Instituto Científico de Ensino Superior e Pesquisa
Faculdade de Tecnologia
Curso de Tecnologia em Redes de Computadores

Elisnaldo Santiago Prazer

IPv6 versus IPv4
Características, instalação e compatibilidade

Brasília
2007

Prazer, Elisnaldo Santiago.
IPv6 versus IPv4: Características, instalação e compatibilidade / Elisnaldo Santiago Prazer; Professor orientador MSc. André Calazans Barreira. – Guará: [s. n.], 2007.
120f. : il.

Monografia (Graduação em Tecnologia em Redes de Computadores) – Instituto Científico de Ensino Superior e Pesquisa, 2007.

I. Calazans, André. II. Título.

Elisnaldo Santiago Prazer

IPv6 versus IPv4
Características, instalação e compatibilidade

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia da Faculdade de Educação do Instituto Científico de Ensino Superior e Pesquisa, como requisito parcial à obtenção do título de Tecnólogo em Redes de Computadores.

Área de concentração: Redes de computadores.

Orientadores: Prof. M. Sc. André Calazans Barreira e M. Sc. Gustavo Fleury Soares

Brasília
2007

Instituto Científico de Ensino Superior e Pesquisa
Faculdade de Tecnologia
Curso de Tecnologia em Redes de Computadores

Elisnaldo Santiago Prazer

Trabalho de Conclusão de Curso intitulado "**IPv6 versus IPv4: Características, instalação e compatibilidade**", avaliado pela banca examinadora constituída pelos seguintes professores:

_____

Prof. M. Sc. André Calazans Barreira Bareira – FACCIG/UnICESP – Orientador

_____

Prof. PHD Oscar Fernando Chaves Santana – FACCIG/UnICESP – Avaliador

_____

Prof. Reinaldo Mangilardo – FACCIG/UnICESP – Avaliador

_____

Prof. M. Sc. Paulo Hansem

Coordenador do Curso de Tecnologia – Redes de Computadores

FACCIG/UnICESP

RESULTADO

( ) APROVADO

( ) REPROVADO

Brasília, 05 de julho de 2007

PROPRIEDADE INTELECTUAL DE TRABALHO DE CONCLUSÃO DE CURSO – CESSÃO DE DIREITOS

Curso de Tecnologia em Redes de Computadores
Centro Tecnológico de Tecnologia da Informação
UnICESP

Título do Trabalho: IPv6 versus IPv4: Características, instalação e compatibilidade.

Autor: Elisnaldo Santiago Prazer

Orientador: Prof. M. Sc. André Calazans Barreira

Data de apresentação do Trabalho: 05 de julho de 2007

Declaramos que o *Centro Tecnológico de Tecnologia da Informação*, por meio da *Coordenação do Curso Tecnologia em Redes de Computadores*, da *Coordenação de Trabalhos de Conclusão de Curso de Tecnologia da Informação* e da *Coordenação Geral de Trabalhos de Conclusão de Curso*, do UnICESP, estão autorizadas a fazer uso do Trabalho por nós desenvolvido para a disciplina de Trabalho de Conclusão de Curso II – TCC II, para:

- Objetivos estritamente acadêmicos, como exposição/apresentação em Seminários ou Simpósios e outros eventos internos ou externos;
- Divulgação interna ou externa, para fins acadêmicos.

_____ _____

DEDICO este estudo a Deus Todo Poderoso, que me proporcionou vida, saúde e recursos necessários para cursar esta faculdade, à minha Esposa Keite de S. V. Prazer e minhas filhas Jade Camille e Maísa Lorena pelo amor e carinho a mim dedicados e pela paciência com minhas ausências inevitáveis durante todo o período deste curso de graduação, aos meus pais que me educaram com amor e carinho, moldando meu caráter de modo a poder, com minhas próprias pernas, conquistar meus objetivos, ao grande amigo Newton Marquês Alcântara que me incentivou e apoiou, se dispondo a me auxiliar com seus conhecimentos e experiência e aos meus orientadores M. Sc Gustavo Fleury e M. Sc. André Calazans que me apoiaram e orientaram com profissionalismo e atenção necessários para obtenção do resultado aqui apresentado.

Agradeço à:
Dataprev – Empresa de Tecnologia e Informações da Previdência Social que dispôs de equipamentos, infra-estrutura e tempo para realização de testes referentes ao objeto deste trabalho.

Leonardo Resende Carvalho, Marcio Prado Arcírio de Oliveira, Neilton Tavares Grangeiro, Rafael Marques Taveira, Elemar Marius Berbigier, Lino Augustus Bandeira, Robson Ricardo de Oliveira Castro, Luis Antonio Gomes Najan e demais colegas do setor de Suporte Técnico, todos bons amigos e colegas de trabalho que compreenderam minhas ausências ao serviço e me apoiaram no desenvolvimento desta pesquisa.

Aos amigos Guilherme Palhares, Wagner Martino, Luiz Cláudio Egito, Marcos Vinicius e demais colegas da turma pelo apoio e companheirismo durante todo o período do curso

Ao corpo docente e funcional desta instituição que, cumprindo seu papel funcional, proporcionou as condições necessárias para o desenvolvimento do curso.

À Professora Karem Kolarik dispôs de seu tempo para tornar a qualidade técnico visual deste trabalho dentro dos padrões impostos pela ABNT e UnICESP e à profesora Helena Alpino Rodrigues que dedicou sua atenção à valorização da língua portuguesa no texto aqui apresentado.

“Não diga a Deus o tamanho do seu problema!
Diga ao seu problema o tamanho de Deus!”
(Autor desconhecido)

“Quem conhece os outros é inteligente.
Quem conhece a si mesmo é iluminado.
Quem vence os outros é forte.
Quem vence a si mesmo é invencível.”
(Tao Te King)

“As pessoas felizes não conseguem tudo o que esperam,
mas querem a maior parte do que conseguem.
Em outras palavras, viram o jogo a seu favor,
escolhendo dar valor às coisas que estão ao seu alcance.
Mantenha um pé na realidade e lute para melhorar as coisas,
e não para torná-las perfeitas.
Não existe perfeição.
As coisas serão o que puderem ser
a partir de nossos esforços.”
(Diener, 1995)

RESUMO

Esta pesquisa tem como objetivo apresentar o protocolo IP em uma versão mais aprimorada – versão 6 – vislumbrando o conhecimento de seus benefícios em relação à versão 4 atualmente utilizada e as dificuldades ainda encontradas para a implementação. Neste estudo, buscamos documentar a prática baseada em uma diversidade de plataformas de sistemas operacionais mais utilizados atualmente no mercado de informática como FreeBSD (baseada em BSD – *Berkeley Software Distribution*), Fedora (baseada em RedHat), Ubuntu (Baseada em Debian) e Windows XP (compatível com a família do Windows 2003 *server*), todas com suporte a Ipv6 já incorporado no *kernel.* Dentre os grandes benefícios do novo protocolo, está a redução de processamento no roteador em função do novo processo de fragmentação do datagrama, a criação de cabeçalho de tamanho fixo, o conceito de cabeçalhos concatenados e o aumento, em quase três vezes, na capacidade de endereçamento de *hosts* e redes.

Palavras-chaves: Protocolo. Compatibilidade. Roteamento. Linux. Windows

ABSTRACT

The goal of this paper is to present the next generation of the Internet Protocol (also known as IPv6), focusing on its benefits as compared to the current version 4 and the challenges still present to its deployment. This study tried to document practical aspects based on a diversity of operating systems widely used nowadays such as FreeBSD (based on BSD - Berkeley Software Distribution), Fedora (based on Red Hat), Ubuntu (based on Debian) and Windows XP (compatible with Windows 2003 server family), all of them supporting IPv6 natively.
Among the great benefits of the new protocol are the load reduction over the router due to the extinguishment of datagram fragmentation, use of fixed length protocol header, the concept of header concatenation and the expansion of hosts and nets addressing in about three times.

key-Words: Protocol. Compatibility. Route. Linux. Windows

LISTA DE FIGURAS

Figura **Página**

LISTA DE TABELAS

Tabela **Página**

LISTA DE SIGLAS

Sigla	Nomenclatura
AH	*Authentication Header*
ARP	*Address Resolution Protocol*
ARPA	*Advanced Research Projects Agency*
ARPANET	Rede de comunicação de dados desenvolvida pela agencia ARPA
BGP	*Border Gateway Protocol*
BIT	Dígito do sistema BINÁRIO de numeração
BOOTP	*Bootstrap*
BSD	*Berkeley Software Distribution*
CBC	*Cipher Block Chaining*
CDMF	*Commercial Data Masking Facility*
CIDR	*Classes Inter-Domain Routing*
CSMA/CD	*Carrier Sense Multiple Access with Collision Detect*
DARPA	*Defense Advanced Research Projects Agency*
DCA	*Defense Communication Agency*
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name Server*
EGP	*Exterior Gateway Protocol*
ESP	*Encapsulation Security Payload*
Ethernet	Tecnologia de rede local que utiliza a tecnologia CSMA/CD
GGP	*Gateway-to-Gateway Protocol*
Hardware	Partes físicas de um sistema computacional, eletrônicas ou não.

HMAC	*Hashed Message Authentication Code*
HOP	Pontos de roteamento pelos quais um datagrama passa em uma transmissão
IAB	*Internet Architecture Board*
IANA	*Internet Assigned Numbers Authority*
ICCB	*Internet Control and Configuration Board*
ICMP	*Internet Control Message Protocol*
IESG	*Internet Engineering Steering Group*
IETF	*Internet Engineering Task Force*
IGMP	*Internet Group Management Protocol*
IGRP	*Internet Gateway Routing Protocol*
Internet	Rede pública
Intranet	Rede privada
IP	*Internet Protocol*
IPng	*Internet Protocol next generation* ou também conhecida como IPv6
IPSec	IP *Security*
IPv4	Versão 4 do protocolo IP
IPv6	Versão 6 do protocolo IP
MAC	*Media Access Control*
MAC	*Message Authentication Code*
Mainframe	Computadores de grande porte
MILNET	Segmento da ARPANET fechado para o serviço militar Norte Americano
MTU	*Maximum Transmission Unit*
NAT	*Network Address Translation*
ND	*Neighbor Discovery*

NDP	*Neighbor Discovery Protocol*
NIC	*Network Interface Card*
Octetos	Conjunto de oito bits
OSPF	*Open Shortest Path First*
PAD	*Packet Assembler/Disassembler*
payload	Parte de dados do cabeçalho IPv6
RFC	*Request for Comments*
RIP	*Routing Information Protocol*
S/L	*Strict/Loose Bit Map*
SA	*Security Association*
SHA	*Secure Hash Algorithm*
SIP	*Simple IP*
SIPP	*Simple IP Plus*
Socket	Interface de *software* para acesso direto de programas aos protocolos
Software	Parte lógica de um sistema computacional
SPI	*Security Parameters Index*
TCP	*Transmission Control Protocol*
TCP	*Transmission Control Protocol*
TLV	*Type-Length-Value*
TOS	*Type Of Service*
UDP	*User Datagram Protocol*
VPN	*Virtual Private Network*
www	*World Wide Web*

SUMÁRIO

Capítulo **Página**

INTRODUÇÃO

Com a grande evolução da computação, o ser humano torna-se cada vez mais dependente dos recursos oferecidos pela tecnologia da informação. Atualmente os eletrônicos e os sistemas informatizados estão sendo compatibilizados, e os grandes responsáveis por essa evolução são os recursos de comunicação de dados.

Antes dos anos 90, cada aparelho possuía função específica, ou seja, aparelho de som para reprodução de músicas, telefone para conversação, máquina fotográfica para registro de imagens e computadores para edição de documentos, programação, execução de aplicativos e comunicação de dados, áreas estas que atendiam apenas a interesses de grandes corporações para interligação de seus *mainframes*.

As redes de informação ganharam espaço nas atividades mais simples do dia-a-dia, ensejando interação entre os mecanismos eletro-eletrônicos. Marcante exemplo disso é a telefonia celular, hoje capaz de trafegar voz, dados, imagem com a utilização de um aparelho telefônico através de redes integradas à *Internet*. Computadores exercem todas as atividades de eletrônicos como conversação, execução de jogos, captura e edição de imagens e vídeos, além da execução de músicas.

Tudo isso foi possível em função do protocolo *Internet Protocol* (IP), que permite a comunicação entre hardwares e sistemas de diferentes tecnologias, o que o tornou muito difundido, fazendo-se necessária a criação de mecanismos de segmentação entre as redes privadas (Intranet) e a pública (Internet).

Às redes privadas, principalmente nos ambientes corporativos, foi permitida a adoção de endereços IP de escolha própria cuja administração é realizada exclusivamente pela organização. Já a rede pública possui uma unidade controladora em cada país a fim de gerenciar a distribuição de IPs impedindo assim a duplicidade. No Brasil, a responsável por essa distribuição é a FAPESP.

A integração dos dois ambientes, redes privadas e públicas, dá-se por dispositivos de *Network Address Translations* (NATs) ou tradução de endereçamento de redes, dispositivos esses que são implantados e administrados pela organização que tem a responsabilidade pelo seu funcionamento.

Mesmo com vários recursos existentes para melhor aproveitamento dos endereços públicos, a quantidade de usuários e empresas que utilizam a Internet cresceu tanto que os mais de 4 bilhões de endereços da *Internet* (excluindo deste quantitativo as faixas reservadas) tendem a esgotar-se em poucos anos.

Este foi o cenário que motivou grupos de pesquisa a trabalharem em um novo modelo de endereçamento TCP/IP a fim de atender à demanda mundial. Este modelo recebeu o nome de *Internet Protocol - next generation* (IPng), e sua versão oficial é a IPv6 que entra em atividade para substituir a atual IPv4.

A justificativa da pesquisa, a par do crescimento exponencial de utilização da Internet, arrima-se na escassez de endereços públicos e maior consumo dos enlaces de telecomunicações.

Com a grande dependência desses recursos, um colapso no sistema acarretaria prejuízos imensuráveis para profissionais liberais e empresas que dependem de tal estrutura para manter seus negócios. A fim de evitar se chegue à situação crítica, foi desenvolvida a versão 6 do protocolo IP.

Com efeito, eleva o IPv6, substancialmente, a quantidade de endereços para a Internet, bem como diminui o uso de banda de rede e reduz informações no cabeçalho de dados.

Muito mais que a ampliação da faixa de endereçamento oficial, o IPv6 visa a melhorar o desempenho das redes, principalmente nos pontos onde se exige maior processamento para passagem dos datagramas entre redes diferentes (roteadores).

Outra questão importante diz respeito ao novo conceito de “cabeçalhos concatenados”. Isso permite um melhor gerenciamento de segurança do datagrama desde as atividades mais simples como um “*echo replay*” (comando ping) até as mais complexas como criptografia e *Virtual Private Network* (VPN), por exemplo.

Nesta ordem de idéia, já se afigura oportuno que todas as grandes corporações iniciassem plano de migração para esta nova tecnologia do protocolo IP. Mudança que poderá ser implementada em longo prazo, uma vez que existem serviços de compatibilização que tornam comunicável as duas versões do protocolo. Até porque essa migração feita de imediato poderia representar alto custo em hardware e software caso os equipamentos da corporação não suportassem IPv6.

1. OBJETIVO GERAL

O Trabalho de Conclusão de Curso (TCC) tem o objetivo precípuo de apresentar um estudo do protocolo IPv6, com destaques para suas inovações em relação ao IPv4.

2. OBJETIVOS ESPECÍFICOS

- Referir as diferenças entre as versões 4 e 6 do protocolo, comparando o modelo didático com pacotes coletados em uma comunicação de dados entre os sistemas operacionais mais utilizados no mercado, onde haja comunicações entre IPs de mesma versão (v6) e versões diferentes (v6 e v4) utilizando algumas técnicas que comprova a compatibilidade entre elas.
- Apresentar um laboratório prático com alguns dos sistemas operacionais mais utilizados, com o funcionamento e a compatibilidade técnica entre estes protocolos, o que poderá servir de base para montagem de laboratórios de estudo e planejamento, como alternativa de diminuição dos impactos, principalmente com a substituição de *hardware* e ajustes de *softwares* ainda não compatíveis

3. METODOLOGIA

Foi realizado um estudo sobre as características da estrutura do protocolo IPv6 baseada em comparação com o IPv4 para melhor compreensão.

Uma rica bibliografia foi objeto de consulta, bem como bons *sites* da *Internet,* a fim de se obterem dados suficientes para um correto entendimento sobre o assunto.

As propostas alusivas à montagem de laboratórios apresentadas neste trabalho estão amplamente fundamentadas no material de estudo e, em alguns

casos, em experiências vividas e compartilhadas por internautas através de artigos ou dicas publicadas em diversos sites.

4. ESCOPO DA PESQUISA

Este trabalho de pesquisa apresenta os prós e os contras da versão 6 do protocolo IP, com fulcro em literaturas diversas, no intuito de orientar os administradores de redes sobre como montar um laboratório de testes para avaliação das possibilidades de configuração do novo IP.

5. REFERENCIAL TEÓRICO

O objetivo aqui é realizar um estudo detalhado sobre a versão 6 do protocolo IP apresentando os serviços de nível 3 mais comuns em uma rede. Este estudo está voltado para a compreensão sobre como esta versão se comporta e o que ela traz de vantagem sobre o atual IPv4.

5.1. APRESENTAÇÃO DO PROTOCOLO TCP/IP

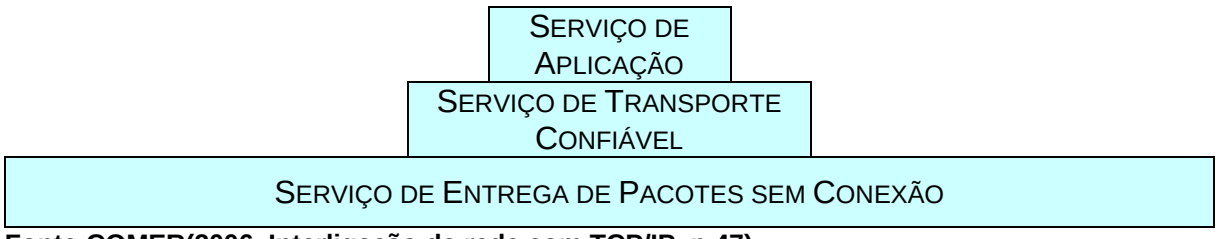
O TCP/IP é um protocolo baseado em 5 camadas (Figura 1) e apresenta as melhores condições de roteamento e serviços de rede. Estes serviços estão divididos em três grupos dependentes entre si conforme Figura 2.

Figura 1 – Modelo de referencia do TCP/IP baseado em 5 camadas

Aplicação	Sistemas e aplicativos
Transporte	Verificação de erros
Internet	Serviços de endereçamento lógico e roteamento
Interface de Rede ou Enlace	Endereçamento físico *Media Access Control*(MAC)
Física ou de *hardware*	Infra-estrutura de rede (cabos, conectores, etc.)

Fonte: própria

Figura .2 – Camadas conceituais dos serviços de internet (TCP/IP)



Fonte:COMER(2006, Interligação de rede com TCP/IP, p.47)

Esta divisão facilita o desenvolvimento das pesquisas no sentido poderem seguir suas linhas de trabalho simultaneamente.

O IP é definido como um sistema de entrega de pacote não-confiável, sem conexão e com a filosofia do melhor esforço. Não-confiável, pois a entrega do pacote não é garantida pelo protocolo, o que garante tal entrega são os protocolos

adjacentes. Sem conexão, pois o pacote não estabelece um único caminho para uma seqüência de quadros, ou seja, um dado pode ser fragmentado e cada fragmento pode ser enviado por caminhos diferentes, sendo reagrupados e reorganizados no destinatário. O melhor esforço é caracterizado pela procura do caminho de menor custo para entrega do datagrama. Esta procura é realizada utilizando-se como base de referência o tráfego, a banda disponível, o retardo do canal, a perda de pacotes e a integridade do link.

Na versão 4, o protocolo é totalmente compatível com os *hardwares* e *softwares* disponíveis no mercado. Já na versão 6, como ainda está em fase de projeto, os primeiros sistemas e equipamentos compatíveis estão começando a aparecer, depois de quase 10 anos de homologação do serviço..

5.2. BREVE HISTÓRICO

O protocolo IP teve sua origem na década de 70 quando o *Defense Advanced Research Projects Agency* (DARPA), um dos maiores financiadores de pesquisa de redes de comutação de pacotes, iniciou os trabalhos para a criação de uma tecnologia para interconexão de redes.

O DARPA desenvolveu sua primeira rede utilizando a tecnologia IP, que foi batizada de ARPANET.

Em 1979, o grupo de pesquisadores interessados no desenvolvimento da tecnologia IP estava tão grande, que foi necessária a criação do comitê *Internet Control and Configuration Board* (ICCB) para coordenar os trabalhos.

Nos anos 80 a ARPANET começou a se expandir a ponto de ocorrer a interligação entre os centros de pesquisa. Como tal expansão apontava para uma grande massa de usuários, a Agência de Defesa nas Comunicações (DCA), que já havia ingressado a ARPANET, determinou a separação desta em uma rede militar (que passou a ser chamada de MILNET) e outra rede de pesquisas (que permaneceu com o mesmo nome).

Nesta mesma época, a versão *Berkeley* ou *Berkeley Software Distribution* (BSD) do UNIX incorporou o TCP/IP ao seu Sistema Operacional. Com efeito, permitiu que mais de noventa por cento dos departamentos de ciência de

computação das universidades estivessem interligados, tornando possível a transferência de arquivos entre elas.

A *Berkeley* também desenvolveu o *socket*, uma interface para que sistemas aplicativos acessassem diretamente protocolos de comunicação, não apenas para o TCP/IP, mas também para outros que já existiam na época.

No final da década de 80, o crescimento da Internet já atingia uma taxa de 15% ao mês, chegando em 2005 com aproximadamente trezentos milhões de computadores distribuídos em 209 países.

O *Internet Architecture Board* (IAB) foi o primeiro órgão regulador a propor criação de padrões de evolução da Internet. Definiu quais protocolos fariam de forma inevitável parte do conjunto TCP/IP e quais seriam as suas políticas oficiais.

Em 1989, milhares de pessoas já tinham a Internet como fonte de negócios, o que dificultava as implementações experimentais de novas idéias. Em função disso, o IAB passou por uma reestruturação e criou um grupo com mais representantes da comunidade., que ficou conhecido como *Internet Engineering Task Force* (IETF) e tornou-se responsável pela padronização de protocolos e por outros aspectos técnicos.

Para se registrar os trabalhos de pesquisa com o TCP/IP, utilizaram-se as *Request for Comments* (RFC)s de forma seqüencial e cronológica, cujos conceitos, padrões e propostas de protocolos são aprovados pela Internet *Engineering Steering Group* (IESG), um departamento do IETF, que oficializa tais registros. Estas RFCs podem ser obtidas no site do IETF (http://www.ietf.org).

O maior fomentador de demandas tecnológicas, não vem do crescente aumento da quantidade de conexões, mas sim da necessidade de tráfego que as novas aplicações e serviços exigem para funcionarem bem na Internet.

Quando a rede mundial de computadores foi implementada, utilizava-se linha de comando para as transações de dados. Com o surgimento dos sistemas operacionais gráficos, vieram as aplicações *Word Wide Web*(www), o que representou um drástico aumento na utilização do canal de comunicação. Como a Internet está constante evolução, as demandas por protocolos e equipamentos mais ágeis não param de aumentar. Para se ter uma idéia, tem-se atualmente voz e vídeo utilizando a mesma tecnologia existente há trinta anos.

Na tabela abaixo apresentamos a expansão do uso da Internet, bem como a quantidade de gerências nacionais que se espalharam pelo mundo, visando uma melhor administração da evolução.

Tabela 1 – Tabela de evolução da *Internet*

Ano	Número de redes	Número de computadores na rede	Número de usuários utilizando a rede	Número de gerentes
1980	10	10^2	10^2	10^0
1990	10^3	10^5	10^6	10^1
2000	10^5	10^7	10^8	10^2
2005	10^6	10^8	10^9	10^3

Fonte:COMER(2006, Interligação de rede com TCP/IP, 5ª ed. P. 07)

Frente ao apresentado na tabela acima, pesquisadores chegaram a uma estimativa de que a disponibilidade de endereços de rede se prolongaria até 2028 se houvesse um bom reaproveitamento dos endereços não utilizados. De qualquer forma, duas evidências são consenso com base na estatística de crescimento da Internet: Aplicações mais sofisticadas como telefonia e videoconferência funcionam muito bem na versão 4 do IP e, os serviços de *Classes Inter-Domain Routing* (CIDR) e *Network Address Translation (*NAT) conseguiram estender os endereços oficiais à quantidade necessária à demanda atual.

Considerando que a versão 4 do IP, totalmente inalterada, funciona há 30 anos de modo eficaz, chegamos à conclusão de que o projeto é bastante flexível e poderoso, uma vez que neste período

> [...] o desempenho do processador aumentou por três ordens de grandeza, os tamanhos típicos de memória aumentaram por um fator maior que 400 e a largura de banda dos enlaces de maior velocidade na Internet aumentaram por um fator de cento e cinqüenta mil. (COMMER, ano 2006 p.370).

À vista de tudo isso, os pesquisadores do IETF convidaram representantes de muitas comunidades para participarem do processo de criação de uma nova versão do TCP/IP. Este novo grupo de trabalho elaborou algumas propostas para base do novo protocolo, dentre as quais a melhor foi denominada *Simple IP*(SIP), que teve sua versão estendida chamada de *Simple IP Plus* (SIPP). Desta proposta originou-se o IPng, que posteriormente foi denominado IPV6, em virtude da versão 5 apresentar uma série de erros e mal entendidos.

5.3. INTERNET PROTOCOL VERSÃO 6 (IPv6)

A versão 6 do protocolo IP, nada mais é do que uma melhoria da versão 4, uma vez que esta última é um projeto muito flexível e estável visto que já está em uso a mais de 30 anos e ainda consegue acompanhar as evoluções dos hardwares, softwares e redes, incorporando novos sistemas e serviços cada vez mais complexos e que exigem cada vez mais das redes.

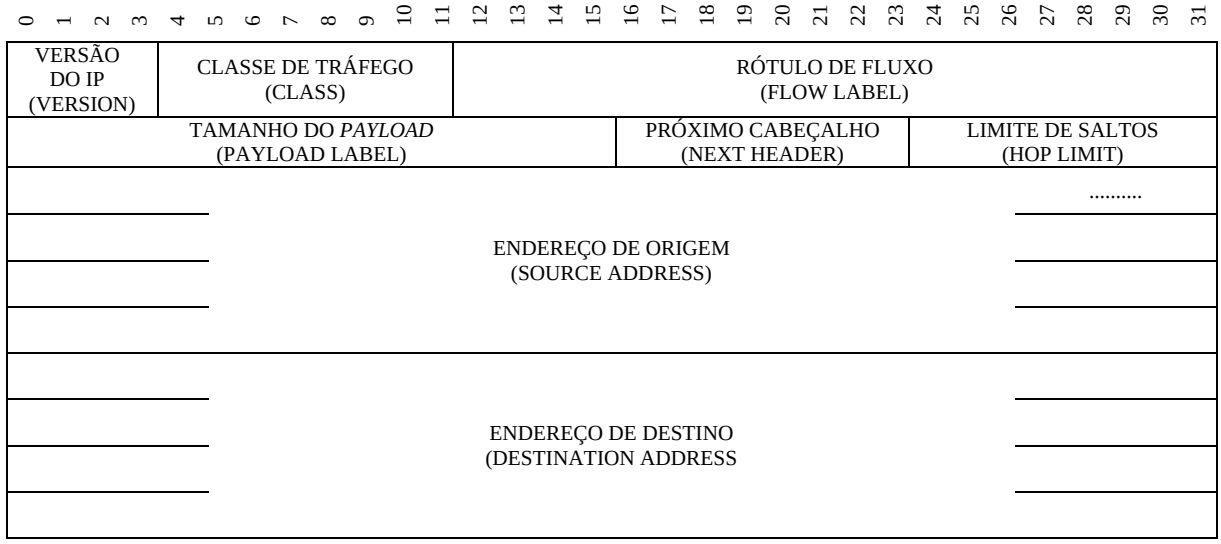
O principal argumento, então, é a re-estruturação do formato de endereçamento de modo a atender a carência de alocações públicas. Como será observado neste trabalho, algumas diferenças de funcionamento apareceram, porém a transparência de comunicação entre as versões foi preservada.

As limitações estão mais presentes nos *firmware* e *softwares* dos equipamentos que, em alguns casos, poderá haver a necessidade de substituição ou simplesmente atualização do sistema.

5.3.1. CABEÇALHO DO DATAGRAMA IPv6

Como podemos observar na figura 3, o novo cabeçalho foi simplificado em relação à versão 4 no que se refere a campos de controle, porém o seu tamanho passou de 20 a 40 *bytes* fixos para qualquer tipo e tamanho de datagrama. Isso se deve, sobretudo, ao novo tamanho do endereço que passou de 4 para 16 octetos como veremos no item 5.3.8.

Figura 3 - Modelo do datagrama IP versão 6



Fonte: Commer (2006,p.372)

Abaixo segue a função de cada campo:

- **VERS –** É praticamente o único campo que não foi modificado, ou seja, tem como função identificar a versão do cabeçalho do datagrama. No caso da versão 6, ele vem setado como 0110.
- **CLASS** – A classe de tráfego possui função semelhante ao campo *Type of Service* (TOS), ou seja, definir a prioridade do datagrama em um mecanismo de roteamento. Na versão 6, é possível ao roteador distinguir tipos de tráfego como:
    - **"Controlado por congestionamento"** (*congetion-controlled*) – Neste caso, os pacotes são retirados da rede quando o índice de utilização do canal está muito alto. Possui 7 níveis de prioridade e seleção sobre quais datagramas serão descartados, seguindo a ordem da tabela 2, ou seja, datagramas de FTP possuem prioridade sobre os datagramas de e-mail, login remoto, sobre os de FTP e assim por diante.
    - **"Não controlado por congestionamento"** (*noncongestion-controlled*) – Os datagramas de prioridade inferior são descartados para priorizar o tráfego de datagramas com prioridades superiores. São sete

níveis numerados de 8 a 15, mas não possuem atribuições especificadas. Esta classe de dados normalmente é caracterizada por informações que exigem transmissão em tempo real, por exemplo, serviços de áudio e vídeo de alta definição.

