
Tutorial de IPv6 abordando problemas e soluções

Fabício Tamusiunas

GTRH – Grupo de Trabalho Formação de
Recursos Humanos / Comitê Gestor Internet BR

IPv6 – visão geral

- Sucessor do IPv4, com 128 bits de tamanho
- Auto configuração (“plug and ping”)
- Implementação de IPsec “mandatório”
- Melhor suporte para Mobile IP que o IPv4
- Cabeçalho simplificado: Apesar do endereçamento ser 4 vezes maior, o cabeçalho é apenas 2 vezes maior.

IPv6 – visão geral

- Endereços são escritos em hexadecimal:
 - FEDC:BA98:7654:3210:FEDC:BA98:7654:0321
- Zeros podem ser truncados:
 - 00FF:0000:0000:0000:0000:9085:9043:1234
 - FF::9085:9043:1234

IPv6 – razões de interesse

- Capacidade de ganho comercial
 - Utilizado comercialmente na Ásia
- Suporte a novos tipos de aplicação
 - Inexistência de NAT, partindo novamente ao princípio de end-to-end
 - Uso potencial de ADSL+802.11b+IPv6 para usuários domésticos
 - Alcançabilidade maior, com recursos de mobilidade avançados
- 3G está preparado para IPv6 (3GPP Release 5)

IPv6 – peer to peer

- Internet atualmente trabalha com foco cliente-servidor
 - Web, e-mail, etc
 - Clientes são geralmente domésticos, usando IP privado e servidores usam IPs válidos globalmente
 - NAT “geralmente” funciona bem neste caso

IPv6 – peer to peer

- Aplicações futuras tendem a usar aplicações cliente-cliente, ou p2p
 - VoIP, envio de mensagens, compartilhamento de arquivos, jogos on-line
 - Implica em dispositivos globalmente endereçados
 - Tendência de servidores em casa
 - Múltiplas web cams
 - Neste caso NAT se torna sensivelmente mais complexo

Sistemas operacionais com suporte nativo a IPv6

- Windows XP/.NET
- *BSD
- Linux
- Solaris 8+
- Symbian OS (PDA)
- Compaq Tru64
- AIX 4.3+
- HP/UX 11.0+
- Irix
- Cisco IOS
- Juniper
- Hitachi
- Zebra, ZebOS
- Ericsson Telebit
- 6WIND (inc. ADSL)
- 3Com
- +outros...

Endereçamentos IPv6

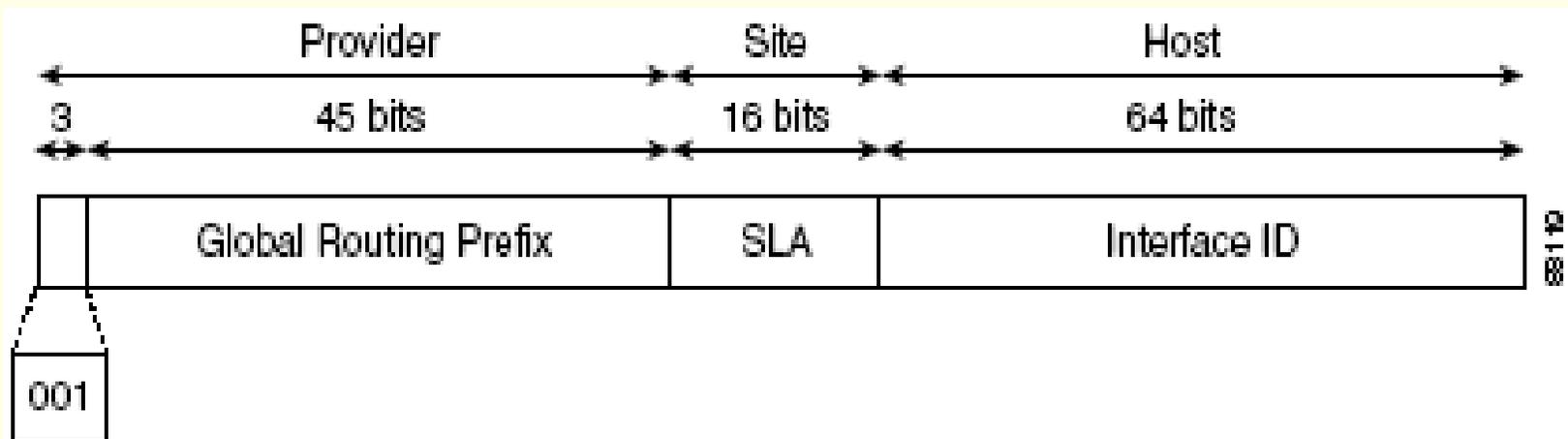
- Broadcast – **NÃO** existe mais no IPv6.
 - Suas funções foram atribuídas a endereçamento multicast
- Multicast – Adquiriu funções novas no IPv6
- Anycast – já vem especificado junto com o protocolo IPv6
- Unicast – Similar ao IPv4

Tipos de endereços unicast

- Global aggregatable address
- Site-local address (possivelmente a ser extinto pelo IETF)
- Link-local address
- IPv4-compatible IPv6 address

Unicast - Agregatable Global Address

- Baseada em tabela global de prefixos IPv6 unicast
- Definido por um prefixo global (fins de roteamento), um identificador de subrede e um identificador de interface



Unicast - Agregatable Global Address

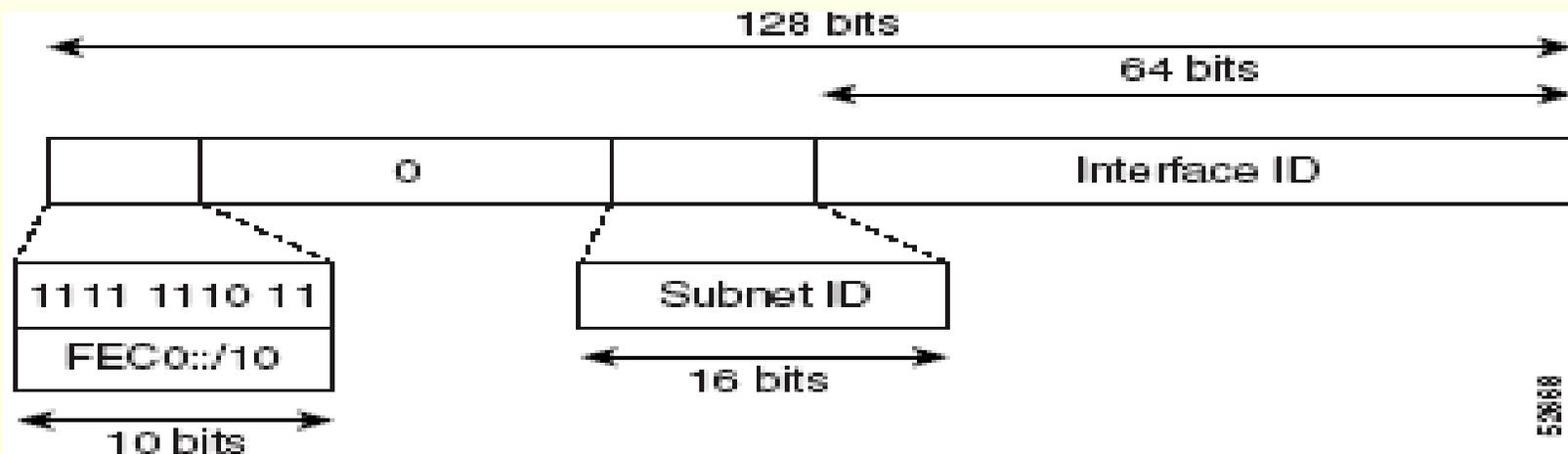
- Com exceção dos endereços começados com os bits “000”, todos os endereços tem um identificador de interface de 64 bits
- Os prefixos entre 2000::/3 e E000::/3 necessitam ter um identificador no formato (EUI)-64
- A RFC 2374 define um padrão para o prefixo global
 - Dividido em Top-Level Aggregator (TLA) e Next-Level Aggregator (NLA)
 - Antigamente existia o campos TLS que foi retirados pela IETF por serem campos baseados em políticas
 - Ainda existem redes IPv6 baseadas na estrutura antiga

Unicast - Agregatable Global Address

- Formação do campo Interface ID
 - Para quem segue o padrão IEEE 802 (interfaces Ethernet e FDDI): os três primeiros octetos (24 bits) vêm do Organizationally Unique Identifier (OUI) dos 48 bits do endereço MAC. Os quartos e quintos octetos (16 bits) tem o valor hexadecimal fixo FFFE e os três últimos octetos (24 bits) são os três últimos octetos do endereço MAC.
 - O construção do Identificados da interface é terminado com o sétimo bit do primeiro octeto setado para 0 (identificador local) ou 1 (identificador global)
 - Para quem não usa IEEE 802 (serial, loopback, ATM, Frame Relay) é usado o primeiro MAC encontrado no roteador como OUI.

