

# O Linux está pronto para o IPv6?

Eduardo Habkost

Mandriva Conectiva

LinuxWorld Conference & Expo, 2006

# Conteúdo

- 1 **Introdução ao IPv6**
  - Histórico
  - Características do IPv6
  - IPv6 Hoje
- 2 **Status do IPv6 no Linux**
  - Histórico
  - O Logo IPv6-ready
  - Resultados dos testes IPv6-ready
- 3 **Como eu posso usar IPv6 hoje?**
  - O IPv6 está pronto para nós?
  - Tecnologias de transição

# Conteúdo

- 1 **Introdução ao IPv6**
  - **Histórico**
  - Características do IPv6
  - IPv6 Hoje
- 2 Status do IPv6 no Linux
  - Histórico
  - O Logo IPv6-ready
  - Resultados dos testes IPv6-ready
- 3 Como eu posso usar IPv6 hoje?
  - O IPv6 está pronto para nós?
  - Tecnologias de transição

# O Início

## Limitações do IPv4:

- Espaço de endereços limitado (32 bits)
- Escalabilidade do sistema de roteamento

## História (1/2)

- ?-1990 Limitações do IPv4 são notadas
- 1992 IPng
- 01/1995 RFC 1752: *The Recommendation for the IP Next Generation Protocol*
- 12/1995 RFC 1883: *Proposed Standard*
- 1998 RFC 2460: *Draft Standard*

## História (2/2)

- 03/1996 6bone – rede global de testes
- 1997 Primeiros prefixos 6bone alocados
- 1999 Primeiros prefixos de produção alocados
- 2005 Agências Federais dos EUA deverão implementar IPv6 até 2008
- 05/2006 Mais de 1200 prefixos / 32 de produção alocados
- 06/06/2006 Fim 6bone

# Conteúdo

- 1 **Introdução ao IPv6**
  - Histórico
  - **Características do IPv6**
  - IPv6 Hoje
- 2 Status do IPv6 no Linux
  - Histórico
  - O Logo IPv6-ready
  - Resultados dos testes IPv6-ready
- 3 Como eu posso usar IPv6 hoje?
  - O IPv6 está pronto para nós?
  - Tecnologias de transição

## Características do IPv6

- Espaço de endereçamento bem maior: 128 bits
- A vida fica mais simples para os roteadores
  - Arquitetura de endereçamento
  - Endereços "agregáveis"
  - *Extension Headers*
  - Fragmentação só nas pontas
- Autoconfiguração é parte do protocolo
- Autoconfiguração *stateless*
- Segurança (IPsec)
- Mobilidade (Mobile IPv6)

## Notação dos endereços IPv6

- XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX
- Hexadecimal
- Abreviando:
  - 1234:0567:0008:0000:0000:0000:0009:0123 (completo)
  - 1234:567:8:0:0:0:9:123 (zeros à direita podem ser omitidos)
  - 1234:567:8::9:123 (bloco de zeros pode ser escrito como ::)
- Endereço de rede: prefixo/bits. Exemplo:  
2001:4815::/32

# Conteúdo

- 1 **Introdução ao IPv6**
  - Histórico
  - Características do IPv6
  - **IPv6 Hoje**
- 2 Status do IPv6 no Linux
  - Histórico
  - O Logo IPv6-ready
  - Resultados dos testes IPv6-ready
- 3 Como eu posso usar IPv6 hoje?
  - O IPv6 está pronto para nós?
  - Tecnologias de transição

# Implementações IPv6

- Solaris
- Windows (2000, XP)
- KAME - Família BSD
- Linux
- Roteadores (Cisco, Hitachi, NEC, e outros)
- ... e outros
- 233 produtos certificados em <http://www.ipv6ready.org>

# Implementações IPv6

- Solaris
- Windows (2000, XP)
- KAME - Família BSD
- Linux
- Roteadores (Cisco, Hitachi, NEC, e outros)
- ... e outros
- 233 produtos certificados em <http://www.ipv6ready.org>

## Infra-estrutura disponível

- Várias redes pelo mundo: mais de 1200 prefixos /32 de produção alocados
- Japão: muitos ISPs já fornecem IPv6
- Tecnologias de transição (Túneis, NAT)

## Por que não estão todos usando?

- Uso de NAT IPv4 em larga escala, para contornar o espaço de endereços pequeno
- Funcionalidades IPv6 "portadas" para IPv4
  - IPsec
  - Autoconfiguração
  - Mobile IPv4
- Espaço de endereçamento
- NAT não resolve totalmente o problema:
  - NAT não é *peer-to-peer*
  - NAT tem *single point of failure*
  - NAT "quebra" alguns protocolos