- ***Flow Label*** – Tem como função definir um controle de fluxo relacionado com o tipo de aplicação. Cada *host*, ao estabelecer uma comunicação com outro em uma rede distinta, informa um valor neste campo. O roteador memoriza-o a fim de criar uma espécie de reserva de banda para essa comunicação. Caso este *host* deixe de usar esse "canal" o roteador libera essa reserva para outras comunicações. Essa reserva dura aproximadamente seis segundos após o encerramento do ultimo datagrama trafegado. Isso acontece para evitar que um *host* tente utilizar o mesmo rótulo para comunicações distintas pelo mesmo roteador. Uma vez estipulado o rótulo de fluxo e respeitando o prazo de validade em um roteador, o serviço de roteamento não irá mais verificar dados do datagrama, já que no primeiro enviado, estas informações já foram armazenadas pelo roteador. A associação entre a CLASSE DE TRÁFEGO, RÓTULO DE FLUXO e ENDEREÇO DE ORIGEM, geram uma identificação única do datagrama em uma comunicação de dados.
- **Tamanho do *payload*** – Informa o tamanho do datagrama sem o cabeçalho. Se este for maior que 64KB (*jumbogramas*), outro cabeçalho (cabeçalho de opção) entrará no datagrama informando o tamanho real.
- ***Next header –*** Apresenta o tipo de cabeçalho que virá logo em seguida ao cabeçalho base (no IPv6 um datagrama pode conter vários cabeçalhos anexados). Este campo equivale de certo modo, ao PROTOCOLO do IPv4. Tanto que os protocolos do nível superior ainda continuam valendo conforme pode ser observado na tabela 3. Porém, o referido campo vai além de informar apenas o protocolo, permite, ainda, a criação de

cabeçalhos de extensão (ou cabeçalhos concatenados) para definir qual o serviço do datagrama.

- ***Hop-Limit*** – É a duração do datagrama em uma rede definida em “saltos” de um roteador para outro. Ele é equivalente ao campo TTL do IPv4, com uma única diferença teórica: No TTL, a unidade de medida é o segundo, mas já foi comprovado que os roteadores atuais conseguem transmitir datagramas entre si, em intervalos de tempo bem inferiores a 1s. Como em cada roteador é decrementado uma unidade no valor deste campo, é recorrente que o tempo de vida seja contado em saltos que duram menos que 1s.
- ***Source Address*** – Este campo de 128 bits, assim como no IPv4, indica qual o endereço de origem do datagrama.
- ***Destination Address*** – Informa o endereço de destino da informação.

Tabela 2 – Tabela de classes de datagramas para tráfego controlado por congestionamento

Valor de prioridade	Especificação da prioridade
0	Prioridade não especificada
1	Tráfego em *background*
2	Transferências não continuadas (ex. e-mail)
3	Reservado para definições futuras
4	Transferências continuadas de grandes volumes (ex. FTP)
5	Reservado para definições futuras
6	Tráfego interativo (ex. logins remotos)
7	Controle de tráfego (ex.: protocolos de roteamento e gerencia de redes)

Fonte: THOMAS(1996 p.98)

Tabela 3 – Tabela de protocolos válida para o campo NEXT HEADER

Código do protocolo*	Descrição
0	Reservado
1	ICMP
2	IGMP
3	GGP
4	IP (encapsulamento IP)
5	*Stream*
6	TCP
8	EGP
11	UDP
29	IPv6
2B	Roteamento
2C	Fragmentação
32	ESP
33	AH
3A	ICMPv6
3B	Sem próximo cabeçalho
3C	Opções de destinação

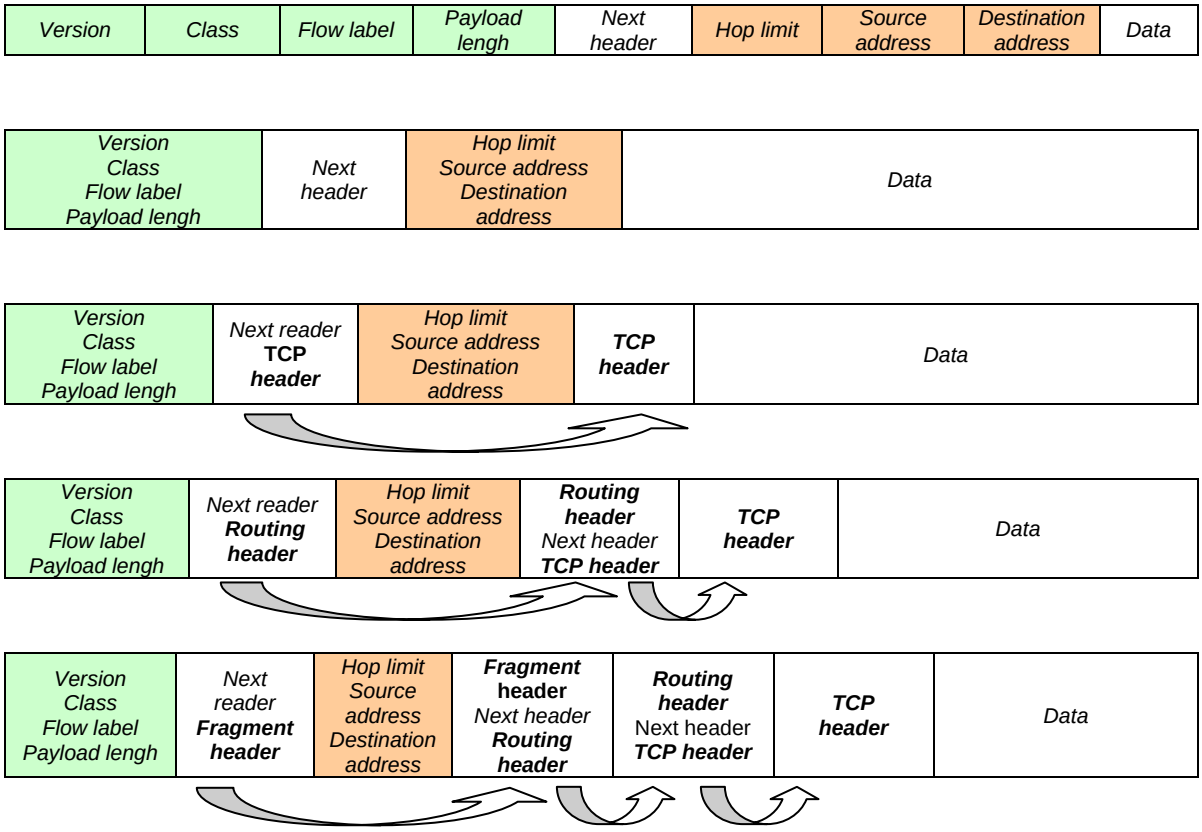
Fonte:MURHAMMER, Martin W. et all(2000 p.49),MILLER(1997, p.40)
***Valores expressos em Hexadecimal**

5.3.1.1. Cabeçalhos Concatenados IPv6

Essa é uma das grandes inovações da versão 6 do IP que, na versão 4, era utilizado apenas na criptografia ESP. Isso permite uma maior flexibilidade ao datagrama que terá como resultado, maior agilidade na comunicação dos dados entre os roteadores.

Como podemos ver na figura 4, os cabeçalhos de extensão e serviços são anexados entre os dados e o endereço de destino. Todos eles possuem um campo PROXIMO CABEÇALHO de 8 bits como o apresentado no cabeçalho principal. Porém a quantidade de cabeçalhos que podem ser anexados é limitada e segue uma ordem numérica (apesar do destino não checar tal ordem, ela facilita nas atividades de roteamento) conforme apresentado na tabela 5.

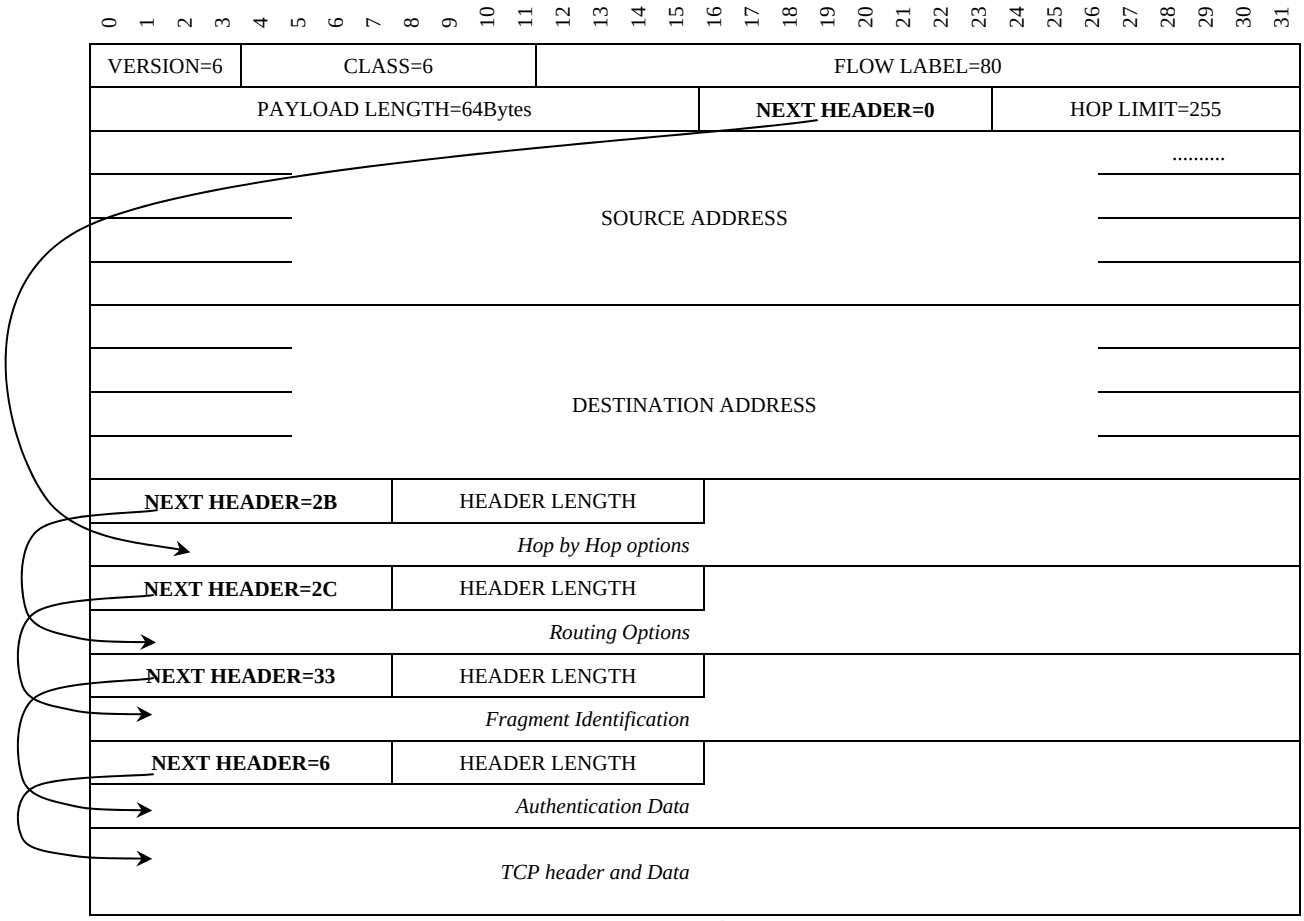
Figura 4 – Conceito de cabeçalhos concatenados



Fonte: Mark (1997,p.41 e 45)

Os cabeçalhos de extensão também possuem campos próprios referentes aos serviços a eles atribuídos. Na próxima figura tem-se uma noção mais detalhada de um datagrama com alguns cabeçalhos de extensão.

Figura 5 – Composição de um datagrama IPv6 com os cabeçalhos de extensão



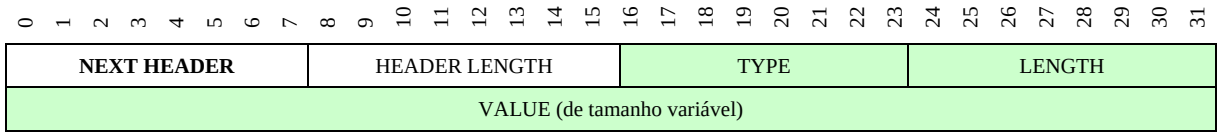
Fonte: STEPHEN(1996, p.108), TCP/IP Tutorial e Técnico(2000, p. 352) e MARK(1997, p.45)

Observe que cada campo PROXIMO CABEÇALHO aponta para o campo de informações do cabeçalho seguinte. Releva mencionar, também, que cada cabeçalho possui um tamanho próprio dependendo do seu tipo e seu valor que sempre será múltiplo de 8. Na seqüência do trabalho, tal noção será pouco a pouco esclarecida.

5.3.2. CABEÇALHO HOP-BY-HOP OPTIONS (SALTO A SALTO)

Este cabeçalho tem por objetivo informar aos “nós” por onde ele passa, incluindo o próprio destinatário, sobre o tamanho do datagrama (se for do tipo JUMBO), características de criação do pacote e ações do roteador conforme sua interpretação do datagrama. Sua opção de extensão é variável no formato *Type-Length-Value* (TLV) segundo observa-se no esquema a seguir:

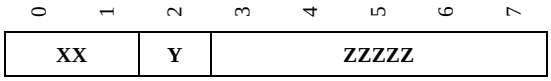
Figura 6 – Composição do cabeçalho *hop-by-hop*



Fonte: MARK(1997, p.46)

- **TYPE** – Possui tamanho de 8 bits e segue a representação abaixo:

Figura 7 – Formato do campo TYPE



Fonte: TCP/IP Tutorial e Técnico (2000,p 353)

  - XX – Informa como um nó que não reconhece o tipo do datagrama deve processá-lo. Veja a tabela abaixo

Tabela 4 – Valores de tipo de tratamento de datagramas pelos nós da rede

Valor	Descrição
00	Pula a opção e continua processando o cabeçalho
01	Descarta o pacote
10	Descarta o pacote e envia uma mensagem ICMP ao remetente de que o pacote não foi reconhecido
11	Descarta o pacote e envia uma mensagem ICMP ao remetente (se o endereço de destino não for um *multicasting)* de que o pacote não foi reconhecido

Fonte: MARK(1997, p.48) e TCP/IP Tutorial e Técnico (2000,p. 353)

  - Y – Informa se o valor do campo VALUE pode ser modificado ao passar pelos nós da rede. Se 0, não pode haver modificação. Se 1 pode haver modificação
  - ZZZZZ – Informa quantos *bytes* de preenchimentos serão utilizados para manter o campo VALUE com tamanho múltiplo de 8. Isso acontece, pois os “limites naturais de palavras” nos processadores modernos são de 16 ou 64 bits. Quando este campo está setado para 0, indica que usaremos PAD1, ou seja, apenas um byte será preenchido. Qualquer outro valor entre 1 e 194 informa a quantidade de *bytes* de preenchimento que foram utilizados. O valor 194 informa que o datagrama é do tipo JUMBO.

- **LENGTH** – Apresenta o tamanho do campo VALUE em bytes
- **VALUE** – Depende do tipo do cabeçalho.

5.3.3. CABEÇALHO DESTINATION OPTIONS

Este cabeçalho tem objetivo quase idêntico ao do cabeçalho *hop by hop,* ou seja, informar sobre o tamanho do datagrama (se for do tipo JUMBO), as características de criação do pacote e as ações do roteador conforme sua interpretação. A diferença está nas informações nele contidas que são de interesse exclusivo do destinatário. A semelhança entre eles é tanta que a maioria das literaturas os trata em um único item.

O cabeçalho Destination Options é opcional e mais utilizado por sistemas de criptografia na comunicação.

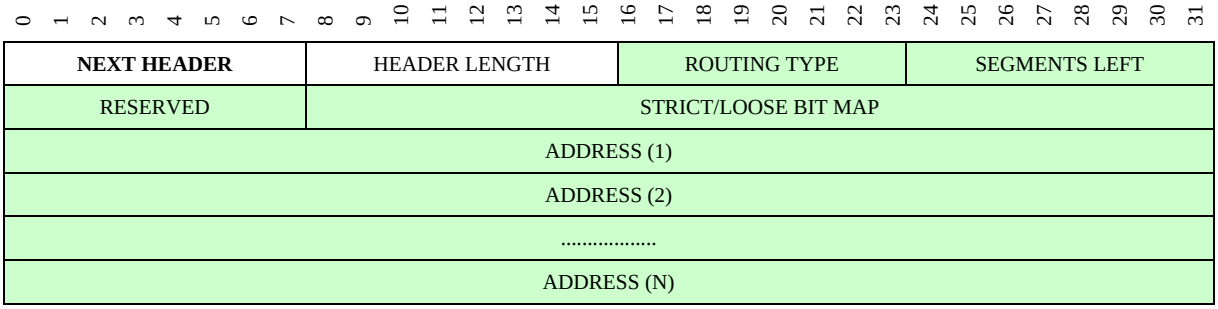
.

5.3.4. CABEÇALHO ROUTING

O cabeçalho de roteamento tem como propósito permitir que o *host* de origem possa definir o caminho pelo qual o datagrama deve passar. Isso é possível, uma vez que é responsabilidade do nó, conhecer todos os *Maximum Transmission Units* (MTU) pelos quais seu pacote trafegará, a fim de retirar dos roteadores a tarefa de fragmentação.

Uma vez conhecendo os MTUs, as informações de largura de banda e confiabilidade, também são recebidas. Então, o *host* manda pacotes que necessitem de mais segurança por um caminho específico e os demais por rotas livres. Vejamos a seguir o diagrama do cabeçalho de roteamento.

Figura 8 – Formato do cabeçalho de roteamento



Fonte: MARK(1997, p.50)

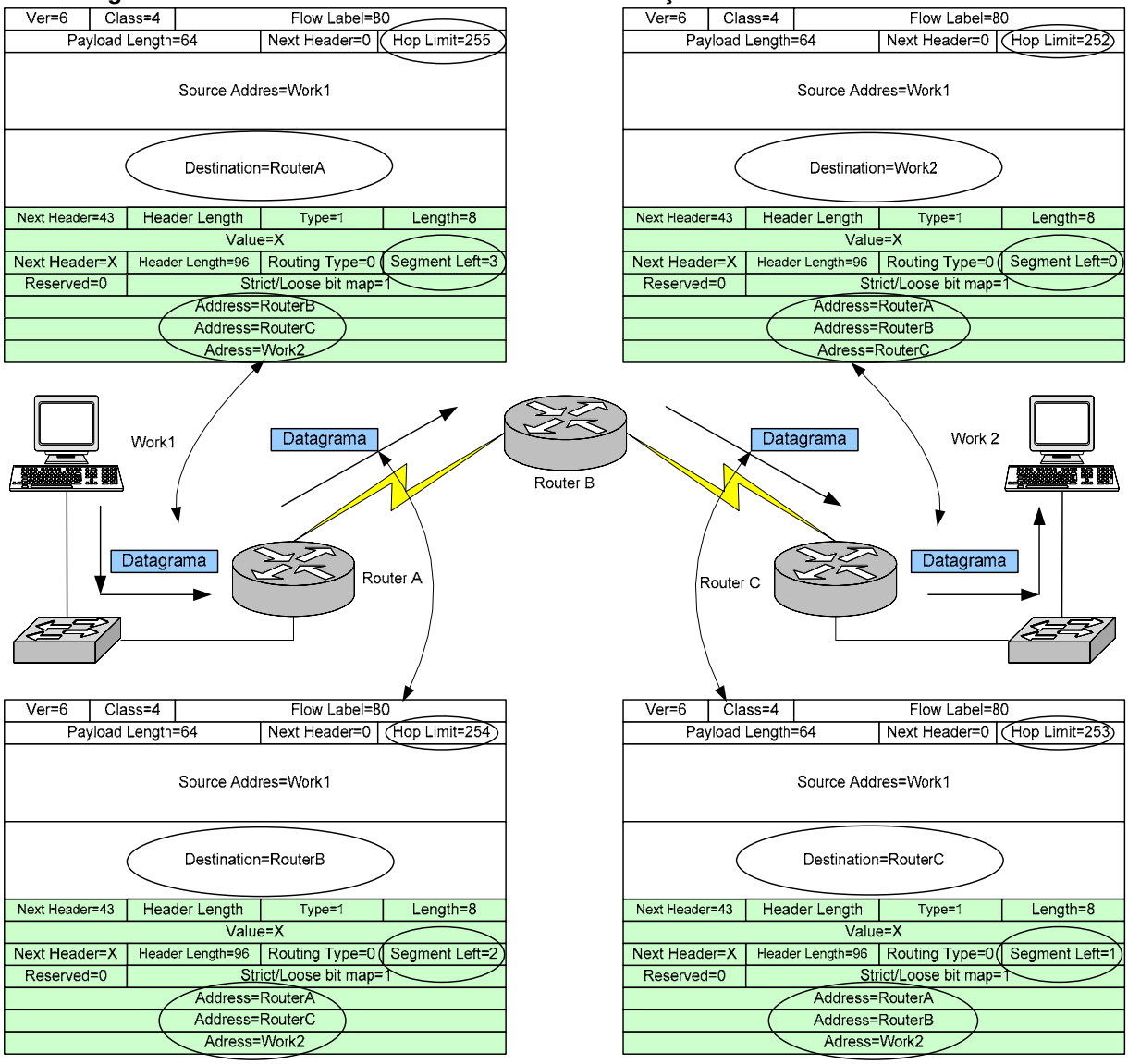
- **ROUTING TYPE** – Define o tipo de roteamento para o datagrama. Atualmente apenas o valor 0 pode ser atribuído. Este valor permite um roteamento de origem Fixo/Livre (*STRICT/LOOSE*).
- **SEGMENTS LEFT** – Indica quantos nós faltam para que o datagrama atinja seu destino. Ele é decrementado toda vez que o datagrama passa por um nó apontado no campo *DESTINATION ADDRESS* do cabeçalho básico.
- **RESERVED** – Inicializado como 0 na transmissão e ignorado na recepção.
- **STRICT/LOOSE BIT MAP(S/L)** – Atualmente existem dois valores básicos para este campo.
    - 1 (*strict*) – A origem especifica o caminho pelo qual o datagrama deverá passar.
    - 0 (*loose*) – Indica que o datagrama pode ser conduzido conforme orientação dos próprios roteadores durante o percurso entre uma origem e um destino, ou seja, não há regras que definam o percurso.
- **ADDRESS 1 a N** – Caso o campo *STRICT/LOOSE BIT MAP* esteja setado para 1, estes campos (pois pode ter vários no mesmo datagrama conforme figura 9) irão determinar os endereços pelos quais o datagrama deverá “viajar”.

Como já foi dito, os campos de endereços representam o “mapa de viagem” do datagrama. À medida que em que o pacote percorre a rede, o campo *SEGMENT LEFT* vai sendo decrementado e o campo *DESTINATION HOST* do cabeçalho

básico vai recebendo o próximo endereço da lista. Veja a ilustração 9 que representa bem essa rotina.

Observe que os campos de endereços vão sendo trocados à medida em que o datagrama passa por um nó de roteamento e cada endereço usado é transferido para o topo da lista.

Figura 9 – Mecanismo de funcionamento do cabeçalho de roteamento com S/L=1



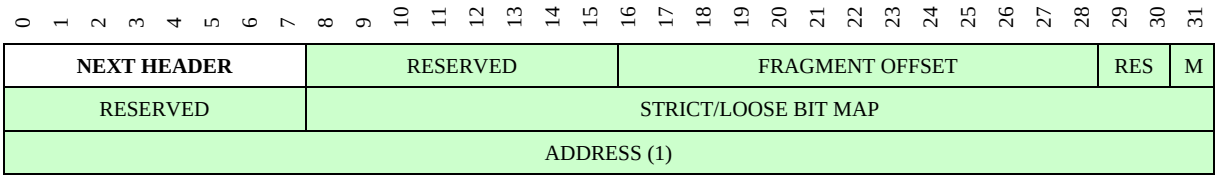
Fonte: Própria

5.3.5. CABEÇALHO FRAGMENT

Como já mencionado na introdução do item 5.3.4, a origem deve conhecer qual o menor MTU do caminho pelo qual o seu datagrama deve passar. Uma vez de posse dessa informação, a origem deverá particionar, se necessário, seu pacote em fragmentos que atendam a esse tamanho. O cabeçalho de fragmentação tem esse propósito.

Esse cabeçalho possui o valor 44 e segue o seguinte formato:

Figura 10 – Formato do cabeçalho de fragmento



Fonte: MARK(1997, p.54)

- **NEXT HEADER** – Indica o endereço do próximo cabeçalho de extensão.
- **RESERVED** – Ainda sem aplicação prática, é inicializado com 0 na origem e ignorado no destino.
- **FRAGMENT OFFSET** – Assim como no IPv4, este campo de 13 bits indica a posição, em unidades de 8 bytes, relativa ao inicio dos dados do fragmento.
- **RES** - Ainda sem aplicação prática, é inicializado com 0 na origem e ignorado no destino.
- **M** – Este campo indica se o datagrama é o último fragmento de uma seqüência. Se “setado” em 1, ainda há fragmentos do mesmo pacote a serem transmitidos. Se 0, não há mais fragmentos.
- **IDENTIFICATION** – Assim como no IPv4, este campo oferece uma identificação única ao fragmento, porém seu tamanho é duas vezes maior.

Um datagrama de fragmentação deve ser montado em duas partes: A parte não fragmentável que consiste do cabeçalho básico, com o endereços de origem e

destino conforme indicado no item 5.3.4 (Cabeçalho *Routing*), seguido dos cabeçalhos de extensão *hop-by-hop, destination* e *routing* e a parte fragmentável, que contém os fragmentos do pacote.

Cada datagrama contendo um fragmento do pacote possui uma parte não fragmentável, como mostra a figura 11.

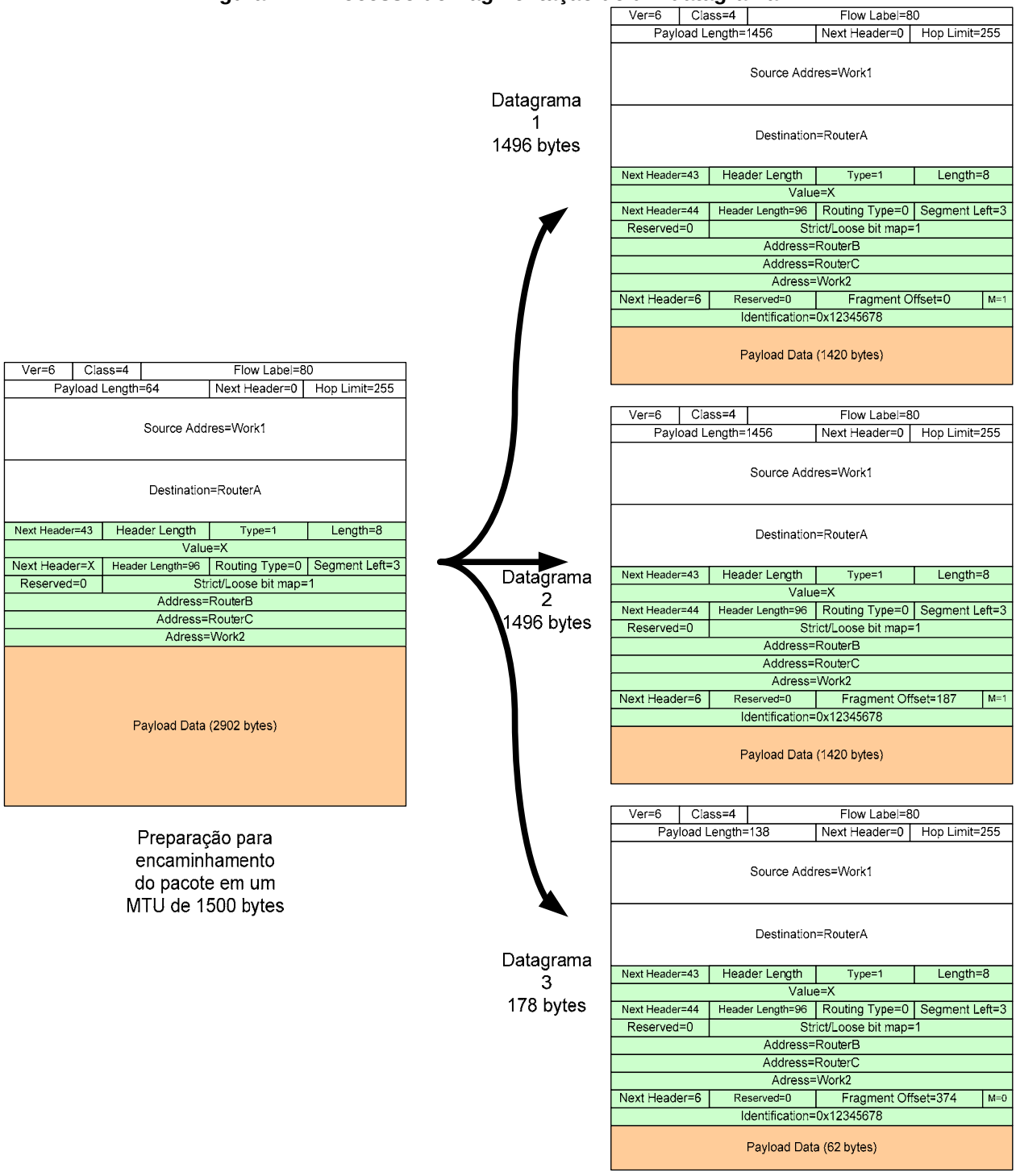
A ilustração apresenta um pacote com 2902 *Bytes* e deve passar em um enlace com MTU de 1500 *Bytes* (incluindo o cabeçalho básico e os cabeçalhos de extensão).

Observe que os datagramas devem possuir tamanhos em múltiplos de 8 *Bytes*. Sendo assim, cada fragmento poderá possuir um tamanho máximo de 1496 *Bytes* (40 *Bytes* do cabeçalho + 1456 *Bytes* de dados).

O campo FRAGMENT OFFSET indica onde começa o datagrama do fragmento que no exemplo apresenta múltiplos de 187 (pois a unidade de medida do OFFSET é palavra de 8 *Bytes*. Logo, 187*8=1496 *Bytes*).

Outra observação importante é o tamanho do *payload*. Seu valor apresenta o tamanho do datagrama sem o cabeçalho básico do IPv6 (40 *Bytes*).

Figura 11 – Processo de fragmentação de um datagrama



Fonte:STEPHEN(1996, p.118)

Para se calcular o tamanho máximo dos datagramas de fragmentação em relação ao menor MTU pelo qual esses irão passar, usa-se a seguinte equação:

$$\frac{MTU(bytes)}{8} = (\text{int}) \times 8 = datagrama(bytes)$$

$$\text{Ex.: } \frac{1500}{8} = 187 \times 8 = 1496bytes$$

Sabendo-se o tamanho dos cabeçalhos básico e de extensão em *bytes*, podemos calcular também, quanto de dados será carregado em cada datagrama. Para este fim deve utiliza-se a seguinte fórmula:

$$datagrama(bytes)\text{-}Cab.basico(bytes)\text{-}cab.ext.1(bytes)\text{-}cab.ext.N(bytes) = payload(bytes)$$

$$\text{Ex.: } 1496\text{-}40(cab.basico)\text{-}8(cab.hop\text{-}by\text{-}hop)\text{-}20(cab.routing)\text{-}8(cab.fragment) = 1420bytes$$

Já o cálculo do OFFSET é dado pela seguinte fórmula:

$$\frac{datagrama(bytes)}{8} = OFFSET$$

$$\text{Ex.: } \frac{1496}{8} = 187$$

Em cada fragmento, o campo OFFSET receberá múltiplos do valor calculado.