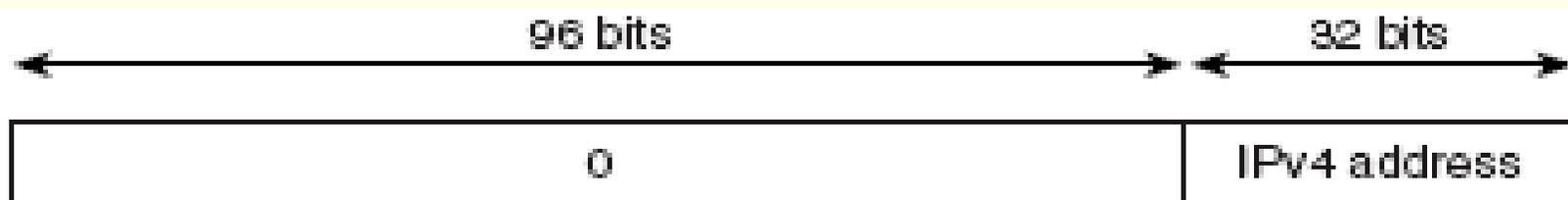
Unicast – Site-Local Address

- Endereço criado para ser usado apenas localmente (restringe a comunicação dentro do domínio)
- Inicia com o prefixo FEC0::/10
- Não é repassado por roteadores



Unicast - IPv4-Compatible IPv6 Address

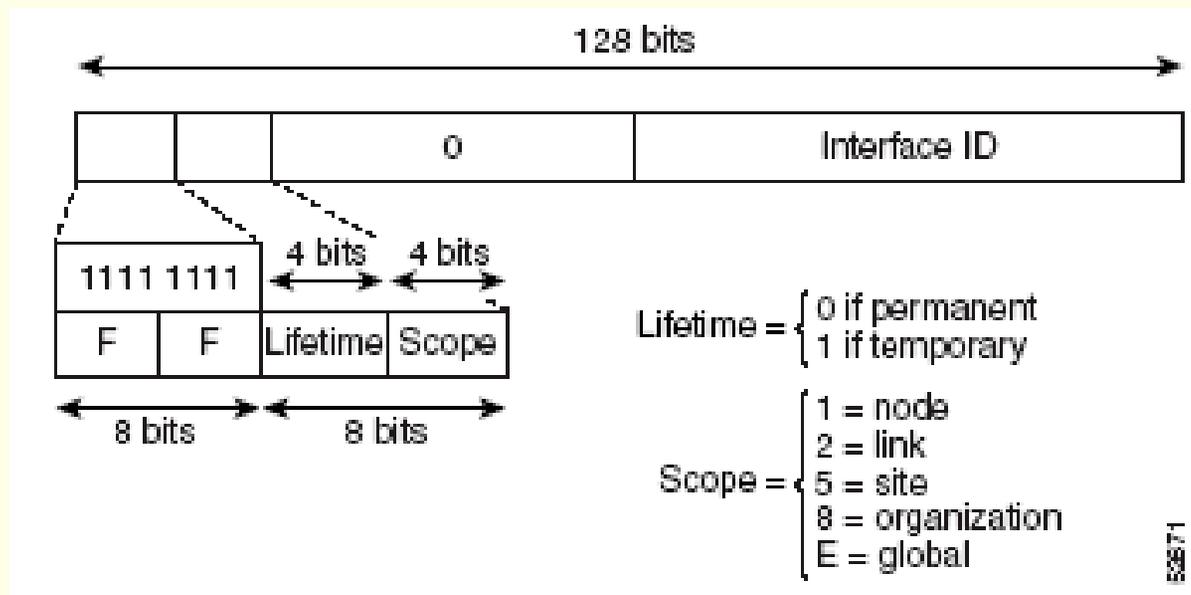
- Endereço unicast com zero nos 96 bits iniciais
- Utilizado em interfaces dual-stack e tunelamento automático
- O endereço IPv4 é embutido nos últimos 32 bits



0:0:0:0:0:0:192.168.30.1
= ::192.168.30.1
= ::C0A8:1E01

Multicast

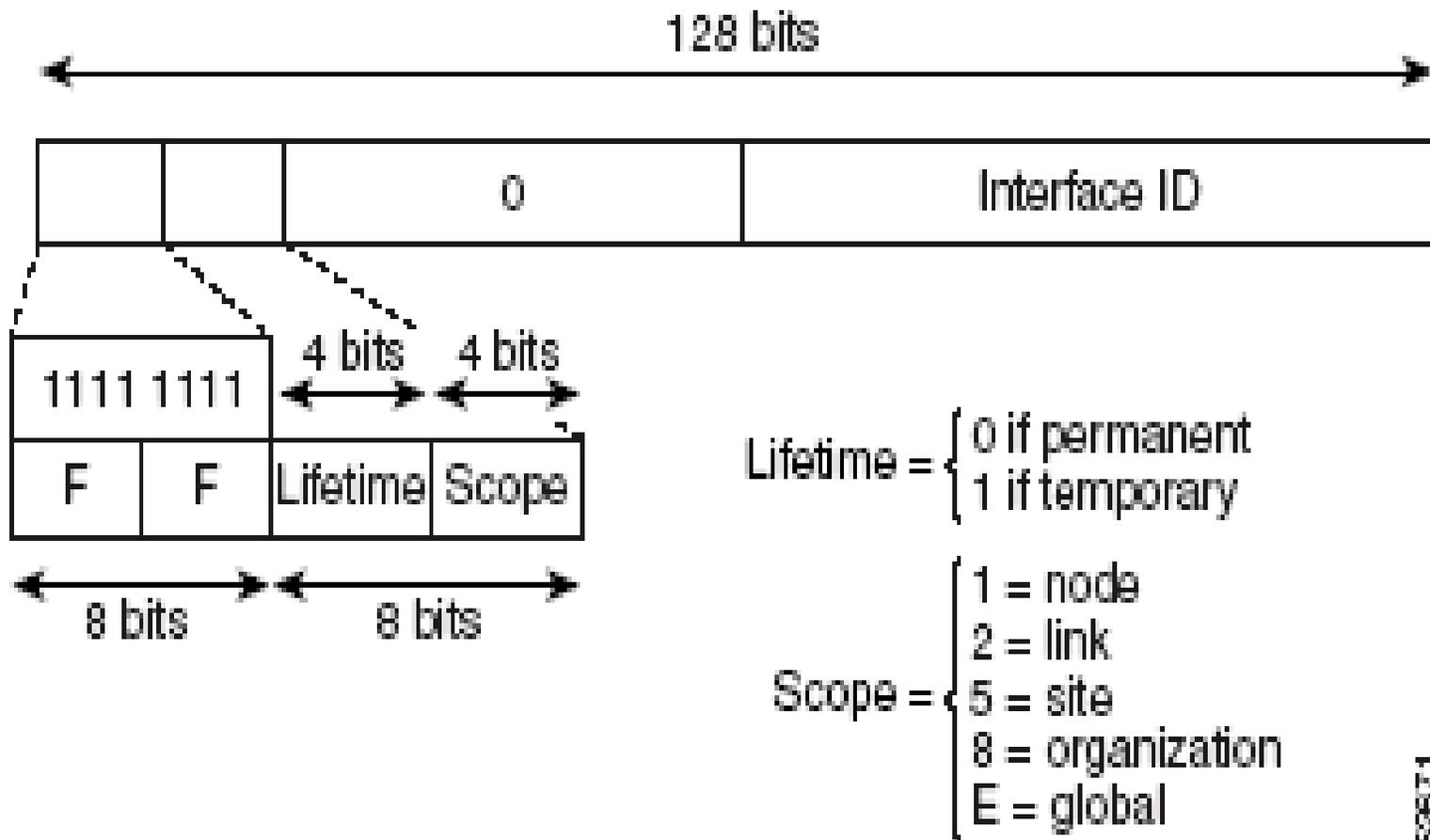
- Endereços multicast IPv6 tem o prefixo FF00::/8



Multicast

- Todos os hosts IPv6 escutam o endereço multicast FF02:0:0:0:0:0:0:1
- Todos os roteadores IPv6 escutam o endereço FF02:0:0:0:0:0:0:2

Multicast



Anycast

- Endereço designado que pode ser uma série de endereços dentro de um caminho
- Sintaticamente são endereços indistinguíveis de um endereço unicast
- Pode ser usado apenas por um roteador



Anycast em roteadores Cisco

```
interface Tunnel0
no ip address
ipv6 address 2001:2b00:300::23
ipv6 address 2001:2b00:300::50 anycast
tunnel source Ethernet0
tunnel mode ipv6ip 6to4
!
interface Ethernet0
ip address 10.0.0.1 255.255.255.0
!
```

IPv6 – Integração e Transição

- Como poderão IPv4 e IPv6 coexistirem ?
 - Incluindo a comunicação entre sistemas IPv4 e IPv6
- Diferentes ferramentas e aspectos para transição
 - Transição a nível de site
 - Transição a nível de ISP
 - Necessidade de entender as relações entre as ferramentas
- Como injetar tráfego nativo IPv6 em uma rede IPv4
 - Incluindo DNS, firewall, e-mail,...

Autoconfiguração

- Autoconfiguração “Stateless”.
 - Todos os nodos recebem o endereço FE02::1
- A autoconfiguração Stateless permite a um nodo inicializar usando o prefixo local e alguns outros números.
 - Estes números provavelmente serão os 48 bits do endereço ethernet
 - Os endereços serão “FE80::48-bit address” (multicast)
- Os hosts enviam uma mensagem de solicitação para todos os roteadores usando o endereço multicast que indica todos os roteadores (FF02::2).
 - Usado para determinar o prefixo e outros parâmetros para roteamento.

Neighbor Discovery

- RFC 2461.
- Os nodos usam Neighbor discovery para descobrir os endereços de link dos nodos vizinhos.
- Encontra links locais de hosts e roteadores.
- Detecta quais vizinhos estão ativos e mudanças em seus endereços.
- ARP não é mais usado no IPv6.
 - Esta é uma solução mais robusta para o ARP.

Neighbor Discovery

- No IPv6, as mensagens de descobrimento utilizam multicast para router discovery, neighbor discovery, etc.
- O IPv6 Neighbor Discovery combina alguns protocolos do IPv4 (ARP, ICMP Router Discovery e ICMP Redirect).
- O IPv4 não tem métodos para detectar quedas no gateway e vizinhos que caíram.

