

CGI.br

Comitê Gestor da Internet no Brasil

O CGI.br - Comitê Gestor da Internet no Brasil - é a principal entidade relacionada à Governança da Internet no país e tem como objetivos:

fomentar o desenvolvimento da Internet no Brasil;

recomendar padrões e procedimentos relacionados à Internet;

coletar, organizar e disseminar informações relacionadas a Internet como, por exemplo, indicadores e estatísticas;

gerenciar os domínios .br a atribuição de números IP no país.

Idealizado pela comunidade acadêmica, por alguns setores da sociedade civil e pelo governo, o CGI.br foi criado por iniciativa deste último, em 1995.

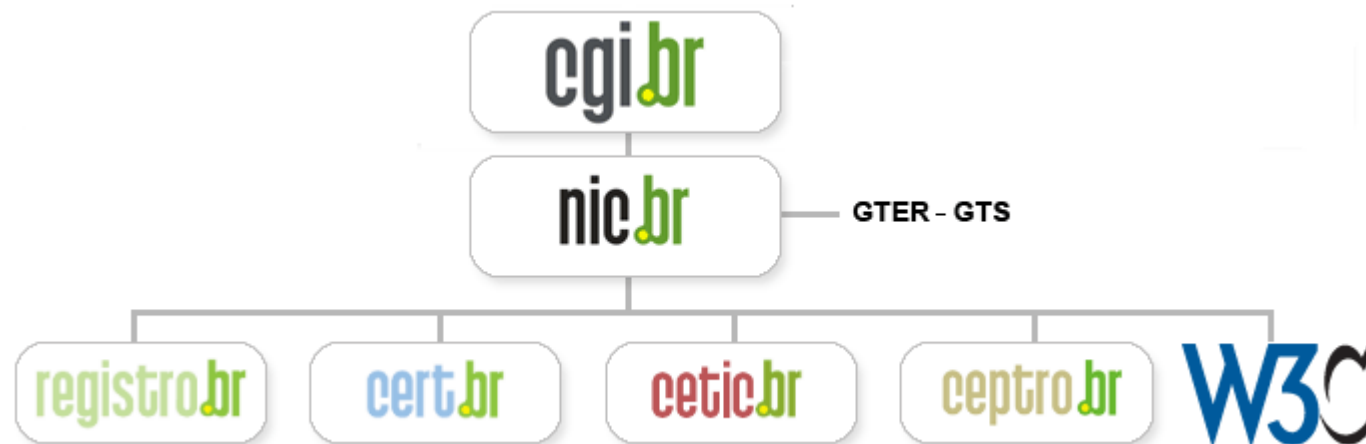
É formado por gente do próprio governo e por voluntários de outros setores, com visões diferentes e complementares sobre a Internet: representantes de usuários, provedores, indústria de software, academia e terceiro setor compõem o grupo.

O CGI.br configura-se como um fórum amplo e representativo para discutir as questões relativas à Internet no Brasil e influenciar positivamente seu desenvolvimento.

CGI.br

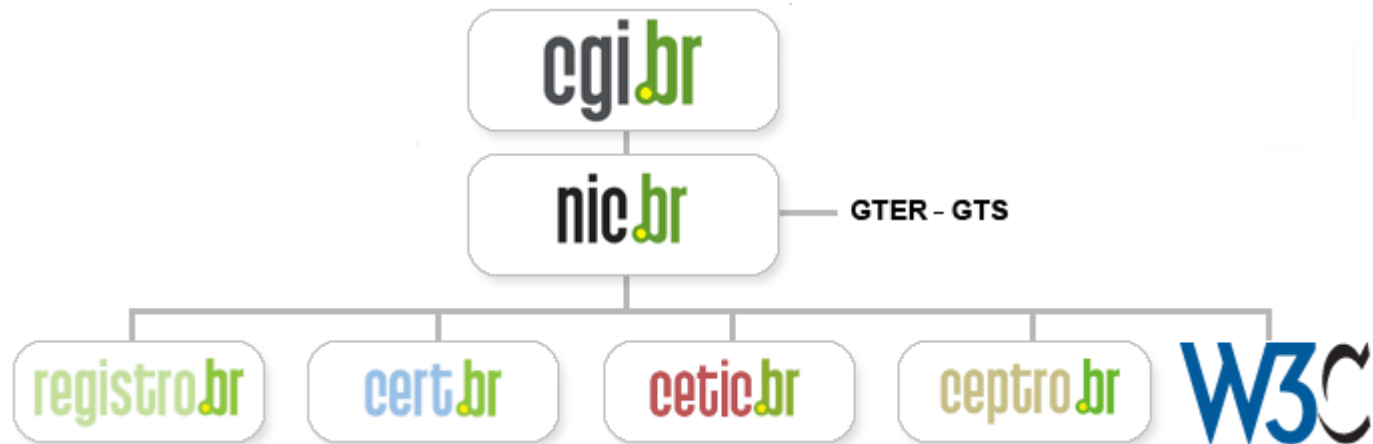
Comitê Gestor da Internet no Brasil

O Comitê Gestor da Internet criou o NIC.br – uma organização sem fins lucrativos – para ajudá-lo a cumprir com suas funções. O Núcleo de Informação e Coordenação do ponto br é o braço executivo do CGI.br.



CGI.br

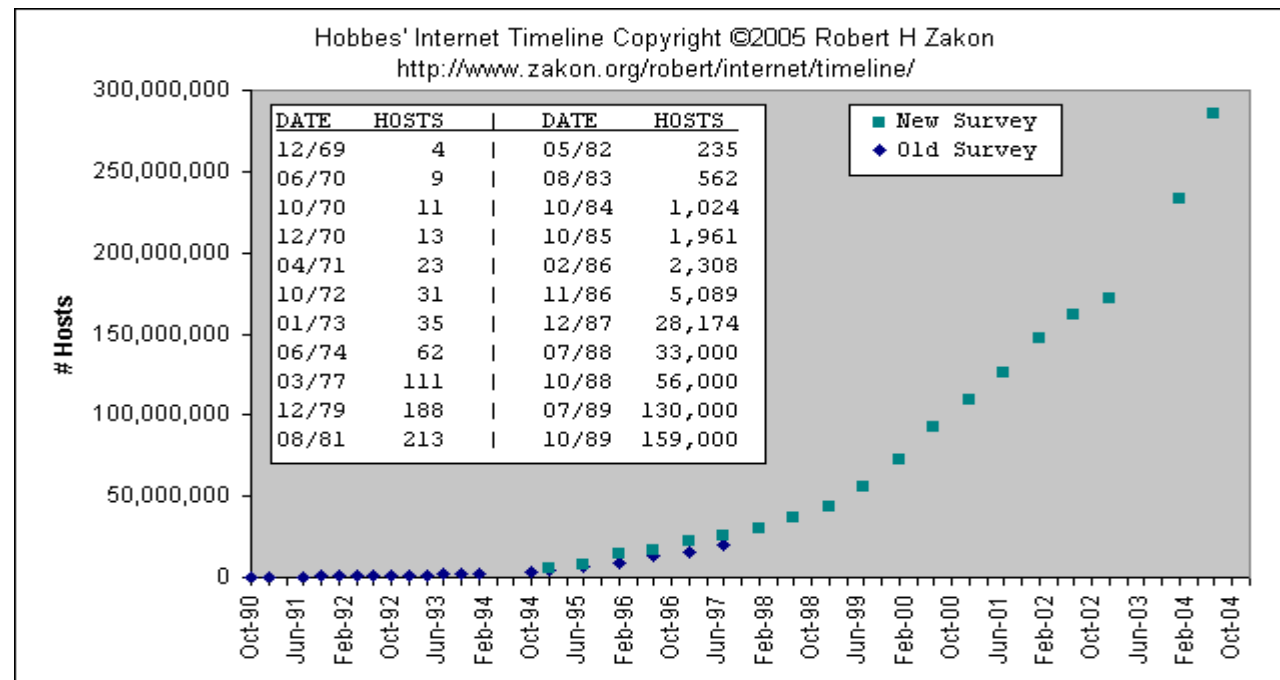
Comitê Gestor da Internet no Brasil



O CEPTRO.br – Centro de Estudos e Pesquisas em Tecnologia de Redes e Operações – é responsável por **projetos que visam melhorar a qualidade da Internet** no Brasil e disseminar seu uso, com especial atenção para seus aspectos técnicos e de infraestrutura. O CEPTRO.br é responsável, por exemplo, pelo PTT Metro, pelo NTP.br, e projeto IPv6.br.