5.3.6. CABEÇALHO AUTHENTICATION

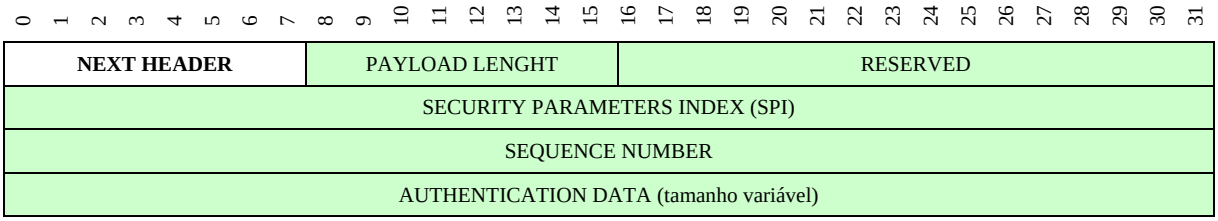
Este é o cabeçalho de extensão responsável pelo provimento de integridade e autenticação dos datagramas. Seu código de acionamento é o 51.

Um datagrama contendo um cabeçalho *Authentication Header* (AH) não pode ser fragmentado pela origem. Caso os equipamentos roteadores detectem comportamento de fragmentação em datagramas com AH, estes serão descartados.

Isso evita tentativas de ataque por “sobreposição de fragmentos”. Neste caso, os pacotes são descartados em nível de IP, o que garante uma considerável redução em “ataques de negação”.

Vejamos o diagrama do cabeçalho de autenticação:

Figura 12 – Formato do cabeçalho de autenticação



Fonte: MARK(1997, p.57)

- **PAYLOAD LENGHT** – Este campo apresenta o tamanho total do cabeçalho em palavras de 32 bits excluindo os campos *NEXT HEADER, PAYLOAD LENGHT, RESERVED* e SPI que possuem tamanho fixo. Seu valor mínimo é 1 indicando que apenas o campo *SEQUENCE NUMBER* será preenchido.
- **RESERVED** – Este campo de 16 bits é destinado para uso futuro. Ele é inicializado com zero mas está incluído no calculo dos dados de autenticação, porém é desprezado pelo destinatário.
- **SECURITY PARAMETERS INDEX** – Este campo, associado com o ENDEREÇO DE DESTINO, definem uma relação que garante segurança à comunicação e identifica diferentes *Security Associations* (SAs) para um mesmo destino. Os valores vão de 0 a 255. O 0 é utilizado para implementação local e os demais valores são reservados pela *Internet Assigned Numbers Authority* (IANA) atualmente chamada de *Internet Corporation for Assigned Names and Numbers* (ICANN), que é a entidade responsável pelo controle de endereços IP oficiais da *Internet* .

  Se utilizado o algoritmo padrão(que atualmente é o *Message Digest 5* (MD5), a autenticação dos dados consiste de 16 *bytes.* Uma chave de autenticação deve ser utilizada pelo computador que inicia a comunicação. Ela deve possuir um tamanho de 128 bits, caso isso não ocorra, vários zeros são incluídos até que a chave

atinja o tamanho mínimo requerido. Como alguns campos do datagrama, como o *hop limit,* são variáveis, esses 128 bits transportam a autenticação dentro do datagrama baseada em cálculos que desprezam esses campos. No destino, o mesmo processo é utilizado. Se os dados enviados e o cálculo realizado forem diferentes, indica que o datagrama não é autentico.

- **SEQUENCE NUMBER** – Trata-se de uma seqüência de crescimento monotônico utilizada para proteger a resposta do destinatário. Este não tem obrigação de interpretar este campo. Quando uma *Security Association* (AS) é estabelecida, no primeiro datagrama enviado esse campo é preenchido com 1 e o seu valor máximo é de $2^{32}-1$. Quando a seqüência atinge seu limite, uma nova SA deve ser iniciada e conseqüentemente uma nova seqüência no campo.
- **AUTHENTICATION DATA** – Este campo de tamanho variável e múltiplo de 32, também é conhecido por *Integrity Check Value* (ICV). Este valor de integridade é calculado na inicialização da SA e despreza os campos mutáveis durante a navegação na rede, considerando seus valores sempre como zeros.

  Vários algoritmos *Message Authentication Code* (MAC) podem ser utilizados para realização destes cálculos. A RFC 1826 recomenda a utilização do MD5, mas o HMAC-MD5-96 e o HMAC-SHA-1-96 também se aplicam.

O cabeçalho de autenticação (AH) pode ser utilizado de duas maneiras:

Modo Transporte – modo utilizado apenas pelos *hosts* e não tratado pelos roteadores, apesar de exigir menos processamento, não autentica os campos mutáveis.

Modo Túnel – utilizado sempre que se utiliza “tunelamento”. Esse conceito se aplica quando um datagrama IP transporta outro como sendo seu conteúdo. Neste caso o AH é calculado depois do “encapsulamento” de um datagrama em outro.

Figura 13 – Encapsulamento IP sobre IP

Cabeçalho IPv6	Dados

Datagrama IP original

Novo Cabeçalho IPv6	Cabeçalho IPv6	Dados

Datagrama "tunelado"

Novo Cabeçalho IPv6	Cabeçalhos de extensão	**Cabeçalho AH**	Cabeçalho IPv6	Dados

Datagrama com cabeçalho AH em modo túnel

Fonte: (TCP/IP Tutorial e Técnico,p.293)

Vale observar que no novo cabeçalho IPv6 da figura, os campos mutáveis não entram nos cálculos do AH conforme já mencionado.

A RFC 1825 que trata sobre o serviço de "tunelamento", não obriga a sua utilização, sendo este um dos motivos pelo qual algumas implementações IP *Security* (IPSec) baseadas nesta requisição, não suportam o AH em modo túnel.

5.3.7. CABEÇALHO ENCAPSULATING SECURYTY PAYLOAD (ESP)

Responsável pelos processos de criptografia, sempre deve ser configurado junto com os serviços de checagem de integridade e autenticação (AH) a fim de se evitar ataques do tipo "criptoanalítico".

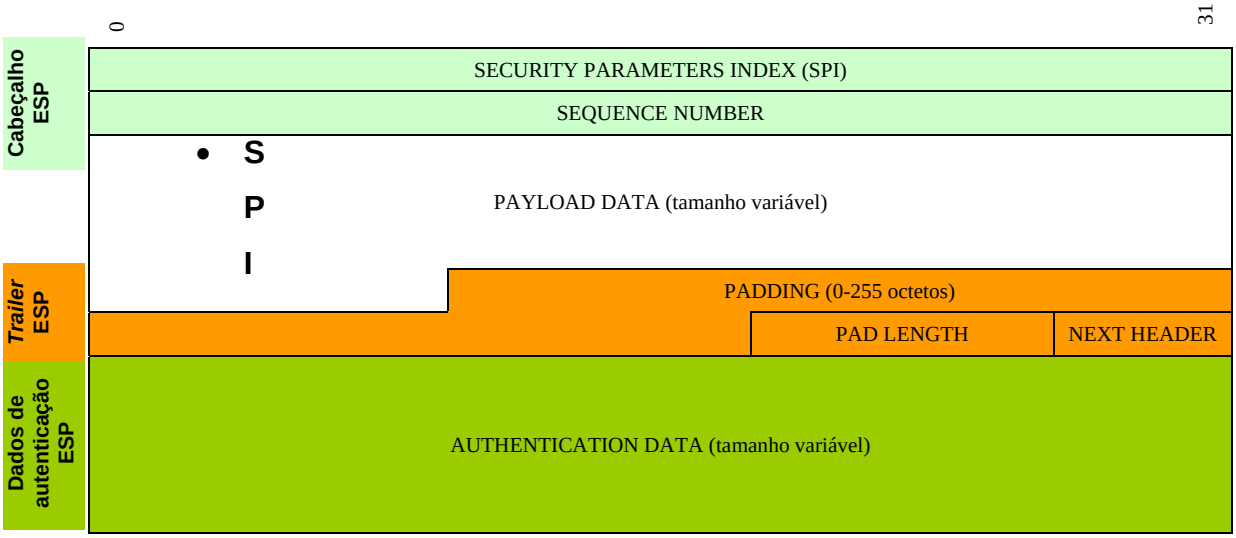
Esse cabeçalho também se aplica a datagramas encaminhados por *gateways*. Isso quer dizer que o roteador de borda irá receber o datagrama e aplicar ESP antes de enviá-lo ao próximo nó remoto. Caso este datagrama seja um fragmento, o roteador irá juntar todos os datagramas do pacote, remontá-lo e aplicar o ESP. No caso do roteador receber um datagrama com ESP implementado, este irá avaliar os campos que indicam fragmentação FRAGMENT OFFSET e M do cabeçalho de fragmentação. Caso eles apresentem valores indicativos de fragmento (OFFSET diferente de 0 e/ou M igual a 1), o datagrama é descartado afim de se evitar o "ataque de duplicação de pacotes".

A "descriptografia" do pacote é realizada pelo último *gateway* antes do destinatário. Quando o "nó de borda" recebe este datagrama autentica-o e verifica

sua integridade. Se estas informações forem pertinentes, então a “descodificação” dos dados é realizada e o datagrama segue para o seu destino. Isso diminui o processamento de dados não válidos e evita o “ataque de negação”.

Vejamos o cabeçalho ESP:

Figura 14 – Formato do cabeçalho criptografia



Fonte: MARK(1997, p.59)

- **SPI** – Segue a mesma definição do campo SPI do cabeçalho de autenticação (item 5.3.6)
- **SEQUENCE NUMBER** – Segue a mesma definição do item 5.3.7, mas uma observação deve ser acrescentada: Como nas especificações originais do ESP, o conceito de número seqüencial não é tratado, as implementações IPSec mais antigas podem não garantir proteção de resposta do destinatário.
- **PAYLOAD DATA** – Este campo de tamanho variável é obrigatório e seu conteúdo consiste em dados provenientes do cabeçalho seguinte chamado pelo campo NEXT HEADER. Os algoritmos DES-DBC, Triplo-DES e o CDMF da IBM são suportados para criptografia.
- **PADDING** – Em razão da maioria dos algoritmos de criptografia terem como requisito um número inteiro como valor do bloco, este campo irá fornecer bits adicionais para a preenchimento. É um

campo opcional uma vez que somente será utilizado caso haja a necessidade de sua finalidade.

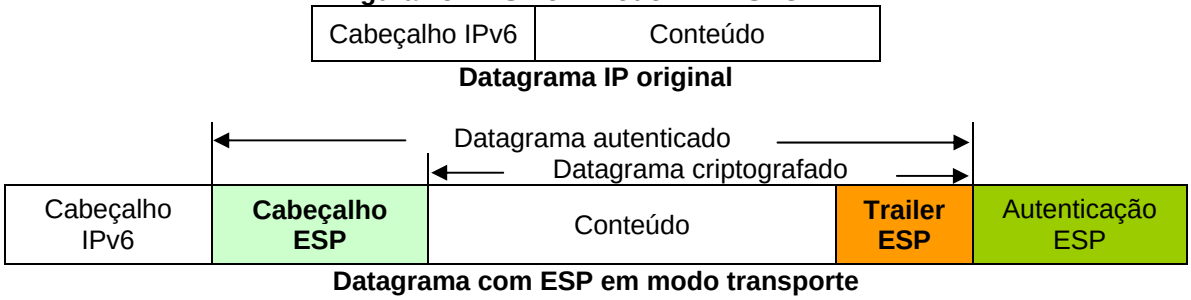
Vale lembrar que a criptografia abrange os campos PAYLOAD DATA, PADDING, PAD LENGTH e NEXT HEADER.

- **PAD LENGTH** – Este campo possui oito bits de tamanho e indica quantos *bytes* foram utilizados para preenchimento. Caso não se tenha utilizado o campo PADDIN, seu valor será 0.
- **NEXT HEADER** – Como o cabeçalho de criptografia é o último antes do protocolo de alto nível (TCP ou UDP por exemplo), este campo irá fornecer um valor que não será de cabeçalho de serviço e sim um dos protocolos IP definidos pelo IANA. (ver tabela 3).
- **AUTHENTICATION DATA** – Este campo contém o Valor de Verificação de Integridade (ICV) que é calculado para o pacote ESP utilizando os campos SPI, SEQUENCE NUMBER, PAYLOAD DATA, PADDING, PAD LENGTH e NEXT HEADER. Este campo somente será utilizado quando a checagem de integridade for solicitada ou quando uma AS for inicializada.

Assim como o AH, o ESP pode ser utilizado em **Modo Transporte** e **Modo Túnel**:

- **Modo Transporte** – O cabeçalho ESP é inserido logo após o cabeçalho IP original. Se o datagrama possuir cabeçalho IPSec, o ESP é inserido antes. O *trailer* e a Autenticação ESP são inseridos ao final do conteúdo conforme pode ser observado no esquema seguinte.

Figura 15 – ESP em modo TRANSPORTE



Datagrama com ESP em modo transporte

Fonte: TCP/IP Tutorial e Técnico,(2000, p.296)

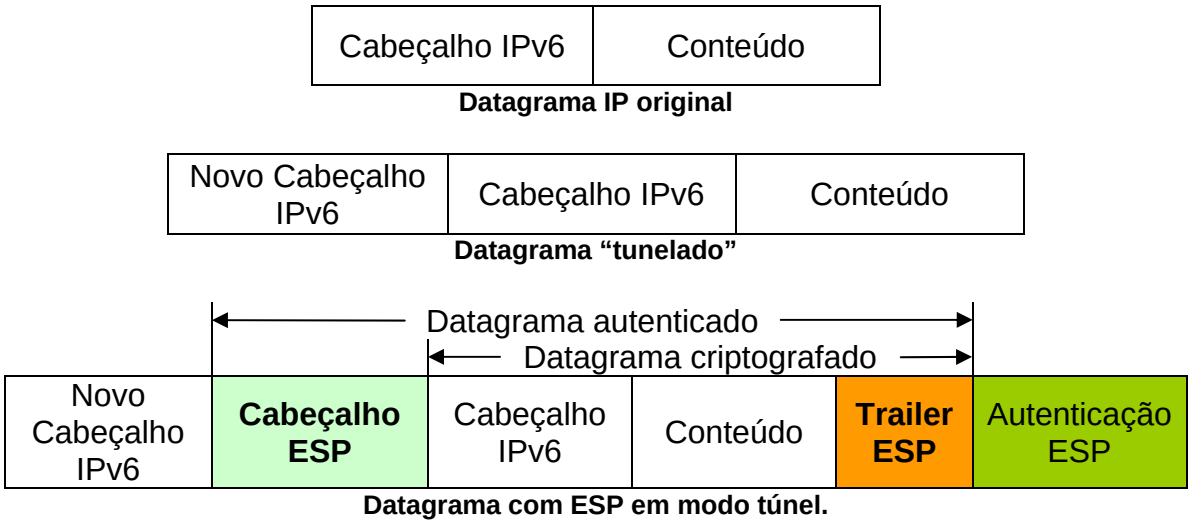
O modo Transporte não prevê encriptação para o cabeçalho IP original, o que torna possível a entrega de pacotes falsos, no entanto, reduz a necessidade de processamento. Sua implementação é feita pelos *hosts* e não pelos *gateways*.

- **Modo Túnel** – Neste modo, assim como na autenticação, a intenção é encapsular um datagrama IP dentro de outro, desta vez, utilizando um processo de criptografia para o datagrama encapsulado, pois o cabeçalho IP do datagrama "encapsulador" continua desprotegido.

  Sempre que uma associação de segurança for requerida entre dois *gateways*, este modo será ativado. Ao contrário do modo transporte, essa implementação se dá apenas em roteadores de fronteira.

Vejamos um esquema de ESP em modo túnel:

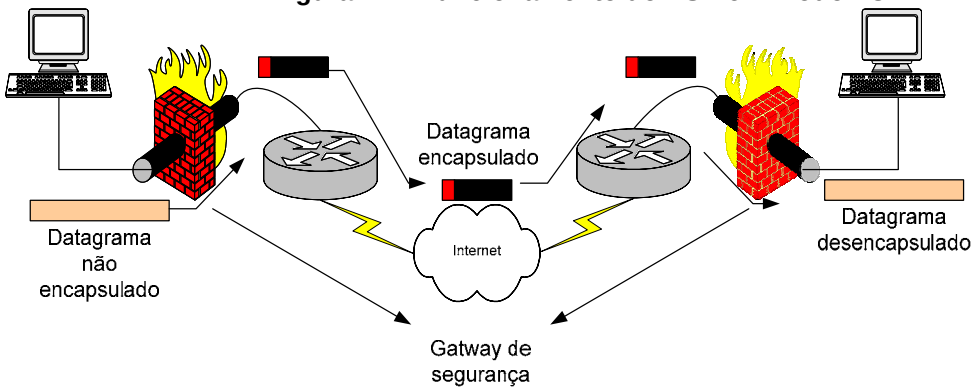
Figura 16 – ESP em modo TÚNEL



Fonte: TCP/IP Tutorial e Técnico (2000, p.297)

Na figura seguinte, podemos verificar uma ilustração de onde o cabeçalho de criptografia atua em modo túnel:

Figura 17 – Funcionamento do ESP em modo TÚNEL



Fonte: STEPHEN (1996, p.124)

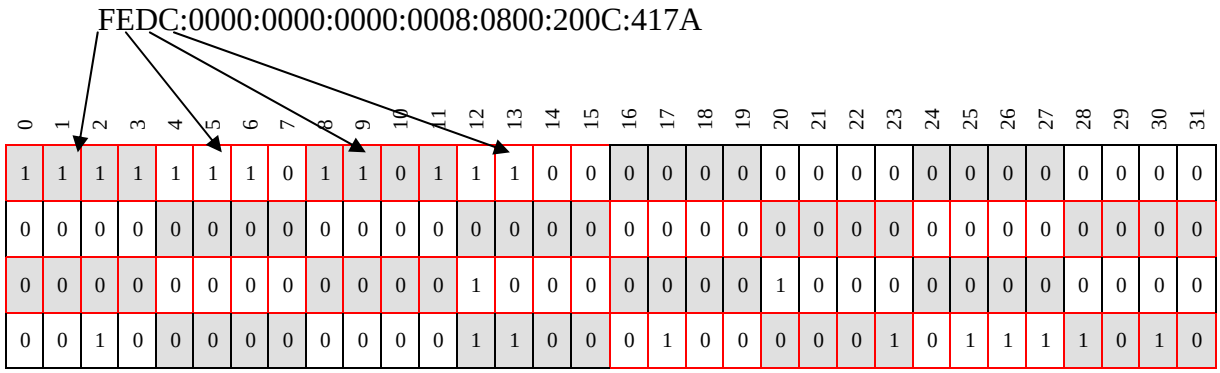
Observe que o *gateway* de segurança cria um novo datagrama e coloca o original em seu campo de dados de modo que este último não pode ser visualizado. Ao chegar no *gateway* de destino, ele desempacota o datagrama original e o lança ao seu destino.

5.3.8. Endereçamento IPv6

Como dito no início deste trabalho, “o IPv6 eleva substancialmente a quantidade de endereços válidos para a Internet , bem como procura reduzir a utilização de banda de rede com redução de informações no cabeçalho de dados”.

Endereçamento IPv6, assim como no IPv4, identifica os nós de uma rede. A grande diferença do novo sistema é que passamos a usar 128*bits* com notação Hexadecimal, contendo em oito grupos de dois *Bytes* e separados por dois pontos (:) como mostra a representação a seguir.

Figura 18 – Endereço IPv6 representado em bits



Fonte: própria

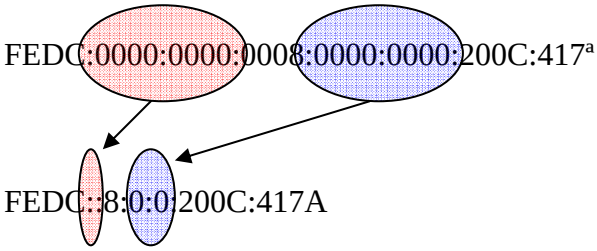
A fim de melhorar a escrita é permitida a simplificação da notação da seguinte maneira: onde houver grupos de zeros, apenas um deles é necessário ser escrito e, os zeros à esquerda de grupos com outros valores, não necessitam ser representados. Veja a representação do endereço acima de forma mais curta:

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

Uma forma mais simplificada de se escrever uma notação de endereço IPv6, é a utilização de um par de “ : “ para representar grupos de zeros consecutivos, conforme apresentado na seqüência:

FEDC::8:800:200C:417A

Cabe ressaltar que somente uma supressão de zeros por “::“ é permitida. Caso apareçam duas seqüências de zeros, apenas uma deverá receber esta representação. A outra, será representada por um 0 normalmente conforme podemos verificar abaixo.



A rede baseada em IPv6 também pode ser subdividida em subredes utilizando notação de barra comum (/) como no IPv4. Alguns prefixos de endereçamento já foram estabelecidos e não devem ser utilizados para endereçamentos comuns. Veja a tabela seguinte:

Tabela 5 – Alocação de prefixos

Alocação	Prefixos binários	Inicio da faixa de endereçamentos em Hexadecimal	Extensão da Máscara em bits	Fração do espaço de endereços geral
Reservado	0000 0000	0:: /8	8	$\frac{1}{256}$
Reservado para NSAP	0000 001	200:: /7	7	$\frac{1}{128}$
Reservado para IPX	0000 010	400:: /7	7	$\frac{1}{128}$
Endereços *Unicast* Globais Agregados	001	2000:: /3	3	$\frac{1}{8}$
Unicast local ao enlace	1111 1110 10	FE80:: /10	10	$\frac{1}{1024}$
Unicast local à instalação	1111 1110 11	FEC0:: /10	10	$\frac{1}{1024}$
Multicast	1111 1111	FF00:: /8	8	$\frac{1}{256}$

Fonte: TCP/IP Tutorial e Técnico (2000, p. 357) e MARK(1997, p.86)

A notação de endereçamento do IPv6 permitiu aos órgãos de controle designar um prefixo para cada país que por sua vez poderá criar sub-prefixos para cara região ou estado.

Vejamos então alguns endereços de uso fixo:

- **Endereço *Unicast*** – Designado para ser entregue a uma única interface. Existem alguns endereços *unicast* para propósitos especiais como segue:
    - ***Loopback*** – Representado pela notação ::1, segue as mesmas regras do IPv4 (127.0.0.1), ou seja, é um endereço virtual que funciona apenas para o próprio *host*.

- ***Não especificado*** – Representado por ::, é utilizado para autoconfiguração do *host* (*Dynamic Host Configuration Protocol* - DHCP por exemplo).
- ***Compatibilidade com IPv4*** – Escrito em notação hexadecimal, separa os octetos do IPv4 em valores hexa como por exemplo: o endereço 10.70.4.10 seria representado por ::0A46:040A. Isso quer dizer que foram utilizados os 32 bits do IPv4 originais e acrescentado 96 zeros à esquerda do endereço.

  Isto é utilizado quando se necessita encaminhar um datagrama de uma rede IPv6 para outra utilizando tunelamento em redes IPv4.
- **IPv4 mapeado** – Utilizado para comunicações entre *hosts* que utilizam versões diferentes do IP. Se um equipamento com endereçamento IPv6 qualquer necessitasse estabelecer conexão com outro IPv4 cujo endereço fosse 10.70.4.10, a notação do endereço de destino ficaria ::FFFF:0A46:040A. Só que para a comunicação funcionar, seria necessário um *gateway* para tradução dos cabeçalhos.
- **Local ao enlace** – São endereços para uso em redes que não se conectam a nenhum outro tipo de rede. Destinado a locais isolados e desprovidos de qualquer serviço de roteamento. Seu prefixo é FE80:: e terminam com o endereço da interface contendo 64 bits.
- **Local à instalação** – Assim como no IPv4, alguns endereços são restrito às corporações, como os 192, 10, 168 e 176, e não podem ser roteados para a *Internet* . Eles possuem uma divisão diferenciada conforme esquema a seguir:

FEC0::(subnet):(endereço da interface)

Onde a subnet possui 16 bits e o endereço da interface, 64 bits.

5.3.8.1. Formato de Endereço Unicast Global

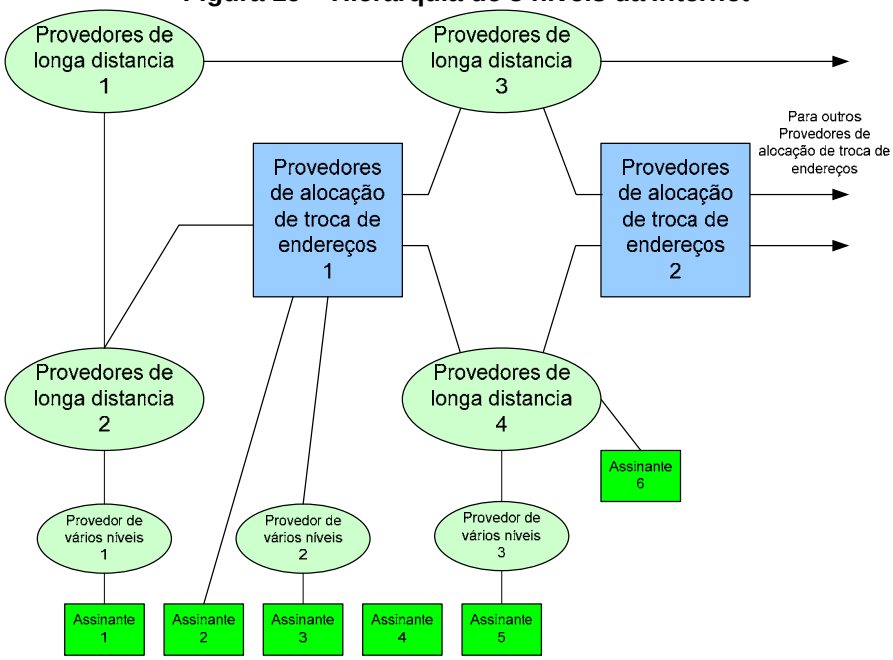
Este formato de endereço está relacionado com o modelo de *Internet* de 3 níveis:

- **Topologia Pública** – Provedores que oferecem serviços de transito público compondo a *Internet .*
- **Topologia de Instalação Local** – Fechado às localidades de uma corporação, mas não provê trânsito público.
- **Identificadores de Interfaces** – Identificam de um modo geral, as interfaces em conexões.

Os provedores de longa distância são responsáveis pelos grandes *backbones* de interligação. Já os provedores de vários níveis, têm por objetivo distribuir conexões para provedores finais e para corporações. Os assinantes por sua vez, são os usuários ou empresas que utilizam os serviços dos provedores.

Veja um esquema que representa a *Internet* em 3 níveis:

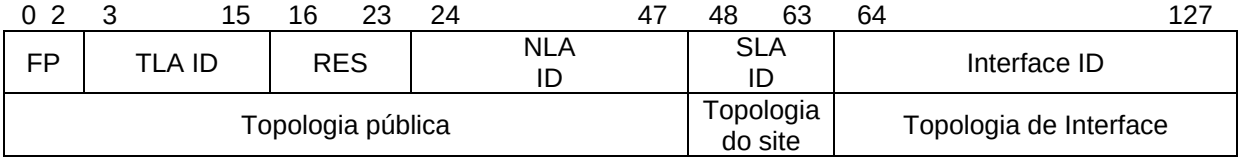
Figura 19 – Hierarquia de 3 níveis da *Internet*



Fonte: TCP/IP Tutorial e Técnico (2000, p. 358) e MARK(1997, p.94)

A próxima ilustração apresenta o formado do endereço unicast global.

Figura 20 – Formato do endereço *Unicast*



Fonte: TCP/IP Tutorial e Técnico(2000, p. 359) e MARK(1997, p.95)

- **FP** (*Format Prefix*) – Prefixo do Formato *Unicast*. Para este formato, seu valor é sempre 001, conforme a tabela 5.
- **TLA ID** (*Top-Level Aggregation Identifier*) – São os níveis superiores na hierarquia de roteamentos, onde cada roteador necessitará de uma entrada TLA ID na sua tabela de roteamento.
- **RES** – Reservado para uso futuro.
- **NLA ID** (*Next Level Aggregation Identifier*) – Utilizado para que as organizações possam criar uma hierarquia própria. Essas organizações não utilizam provedoras e podem designar um TLA ID próprio.
- **SLA ID** (*Site Level Aggregation Identifier*) – Este campo permite a criação de hierarquia local para a organização, podendo se definir até 65.535 subredes na corporação.
- **INTERFACE ID** (*Interface Identifier*) – Identifica qual o tipo de interface para o *link* de comunicação(*Ethernet, Tolken Ring, etc*).

5.3.8.2. Formato de Endereço *Anycast*

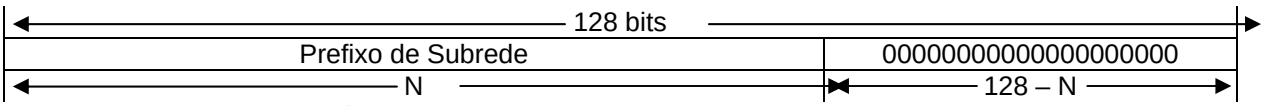
Trata-se de um endereço preparado para comunicação com múltiplas interfaces, normalmente em nós diferentes desde que estejam bem próximos. A distancia pode ser definida pelo protocolo de roteamento.

A RFC 1884 define aplicações de uso para o endereço *anycast* no que se refere a identificar grupos de roteadores que pertençam ao mesmo provedor de serviço de *Internet* , na mesma sub-rede e dentro do mesmo domínio de roteamento.

Endereços *anycast* só podem ser designados por roteadores e não devem ser utilizados como de origem de pacotes IPv6.

Este endereço se inicia com a parte referente à sub-rede de tamanho variável (pois depende da máscara aplicada) seguida da parte referente a *host* preenchida com zeros. A intenção é utilizá-lo para a comunicação de um nó com membros de grupos de sub-redes remotas.

Figura 21 – Formato do endereço *Anycast*



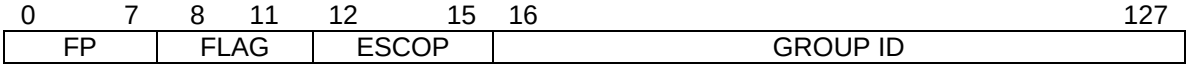
Fonte: TCP/IP Tutorial e Técnico (2000, p. 359) e MARK (1997, p.95)

5.3.8.3. Formato de Endereço *Multicast*

Este endereço é designado para se identificar um conjunto de *hosts* cujos pacotes enviados para o endereço *multicast* serão recebidos por todos de um determinado grupo.

