EXPLORANDO OS ENDEREÇOS PRIVADOS (ULA) DO IPV6

Lucas Venezian Povoar[1]
Emerson Barea[2]

Resumo: A nova geração do protocolo de internet, o IPv6, é caracterizada pela implementação de novos recursos e o aprimoramento de outros já existentes em sua versão anterior, o IPv4. Apesar da relação aparente pela nomenclatura, não podemos tratar o IPv6 apenas como uma atualização do IPv4, visto que seu desenvolvimento se deu sob uma nova necessidade de mercado e um contexto tecnológico diferente. Um dos pontos evolucionários do IPv6 é a variedade relativamente grande de tipos de endereços que podemos atribuir às interfaces, tornando o assunto extenso e complexo. Um dos elementos contidos no escopo do IPv6 é o *Unique Local* IPv6 *Unicast Address* (ULA), que contêm diversos detalhes a serem considerados, tais como: componentes do endereço, algoritmos de geração do identificador global, probabilidade de colisão, protocolos de atribuição de endereços às interfaces e suas vantagens e desvantagens. Portanto, este estudo tem a finalidade de abordar essa gama de assuntos relativos ao endereço ULA do IPv6, além de expor uma nova ferramenta, denominada UniLaGenJ, com a finalidade de gerar prefixos globais dos endereços privados, e junto a isso ilustrar sua aplicação com estudos de casos, que também expõem a utilização do protocolo de descoberta de vizinhança (*Neighbourhood Discovery*).
Palavras-Chave: Protocolo de Internet, Unique Local IPv6 Unicast Address, IPng

Abstract: *The new generation of Internet Protocol, IPv6, is characterized by the implementation of new features and improving existing ones in the previous version, the IPv4. Despite the apparent nominal relationship, we cannot assume that IPv6 is only an upgrade of IPv4, since its development took place under a new market needed and a different technology context. One of the evolutionary points of IPv6 is the relatively large variety of types of addresses that we can assign to the interfaces, making it an extensive and complex context. Thus, one of the elements contained in the scope of the v6 is*

---

[1]Universidade Estadual Paulista "Júlio de Mesquita Filho" Campus de Ourinhos (UNESP), lucas@ourinhos.unesp.br.
[2]Professor da Faculdade de Tecnologia de Ourinhos (FATEC), emerson.barea@fatec.sp.gov.br.

the Unique Local IPv6 Unicast Address, which contains many details to be considered, such as: address components, algorithms of global identification generation, collision probability, address assignment interfaces protocols, their advantages and disadvantages. Therefore, this study aims to approach this range of issues relating to IPv6 ULA address, and expose a new tool, called UniLaGenJ, aiming to generate global prefixes of private addresses, and along with that illustrate their application in case studies, which also expose the use of neighborhood discovery protocol.
Keywords: *Internet Protocol, Unique Local IPv6 Unicast Address, IPng*

1. Introdução

Desde primeiro de janeiro de 1983, quando o modelo TCP/IP passou a ser adotado como padrão oficial da antiga ARPANET, a evolução da internet vem ocorrendo de forma explosiva e quase que independente. Devido a essa evolução, há mais de vinte anos foi observada a necessidade de implementar novos recursos, bem como proporcionar a melhoria de outros já existentes na estrutura de endereçamento utilizada, motivados principalmente pelo esgotamento eminente dos endereços IPv4 disponíveis para novas alocações. Segundo Santos (2010) atualmente existem somente sete blocos IPv4 /8 disponíveis na IANA (*Internet Assigned Numbers Authority*), bem como problemas causados pela complexidade das tabelas de roteamento na Internet. Destarte, em 1990 a IETF (*Internet Engineering Task Force*) iniciou um projeto intitulado IPng (*Internet Protocol New Generation*) com a finalidade de gerar uma nova estrutura de endereçamento, sendo assim, foi aberta a chamada para submissão de propostas através da RFC 1550 (BRADNER e MANKIN, 1993). Em 1992, após a classificação dos trabalhos, havia sete modelos em análise, contudo a versão inicial do IPv6 foi criada a partir da combinação da proposta de Deering e Francis de 1993 (TANENBAUM, 2003).