# Conteúdo

- 1 Introdução ao IPv6
  - Histórico
  - Características do IPv6
  - IPv6 Hoje
- 2 Status do IPv6 no Linux
  - **Histórico**
  - O Logo IPv6-ready
  - Resultados dos testes IPv6-ready
- 3 Como eu posso usar IPv6 hoje?
  - O IPv6 está pronto para nós?
  - Tecnologias de transição

# Histórico do IPv6 no Linux

- Desde a versão 2.1.x (1996)
  - Qualidade deixava a desejar
- 2000: Lançado Projeto USAGI
  - Objetivo: Melhorar código IPv6 no Linux
- Logo IPv6-ready
- 02/2004: USAGI: *Phase-1* no kernel 2.6
- 09/2004: USAGI: *Phase-1* no kernel 2.4
- 05/2005: Kernel 2.6.11-rc2 recebe o selo *IPv6-ready Phase-1*
- 06/2005: Kernel 2.6.12: passa no *self-test Phase-2* (porém ainda não submetido para a certificação)

# Histórico do IPv6 no Linux

- Desde a versão 2.1.x (1996)
  - Qualidade deixava a desejar
- 2000: Lançado Projeto USAGI
  - Objetivo: Melhorar código IPv6 no Linux
- Logo IPv6-ready
- 02/2004: USAGI: *Phase-1* no kernel 2.6
- 09/2004: USAGI: *Phase-1* no kernel 2.4
- 05/2005: Kernel 2.6.11-rc2 recebe o selo *IPv6-ready Phase-1*
- 06/2005: Kernel 2.6.12: passa no *self-test Phase-2* (porém ainda não submetido para a certificação)

## Histórico do IPv6 no Linux

- Desde a versão 2.1.x (1996)
  - Qualidade deixava a desejar
- 2000: Lançado Projeto USAGI
  - Objetivo: Melhorar código IPv6 no Linux
- Logo IPv6-ready
- 02/2004: USAGI: *Phase-1* no kernel 2.6
- 09/2004: USAGI: *Phase-1* no kernel 2.4
- 05/2005: Kernel 2.6.11-rc2 recebe o selo *IPv6-ready Phase-1*
- 06/2005: Kernel 2.6.12: passa no *self-test Phase-2* (porém ainda não submetido para a certificação)

# Conteúdo

- 1 Introdução ao IPv6
  - Histórico
  - Características do IPv6
  - IPv6 Hoje
- 2 **Status do IPv6 no Linux**
  - Histórico
  - **O Logo IPv6-ready**
  - Resultados dos testes IPv6-ready
- 3 Como eu posso usar IPv6 hoje?
  - O IPv6 está pronto para nós?
  - Tecnologias de transição

## O Logo *IPv6-ready*

Programa de certificação para implementações IPv6.

- Funções: *host e router*
- *Self-test* e interoperabilidade
- Fase 1
  - Desde 09/2003
  - Funções básicas (*core protocols*)
  - itens *MUST*
- Fase 2
  - Desde 02/2005
  - Itens *MUST* e *SHOULD*
  - IPsec, Mobile IPv6, outros
- Testes automatizados nos laboratórios USAGI



## O Logo *IPv6-ready*

Programa de certificação para implementações IPv6.

- Funções: *host e router*
- *Self-test* e interoperabilidade
- Fase 1
  - Desde 09/2003
  - Funções básicas (*core protocols*)
  - itens *MUST*
- Fase 2
  - Desde 02/2005
  - Itens *MUST* e *SHOULD*
  - IPsec, Mobile IPv6, outros
- Testes automatizados nos laboratórios USAGI



## O Logo *IPv6-ready*

Programa de certificação para implementações IPv6.

- Funções: *host e router*
- *Self-test* e interoperabilidade
- Fase 1
  - Desde 09/2003
  - Funções básicas (*core protocols*)
  - itens *MUST*
- Fase 2
  - Desde 02/2005
  - Itens *MUST* e *SHOULD*
  - IPsec, Mobile IPv6, outros
- Testes automatizados nos laboratórios USAGI



## O Logo *IPv6-ready*

Programa de certificação para implementações IPv6.

- Funções: *host e router*
- *Self-test* e interoperabilidade
- Fase 1
  - Desde 09/2003
  - Funções básicas (*core protocols*)
  - itens *MUST*
- Fase 2
  - Desde 02/2005
  - Itens *MUST* e *SHOULD*
  - IPsec, Mobile IPv6, outros
- Testes automatizados nos laboratórios USAGI



## O Logo *IPv6-ready*

Programa de certificação para implementações IPv6.

