

IPv6 e IPv4 em ambiente retrô com DSLAM

XI BSDDAY - UFFRJ
Mauro Paes Corrêa



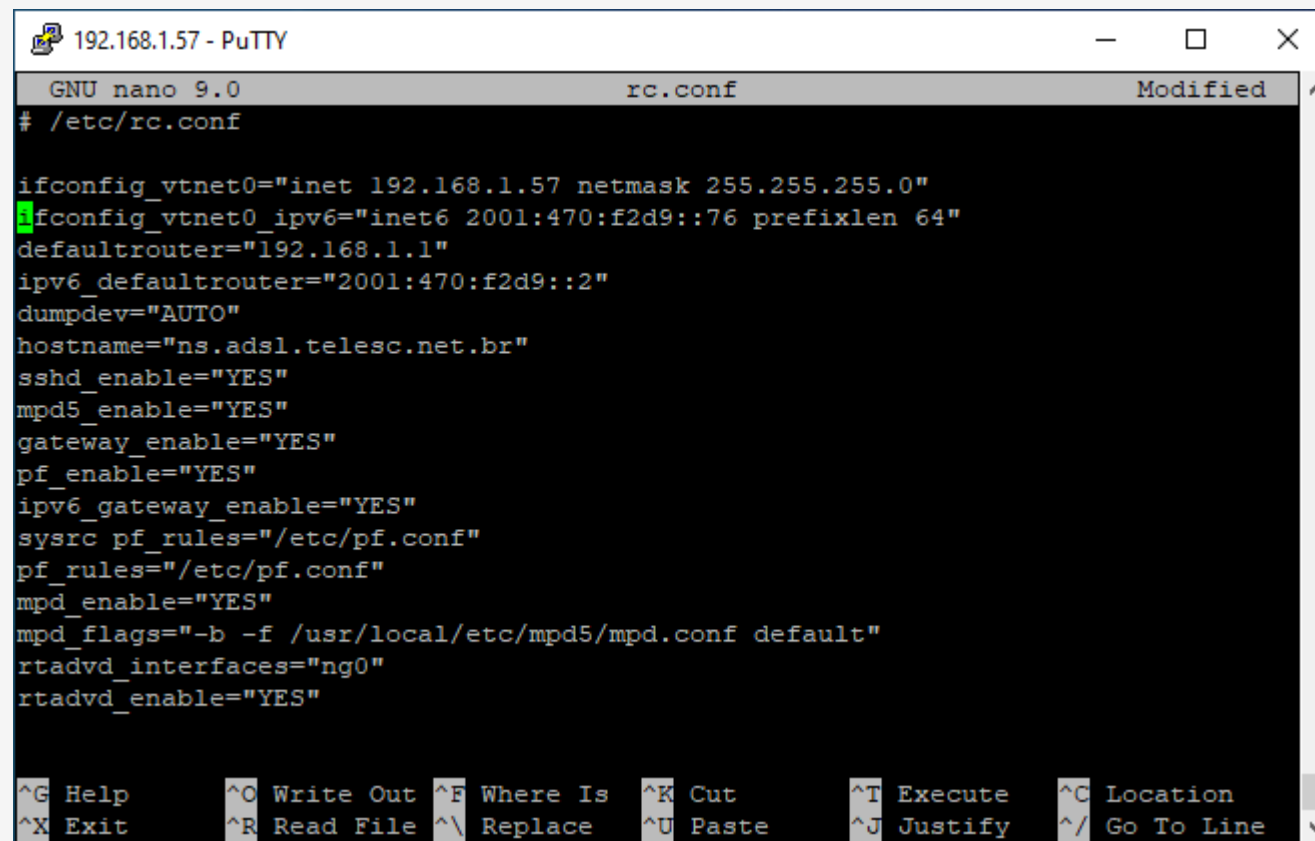
Como é a implementação do mpd no FreeBSD?