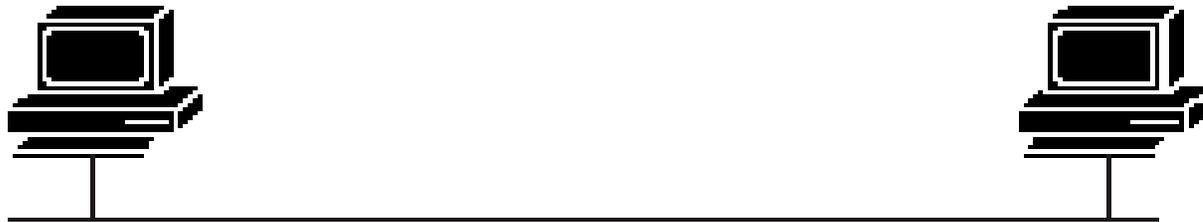
Neighbor Discovery

- O IPv6 detecta suspeita de quedas de links e links que caíram.
- Usando endereço de link local é possível identificar como anda o relacionamento com os roteadores e manter, mesmo se o endereço do provedor mudar.
- A definição de endereços é realizada na camada ICMP.

Funções do Neighbor Discovery

- Descoberta do Roteador (gateway)
- Descoberta do prefixo
- Descoberta de parâmetros
- Autoconfiguração de endereço
- Definição de endereço
- Determinação do “próximo salto”
- Detecção de vizinhos que não estão respondendo
- Detecção de endereços duplicados
- Redirecionamento

Neighbor Solicitation



ICMPv6 Type = 135

Src = A

Dst = solicited-node multicast of B

Data = link-layer address of A

Query = what is your link address?



ICMPv6 Type = 136

Src = B

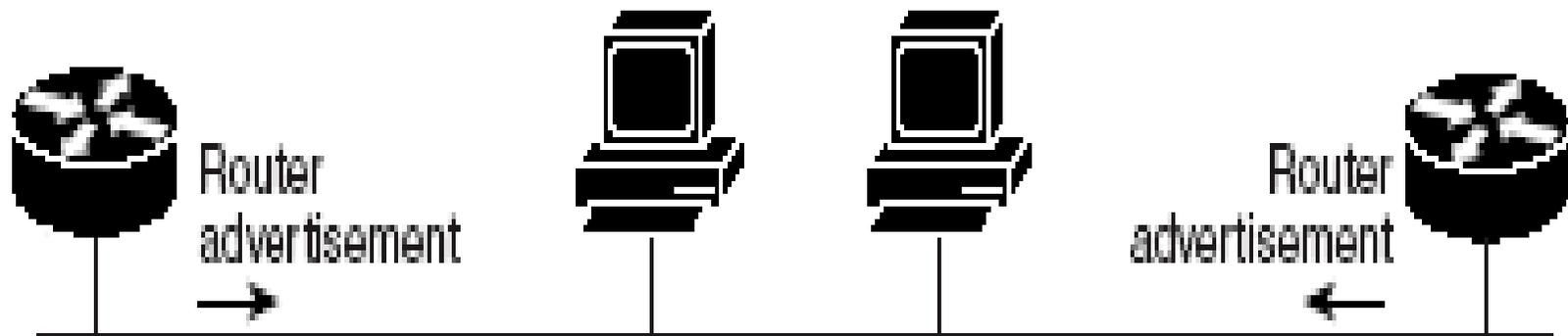
Dst = A

Data = link-layer address of B



A and B can now exchange
packets on this link

Neighbor Advertisement



Router advertisement packet definitions:

