalação e configuração. Instação de placa Ethernet e drivers Com relação aos procedimentos de instalação e configuração da placa Ethernet, não se tem nenhuma restrição ou recomendação. Esse processo pode ser realizado normalmente.

Passando para a instalação dos drivers deve-se selecionar "Meu Computador" → "Painel de Controle" → "Rede".

Posteriormente na opção "Adaptadores" seleciona-se a alternativa "Adicionar...". Em seguida, a caixa "Selecione Adaptador de Rede" aparecerá. Basta então selecionar "Com disco...". Com a opção "Insira o Disco”, inserir o disco criado na etapa anterior e direcionar o caminho para o mesmo. Com a conclusão desses passos a caixa de opção *Select OEM Option* aparecerá conforme ilustra a figura 5-2. Confira as seguintes opções de drivers existentes:

"3Com EtherLink 3 3C509 Adapter(IPv6)" → para 3C509/ISA NIC

"3Com EtherLink 3 3C589 Adapter(IPv6)" → para3C589,3C589A/B/C PCMCIA Card

"3Com EtherLink 3 3C589D Adapter(IPv6)" → para 3C589D PCMCIA Card

E selecione a opção desejada.

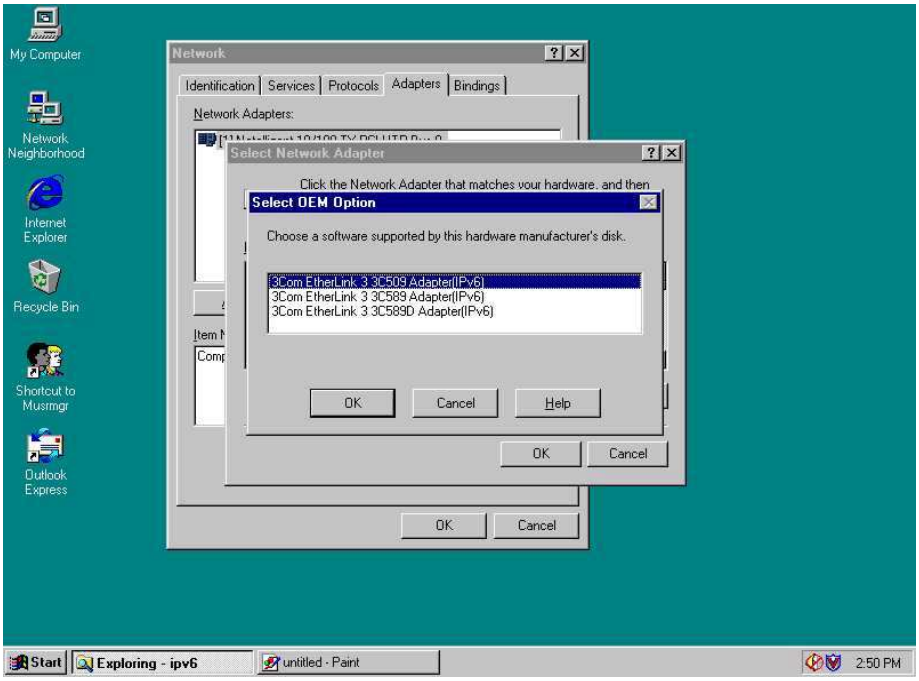


Figura 5-2 Seleção do driver Toolnet6

Selecionado o *driver* desejado, a caixa chamada "3Com EtherLink3 3C509 Adapter Card Setup(IPv6)" surgirá, como na próxima figura:

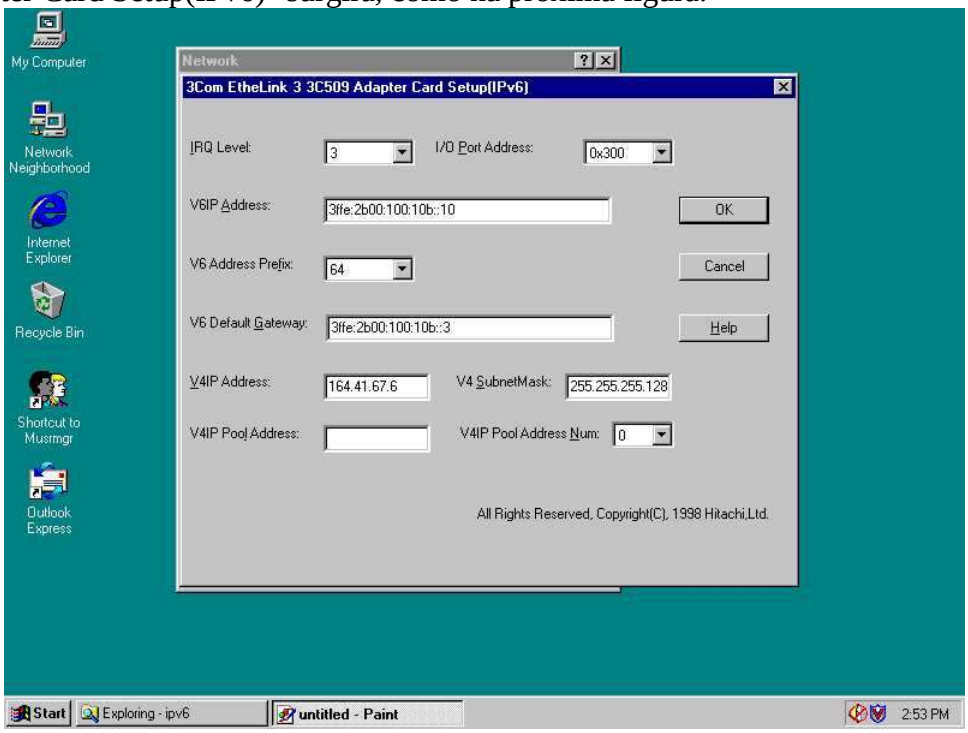


Figura 5-3 Configuração do túnel usando Toolnet6

Configuração dos parâmetros toolnet6

Nessa caixa de diálogo, nove parâmetros surgem para serem configurados:

(a) IRQ level – requerida por hardware (decimal). (b) Endereço de porta I/O - (hexadecimal).

(c) Endereço IPv6. (d) Tamanho do Prefixo IPv6(decimal). (e) Endereço IPv6 do Default Gateway.

(f) Endereço IPv4. (g) Máscara Subrede IPv4.