- Funções: *host e router*
- *Self-test* e interoperabilidade
- Fase 1
  - Desde 09/2003
  - Funções básicas (*core protocols*)
  - itens *MUST*
- Fase 2
  - Desde 02/2005
  - Itens *MUST* e *SHOULD*
  - IPsec, Mobile IPv6, outros
- Testes automatizados nos laboratórios USAGI



## O Logo *IPv6-ready*

Programa de certificação para implementações IPv6.

- Funções: *host e router*
- *Self-test* e interoperabilidade
- Fase 1
  - Desde 09/2003
  - Funções básicas (*core protocols*)
  - itens *MUST*
- Fase 2
  - Desde 02/2005
  - Itens *MUST* e *SHOULD*
  - IPsec, Mobile IPv6, outros
- Testes automatizados nos laboratórios USAGI



# Conteúdo

- 1 Introdução ao IPv6
  - Histórico
  - Características do IPv6
  - IPv6 Hoje
- 2 Status do IPv6 no Linux
  - Histórico
  - O Logo IPv6-ready
  - Resultados dos testes IPv6-ready
- 3 Como eu posso usar IPv6 hoje?
  - O IPv6 está pronto para nós?
  - Tecnologias de transição

## Histórico do IPv6 no Linux

- Desde a versão 2.1.x (1996)
  - Qualidade deixava a desejar
- 2000: Lançado Projeto USAGI
  - Objetivo: Melhorar código IPv6 no Linux
- Logo IPv6-ready
- 02/2004: USAGI: *Phase-1* no kernel 2.6
- 09/2004: USAGI: *Phase-1* no kernel 2.4
- 05/2005: Kernel 2.6.11-rc2 recebe o selo *IPv6-ready Phase-1*
- 06/2005: Kernel 2.6.12: passa no *self-test Phase-2* (porém ainda não submetido para a certificação)

## Histórico do IPv6 no Linux

- Desde a versão 2.1.x (1996)
  - Qualidade deixava a desejar
- 2000: Lançado Projeto USAGI
  - Objetivo: Melhorar código IPv6 no Linux
- Logo IPv6-ready
- 02/2004: USAGI: *Phase-1* no kernel 2.6
- 09/2004: USAGI: *Phase-1* no kernel 2.4
- 05/2005: Kernel 2.6.11-rc2 recebe o selo *IPv6-ready Phase-1*
- 06/2005: Kernel 2.6.12: passa no *self-test Phase-2* (porém ainda não submetido para a certificação)

## Histórico do IPv6 no Linux

- Desde a versão 2.1.x (1996)
  - Qualidade deixava a desejar
- 2000: Lançado Projeto USAGI
  - Objetivo: Melhorar código IPv6 no Linux
- Logo IPv6-ready
- 02/2004: USAGI: *Phase-1* no kernel 2.6
- 09/2004: USAGI: *Phase-1* no kernel 2.4
- 05/2005: Kernel 2.6.11-rc2 recebe o selo *IPv6-ready Phase-1*
- 06/2005: Kernel 2.6.12: passa no *self-test Phase-2* (porém ainda não submetido para a certificação)

## Histórico do IPv6 no Linux

- Desde a versão 2.1.x (1996)
  - Qualidade deixava a desejar
- 2000: Lançado Projeto USAGI
  - Objetivo: Melhorar código IPv6 no Linux
- Logo IPv6-ready
- 02/2004: USAGI: *Phase-1* no kernel 2.6
- 09/2004: USAGI: *Phase-1* no kernel 2.4
- 05/2005: Kernel 2.6.11-rc2 recebe o selo *IPv6-ready Phase-1*
- 06/2005: Kernel 2.6.12: passa no *self-test Phase-2* (porém ainda não submetido para a certificação)

## Histórico do IPv6 no Linux

- Desde a versão 2.1.x (1996)
  - Qualidade deixava a desejar
- 2000: Lançado Projeto USAGI
  - Objetivo: Melhorar código IPv6 no Linux
- Logo IPv6-ready
- 02/2004: USAGI: *Phase-1* no kernel 2.6
- 09/2004: USAGI: *Phase-1* no kernel 2.4
- 05/2005: Kernel 2.6.11-rc2 recebe o selo *IPv6-ready Phase-1*
- 06/2005: Kernel 2.6.12: passa no *self-test Phase-2* (porém ainda não submetido para a certificação)

# Conteúdo

- 1 Introdução ao IPv6
  - Histórico
  - Características do IPv6
  - IPv6 Hoje
- 2 Status do IPv6 no Linux
  - Histórico
  - O Logo IPv6-ready
  - Resultados dos testes IPv6-ready
- 3 Como eu posso usar IPv6 hoje?
  - O IPv6 está pronto para nós?
  - Tecnologias de transição

# O IPv6 está pronto para nós?

Como usar IPv6 com a tecnologia de hoje?