A implementação do IPv6 no FreeBSD, é fácil de ser realizada, mas é preciso ter os seguintes conhecimentos:

- O básico de Ipv4 e Ipv6, para compreender as diferenças entre os protocolos e a forma de trabalho dentro do mpd
- Estudar qual cenário será utilizado (bridge, ou router)
- E qual gateway será usado

E principalmente!!!!

- **Garantir um /etc/rc.conf bem configurado (e o sysctl.conf)**



```
192.168.1.57 - PuTTY
GNU nano 9.0 rc.conf Modified
# /etc/rc.conf

ifconfig_vtnet0="inet 192.168.1.57 netmask 255.255.255.0"
ifconfig_vtnet0_ipv6="inet6 2001:470:f2d9::76 prefixlen 64"
defaultrouter="192.168.1.1"
ipv6_defaultrouter="2001:470:f2d9::2"
dumpdev="AUTO"
hostname="ns.adsl.telesc.net.br"
sshd_enable="YES"
mpd5_enable="YES"
gateway_enable="YES"
pf_enable="YES"
ipv6_gateway_enable="YES"
sysrc pf_rules="/etc/pf.conf"
pf_rules="/etc/pf.conf"
mpd_enable="YES"
mpd_flags="-b -f /usr/local/etc/mpd5/mpd.conf default"
rtadvd_interfaces="ng0"
rtadvd_enable="YES"

^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

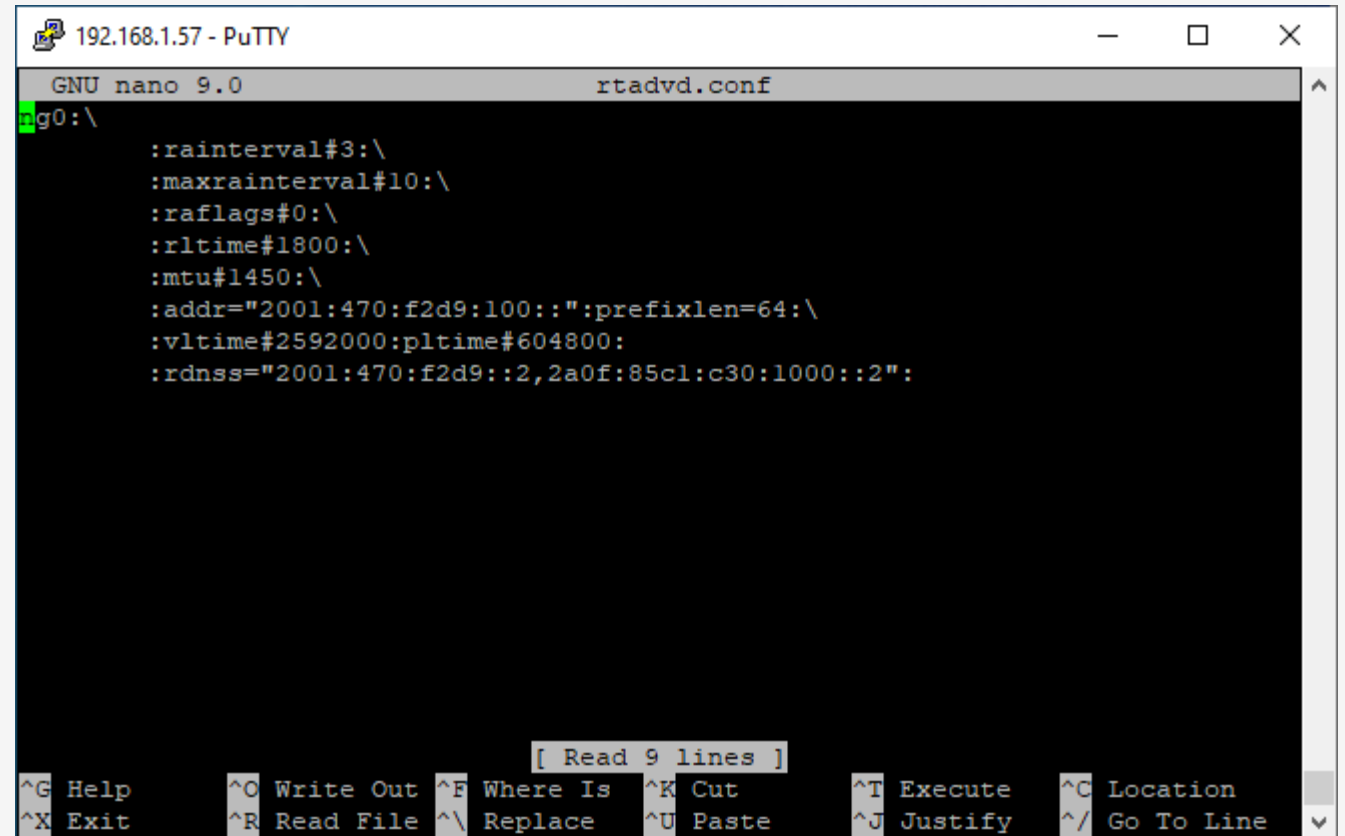


Além do mpd, é preciso de algo a mais?

Sim, precisamos anunciar o Ipv6, tanto a rede que o modem ADSL vai receber, como de que a vizinhança "saiba" que a rede existe e é comunicável.

Fazemos isso instalando o rtdvd.

Na sequência, configuramos o firewall pf, e o mpd, garantindo então a configuração do /etc/pf.conf, dos arquivos em /usr/local/etc/mpd5 e do arquivo único do rtdvd, o rtdvd.conf, garantindo a plena automação do processo.