(h) Especificar o endereço IPv4 de início do *pool*. (i) O número de endereços do *pool*. Assim que concluída essa etapa, caso o sistema pergunte o tipo de placa Ethernet basta escolher dentro das alternativas. Figura 5-4.

3C509/ISA NIC → "ISA" # 3C589/PCMCIA Card → "PCMCIA"

Assim que a instalação estiver completa, o gerenciador NAT será automaticamente iniciado para que se possa configurar a tabela.

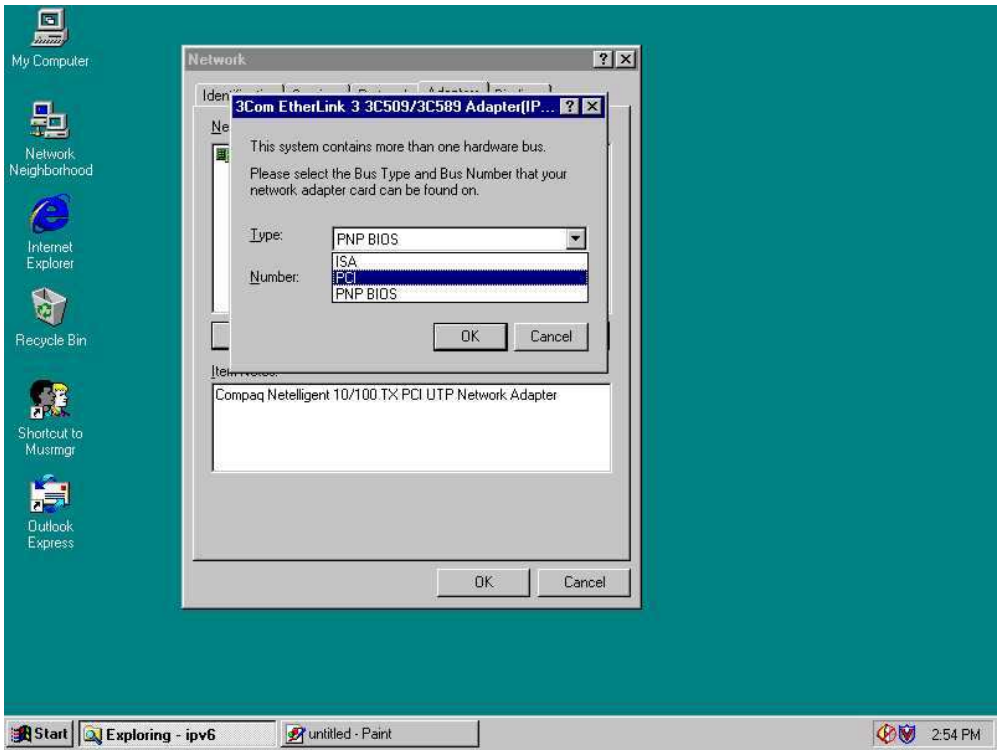


Figura 5-4 Tipo de placa implementação Toolnet6

Configuração da tabela NAT Faz o mapeamento de registros entre endereços IPv4 e IPv6. Finalizado o preenchimento do gerenciador NAT deve-se partir para a configuração das propriedades TCP/IP.

Configuração das propriedades TCP/IP Na janela "Microsoft TCP/IP" Properties, deve-se configurar três parâmetros: endereço IP, máscara de subrede e Default Gateway. Deve-se tomar o cuidado de utilizar os mesmos valores inseridos anteriormente. Por fim, basta reiniciar a máquina.

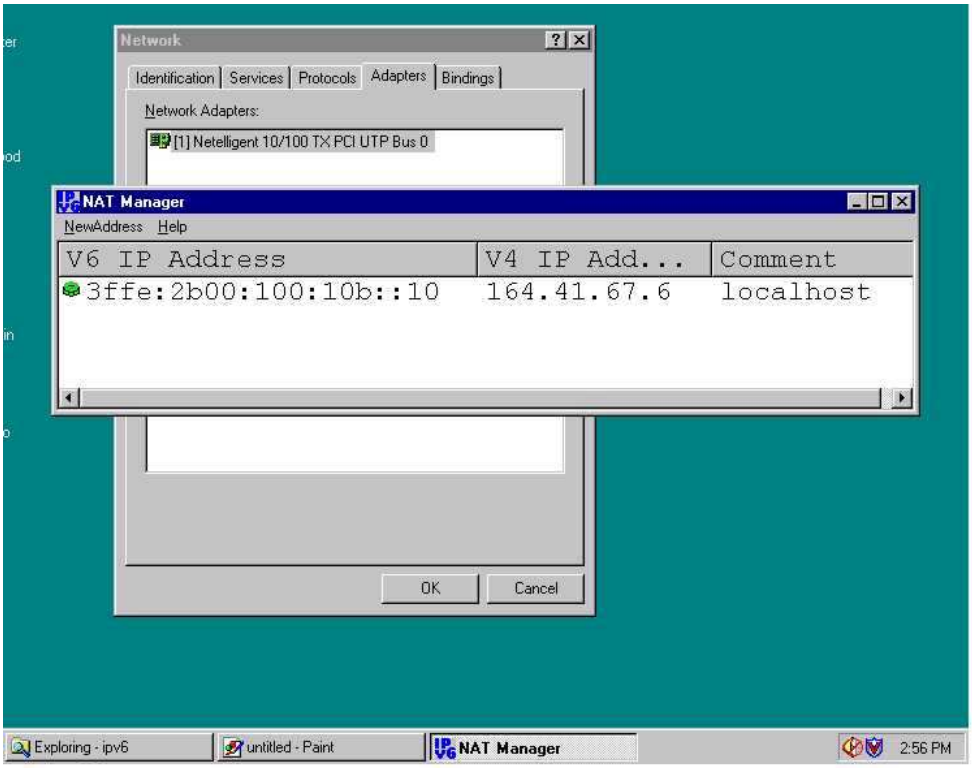


Figura 5-5 Configuração gerenciador NAT

6. PROJETOS ESPECIAIS

Em nossas pesquisas encontramos várias referências a projetos especiais envolvendo o IPv6.