Por que IPv6

- Em 1983 a Internet era uma rede acadêmica com aproximadamente 100 computadores...
- Em 1993 iniciou-se seu uso comercial.
- O crescimento foi exponencial!
- O crescimento, aliado à política vigente de alocação de endereços, faria com que esses se esgotassem num prazo de 2 ou 3 anos. Previam-se um colapso no crescimento da rede!



Por que IPv6

Distribuição histórica de IPv4

Endereços Ipv4 tem 32 bits: X.X.X.X

(/8)

- Sub-redes Classe A:
de 00000000.X.X.X 0.*.*.*
até 01111111.X.X.X 127.*.*.*
(128 segmentos com 16M de endereços cada)
(/16)
- Sub-redes Classe B:
de 10000000.00000000.X.X 128.0.*.*
até 10111111.11111111.X.X 191.255.*.*
• (16K segmentos com 64K endereços cada)
(/24)
- Sub-redes Classe C:
de 11000000.00000000.00000000.X 192.0.0.*
até 11011111.11111111.11111111.X 213.255.255.*
• (2M segmentos com 256 endereços cada)
- Os 32 /8 restantes reservados para Multicast (16)
e para IANA (16)

Mapa da Internet

Por que IPv6

Tecnologias como:

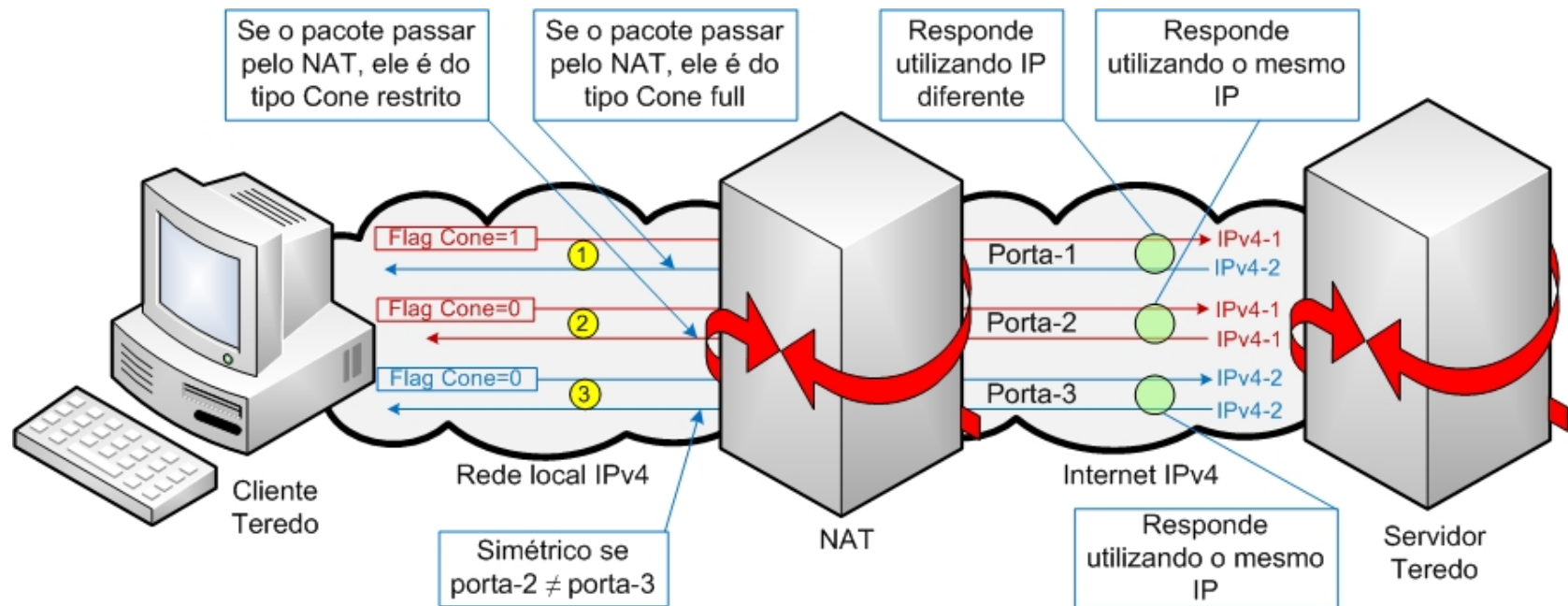
- CIDR (roteamento sem uso de classes – permite um melhor aproveitamento dos endereços disponíveis)
- RFC 1918 (endereços privados – permite o uso de endereços não válidos na Internet nas redes corporativas)
- NAT (tradução de endereços – permite que com um endereço válido na Internet apenas, toda uma rede de computadores usando endereços privados seja conectada, mas com várias restrições)
- DHCP (alocação dinâmica de endereços IP – permite que provedores reutilizem endereços Internet para conexões não permanentes)

Por que IPv6

- Alguns questionam porque não utilizar o NAT indefinidamente, mas ele foi concebido como uma solução provisória!
- O NAT acaba com o modelo de funcionamento fim a fim, trazendo complicações ou impedindo o funcionamento de uma série de aplicações.
- O NAT tem alguns problemas técnicos:
 - Não é fácil manter o estado do NAT no caso de falha em um dos hosts.
 - O NAT não funciona bem com o IPsec.
 - O NAT não escala bem.
 - O NAT dá uma falsa sensação de segurança (Comporta-se como um stateful firewall)