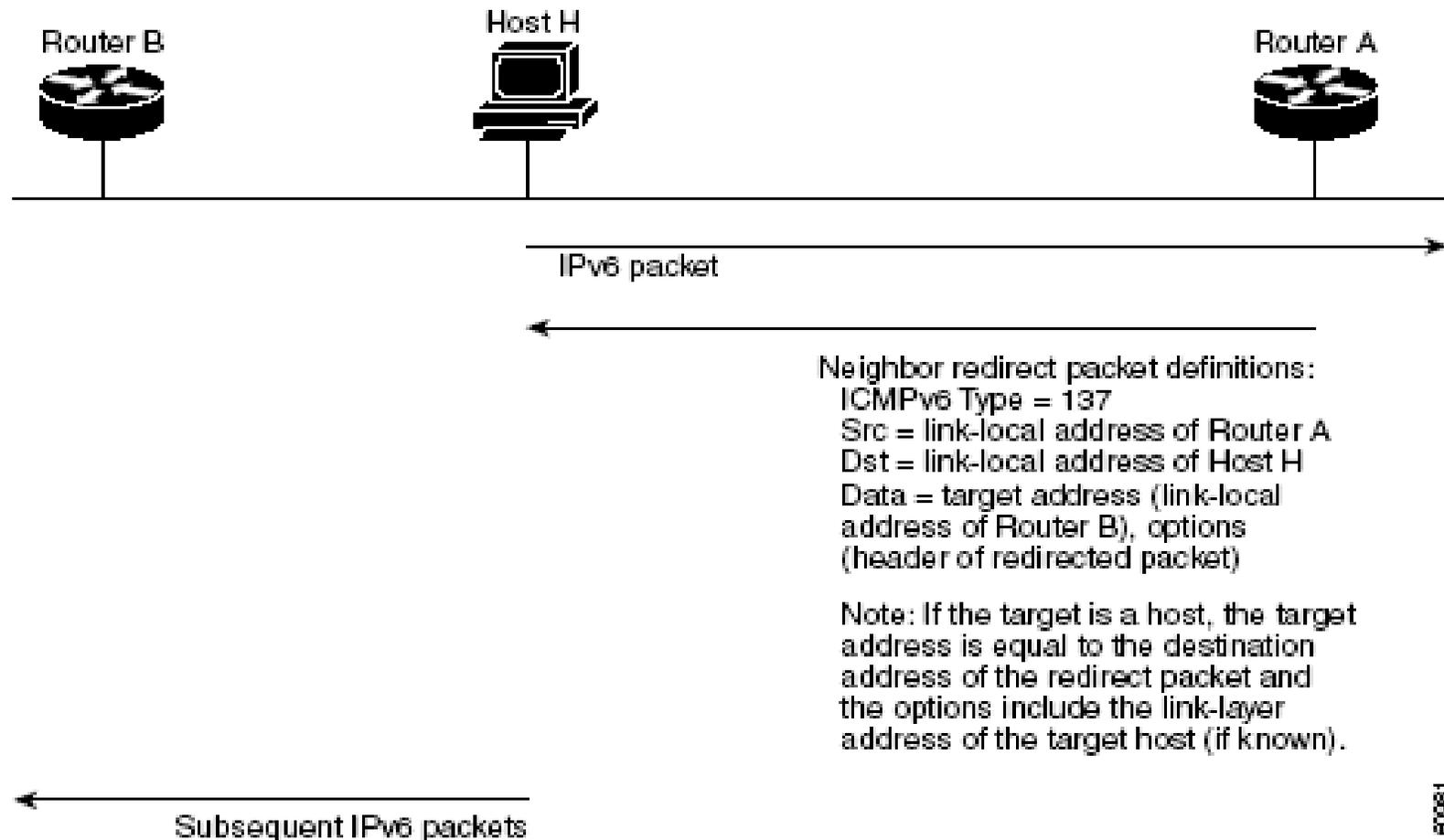
ICMPv6 Type = 134

Src = router link-local address

Dst = all-nodes multicast address

Data = options, prefix, lifetime, autoconfig flag

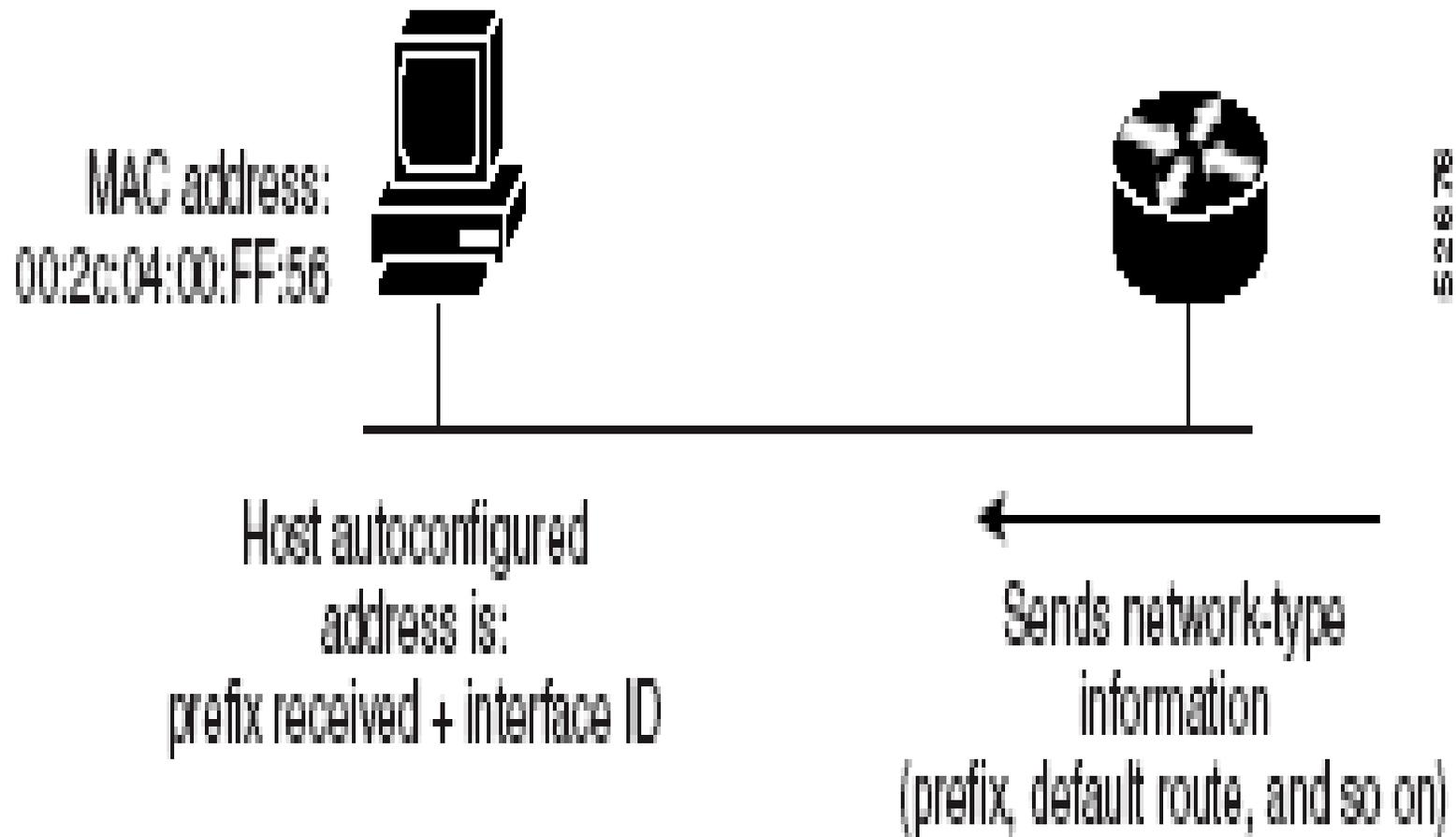
Neighbor Redirect



Definição de Endereços

- Consiste em enviar uma mensagem de Neighbor Solicitation e esperar por uma resposta.
 - Todos os nodos iniciam juntando o endereço multicast genérico e o nodo de endereço multicast solicitado
 - O endereço do nodo solicitado é feito pelo prefixo de 96 bits FF02:0:0:0:0:1 e colocado nos 32 bits mais baixos o endereço de destino.
 - Isto permite uma faixa de FF02:1:0:0 até FF02:1:FFFF:FFFF
 - O endereço destino completo está encaixado no pacote ICMP.

Autoconfiguração Stateless



Configuração de interface IPv6 em Windows XP

- Para instalar a pilha IPv6 no Windows XP/.NET (Windows 2003)
 - Durante o command prompt digitar `ipv6 install`
 - Assume configuração stateless por default
- As demais configurações podem ser feitas com o `netsh` (disponível a partir do Service Pack 1)

Windows XP - netsh

- Para configurar ou consultar dados referentes a interfaces IPv6, usa-se netsh ipv6 e os comandos:
 - 6to4 - Changes to the `netsh interface ipv6 6to4' context.
 - add - Adds a configuration entry to a table.
 - delete - Deletes a configuration entry from a table.
 - dump - Displays a configuration script.
 - help - Displays a list of commands.
 - install - Installs IPv6.
 - isatap - Changes to the `netsh interface ipv6 isatap' context.
 - renew - Restarts IPv6 interfaces.
 - reset - Resets IPv6 configuration state.
 - set - Sets configuration information.
 - show - Displays information.
 - uninstall - Uninstalls IPv6.

Windows XP - netsh

- Informações sobre o comando netsh interface ipv6 6to4
 - dump - Displays a configuration script.
 - reset state. - Resets all 6to4 configuration
 - set - Sets configuration information.
 - show - Displays information.

Windows XP - netsh

- Informações sobre o comando netsh interface ipv6 6to4 set
 - set interface - Sets 6to4 interface configuration information.
 - set relay - Sets 6to4 relay information.
 - set routing - Sets 6to4 routing information.
 - set state - Sets the 6to4 state.

Configuração de IPv6 em Linux

- Baseado na distribuição RedHat/Fedora
 - Usado também em outras (Mandrake, ...)
- Edita-se o arquivo /etc/network
 - Coloca-se a linha NETWORKING_IPV6=yes
- Reiniciam-se as configurações de rede
 - Executa-se o comando:
 - service network restart

Configuração de IPv6 em Linux

- A configuração padrão é buscar o endereço automaticamente via Neighbor Discovery
- Pode-se optar por configurar informações específicas sobre IPv6, editando-se o arquivo `/etc/sysconfig/networking/ifcfg-eth0`, no caso para a interface de rede eth0

Opções do arquivo

/etc/sysconfig/networking/ifcfg-eth0

- IPV6INIT=yes|no
 - Controla configuração desta interface (eth0)
- IPV6ADDR=<ipv6address>/<prefixlength>
 - Especifica o endereço IPv6 primário desta interface
- IPV6ADDR_SECONDARIES="<list of ipv6 addresses>"
 - (opcional) Especifica os endereços IPv6 secundários desta interface
- IPV6_ROUTER=yes|no
 - Controla a autoconfiguração IPv6, no caso de múltiplos routers (multi-homed)
- IPV6_AUTOCONF=yes|no
 - Controla a autoconfiguração
- IPV6_MTU=<MTU for IPv6>
 - (opcional) Especifica o MTU da interface (geralmente o MTU é especificado automaticamente)
- IPV6FORWARDING=yes|no
 - Repassa pacotes IPv6

Configuração de IPv6 em FreeBSD

- O FreeBSD já possui suporte nativo a IPv6
- Deve ser usado o arquivo `/etc/rc.conf` para a configuração
- A configuração é carregada quando o sistema é iniciado

Opções do arquivo /etc/rc.conf

- `ipv6_enable=<"YES|NO">`
 - Habilita pilha IPv6
- `ipv6_network_interfaces=<"interface">`
 - Habilita IPv6 na interface especificada. Se for colocado auto no lugar da interface, a interface será configurada automaticamente (autoconfiguração Serverless).
- `ipv6_gateway_enable=<"YES|NO">`
 - Habilita o forward de pacotes IPv6
- `ipv6_prefix_<interface>=<"prefixo">`
 - Mostra o prefixo a ser usado na formação do endereço IPv6. O sufixo é formado pelo endereço MAC.
- `ipv6_defaultrouter=<"endereço IPv6">`
 - Especifica o gateway default da rede IPv6
- `ipv6_router=<"caminho">`
 - Especifica o caminho para o daemon de roteamento (pode ser utilizado o daemon route6d)

Opções do arquivo /etc/rc.conf

- `ipv6_router_flags=<"flags">`
 - Flags a serem passados para o daemon de roteamento
- `prefixcmd_enable=<"YES|NO">`
 - Utiliza o prefixo especificado prefixo do roteador
- `rtadvd_enable=<"YES|NO">`
 - Habilita Router Adversiment
- `rtadvd_interfaces=<"interface">`
 - Define a interface por onde sairão os pacotes de Router Adversiment
- `mroute6d_enable=<"YES|NO">`
 - Habilita o roteamento multicast em IPv6
- `mroute6d_program=<"caminho">`
 - Caminho do daemon de roteamento IPv6. Geralmente é utilizado o daemon pim6d

Opções do arquivo /etc/rc.conf

- `mroute6d_flags=<"flags">`
 - Flags a serem passados para o daemon de roteamento.
- `gif_interfaces=<"lista de tuneis">`
 - Lista de túneis GIF. Te utilizado NO, desabilita túneis GIF.
- `gifconfig_<tunnel gif>=<"IPv4_1 IPv4_2">`
 - Configura tunel gif. IPv4_1 e IPv4_2 são as pontas.
- `stf_interface_ipv4addr=<"endereço IPv4">`
 - Endereço IPv4 para especificação de túnel do tipo 6to4
- `stf_interface_ipv4plen=<"0-31">`
 - Tamanho do prefixo a ser utilizado no túnel 6to4.
- `stf_interface_ipv6_ifid=<"endereço IPv6">`
 - Identificador da interface local em túneis do tipo 6to4. Pode ser utilizado o valor "auto", para configuração automática.

Opções do arquivo /etc/rc.conf

- `stf_interface_ipv6_slaid=<"Endereçamento SLA">`
 - Endereçamento SLA (Site Level Aggregator) a ser utilizado no túnel 6to4.
- `ipv6_ipv4mapping=<"YES|NO">`
 - Habilita mapeamento de endereços IPv4 dentro de endereços IPv6, como por exemplo, `::ffff:a.b.c.d`
- `ipv6_firewall_enable=<"YES|NO">`
 - Habilita firewall IPv6
- `ipv6_firewall_script=<"caminho">`
 - Caminho do script para firewall IPv6. Geralmente é utilizado `"/etc/rc.firewall6"`.
- `ipv6_firewall_type=<"tipo">`
 - Tipo de firewall IPv6
- `ipv6_firewall_quiet=<"YES|NO">`
 - Habilita o recurso de esconder as regras da firewall
- `ipv6_firewall_logging=<"YES|NO">`
 - Habilita o LOG da firewall IPv6.
- `ipv6_firewall_flags=<"flags">`
 - Envia flags para o daemon de firewall (ip6fw).

Configuração de interface IPv6 em roteadores Cisco

```
enable
configure terminal
interface ethernet 0/0
    ipv6 address 2001:0db8:0:1::/64 eui-64
exit
ipv6 unicast-routing
```

Interconexão IPv4/IPv6

- IPv6 sobre túnel IPv4 – o processo de examinar um datagrama IPv6 e colocá-lo um cabeçalho IPv4 para que ele possa transitar por roteadores IPv4
 - Túnel configurado – O endereço de “endpoint” do túnel é determinado pelo nodo que está encapsulando.
 - Túnel automático – O endereço de “endpoint” é determinado pelo endereço IPv4 que está dentro do pacote IPv6.