Atualmente existe uma grande quantidade de documentos focando pontos distintos do IPv6, mas devido principalmente à sua abrangência e complexidade, o estudo aprofundado sobre pontos específicos do seu funcionamento acaba sendo uma tarefa complexa e consumidora de muito tempo de pesquisa.

Este estudo descreve o *Unique Local* IPv6 *Unicast Address*, um tipo de endereço do IPv6 cuja finalidade é possibilitar a comunicação entre componentes de uma mesma sub-rede local e entre sub-redes locais distintas. Também é apresentada a aplicação prática dessa tecnologia, abordando pontos

como a estruturação dos endereços e as rotinas a serem executadas para a configuração dos componentes de uma rede.

Contudo, para o bom entendimento deste estudo, é necessário que o leitor tenha o conhecimento prévio do formato e notação utilizados no endereçamento IPv6, descrito por Hinden e Deering (2003).

2. Unique Local IPv6 Unicast Address

O *Unique Local* IPv6 *Unicast Address*, ou simplesmente ULA, é descrito por Hinden e Haberman (2005). Sua estrutura é composta pelos seguintes elementos: prefixo, valor de *L*, identificador global, identificador de sub-rede e identificador de interface, como ilustrado pela Figura 1.

O prefixo é formado por sete bits e tem, obrigatoriamente, o valor FC00::/7. O valor de *L* possui um único bit, portanto poderá assumir somente os valores zero e um. Este último será determinado a *L* quando houver uma atribuição local do endereço IP e o valor zero poderá ter sua regra de atribuição definida futuramente. O identificador global é composto por quarenta bits, valor este gerado pseudo-aletoriamente segundo alguns critérios descritos posteriormente e utilizado para criar um prefixo global único de 48 bits, ou seja, a junção do prefixo, do valor de *L* e do identificador global, respectivamente.

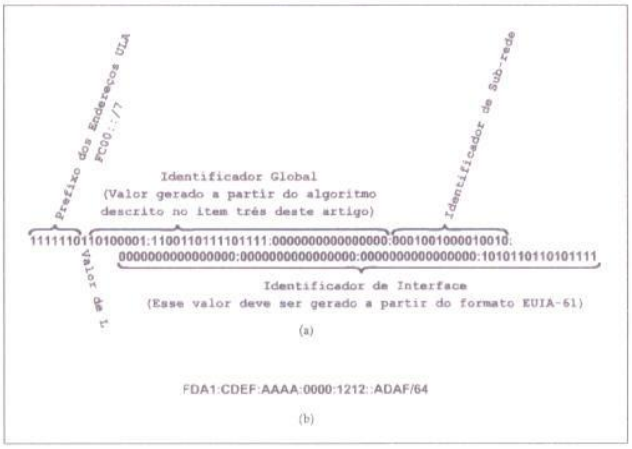

Figura 1 - *Unique Local* IPv6 *Unicast Address* descrito em formato binário (a) e descrito em formato hexadecimal (b).

O identificador de sub-rede é definido arbitrariamente, descrevendo a rede ao qual o endereço pertence dentro do sistema. Por fim, o identificador da interface é atribuído seguindo as especificações da RFC 3513 (HINDEN e DEERING, 2003), onde é afirmado que esse valor **não pode** ser repetido dentro de uma sub-rede e **deve** ter 64 bits de tamanho (com exceção dos endereços que se iniciam com o valor binário 000), construídos a partir do formato *Modified* EUIA-64, também descrito e exemplificado pelo apêndice A do mesmo documento.