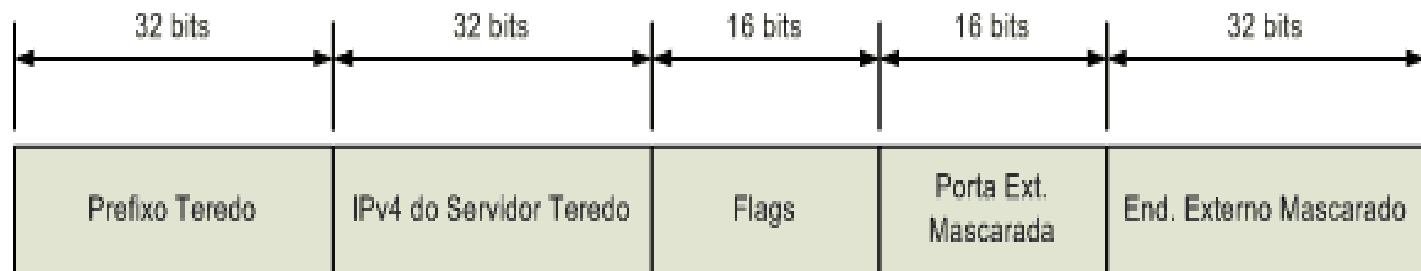
Teredo

Setup inicial



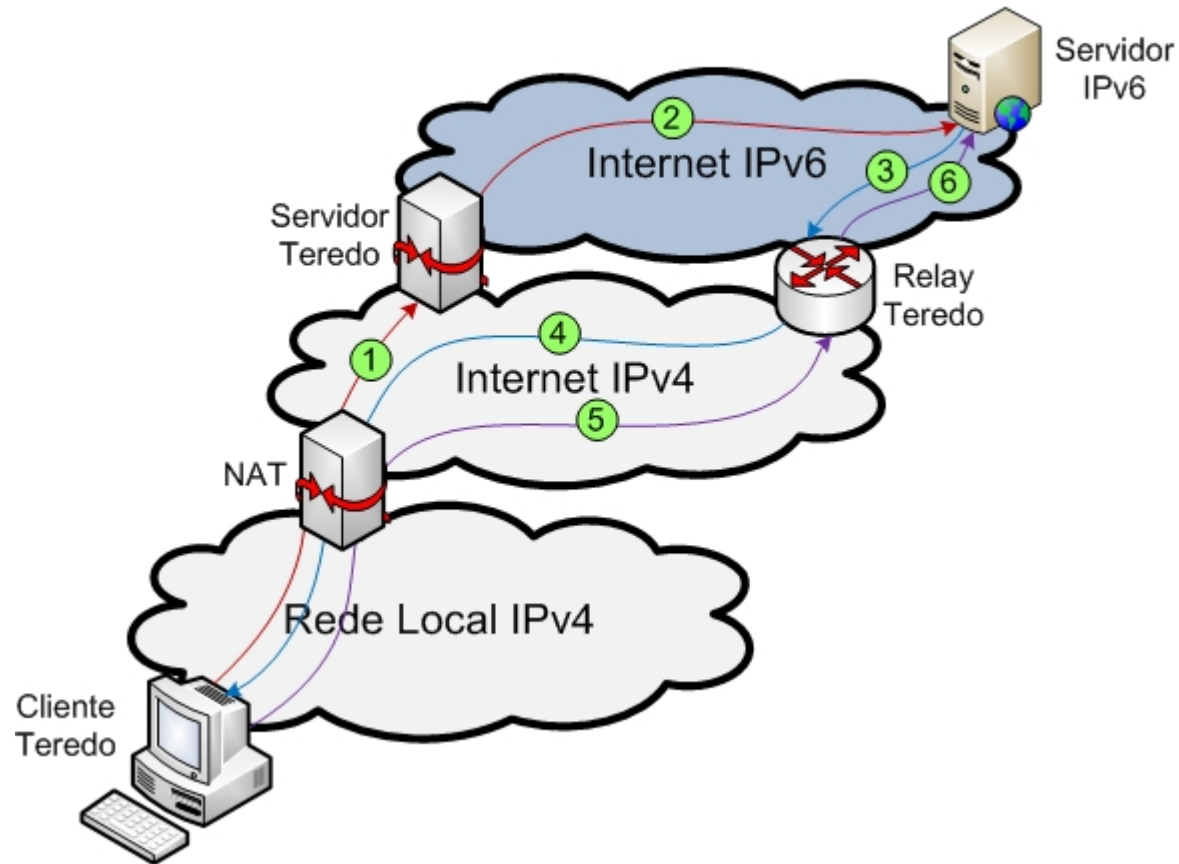
Teredo

Endereçamento do cliente IPv6/Teredo



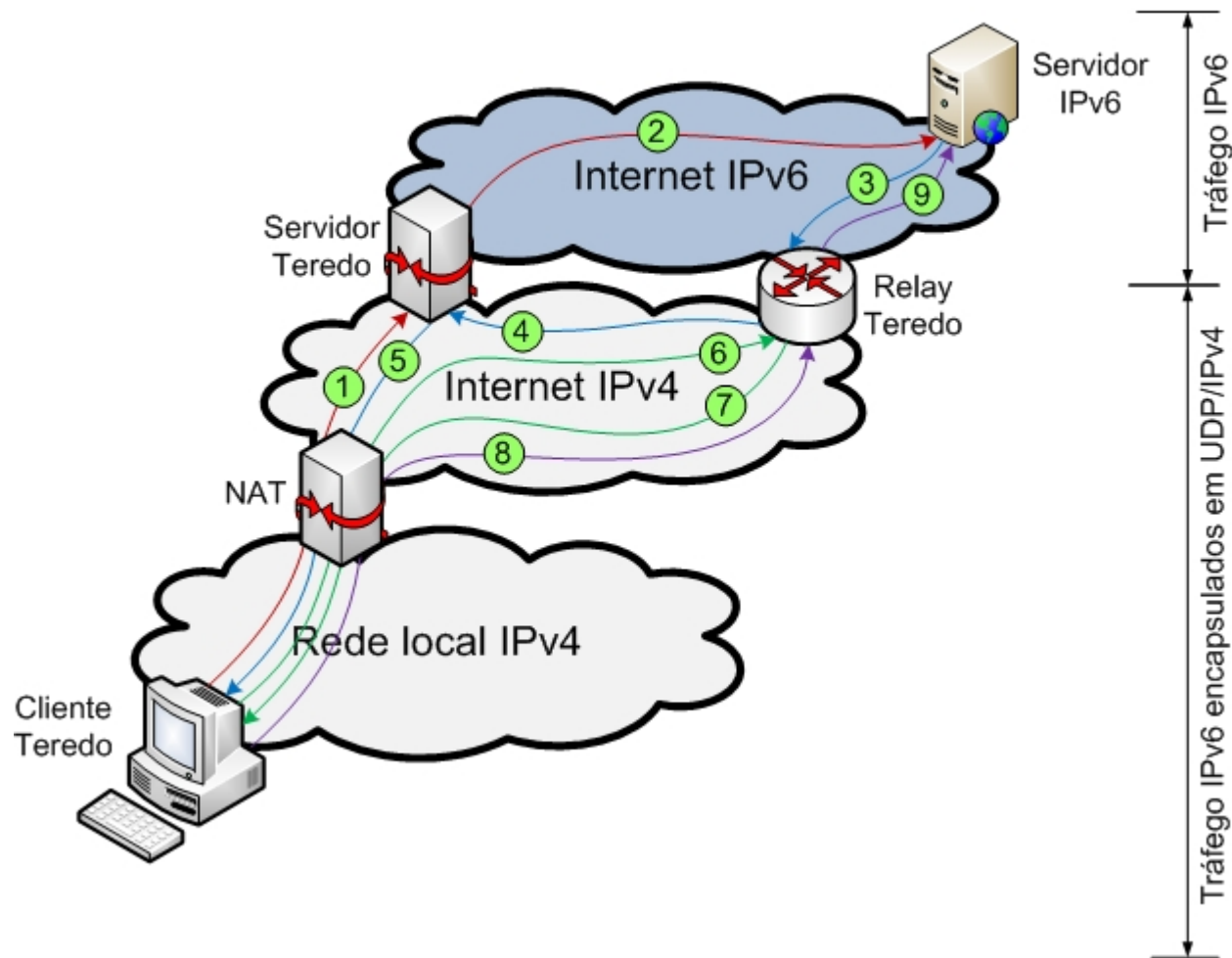
Teredo

Cliente Teredo para servidor IPv6 através de NAT tipo cone



Teredo

Cliente Teredo para servidor IPv6 através de NAT tipo restrito



Teredo

O principal problema de segurança quando se utiliza o Teredo é que seu tráfego pode passar despercebido pelos filtros e firewalls se os mesmos não estiverem preparados para interpretá-lo, sendo assim, os computadores e a rede interna ficam totalmente expostos à ataques vindos da Internet IPv6. Para resolver este problema, antes de implementar o Teredo, deve se fazer uma revisão nos filtros e firewalls da rede ou pelo menos dos computadores que utilizarão esta técnica. Além deste problema, ainda temos os seguintes:

- O cliente Teredo divulga na rede a porta aberta por ele no NAT e o tipo de NAT que ele está utilizando, possibilitando assim um ataque através dela;
- A quantidade de endereços Teredo é bem menor que os IPv6 nativos, facilitando assim a localização de computadores vulneráveis;
- Um ataque por negação de serviço é fácil de ser aplicado tanto no cliente quanto no relay;
- Devido ao método de escolha do Relay pelo host de destino, pode se criar um Relay falso e utilizá-lo para coletar a comunicação deste host com os seus clientes.

Configurando o Linux como cliente Teredo

Edite o arquivo de configuração do miredo:

```
# nano /usr/local/etc/miredo/miredo.conf
```

Especifique o nome ou IP do servidor que você irá utilizar, você pode utilizar até 2 servidores, sendo que o primeiro é especificado pela entrada “ServerAddress” e o segundo por “ServerAddress2” seguida do IP ou nome do servidor:

```
#!/usr/local/sbin/miredo -f -c
```

```
# Nome da interface utilizada no tunel.  
InterfaceName teredo
```

```
#Dependendo das regras do seu firewall/NAT ou tipo de NAT,  
#voce precisa fixar a porta e o endereço IP a ser utilizado  
#BindPort 3545  
#BindAddress 192.0.2.100
```

```
#Servidores Teredo a serem utilizados(o maximo é 2)  
ServerAddress teredo-debian.remlab.net  
ServerAddress2 teredo.ipv6.microsoft.com
```

Configurando FreeBSD como cliente Teredo

Execute o seguinte comando para instalar o pacote do Miredo:

```
# pkg_add -r miredo
```

Edite o arquivo `/usr/local/etc/miredo-server.conf`

Insira as seguintes linhas ou altere os seguintes parâmetros:

```
#IP's onde o servidor estará ativo  
ServerBindAddress <IP-1>  
ServerBindAddress2 <IP-2>
```

Entre no diretório `/usr/local/etc/rc.d` e execute o seguinte comando(isto é necessário devido a um bug no pacote de instalação):

```
# ln miredo-server miredo_server
```

Crie o diretório onde o Miredo irá gravar o seu PID:

```
# mkdir /usr/local/var  
# mkdir /usr/local/var/run
```

Edite o arquivo de configuração do miredo:

```
#edit /usr/local/etc/miredo/miredo.conf
```

Especifique o nome ou IP do servidor que você irá utilizar, você poderá especificar até 2, sendo que o primeiro é pela entrada “ServerAddress” e o segundo por “ServerAddress2” seguida do IP ou nome do servidor:

```
#!/usr/local/sbin/miredo -f -c
```

```
# Nome da interface utilizada no tunel.  
InterfaceName teredo
```

```
#Dependendo das regras do seu firewall/NAT ou tipo de NAT,  
#voce precisa fixar a porta e o endereço IP a ser utilizado  
#BindPort 3545  
#BindAddress 192.0.2.100
```

```
#Servidores Teredo a serem utilizados(o máximo é 2)  
ServerAddress teredo-debian.remlab.net  
ServerAddress2 teredo.ipv6.microsoft.com
```