O projeto KAME é desenvolvido por seis empresas no Japão, a saber (em ordem alfabética),

Fujitsu Limited
Hitachi, Ltd.
Internet Initiative Japan Inc.
NEC Corporation
Toshiba Corporation
Yokogawa Electric Corporation

Sua finalidade é desenvolver uma pilha de *softwares* gratuitos com ênfase em IPv6/ IPSec para as diversas variantes BSD, através de um esforço conjunto dos pesquisadores das empresas participantes. O projeto começou em abril de 1998 com a previsão de durar apenas dois anos, mas foi estendido até março de 2004.

O projeto USAGI (Universal Playground for IPv6) é composto por voluntários, na maioria japoneses, e tem a finalidade de desenvolver uma pilha IPv6 de qualidade para o Linux.

O projeto TAHI também é desenvolvido no Japão e sua finalidade é desenvolver uma metodologia de testes para a tecnologia IPv6 trabalhando conjuntamente com os projetos KAME e USAGI.

7. CONCLUSÃO

A entrada da rede da UnB – RedUnB no 6Bone, tem grande importância para o meio acadêmico em geral. Por meio dela, a Universidade cumpre o seu papel para com a sociedade passando a contribuir para a fase de testes de campo, o desenvolvimento e o aperfeiçoamento do protocolo IPv6 no Brasil e no mundo.

A UnB, por meio desse projeto, passa a fazer parte de uma rede virtual existente sobre a rede física IPv4 da atual Internet. Essa rede virtual é composta de redes locais IPv6 ligadas entre si por túneis ponto-a-ponto IPv6 sobre IPv4. Os túneis são formados por roteadores com pilha dupla (IPv6 e IPv4) com suporte para roteamento estático e dinâmico (RIPng e BGP4+), e as redes locais IPv6 são compostas por estações com sistemas operacionais com suporte a IPv6 ou com pilha dupla.

Na busca de uma melhor configuração para o kernel das máquinas Linux do laboratório, constatou-se que, em termos de flexibilidade e ferramentas, o Linux supera consideravelmente o Windows, que foi o outro sistema operacional pesquisado. Tivemos alguns problemas/dificuldades no processo de instalação e configuração do sistema nas máquinas, o que nos fez ganhar experiência em *troubleshooting*.

A solução para os sistemas Microsoft possui a vantagem da facilidade de instalação e manutenção, características registradas da empresa. Por outro lado, essa solução não tem a robustez daquela utilizada no Linux sendo adotada apenas nas máquinas clientes durante o projeto.

Com relação à configuração do DNS para suporte IPv6, verificamos que apenas a inclusão dos registros do tipo AAAA no próprio arquivo de zonas existente em um domínio é suficiente. A essência do sistema e da lógica funcional não foi alterada.

Os testes de algumas implementações para IPv6 efetuados se mostraram confiáveis e condizentes com a teoria confrontada. Apenas algumas limitações no sentido da falta de suporte por conta de alguns aplicativos foram encontradas.

Ainda no intuito de avanço nas pesquisas do protocolo IPv6, descrevemos um conjunto de propostas para futuros projetos e estudos:

1. Testes em conexões multihomed;
2. Roteamento com BGP4+, RIPng e IGRPng;
3. Aplicações multicasting;
4. Aplicações multimídia;
5. Tecnologia e aplicações anycast;
6. NAT (Network Address Translation) de IPv6 para IPv4 e vice-versa;
7. DHCPv6;
8. IPSec.

Contrariando até algumas estimativas de que os endereços IPv4 se esgotariam em 1999, o uso da estratégia CIDR (Classless Interdomain Routing) está fornecendo uma maior sobrevida ao espaço de endereçamento IPv4. Mas, com o vertiginoso

crescimento da Internet no cenário mundial e com o desenvolvimento das tecnologias e aplicações IPv6, cada vez fica mais evidente que a migração para IPv6 é inevitável.

Todavia, porque o IPv6 ainda não é utilizado em grande escala?

Dentre os pontos que são de maior consenso na comunidade IPv6 podemos destacar:

1. A crise financeira enfrentada pelas companhias;
2. O pouco ou falta de suporte com relação ao Sistema Operacional (exemplo: vanilla 2.4 Linux Kernel);
3. Falta de suporte em aplicações;
4. Falta de suporte a *firewalls* comerciais.

Há até mesmo aqueles que dizem que não existe uma necessidade mundial real para o uso de endereços, apenas nos casos de 3G (mobile phones) ou comunicação em automóveis.

Contudo, a UnB estará preparada para qualquer transição que venha surgir nesse âmbito...

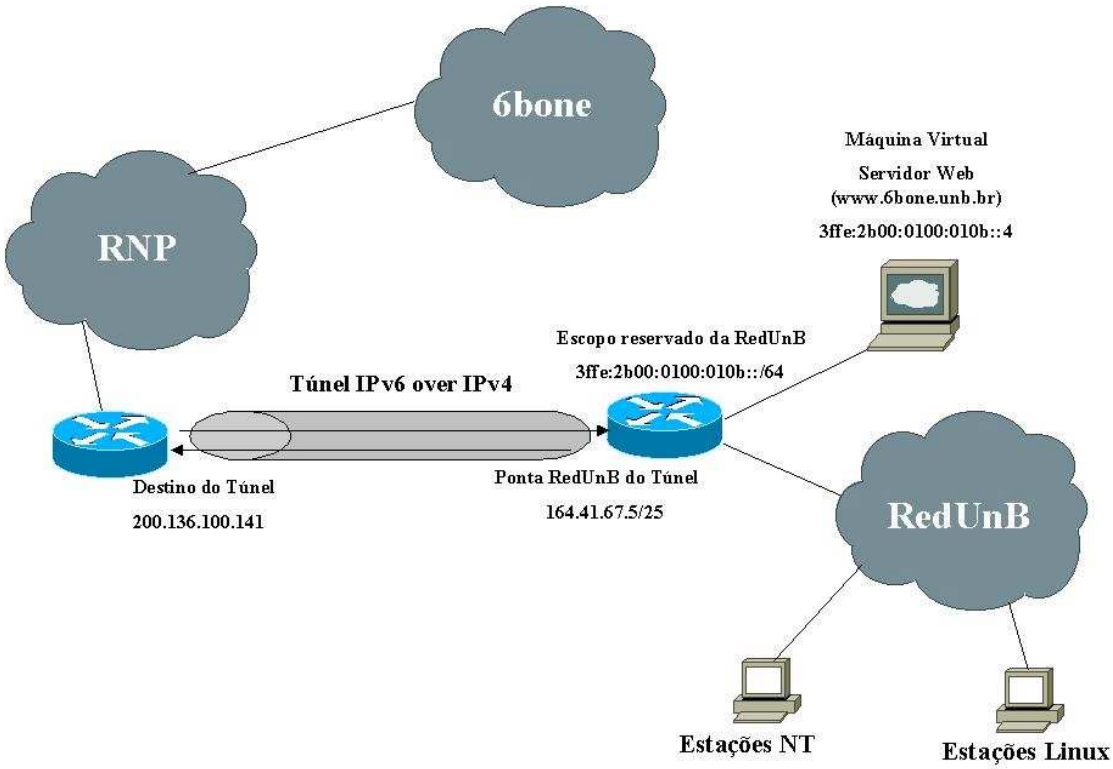


Figura 7-1 Interligação da RedUnB ao Br6Bone

"Os peritos não têm dúvidas em rotulá-la como a migração da década. E assim será, pois a transição mundial para IPv6 demorará cerca de dez anos até ser concluída."