Existem três maneiras distintas de se atribuir um ULA a uma interface: via DHCPv6, manualmente ou através do protocolo de descoberta de vizinhança, descrito por Narten, Nordmark e Simpson (1998). Portanto, o ULA não é gerado automaticamente como o *Link Local Address*[3]. Também é importante destacar que não é possível rotear o ULA na internet, sendo assim, ele pode ser utilizado somente para comunicação interna e entre empresas que possuam comunicação direta ou através de uma VPN configurada adequadamente.

3. Algoritmo de Geração Pseudo-aleatória do Identificador Global

O identificador global é gerado aleatoriamente obedecendo ao critério proposto por Hinden e Haberman (2005, p. 5), composto pelos seis passos a seguir: (a) obter a hora atual no formato NTP 64 bits, descrito pela RFC 1305 (MILLS, 1992); (b) obter o identificador EUIA-64 do sistema onde o algoritmo será executado - na falta desse, seu valor deve ser obtido a partir de um endereço MAC de 48 bits segunda a RFC 3513 (HINDEN e DEERING, 2003) e caso esse último também não exista deve ser utilizado o valor de um identificador equivalente, como por exemplo, o número serial do sistema; (c) concatenar os valores obtidos em (a) e (b), respectivamente; (d) calcular o valor *hash,* SHA-1 (EASTLAKE 3rd e JONES, 2001) de 160 bits, a partir dos valores concatenados no terceiro passo; (e) utilizar os últimos 40 bits significantes do valor *hash* como identificador global e; (f) concatenar o prefixo FC00::/7, o valor de *L*, definido como 1, e os 40 bits do identificador global, gerando assim o prefixo do endereço local do IPv6.

---

[3]Endereço gerado automaticamente e utilizado para a comunicação de componentes pertencentes somente ao mesmo enlace. Sua descrição completa é feita por Thomson, Narten e Junmei (2007).

4. Probabilidade de Colisão

Observando a propriedade pseudo-aleatória de geração do ULA, devemos considerar a possibilidade de haver colisão entre os endereços utilizados nas interfaces e, portanto, torna-se necessário conhecer quais são as reais chances da ocorrência desse fenômeno, informação que pode ser obtida através da fórmula descrita por Hiden e Haberman (2005):

$$P = 1 - \lim_{n \to \infty} \left(1 + \frac{\frac{-N^2}{2} / 2^{L+1}}{n}\right)^n \quad (1)$$

onde P é a probabilidade de colisão de endereços; N é o número de conexões; L é o número de bits do identificador global e e é a constante de Euler[4].

Observando os resultados na figura 2 é notável que a chance de colisão de endereços locais únicos é extremamente pequena, mesmo quando utilizado sobre um grande número de conexões.

Figura 2 - Distribuição da probabilidade de colisão em função do número de conexões para todo $L = 40$.

5. Aplicando o Unique Local Ipv6 Unicast Address

Esta seção descreve uma ferramenta em Java com a finalidade de gerar o prefixo global dos endereços ULA e posteriormente expõe estudos de caso que

[4]Valor aproximadamente igual a 2,71828183.

a apliquem. Deste modo, as demonstrações de sua utilização também têm a finalidade de expor os conceitos de atribuição de endereços ULA IPv6 às interfaces através de duas maneiras distintas: atribuição de endereços efetuada manualmente e através do protocolo *Neighbourhood Discovery* (NARTEN, NORDMARK e SIMPSON, 1998).

5.1 A Ferramenta UniLaGenJ

Foi desenvolvida, com base na seção 2 deste estudo, uma ferramenta em Java, nomeada UniLAGenJ[5], com a finalidade de gerar o identificador global do ULA, a qual foi utilizada para a aplicação de tal endereço em diferentes contextos. Sua estrutura estática é exposta na figura 3 através do diagrama de classe da linguagem de modelagem UML 2.0 (BOOCH, RUMBAUGH e JACOBSON, 2005).