Interconexão IPv4/IPv6

- **Dual-stack**: Um nodo que roda tanto IPv4 quanto IPv6 em uma mesma interface.
- **NAT-PT**: Tradutor de protocolo que é colocado na frente de endereços que são somente IPv4 ou IPv6.
- **DSTM**: Dual Stack Transition Mechanism (draft, baseia-se em tunelamento/DHCPv6).
- **Proxy**: Recebe pedidos de endereços IPv4, retornando páginas IPv6 e vice-versa

Dual-Stack

- Necessária em qualquer sistema de interconexão entre os protocolos em alguma parte do backbone
- Estações que fazem a interconexão devem ter Dual Stack
- Baseia-se na idéia de a mesma interface de rede ter ambos os protocolos IPv4 e IPv6

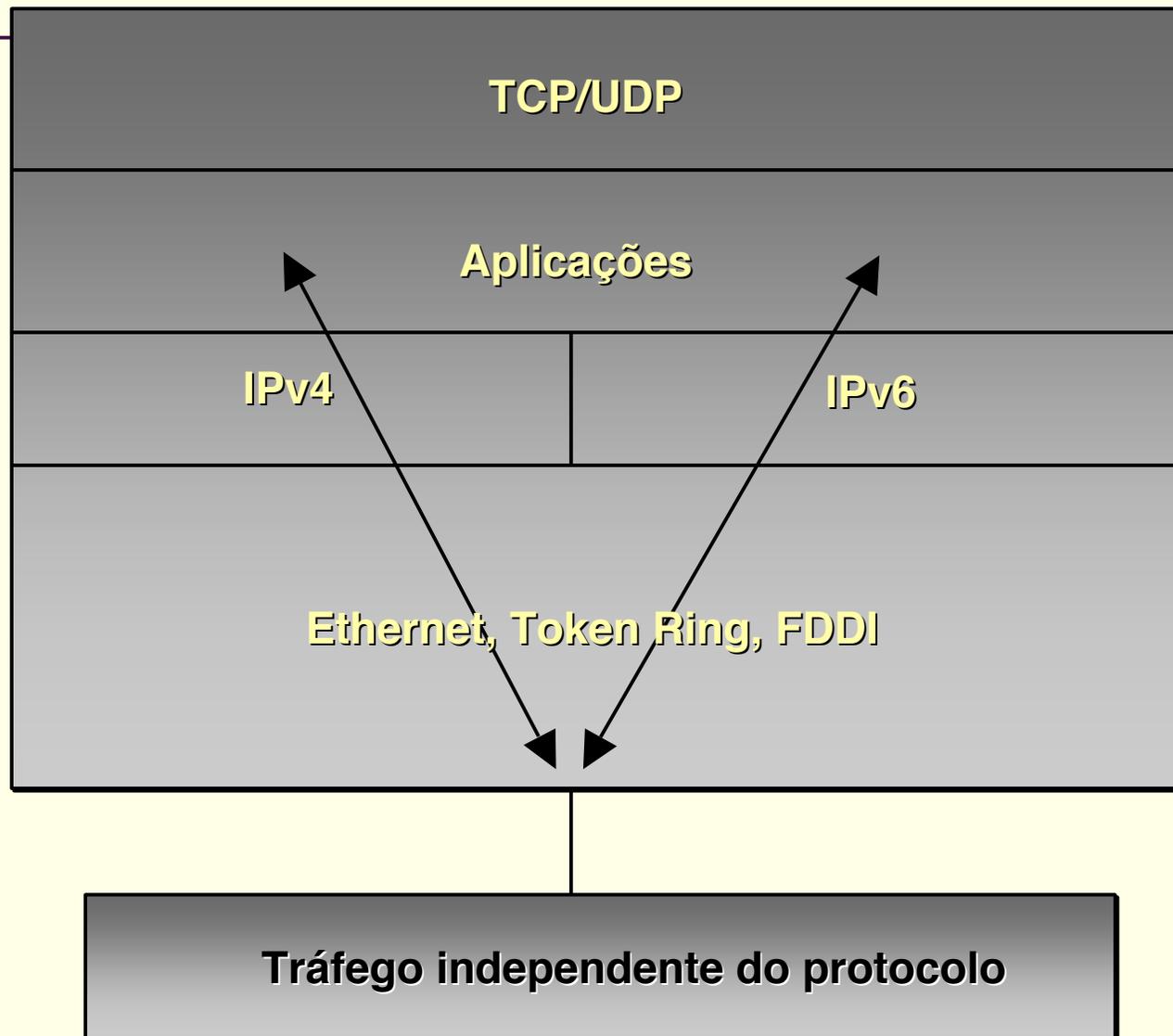
Dual-Stack

- Exemplo de Interface Dual-Stack em roteador Cisco
- Endereço IPv4 unicast:
200.132.0.17
- Endereço IPv6 unicast:
**2001:4A0:300:0:20
2:4AFF:FE64:8820**

```
FastEthernet1/0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::202:4AFF:FE64:8820
Global unicast address(es):
  2001:4A0:300:0:202:4AFF:FE64:8820, subnet is 2001:4A0:300::/64
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF64:8820
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

```
FastEthernet1/0/0 is up, line protocol is up
Hardware is cyBus FastEthernet Interface, address is 0002.4a64.8820 (bia 0002.4a64.8820)
Internet address is 200.132.0.17/25
MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
  reliability 255/255, txload 88/255, rxload 44/255
Encapsulation ARPA, loopback not set
```

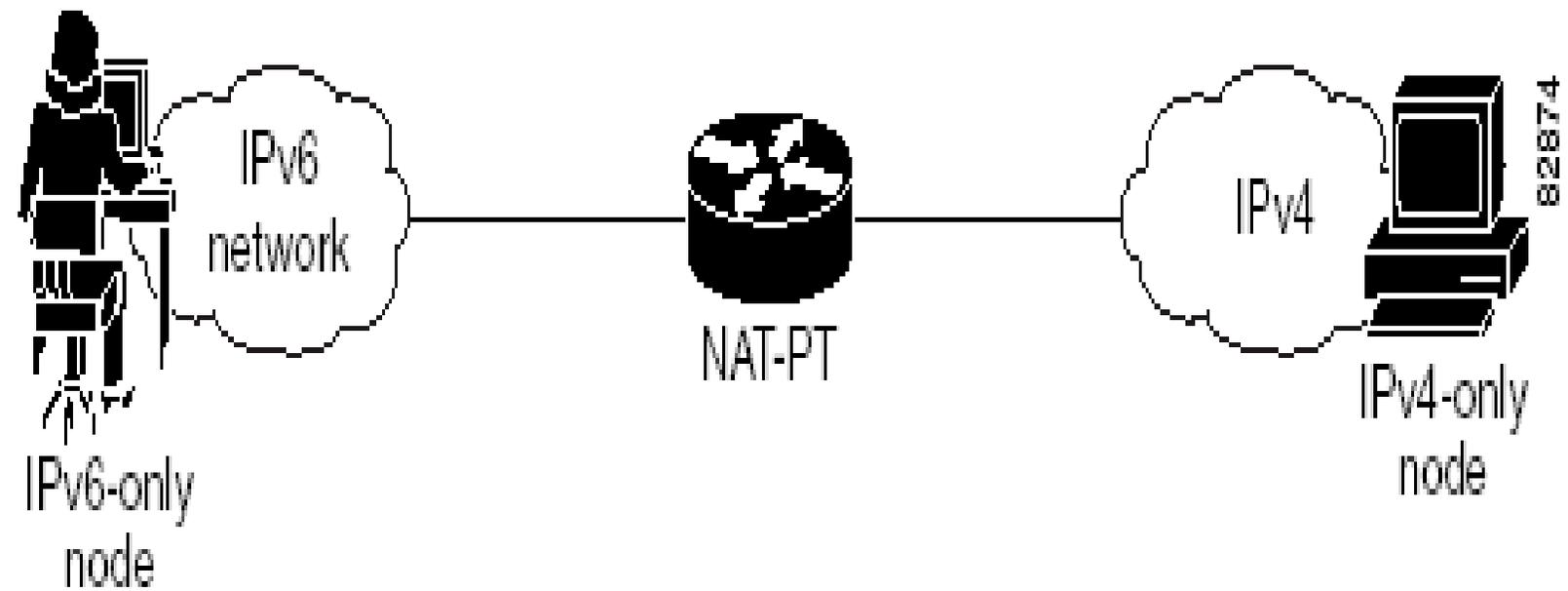
Dual-Stack



NAT-PT

- RFC
- Network Address Translation – Protocol Translation
- Utiliza faixas de endereço IPv4 para associação com nodos de endereço IPv6
- É baseado em um prefixo IPV6 escolhido para este fim
- Deve ser sempre utilizado em conjunto com algum DNS-ALG

NAT-PT



NAT-PT

- Pode ser:
 - Tradicional
 - Abertura de conexões de forma unidirecional, indo da rede IPv4 para a rede IPv6
 - Basic-NAT-PT
 - NAT-PT (PAT – Port Address Translation)
 - Bidirecional
 - Conexões abertas em ambos os sentidos

NAT-PT e FreeBSD

- Baseado do projeto KAME
- Melhor implementação por software existente
- Serve para demais distribuições BSD
- Dentro do arquivo de configuração do kernel deve-se colocar as opções
 - options INET6
 - options NATPT
- Após recompila-se o kernel

NAT-PT e FreeBSD

- A configuração do kernel é feita com o software natptconfig

mostra as traduções ativas

```
# natptconfig show xlate
```

mostra as regras correntes

```
# natptconfig show rules
```

habilita o mapeamento

```
# natptconfig map enable
```

Desabilita o mapeamento

```
natptconfig map disable
```

NAT-PT e roteadores CISCO

- Iniciou no IOS 12.2(13)T e se firmou no IOS 12.3(2)T
- Permite NAT-PT dinâmica e estática
- Segue as RFCs 2765 e 2766

NAT-PT e roteadores CISCO

- Exemplo de configuração