A próxima figura apresenta o formato do cabeçalho *multicast*:

Figura 22 – Formato do endereço *Multicast*



Fonte:TCP/IP Tutorial e Técnico (2000, p. 359) e MARK (1997, p.95)

- **FP** – O prefixo do formato para *multicast* é sempre 1111 1111.
- **FLAG** – Pode apresentar dois valores:
  - Se 0000, o endereço é designado permanentemente por uma autoridade de numeração.
  - Se 0001, o endereço pode ser estabelecido por aplicativos conforme a necessidade. Ao término da utilização, o endereço é liberado para ser reutilizado por outro sistema.
- **ESCOP** – Composto de 4 bits, indica o escopo do *multicast* conforme a tabela seguinte.

Tabela 6 – Valores possíveis para o campo ESCOP

Valor de escopo	Descrição
0	Reservado
1	Restrito ao nó local
2	Restrito a nós no enlace local
3	Não utilizado
4	Não utilizado
5	Restrito ao *site* local
6	Não utilizado
7	Não utilizado
8	Restrito à organização
9	Não utilizado
A	Não utilizado
B	Não utilizado
C	Não utilizado
D	Não utilizado
E	Escopo Global
F	Reservado

Fonte: TCP/IP Tutorial e Técnico (2000, p. 360) e MARK (1997, p.102)

- **GROUP ID** – Valor que identifica o grupo de *multicast*. Alguns endereços deste grupo possuem uma função predefinida com objetivos especiais e não devem ser utilizados, conforme apresentados na tabela 7. As tabelas 8, 9 e 10 apresentam os valores de uso comum.
-

Tabela 7 – Valores *Multicast* reservados

Endereços *multicast* reservados e que não devem ser utilizados
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

Fonte: MARK (1997, p.103)

Tabela 8 – Valores *Multicast* para todos os nós (escopos 1 e 2)

Endereços *multicast All nodes*	Descrição
FF01:0:0:0:0:0:0:1	Todas a interfaces locais ao nó
FF02:0:0:0:0:0:0:1	Todas a interfaces locais ao enlace

Fonte: MARK (1997, p.103)

Tabela 9 – Valores *Multicast* para todos os nós (escopos 1, 2 ou 5)

Endereços *multicast All nodes*	Descrição
FF01:0:0:0:0:0:0:2	Todos os roteadores locais ao nó
FF02:0:0:0:0:0:0:2	Todos os roteadores locais ao enlace
FF05:0:0:0:0:0:0:2	Todos os roteadores locais ao *site*

Fonte: MARK (1997, p.104)

Tabela 10 – Valores Multicast com outras funções

Endereços *multicast All nodes*	Descrição
FF02:0:0:0:0:0:0:B	Agentes móveis locais ao enlace
FF02:0:0:0:0:0:1:2	Todos os agentes DHCP locais ao enlace
FF05:0:0:0:0:0:1:3	Todos os servidores DHCP locais à instalação

Fonte: MARK (1997, p.103)

5.3.9. IEEE EUI-64

O *Institute of Eletrical and Eletronics Engineers* (IEEE) está trabalhando no sentido de ampliar o tamanho do endereço físico das interfaces de rede (endereços MAC). Segundo Mark A. Miller, em seu livro "*Implementing* IPv6, *Migrating to the next generation Internet protocols*", a proposta é ampliar o atual endereço MAC de 48 para 64 bits de tamanho inserindo 2 *Bytes* entre o prefixo que representa a identificação da companhia que fabrica o *hardware* e a extensão. Isso elevaria a capacidade de endereçar *hardwares* de rede em aproximadamente um trilhão de endereços. Uma das idéias seria utilizar este novo endereçamento incorporando-o no campo INTERFACE ID do endereço IPv6 a fim de facilitar a implementação em redes pequenas e isoladas.

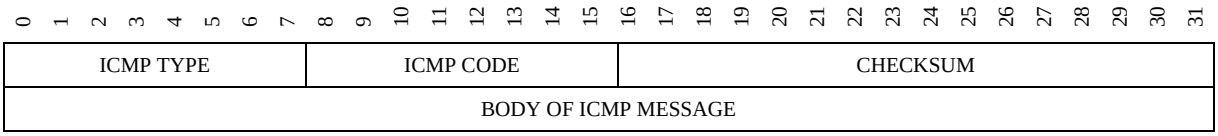
5.3.10. INTERNET CONTROL MESSAGE PROTOCOL VERSION 6 (ICMPv6)

O protocolo ICMP foi desenvolvido para transporte de diagnósticos sobre a transação de datagramas pela rede. Ele é responsável pelas funções de relatar erros de entrega de datagramas, atualizar tabelas de rotas entre outras.

Com a versão 6 do protocolo, as funções do IGMP e do ARP foram incorporadas a ele, o que o tornou muito mais robusto.

O cabeçalho de extensão ICMPv6 é acionado pelo número 58 conforme tabela 3. Seu formato se apresenta na próxima figura:

Figura 23 – Formato do cabeçalho ICMPv6



Fonte: STEPHEN (1996, p.128)

- **ICMP TYPES** – Este campo é provido da parte mais pesada da identificação. Ele é composto por quinze valores distintos que divide as mensagens em duas classes bem definidas:
  - A primeira classe com os 127 primeiros valores que apresentam as mensagens de erro.
  - E a segunda onde estão presentes os valores a partir de 128 que apresentam mensagens de informação.

A tabela 11 apresenta alguns valores de mensagens ICMP

Tabela 11 – Valores e mensagens ICMPv6

Valor	Mensagem
1	Destination Unreachable error message (destino não encontrado)
2	Packet Too Big error message (pacote muito grande para o canal)
3	Time Exceeded error message (tempo de resposta excedido)
4	Parameter Problem error message (problemas com os parâmetros)
128	Echo Request message (solicitação de resposta)
129	Echo Replay message (resposta à solicitação)
130	Group Membership query (associação de um membro a um grupo)
131	Group Membership report (informações sobre o membro de um grupo)
132	Group Membership termination (saída do membro do grupo)
133	Router Solicitation (solicitação de roteamento)
134	Router Advertisement (aviso de roteamento)
135	Neighbor Solicitation (busca de vizinhos)
136	Neighbor Advertisement (anuncio de vizinho)
137	Redirect message (aviso de redirecionamento)

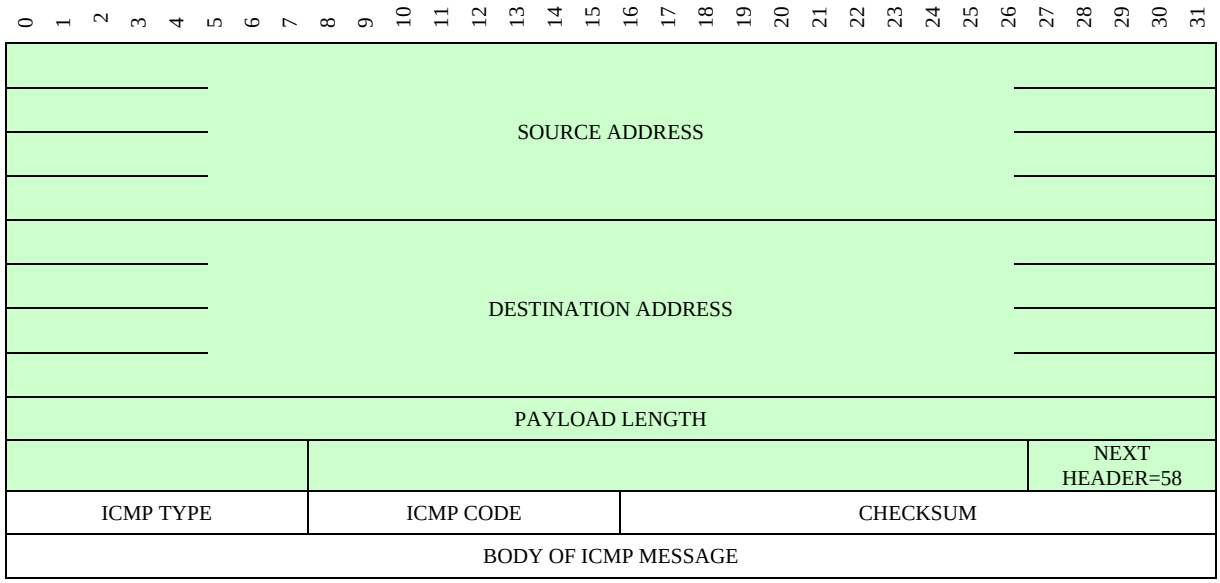
Fonte: STEPHEN (1996, p.129)

- **CHECKSUM** – Destinado a proteger as mensagens ICMP contra corrupção. Para atender a esta demanda, é utilizado o TCP como nível de transporte.

  Antes de se enviar o datagrama, o *checksum* é calculado e inserido neste campo. Se o valor resultante não for múltiplo de 8 *bits*, um *byte* extra é inserido imaginariamente no cálculo deste campo porém não é enviado com o pacote.
- **BODY OF ICMP MESSAGE** – Trata-se do corpo da mensagem ICMP no qual serão levados os dados de informações ou erros.

Alguns sistemas utilizam-se de um pseudo cabeçalho (*pseudo header*) para envido de suas mensagens. Este é composto de Endereço de Origem e Destino, Tamanho do Conteúdo (*payload*) e o Próximo Cabeçalho já setado para o ICMP (valor 58) e localiza-se entre o último cabeçalho de extensão e o ICMP conforme podemos ver na figura seguinte.

Figura 24 – Formato do pseudo cabeçalho ICMPv6



Fonte: STEPHEN (2000,p.130)

Na criação deste pseudo cabeçalho, o Endereço de Destino deve ser preenchido com o valor que ele deverá conter quando lá chegar. Por este motivo, se estiver utilizando um Cabeçalho de Roteamento, este endereço será diferente do original quando estiver no destino.

Quando o datagrama é recebido pelo destino, um cálculo reverso do *checksum* é realizado. Se o resultado for 0xFFFF, o datagrama será aceito. Caso contrário será invalidado o *checksum* e conseqüentemente o datagrama.

5.3.11. NEIGHBOR DISCOVERY PROTOCOL (NDP)

Esta é uma das grandes funcionalidades do novo ICMP. Ela substitui o protocolo de resolução de endereços (ARP), o ICMP de descoberta de roteador do protocolo de mensagens e o redirecionamento ICMP, que são usados no IPv4. Isso possibilita aos nós da rede determinarem os MTUs nos quais seus vizinhos que executam tarefas de roteamento ou redirecionamento estão conectados, configurarem seus endereços automaticamente (DHCPv6), descobrir quem são os *gateways* de sua rede local, determinar o melhor caminho para envio de seus datagramas a partir de informações dos roteadores vizinhos, entre outras.

Combinando essas funcionalidades, temos um eficiente disseminador de informações na rede.

Cinco tipos de mensagens ICMPv6 foram criados pra facilitar o trabalho do NDP:

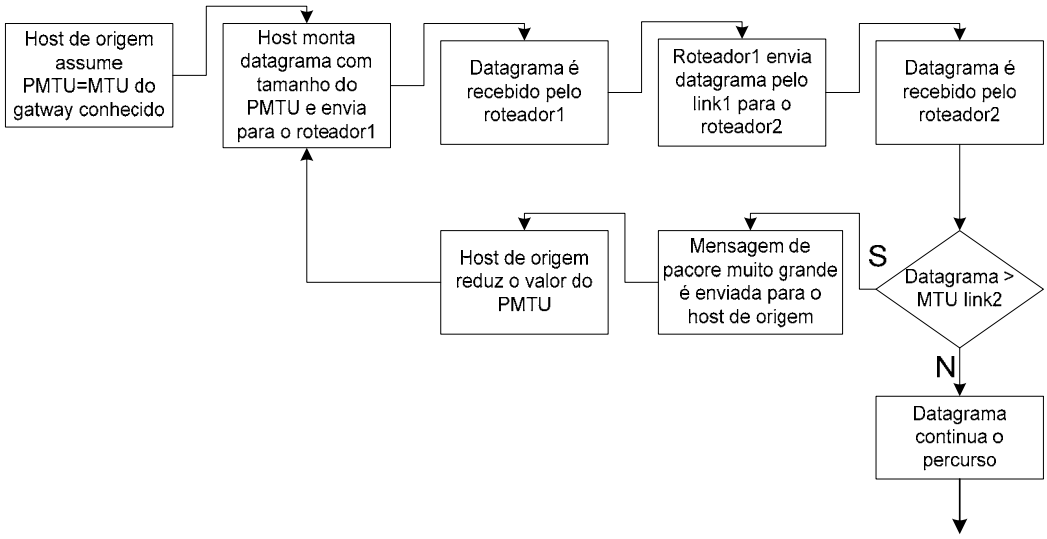
- **Mensagens de Solicitação de Roteador (*Router Solicitation message*)** – Essa solicitação é enviada pelo *host* a fim de solicitar aos roteadores que enviem informações sobre suas rotas. Isso o auxiliará da definição de um *gateway* padrão.
- **Mensagens de Anúncios de Rota (*Router Advertisement message*) –** Tem como objetivo a divulgação de informações sobre rotas dos roteadores. A disseminação destes dados é feita periodicamente ou quando há uma **Mensagem de Solicitação de Roteador**.
- **Mensagem de Solicitação de Vizinhança (*Neighbor Solicitation message*)** – Solicitado pelos nós de origem via *multicast*, requer informações dos nós vizinhos sobre endereçamentos de portas de saída da rede local, os quais são repassados adiante para outros nós. Para se saber a distância a que estão destes vizinhos, utiliza-se mensagens do tipo *unicast*.
- **Mensagem de Anúncio de Vizinhança (*Neighbor Advertisement message*)** – É a resposta à **Mensagem de Solicitação de Vizinhança** enviada ao nó solicitante.
- **Mensagens de Redirecionamento** – O roteador utiliza-se destas mensagem pra informar aos *hosts* os melhores caminhos pelos quais eles deverão enviar seus datagramas a fim de atingirem seus destinos.

5.3.11.1. Processo de descoberta dos MTUs dos caminhos (*Path* MTU *Discovery Process*)

No IPv6, a responsabilidade pela fragmentação do pacote de dados é do *host* e não mais do roteador. Sendo assim, o *host* necessita saber qual o valor do menor MTU por onde cada fragmento de seu pacote irá passar. Desta maneira, o *host* pode identificar qual o melhor canal de dados e o MTU mais adequado para o enviá-los com mais eficiência.

O processo se baseia em tentativa e erro. Em uma primeira comunicação, um nó assume o MTU do roteador que ele já conhece como sendo o melhor caminho para enviar um datagrama para um *host* que está em uma rede que ele não conhece. Quando ele envia o datagrama, este roteador vai recebê-lo e encaminhar para o próximo nó que ele conhece. Se o próximo canal de comunicação possuir um MTU igual ou superior, o datagrama segue o curso. Caso esta unidade não suporte o datagrama, uma mensagem de “pacote muito grande” (Packt too Big), é enviada ao host de origem. Com isso, este equipamento irá alimentar sua tabela de rotas com as informações aprendidas e re-fragmentar o pacote. Este procedimento se repete até que o datagrama chegue ao seu destino. Uma vez passado o primeiro pacote, o host já terá conhecimento de que, para atingir a rede especifica, o tamanho do datagrama deverá ser sempre aquele. Veja o fluxo:

Figura 25 – Processo de reconhecimento do MTU dos links



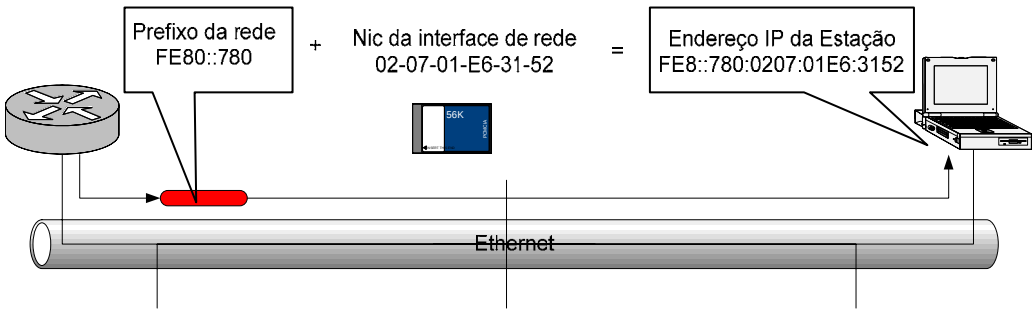
Fonte: MARK (1997, p.153)

5.3.11.2. Autoconfiguração de endereços

No IPv6, possuem duas maneiras de se alocar endereços automaticamente:

- **Com estado** – um *host* obtém seu endereço IPv6, informações e parâmetros de configuração a partir de um servidor DHCPv6 (ver item 5.3.11.3);
- **Sem estado –** trata-se de uma inovação do IPv6 para sites que não necessitam conhecer seus endereços IP. Para tal são utilizados os endereços MAC dos adaptadores de rede associados às informações fornecidas pelos roteadores. Os prefixos IPv6 para este tipo de endereçamento devem começar com 1111 1110 10 conforme tabela 5. Para este tipo de endereçamento, as chances de se encontrar um endereço duplo, quase não existem, uma vez que os NICs, teoricamente, são únicos no mundo. Porém, caso aconteça de se encontrar uma placa com mesmo endereço físico de outra na mesma rede, ou se substitui a interface, ou se força um endereço manualmente. Veja a ilustração:

Figura 26 – Processo de auto configuração de endereço “sem estado”



Fonte: STEPHEN(1996, p.147)

5.3.11.3. DHCPv6

Assim como na versão 4 do protocolo IP, sua função é distribuir endereços IP e parâmetros de configuração para os *hosts* de uma rede. Porém algumas inovações foram incluídas. Vejamos as diferenças:

- As estações configuradas para funcionarem em redes IPv6 com DHCP utilizam chamadas *multicast* (ao invés de *broadcast* como no IPv4) para localizar servidores do serviço na rede;
- **Não é mais necessária a configuração de *gateway default* no servidor DHCP, pois as estações utilizam o serviço de "busca de vizinhos" (*neighbor discovery*);**
- O servidor pode encaminhar mensagens de reconfiguração dos endereços distribuídos, sem a necessidade de aguardar uma nova solicitação por parte da estação.

O serviço DHCPv6 é baseado em arquitetura Cliente/Servidor e possui 4 tipos funcionais:

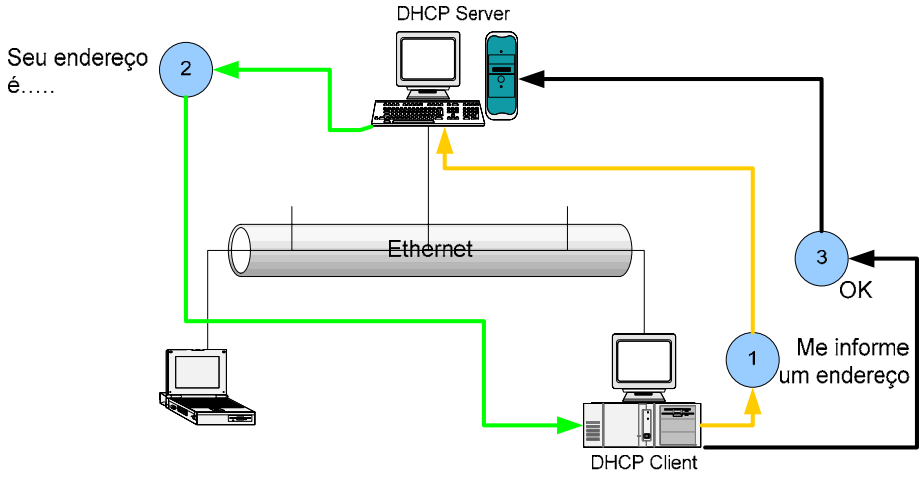
- ***Client* –** o nó que requisita e obtém os parâmetros de configuração;
- ***Server* –** o nó que responde às requisições fornecendo o endereço ou o prefixo (para o caso de alocação de endereço sem estado), e outros parâmetros de configuração (que podem ser implementados em um modo misto – parte com e parte sem estado);
- ***Relay* –** o nó que serve de intermédio de entrega de mensagens DHCPv6 entre clientes e servidores;
- ***Agent* –** nó que pode atuar como cliente ou servidor.

A comunicação DHCPv6 pode ocorrer de 3 modos diferentes dependendo dos tipos funcionais envolvidos:

1. *Interação entre Clientes e servidores de modo direto* – Neste caso, o nome do *host* não necessita ser conhecido. Esta

comunicação se dá em três níveis como mostra a figura que se segue:

Figura 27 – Processo fornecimento de endereço cliente/servidor



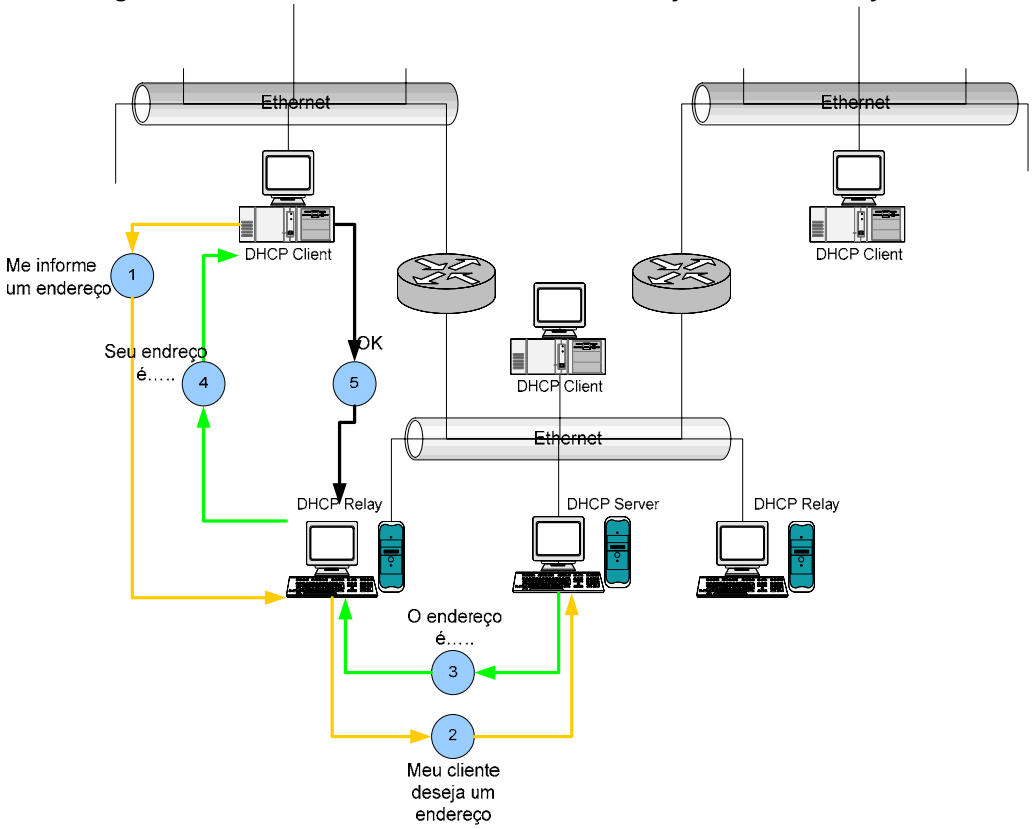
Fonte: Própria

2. *Utilizando um agente Relay* – Neste caso, um agente de interação entre o cliente e o servidor intermedeia a negociação do endereço. Este agente nunca responde diretamente a uma requisição do cliente. Isso se aplica quando temos duas redes físicas distintas ligadas a uma terceira e não temos a intenção de colocar dois servidores DHCP para disponibilizar endereços conforme exemplificado na figura 28.

3. *Utilização do serviço DHCP associado com o serviço de resolução de nomes (DNS-Domain Name Server)* – Neste caso, quando o *host* conhece seu nome de domínio, ele pode solicitar ao servidor DHCP um endereço a partir de seu nome. Este servidor irá realizar uma consulta ao servidor DNS para saber se seu nome já esta associado a algum IP. Em caso positivo, o servidor DHCP verifica a alocação do endereço e o repassa ao cliente. Caso o DNS desconheça o endereço do nome, o DHCP reserva um endereço dinâmico para o cliente, informa essa locação ao servidor DNS e em

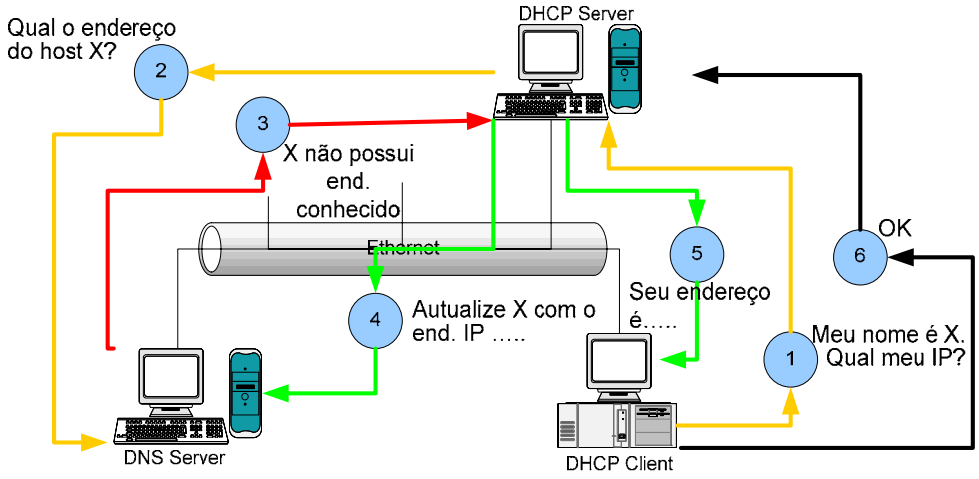
seguida o repassa ao cliente. (funcionamento similar ao WINS).

Figura 28 – Processo fornecimento de endereço com a utilização do DHCP *Relay*



Fonte: Própria

Figura 29 – Processo fornecimento de endereço combinando DHCP e DNS



Fonte: Própria

5.4. COMPATIBILIDADE ENTRE IPv4 E IPv6

Em virtude da semelhança entre as duas versões do protocolo IP, há a possibilidade de que elas funcionem em conjunto. As preocupações não se prendem somente ao formato do endereço, mas a compatibilidade entre os , a qual deve ser igualmente ajustada.

Em termos de endereço, o IPng define dois modos para compatibilizar o IPv6 com o IPv4:

- **IPv4 compatível** – Trata-se do processo de conversão de endereços propriamente dito - IPv4 para IPv6 e vice versa. São utilizados para comunicação entre duas redes IPv6 com a utilização de uma rede IPv4 como caminho de passagem sem a utilização de “tunelamento”. O processo de conversão de endereços se dá com a inclusão de um endereço IPv4 como, por exemplo, 10.70.4.5 com os 96 bits que faltam para 128bits (tamanho padrão do IPv6) inseridos à esquerda “setados” em zero da seguinte maneira: 0:0:0:0:0:0:0A:46:04:05 ou simplesmente ::0A:46:04:05. Deste modo, quando o roteador que receber o datagrama, retirar os 96 zeros, ele entrega um pacote convertido para IPv4 na rede compatível.
- **IPv4 mapeado** – Este endereçamento foi desenvolvido para permitir a comunicação entre um *host* puramente IPv6 com outro puramente IPv4. Seu formato consiste de 80 bits setados em zero, 16 bits em um e o endereço IPv4 do destino. A apresentação deste endereço ficaria da seguinte maneira: 0:0:0:0:0:FFFF:0A46:0405.

Apesar da semelhança entre os resultados, o complemento 0:0:0:0:0:FFFF é o parâmetro que indica que o endereço é mapeado, e não compatível.

Quanto aos cabeçalhos, a maioria dos campos se mapeia diretamente e o que não é possível faz-se um processo de conversão. A tabela a seguir apresenta uma lista de compatibilidades entre os cabeçalhos:

Tabela 12 – Processo fornecimento de endereço combinando DHCP e DNS

Protocolo	Campo	Compatibilidades
IPv4	*HLength*	Calculado conforme as opções existentes no cabeçalho do IPv6
IPv6	*Class(valor > 7)*	O IPv6 recebe os datagramas IPv4 com prioridade 0
IPv4	*Type Of Service*	Este campo é ignorado pelo IPv6 durante a conversão.
IPv4	*Flags e OffSet Fragment*	Quando recebe um datagrama com cabeçalho de fragmentação, os 16 bits menos significativos do cabeçalho de extensão de fragmentação referente às informações do serviço entram como valores deste campo
IPv6	*Fragment OffSet e M*	Recebem os valores do datagrama IPv4 normalmente
IPv6	*Flow Labels*	Ignorado quando da conversão para IPv4 e zerado quando do inverso

Fonte: STEPHEN (1996, p.428)

Ainda há a possibilidade de se ter um host com pilha dupla, ou seja, que é capaz de conversar tanto com equipamentos IPv6 quanto com equipamentos IPv4.