Inicialize-o com o seguinte comando para verificar se está tudo ok:

```
# /usr/local/sbin/miredo -f
```

Configurando Windows como Cliente Teredo

Por motivos de segurança, faça todas as atualizações disponíveis através do Windows Update;

Instale o suporte ao IPv6 utilizando o seguinte comando:

```
> netsh int ipv6 install
```

Para configurar o cliente, simplesmente execute o comando abaixo em uma janela do DOS ou clicando em “Iniciar” e depois “Executar”

```
> netsh int ipv6 set teredo <tipo de NAT> <ip do servidor teredo>
```

Sendo que o tipo de NAT poderá ser “cliente” ou “enterpriseclient”

Para verificar a conexão e configuração do cliente, execute o seguinte comando:

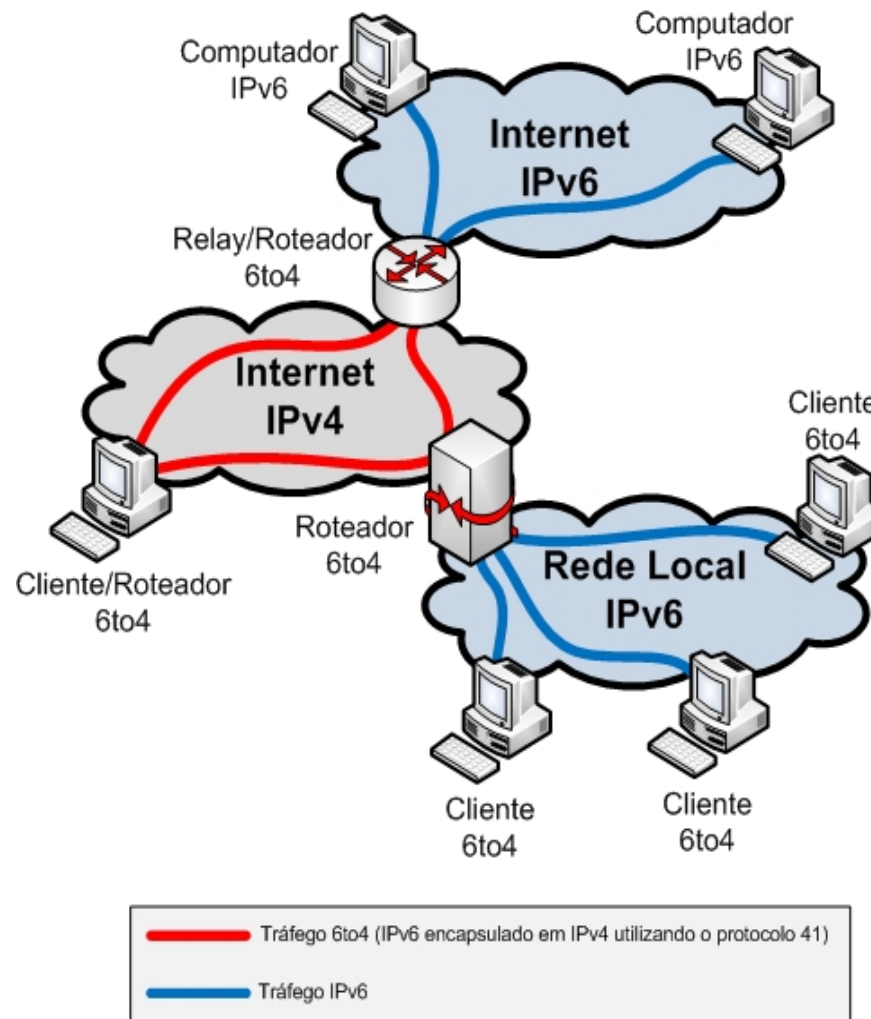
```
> netsh int ipv6 show teredo
```

No caso do erro abaixo (destacado em verde), utilize o tipo de NAT “enterpriseclient”

```
C:\Documents and Settings\Projetos>netsh int ipv6 show teredo
Parâmetros Teredo
-----
Tipo                : client
Nome do servidor    : 200.160.1.100
Intervalo de atualização de cliente: default
Porta cliente       : default
Estado              : offline
Erro                 : cliente é uma rede gerenciada
```

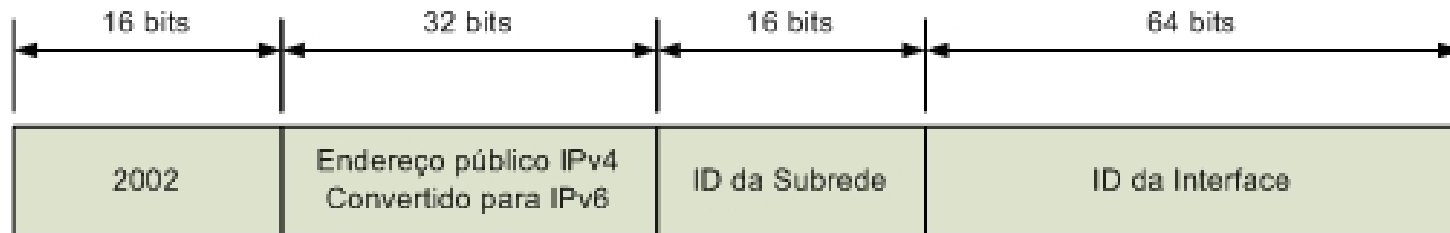
6to4

Componentes básicos



6to4

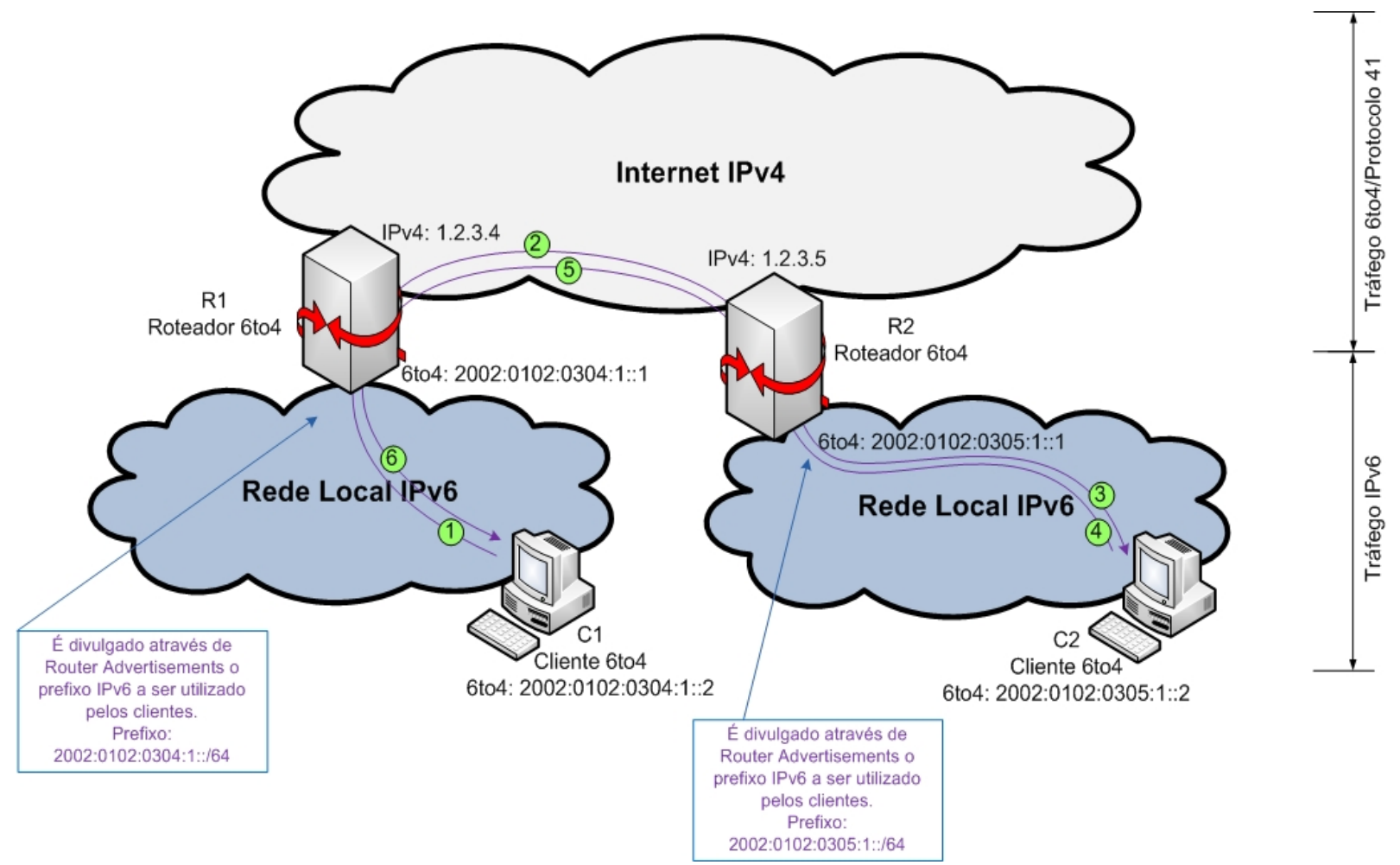
Endereçamento



- O prefixo 6to4 é sempre 2002, conforme definição da IANA;
- O próximo campo, IPv4 público do cliente, convertido para hexadecimal;
- O ID da subrede é utilizado apenas para segmentar a rede 6to4;
- O ID da interface pode ser igual ao segundo campo (IPv4 convertido para hexadecimal) no caso da configuração automática do Windows Vista e Server 2008 ou então 1, 2, 3, 4... no caso de configuração manual ou do Linux e BSD.