```
enable
```

```
configure terminal
```

```
ipv6 nat prefix 2003:0db8::/96
```

```
interface ethernet 3/1
```

```
    ipv6 address 2003:0db8:yyyy::9/64
```

```
    ipv6 nat
```

```
    exit
```

```
interface ethernet 3/3
```

```
    ip address 192.168.30.9 255.255.255.0
```

```
    ipv6 nat
```

```
    exit
```

DNS-ALG

- Papel crucial na idéia de trabalhar com NAT-PT
- Funciona na camada de aplicação (Application Level Gateway)
- RFC 2766
- Cada cada consulta de DNS dá uma resposta diferente, dependendo do protocolo que fez a consulta (IPv4 ou IPv6)

POP-RS – Rede IPv6

- **Blocos IPv6 reservado ao POP-RS**
 - 2001:4A0:300::/48 – Bloco IPv6 de produção do POP-RS
 - 2001: 4A0:300::/64 – Bloco SLA utilizado na rede local do POP-RS
 - 2001: 4A0:300:1::/64 – Bloco SLA utilizado na rede piloto, em estações apenas com IPv6
 - 2001: 4A0:300:2::/64 – Bloco SLA utilizado na rede piloto para o mapeamento interno de IPv6 para IPv4
- **Bloco IPv4 reservado para fazer NAT-PT**
 - 200.132.15.0/24 – Rede IPv4 para o mapeamento de IPv4 para IPv6

POP-RS – Configuração do roteador IPv6 de borda

```
ipv6 unicast-routing
!  
interface FastEthernet1/0/0  
  ipv6 address 2001:4A0:300::/64 eui-64  
!  
interface ATM4/0/0.103 point-to-point  
  ipv6 address 2001:4A0:0:1::D9/126  
  ipv6 rip PILOTO-IPv6 enable  
  ipv6 rip PILOTO-IPv6 summary-address 2001:4A0:300::/48  
!  
interface ATM4/0/0.111 point-to-point  
  ipv6 address 2001:4A0:0:1::D5/126  
  ipv6 rip PILOTO-IPv6 enable  
  ipv6 rip PILOTO-IPv6 summary-address 2001:4A0:300::/48  
!  
ipv6 route 2001:4A0:300:1::/64 2001:4A0:300:0:210:B5FF:FE54:A5C  
!
```

POP-RS – Configuração do NAT-PT (KAME)

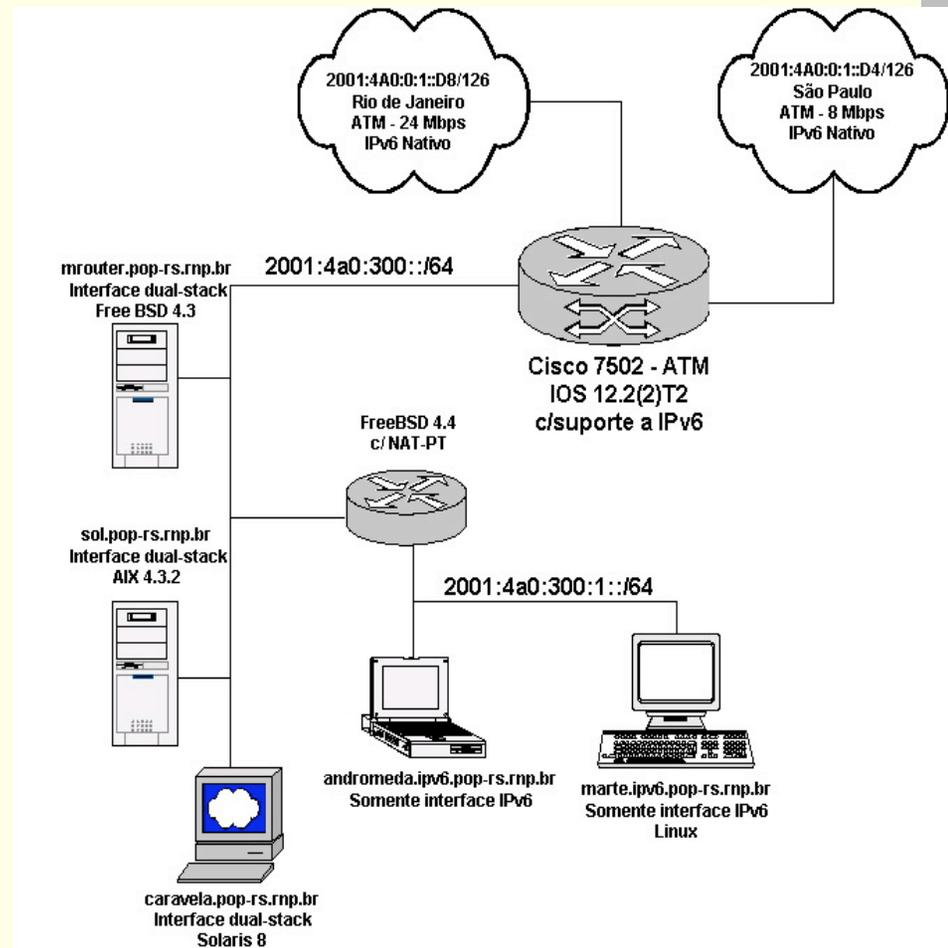
prefix 2001:4a0:300:2::

**map from daddr 200.132.15.2 to daddr
2001:4a0:300:1:250:8bff:fec0:8b2d**

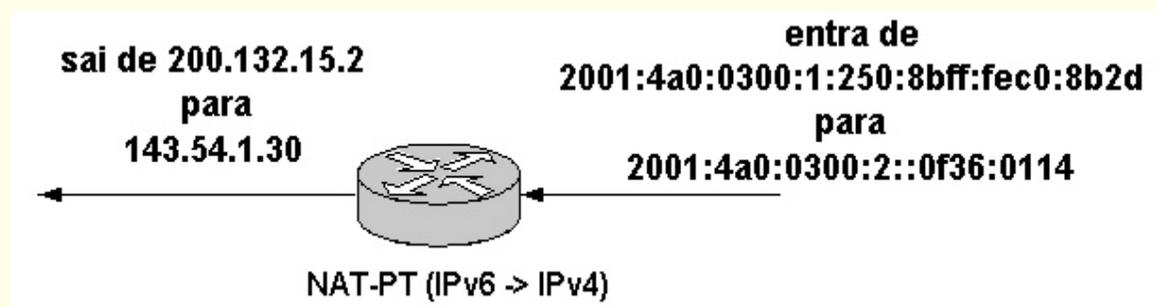
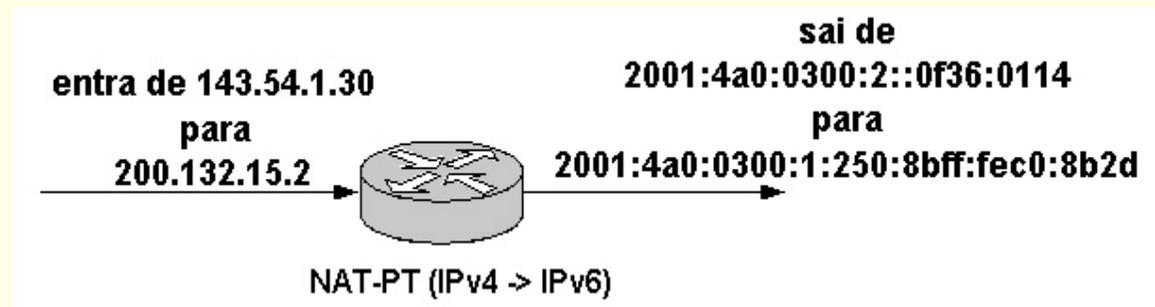
**map from 2001:4a0:300:1:250:8bff:fec0:8b2d to
200.132.15.2**

map enable

POP-RS – Esboço IPv6



POP-RS – Tráfego NAT-PT



TOTd

- Aplicação que funciona como DNS-ALG
- Serve apenas de middleware entre a aplicação e um DNS válido
- Desenvolvido pelo projeto Vermicelli (<http://www.vermicelli.pasta.cs.uit.no/ipv6/software.html>)

TOTd

- Exemplo de configuração

```
forwarder 200.132.0.132 port 53
```

```
prefix 2001:4a0:300:2::
```

```
retry 300
```

Filtros de pacotes IPv6

- Não existem muitos filtros para IPv6 aplicados hoje
 - A maior preocupação é com o “colocar para funcionar” e não se aplica os filtros devidos
- Existe a idéia que pouca gente usa IPv6 e não existe muitos problemas com segurança
- A maioria das firewalls comerciais não tem suporte a IPv6
- Atualmente vem aumentando o número de filtros para IPv6

Filtros no Windows XP/.NET

- Necessita do Advanced Networking Pack for Windows XP
- Filtros feitos usando o software netsh
- Exemplo:
 - netsh firewall> set adapter Local Area Connection filtering=enable
 - netsh firewall> set adapter Local Area Connection port 80=enable protocol=tcp

Filtros no Linux

- Utiliza Iptables-IPv6 para escrita das regras no kernel
- Pode ser instalado através de pacotes (RPMS, apt-get, ...)
- Para carregar o módulo de firewall IPv6
 - modprobe ip6_tables
- Exemplo de regra:
 - `ip6tables -A INPUT -i sit+ -p tcp -s 3ffe:ffff:100::1/128 --sport 512:65535 --dport 22 -j ACCEPT`

HTTP - Apache

- Versão 1.3.x necessita de patch para trabalhar com IPv6
- Versão 2 tem suporte nativo a IPv6
- Possibilidade de VirtualHosts IPv4 e IPv6

HTTP – Apache VirtualHosts (httpd.conf)

- Definição de IPs a serem usados
 - NameVirtualHost 200.132.0.15
 - NameVirtualHost 200.19.246.22
 - NameVirtualHost
[2001:4a0:300:0:2a0:24ff:feba:f1e2]
- Definição das portas
 - Listen 200.132.0.15:80
 - Listen 200.19.246.22:80
 - Listen [2001:4a0:300:0:2a0:24ff:feba:f1e2]:80

HTTP – Apache VirtualHosts (httpd.conf)

■ Definição do VirtualHost IPv6