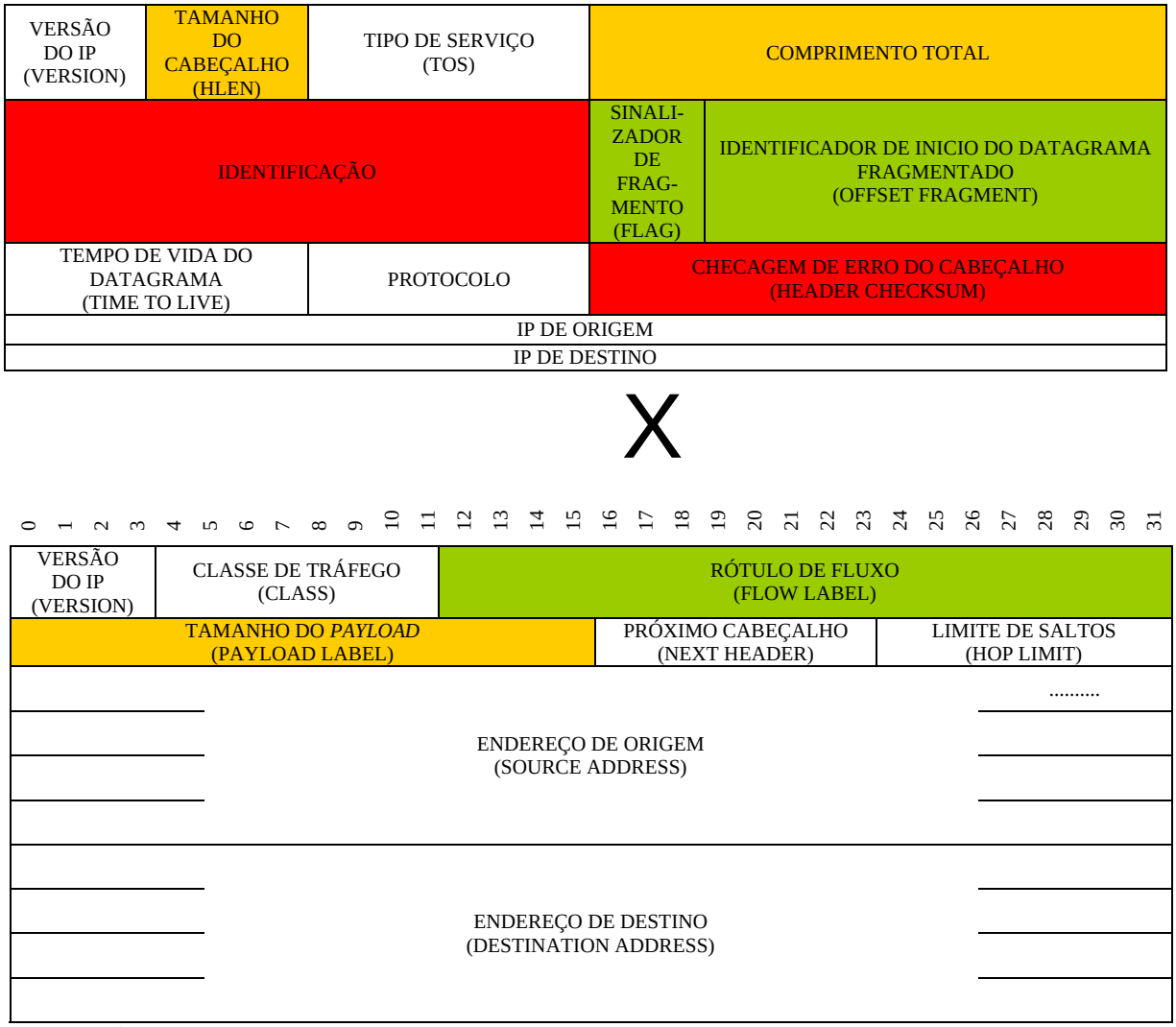
5.5. COMPARANDO AS VERSÕES DO PROTOCOLO

Quando se comparam os dois cabeçalhos conforme apresentado na figura 30, observa-se que muitos campos do IPv4 foram suprimidos devido a funções re-projetadas no IPv6.

Campos preservados: **(TCP/IP-Tutorial(2000, p. 350))**

- Versão do IP
- Tipo de Serviço, agora apresentado como Classe;
- Tempo de vida, nomeado de Limite de Saltos;
- Protocolo, denominado de Próximo Cabeçalho;
- Endereço de Origem;
- Endereço de Destino.

Figura 30 – Comparação entre os cabeçalhos IP versão 6 e 4



Fonte:Própria

Campos integrados:

- SINALIZADOR DE FRAGMENTO e IDENTIFICADOR DE INICIO DO DATAGRAMA FRAGMENTADO – Agora chamada de RÓTULO DE FLUXO

Campos Substituídos:

- TAMANHO DO CABEÇALHO e COMPRIMENTO TOTAL DO DATAGRAMA – No IPv6, o cabeçalho possui tamanho fixo, não sendo mais necessário o cálculo deste. O campo

TAMANHO DO PAYLOAD informa qual o tamanho do datagrama já subtraindo os 40Bytes do cabeçalho.

Campos Eliminados:

- IDENTIFICAÇÃO – Os cabeçalhos de extensão possuem informações mais completas sobre fragmentação, eliminando assim a necessidade de tal campo.
- CHECAGEM DE ERROS – Entendendo que os níveis superiores tratam a questão de checagem de erros, o IPv6 não trata mais as questões de garantia, apesar de existir um cabeçalho opcional de autenticação que certifica se o datagrama foi recebido sem erros.

5.6. REFLEXÕES SOBRE ALGUNS MOTIVOS PARA MIGRAR DE IPv4 PARA IPv6

Como vimos ao longo deste estudo, existe uma série de fatores que irá influenciar na decisão das corporações em migrarem sua rede para este novo sistema de comunicação de nível 3.

Segundo os autores do livro ***TCP/IP, Tutorial e técnico***, podem ser dois os motivos: “necessidade de se ter os requisitos apropriados para os novos recursos que exigem o IPv6, ou a exaustão do espaço de endereços IPv4.”

Eles alegam que em grandes organizações com um parque computacional muito elevado, poderia ser muito oneroso tomar a decisão de troca de tecnologia, uma vez que a maioria dos seus equipamentos de roteamento e até mesmo outros tipos de hardware teriam que ser substituídos. Para estas empresas, a única motivação, seria a limitação do mercado ao uso das conexões e serviços em IPv6. Caso isso não acontecesse, o melhor que teriam a fazer, seria criar mecanismos de compatibilização, conforme apresentado no item anterior, para permitir que seu ambiente computacional não deixasse de falar com clientes e fornecedores e até

mesmo para receber os recursos necessários para o bom andamento do seu negócio.

Sabendo-se que o protocolo ainda está em desenvolvimento, mesmo as novas corporações devem ficar atentas antes de tomarem a decisão de implementar sua rede com uma plataforma puramente IPv6, mesmo porque elas terão a necessidade de se conectar com o mundo que ainda é IPv4.

Para pequenas empresas, onde a conversão pode ser feita de maneira rápida, afigura-se boa a opção de implantarem redes IPv6, desde que nelas existam mecanismos de compatibilização com redes IPv4. Vantagem relevante, visto que os mecanismos de distribuição de endereços são facilmente automatizados sem a necessidade de custos adicionais com servidor de DHCP conforme apresentado durante o estudo.

A implementação da versão 6 do protocolo IP visa melhorar o tráfego nos links WAN diminuindo a carga de processamento nos roteadores, bem como o aumento significativo na faixa de endereçamento lógico.

Àqueles que quiserem implantar Ipv6 em seus ambientes, é recomendado que, antes do início desse trabalho, montem laboratórios por meio dos quais se possam descobrir e entender o que estiver sendo configurado..

Nos laboratórios aqui apresentados, poderá ser observado que há muitas diferenças de implantação para cada plataforma de Sistema Operacional utilizada, o que dificulta a implantação direta. Uma boa parte das publicações disponíveis não é funcional no ambiente proposto neste trabalho devido à diferença de plataforma.

6. LABORATÓRIO PRÁTICO

O laboratório montado para realização desta parte do trabalho, visa criar um ambiente em IPV6 para análise das funções mais utilizadas em uma rede de comunicação de dados, baseado na teoria apresentada anteriormente bem como apresentar algumas aplicações técnicas que permitam uma convivência entre redes IPv6 e IPv4.

O ambiente é constituído por 4 computadores utilizando os seguintes sistemas operacionais:

Windows XP Professional
Linux Fedora 5
Linux Free BSD 5
Linux Ubuntu 6

6.1. CONFIGURAÇÃO DOS LABORATÓRIOS DE TESTES

Os testes serão divididos em quatro laboratórios com o objetivo de testar as compatibilidades entre *hardware* e *software* em algumas atividades básicas no cotidiano de utilização das redes.

Nestes testes também serão realizadas capturas mediante emprego de ferramentas de *sniffer*, a fim de exibirem-se os cabeçalhos de dados e comportamento do protocolo e poder-se comparar com o contido na descrição teórica.

Os laboratórios estarão equipados da seguinte maneira:

- **Laboratório 1** – Os equipamentos estarão ligados a um *HUB* e um *switch* com dois objetivos:

    - Demonstrar que o protocolo é compatível com equipamentos nível 1 e
    - Testar os comandos de rede para teste de conexão como PING6, comandos de transferência de arquivos como SCP, FTP, e comandos de administração remota como SSH e TELNET.

- **Laboratório 2** – Utilizar-se-á equipamento para roteamento com o sistema QUAGGA entre redes Ipv6 distintas.

- **Laboratório 3** – Este laboratório tem como objetivo apresentar que há mecanismos que compatibilizam interconexões entre as versões 4 e 6

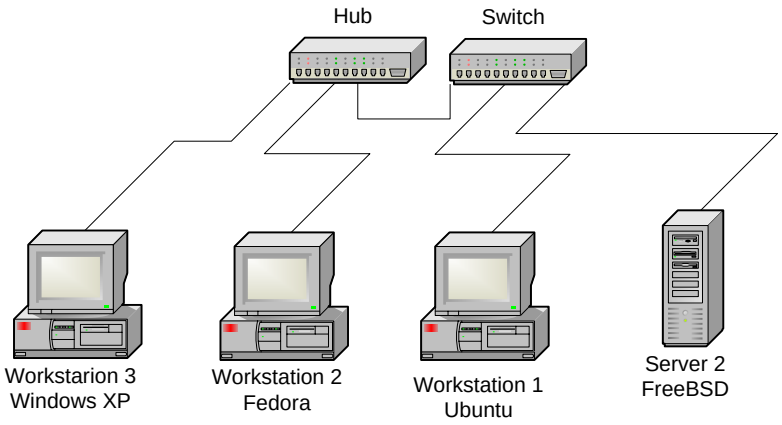
do protocolo, o que indica haver a possibilidade de uma migração tranqüila.

6.2. LABORATÓRIO 1

6.2.1. CENÁRIO:

Este laboratório consiste de uma rede em um único domínio de colisão utilizando HUB e SWITCH conforme apresentado na figura abaixo.

Figura 31 - Diagrama de interligação de equipamentos do LAB 1



Fonte:Própria

6.2.2. OBJETIVO:

Apresentar a compatibilidade do protocolo com equipamentos L1 e L2, bem como testar e analisar o comportamento dos datagramas utilizando comandos básicos de rede.

6.2.3. PROCEDIMENTO:

Consiste em descrever os passos de montagem deste laboratório, referente à instalação e configuração dos equipamentos envolvidos.

6.2.4. TABELA DE CONFIGURAÇÕES

Os equipamentos servidores e estações receberão as seguintes configurações:

Tabela 13 – Endereçamento dos *hosts do LAB 1*

Nome da Estação	Descrição	End. Ipv6	End. Ipv4
Fedora-WSLin100	WorkStation Linux Fedora 5	fec0::a001:100/32	----------
Ubuntu-WSLin200	WorkStation Linux Ubuntu 6	fec0::a001:200/32	----------
Windows-WSWin300	WorkStation MS Win XP pro	fec0::a001:300/32	----------
FreeBSD-SRVBSD500	Servidor FreeBSD 5	fec0::a001:500/32	----------

Fonte: Própria

6.2.5. CONFIGURAÇÃO DOS EQUIPAMENTOS LINUX:

- Proceder a instalação do Linux normalmente com todas as atualizações e pacotes necessários;
- Instalar o serviço SSH, TELNET e FTP;
- Verificar se a versão do *kernel* do Linux é igual ou superior a 2.4.0. Somente estas versões são compatíveis com o Ipv6. Para isso, vá à linha de comando e digite

 # uname –r

Figura 32 – Apresentação do comando UNAME

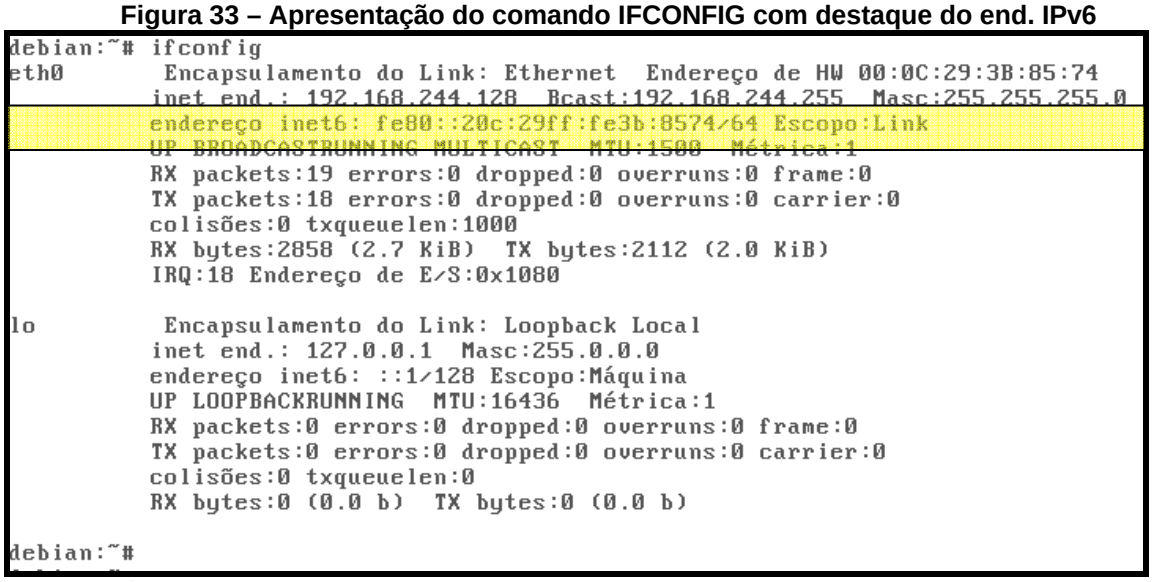
```
FreeBSD-SRVBSD500# uname -r
6.2-RELEASE
```

Fonte: Própria

- Conectar como usuário root;
- Verificar se o protocolo Ipv6 já está ativo digitando

 # ifconfig

Figura 33 – Apresentação do comando IFCONFIG com destaque do end. IPv6



Fonte: Própria

- Verificar se aparece uma linha semelhante à destacada na figura 33. Esta linha indica que o seu Sistema Operacional já está compatível com o protocolo Ipv6 e já adicionou um endereço automático baseado no MAC *Address*, utilizando-se da parte referente ao numero seqüencial do endereço físico
- Caso não esteja ativado carregar o protocolo, digitando

  # insmod ipv6
  # modprob ipv6

- Editar o arquivo /etc/modules e Acrescentar ipv6 ao final do texto para que o suporte ao protocolo carregue automaticamente quando o servidor for reinicializado;
- Para inserir um endereços Ipv6 na interface de rede do equipamento de maneira provisória (enquanto a máquina estiver ligada), digite:

- Nos sistemas operacionais **Linux**

 # ifconfig [interface de rede] inet6 add [endereço IPv6 / quantidade de dígitos de máscara];

- Nos sistemas **BSD**

 # ifconfig [interface de rede] inet6 [endereço IPv6/quantidade de dígitos da máscara] add

- Para configurar permanentemente proceder da seguinte maneira

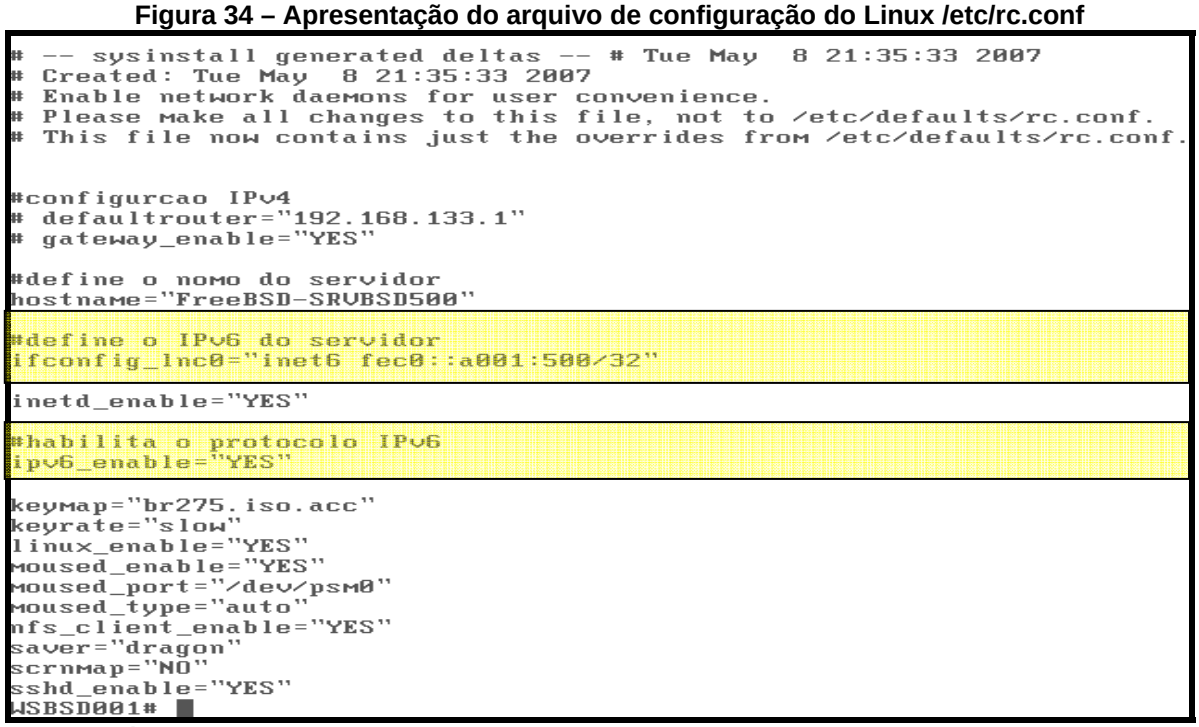
  - FreeBSD

    - Editar o arquivo ***/etc/rc.conf***
    - Acrescentar a linha:

ipv6_enable="YES"
ifconfig_[interface]="inet6 [endereço]/[quantidade de dígitos de máscara]"

    - O arquivo final deverá ficar semelhante ao apresentado na seqüência:

Figura 34 – Apresentação do arquivo de configuração do Linux /etc/rc.conf



Fonte: Própria

- Ubuntu (sistemas baseados em Debian de um modo geral)

  - Editar o arquivo /etc/network/interfaces
  - Localizar a linha ***auto eth0***
  - Logo abaixo, acrescentar as linhas:

Iface eth0 inet6 static
address [endereço IPv6]
netmask [quantidade de dígitos de máscara]

  - O arquivo final deverá ficar semelhante ao que se segue:

Figura 35 – Apresentação do arquivo de configuração do Linux /etc/network/interfaces

```
iface eth1 inet dhcp

auto eth2
iface eth2 inet dhcp

#auto ath0
#iface ath0 inet dhcp

auto wlan0
iface wlan0 inet dhcp


# iface eth0 inet dhcp

#Configuracao da interface eth0
auto eth0

#configura interface com IPv6 e endereco fixo
iface eth0 inet6 static

#configura endereco IPv6
address fec0::a001:200
netmask 32
"/etc/network/interfaces" 27 lines, 363 characters written
root@Ubuntu-WSLin200:~# _
```

Fonte: Própria

- Fedora (sistemas baseados em RedHat)

  - Editar o arquivo ***/etc/sysconfig/networking/devices/ifcfg-eth0***
  - Acrescentar a linha:

BOOTPROTO=none

IPV6INIT=yes

IPV6ADDR= [endereço]/[quantidade de dígitos de máscara]

  - Como o arquivo já pertence à interface eth0, não é necessário informá-la na configuração.
  - O arquivo final deverá ficar semelhante ao apresentado na figura 36:

Figura 36 – Apresentação do arquivo de configuração do Linux etc/sysconfig/networking/devices/ifcfg-eth0

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]

#configura a interface
DEVICE=eth0

#desabilita o DHCP
BOOTPROTO=none

#configura o endereço de hardware
HWADDR=00:0c:29:16:d2:f2

#habilita carga durante a inicializacao
ONBOOT=yes

#cria a identificacao da maquina
DHCP_HOSTNAME=Fedora-WSLIN100

USERCTL=no

#habilita o protocolo IPv6
IPV6INIT=yes

PEERDNS=yes
TYPE=Ethernet

#atribui o endereco IPv6
IPV6ADDR=fec0::a001:100/32

#configuracoes do IPv4
# IPADDR=10.70.0.241
# NETMASK=255.255.255.0

[root@localhost devices]#
```

Fonte: Própria

- Editar também o arquivo **/etc/sysconfig/network**
- Acrescentar as linhas

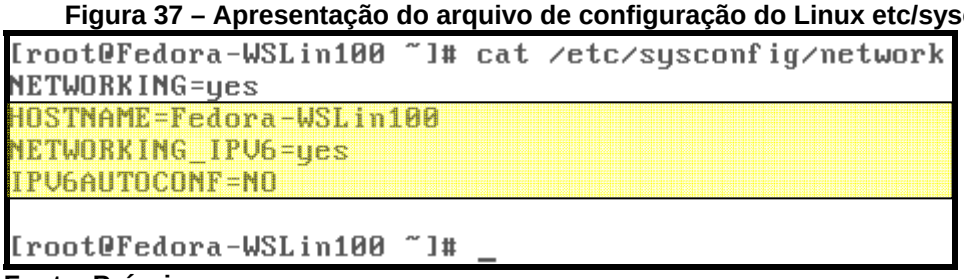
HOSTNAME=[nome da máquina]

NETWORKING_IPV6=yes

IPV6AUTOCONF=no

- O arquivo final deverá ficar semelhante ao apresentado na figura 37:

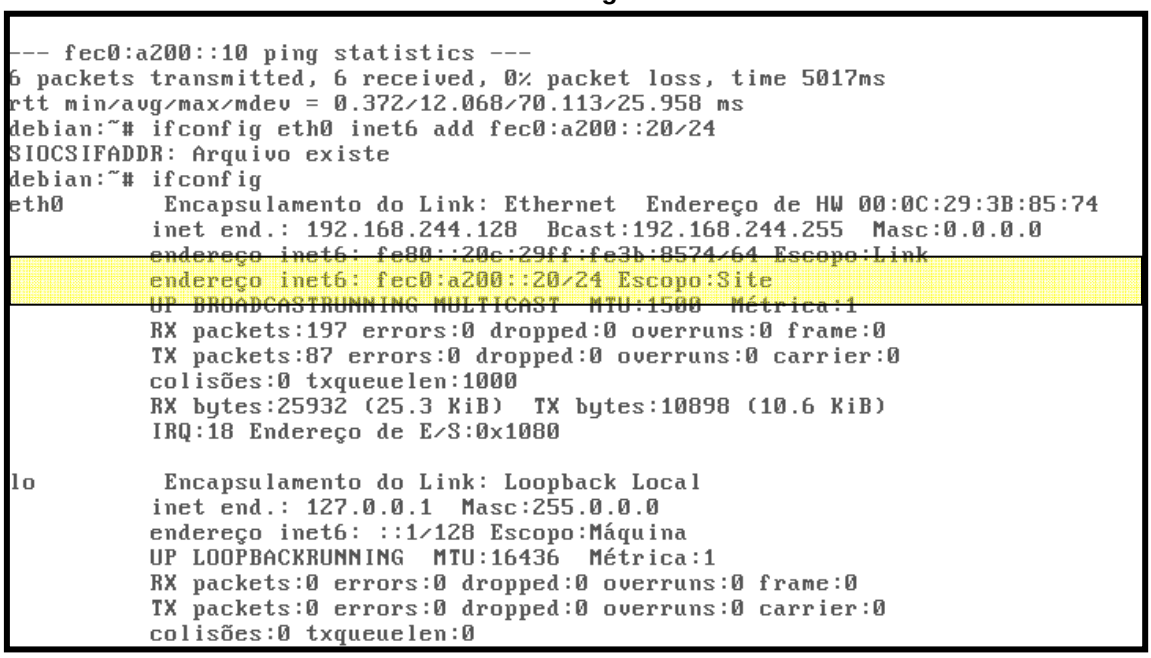
Figura 37 – Apresentação do arquivo de configuração do Linux etc/sysconfig/network



Fonte: Própria

- Efetuar um *boot* no sistema com o comando ***init 6***
- Após a recarga, para todas as versões de Linux e BSD, executar o comando ***ifconfig*** e confirmar se o endereço configurado foi acrescentado;

Figura 38 – Apresentação do comando IFCONFIG com destaque do end. IPv6 recém configurado



Fonte: Própria

- Para remover o endereço criado originalmente pelo sistema, digitar:

o Nos sistemas operacionais Linux

 # ifconfig [interface de rede] inet6 del [endereço ipv6 a ser removido/quantidade de dígitos de máscara]

- Nos sistemas BSD

#ifconfig [interface de rede] inet6 [endereço IPv6/quantidade de dígitos da máscara] delete

6.2.6. CONFIGURAÇÃO DOS EQUIPAMENTOS WINDOWS:

- Ir ao painel de controle do Windows;
- Localizar o ícone conexões de rede;
- Selecionar a placa de rede a qual irá ser configurado o Ipv6;
- Com a tecla direita do mouse, selecionar ‘Propriedades’;
- Desmarcar a opção ‘Protocolo TCP/IP’;
- Selecionar com o mouse o botão ‘Instalar’;
- Selecionar ‘protocolo’ e “clicar” em ‘adicionar;
- Selecionar ‘Microsoft TCP/IP versão 6’ e ‘ok’;
- Voltando à tela anterior, “clicar” em ok novamente;
- Verificar as interfaces de rede do sistema indo ao prompt de comando (DOS) e digitando:

C:\>netsh interface ipv6 show

Figura 39 – Apresentação do comando netsh interface ipv6 show

```
C:\>netsh interface ipv6 show interface
      Índ  Med   MTU    Estado        Nome
      ---  ---  -----  ------------  -----
       3    0   1500  Conectado     Conexão local
       2    0   1500  Desconectado  Conexão de rede sem fio
       1    0   1500  Desconectado  Conexão de rede sem fio
```

Fonte: Própria

- Identificar e anotar o índice (Índ) da interface de rede a ser configurada;
- Configurar o novo endereço IP digitando no prompt de comando:

C:\> netsh interface ipv6 add address interface=(índice anotado ou nome da interface) address=(endereço ipv6 a ser inserido);

- Conferir se o novo endereço foi realmente inserido digitando

C:\> ipconfig

Figura 40 – Apresentação do comando ipconfig

```
C:\>ipconfig

Configuração de IP do Windows

Adaptador Ethernet Conexão local:

   Sufixo DNS específico de conexão  . :
   Endereço IP . . . . . . . . . . . . : 192.168.244.1
   Máscara de sub-rede . . . . . . . . : 255.255.255.0
   Endereço IP . . . . . . . . . . . . : fec0:200::10%2
   Endereço IP . . . . . . . . . . . . : fe80::250:56ff:fec0:1%5
   Gateway padrão. . . . . . . . . . . :
```

Fonte: Própria

OBS.: Este comando só terá efeito se o IPv4 também estiver habilitado. Caso contrário, o comando para verificação do endereço deverá ser ***netsh interface ipv6 show address.***

Figura 41 – Apresentação do comando ipconfig

```
C:\Documents and Settings\i9526583\Desktop>netsh interface ipv6 show address

Consultando estado ativo...
Interface 3: Teredo Tunneling Pseudo-Interface
Tipo end  Estado DAD    Vida vál.   Vida pref.   Endereço
--------  ----------    ----------  ----------   --------------------------
vínculo   Preferencial  infinite    infinite     fe80::ffff:ffff:fffd

Interface 2: Conexão local

Tipo end  Estado DAD    Vida vál.    Vida pref.   Endereço
--------  ----------    ----------   ----------   --------------------------
Manual    Preferencial  infinite     infinite     fec0::a001:300
vínculo   Preferencial  infinite     infinite     fe80::250:56ff:fec0:1