- A transição é um processo lento
- Aplicações: a maioria já suporta IPv6, portar é simples
- Infra-estrutura IPv4 existente muito grande
- Mas tecnologias de transição facilitam nossa vida

# Conteúdo

- 1 Introdução ao IPv6
  - Histórico
  - Características do IPv6
  - IPv6 Hoje
- 2 Status do IPv6 no Linux
  - Histórico
  - O Logo IPv6-ready
  - Resultados dos testes IPv6-ready
- 3 Como eu posso usar IPv6 hoje?
  - O IPv6 está pronto para nós?
  - **Tecnologias de transição**

# Usando IPv6 aproveitando a infra-estrutura IPv4

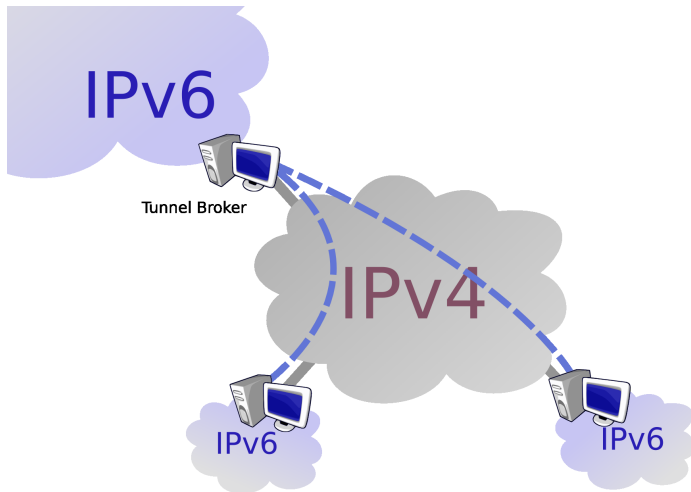
Para ligar-se à *Internet IPv6*:

- Túneis ligando redes IPv6 através de IPv4
  - Túneis manuais
  - Túneis automáticos
    - 6to4 – Túneis "para o mundo"
    - ISATAP – Túneis automáticos dentro da rede interna

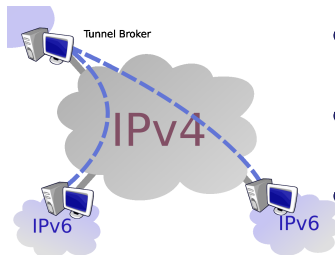
Para continuar utilizando a *Internet IPv4*:

- *Dual-stack*: suportar ambos os protocolos
- *NAT*: hosts *ipv6-only* podem enxergar a rede IPv4 como se fosse IPv6 (endereço especial: `::ffff:1.2.3.4`)
- *Dual-stack* é a solução mais comum

# Túneis

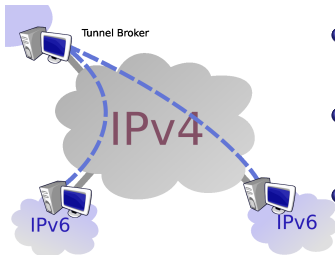


## Túneis (2/2)



- Infra-estrutura de roteamento IPv4 é utilizada para comunicação IPv6
- Pacotes IPv6 são encapsulados em pacotes IPv4
- Os pacotes IPv4 são enviados para um *Tunnel Broker*
- *Tunnel Broker* desencapsula o pacote IPv6 e encaminha

## Túneis manuais



- Um túnel a um *Tunnel Broker* específico é configurado
- Uma das primeiras soluções disponíveis. Já foi bastante comum
- Desvantagem: todos os pacotes IPv6 passam pelo *Tunnel Broker* – *Triangular Routing*

## Túneis automáticos

- O endereço IPv4 do *Tunnel Broker* é determinado automaticamente:
  - a) Pela infra-estrutura de roteamento IPv4; ou
  - b) Pelo endereço de destino IPv6, que geralmente contém um endereço IPv4 *embutido*
- **ISATAP** utiliza (b)
- **6to4** utiliza (a) e (b)

## Túneis automáticos

- O endereço IPv4 do *Tunnel Broker* é determinado automaticamente:
  - a) Pela infra-estrutura de roteamento IPv4; ou
  - b) Pelo endereço de destino IPv6, que geralmente contém um endereço IPv4 *embutido*
- **ISATAP** utiliza (b)
- **6to4** utiliza (a) e (b)