```
<VirtualHost [2001:4a0:300:0:2a0:24ff:feba:f1e2]>  
  ServerAdmin suporte@pop-rs.rnp.br  
  DocumentRoot /home/www/ipv6_ipv6  
  ServerName ipv6.pop-rs.rnp.br  
  ErrorLog logs/ipv6_ipv6-error_log  
  CustomLog logs/ipv6_ipv6-access_log common  
</VirtualHost>
```

HTTPD – formato de logs do Apache

- Logs ficam no formato IPv6

```
2001:4a0:300:0:d12:6681:3be:899f - -  
[26/Nov/2003:16:02:09 -0200] "GET /2.rnp.gif  
HTTP/1.1" 304 0
```

```
2001:4a0:300:0:d12:6681:3be:899f - -  
[26/Nov/2003:16:02:09 -0200] "GET /m-3.gif  
HTTP/1.1" 304 0
```

```
2001:4a0:300:0:d12:6681:3be:899f - -  
[26/Nov/2003:16:02:09 -0200] "GET /bul2.gif  
HTTP/1.1" 304 0
```

HTTPD – formato de logs do Apache

- No apache 1.3.x, mesmo quando a interface acessada é IPv4, o endereço no log fica no tipo IPv4-compatible IPv6 address
::200.132.0.118 - - [26/Nov/2003:16:02:09 -0200]
"GET /bul2.gif HTTP/1.1" 304 0
- Pode gerar problemas com analisadores de logs, como programas para estatísticas
- Este problema não ocorre com o Apache 2

DNS – Windows .NET

The screenshot shows the DNS Management console window titled "dnsmgmt - [DNS\LUA\Forward Lookup Zones\ad.tche.br]". The left pane displays a tree view of the DNS hierarchy, with "ad.tche.br" selected under "Forward Lookup Zones". The right pane shows a list of 12 records for the "ad.tche.br" zone.

Name	Type	Data
_msdcs		
_sites		
_tcp		
_udp		
DomainDnsZones		
ForestDnsZones		
(same as parent folder)	Start of Authority (SOA)	[151], lua.ad.tche.br., hostmaster.
(same as parent folder)	Name Server (NS)	lua.ad.tche.br.
(same as parent folder)	Host (A)	200.132.0.109
lua	Host (A)	200.132.0.109
marTE	Host (A)	200.132.0.118
marTE	IPv6 Host (AAAA)	2001:04a0:0300:0000:0202:55ff:fe5d:1f11

DNS - BIND

- Utilizado para resolução de nomes e DNS reverso
- Suporte a IPv6 desde a versão 8
- Suporte endereços DNS A6 e AAAA
- Suporte a DNS reverso INT e ARPA

DNS - BIND

- Configuração do named.conf

```
#  
# Configuração de DNS para o domínio  
# ipv6.pop-rs.rnp.br  
#  
zone "ipv6.pop-rs.rnp.br" {  
    type master;  
    allow-transfer {  
        200.132.0.0/24;  
        2001:4a0:300::/64;  
    };  
    also-notify {  
        200.132.0.105;  
    };  
    file "zone.ipv6.pop-rs.rnp";  
    allow-query { any; };  
};
```

DNS - BIND

- Configuração do named.conf

```
#  
# Configuração do DNS Reverso  
# baseado em ip6.arpa  
# faixa da zona: 2001:4a0:300::/48  
#  
zone "[x200104A00300/48].ip6.arpa" {  
    type master;  
    allow-transfer {  
        200.132.0.0/24;  
        2001:4a0:300::/64;  
    };  
    also-notify {  
        200.132.0.105;  
    };  
    file "zone.ipv6.pop-rs.rnp_rev_arpa";  
    allow-query { any; };  
};
```

DNS - BIND

- Configuração do named.conf

```
#  
# Configuração do DNS Reverso  
# baseado em ip6.int (usado ainda muito atualmente (20/02/2002))  
# porém tende a cair em desuso  
# Usado para manter compatibilidade com aplicações mais antigas  
# faixa: 2001:4a0:300::/48  
#  
zone "0.0.3.0.0.a.4.0.1.0.0.2.ip6.int" {  
    type master;  
    allow-transfer {  
        200.132.0.0/24;  
        2001:4a0:300::/64;  
    };  
    also-notify {  
        200.132.0.105;  
    };  
    file "zone.ipv6.pop-rs.rnp_rev_int";  
    allow-query { any; };  
};
```

DNS - BIND

- Arquivo zone.ipv6.pop-rs.rnp

```
...
$ORIGIN ipv6.pop-rs.rnp.br.
caravela      AAAA  2001:4a0:300:0:204:acff:fe66:212d
              A6   64  ::204:acff:fe66:212d redelocal
equitext      CNAME lupus
lepus         A    200.132.0.103
              AAAA  2001:4a0:300:0:201:3ff:fedd:2a9f
              A6   64  ::201:3ff:fedd:2a9f redelocal
pampa         A    200.19.246.22
              AAAA  2001:4a0:300:0:2a0:24ff:feba:f1e2
              A6   64  ::2a0:24ff:feba:f1e2 redelocal
lupus         AAAA  2001:4a0:300:0:2d0:9ff:fea6:230e
              A6   64  ::2d0:9ff:fea6:230e redelocal
marte        AAAA  2001:4a0:300:0:202:55ff:fe5d:1f11
              A6   64  ::202:55ff:fe5d:1f11 redelocal
...
```

DNS - BIND

- Arquivo zone.ipv6.pop-rs.rnp_rev_arpa

```
...
\[x200104A00300/48].ip6.arpa IN SOA lepus.pop-rs.rnp.br.
    root.lepus.pop-rs.rnp.br. (
        2003112401 ; serial
        10800     ; refresh (3 hours)
        3600     ; retry (1 hour)
        604800  ; expire (1 week)
        60      ; minimum (1 minute)
    )
    NS    lepus.pop-rs.rnp.br.
    NS    lupus.pop-rs.rnp.br.
...
$ORIGIN \[x200104A003000000/64].ip6.arpa.
\[x020255FFFE5D0B13/64] PTR    lua.ipv6.pop-rs.rnp.br.
\[x02a024fffebaf1e2/64] PTR    pampa.ipv6.pop-rs.rnp.br.
\[x020255FFFE5D1F11/64] PTR    marte.ipv6.pop-rs.rnp.br.
...
```

DNS - BIND

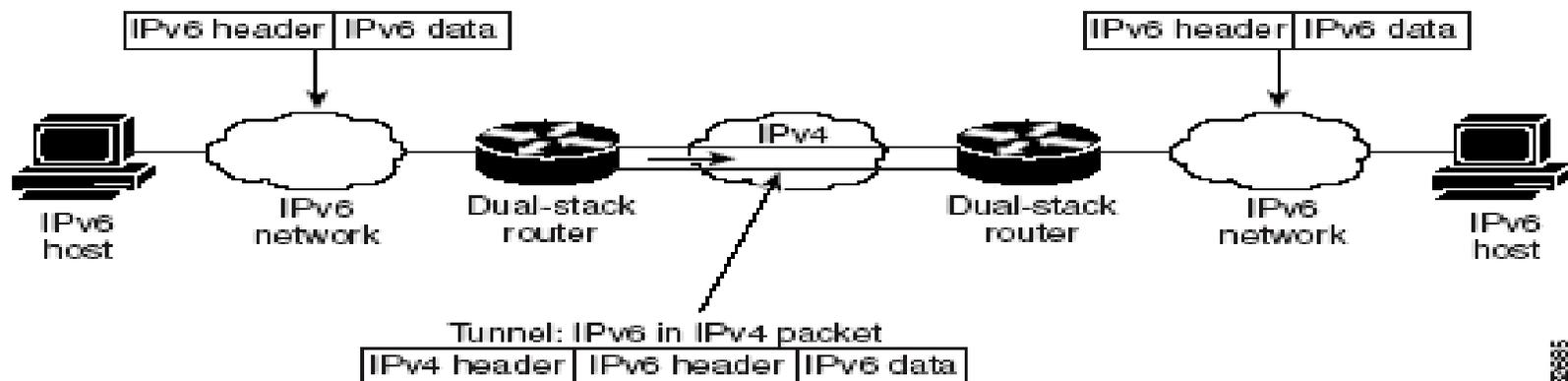
- Arquivo zone.ipv6.pop-rs.rnp_rev_int

```
...
0.0.3.0.0.a.4.0.1.0.0.2.ip6.int IN SOA lepus.pop-rs.rnp.br. root.lepus.pop-
rs.rnp.br. (
                2003112401 ; serial
                10800      ; refresh (3 hours)
                3600       ; retry (1 hour)
                604800     ; expire (1 week)
                60         ; minimum (1 minute)
                )
                NS      lepus.pop-rs.rnp.br.
                NS      lupus.pop-rs.rnp.br.
...
■ $ORIGIN 0.0.0.0.0.3.0.0.a.4.0.1.0.0.2.ip6.int.
■ 1.1.f.1.d.5.e.f.f.f.5.5.2.0.2.0 PTR marte.ipv6.pop-rs.rnp.br.
■ 2.e.1.f.a.b.e.f.f.f.4.2.0.a.2.0 PTR pampa.ipv6.pop-rs.rnp.br.
■ 3.1.b.0.d.5.e.f.f.f.5.5.2.0.2.0 PTR lua.ipv6.pop-rs.rnp.br.
...
```

Tunelamento

- Baseia-se na idéia de encapsular tráfego IPv6 dentro de tráfego IPv4, fazendo a junção de nuvens IPv6 a Internet IPv6

Figure 34 Overlay Tunnels



Tunelamento

- Podem ser:
 - Tunelamento IPv6 manualmente configurado
 - Tunelamento GRE (IPv6 over IPv4)
 - Em desuso
 - Tunelamento automático 6to4
 - Tunelamento automático IPv4-compatível
 - Em desuso
 - TSP
 - Usado pelo freenet6

Tunelamento IPv6 manualmente configurado

- Túneis também conhecidos como estáticos
- Manualmente configurados pelos administradores, possuindo endereços IPv4 e IPv6 fixos.
- Suportado pela maioria dos sistemas operacionais
- Um dos primeiro tipo de túneis IPv6 existentes e é comumente usado
- Point-to-point

Tunelamento IPv6 manualmente configurado

- Exemplo com Roteadores Cisco