Interface 1: Loopback Pseudo-Interface
```

Fonte: Própria

6.2.7. INSTALAÇÃO DA FERRAMENTA DE SNIFFER (WIRESHARK) NO LINUX OU WINDOWS:

- No Linux DEBIAN
  - Colocar o equipamento na rede com acesso à *internet*
  - Executar o comando *apt-get install wireshark*

- No Windows
  - Realizar *dowload* do sistema wireshark-setup-0.99.5.exe
  - Executar o arquivo com <iniciar> <executar><localizar>
  - Ir até o diretório onde realizou o *dowload*
  - Executar o arquivo wireshark-setup-0.99.5.exe
  - Aceitar todas as opções “clicando” em <next>
  - Clicar em <finish>

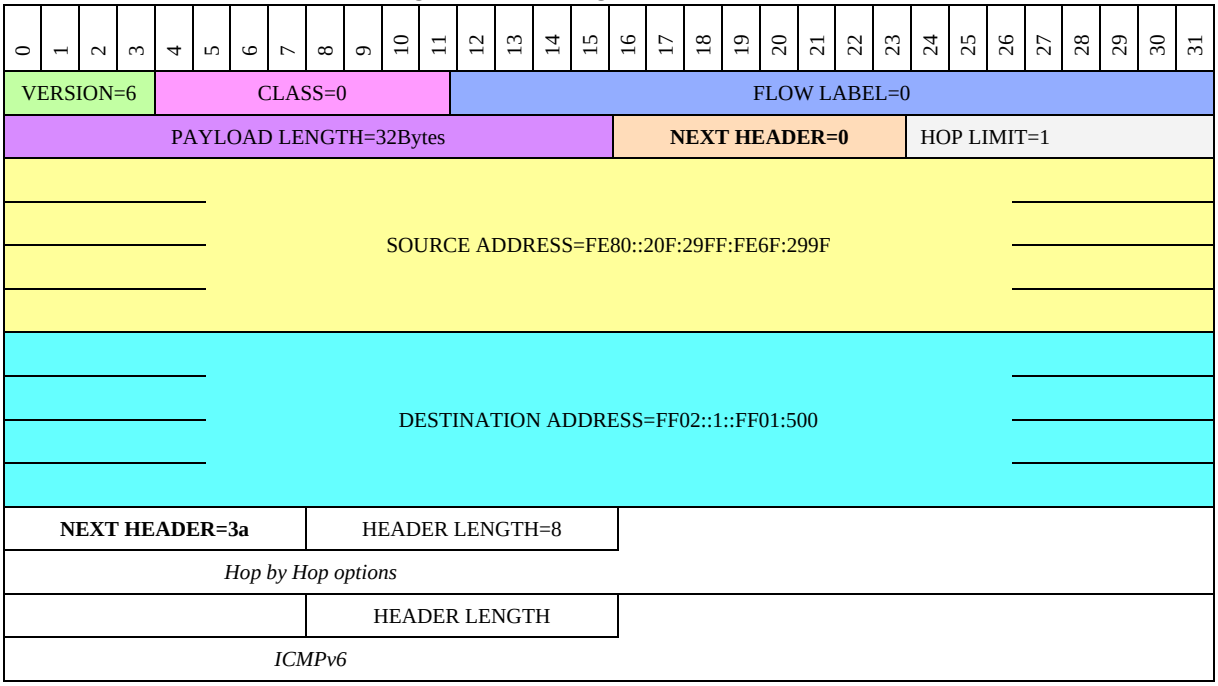
6.2.7.1. Entendendo a captura do WIRESHARK:

O software WIRESHARK é uma ferramenta de depuração de dados de rede que tem como finalidade capturar, de modo transparente, uma amostra dos

datagramas que estão passando no barramento. Trata-se de uma ferramenta livre e gratuita com um excelente grau de eficiência.

A seguir é apresentado um modelo didático do datagrama IPv6 para comparação com a figura 43 que expõe tela com um datagrama capturado pelo WIRESHARK. O objetivo é visualizar o modelo com uma informação real comprovando o exposto no item 7.4.1 do referencial teórico.

Figura 42 – Datagrama de multicast ICMP

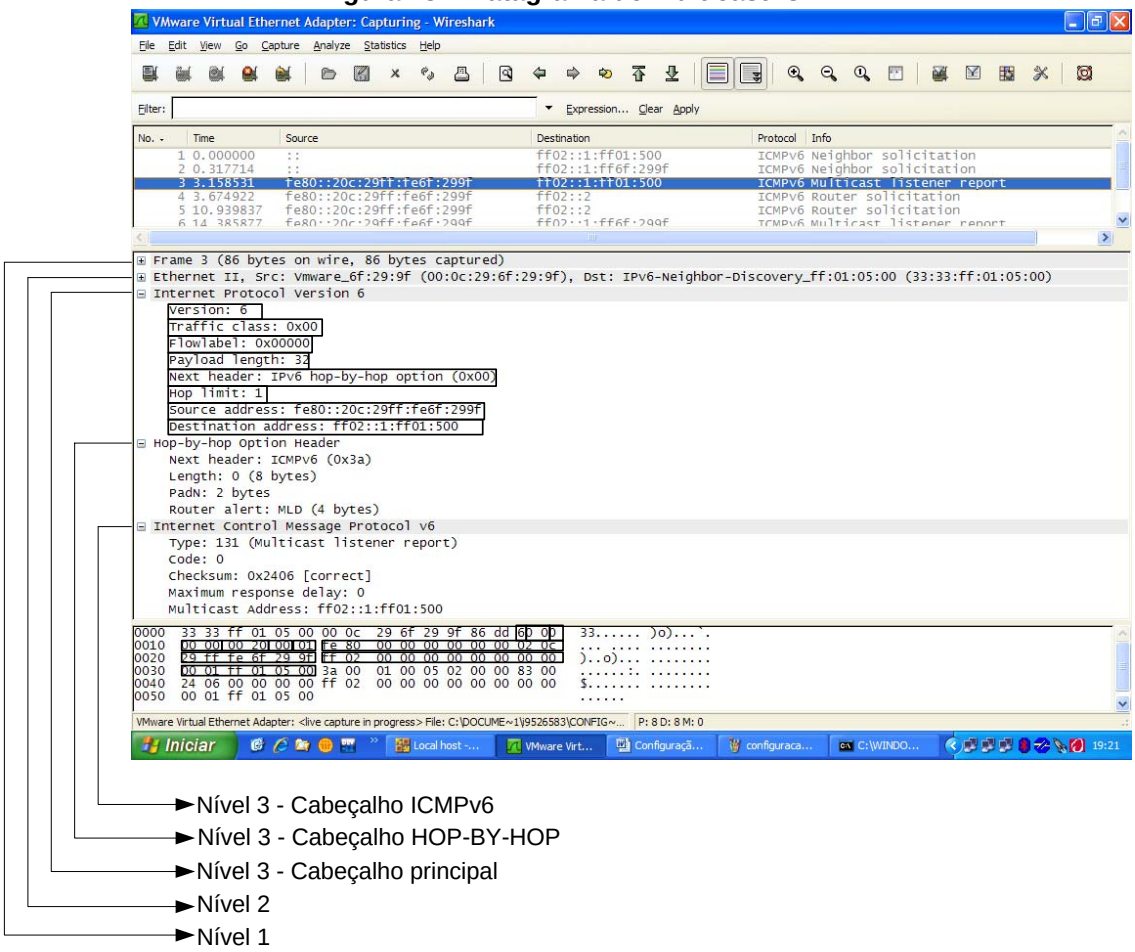


Fonte: Própria

Como podemos observar na figura 43, o datagrama capturado possui exatamente a estrutura apresentada no modelo didático do referencial teórico.

O entendimento dos campos do cabeçalho IPv6 será de fundamental importância para a compreensão dos laboratórios que se seguem.

Figura 43 – Datagrama de multicast ICMP



Fonte: Própria

6.2.7.2. Instalação do HUB e do SWITCH:

Para instalação do HUB e do SWITCH, basta ligá-los à rede elétrica e utilizar um cabo de rede para cada equipamento conforme apresentado na figura 1.

Vale lembrar que estes dispositivos funcionam nos níveis 1 e 2 do modelo OSI. O que significa dizer que ele deve ser compatível com o novo modelo do IP por estarem nos níveis inferiores ao do IP sem se preocupar com o tipo de protocolo de comunicação.

Em função desta característica, eles serão utilizados para comprovação das funcionalidades deste protocolo utilizando-se de serviços básicos de funcionamento de uma rede de dados.

6.2.8. Testes:

O objetivo aqui é testar o comportamento de alguns sistemas de uso em redes com a utilização do protocolo IPV6.

Para os primeiros testes, deverão ser executados os procedimentos abaixo:

- Carregar o sistema WHIRESHARK em um dos equipamentos utilizados no laboratório antes de ligar os demais envolvidos conforme procedimentos abaixo:
- Desligar os equipamentos configurados com o protocolo IPV6 que por ventura ainda estejam ligados;
- Conectar o equipamento com a ferramenta de análise de pacotes ao HUB do laboratório montado;
- Configurar o WHIRESHARK para coletar informações utilizando a opção ***Capture -> Interfaces;***
- Identificar a interface que está conectada ao HUB para a captura de pacotes na rede;
- Efetuar um “click” no botão <start>

Nesta fase, serão realizados os testes das funcionalidades do protocolo utilizando os comandos PING, TELNET, FTP, SSH, e SCP. Serão originados, entres as estações, tráfegos de dados com monitoração utilizando a ferramenta WIRESHARK para captura dos datagramas e comparação com o exposto na parte teórica.

Comando Ping6

Com a versão 6 do IP, o comando PING foi modificado para PING6 para diferenciar comandos entre as versões. Para testá-lo, seguir os passos que se segue:

- Configurar filtro para os protocolo IPv6
- Proceder o comando abaixo em uma estação Linux contra um endereço qualquer de outra estação:

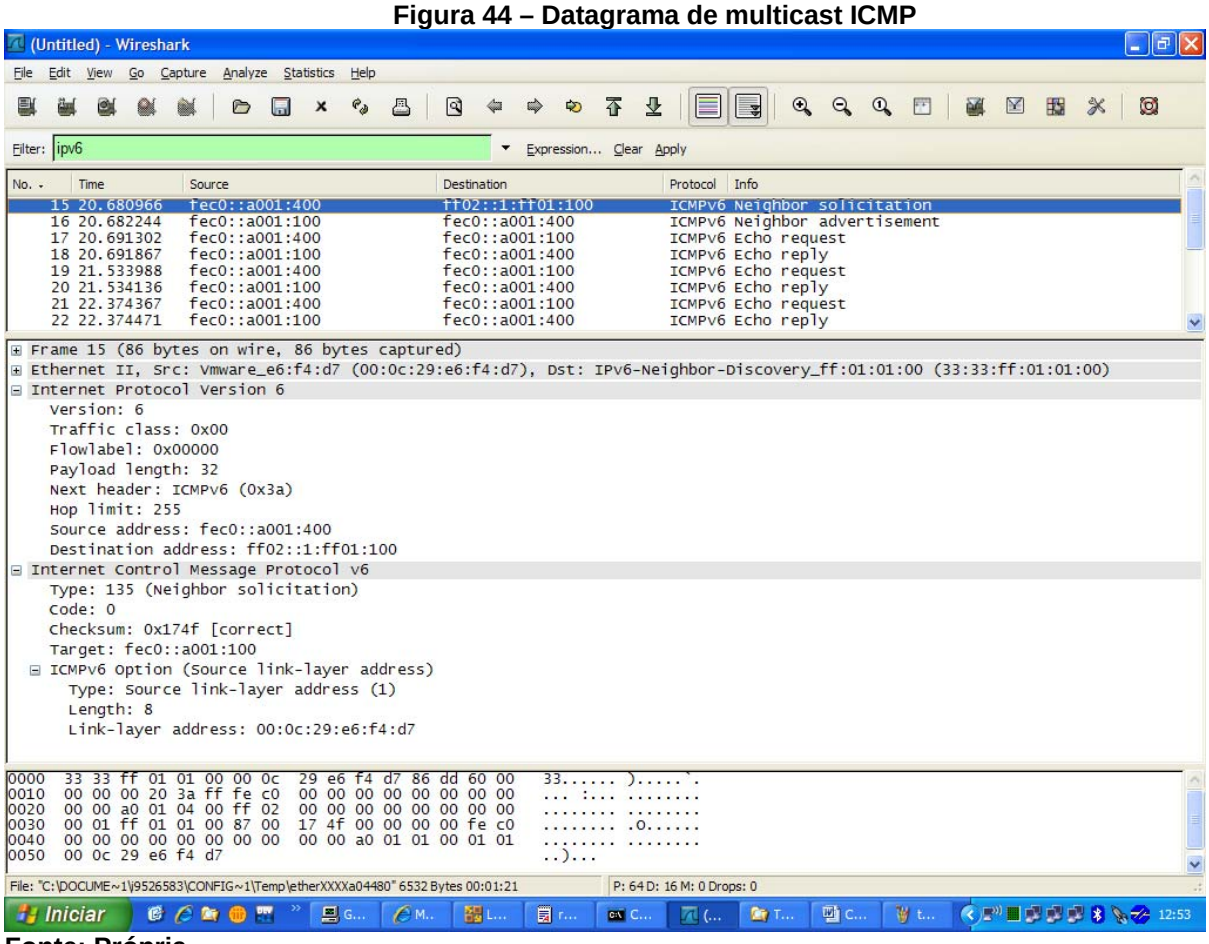
  - Linux e FreeBSD

 # ping6 (endereço Ipv6)

  - Windows

C:\> ping6 <endereço ipv6> % <índice da porta>

Figura 44 – Datagrama de multicast ICMP



Fonte: Própria

Conforme figura acima, podemos observar que, inicialmente, é realizada uma solicitação NDP para identificação e alimentação da tabela de associação

MAC/IP. Em seguida, os comandos ICMPv6, *echo request* e *echo replay* são apresentados com os respectivos endereços de origem e destino.

Na parte de detalhamento do pacote, podemos observar o campo version informando a versão 6 do IP e o *next header* indicando que o próximo cabeçalho é do tipo ICMPv6 com o valor 3ª, que corresponde a 58 em decimal, conforme apresentado na tabela 3 do referencial teórico.

Já no cabeçalho ICMPv6 podemos observar o tipo 135 indicando o anuncio de solicitação de vizinho (tabela 11) e também o tamanho (*length*) dado em *bytes* seguindo o tamanho original do PING da versão 4.

OBS.: Os 64 bits (8 *bytes*) apresentados no campo *length* não incorporam o cabeçalho principal que possui um tamanho fixo de 40 *bytes* (item 5.4.1)**.**

É interessante notar que no IPv6, não mais tratamos com tabelas ARP. Em seu lugar foi instituída a tabela NEIGHBOR que possui a mesma finalidade.

Assim como a primeira, os relacionamentos possuem um prazo de validade que, nos sistemas LINUX, foi constatado ser muito curto (cerca de menos de um minuto).

Por este motivo, para se visualizar a tabela NEIGHBOR é necessário que se execute alguma chamada ao *host* de destino (um ping por exemplo) e em seguida se proceda o comando de verificação conforme segue abaixo:

- Windows

c:\> netsh interface ipv6 show neighbors

Figura 45 – Saída do comando de verificação de vizinhos

```
C:\>netsh interface ipv6 show neighbors
Interface 7: VMware Network Adapter VMnet1
Endereço Internet                              Endereço físico    Tipo
---------------------------------------------  -----------------  -----------
fec0::a001:100                                 00-0c-29-16-d2-f2  Obsoleto
fec0::a001:301                                                    Incompleto
fec0::a001:500                                                    Incompleto
fe80::250:56ff:fec0:1                          00-50-56-c0-00-01  Permanente
fec0::a001:302                                 00-50-56-c0-00-01  Permanente
```

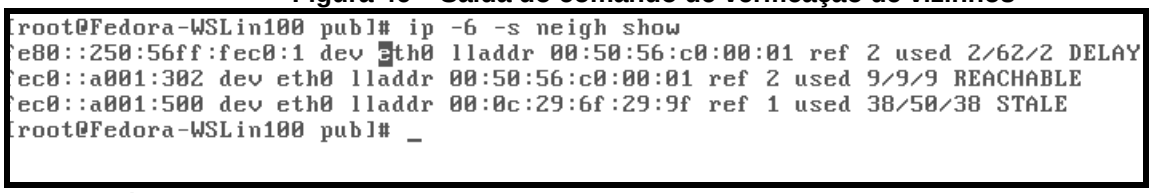
Fonte: Própria

Observe que na coluna Tipo, aparece informações de “obsoleto”, “Incompleto” e “Permanente”. Estas informações são referentes ao tempo de validade. Obsoleto é quando o endereço na tabela ND já está vencido, Incompleto refere-se a endereços que ele não conseguiu resolver o MAC e Permanente é quando o endereço pertence ao próprio host.

- Linux

 # ip -6 –s neigh show

Figura 46 – Saída do comando de verificação de vizinhos

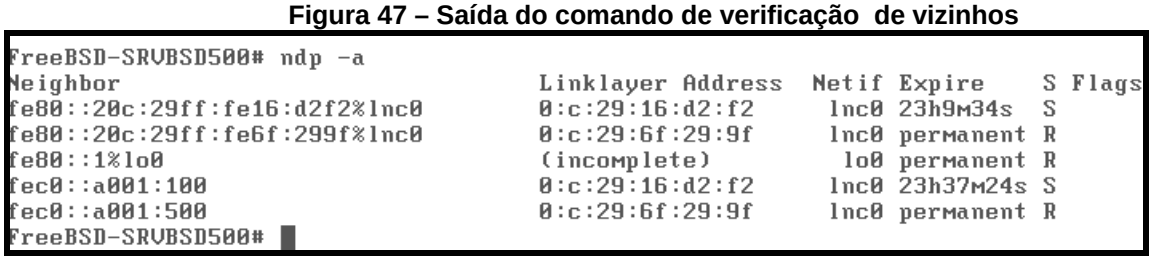


Fonte: Própria

- FreeBSD

 # ndp –a

Figura 47 – Saída do comando de verificação de vizinhos



Fonte: Própria

Telnet

O comando telnet é padrão, porém algumas distribuições podem solicitar o parâmetro -6 para diferenciar da versão 4.

- Configurar filtro para os endereços envolvidos;
- Configurar filtro para os protocolo IPv6
- Proceder ao comando abaixo em uma estação **Linux** contra um endereço qualquer de outra estação:

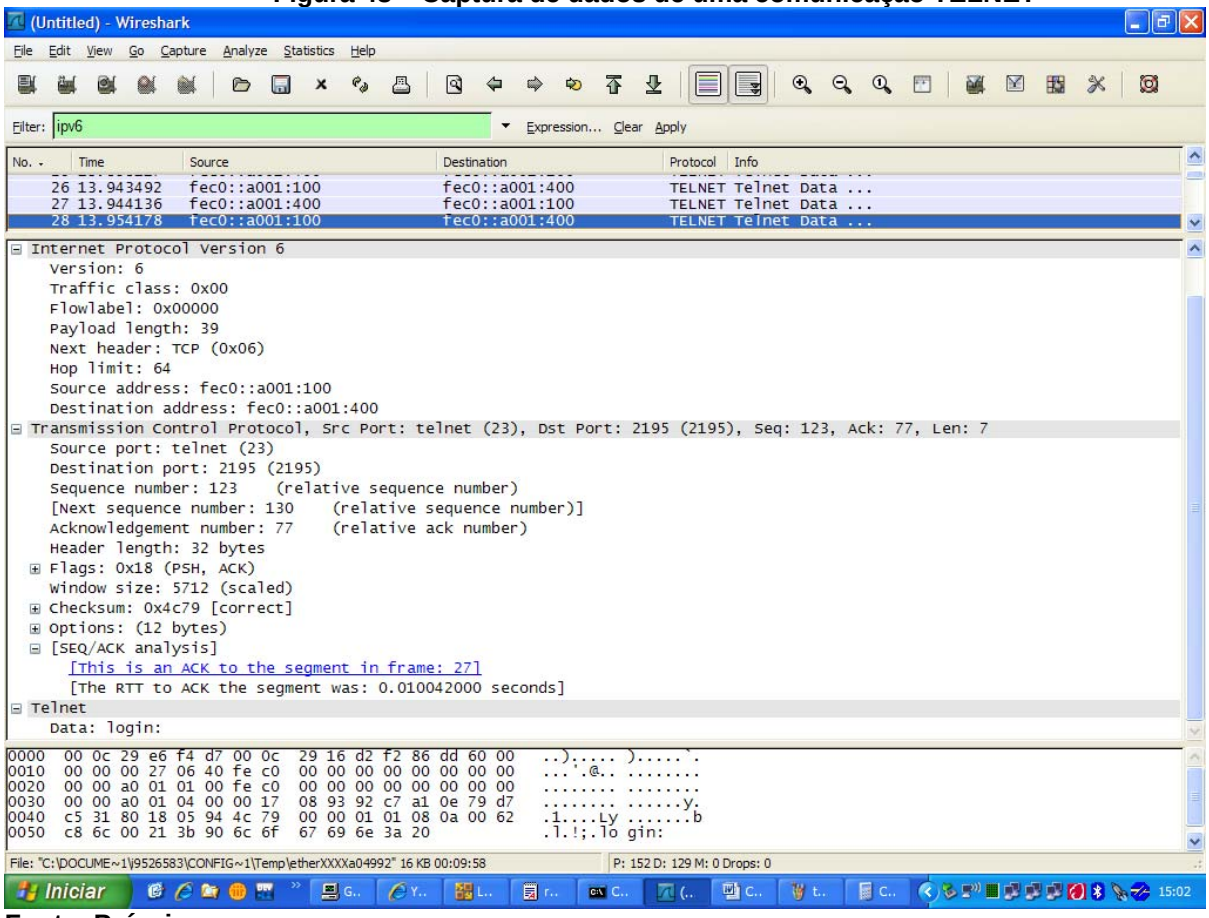
o LINUX

 # telnet <endereço ipv6>

o Windows

C:\> telnet <endereço ipv6>%<indice da interface de rede>

Figura 48 – Captura de dados de uma comunicação TELNET



Fonte: Própria

Nota-se na figura acima que o serviço TELNET funciona da mesma maneira que no IPv4, ou seja, os dados passam abertos, sem criptografia (na janela de detalhes do pacote, ao final). Observamos também que continua operando sobre o TCP na mesma porta (23).

Para que este serviço funcione, é necessário informar ao servidor que ele deve receber conexões TELNET com endereços IPv6. No LINUX Fedora, a linha flags do arquivo /etc/xinetd.d/telnet deve estar da seguinte maneira: **flags = IPv6 REUSE**. Isto permitirá que estações IPv6 possam realizar telnet.

Como a comunicação é transparente, ou seja, sem criptografia, se é possível montar a transação realizada pelo operador a partir dos dados coletados conforme mostrados abaixo:

Figura 49 – Remontagem de uma captura TELNET realizada com o WireShark

```
.......... ..!.."..'...
..... ..#..'
........!.."...
..#....P....
.. .....'.........
.. .38400,38400....'.......linux..
...
...
...Fedora Core release 5 (Bordeaux)

Kernel 2.6.15-1.2054_FC5 on an i686


...
login:
e
e
l
l
i
i
s
s
n
n
a
a
l
l
d
d
o
o

.


Password:
e
l
i
s
n
a
l
d
```

```
o

.


Last login: Tue May 15 10:26:51 from 0.0.0.0


[elisnaldo@Fedora-WSLin100 ~]$
s
s
u
u

.


Senha:
t
t
e
e
s
s
t
t
e
e

.


[root@Fedora-WSLin100 elisnaldo]#
e
e
x
x
i
i
t
t

.


exit


[elisnaldo@Fedora-WSLin100 ~]$
```

Fonte: Própria

Podemos observar que os caracteres digitados são “ecoados” entre a origem e o destino.

FTP

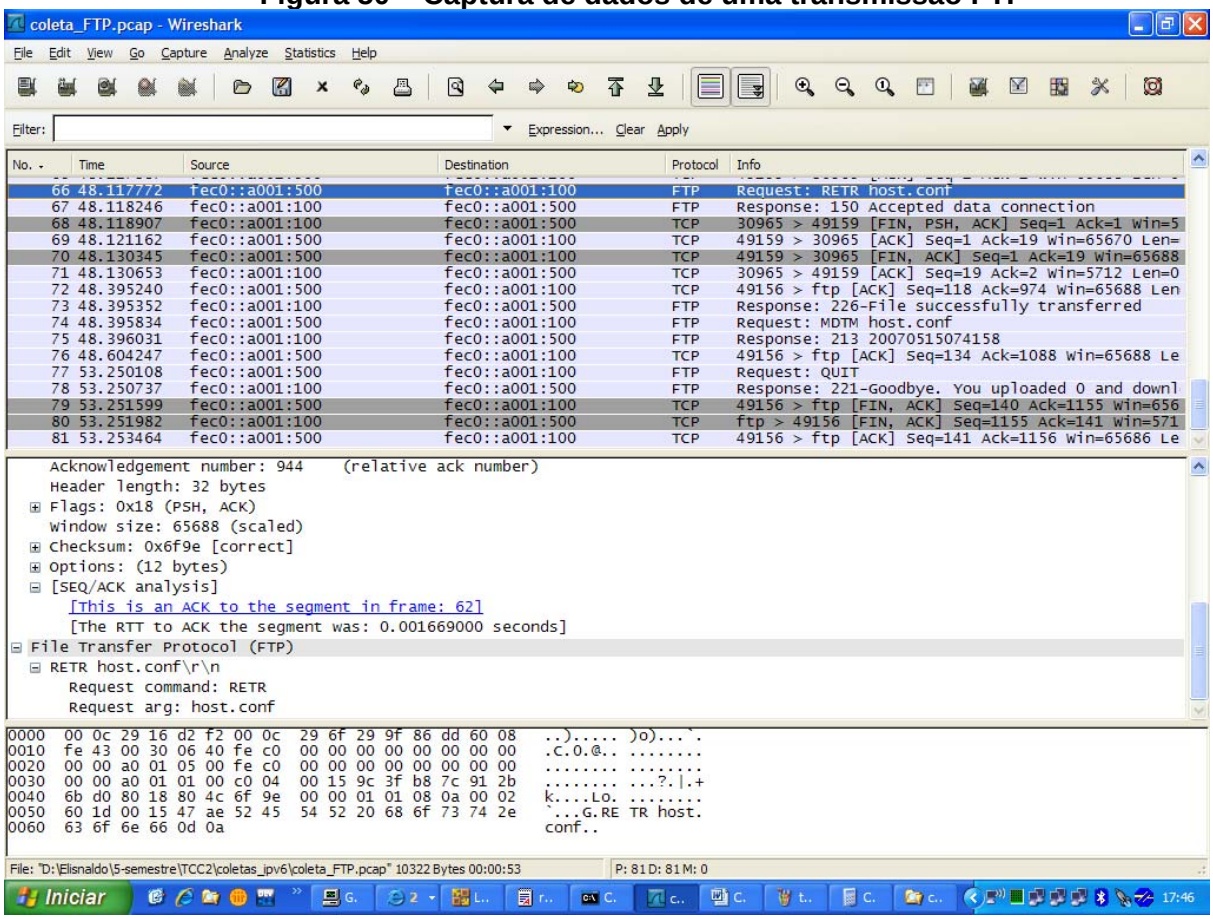
Com a versão 6 do IP, o comando PING foi modificado para PING6 afim de diferenciar comandos entre as versões. Para testá-lo, o processo abaixo deve ser seguido:

- Configurar filtro para os endereços envolvidos;
- Configurar filtro para o protocolo IPv6
- Proceder o comando abaixo em uma estação **Linux** contra um endereço qualquer de outra estação:

    - FreeBSD

 # ftp -6 (endereço Ipv6)

Figura 50 – Captura de dados de uma transmissão FTP



Fonte: Própria

Na figura 50, podemos observar o inicio da transmissão do arquivo host.conf na linha 66, seu término na linha 73, o pedido de desconexão na linha 77 e o fim da comunicação na linha 81.

Os dados passam de modo transparente podendo-se até remontar a transmissão como segue:

Figura 51 – Remontagem de uma captura FTP realizada com o WireShark

```
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 04:50. Server port: 21.
220-Only anonymous FTP is allowed here
220 You will be disconnected after 15 minutes of inactivity.
USER anonymous
230 Anonymous user logged in
SYST
215 UNIX Type: L8
FEAT
211-Extensions supported:
 EPRT
 IDLE
 MDTM
 SIZE
 REST STREAM
 MLST type*;size*;sizd*;modify*;UNIX.mode*;UNIX.uid*;UNIX.gid*;unique*;
 MLSD
 ESTP
 PASV
 EPSV
 SPSV
 ESTA
 AUTH TLS
 PBSZ
 PROT
 UTF8
211 End.
PWD
257 "/" is your current location
EPSV
229 Extended Passive mode OK (|||24868|)
LIST
150 Accepted data connection
226-Options: -a -l
226 3 matches total
CWD pub
250 OK. Current directory is /pub
PWD
257 "/pub" is your current location
```