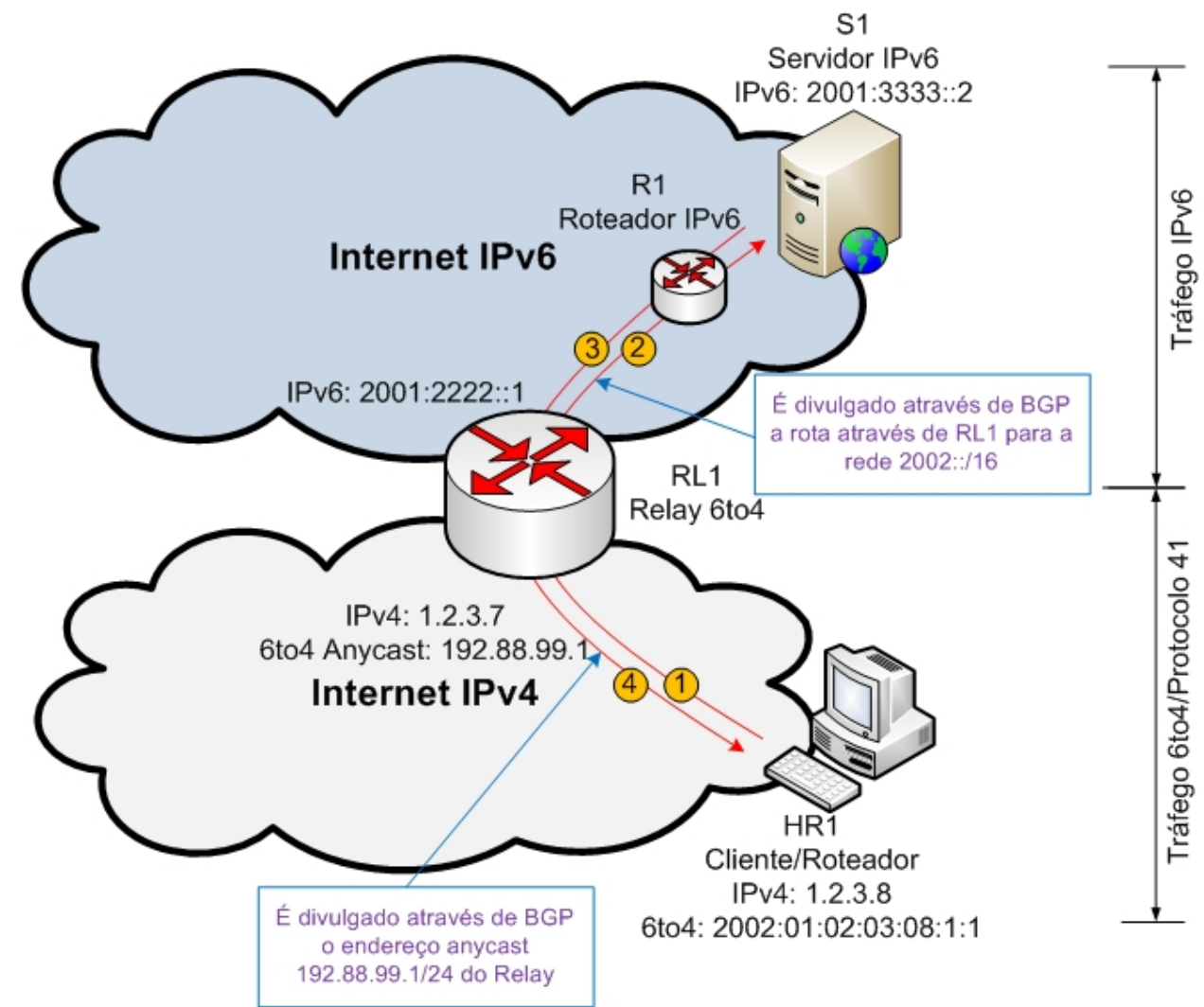
6to4

Comunicação Cliente 6to4 com Cliente 6to4 em redes diferentes



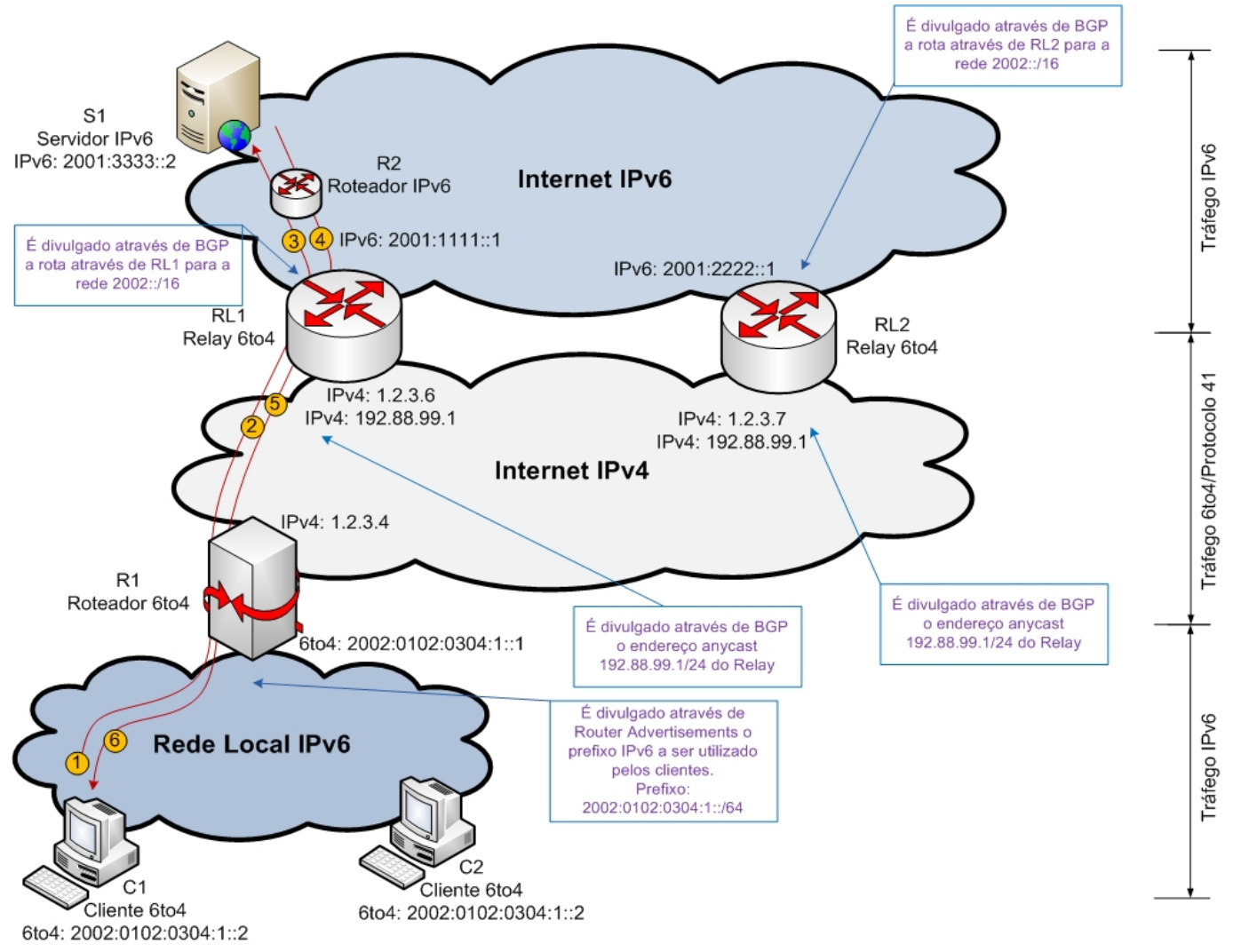
6to4

Comunicação Cliente/Roteador 6to4 com servidor IPv6



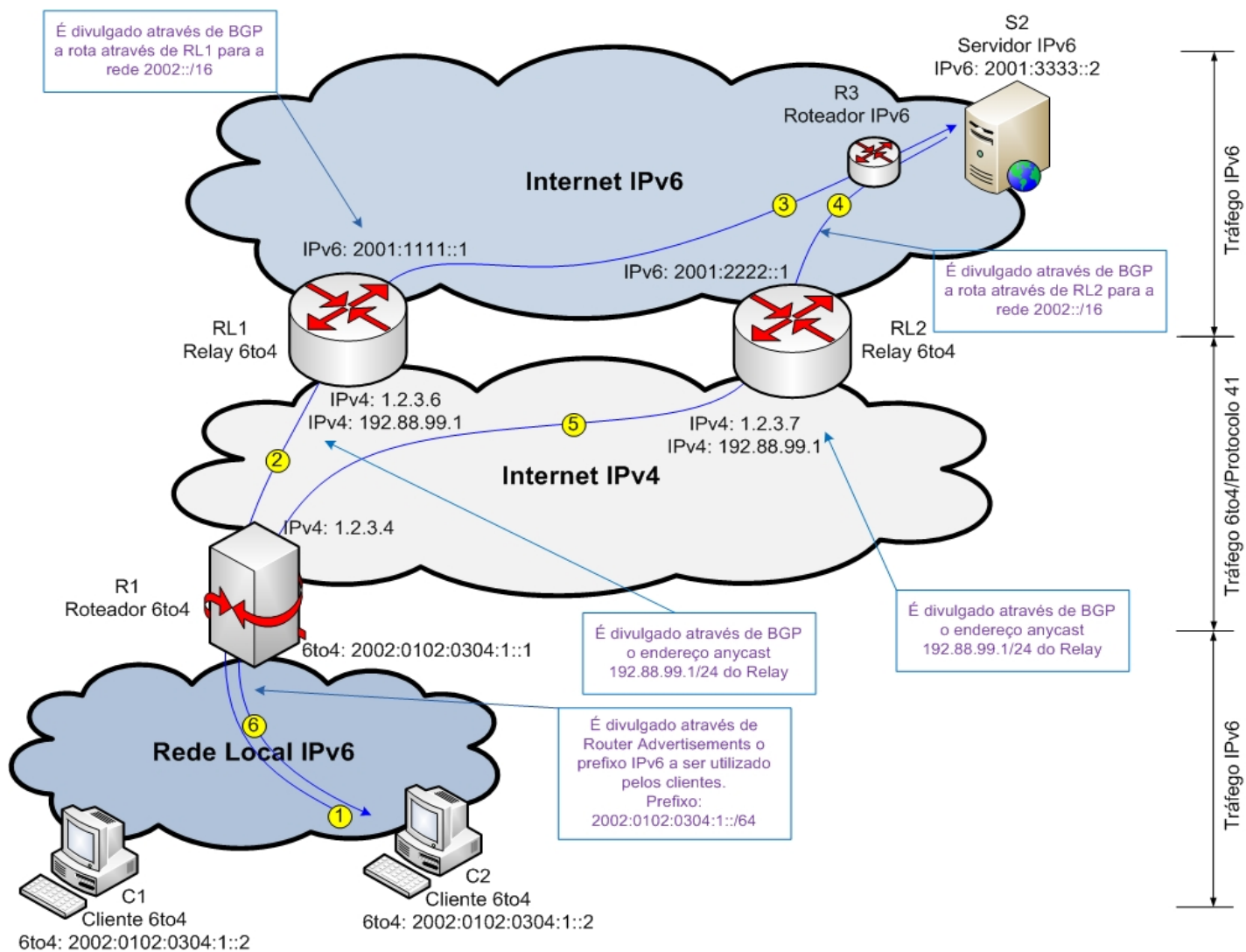
6to4

Comunicação Cliente 6to4 com servidor IPv6 utilizando apenas um Relay 6to4 (Rota de ida e volta iguais):



6to4

Comunicação Cliente 6to4 com servidor IPv6 utilizando dois relays 6to4 diferentes (Rota de ida e volta diferentes)