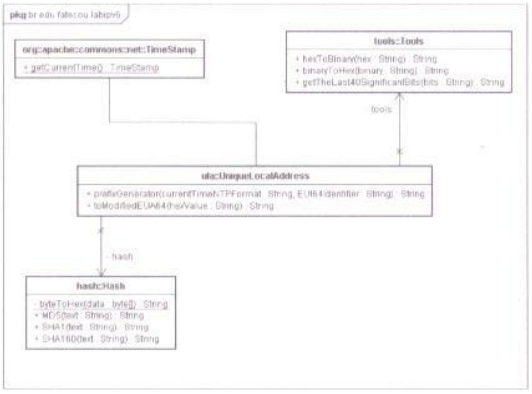

Figura 3 - Diagrama de classes da ferramenta UniLAGenJ.

Estruturamos essa ferramenta da seguinte maneira: a classe *Hash* contém métodos responsáveis por gerar o código MD5, SHA1 de 160 bits e converter bytes para hexadecimal, auxiliando no processo final de geração do prefixo global. A classe *Tools* possui métodos com a finalidade de converter bases numéricas, mais precisamente, converter hexadecimal em binário e

[5]Essa ferramenta está disponível no seguinte endereço: http://tiny.cc/xur8x.

vice-versa, além de conter o método responsável por extrair os últimos quarenta bits significantes de um conjunto de bits. A classe *TimeStamp*, do pacote org.apache.commons.net, contém o método responsável por fornecer o valor do tempo corrente no formato NTP 64 bits, descrito pela RFC 1305 (MILLS, 1992), valor este utilizado com parâmetro para geração do prefixo global. E por último, a classe *UniqueLocalAddress*, possui dois métodos: o primeiro, *toModifiedEUI64*, tem a finalidade de aceitar um MAC *address* como parâmetro de entrada e gerar um valor no formato *Modified* EUI-64 descrito pelo apêndice A da RFC 3513 (HINDEN e DEERING, 2003) e o método responsável pelo geração do prefixo global, *prefixGenerator*, que aceita o valor do tempo corrente no formato NTP 64 bits e o *Modified* EUI-64 como parâmetros de entrada e retorna o prefixo global de 40 bits.

Uma das principais vantagens existentes na utilização dessa ferramenta é a de não haver necessidade de uma interface gráfica sendo possível, portanto, usufruir de seus recursos através de um terminal de comandos, como o *Shell* de sistemas da família Unix ou o *Command* de sistemas família Windows, além de ser multiplataforma, havendo somente a necessidade de uma JVM instalada.

5.2 Estudos de Caso

Esta seção possui a finalidade de expor alguns estudos de caso com a utilização da ferramenta UniLAGenJ para a aplicação de endereços locais únicos do IPv6, bem como na explanação do protocolo *Neighbourhood Discovery* como estratégia de atribuição de endereço. Cada estudo contém um objetivo específico, onde as tarefas executadas podem conter procedimentos úteis aos estudos de caso posteriores. Os códigos expostos devem ter os termos até o símbolo '#' descartados, uma vez que possuem a simples finalidade de informar em qual elemento do estudo de caso foram executados.

Estudo de Caso 01: seu objetivo é gerar endereços locais únicos do IPv6 através da ferramenta UniLAGenJ e atribuí-los às interfaces de dois nós na mesma sub-rede, tornando possível sua comunicação.

Como proposto pela RFC 4193 (HINDEN e HABERMAN, 2005, p. 5), é necessário calcular o prefixo global do ULA, sendo que tal procedimento é executado pela ferramenta UniLAGenJ. Depois de gerado o prefixo global, é necessário definir os valores do identificador de sub-rede e dos identificadores das interfaces. A concatenação desses valores forma o ULA, finalizando assim a sua composição.