```
192.168.1.57 - PuTTY
GNU nano 9.0 rtdvd.conf
g0:\
:rainterval#3:\
:maxrainterval#10:\
:raflags#0:\
:rtime#1800:\
:mtu#1450:\
:addr="2001:470:f2d9:100::":prefixlen=64:\
:vlttime#2592000:pltime#604800:
:rdns="2001:470:f2d9::2,2a0f:85c1:c30:1000::2":

[ Read 9 lines ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```



Bom, se o rtadvd anuncia as redes, o que faz o mpd?

O mpd, na sua versão 5.9, faz várias funções, dentre as principais:

- Garante a oferta do PPPoE na rede, “jogando” a sinalização na interface do servidor;
- Autenticar o usuário;
- Disponibilizar tanto o IPv4 quanto o IPv6
- Pode permitir a bilhetagem do usuário, das mais diversas formas.

```
192.168.1.57 - PuTTY
GNU nano 9.0 mpd.conf
Startup:
# Usuário de gerência do console do MPD5
set user admin mauro admin
# Configurações de console e interface web
set console self 127.0.0.1 5005
set console open
set web self 0.0.0.0 5006
set web open

default:
load pppoe_server

pppoe_server:
# 1. CONTEXTO DO BUNDLE (Camada 3)
create bundle template B
set bundle maximum 1
# [ip-pool] Seus IPs IPv4 do Accel-PPP (Gateway 10.1.50.1 e range para
set ipcp ranges 10.1.50.1/32 10.1.50.10/32
set ipcp dns 192.168.1.1 8.8.8.8

^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```



E como ele “entende” o roteamento de forma automática?

Quando o usuário se loga, o mpd dispara um script para “garantir” a rota do usuário dentro do túnel PPPoE e ao mesmo tempo o servidor já está preparado para fazer o NAT das conexões em IPv4. Assim, temos IPs reais em IPv6 e um NAT em Ipv4, entregando o que chamamos de dual-stack.

Como estamos usando uma conexão ADSL, diminuimos o MTU e MRU para 1450, garantindo melhor a entrega de pacotes, em todo o processo.