8. BIBLIOGRAFIA

[1] COMER, D.E. *Internetwoking with TCP/IP*, 4th ed., Prentice Hall, 2000, cap 24 e 33.

[2] STALLINGS, W. *Data and Computer Commun*ications. 6th ed., Prentice Hall,1999, cap.15

[3] ALBITZ, Paul e LIU, Cricket. *DNS and BIND*, 2nd ed., O'Reilly & Associates, Inc.

[4] IPv6 Org - Site < http://www.IPv6.org> Último acesso em: 18 de ago. 2002

[5] IPv6 Resource Centre - Site < http://www.cs-IPv6.lancs.ac.uk> Último acesso em: 01 de ago. 2002

[6] IPv8 (Galaxy/StarGate) Reference Guide - Site < http://ipv8.vrx.net/> Último acesso em: 12 de ago. 2002

[7] Linux IPv6 – Site < http://www.bieringer.de/linux/IPv6/index.html> Último acesso em: 18 de ago. 2002

[8] Research Microsoft - Site < http://www.research.microsoft.com/msripv6/> Último acesso em: 18 de ago. 2002

[9] IPv6 - Site < http://www.IPv6.mfa.eti.br/> Último acesso em: 10 de ago. 2002

[10] Overview Usagi Project - Site < http://www2.linux-IPv6.org/~sekiya/IETF49/> Último acesso em: 22 de jul. 2002

[11] Site < http://ebs.jindai.net/v6rpm.html> Último acesso em: 15 de jul. 2002

[12] Kame Project - Site < http://www.kame.net> Último acesso em: 18 de ago. 2002

[13] IPv6 Forum - Site < http://www.ipv6forum.com/> Último acesso em: 05 de ago. 2002

[14] Hitachi Project - Site < http://www.hitachi.co.jp> Último acesso em: 02 de ago. 2002

[15] Site < http://playground.sun.com> Último acesso em: 18 de ago. 2002

[16] IP Version 6 Working Group (IPv6) - Site < http://www.ietf.org/html.charters/IPv6-charter.html> Último acesso em: 04 de ago. 2002

[17] IP Version 6 (IPv6) - Site < http://playground.sun.com/pub/ipng/html > Último acesso em: 18 de ago. 2002

[18] WIDE v6 working group - Site < http://www.v6.wide.ad.jp/> Último acesso em: 20 de jul. 2002

[19] Freenet6 - Site < http://www.freenet6.net/> Último acesso em: 18 de ago. 2002

[20] ESnet IPv6 - Site < http://www-6bone.es.net/> Último acesso em: 07 de ago. 2002

[21] 6bone RNP - Site < http://www.6bone.rnp.br> Último acesso em: 18 de ago. 2002

[22] IPv6 Research and Education Networks - Site < http://www.6ren.net/> Último acesso em: 18 de ago. 2002

[23] Ca*Net3 IPv6 - Site < http://www.6pop.canet2.net/> Último acesso em: 07 de ago. 2002

[24] WIDE Project - Site < http://www.wide.ad.jp/> Último acesso em: 18 de ago. 2002

[25] TAHI Project - Site < http://www.tahi.org/> Último acesso em: 18 de ago. 2002

[26] NetBSD IPv6 Networking - Site < http://bofh.st/IPv6/apps.shtml> Último acesso em: 19 de jul. 2002

[27] ISC. *ISC BIND*. Site < www.isc.org/isc/bind.html > Último acesso em: 08 de ago 2002.

[28] IETF RFC 1809 Site < http://www.ietf.org/rfc/rfc1809.txt?number=1809 > Último acesso em: 18 de ago 2002.
[29] IETF RFC 1881 Site < http://www.ietf.org/rfc/rfc1881.txt?number=1881 > Último acesso em: 18 de ago 2002.
[29] IETF RFC 1933 Site < http://www.ietf.org/rfc/rfc1933.txt?number=1933 > Último acesso em: 18 de ago 2002.
[30] IETF RFC 2373 Site < http://www.ietf.org/rfc/rfc2373.txt?number=2373 > Último acesso em: 18 de ago 2002.
[31] IETF RFC 2460 Site < http://www.ietf.org/rfc/rfc2460.txt?number=2460 > Último acesso em: 18 de ago 2002.
[32] IETF RFC 2461 Site < http://www.ietf.org/rfc/rfc2461.txt?number=2461 > Último acesso em: 18 de ago 2002.
[33] IETF RFC 2462 Site < http://www.ietf.org/rfc/rfc2462.txt?number=2462 > Último acesso em: 18 de ago 2002.
[34] IETF RFC 2463 Site < http://www.ietf.org/rfc/rfc2463.txt?number=2463 > Último acesso em: 18 de ago 2002.
[35] IETF RFC 2464 Site < http://www.ietf.org/rfc/rfc2464.txt?number=2464 > Último acesso em: 18 de ago 2002.
[36] IETF RFC 2471 Site < http://www.ietf.org/rfc/rfc2471.txt?number=2471 > Último acesso em: 18 de ago 2002.
[37] IETF RFC 2473 Site < http://www.ietf.org/rfc/rfc2473.txt?number=2473 > Último acesso em: 18 de ago 2002.
[38] IETF RFC 3068 Site < http://www.ietf.org/rfc/rfc3068.txt?number=3068 > Último acesso em: 18 de ago 2002.