## Túneis automáticos

- O endereço IPv4 do *Tunnel Broker* é determinado automaticamente:
  - a) Pela infra-estrutura de roteamento IPv4; ou
  - b) Pelo endereço de destino IPv6, que geralmente contém um endereço IPv4 *embutido*
- ISATAP utiliza (b)
- 6to4 utiliza (a) e (b)

## Túneis automáticos

- O endereço IPv4 do *Tunnel Broker* é determinado automaticamente:
  - a) Pela infra-estrutura de roteamento IPv4; ou
  - b) Pelo endereço de destino IPv6, que geralmente contém um endereço IPv4 *embutido*
- **ISATAP** utiliza (b)
- **6to4** utiliza (a) e (b)

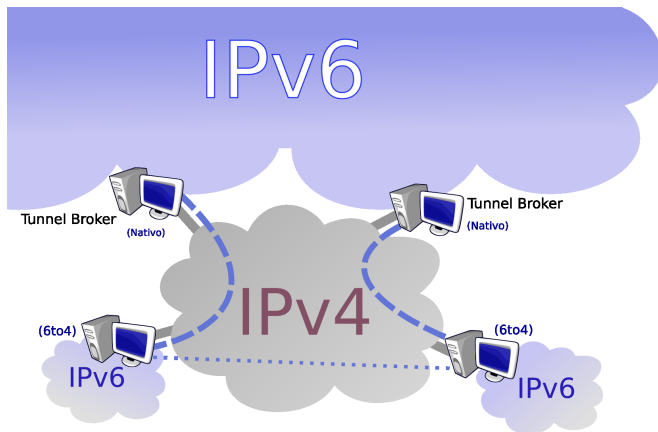
## Túneis automáticos

- O endereço IPv4 do *Tunnel Broker* é determinado automaticamente:
  - a) Pela infra-estrutura de roteamento IPv4; ou
  - b) Pelo endereço de destino IPv6, que geralmente contém um endereço IPv4 *embutido*
- **ISATAP** utiliza (b)
- **6to4** utiliza (a) e (b)

## 6to4 (1/3)

- RFC 3056
- O túnel é feito para um endereço *anycast* IPv4 (192.88.99.1)
- A infra-estrutura de roteamento IPv4 irá entregar o pacote IPv4 do túnel para o *Tunnel Broker* mais próximo
- O prefixo 2002::/16 é reservado para 6to4

## 6to4 (2/3)



## 6to4 (3/3)

- Cada endereço IPv4 ganha um prefixo IPv6 de 48 bits
- 65.536 sub-redes
- O prefixo é derivado do endereço IPv4: 1 . 2 . 3 . 4 vira 2002:0102:0304::
- Comunicação de hosts 6to4 com não-6to4: através do *Tunnel Broker*
- Comunicação de hosts 6to4 com 6to4: diretamente para o host IPv4 “dono” do endereço

## 6to4: Vantagens

- *Triangular Routing*:
  - Evitado em comunicação 6to4 com 6to4
  - Diminuído em comunicação 6to4 com não-6to4 (*Tunnel Broker* mais próximo é utilizado)
- Permite transição gradual. Mesmo que seu ISP não forneça IPv6, você já pode utilizar
- Como o *Tunnel Broker* é um endereço *anycast*, os ISPs podem gradualmente fornecer *Tunnel Brokers* em suas redes

## Configuração de 6to4

- Várias distribuições têm suporte para 6to4 nos seus scripts de inicialização de rede
  - Conectiva 10, Mandriva 2006.0
  - SuSE
  - Debian
  - Gentoo
  - outros
- Caso a distribuição não tenha suporte, um script simples pode fazer o trabalho
- <http://www.wlug.org.nz/6to4>

## Configuração de 6to4: Mandriva

```
/etc/sysconfig/network
```

```
NETWORKING_IPV6=yes  
IPV6_DEFAULTDEV=tun6to4
```

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

```
IPV6INIT=yes  
IPV6TO4INIT=yes
```

# Resumo

- O Linux *está* pronto para o IPv6
- Estamos em transição, mas é possível utilizar a infra-estrutura IPv4 existente para se comunicar com a *Internet IPv6*

## Referências

- **USAGI:** <http://www.linux-ipv6.org>
- **USAGI Test Lab:** <http://testlab.linux-ipv6.org>  
(acessível apenas por IPv6)
- **Kame:** <http://www.kame.net>
- **IPV6-Ready Logo Program:**  
<http://www.ipv6ready.org>
- **6to4 setup:** <http://www.wlug.org.nz/6to4>
- **Contato:** [ehabkost@mandriva.com](mailto:ehabkost@mandriva.com)