```
192.168.1.57 - PuTTY
GNU nano 9.0          ipv6-up.sh
^G Actually assign the IPv6 prefix
^X if [ "$PROTO" == "inet6" ] || [ -z "$PROTO" ]
^O then
^R     sleep 2
^F
^E     # 1. Atribui o IP de transporte do FreeBSD na interface ngX
^C     ifconfig $INTERFACE inet6 $SUBNET::1 prefixlen 64
^_
^/     # O PULO DO GATO: Desativa a recepção de RA na ngX para matar o conflito
^T     sysctl net.inet6.ip6.ifsubmit=1 2>/dev/null
^I     ifconfig $INTERFACE inet6 -accept_rtadv
^L
^K     # 2. Escreve o arquivo de configuração do rtadvd com a flag mestre
^H     printf "%s:\\\\n\t:rainterval#3:\\\\n\t:maxrainterval#10:\\\\n\t:raflags#0
^N
^M     # 3. Mata resíduos antigos
^B     if [ -f /var/ppp-v6/$INTERFACE.pid ]; then
^J         kill `cat /var/ppp-v6/$INTERFACE.pid` 2>/dev/null
^U     fi
^_
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^/ Go To Line
```



Vendo na prática a conexão

Quando o usuário e senha são conferidos com sucesso, a interface ng0, utilizada pelo mpd, começa a transmitir informações. Na imagem, resolvemos “brincar” um pouco mais, reduzindo o mtu para 1280, se comportando muito bem.

Observe que temos o inet com o IPv4 e o inet6, com o endereçamento IPv6

Dessa forma, com a interface levantada é possível utilizar regras de firewall, de acordo com a necessidade da estrutura.

```
192.168.1.57 - PuTTY
options=ec079b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,TSO4,TSO6,LRO,VLAN_HWTSO,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6,HWSTATS>
ether bc:24:11:a3:9e:89
inet 192.168.1.57 netmask 0xffffffff broadcast 192.168.1.255
inet6 fe80::be24:11ff:fea3:9e89%vtnet0 prefixlen 64 scopeid 0x1
inet6 2001:470:f2d9::76 prefixlen 64
media: Ethernet autoselect (10Gbase-T <full-duplex>)
status: active
nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
lo0: flags=1008049<UP,LOOPBACK,RUNNING,MULTICAST,LOWER_UP> metric 0 mtu 16384
options=680003<RXCSUM,TXCSUM,LINKSTATE,RXCSUM_IPV6,TXCSUM_IPV6>
inet 127.0.0.1 netmask 0xff000000
inet6 ::1 prefixlen 128
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
groups: lo
nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
ng0: flags=10088d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu 1280
options=0
inet 10.1.50.1 --> 10.1.50.10 netmask 0xffffffff
inet6 fe80::be24:11ff:fea3:9e89%ng0 prefixlen 64 scopeid 0x3
inet6 2001:470:f2d9:100::1 prefixlen 64
nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
root@ns:/usr/local/etc/mpd5 #
```



Testando na prática - DSLAM

No nosso ambiente, temos a seguinte estrutura:

Um DSLAM AN2000B, que foi muito popular por ofertar ADSL E ADSL2, chegando a velocidades de até 24 mega de download e 1,5 de upload.

Curiosidade: o controle de velocidade era baseado em “profiles” no DSLAM

Este equipamento funciona nativamente com PPPoE, com uma saída de 10/100 como uplink. Não há manual, e tudo foi configurado via telnet.



O modem ADSL

Modelo de testes – TP-Link TD-VG5611

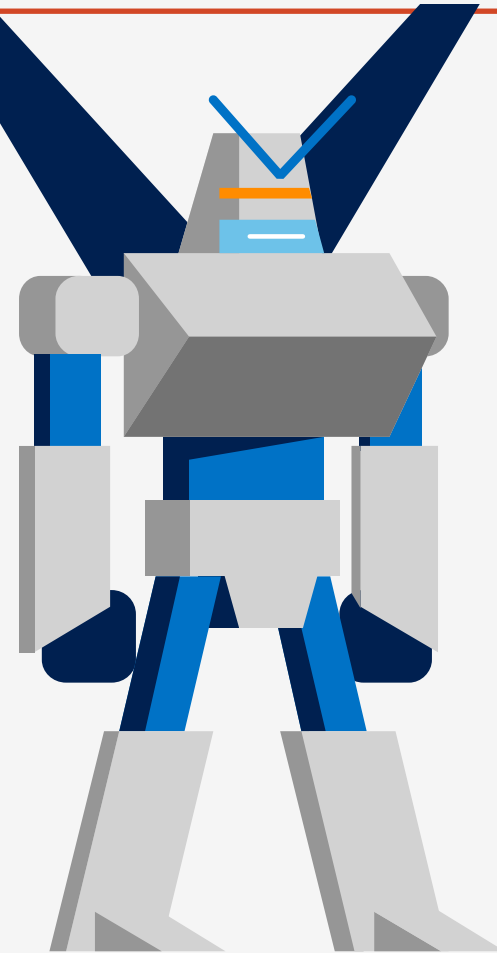
- 1 Escolhido este modelo pelo fato de suportar IPv4, IPv6 e ter Wifi, além de uma porta Voip
- 2 Sua configuração é fácil, e tem várias opções de configuração, suportando todas as tecnologias de ADSL do DSLAM UTSTARCOM AN2000B
- 3 Usamos um modelo que foi popular na Algar Telecom, no início da descontinuidade da ADSL.



Vamos ver tudo em ação?

Agora vamos mostrar o usuário conectado, os prints do modem ADSL e o tracert em IPv4, IPv6 assim como os pings, além de testes de validação em sites famosos, como o ip6.test.

BORA FREEBSD!



Configurando o modem

No mpd foi configurado um usuário e senha, e o VPI/VCI precisam seguir o padrão designado para esta rede, conforme tela ao lado, habilitando o SLAAC para IPv6.



Internet Connection Type: PPPoE

Username: mauro

Password:

Confirm Password:

Connection Mode: Auto On Demand Manually

Authentication Type: Auto


IPv4: Enable

Default Gateway: Current Connection

IPv6: Enable

Addressing Type: SLAAC

IPv6 Default Gateway: Current Connection



Vendo se deu tudo certo...

O usuário e senha precisam ser validados pelo mpd e podemos acompanhar pela console ou log dele.

Quando ele valida, recebemos o IPv4 e IPv6.

Daí para frente, toda a parte de roteamento é ativada, permitindo a navegação