9. ANEXOS

9.1. ANEXO A

Termo de Adesão

Brasília – DF, 13/07/2002

À Rede Nacional de Pesquisa
A/C Marcel R. Faria
Centro de Engenharia e Operações - CEO/RNP
marcel@rnp.br
Fax: (19) 3787-3301

De: Sr(a). Rafael Timóteo de Sousa Júnior
Universidade de Brasília-UnB/Dep. Eng. Elétrica – Laboratório de Redes
Avenida L3 Norte, Faculdade de Tecnologia
Dep. Eng. Elétrica – LabRedes – Sala B1-01
Asa Norte
(61) 307-2308 R.:238
desousa@redes.unb.br

Prezado Senhor,

Solicitamos da Rede Nacional de Pesquisa a alocação de um endereço IPv6 para uso em projeto de pesquisa e participação no Backbone IPv6 Brasileiro.

Desde já nos comprometemos a seguir as recomendações e exigências do Grupo de Trabalho NGTRANS (Next Generation Transition) da IETF (http://www.ietf.org/html.charters/ngtrans-charter.html), do 6Bone (http://www.6bone.net), inclusive no que se refere às especificações do RFC 2772 - "6Bone Routing Practice" (http://www.ietf.rnp.br/ftp/rfc/rfc2772.txt) - e outras recomendações que por ventura sejam definidas pela Rede Nacional de Pesquisa ou pela IETF. Também nos comprometermos a disponibilizar informações técnicas de pesquisas e testes relevantes para os projetos envolvidos.

Atenciosamente,
Rafael Timóteo de Sousa Júnior
Coordenador do curso de Engenharia de Redes de Comunicação
Universidade de Brasília-UnB

9.2. ANEXO B

```
FORMULARIO DE SOLICITACAO DE NUMERO IPv6 RNP (20011120)
=======================================================

# Instrucoes gerais:
# . Enviar o formulario preenchido para IPv6-adm@rnp.br.
# . Nao utilize caracteres acentuados ou cedilha.
# . Todas as linhas iniciadas com um "#", como a presente, sao
#   comentarios e podem ser removidas antes do envio do
#   formulario.

1. Nome da organizacao que solicita a rede
------------------------------------------

organization.........: Universidade de Brasilia - UnB
departament/unit.....: Dep. Engenharia Eletrica - Laboratorio de
Engenharia de Redes
address..............: Avenida L3 Norte, Faculdade de Tecnologia
address..............: Dep. Eng. Eletrica - LabRedes - Sala B1-01
city.................: Brasilia
state/province.......: DF
postal Code..........: 70910-900
country..............: Brasil


2. Caracteristicas da organizacao e da utilizacao da rede
---------------------------------------------------------

# ES - ensino superior, graduacao ou pos-graduacao

type1: ES


# Caso tenha selecionado EB, ES, GO ou IP, escolha tambem  uma das
# opcoes abaixo. Caso contrario, deixe em branco ou remova a linha
# "type2".

# PR - entidade privada
# FE - subordinada ao governo federal
# ET - subordinada a governo estadual
# MU - subordinada a governo municipal
```

```
type2: FE


3. Contato administrativo
-------------------------

# IMPORTANTE: o contato administrativo deve necessariamente ser uma
# pessoa vinculada a organizacao que solicita a rede. O endereco
postal
# e telefone dessa pessoa serao interpretados como sendo os da
propria
# organizacao que solicita a rede.

# Caso a pessoa ja esteja cadastrada na base de dados da RNP, coloque
# aqui apenas o seu "nic-handle" e deixe os outros campos dessa
# secao em branco.

nic-hdl:

# Caso nao haja cadastramento previo, forneca as informacoes abaixo.
# Como parte do processo de registro, lhe sera devolvido um nic-
handle
# (identificador unico) para a pessoa especificada. Veja exemplo:

person: Rafael Timoteo de Sousa Junior
address: Avenida L3 Norte, Faculdade de Tecnologia
address: Dep. Eng. Eletrica - LabRedes - Sala B1-01
address: Asa Norte
address:
phone: +55 61 307-2308 R.:238
fax-no: +55 61 274-6651
e-mail: desousa@unb.br
notify: desousa@unb.br



4. Contato tecnico
------------------

# Caso o contato tecnico e administrativo sejam a mesma pessoa,
# deixar os campos abaixo em branco.
#
# A pessoa de contato tecnico pode nao ser formalmente vinculada
# a organizacao que solicita a rede, podendo ter endereco, telefone e
# fax diferentes, e mesmo trabalhar regularmente em outra
organizacao.
#
# Caso a pessoa ja esteja cadastrada na base de dados da RNP, coloque
# aqui apenas o seu "nic-handle" e deixe os outros campos dessa
# secao em branco.

nic-hdl:

person: Robson de Oliveira Albuquerque
address: Avenida L3 Norte, Faculdade de Tecnologia
address: Dep. Eng. Eletrica - LabRedes - Sala B1-01
address: Asa Norte
address:
phone: +55 61 307-2308 R.:238
fax-no: +55 61 274-6651
e-mail: robson@redes.unb.br
notify: robson@redes.unb.br
organization: UnB - Universidade de Brasilia
```

5. Nome da rede

Nome da rede. Nao deve conter, no maximo, 14 caracteres
alfanumericos. Veja exemplo:

netname: RedUnB

6. Breve descricao da rede IPv6

Campo obrigatorio. Necessario para avaliacao do
pedido. Formato livre.

O Laboratorio de Engenharia de Redes de Comunicacao da UnB participa da Infra-estrutura Internet2 para o Brasil (RNP2), sendo o nodo central da REMAV Projeto Infovia de Brasília (http://www.rnp.br/remav/consorcios/brasilia.html), cuja coordenacao encontra-se sob responsabilidade do Prof. Rafael Timoteo de Sousa Jr. O LabRedes localiza-se em uma area de cerca de 500m2, contendo cerca de 80 computadores clientes, 10 computadores servidores, 1 rede local com cabeamento estruturado em UTP cat.5 e fibras ppticas, com 5 concentradores ATM a 155Mbps. Para integrar o projeto 6Bone da RNP, o LabRedes tem a intencao de disponibilizar aproximadamente 20 maquinas baseadas em Linux e com suporte IPv6. Sera utilizada uma maquina (164.41.67.5) como tunnel endpoint no link UNB-RNP, que atualmente é composto de um canal a 2Mbps e outro a 34Mbps.