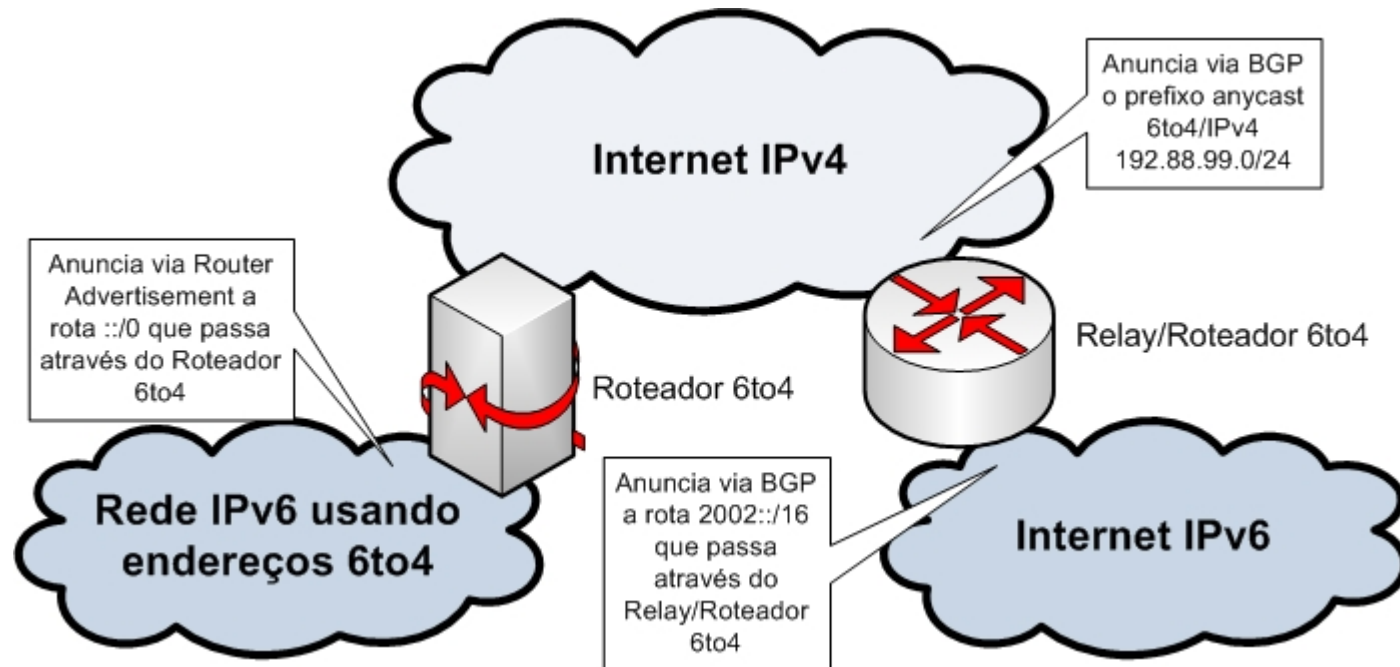
6to4

Segurança

- O Relay roteador não verifica os pacotes IPv6 que estão encapsulados em IPv4, apesar dele os encapsular e desencapsular;
- O spoofing de endereço é um problema grave de túneis 6to4, podendo ser facilmente explorado;
- Não há um sistema de autenticação entre o roteador e o Relay roteador, facilitando assim a exploração de segurança através da utilização de Relays roteadores falsos.