```
192.168.1.57 - PuTTY
[vtnet0-2] LCP: state change Ack-Sent --> Opened
[vtnet0-2] LCP: auth: peer wants nothing, I want CHAP
[vtnet0-2] CHAP: sending CHALLENGE #1 len: 41
[vtnet0-2] LCP: LayerUp
[vtnet0-2] CHAP: rec'd RESPONSE #1 len: 26
[vtnet0-2]   Name: "mauro"
[vtnet0-2] AUTH: Trying INTERNAL
[vtnet0-2] AUTH: INTERNAL returned: undefined
[vtnet0-2] CHAP: Auth return status: undefined
[vtnet0-2] CHAP: Response is valid
[vtnet0-2] CHAP: Reply message: Welcome
[vtnet0-2] CHAP: sending SUCCESS #1 len: 11
[vtnet0-2] LCP: authorization successful
[vtnet0-2] Link: Matched action 'bundle "B" ""'
[vtnet0-2] Creating new bundle using template "B".
[B-1] Bundle: Interface ng0 created
[vtnet0-2] Link: Join bundle "B-1"
[B-1] Bundle: Status update: up 1 link, total bandwidth 64000 bps
[B-1] IPCP: Open event
[B-1] IPCP: state change Initial --> Starting
[B-1] IPCP: LayerStart
[B-1] IPV6CP: Open event
[B-1] IPV6CP: state change Initial --> Starting
[B-1] IPV6CP: LayerStart
```



Testando a conectividade

Escolhemos um perfil (profile) no DSLAM de 6 Mega por 512k de upload, simulando um ambiente “empresarial”, com baixa latência, como você verá agora. Os testes foram feitos com IPv4 e IPv6, sendo observada ampla estabilidade e velocidade nos dois protocolos.



Tracert e ping em Ipv6