7. Espaco solicitado

Informar o tamanho do prefixo IPv6 solicitado:
NLA (/48), SLA (/56) ou Leaf Site (/64).
#
Alocacoes inicias costumam ser de /64. Veja exemplo:
#

Length of IPv6 prefix......: /64

Justifique, em formato livre, a quantidade de espaco de
enderecamento solicitado. O correto preenchimento
desse item eh fundamental no julgamento da solicitacao.

Sendo esse, um projeto inicial, achamos por bem utilizar o tamanho descrito.

8. Dados do Servidor de Nomes

Dados dos servidores de nomes. Deve ser fornecido pelo
menos um hostname. Veja exemplo:

Primary IP6.INT Server Hostname......: dns.redes.unb.br
Secondary IP6.INT Server Hostname....:

9. IP da interface de Tunel

Endereco IP da interface de tunel. Exemplo:
#

tunnel endpoint: 164.41.67.5

10. Outras informacoes

Coloque aqui informacoes que lhe parecam importantes a respeito
da organizacao ou experimentos que deseje conduzir.
Este campo eh opcional e em formato livre.

A UnB e uma das principais instituicoes federais de ensino e pesquisa. O Departamento de Engenharia Eletrica da UnB oferta 3 cursos de graduacao, entre os quais o o Curso de Engenharia de Redes de Comunicacao. O Departamento tem ainda um programa de pos-graduacao com mestrado acadêmico, mestrado profissionalizante, doutorado e 6 especializacoes nas areas de tecnologias da informacao e das comunicacoes. Tal programa e avaliado com nota 4 pela CAPES.
Em funcao da participacao do LabRedes no projeto da REMAV local, objetiva-se dar continuidade a experimentos em realizacao no contexto da REMAV, especificamente: 1.Experimento Bibliotecas Virtuais Colaborativas; 2.Experimento Disponibilizacao de Informacoes Governamentais Georeferenciadas; 3.Experimento Disponibilizacao de Informacoes sobre Recursos Geneticos e Controle de Pragas; 4.Experimento Educacao via Rede Multimídia; 5.Experimento Gestao Tecnologica e Comercial de uma Rede de Alta Velocidade e seus Servicos; 6.Experimento Telemedicina - Prontuario Eletronico do Paciente. Com tais experimentos, ha possibilidades de estudos acerca de computacao distribuída, QoS, engenharia de trafego, agentes moveis e balanceamento de carga.

-------------------------<corte aqui>--------------------------------

GUIA PARA PREENCHIMENTO

```
**********************************************************************
*                                                                    *
* Duvidas, criticas ou sugestoes, relativas a esse formulario,       *
* favor endereca-las a                                               *
*                                                                    *
*                    IPv6-adm@rnp.br                                 *
*                                                                    *
* Serao bem vindas.                                                  *
*                                                                    *
**********************************************************************
```

Os campos telefone e e-mail sao indispensaveis.

O CONTATO ADMINISTRATIVO TEM QUE SER OBRIGATORIAMENTE UMA PESSOA VINCULADA A ORGANIZACAO QUE SOLICITA A REDE, PESSOA ESSA QUE SEJA RESPONSAVEL PELA CONECTIVIDADE A INTERNET DE SUA ORGANIZACAO.
OS DADOS DE ENDERECO E TELEFONE DO ADMIN-C SERAO CONSIDERADOS COMO SENDO OS DA ORGANIZACAO.

Tipicamente, o contato-admin sera o gerente de processamento de dados ou de redes, ou o chefe do CPD da organizacao solicitante.

Tipicamente, o contato-tecnico sera o tecnico diretamente envolvido com o gerenciamento da rede da organizacao e sua conexao a Internet. Essa pessoa sera procurada sempre que ocorram problemas ou duvidas relacionadas ao funcionamento adequado da rede.

Por imediato entende-se um periodo de ate 60 dias apos a entrada em operacao de rede.

---- **** ----

9.3. ANEXO C

Para mantermo-nos fixos no objetivo do projeto e aproveitar melhor o tempo que tínhamos disponível fizemos um diário de projeto documentando todas as configurações realizadas durante a nossa trajetória.

Dia	22/05/2002
Assunto	Contato RNP

- Após a definição do conteúdo e da metodologia de um projeto final de graduação, juntamente com a professora orientadora Cláudia Jacy Barenco, iniciamos o processo de coleta de informações sobre os contatos na RNP. Inicialmente solicitamos o auxílio de Marcelino Cunha – RNP/DF [marcelino@na-df.rnp.br] que nos repassou o nome do responsável pelo projeto: Marcel Rodrigues de Farias [marcel@rnp.br].
- Entramos em contato com o responsável descrito solicitando documentação para agilizarmos todo o processo de cadastro e avaliação.

Dia	11 e 12/06/2002
Assunto	Instalação Red Hat 7.2

- Após conseguir a alocação de uma máquina no labredes para o desenvolvimento do projeto, iniciamos a instalação do Red Hat 7.2. A instalação não pode ser concluída devido ao fato de ficarmos recebendo mensagens de erro se referindo ao ANACONDA.

Dia	14/06/2002
Assunto	Instalação Red Hat 7.1

- Reiniciamos a tentativa de instalação agora com o Red Hat 7.1 conseguido com os bolsistas do próprio LabRedes. A instalação foi concluída com sucesso.

Dia	18/06/2002
Assunto	Compilação kernel

- Iniciamos a configuração da máquina para ter suporte IPv6. Fizemos as configurações seguindo a documentação obtida no site do Br6Bone e recompilamos o novo Kernel para suporte ao protocolo IPv6. Foram encontrados alguns erros durante a compilação

make [2]: *** [dummy.o] Error 1
make [2]: Leaving directory ‘/usr/src/linux-2.4.2/drivers.net’
make [1]: *** [-modsubdir-net] Error 2
make [1]: Leaving directory ‘/usr/src/linux-2.4.2/drivers’

make : *** [-mod-drivers] Error 2

Dia	21/06/2002
Assunto	Compilação kernel

- Os erros encontrados anteriormente foram corrigidos e a recompilação do Kernel foi efetivada sem maiores problemas. Realizamos posteriormente a configuração do arquivo /etc/lilo.conf. Após reinicializar a máquina com a nova imagem – linux IPv6, a tela parava na seguinte mensagem "*UNCOMPRESSING KERNEL"*. Concluimos que havíamos nos deparado com um possível conflito de hardware.
- Entramos em contato com Marcel Rodrigues de Faria [marcel@rnp.br] para perguntar sobre a necessidade de interface IPv6 na máquina DNS. A resposta foi que não e foi obtida no dia 15/07/2002.