6to4

Implementação



6to4

Cliente/Roteador - Linux

- Instale o suporte ao IPv6:

```
# modprobe ipv6
```

- Ative o roteamento IPv6, editando o arquivo /etc/sysctl.conf e adicionando a seguinte linha:

```
net.ipv6.conf.default.forwarding=1
```

- Converta o endereço IPv4 para Ipv6/6to4 utilizando o seguinte comando:

Exemplo de conversão do IPv4 207.192.20.30 para 6to4:

```
# printf "2002:%02x%02x:%02x%02x::1\n" 207 192 20 30
```

6to4

Cliente/Roteador - Linux

- No caso do Debian e Ubuntu, edite o arquivo `/etc/network/interfaces` e acrescente a interface 6to4 conforme o seguinte exemplo:

```
auto sit0
iface sit0 inet6 static
    address 2002:c000:0203::1 # IPv4 convertido para 6to4
    netmask 16
    gateway ::192.88.99.1 # endereço do relay a ser utilizado
```

- Nos outros casos você pode utilizar um script para ativar o túnel 6to4, sendo assim, faça o download do script utilizando o seguinte comando:

```
# wget -c http://sites.inka.de/bigred/sw/6to4
```

6to4

Cliente/Roteador - Linux

Se você não for utilizar o relay padrão que é obtido via anycast (192.88.99.1), você terá que alterar duas variáveis no script:

```
REMOTE4=<IPv4 do Relay>  
REMOTE6=<IPv6 6to4 do Relay>
```

Com tudo configurado, você já poderá iniciar o tunel 6to4 executando o seguinte comando:

```
# ./6to4 up <IPv4 da Interface ligada à Internet> <interface ligada à sua rede local>
```

Exemplo: # ./6to4 up 200.192.170.10 eth0

Para desativar o túnel, você executa o seguinte comando:

```
# 6to4 down <IPv4 da Interface ligada à Internet> <interface ligada à sua rede local>
```

Exemplo: # 6to4 down 200.192.170.10 eth0

6to4

Cliente/Roteador - FreeBSD (7.0 e 6.3)

- Edite o arquivo /etc/rc.conf executando o seguinte comando:
- Acrescente ou altere as seguintes configurações para ativar o IPv6 e o 6to4:

```
ipv6_enable="YES"  
ipv6_network_interfaces="auto"  
ipv6_defaultrouter="2002:c058:6301::" # IPv6/6to4 do relay a ser utilizado  
stf_interface_ipv4addr="201.111.222.123" #IPv4 do computador
```

- Depois de reiniciar o computador, uma interface stf0 deverá ser inicializada automaticamente com um endereço 6to4 e todas as rotas também deverão já estar configuradas. Agora podemos verificar se tudo está funcionando corretamente executando o seguinte comando:

```
# traceroute6 ipv6.google.br
```

ou se o DNS não estiver resolvendo IPv6

```
# traceroute6 2001:4860:0:2001::68
```

6to4

Roteador - Linux

- Instale o suporte ao IPv6:

```
# modprobe ipv6
```

- Ative o roteamento IPv6, editando o arquivo /etc/sysctl.conf e adicionando a seguinte linha:
net.ipv6.conf.default.forwarding=1

- Faça o download do script de configuração em utilizando o seguinte comando:

```
# wget -c http://sites.inka.de/bigred/sw/6to4
```

- Se você for utilizar um relay diferente do padrão 192.88.99.1, é necessário que você modifique o a configuração do script de inicialização do roteador 6to4. Para configurá-lo, você precisará converter o endereço do relay para o formato do 6to4, para isto, utilize o seguinte comando:

Exemplo de conversão do IPv4 207.192.20.30 para 6to4:

```
# printf "2002:%02x%02x:%02x%02x::\n" 207 192 20 30
```

Depois disso, edite o script e altere as seguintes variáveis:

```
REMOTE4=<IPv4 do Relay>
```

```
REMOTE6=<IPv6 6to4 do Relay>
```

6to4

Roteador - Linux

- Instale o serviço de router advertisement:
Para o Debian e Ubuntu utilize o seguinte comando:

```
# apt-get install radvd
```

- Configure o Radvd editando ou criando o arquivo `/etc/radvd.conf` com o seguinte conteúdo:

```
interface eth0 { # ajuste de acordo com a interface conectada a sua rede local
    AdvSendAdvert on;
    MinRtrAdvInterval 20;
    MaxRtrAdvInterval 60;
    AdvLinkMTU 1400; # ajuste de acordo com suas necessidades
    prefix 2002::/64 {
        AdvOnLink off;
        AdvAutonomous on;
        AdvRouterAddr on;
        Base6to4Interface tun64;
        AdvPreferredLifetime 90;
        AdvValidLifetime 120;
    };
};
```

6to4

Roteador - Linux

- Com tudo configurado, você já poderá iniciar o seu roteador executando o seguinte comando:

```
# 6to4 up <IPv4 da Interface ligada à Internet> <interface ligada à sua rede local>
```

Exemplo:

```
# 6to4 up 200.192.170.10 eth0
```

- Para testar se o roteador 6to4 está com conectividade à rede IPv6, execute o seguinte comando:

```
# traceroute6 ipv6.google.com
```

ou se o DNS não estiver resolvendo IPv6

```
# traceroute6 2001:4860:0:2001::68
```

- Para testar se o roteador esta funcionando corretamente, coloque um computador com suporte a IPv6 na rede local e execute novamente o comando acima, ela deverá pegar automaticamente um IPv6 6to4 conforme o prefixo anunciado pelo Radvd.

6to4

Roteador - Freebsd (7.0 e 6.3)

- Converta o IPv4 da interface externa para o formato 6to4/IPv6:

Exemplo de conversão do IPv4 207.192.20.30 para IPv6/6to4:

```
# printf "2002:%02x%02x:%02x%02x::\n" 207 192 20 30
```

- Converta o IPv4 do Relay 6to4 para o formato 6to4/IPv6:

Exemplo de conversão do IPv4 192.88.99.1 para IPv6/6to4:

```
# printf "2002:%02x%02x:%02x%02x::\n" 192 88 99 1
```

6to4

Roteador - FreeBSD (7.0 e 6.3)

- Acrescente ou altere as seguintes linhas no arquivo /etc/rc.conf para ativar o suporte ao IPv6 e a interface virtual 6to4. Os itens em vermelho devem ser configurados de acordo com sua configuração de rede:

```
##### IPv4 #####
ifconfig_inc0="inet 123.123.123.123 netmask 255.255.255.0" # IPv4 Externo
##### IPv6 #####
ipv6_enable="YES" # habilita o suporte ao IPv6
ipv6_network_interfaces="auto" # Interfaces onde o IPv6 será ativado.
ipv6_gateway_enable="YES" # Ativa o roteamento
##### 6to4 #####
ipv6_default_interface="stf0" # Interface virtual 6to4
ipv6_ipv4mapping="YES" # habilita o mapeamento IPv4->IPv6, possibilitando a utilização
# de endereços do tipo ::ffff:a.b.c.d
ipv6_defaultrouter="2002:c058:6301::" # Endereço do Relay 6to4 geralmente 192.88.99.1
ipv6_ifconfig_inc1="2002:7b7b:7b7b:1::1 prefixlen 48" # ipv6 da rede interna, baseado no ipv4
externo
stf_interface_ipv4addr="123.123.123.123" # Mesmo IPv4 da interface Internet IPv4
stf_interface_ipv4plen="0" # Tamanho do prefixo do endereço 6to4/IPv4, geralmente 0
stf_interface_ipv6_ifid="0:0:0:2" # Número final do IP a ser utilizado na interface stf0.
##### Router Advertisement #####
rtadvd_enable="YES" # Habilita o anuncio de roteador
rtadvd_interfaces="inc1" # Ativa na interface interna
```

6to4

Roteador - FreeBSD (7.0 e 6.3)

- Configure o serviço rtadvd editando o arquivo /etc/rtadvd.conf e acrescentado as seguintes linhas:

```
Inc1::\n        :addr="2002:7b7b:7b7b:1::":prefixlen#64
```

Sendo que:

- Inc1 é a interface interna(IPv6) do roteador;
 - "2002:7b7b:7b7b:1::" é o prefixo a ser utilizado na rede interna, sendo que o "1::" especifica a subrede da rede 6to4, e é o IPv4 da interface externa convertido para IPv6/6to4;
 - Prefixlen#64 é comprimento do prefixo da subrede.
- Depois disso, reinicie o computador e verifique se a interface virtual 6to4 foi criada, se o serviço rtadvd foi iniciado e se as rotas foram criadas.

6to4

- Windows Vista já ativa automaticamente o cliente 6to4 quando ele possui endereço Ipv4 Público.