```
enable
```

```
configure terminal
```

```
interface tunnel 0
```

```
    ipv6 address 2001:2b00:0027:1::3/127
```

```
    tunnel source ethernet 0
```

```
    tunnel destination 192.168.30.1
```

```
    tunnel mode ipv6ip
```

```
exit
```

Tunelamento IPv6 manualmente configurado

- Exemplo com Linux

```
/sbin/ip tunnel add sit1 mode sit ttl 4 remote  
192.168.1.2
```

```
/bin/ip link set dev sit1 up
```

```
/sbin/ip -6 route add 2001:2b00:27:2::/64 dev sit1  
metric 1
```

Tunelamento GRE (IPv6 over IPv4)

- O padrão GRE pode encapsular qualquer protocolo dentro de IPv4
- Deve ser feito de forma manual a configuração
- O protocolo GRE funciona como o carregador do IPv6 e o IPv4 funciona como o protocolo de transporte
- Muito pouco usado atualmente

Tunelamento automático 6to4

- Descrito na RFC 3056
- Permite que domínios IPv6 isolados se comuniquem com outros domínios IPv6 através da rede IPv4, no modo **point-to-multipoint**
- Roteadores não são configurados em pares porque usam a estrutura IPv4 como um NBMA (NonBroadcast MultiAccess link)
- O endereço IPv4 embutido no IPv6 é usado para encontrar a outra ponta do túnel automático
- Deve ser configurado em um roteador de borda da rede IPv6 isolada

Tunelamento automático 6to4

- É usado um relay router com endereço IPv4 anycast (192.88.99.1)
- Usa o prefixo global 2002:wwxx:yyzz::/48
 - wwxyyzz = hexadecimal do IPv4 válido
- O endereço IPv6 completo fica
 - 2002:wwxx:yyzz:[SLA ID]:[Interface ID]

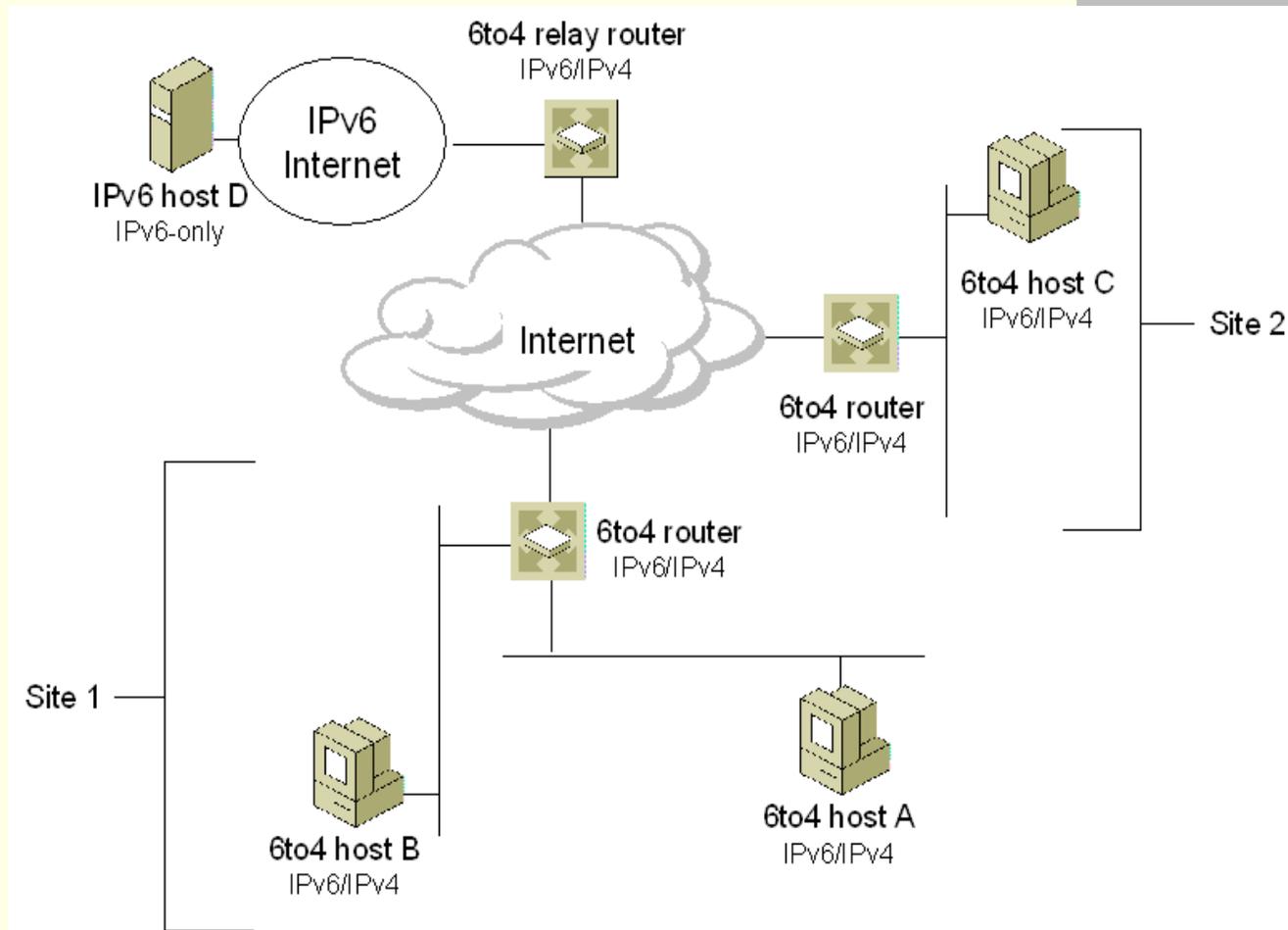
Tunelamento automático 6to4

- Dividido nos componentes:
 - 6to4 host
 - Host configurado com pelo menos um endereço 6to4. Não requer configuração manual
 - 6to4 router
 - Roteador com suporte a tunelamento 6to4, encarregado de rotear os pacotes. Requer configuração manual.
 - 6to4 relay router
 - Faz a ligação do tráfego 6to4 para a Internet IPv6.

Tunelamento automático 6to4

- O roteador 6to4 possui uma rota 2002::/16 para rotear para outras redes 6to4 e uma rota default (:::/0) para rotear pelo router relay 6to4

Tunelamento automático 6to4



Tunelamento automático 6to4

- Exemplo com Roteadores Cisco

```
enable
```

```
configure terminal
```

```
interface tunnel 0
```

```
    ipv6 address 2000:c0a8:6301::1/64
```

```
    tunnel source ethernet 0
```

```
    tunnel mode ipv6ip 6to4
```

```
exit
```

```
ipv6 route 2002::/16 tunnel 0
```

Tunelamento automático 6to4

- Exemplo com Linux (“braçalmente” !!!)

```
/sbin/ip tunnel add tun6to4 mode sit ttl 5 remote  
any local 200.132.0.15
```

```
/sbin/ip link set dev tun6to4 up
```

```
/sbin/ip -6 addr add 2002::/16 dev tun6to4
```

```
/sbin/ip -6 route add 2000::/3 via ::192.88.99.1 dev  
tun6to4 metric 1
```

Tunelamento automático 6to4

- Exemplo com Linux (editando o arquivo `/etc/sysconfig/networking/ifcfg-eth[0..x]`, onde x é a interface ethernet)
 - `IPV6TO4INIT=yes|no`
 - Controla configuração de túneis do tipo 6to4
 - `IPV6TO4_RELAY=<ipv4address|ipv6to4address>`
 - Endereço IPv4 ou IPv6to4 da outra ponta do túnel
 - `IPV6TO4_IPV4ADDR=<ipv6address>`
 - Sobrescreve o endereço IPv4 local na formação do endereço IPv6
 - `IPV6TO4_ROUTING="eth0-:f101::0/64 eth1-:f102::0/64"`
 - Informação sobre a configuração de sobredes locais
 - `IPV6TO4_CONTROL_RADVD=yes|no`
 - Controla anúncios de Router Adversiment (opcional)
 - `IPV6TO4_RADVD_PIDFILE=file`
 - Arquivo com o número do identificador de processo do daemon radvd (responsável pelo Router Adversiment)1

Tunelamento automático IPv4- compatível

- Usado no início com IPv6. Hoje está em desuso
- É composto pelos 96 bits iniciais zerados, mais os 32 bits finais compostos pelo endereço IP do host
- Pode ser configurado entre roteadores de borda e entre um roteador de borda e um host.

Freenet6

- Usa o protocolo tsp
 - Configura automaticamente túneis usando comandos do sistema operacional
 - Draft proposta
 - Usa as informações contidas em arquivo de configuração (tspc.conf) e um programa chamado tspc[.exe]
 - Funciona com vários sistemas operacionais
 - Possui templates para Windows, Linux e outros sistemas operacionais

Arquivo tspc.conf

```
tsp_version=1.0.1  
tsp_dir=.  
client_v4=auto|1.1.1.1  
userid=usuario1  
passwd=senha1  
template=WindowsNT-2K  
retry_delay=0  
if_tunnel=PPP
```

Problemas na implementação de IPv6 no POP-RS

- Problemas com interconexão entre prefixos 2002,2001 e 3FFE
- Problemas com documentação de servidores IPv6, especialmente VirtualHosts (Apache)
- Problemas com documentação de sistemas operacionais, como AIX 3.2
 - AIX 3.2 não tinha uma autoconfiguração eficiente

Dúvidas, questões, discussões, ...

E-mail para contato (com SpamAssassin na frente ;-)

fabricao@gtrh.tche.br