Dia	25/06/2002
Assunto	Kernel e Placa de rede

- A próxima etapa foi a instalação de uma placa de rede da 3Com modelo 905B na máquina ponta do túnel. Após a instalação, demos *boot* na máquina e duas falhas ocorreram:

* TURNING ON PROCESS ACCOUNTING ACCTON: FUNCTION NOT IMPLEMENTED [FAILED]
* DELAYING ETH0 INICIALIZATION [FAILED]
* DELAYING ETH1 INICIALIZATION [FAILED]

- Foi feita a instalação dos módulos de suporte para as placas 3Com como *built-in* e nova compilação do Kernel. Cabe ressaltar que cada compilação do Kernel gasta por volta de 2 horas.

Dia	26/06/2002
Assunto	Kernel e Placa de rede

- Os erros encontrados no dia anterior ainda permaneciam. Alteramos a instalação dos módulos de suporte para as placas 3Com para o tipo "M" [modules] e nova compilação do Kernel foi feita.

Dia	27/06/2002
Assunto	Kernel e Placa de rede

- Os erros anteriores desapareceram. Fizemos os testes e constatamos a funcionalidade plena da placa [*ping loopback* e endereço IPv4]. Comprovamos a geração do endereço "link-local" IPv6 [***fe80::***210:4bff:fe35:ca36***/10***] através do próprio MAC da máquina [00:10:4b:35:ca:36].

Dia	28/06/2002
Assunto	Kernel e Placa de rede

- Autorizados a utilizar uma porta do switch UnB, efetuamos os testes de conectividade com a Internet, testes que funcionaram perfeitamente.
- Realizamos a inclusão do *alias eth0 3c59x* no arquivo ***/etc/modules.conf***

Dia	02/07/2002
Assunto	Placa de rede e aperfeiçoamentos

- Hoje fizemos a adição de outra placa de rede.
- Posteriormente fizemos testes IPv6 na própria placa para verificar seu funcionamento.
- Iniciamos o aperfeiçoamento da parte gráfica do linux configurando o Vídeo.
- Por fim fizemos download de alguns aplicativos com suporte IPv6 para usarmos no projeto.
- Verificamos o processo de desabilitação de serviços que eram considerados de risco [estudo].

Dia	03/07/2002
Assunto	Netscape 6.2

- Tínhamos como proposta fazer a instalação do Netscape 6.2 baixado da própria página (www.netscape.com). Seguindo as orientações descritas no próprio site, efetuamos a instalação do mesmo sem problemas. Ao tentarmos acessar algumas páginas da internet constatamos erros durante a resolução. Local de instalação: */usr/local/netscape*
- Foram realizadas as alterações no arquivo ***/etc/hosts***.

Dia	04/07/2002
Assunto	*Sound*

- Ainda tínhamos constatado um erro na placa de som quando desligávamos a máquina. Foi diagnosticado que o problema era similar ao da placa de rede, encontrado no dia 25/06/2002. Realizamos a correção seguindo passos parecidos aos anteriormente realizados. Efetuamos a instalação da opção *sound* como módulo.
- Realizamos o contato com Peter Bieringer [pb@bieringer.de] para verificarmos a autenticidade da informação contida em sua página a respeito do suporte Netscape 6.2 ao protocolo IPv6. A resposta obtida foi afirmativa.

Dia	09/07/2002
Assunto	Máquina

- Conseguimos uma nova máquina para efetuar a realização do projeto [Pentium 233MHz, 64M RAM].
- Refizemos as instalações e configurações para suporte IPv6.

Dia	10/07/2002
Assunto	Netscape 6.2

- Refizemos a instalação do Netscape 6.2. Os mesmos problemas de acessibilidade em alguns sites continuaram aparecendo.

Dia	12/07/2002
Assunto	Netscape 6.2 e DNS

- Entramos em contato com Allan Edgard Silva Freitas [allan@cefetba.br] para verificarmos o problema do Netscape 6.2. Ele orientou para que verificássemos a conecitvidade da máquina. Verificamos o alcance de sites IPv6 pelo endereço e pelo DNS. Após os testes concluímos o problema do software deveria ser descartado.
- Primeiro contato com responsável técnico da RNP na parte DNS Thiago Alves da Silva [thiago@na-cp.rnp.br]. Nos foram passados alguns materiais para estudo na implantação do DNS (BIND) com suporte IPv6.

Dia	17/07/2002
Assunto	Ethereal

- Efetuamos a instalação de um analisador de pacotes para podermos avaliar o que está ocorrendo com os pacotes na rede. Escolhido o software a ser utilizado (*ethereal*), fizemos seu download [www.kame.net] e posterior instalação. Os pacotes instalados foram os seguintes:

  ethereal -base-0.9.5-1.i386.rpm
  ethereal -gtkt-0.9.5-1.i386.rpm
  ethereal -kde--0.9.5-1.i386.rpm
  ethereal -usermode-0.9.5-1.i386.rpm

Dia	19/07/2002
Assunto	Formulários

- Após a obtenção de todos os dados, e de realizados os procedimentos necessários

para preenchimento da documentação da RNP, foi feito, por parte do prof. Rafael Timóteo de Souza Júnior, o envio do termo de adesão e da solicitação do endereço IPv6.

Dia	23/07/2002
Assunto	Scripts

- Realizamos a inclusão de *scripts* de inicialização no arquivo ***/etc/rc.d/rc.local*** que serão usados na configuração do túnel com a RNP.
- Posteriormente foi realizada a instalação do linux 7.2 em um laptop Compaq Presario para que pudéssemos trabalhar em melhorias durante o tempo que não estivéssimos no laboratório. A instalação foi realizada com sucesso.

Dia	25/07/2002
Assunto	Informações Técnicas

- Realizamos o contato com o serviço de informações da RNP solicitando informações a respeito do backbone como um todo, pedindo detalhes que auxiliem na visualização de todo o processo envolvido nesse projeto. Nos foi retornado o código do serviço de atendimento da própria RNP [Atendimento RNP #6518].