- Windows XP e Windows 2003:

 - Primeiro faça todas as atualizações via Windows Update;

 - Ative o suporte ao IPv6 executando o seguinte comando:

 - > netsh int ipv6 install

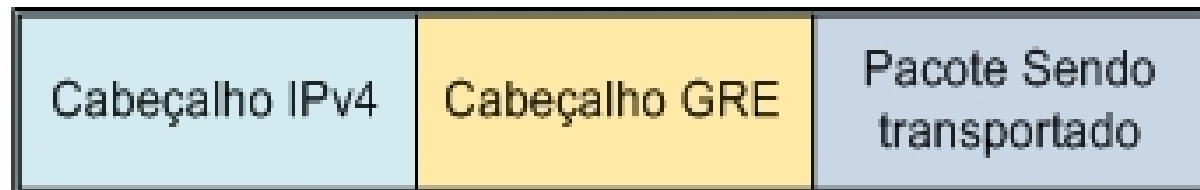
- Ative e configure o 6to4 executando o seguinte comando:

 - > netsh int ipv6 6to4 set relay <IPv4 do relay> enabled <MTU>

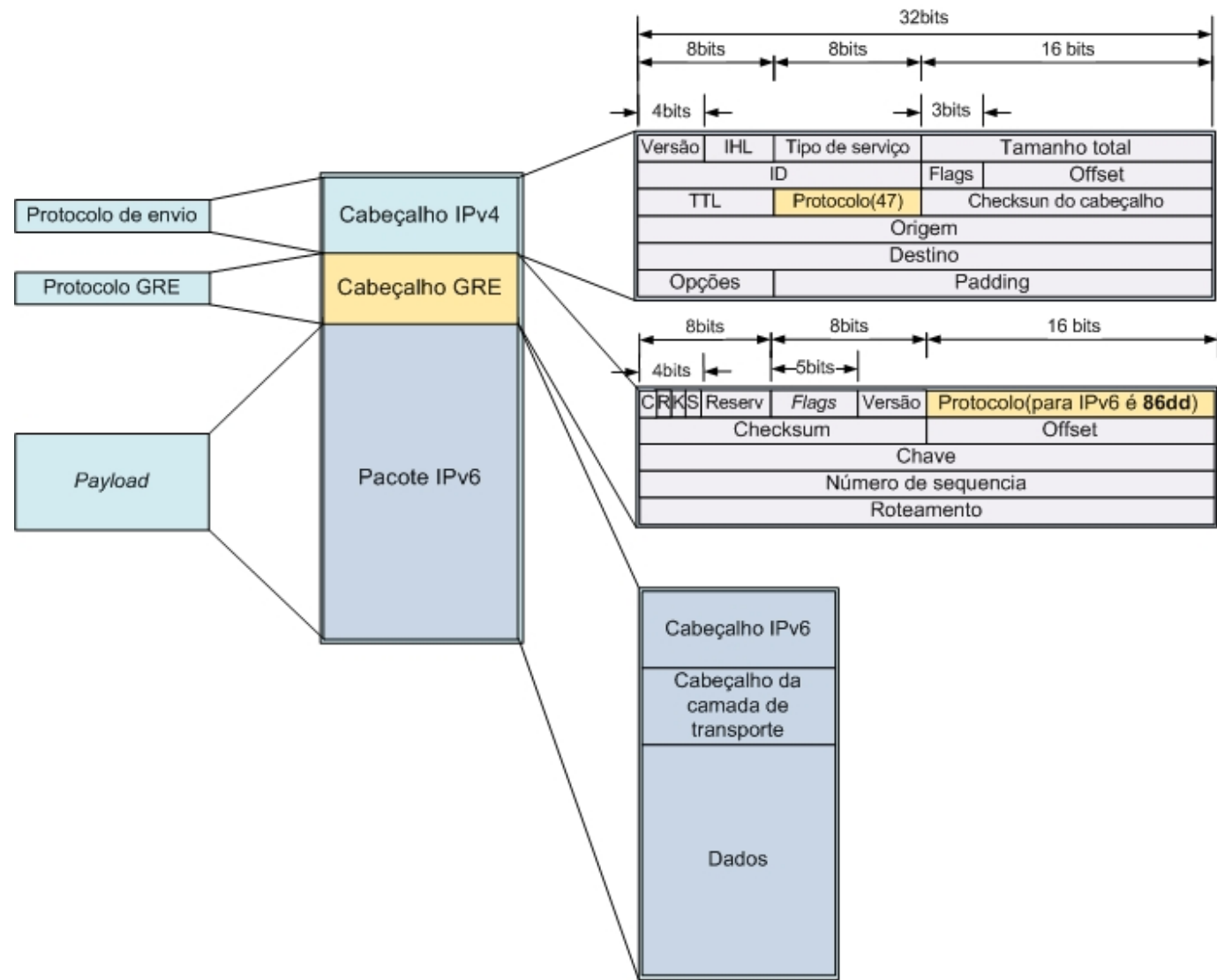
Exemplo utilizando o relay padrão anycast:

 - > netsh int ipv6 6to4 set relay 192.88.99.1 enabled 1440

Túneis GRE



Túneis GRE



Túneis GRE

Linux

O primeiro passo é verificar se o módulo GRE está carregado, executando o seguinte comando:

```
# lsmod | grep gre
```

Se não estiver carregado, execute o seguinte comando para carregá-lo:

```
# modprobe ip_gre
```

Depois disso, podemos configurar a entrada do túnel executando o seguinte comando:

```
# ip tunnel add <Nome do túnel> mode gre remote <IPv4 remoto> local <IPv4 local> ttl 255  
# ip link set <Nome do túnel> up  
# ip addr add <IPv6 local> dev <Nome do túnel>  
# ip route add <IPv6 da rede remota> dev <Nome do túnel>
```

Na outra ponta do túnel, executamos os mesmos comandos, mas, com inversão dos IP's de destino com os de origem:

```
# ip tunnel add <Nome do túnel> mode gre remote <IPv4 remoto> local <IPv4 local> ttl 255  
# ip link set <Nome do túnel> up  
# ip addr add <IPv6 local> dev <Nome do túnel>  
# ip route add <IPv6 da rede remota> dev <Nome do túnel>
```

Para testar o túnel, dê um ping em algum host da outra ponta:

```
# ping6 <IPv6 de um host remoto>
```

Túneis GRE

Linux

Edite o arquivo `/etc/sysctl.conf` e adicione as seguintes linhas para ativar o encaminhamento de pacotes IPv4 e IPv6:

```
net.inet.ip.forwarding=1  
net.inet6.ip6.forwarding=1
```

Edite o arquivo `/etc/rc.conf` e adicione as seguintes linhas para criar e configurar automaticamente o túnel:

```
cloned_interfaces="gre<Número do túnel>"  
ifconfig_gre<Número do túnel>="inet <IPv4 virtual local> <IPv4 virtual remoto> netmask <Mascara da rede virtual> tunnel <IPv4 real local> <IPv4 real remoto>"  
ipv6_ifconfig_gre<0>="<IPv6 local/Prefixo>"  
ipv6_static_routes="gre<Número do túnel>"  
ipv6_route_gre<Número do túnel>="<Prefixo da rede IPv6 remota>:: -interface gre<Número do túnel>"
```

Na outra ponta, temos que fazer a mesma coisa, só ajustando os números IP's utilizados:

```
cloned_interfaces="gre<Número do túnel>"  
ifconfig_gre<Número do túnel>="inet <IPv4 virtual local> <IPv4 virtual remoto> netmask <Mascara da rede virtual> tunnel <IPv4 real local> <IPv4 real remoto>"  
ipv6_ifconfig_gre<0>="<IPv6 real local/Prefixo>"  
ipv6_static_routes="gre<Número do túnel>"  
ipv6_route_gre<Número do túnel>="<Prefixo da rede IPv6 remota>:: -interface gre<Número do túnel>"
```

Para testar o túnel, dê um ping em algum host da outra ponta:

```
# ping6 <IPv6 da outra ponta>
```

DHCPv6

```
#opções para pedidos DHCPv6 recebidos no interface eth0
interface eth0 {
#os clientes devem renovar o endereço passados 3 horas
renew-time 10800;
#o servidor enviará o endereço do servidor DNS
option dns_servers 2001:DB8:EDE::2;
#opções para o cliente com MAC=00:00:00:00:EE:EE
#a esse cliente será atribuído o endereço 2001:DB8:EDE::10/64
  host host0 {
    duid 00:00:00:00:EE:EE;
    address {
      2001:DB8:EDE::10/64;
    };
  };
#opções que definem os endereços para atribuir aos cliente os
#endereços 2001:DB8:EDE::10 até 2001:DB8:EDE::20
  link C{
    range 2001:DB8:EDE::10 to 2001:DB8:EDE::20/64;
  };
};
```

DNSv6

```
;  
; Local hosts  
; -----  
host1          IN      AAAA    3FFE:800::2A8:79FF:FE32:1982  
               IN      AAAA    3FFE:800::80  
www            IN      CNAME   host1.v6.ipv6domain-tottaro.org.  
;  
host2          IN      AAAA    2001:200:1000:0:25F:23FF:FE80:1234  
;  
host3          IN      AAAA    3FFE:801:1000::2EF:6FFF:FE11:2222  
host4          IN      AAAA    3FFE:801:2000:100:280:9AFF:FE80:3333  
;
```