```
EPSV
229 Extended Passive mode OK (|||46502|)
LIST
150 Accepted data connection
226-Options: -a -l
226 3 matches total
TYPE I
200 TYPE is now 8-bit binary
SIZE host.conf
213 17
EPSV
229 Extended Passive mode OK (|||30965|)
RETR host.conf
150 Accepted data connection
226-File successfully transferred
226 0.005 seconds (measured here), 3.42 Kbytes per second
MDTM host.conf
213 20070515074158
QUIT
221-Goodbye. You uploaded 0 and downloaded 1 kbytes.
221 Logout.
```

Fonte: Própria

SSH

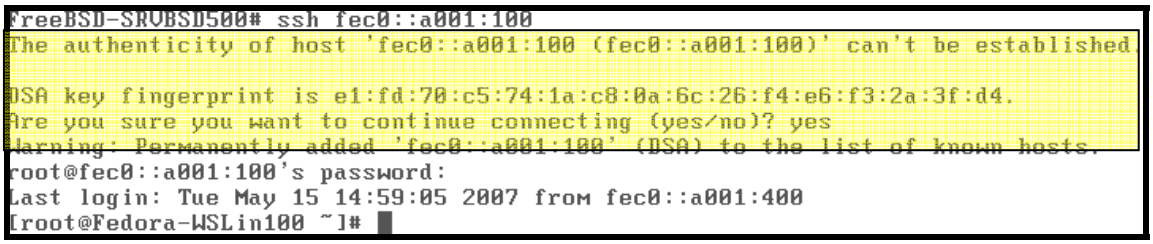
O comando SSH segue a mesma sintaxe da versão 4. Ele provê criptografia na administração remota de equipamentos com SO UX. O Windows não possui uma ferramenta nativa para este comando, porém, quem tiver a intenção de acessar uma máquina LINUX utilizando essa segurança, deverá utilizar o aplicativo PUTTY FOR WINDOWS que é gratuito e funciona muito bem com a versão 4 do protocolo IP. Apesar de a ferramenta possuir suporte ao Ipv6, sua interface gráfica não endente o endereço desta natureza. Isso é um BUG retratado pelo próprio fabricante para o Windows Vista, porém, os mesmos erros foram encontrados no XP e 2003. (http://www.chiark.greenend.org.uk/~sgtatham/putty/wishlist/vista-ipv6.html 06/06/2007.

- Configurar filtro para os endereços envolvidos;
- Configurar filtro para o protocolo IPv6
- Proceder ao comando abaixo em uma estação **Linux** contra um endereço qualquer de outra estação:

- Linux e FreeBSD

 # ssh <endereço Ipv6>

Figura 52 – Apresentação da primeira conexão SSH em uma estação Fedora



Fonte: Própria

Um detalhe deve ser observado: quando se trata da primeira conexão, assim como no IPv4, o sistema local irá informar que a autenticidade do *host* não foi estabelecida e que, para este, uma nova chave de criptografia está sendo acionada na lista local de maneira permanente. Isto faz parte da segurança da conexão.

Por padrão, o sistema remoto assume o usuário local como *login name*. Caso haja a necessidade de se efetuar a conexão com um usuário diferente do usuário que está estabelecendo a chamada, pode-se utilizar o comando:

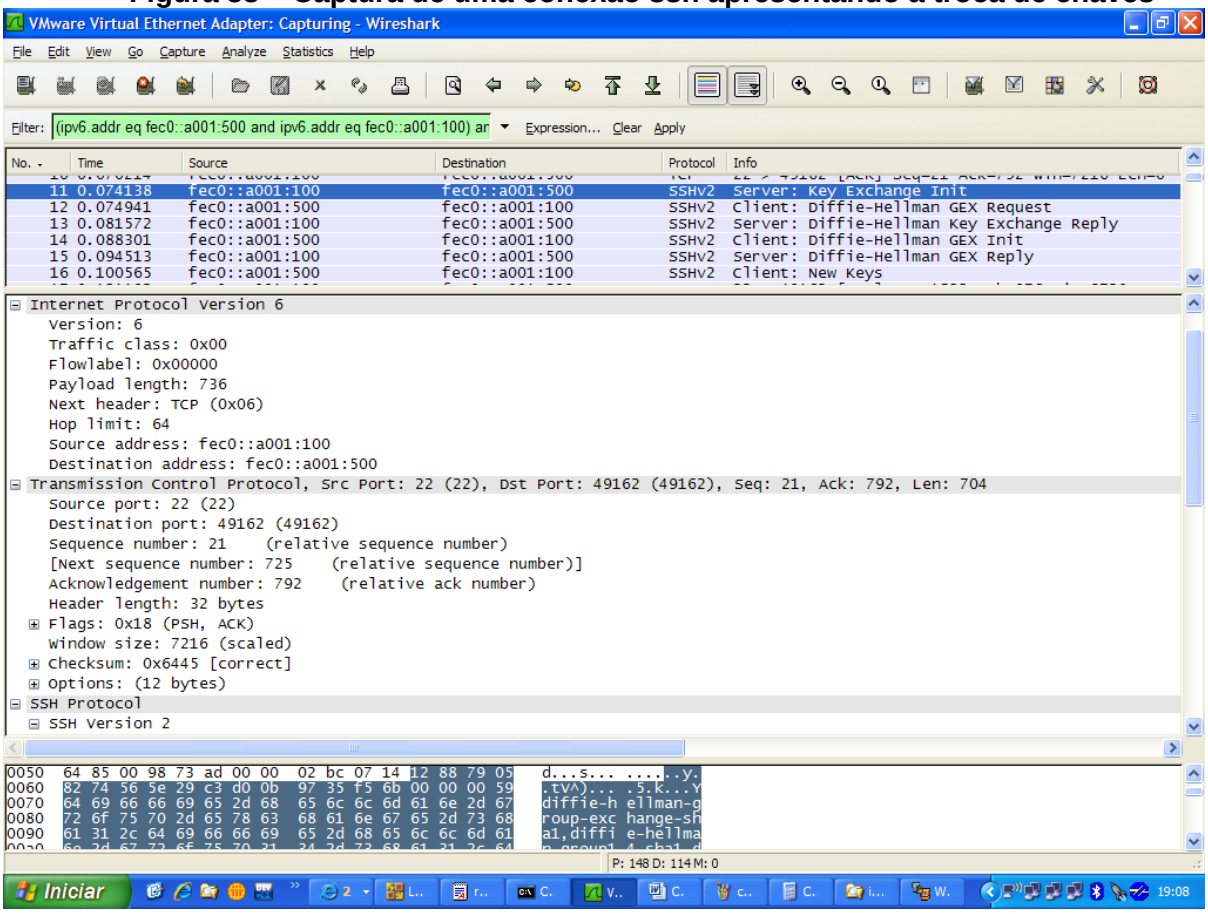
- Linux

 # ssh <endereço Ipv6> –l <nome do usuário>

- Windows (é necessário o arquivo putty.exe)

C:\> putty -6 –ssh <endereço IPv6>%<índice da interface>

Figura 53 – Captura de uma conexão ssh apresentando a troca de chaves



Fonte: Própria

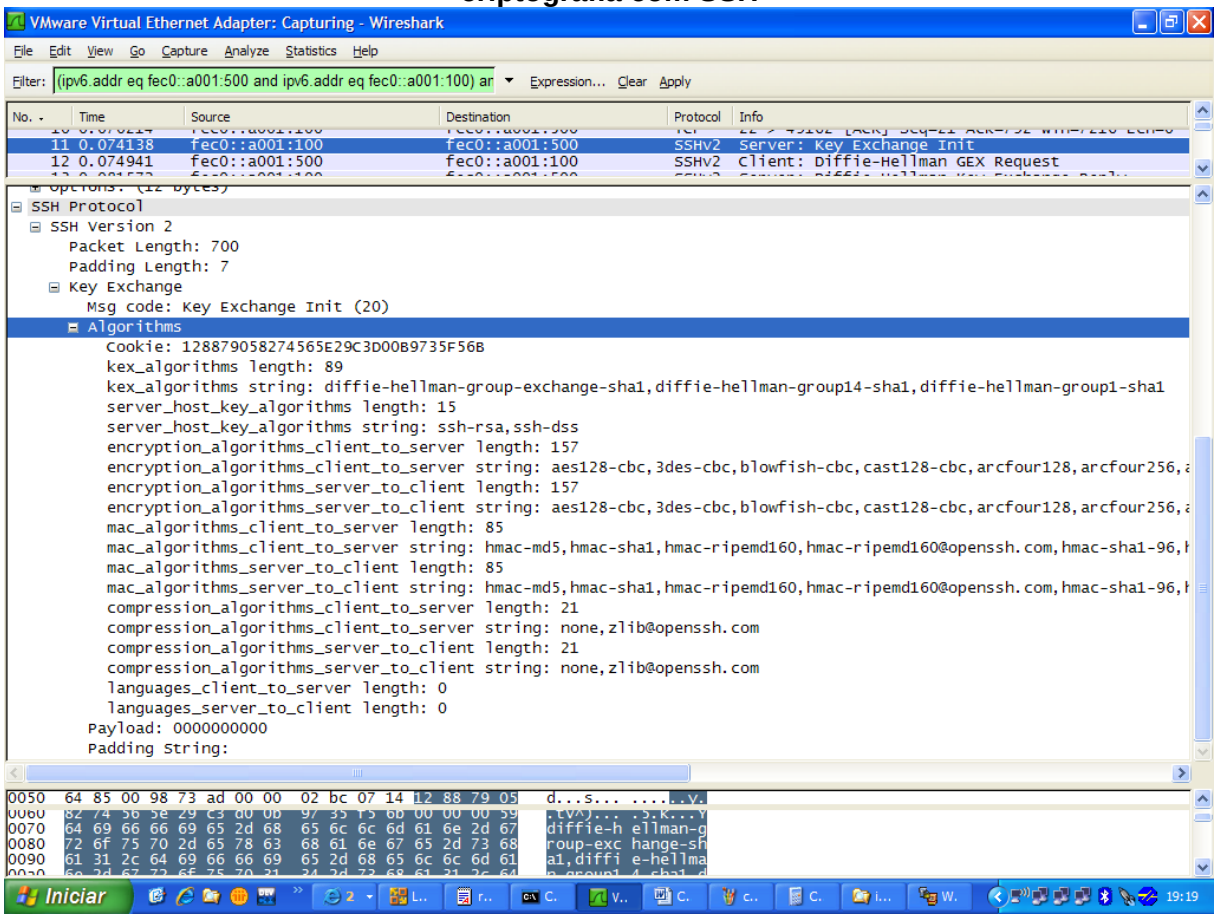
Como se trata de um processo com criptografia, ambos os equipamentos (servidor e estação) devem trocar chaves públicas entre si. No caso particular, a chave utilizada é a ***Diffie-Hellman,*** valendo registrar que este processo está indicado na coleta acima entre as linhas 11 e 16.

O detalhamento do datagrama da linha 11 é o que aparece na figura. Pode-se observar que, bem ao final, aparece o *ssh protocol* que se apresenta melhor detalhamento abaixo.

> "O algoritmo Diffie-Hellman foi o primeiro algoritmo de chave pública a ser inventado. Isto significa que seus autores também são os donos da idéia. O algoritmo pode ser usado para a distribuição de chaves, mas não para cifrar ou decifrar mensagens. Sua segurança reside na dificuldade de calcular logaritmos discretos num campo finito comparada com a facilidade de realizar exponenciações no mesmo

campo.”(Viktoria Tkotz - http://www.numaboa.com/content/view/353/57/ - 13/06/2007).

Figura 54 – Detalhamento da troca de chaves apresentando os algoritmos de criptografia com SSH



Fonte: Própria

SCP

O comando SCP, é utilizado para de transferência de arquivos com criptografia associado ao SSH, valendo ressaltar, aliás, que a porta de conexão é a mesma. O sistema para o Windows é o WinSCP, porém , porém, uma vez que nos testes, não funcionou com IPv6, recomenda-se:

- Configurar filtro para os endereços envolvidos;
- Configurar filtro para os protocolos IPv6
- Proceder ao comando abaixo em uma estação **Linux** contra um endereço qualquer de outra estação:

Para fazer *download*, ou seja, para copiar o arquivo da máquina remota para a local, usar o seguinte comando:

- **Linux**

 # scp -6 usuário@[<endereçoIPv6>]:<nome do arquivo de origem> <nome do arquivo de destino>

- **FreeBSD**

 # scp <usuário@\[<endereçoIPv6>\]:<nome do arquivo de origem> <nome do arquivo de destino>

Para fazer *upload*, ou seja, para copia o arquivo da maquina local para a remota, o comando é:

- **Linux**

 # scp -6 <nome do arquivo de origem> usuário@[<endereçoIPv6>]: <nome do arquivo de destino>

- **FreeBSD**

 # scp <nome do arquivo de origem> usuário@\[<endereçoIPv6>\]: *<nome do arquivo de destino>*

Figura 55 – Comando SCP no Linux

```
root@Ubuntu-WSLin200:~# scp -6 elisnaldo@[fec0::a001:100]:som.mp3 som.mp3
elisnaldo@fec0::a001:100's password:
som.mp3                                       100% 3559KB   1.7MB/s   00:02
root@Ubuntu-WSLin200:~# ls
som.mp3
root@Ubuntu-WSLin200:~#
```

Fonte: Própria

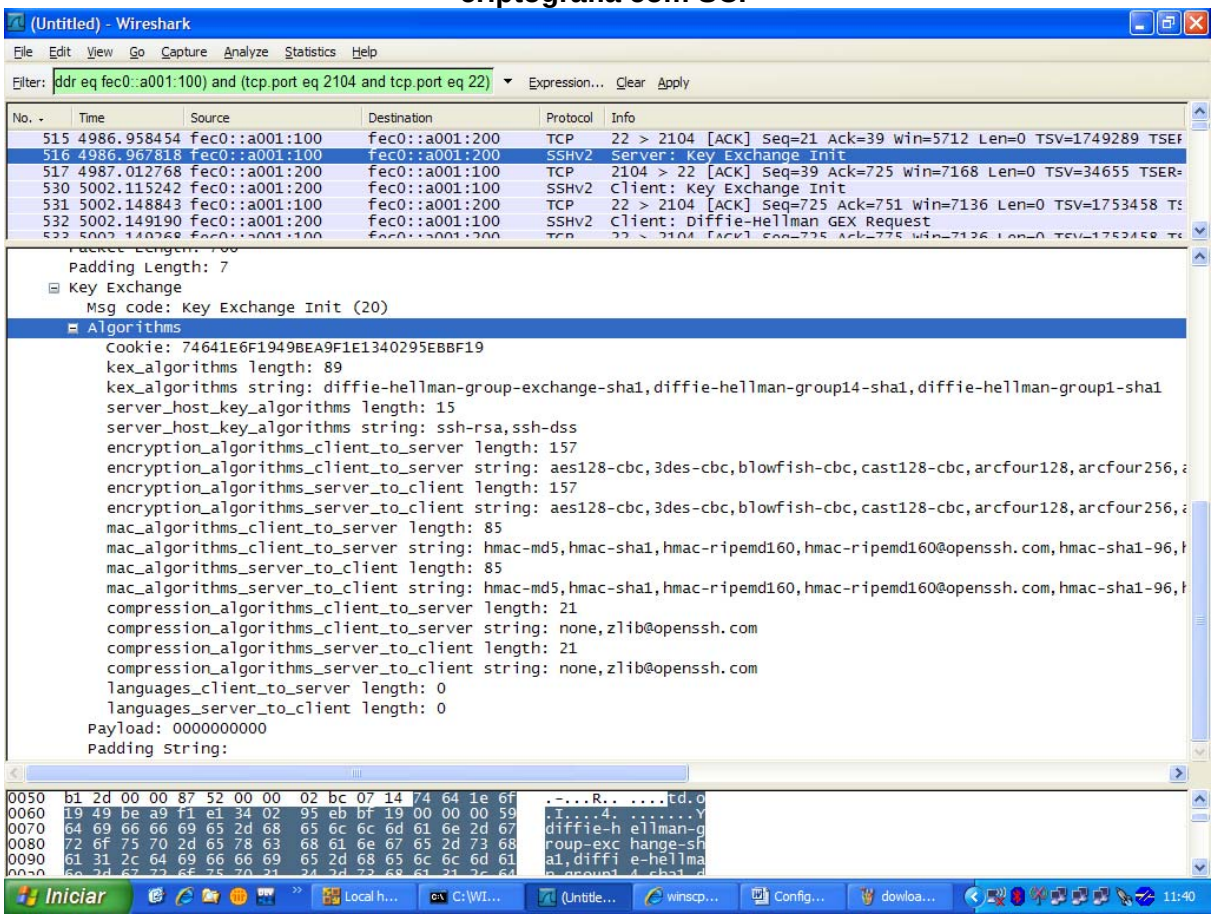
Figura 56 – Comando SCP no FreeBSD

```
FreeBSD-SRVBSD500# scp elisnaldo@\[fec0:c1::c003:200\]:som.mp3 som.mp3
Password:
som.mp3                                       100% 3559KB    3.5MB/s   00:00
FreeBSD-SRVBSD500# ls
.Xauthority     .login          .mailrc         .shrc
.cshrc          .login_conf     .profile        som.mp3
.fonts.cache-1  .mail_aliases   .rhosts
FreeBSD-SRVBSD500# cp som.mp3 som1.mp3
FreeBSD-SRVBSD500# scp som1.mp3 elisnaldo@\[fec0:c1::c003:200\]:som1.mp3
Password:
som1.mp3                                      100% 3559KB    3.5MB/s   00:01
FreeBSD-SRVBSD500#
```

Fonte: Própria

Como pode ser observado na figura 57, as chaves de criptografia são as mesmas utilizadas no SSH.

Figura 57 – Detalhamento da troca de chaves apresentando os algoritmos de criptografia com SCP



Fonte: Própria

6.2.9. Conclusão do primeiro laboratório:

O objetivo principal deste laboratório foi demonstrar que equipamentos de nível 1 e 2 (*hubs*, *bridges*, *transceivers* de mídia, *switches,* etc.) funcionam muito bem para pequenas redes, mas tecnicamente, estes equipamentos continuam sendo ineficientes em função de suas características operacionais.

Observamos também que não existe um padrão de configuração e modo de execução por linha de comando para todos os sistemas operacionais. Nas três plataformas utilizadas, tornaram-se notórias as grandes diferenças de utilização. O maior problema foi com o aplicativo SCP para transferência de arquivos com criptografia.

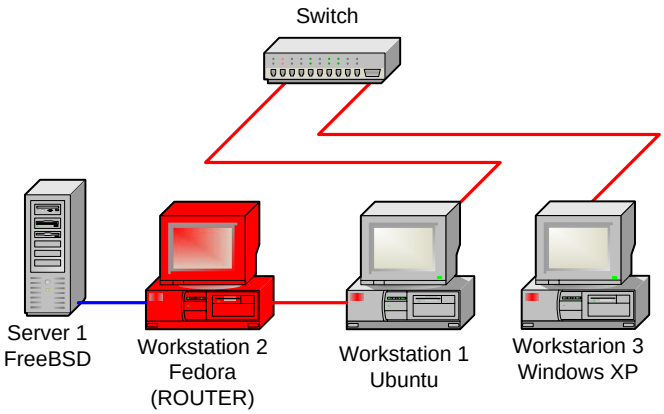
Importa mencionar a dificuldade de se encontrar documentação funcional dos sistemas para IPv6. Para que os testes fossem realizados com sucesso, centenas de sites foram pesquisados a fim de se encontrar a solução para cada problema ocorrente. Nas referencias bibliográficas, estão apresentados apenas aqueles que contém alguma alternativa eficaz. Com exceção do serviço SSH/SCP que não são nativos no Windows, o site da Microsoft é o que possui uma melhor documentação de configuração dos sistemas por eles desenvolvidos. O ***TechNe***t apresenta as informações de maneira clara e bem estruturada (ver referencias bibliográficas).

6.3. LABORATÓRIO 2

6.3.1. CENÁRIO:

Este laboratório consiste de duas redes IPv6 roteadas através de um servidor roteador utilizando o sistema QUAGGA conforme apresentado na figura 58.

Figura 58 - Diagrama de interligação de equipamentos do laboratório de testes 2



Fonte: Própria

6.3.2. OBJETIVO:

Este laboratório tem dois objetivos: apresentar o funcionamento dos serviços de encaminhamento de pacotes através de um roteador e confirmar que é necessário a atualização dos *firmwares* e *softwares* dos equipamentos que atuam como roteadores de pacotes em nível 3

6.3.3. PROCEDIMENTOS:

Aqui vamos descrever os passos de montagem deste laboratório, no que se refere à instalação e configuração dos equipamentos envolvidos.

6.3.4. TABELA DE CONFIGURAÇÕES

Os equipamentos servidores e estações receberão as seguintes configurações:

Tabela 14 – Endereçamento dos *hosts do LAB 2*

Nome da Estação	Descrição	End. Ipv6	End. Ipv4
Windows-WSWin300	WorkStation MS Win XP pro	fec0:a1::a003:300/32	----------
Ubuntu-WSLin200	WorkStation Linux Ubuntu 6	fec0:a1::a003:200/32	----------
Fedora-WSLinROT	WorkStation Linux Fedora 5	fec0:a1::a003:100/32 fec0:b1::b003:100/32	----------
FreeBSD-SRVBSD500	Servidor FreeBSD 5	fec0:b1:b003:200/32	----------

Fonte: Própria

6.3.5. CONFIGURAÇÃO DO EQUIPAMETO LINUX COMO ROTEADOR:

Para configurar este laboratório, será utilizado o software QUAGGA rodando no SO Fedora. Este sistema funciona como um roteador completo permitindo aplicação de *Access List* (ACL), roteamento estático, e dinâmico com BGP, OSPF ou RIP.

Atualmente, na versão utilizada aqui (1.1.1.1), já possui suporte a IPv6.

Como o IPv4 não interpreta endereços IPv6 de forma direta (somente com configurações adequadas que veremos mais à frente), o sistema QUAGGA, que já possui uma interface própria para reconhecimento de endereços IPv6, será configurado com os arquivos de configuração específicos desta versão.

- Proceda a instalação do QUAGGA na estação Fedora, utilizando o instalador de pacotes do sistema:
  - Certifique que seu equipamento está com as duas interfaces de rede ativas utilizando o comando *mii-tool*;
  - Vá para a interface gráfica (<ctrl>+<alt>+<f7> ou *startx* caso a interface X do SO ainda não esteja carregada);
  - Logue como root;
  - Vá em “Aplicações”, “add/remove software”;
  - “Click” na lupa e digite “Quagga” <enter>
  - Selecione o pacote e “click” em <aplicar>
  - Configurar as interfaces de rede do SO com os repectivos endereços

Configuração de roteamento estático utilizando o Quagga

Editar arquivo /etc/quagga/ zebra.conf e acrescentar:

hostname RouterIPv6-1
password teste
enable password teste

Digitar os comandos:

 # *service zebra start*

 # *telnet localhost zebra* (obs.: Não funciona com ::1)

password
RouterIPv6-1> enable
password
RouterIPv6-1# configure terminal
RouterIPv6 (config)# interface eth0 (configurar o interface eth0)
RouterIPv6-1 (config-if)# ipv6 address <endereço IPv6>/<máscara> (endereço da interface ligada diretamente à rede)

RouterIPv6-1 (config-if)# ipv6 nd prefix <endereço da rede configurada anteriormente>/<máscara> (prefixo de rede para anuncio de rotas)

RouterIPv6-1 (config-if)# ipv6 nd suppress-ra (desativa o envio de anuncio de rotas pois estamos utilizando rotas estáticas)

RouterIPv6-1 (config-if)# exit (voltar ao menu anterior)
RouterIPv6-1 (config)# interface eth1 (configurar o interface eth1)
RouterIPv6-1 (config-if)# ipv6 address <endereço IPv6>/<máscara> (endereço da interface ligada diretamente à rede)

RouterIPv6-1 (config-if)# ipv6 nd prefix <endereço da rede configurada na linha anterior>/<máscara> (prefixo de rede para anuncio de rotas)

RouterIPv6-1 (config-if)# ipv6 nd suppress-ra (desativa o envio de anuncio de rotas pois estamos utilizando rotas estáticas)

RouterIPv6-1 (config-if)# exit (voltar ao menu anterior)
RouterIPv6-1 (config)# ipv6 forwarding (activa o modo de encaminhamento do router)

RouterIPv6-1# write file (grava as configurações no "startup-config" – "zebra.conf")
RouterIPv6-1# exit (nesta fase, termina a sessão telnet)

Editar o arquivo "*forwarding"* mudando alterando o valor 0 para 1

 # echo "1" > /proc/sys/net/ipv6/conf/all/forwarding

- Lembrar de configurar o *gatway default* do *host* com o endereço da interface do roteador que está conectada na mesma rede do *host.* (apesar de ter sido apresentado no referencial teórico de que não haveria mais necessidade deste tipo de configuração, no laboratório não funcionou sem a mesma)

    - **Linux**

$ route –A inet6 add default gw <endereço da interface do roteador conectada à rede do host que está sendo configurado>

    - **FreeBSD**

 # route –n add –net –inet6 <endereço da rede/mascara> -inet6 <endereço do gateway>

- Para verificar a rota criada, proceda o comando:

  - **Linux**

$ route –A inet6 -n

  - **FreeBSD**

#netstat –r

Figura 59 – Tabela de roteamento do equipamento Ubuntu