Para este estudo de caso, foi gerado o prefixo global FD9C:692:4226::/48 com a ferramenta UniLAGenJ com a execução do

seguinte comando: *java -jar unilagenj.jar 001FE2A551AE*, sendo o parâmetro utilizado para a criação do identificador global o endereço MAC de 48bits de uma determinada interface do estudo de caso. O valor :0: foi definido como identificador de sub-rede e o fragmento 1::1 foi atribuído ao identificador da interface do *Host-01* e o valor 1::2 à interface do *Host-02*. Deste modo, o código abaixo descreve o procedimento necessário para atribuir os endereços gerados às interfaces dos componentes da rede:

```
host-01~# ip -6 addr add FD9C:692:4226:0:1::1/64 dev eth0
host-02~# ip -6 addr add FD9C:692:4226:0:1::2/64 dev eth0
```

Estudo de Caso 02: seu objetivo é atribuir manualmente endereços locais únicos IPv6 às interfaces de duas máquinas em sub-rede distintas, conectadas entre si através de um roteador Linux, como ilustra a figura abaixo:

Figura 4 - Topologia do *Estudo de Caso 02*. Composição: Venezian e Barea (2011).

O procedimento para composição dos ULAs utilizados nas interfaces do *Host-01* e *Host-02* deve ser exatamente o mesmo descrito no *Estudo de Caso 01*, observando apenas a necessidade de se modificar o prefixo de sub-rede devido ao fato das máquinas estarem em sub-redes distintas. Também é possível calcular um novo prefixo global para atribuição do ULA, porém, esse processo não é obrigatório. As interfaces do roteador devem ter um ULA válido para cada sub-rede, de forma a prover um caminho válido de comunicação entre as máquinas. O roteador também deve ter a capacidade de repassar os dados transmitidos através de uma interface para outra. O código abaixo apresenta o procedimento de configuração do roteador Linux, atribuindo os endereços locais únicos e habilitando o roteamento entre as interfaces:

```
roteador:~# ip -6 addr add FD9C:0692:4226:0:1::2/64 dev eth0
roteador:~# ip -6 addr add FD9C:0692:4226:1:1::2/64 dev eth0
roteador:~# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

O próximo passo é informar aos outros elementos das sub-redes que

toda comunicação entre as sub-redes deve ocorrer através do roteador. Essa configuração é feita com a inclusão de rotas nas máquinas clientes da topologia, como apresentado no código abaixo:

```
host-01:~# ip -6 route add default via FD9C:692:4226:0:1::2
host-02:~# ip -6 route add default via FD9C:692:4226:1:1::2
```

Com isso, as máquinas clientes podem se comunicar internamente com outras máquinas da sua própria sub-rede, bem como com máquinas de sub-redes distintas através do roteador.

Estudo de Caso 03: seu objetivo é atribuir endereços locais únicos IPv6 às interfaces de duas máquinas em sub-redes distintas, conectadas entre si através de um roteador Linux utilizando o protocolo *Neighbourhood Discovery* (NARTEN, NORDMARK e SIMPSON, 1998).
A topologia utilizada neste estudo de caso é a mesma utilizada pelo estudo de caso 2, havendo distinção em relação à estratégia de atribuição de endereços locais únicos e nos sistemas utilizados no roteador. Desse modo, foi empregado o pacote de softwares de roteamento *Quagga*[6], serviço responsável por prover suporte ao software GNU Zebra[7], que possui a finalidade de fornecer gerenciamento a tecnologia TCP/IP baseado em protocolos de roteamento e a implementação dos protocolos OSPFv2, OSPFv3, RIP v1 e v2, RIPng e BGP-4.
Sendo assim, antes de explanarmos os códigos referentes às configurações do *Neighbourhood Discovery*, descreveremos sua configuração mínima para o objetivo deste estudo de caso. Portanto, para instalar o *Quagga* no sistema operacional utilizado em nossos estudos, basta executar o seguinte comando: apt-get install quagga. Após sua instalação, deve-se alterar o conteúdo do arquivo */etc/quagga/zebra.conf* com os seguintes valores:

```
! nome do roteador Quagga
hostname Router
! senha para acesso ao roteador
password zebra
! senha para acesso privilegiado ao roteador
enable password zebra
! direciona os relatórios do roteador para a tela do usuário
log stdout
```

[6]Software que une vários protocolos de roteamento. Mais informações sobre o Quagga podem ser encontradas em http://www.quagga.net/.

[7]Software de roteamento distribuído sob a *General Public License*. Mais informações sobre o GNU Zebra podem ser encontradas em http://www.zebra.org/.

Como o *Quagga* tem a função de tonar a máquina hospedeira de seu serviço em um roteador, é importante que toda interface instalada no *host* seja definida nele, para que assim possa utilizá-las em suas tarefas. Dessa maneira, a última configuração para que o *Quagga* funcione corretamente é feita no arquivo */etc/quagga/daemons*, tendo a finalidade de habilitar o serviço zebra do suíte de aplicativos de roteamento, com os seguintes valores: zebra=yes. Depois de finalizadas as configurações, é necessário reiniciar o serviço executando o comando: /etc/init.d/quagga restart. Uma mensagem equivalente a abaixo deverá ser exibida:

```
Stopping Quagga daemons (prio:0): zebra (bgpd) (ripd) (ripngd)
(ospfd) (ospf6d) (isisd).
Removing all routes made by zebra.
Nothing to flush.
Loading capability module if not yet done.
Starting Quagga daemons (prio:10): zebra.
```

Para acessar o *Quagga* é necessário ter o aplicativo cliente *telnet* devidamente instalado e, caso não esteja, basta executar o seguinte código: apt-get install telnet. Depois de obter o servidor *Quagga* e o aplicativo cliente *telnet* devidamente instalados, configurados e com seus serviços inicializados, devemos configurar o roteador, para isso devemos primeiramente acessá-lo através do código:

```
roteador:~# telnet localhost zebra;
```

Nele, *localhost* é o endereço do servidor contendo o serviço *Quagga* e *zebra* é o nome do roteador, definido no arquivo de configuração */etc/quagga/zebra.conf*. Após a execução do comando acima digite sua senha definida nesse último arquivo, que neste estudo de caso foi definido *zebra*. Portanto, para iniciar sua configuração basta executar os seguintes comandos:

```
router> enable
Passwword: zebra
router# configure terminal
```

A primeira linha habilita o modo de edição do roteador, sendo necessário informar a senha de acesso e o comando *configure terminal* é utilizado para acessar o modo de edição. Note que os elementos *router>, Password:* e *router#* são informativos do aplicativo e não códigos a serem executados.

Para a configuração do protocolo *Neightbourhood Discovery* são necessárias três definições em cada interface que se deseja utilizar esse protocolo: (a) habilitar o protocolo *Router Advertisement*; (b) atribuir uma

prefixo de, **obrigatoriamente**, sessenta e quatro bits e; (c) atribuir um endereço ULA a interface, também com sessenta e quatro bits, obrigatoriamente. Sendo assim, para tal tarefa é necessário executar os comandos abaixo:

```
router(config)# interface eth0
router(config-if)# no ipv6 nd suppress-ra
router(config-if)# ipv6 nd prefix FD9C:692:4226:0::/64
router(config-if)# ipv6 address FD9C:692:4226:0:1::1/64
```

A primeira linha é utilizada para acessar o modo de edição da interface *eth0* do roteador; o segundo comando habilita o protocolo *Router Advertisement*; a próxima linha define o prefixo que será utilizado nos *hosts* conectados à interface *eth0*, para compor o seu respectivo ULA; e o último comando define o endereço ULA de sessenta e quatro bits da interface *eth0* do roteador. Para compor os sessenta e quatro bits restantes do endereço dos *hosts* é utilizada a estratégia do *Modified* EUIA-64, definido no apêndice A de Hinden e Deering (2003), por isso existe a obrigatoriedade do prefixo definido no roteador ter sessenta e quatro bits. O resultado final dessa configuração pode ser exibido executando o seguinte código: router# show running-config, o que deverá trazer um resultado equivalente ao descrito a seguir:

```
Current configuration:
!
hostname zebra
enable password zebra
log stdout
!
interface eth0
  ipv6 address fd9c:692:4226:0:1::1/64
  no ipv6 nd suppress-ra
  ipv6 nd prefix fd9c:692:4226:1::/64
!
interface lo
!
line vty
!
end
```

Para salvar as configurações do roteador é necessário executar o comando *write*. Deste modo, para a auto-configuração dos equipamentos é necessário somente conectá-los ao roteador através da interface definida e assim um ULA será atribuído a eles. Lembrando que para cada interface do roteador onde se deseja utilizar o protocolo *Neighbourhood Discovery* os procedimentos descritos neste estudo de caso devem ser repetidos em cada

uma, havendo modificações somente nos nomes das interfaces, prefixos e endereços definidos nos códigos descritos neste estudo de caso.

6. Considerações Finais

Este estudo teve por finalidade expor, de forma clara e objetiva, os conceitos e aplicações do *Unique Local* IPv6 *Unicast Address*, um recurso importante para situações que envolvam a necessidade de confidencialidade de informações disponíveis em uma rede, ou até mesmo para conexões mais seguras entre sítios distintos através de conexões diretas ou de uma VPN (*Virtual Private Networt*). Também foram expostos alguns recursos relativos ao IPv6, como por exemplo, o protocolo *Neighbourhood Discovery*, que possui a finalidade de atribuir endereços automaticamente aos componentes conectados à rede. Além disso, foi apresentada a ferramenta UniLaGenJ, desenvolvida com a finalidade de gerar o prefixo global dos endereços ULA do IPv6, tornando-se um recurso importante para sua aplicação. Deste modo é possível notar a importância do entendimento e conhecimento dos recursos presentes nesse novo protocolo, o qual resolverá inúmeros problemas recorrentes de sua versão anterior, o IPv4.

7. Referências

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML**: guia do usuário. 2. ed. Rio de Janeiro. Elsevier, 2005.

BRADNER, S.; MANKIN, A. **IP: Next Generation (IPng) White Paper Solicitation**. **Internet Engineering Task Force**. 1993. Disponivel em: <http://www.ietf.org/rfc/rfc1550>. Acesso em: 28 novembro 2010.

EASTLAKE 3RD, D.; JONES, P. **US Secure Hash Algorithm 1 (SHA1)**. 2001. Disponivel em: <http://www.ietf.org/rfc/rfc3174>. Acesso em: 7 outubro 2010.

HINDEN, R. M.; HABERMAN, B. **Unique Local IPv6 Unicast Addresses**. 2005. Disponivel em: <http://www.ietf.org/rfc/rfc4193>. Acesso em: 22 setembro 2010.

HINDEN, R.; DEERING, S. **Internet Protocol Version 6 (IPv6) Addressing Architecture**. 2003. Disponivel em: <http://www.ietf.org/rfc/rfc3513>. Acesso em: 22 setembro 2010.

MILLS, D. **Network Time Protocol (Version 3) Specification, Implementation and Analysis**. 1992. Disponivel em: <http://www.ietf.org/rfc/rfc1305>. Acesso em: 22 setembro 2010.

NARTEN, T.; NORDMARK, E.; SIMPSON, W. **Neighbor Discovery for IP Version 6 (IPv6)**. 1998. Disponivel em: <http://www.ietf.org/rfc/rfc2461>. Acesso em: 1 outubro 2010.

SANTOS, R. R. D. O fim está próximo. **IPv6.br**, 2010. Disponivel em: <http://www.ipv6.br/IPV6/ArtigoNoticiasOFimEstaProximo>. Acesso em: 12 dezembro 2010.

TANENBAUM, A. S. **Redes de Computadores**. Tradução de Vandenberg D. de Souza. Rio de Janeiro: Elsevier, 2003.