Dia	30/07/2002
Assunto	Interface de rede interna

- Para que pudéssemos fazer da máquina um roteador, fizemos a instalação de uma nova interface – eth1 (placa IBM) que será ligada à rede IPv6 da UnB. O endereço link-local IPv6 gerado automaticamente foi: ***fe80::200:b4ff:fea2:9782/10 – eth1*** . Já o endereço eth0 gerado foi: ***fe80::220:35ff:fe71:982e/10***.
- Foi realizado o download e instalação do software Mozilla [www.mozilla.com]. Essa instalação tem como propósito verificar "possíveis falhas" do Netscape6 no que diz respeito ao suporte.
- Outra instalação e configuração foi feita: a do software cliente chamado Freenet6. Este foi o primeiro serviço de servidor de túnel público amplamente utilizado para delegar automaticamente um endereço IPv6 único a qualquer host já conectado a uma rede IPv4. A instalação foi concluída com sucesso gerando uma interface virtual e endereço IPv6 do túnel e IPv4 do servidor. Testes mais aprofundados não foram feitos.
- Um teste simples foi realizado em nossa interface. Após análise na documentação, foi descoberto que o ping6 com o endereço link-local não era roteável, nos levando então a realizar o seguinte teste: adicionar o endereço inet6 fec0::2 à uma interface e pingá-la com o ping6. O teste funcionou.
- Recebimento dos endereços alocados para a UnB no BR-6bone no final do dia.

Dia	31/07/2002
Assunto	Túnel RNP

- Realizamos, de posse do documento da RNP que nos alocava um prefixo e designava os endereços relevantes, a configuração do túnel. Utilizamos o script já descrito e criado no dia 23/07/2002 e efetuamos os testes previstos para se verificar. Tudo funcionou perfeitamente [ping6, traceroute6, tracepeth6, www].

Dia	1/08/2002
Assunto	Informações técnicas

- Recebemos a resposta da RNP sobre informações técnicas a respeito do projeto Br 6Bone. As informações foram incompletas. Contudo, a UnB já faz parte do hall de endereços alocados no *backbone* Br6bone [http://www.6bone.rnp.br/cgi-bin/nets.pl]
- A máquina ponta do túnel foi configurada de modo a ter suporte *multicast*.
- Iniciamos testes após configuração do DNS para suporte IPv6. Tivemos alguns problemas.

Dia	2/08/2002
Assunto	DNS e Roteamento

- Concluímos os testes do BIND instalado em um PC do labredes. Os erros ocorridos durante as configurações do dia anterior foram erros de sintaxe. Alguns comandos não são aceitos no BIND IPv6.
- Concluída a etapa DNS do projeto, partimos para a realização de variações no que diz respeito à topologia labredes. Sendo uma das propostas por parte da gerência do labredes colocar o labredes-linux com suporte e endereços IPv6, iniciamos a configuração de uma máquina nessas condições. Alguns detalhes da configuração da tabela de roteamento precisam ser avaliadas.

Dia	06/08/2002
Assunto	Roteamento

- Foi colocada uma nova máquina Linux para acessar o túnel via Router Dual Stack. A configuração foi toda realizada.

Dia	09/08/2002
Assunto	Roteamento

- Efetuamos as configurações das tabelas de roteamento e das permissões de rota.
- Concluído esse procedimento, fizemos os mesmos testes de conectividade que usados no caso do roteador. Os testes foram um sucesso. A máquina acessa o túnel e utiliza o DNS da UnB.

Dia	10/08/2002
Assunto	Roteamento e NT

- Trabalhamos durante todo o dia na tentativa de configurar as máquinas do labredes – linux para suporte IPv6. As máquinas apresentaram problemas que não sabemos ao certo se estão relacionados ao Kernel.
- Efetuamos as instalações e configurações de duas soluções IPv6 para NT: Toolnet6 e MSR IPv6. A segunda solução foi considerada superior e com menos dificuldades de implementação. A máquina foi configurada como túnel da RNP para efetuarmos os testes que tiveram êxito.

Dia	12/08/2002
Assunto	Roteamento

- Os estudos a respeito do problema das máquinas do labredes – linux se prolongaram sem efetivamente constatarmos falha nos processos de configuração.
- Nos debruçamos sobre a monografia.

Dia	13/08/2002
Assunto	DNS

- Solicitamos aos responsáveis da RNP pelo DNS um teste para comparação da qualidade do serviço DNS usado no projeto. Por questões de segurança, a idéia foi abandonada e negada pela RNP.

Dia	14/08/2002
Assunto	Monografia

- Instalamos o Mozilla na máquina NT para efetuarmos os acessos com suporte HTTP (não funcionou).
- Começamos a padronizar e formatar a monografia. Esse trabalho durou a tarde e noite a dentro.

Dia	15/08/2002
Assunto	IE 6.0

- Através de listas de discussão, buscamos informações para o não funcionamento do IE 6.0 com suporte ao IPv6.

Dia	17/08/2002
Assunto	IE 6.0

- Obtivemos a informação de que realmente o IE 6.0 não continha suporte Http

IPv6. A solução proposta foi de substituir a versão do IE.

Dia	17/08/2002
Assunto	IE 6.0

- Efetuamos o *downgrade* do IE para a versão 5.5 em algumas máquinas do LabNT e substituímos o arquivo Wininet.dll pelo baixado junto com o driver IPv6 da Microsoft. Tudo funcionou perfeitamente.

Dia	18/08/2002
Assunto	IPv6

- Buscamos maiores informações a respeito dos porquês do IPv6 não estar em plena utilização. Várias pessoas de renome foram contactadas e as respostas foram similares.

Dia	19/08/2002
Assunto	QoS

- Entramos em contato com Maurílio Alves (Mestre pela UFSC) que desenvolvera pesquisa sobre QoS com IPV6.

Dia	19/08/2002
Assunto	Monografia

- Trabalhamos nos ajustes finais da monografia.

Dia	23/08/2002
Assunto	Monografia e servidor Apache

- Entrega da monografia e início da instalação do servidor Apache com suporte IPv6 nativo.

Dia	24/08/2002
Assunto	Servidor Apache

- Conclusão da instalação do servidor Apache com suporte IPv6 nativo.

Dia	26/08/2002
Assunto	Servidor Apache

- Etapa de criação da página destinada ao projeto final (https://www.redes.unb.br.Projetos_grad/ipv6/pojeto .html)

Dia	28/08/2002
Assunto	Slides

- Etapa de criação da apresentação para defesa do projeto final de graduação.