```
root@Ubuntu-WSLin200:~# route -A inet6 add default gw fec0:a1::a003:100
root@Ubuntu-WSLin200:~# route -A inet6 -n
Tabela de Roteamento IPv6 do Kernel
Destination                                 Next Hop
    Flags Metric Ref    Use Iface
::1/128                                     ::
    U     0      0        1 lo
fec0:a1::a003:200/128                       ::
    U     0      783       1 lo
fec0:a1::20c:29ff:fee6:f4d7/128             ::
    U     0      10       1 lo
fec0:a1::/64                                ::
    U     256    19       0 eth1
ff00::/8                                    ::
    U     256    0        0 eth1
::/0                                        fec0:a1::a003:100
    UG    1      599       0 eth1
::/0                                        fe80::20c:29ff:fe16:d2fc
    UGDA  1024   0        0 eth1
root@Ubuntu-WSLin200:~# _
```

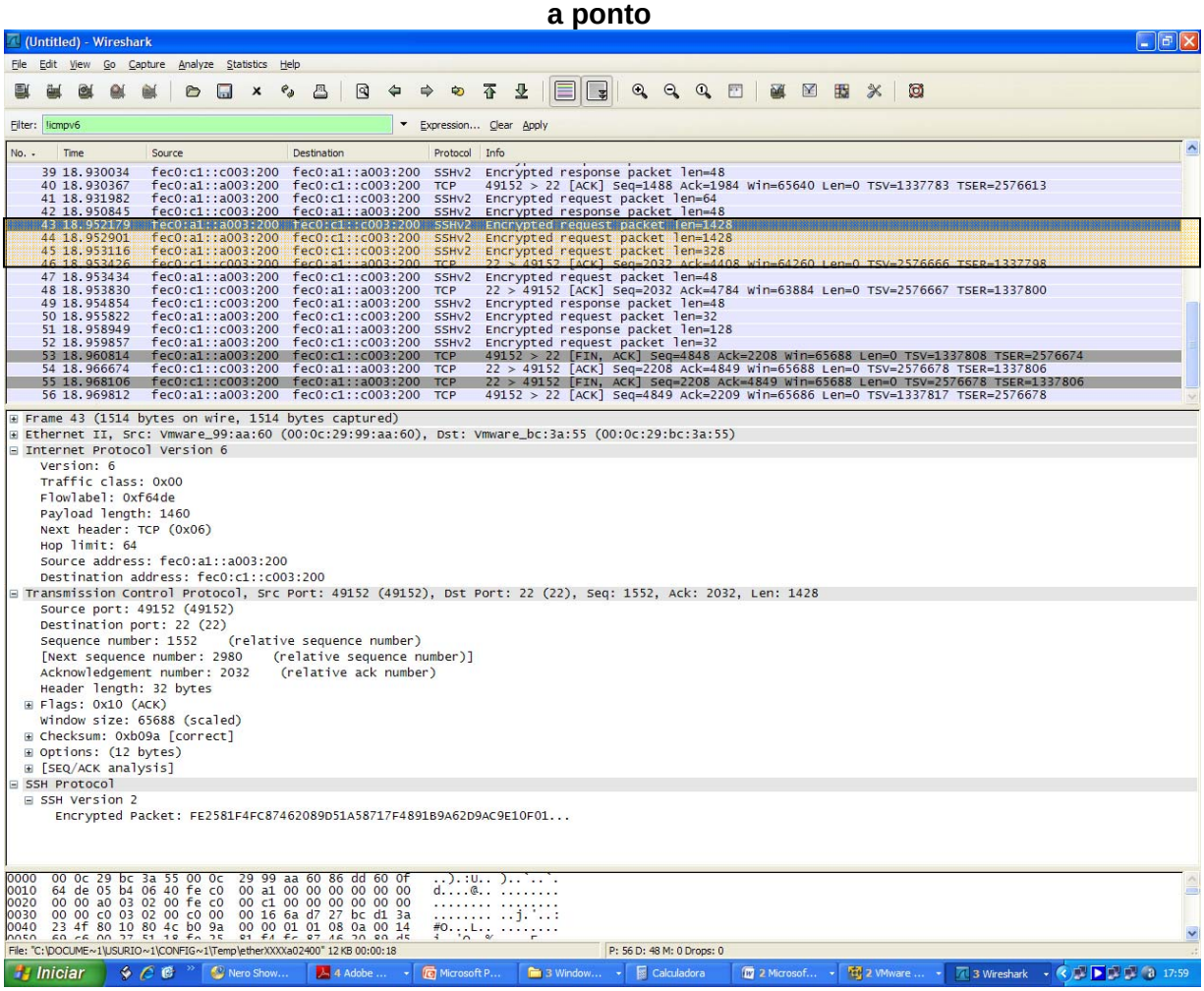
Fonte: Própria

- A fim de demonstrar que fragmentação do pacote é realizada na estação, a proposta era configurar dois roteadores interligados com um MTU menor que 1500 e fazer análise com o WIRESHARK nas extremidades externas da rede e provar que o host A envia o datagrama para B no tamanho do menor MTU. Porém, por limitações do *software* VMWare, o qual está sendo utilizado para os laboratórios, quando se tentou reduzir o MTU de um dos lados do roteador, as redes param de se comunicar.
- A figura na seqüência apresenta a fragmentação e um datagrama de 3146 *Bytes* passando por um canal de comunicação com MTU de

1500 *Bytes*. Conforme apresentado na figura, o tamanho do datagrama é de 1428. Este valor é um pouco menor que o indicado no item 5.3.5 (que é de 1496), mas isso se deve ao fato de que os dados coletados possuem criptografia, o que consomem a diferença.

A figura 60 apresenta a transferência de um arquivo de 3.64MB utilizado para se aferir a fragmentação do pacote de dados. Pela figura, observamos o comportamento de transferência. Isso pode ser confirmado pela quantidade de datagramas de *request* em relação aos de confirmação (*ack*). Não foi possível visualizar o campo *fragment offset* apresentado no item 5.4.5 do referencial teórico, porém no cabeçalho TCP, pode ser visto os campos de *sequence number*, *next sequence number* e *acknowledgement number* que indicam a seqüência de datagramas transmitidos e a quantidade de confirmações.

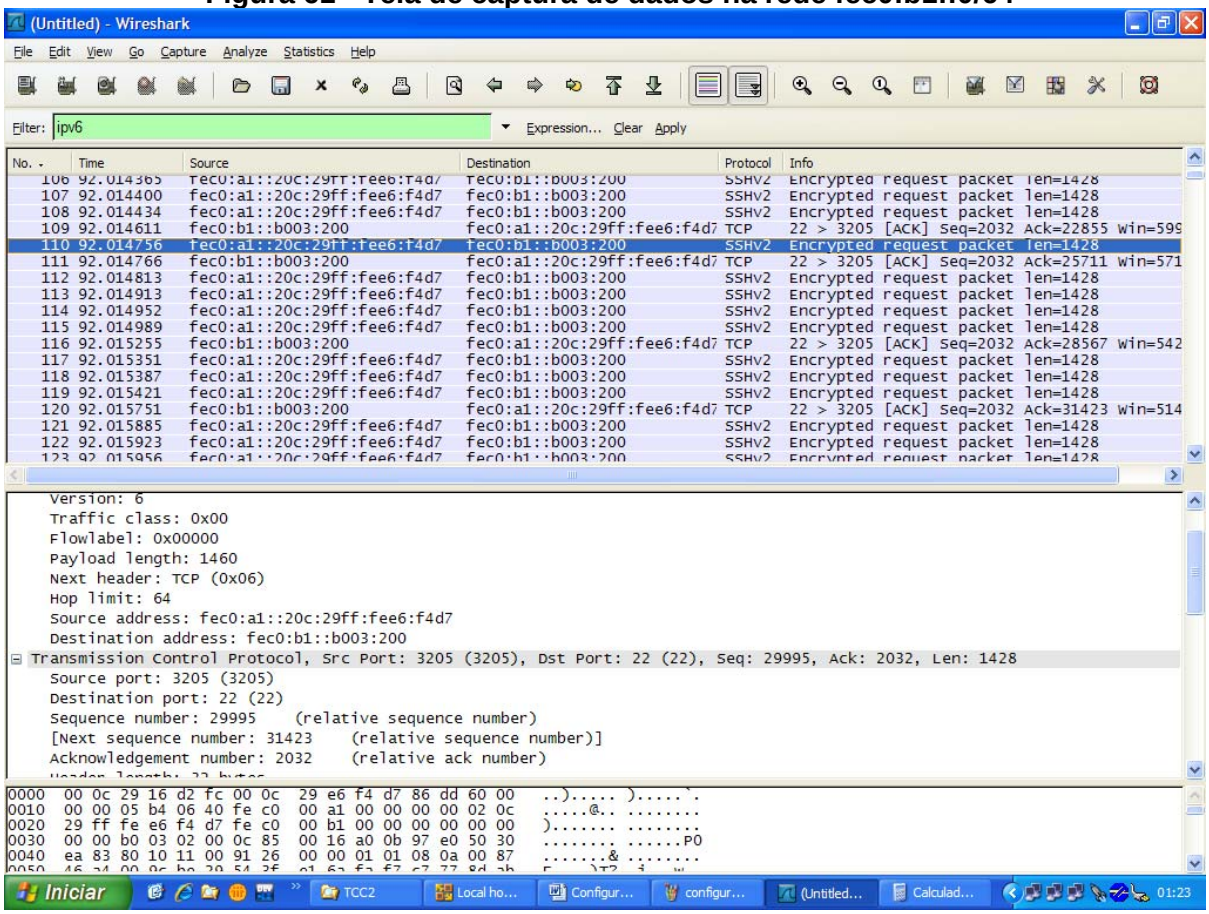
Figura 60 - Transmissão do arquivo teste.html de 3.640 Bytes com MTU de 1500 ponto a ponto



Fonte: Própria

Um detalhe a ser observado na coleta de dados é que o endereço de origem, pertence à rede fec0:b1::0, porém não é o configurado no equipamento roteador, mas um novo atribuído pelo sistema com a parte de seqüencial do endereço MAC. Isso se chama “endereço automático” que pode ser desabilitado na configuração permanente da interface apresentada no item 6.2.5.

Figura 61 - Tela de captura de dados na rede fec0:b1::0/64



Fonte: Própria

6.3.6. CONCLUSÃO DO SEGUNDO LABORATÓRIO:

Com este laboratório, verificou-se que é possível se criar roteamento entre redes IPv6. Vale ressaltar que a fragmentação dos pacotes em datagramas agora é realizada pela estação, conforme apresentado na teoria. Por limitações do *software* de roteamento utilizado que não permitiu a redução do tamanho do MTU em um dos barramentos do roteador, não foi possível demonstrar tal funcionamento.

6.4. LABORATÓRIO 3

Neste laboratório, serão tratadas as questões de compatibilidade entre as pilhas IP v4 e v6. Tais compatibilidades estão previstas nas RFCs *Stateless* IP/ICMP *Translation* (SIIT) 2765, *Network Address Trasnlation*/*Protocol Translation* (Nat-PT) 2766, *Transport Relay Translation* (TRT) 3142. Os testes seguem à seguinte ordem:

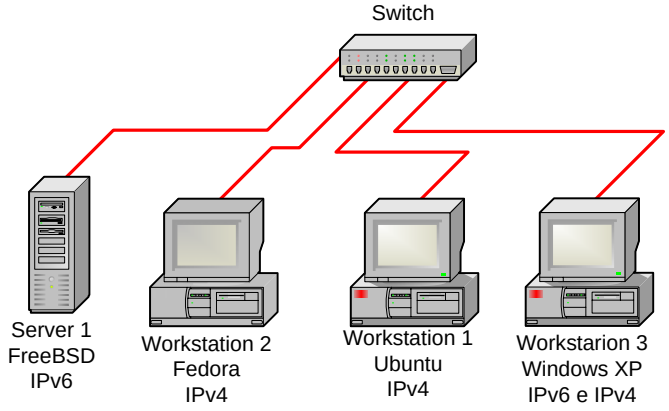
- **Cenário 1:** Equipamentos com pilha dupla – Neste cenário, teremos equipamentos com IPs da versão 4 e 6 simultaneamente conversando com outros somente 4 e somente 6.

- **Cenário 2:** Uma rede com dois roteadores IPv4 funcionando como túnel para uma comunicação IPv6 “tunelada” (6to4).

6.4.1. CENÁRIO 1

O objetivo deste laboratório é apresentar a possibilidade de se ter endereços IPv4 e IPv6 distintos configurados na mesma interface de rede (*Network Interface Card* - NIC), chamado tecnicamente de Pilha Dupla. Isso permite compatibilizar sistemas a ambas as versões do protocolo IP, bem como proporciona uma maneira bem adequada para um processo de migração.

Figura 62 - Diagrama de interligação de equipamentos do laboratório de testes 3-1



Fonte: Própria

6.4.1.1. Objetivo:

Configurar um equipamento com dois endereços IP de versões diferentes comunicando-se com equipamentos endereçados em IPv4 ou IPv6 simultaneamente.

6.4.1.2. Procedimentos:

Aqui vamos descrever os passos de montagem deste laboratório, no que se refere à instalação e configuração dos equipamentos envolvidos.

Tabela 15 – Endereçamento dos *hosts do LAB 3-1*

Nome da Estação	Descrição	End. Ipv6	End. Ipv4
Fedora-WSLin100	WorkStation Linux Fedora 5	----------	192.168.1.2/24
Ubuntu-WSLin200	WorkStation Linux Ubuntu 6	fec0:100::10/64	----------
Windows-WSWin300	WorkStation MS Win XP pro	fec0:100::20/64	192.168.1.1/24

Fonte: Própria

6.4.1.3. Configuração dos equipamentos:

Neste cenário, as configurações são as mesmas realizadas no laboratório um, ou seja, configuração de endereços nas interfaces dos equipamentos. O que difere é a colocação de dois endereços IP em um único equipamento, no caso, o Windows XP. Poderia ser em qualquer um dos apresentados acima.

Para configurar o Windows, proceder da seguinte maneira

C:> netsh interface ipv6 add address “nome da interface” <endereço IPv6>
C:> netsh interface ipv4 add address “nome da interface” <endereço ipv4>

Para configuração das máquinas Linux, proceder o comando;

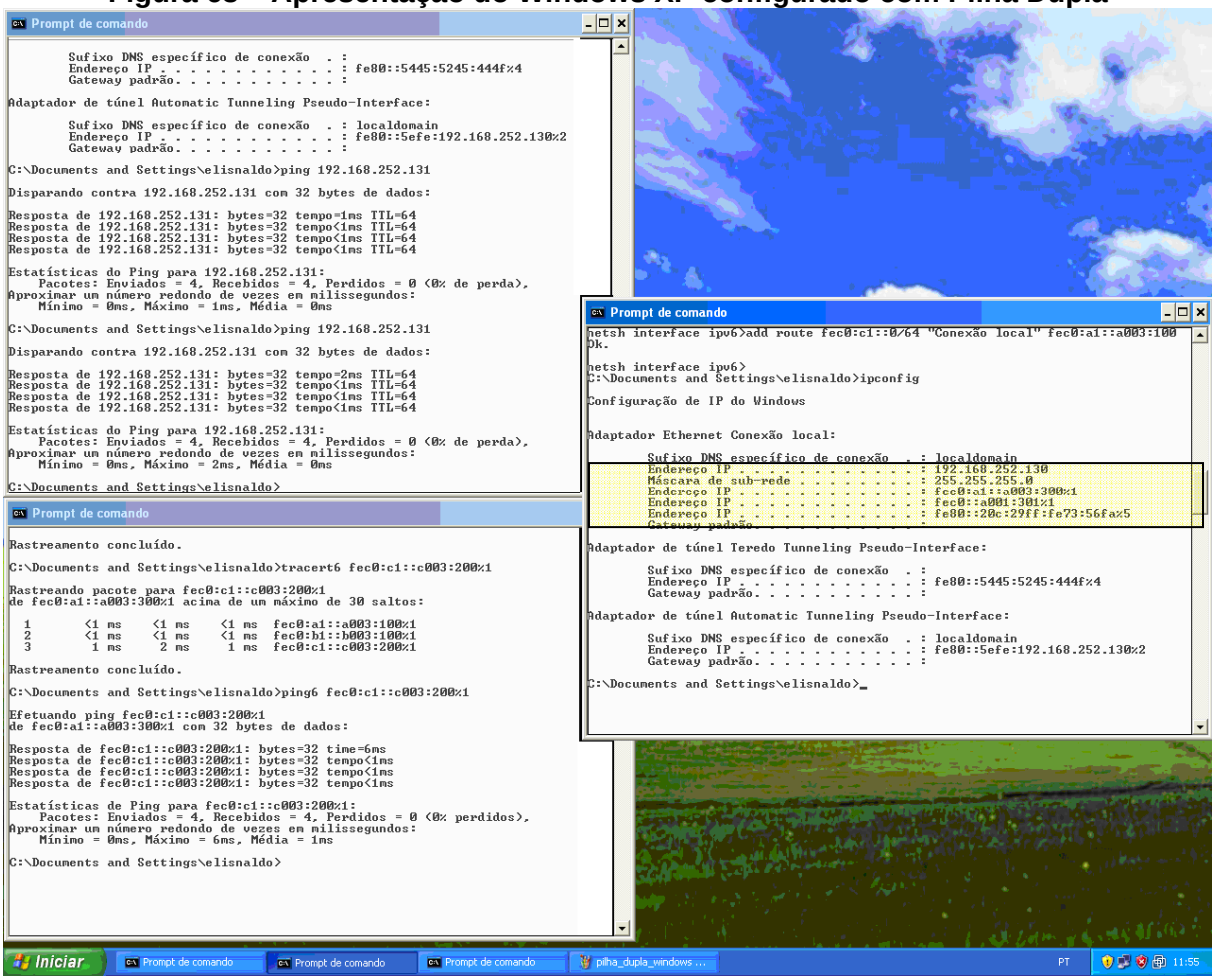
 # ifconfig eth0 inet6 add <endereço ipv6>/<máscara de rede>

 # ifconfig eth0 <endereço ipv4>/<máscara de rede>

6.4.1.4. Procedimentos de teste:

O teste se resume em executar um comando PING para cada estação com os respectivos endereços conforme apresentado na próxima figura:

Figura 63 – Apresentação do Windows XP configurado com Pilha Dupla



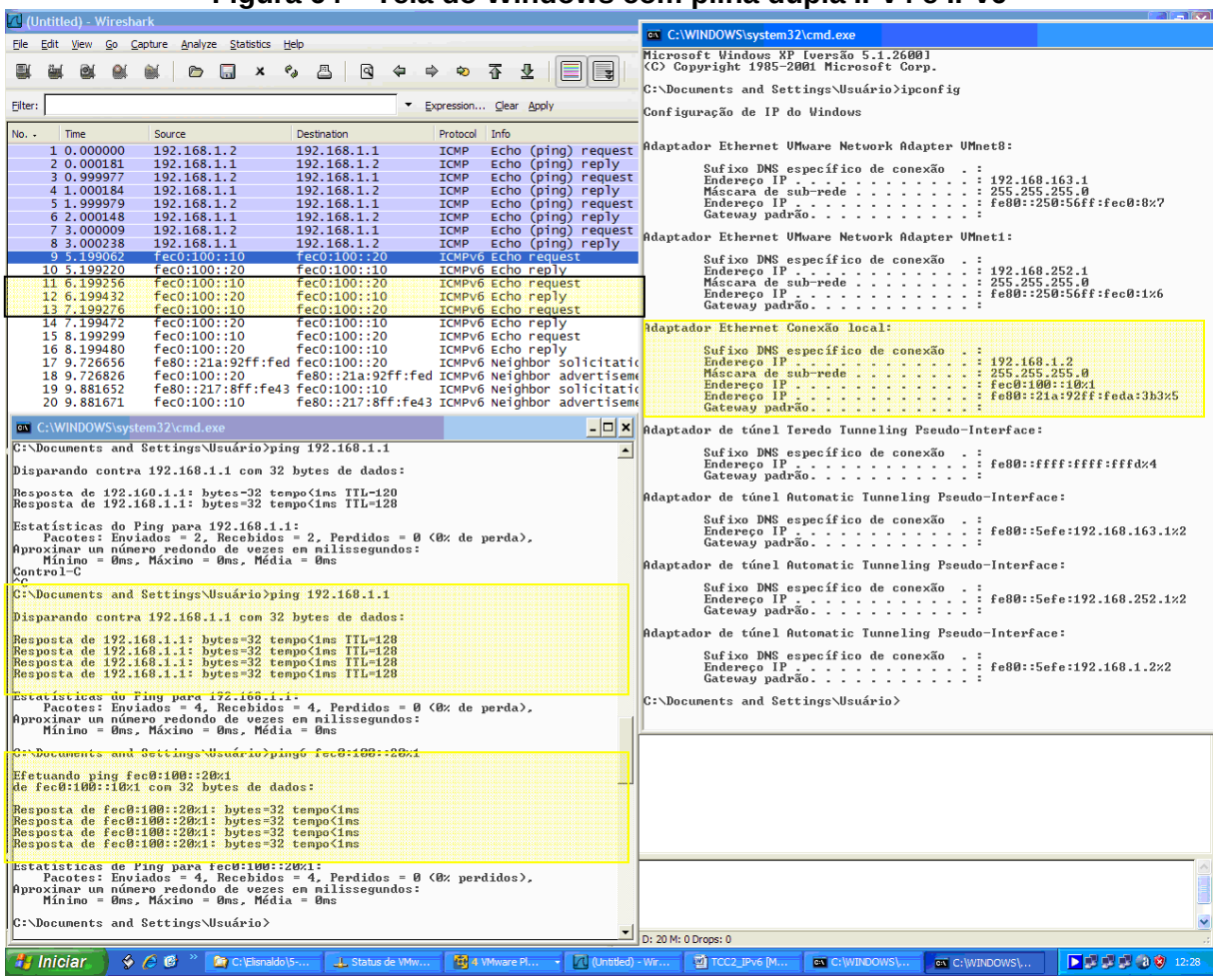
Fonte: Própria

Como podemos observar na janela da direita da figura 63, o equipamento apresenta endereços IPv6 e IPv4 respectivamente na mesma interface “Conexão Local”. Nas duas janelas da esquerda, estão apresentados comandos de “ping” e “tracert” para os endereços IPv4 e IPv6 respectivamente.

Obs.: Os endereços vitos na figura diferem dos apresentados na tabela devido a terem sido configurados em ambiente diferente.

Na próxima figura, veremos o comando PING sendo executado e coletado pelo WIRESHARK.

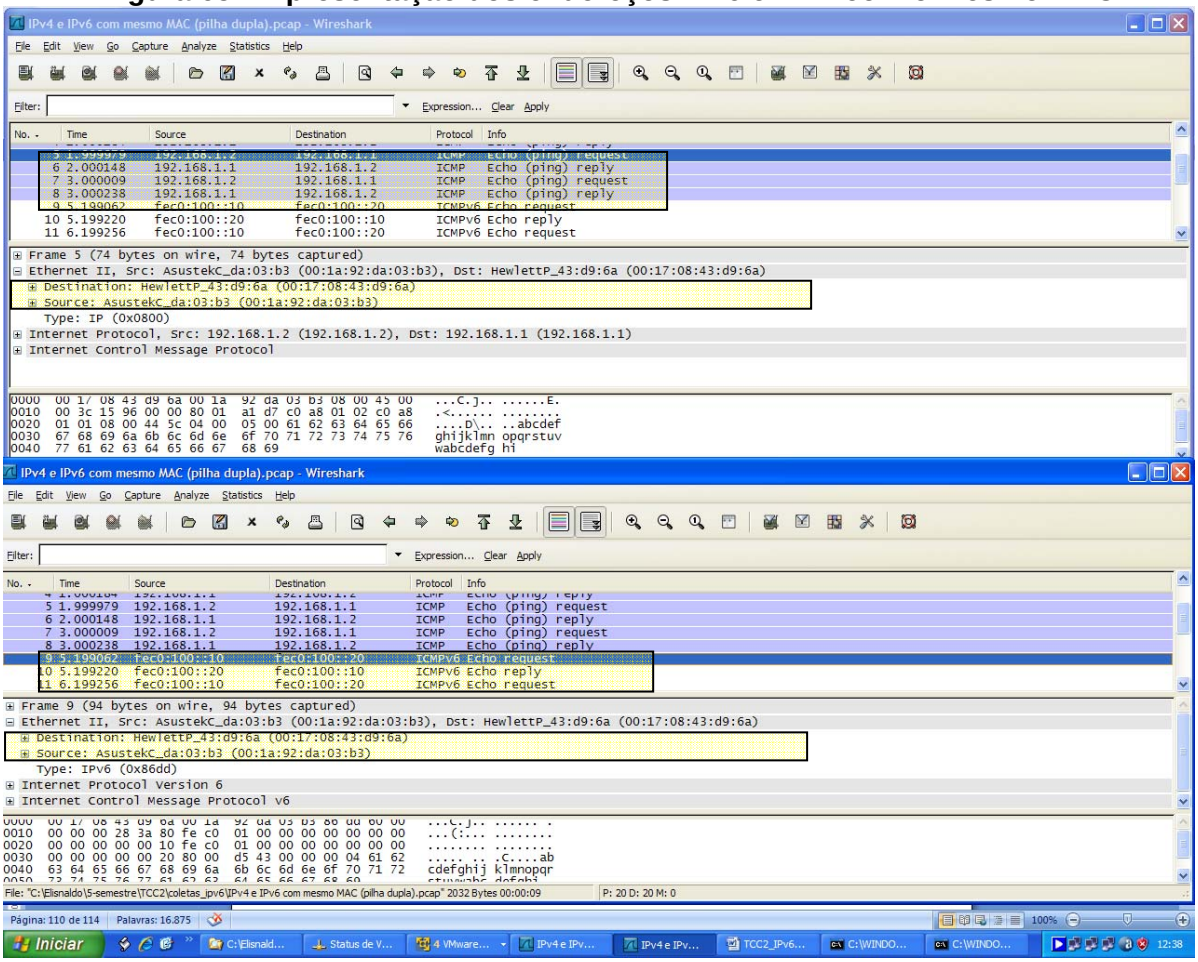
Figura 64 – Tela do Windows com pilha dupla IPv4 e IPv6



Fonte: Própria

Nesta figura, temos ao fundo, a tela de captura do WIRESHARK coletando os pacotes de *ping* executados na janela do DOS da esquerda. A janela da direita apresenta os endereços de rede da interface “Conexão Local”.

Figura 65 – Apresentação dos endereços IPv6 e IPv4 com o mesmo MAC



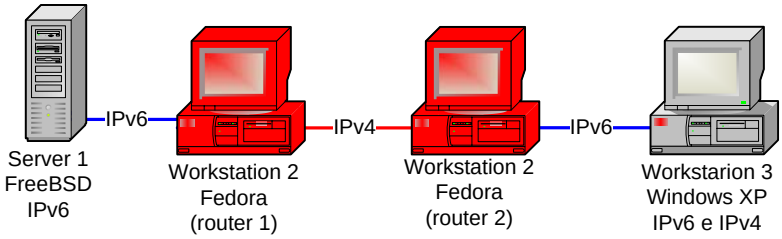
Fonte: Própria

Na figura acima, temos apresentada a mesma tela de captura da figura anterior, porém aberto em duas janelas a fim de apresentar que os IPs estão com o mesmo MAC conforme destacado.

6.4.2. CENÁRIO 2:

O objetivo deste laboratório é apresentar a possibilidade de se interligar duas redes IPv6 utilizando um canal IPv4 com a utilização da técnica de “Tunelamento”.

Figura 66 - Diagrama de interligação de equipamentos do laboratório de testes 3-2



Fonte: Própria

6.4.2.1. Objetivo:

Interligar duas redes IPv6 utilizando dois roteadores Linux IPv4.

6.4.2.2. Procedimentos:

Aqui vamos descrever os passos de montagem deste laboratório, no que se refere a instalação e configuração dos equipamentos envolvidos.

Tabela 16 – Endereçamento dos *hosts do LAB 3-2*

Nome da Estação	Descrição	End. Ipv6	End. Ipv4
Fedora-Router100	WorkStation Linux Fedora 5	----------	192.168.1.2/24 10.1.1.1/16
Fedora-Router200	WorkStation Linux Fedora 5	----------	192.168.1.2/24 10.2.1.1/16
FreeBSD	WorkStation MS Win XP pro	fec0:100::20/64	----------
FreeBSD	WorkStation MS Win XP pro	fec0:100::20/64	----------

Fonte: Própria

6.4.2.3. Configuração dos equipamentos:

Para configuração deste cenário, foram utilizados dois roteadores estáticos com Fedora e habilitado o serviço de tunelamento automático configurado da seguinte maneira;

Configurar as interfaces de rede:

 # ifconfig eth0 inet6 add <endereço ipv6 da rede A>/<mascara de subrede>
 # ifconfig eth1 <endereço ipv4 do link de ligação>/<máscara de subrede>

Configurar o túnel IPv4 na interface virtual “sit1”:

 # ip tunel add sit1 mode sit remote <endereço ip do roteador remoto ao túnel>
 # ip link set dev sit1 up

Habilitando o roteamento para o túnel:

 # echo 1 > /proc/sys/net/ipv6/conf/all/forwarding

Criando rota para a rede do outro lado do túnel:

 # route –A inet6 add <endreço ipv6 da rede do outro lado do túnel> sit1

Este procedimento deve ser realizado nos dois roteadores sempre apontando o endereço do túnel para o roteador remoto

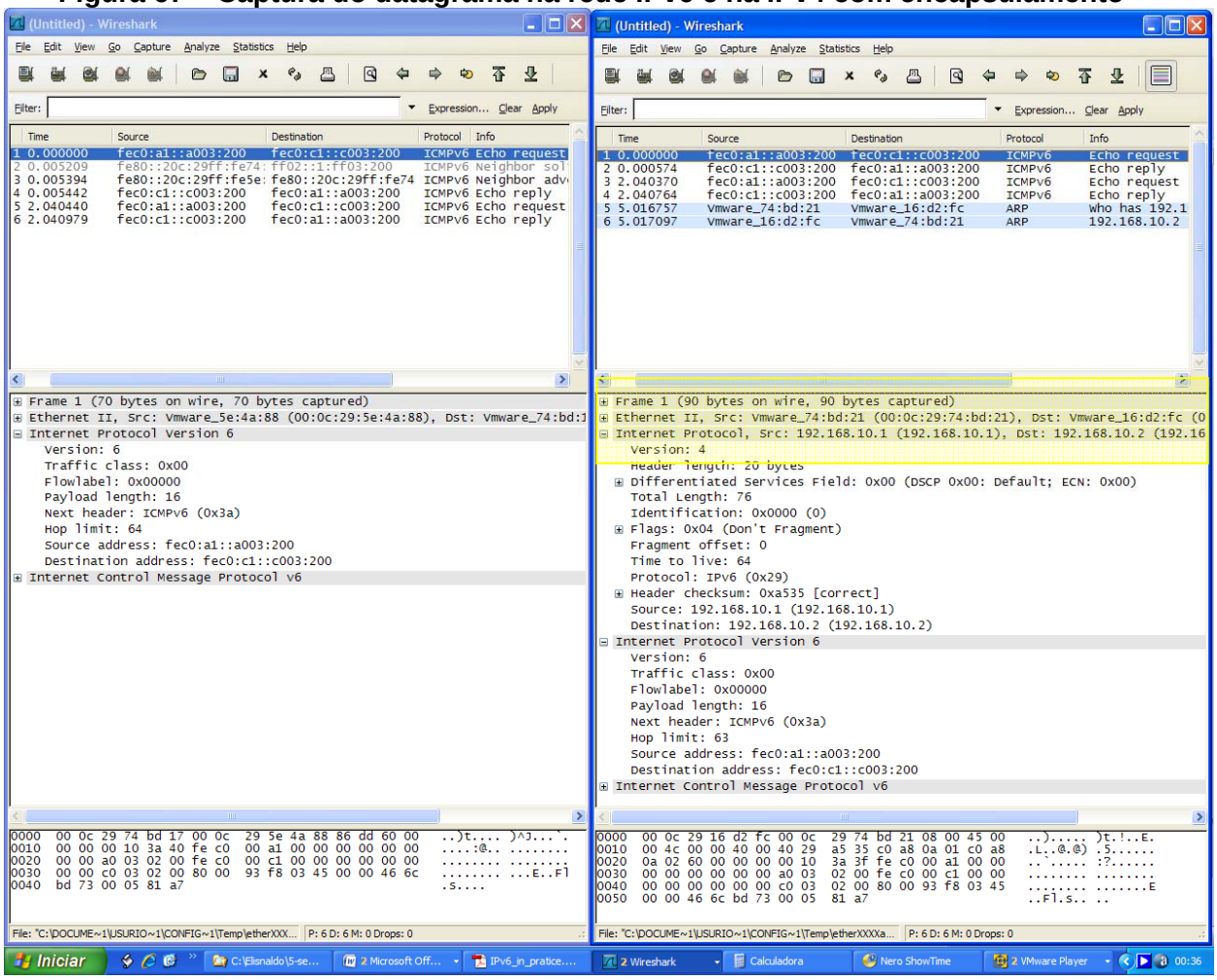
Nas estações das redes IPv6 configurar a rota padrão para o roteador que provê o túnel IPv4

6.4.2.4. Procedimentos de teste:

Para demonstração do funcionamento foi realizado o comando PING da estação IPv6 da rede A para a outra da rede B e coletado datagramas na rede IPv6

e no link IPv4 apresentando sua passagem sem e com “encapsulamento” conforme apresentado na figura abaixo

Figura 67 – Captura do datagrama na rede IPv6 e na IPv4 com encapsulamento



Fonte: Própria

6.4.3. CONCLUSÃO DO TERCEIRO LABORATÓRIO:

Com este laboratório, ficou comprovado a possibilidade de se manter as duas versões do protocolo IP convivendo em paralelo.

Além destes modos, conforme apresentado no item 5.4, possuem outras maneiras de se manter compatibilidades entre as versões que não foram apresentadas aqui mas podem ser utilizadas desde que, conforme já mencionado, se teste exaustivamente antes de implantar em uma rede de produção.

7.CONCLUSÃO

Apesar das grandes vantagens com a implementação da versão 6 do protocolo IP encontradas neste trabalho, muita cautela antes de se aventurar por este caminho. O serviço ainda está em desenvolvimento e grandes mudanças ainda serão implementadas à medida que a grande massa de usuários iniciar a utilização. Os *hardwares* e *softwares* atuais, já estão sendo desenvolvidos com compatibilidade para as duas versões. Cisco, 3Com, Foudry, e outros, já desenvolvem roteadores e equipamentos que atuam com pilha dupla. A maioria das distribuições Linux e o próprio WindowsXP, 2003 e Vista já oferecem suporte para esta versão, apesar da Microsoft ainda não permitir configuração somente IPv6 sem a instalação do pacote IPv4 conforme publicado no site http://www.microsoft.com/brasil/technet/centralwindows/artigos/ipv6over.mspx.

> O uso do IPv6 é suportado atualmente apenas quando o IPv4 também é instalado. As internetworks TCP/IP são suscetíveis a uma variedade de ataques possíveis, de ataques passivos (como a escuta privada) a ataques ativos (como as ataques de denial-of-service) (MICROSOFT, 2007).

Mesmo com as diferenças encontradas entre os formatos referentes a cada versão, ficou constatado que há a possibilidade de se integrar redes de diferentes versões.

Com a parte prática, pode ser observado que se trata de uma implementação ainda em desenvolvimento em todas as plataformas. Isto pode ser comprovado pela dificuldade em se encontrar documentação clara sobre os modos de configuração em todos os sistemas operacionais.

Já existem publicações específicas sobre o assunto conforme pode ser observado na relação bibliográfica deste trabalho, porém, muitas informações apresentadas, dependem de um melhor estudo do protocolo e dos métodos de funcionamento do mesmo.

Um bom estudo das RFCs pode trazer melhores esclarecimentos sobre as dificuldades encontradas nos procedimentos. Nem todas as funcionalidades do novo

protocolo IP foram testadas neste trabalho a fim de evitar um prolongamento das atividades mediante às dificuldades encontradas para implantação.

Apesar de não ter sido possível a implementação de algumas funcionalidades apresentadas no referencial teórico, em função de limitações do laboratório virtual (utilizando a ferramenta VMWare) para realização das pesquisas necessárias, o estudo teve seu papel em apresentar que existe uma nova possibilidade de endereçamento que esta permite de algum modo, uma migração lenta e bem estruturada.

Como autor deste trabalho, senti-me realizado por desenvolver esta pesquisa sobre um assunto que poderá, nos próximos anos, encontrar espaço para se impor e permitirá aos que possuírem domínio no assunto, uma posição de destaque no mercado de consultoria para migração.

REFERENCIAL BIBLIOGRÁFICO

COMMER, Douglas E..**Interligação de redes com TCP/IP**. Vol.1, 5 ed. tradução Daniel Vieira. Rio de Janeiro: Campus,2006.

MILLER, Mark A..**Implementing IPv6: Migrating to the Next Generation Internet Protocols**. New York,USA:M&T Books,1998.

MURHAMMER, Martim W. et. all. **TCP/IP Tutorial e Técnico**. São Paulo: Makron Books,2000

NAUGLE, Matthew G.. **Guia ilustrado do TCP/IP: Uma anatomia completa das redes TCP/IP e do conjunto de protocolos IP em um formato de referência rápida**. São Paulo:Berkley,2001.

STOCKEBRAND,Benedikt. **IPv6 in Practice A Unixer's Guide to the Next Generation Internet.** New York, USA: Springer-Verlag Berlin Heidelberg 2007

THOMAS, Stephen A..**IPng and TCP/IP Protocols: Implementing the Next Generation Internet** .Canadá,USA:Wiley Computer Publishing, 1996.

ZHOU, Rui et. all. **IPV4 TO IPv6 Migration**. Freiburg: Institute for Computer Science . Wien: Albert-Ludwigs-University Wien, 2004 Disponível em http://www.ks.uni-freiburg.de/download/studienarbeit/WS03/IPv4-IPv6-Migration.pdf: acessado em:16/06/2007

SCHUH, Sylvia. **Migrating Small Business Networks To IPv6**.Austria, 2006. 287p. Magister der Sozial- undWirtschaftswissenschaften, Universität Wien:2006. Disponível em: http://stud3.tuwien.ac.at/~e0000195/DA/DA_gesamt_v1.01.pdf: acessado em: 16/06/2007

RFC-1825 ***Security Architecture for the Internet Protocol*** Disponível em http://www.ietf.org/rfc/rfc1825.txt acessado em: 16/06/2007

RFC-1884 ***IP Version 6 Addressing Architecture*** Disponível em http://www.ietf.org/rfc/rfc1884.txt: acessado em: 16/06/2007

RFC-2765 ***Stateless IP/ICMP Translation*** **(SIIT)** Diponível em http://www.ietf.org/rfc/rfc2765.txt?number=2765: acessado em: 16/06/2007

RFC-2766 ***Network Address Trasnlation/Protocol Translation*** **(*Nat-PT)*** Disponível em http://www.ietf.org/rfc/rfc2766.txt?number=2766: acessado em: 16/06/2007

RFC-3142 ***Transport Relay Translation*** **(TRT)** Disponível em http://www.ietf.org/rfc/rfc3142.txt?number=3142: acessado em: 16/06/2007

OUTRAS FONTES

DHCPv6. Disponível em
http://thesource.ofallevil.com/technet/prodtechnol/windowsserver2003/pt-pt/library/ServerHelp/5a528933-a78d-4588-8aa1-b158957ba2d5.mspx?mfr=true
Acessado em 16/06/2007

Neighbor Discovery Protocol. Disponível em
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/pt-br/library/ServerHelp/55049e22-2aa7-44c9-ab4b-97cf98b96ad2.mspx?mfr=true
Acessado em 16/06/2007

Configuração do ambiente MS-Windows. Disponível em
http://technet2.microsoft.com/WindowsServer/pt-BR/Library/85a03ee2-9db3-427a-bd1e-5ed5c76002fd1046.mspx?mfr=true
Acessado em 16/06/2007

Restrições do Windows XP e 2003 em relação ao IPv6
http://www.microsoft.com/brasil/technet/centralwindows/artigos/ipv6over.mspx
Acessado em 16/06/2007

Comando SCP no FreeBSD
http://www.enterprisenetworkingplanet.com/netsp/article.php/3634596
Acessado em 16/06/2007

Configurando o IPv6 no FreeBSD. Disponível em
http://www.free.bsd.com.br/noticia.php3?CAD=1&NOT=139
Acessado em 16/06/2007

Comandos do protocolo FTP
http://www.nsftools.com/tips/RawFTP.htm. Disponível em
Acessado em 16/06/2007

Como verificar a tabela de vizinhos do IPv6. Disponível em
http://www.linux.com/howtos/Linux+IPv6-HOWTO/x1181.shtml
Acessado em 16/06/2007

Guia para iniciantes no FreeBSD. Disponível em
http://freebsd.ag.com.br/sessao8_5.html
Acessado em 16/06/2007

Comandos de FTP para IPv6. Disponível em
http://www.fccn.pt/files/documents/D2.10.PDF?947cda2253a1dc58fe23dc95ac31cbed=a953313be5c7e0090d9f54bf5cae09b6
Acessado em 16/06/2007

Configurações e comandos IPv6 para Linux. Disponível em
http://www.enterprisenetworkingplanet.com/netsp/article.php/3634596
Acessado em 16/06/2007

Roteamento IPv6 com Zebra. Disponível em
http://www.rnp.br/newsgen/0207/ipv6_ospfv3_bgp4.html
Acessado em 16/06/2007

Manual do sistema WireShark. Disponível em
http://www.wireshark.org/download/docs/user-guide-a4.pdf
Acessado em 16/06/2007