```
C:\Windows\System32>ping www.gnx.com.br
```

```
Disparando www.gnx.com.br [2001:470:f2d9::7] com 32 bytes de dados:
```

```
Resposta de 2001:470:f2d9::7: tempo=6ms
```

```
Resposta de 2001:470:f2d9::7: tempo=6ms
```

```
Resposta de 2001:470:f2d9::7: tempo=5ms
```

```
Resposta de 2001:470:f2d9::7: tempo=5ms
```

```
Estatísticas do Ping para 2001:470:f2d9::7: Administrador: cmd
```

```
Pacotes: Enviados = 4, F
```

```
perda),
```

```
Aproximar um número redondo
```

```
C:\Windows\System32>tracert www.gnx.com.br
```

```
Rastreando a rota para www.gnx.com.br [2001:470:f2d9::7]
```

```
com no máximo 30 saltos:
```

1	<1 ms	<1 ms	<1 ms	2001:470:f2d9:1001:76da:88ff:fe69:c48c
2	6 ms	5 ms	6 ms	dslflex.telesc.net.br [2001:470:f2d9:100::1]
3	6 ms	6 ms	6 ms	sc-108.gnx.com.br [2001:470:f2d9::7]



Tracert e ping em ipv4

```
C:\Windows\System32>ping -4 www.gnx.com.br
```

```
Disparando www.gnx.com.br [138.117.192.72] com 32 bytes de dados:
```

```
Resposta de 138.117.192.72: bytes=32 tempo=6ms TTL=60
```

```
Resposta de 138.117.192.72: bytes=32 tempo=6ms TTL=60
```

```
Resposta de 138.117.192.72: bytes=32 tempo=6ms TTL=60
```

```
Resposta de 138.117.192.72: bytes=32 tempo=6ms TTL=60
```

```
C:\Windows\System32>tracert -4 www.gnx.com.br
```

```
Estadísticas do Ping para 1
```

```
Pacotes: Enviados = 4,  
perda),
```

```
Aproximar um número redondo
```

```
Mínimo = 6ms, Máximo =
```

```
Rastreando a rota para www.gnx.com.br [138.117.192.72]  
com no máximo 30 saltos:
```

1	<1 ms	<1 ms	<1 ms	192.168.34.1
2	5 ms	5 ms	5 ms	corelan0-dsl.telesc.net.br [10.1.50.1]
3	6 ms	5 ms	5 ms	pfSense.gnx.com.br [192.168.1.1]
4	7 ms	6 ms	6 ms	leo.gnx.com.br [138.117.192.72]
5	7 ms	6 ms	6 ms	leo.gnx.com.br [138.117.192.72]



Teste de velocidade

SHARE

Result ID 19251887709 RESULTS SETTINGS

DOWNLOAD Mbps **6.28** **UPLOAD** Mbps **0.38**

Ping ms 477 — 368

GO

Connections
Multi

Tac Telecom
Jaguaruna
[Change Server](#)

Ateky
138.117.192.72

HOW LIKELY IS IT THAT YOU WOULD RECOMMEND ATEKY TO A FRIEND OR COLLEAGUE?

0 1 2 3 4 5 6 7 8 9 10

Not at all likely Extremely Likely



Visualizando os ips...



ip6.biz

IPv4 connectivity

IPv4	Supported	?
Address	138.117.192.72	?
Hostname	leo.gnx.com.br	?
ISP	AS262316 Ateky Internet Eireli Me	?

IPv6 connectivity

IPv6	Supported	?
ICMP	Reachable (336 ms)	?
Address	2001:470:f2d9:1001:b926:2dab:b2f7:ac17	?
Hostname	Test failed	?
ISP	AS6939 Hurricane Electric LLC	?

Usamos tudo para facilitar a sua experiência

Para simular o ambiente em Ipv6, usamos um tunnelbroker da Hurricane Electric, que é gratuito e com ampla documentação para novos usuários, inclusive o FreeBSD, e o mpd pode ser usado para permitir conexão PPPoE direto em sua rede, onde não há oferta de conectividade em DHCP e DHCPv6, sendo o PPPoE a fonte de conectividade. É um ambiente que roda em ethernet, rádio, fibra...

Mesmo utilizando hardware legado como um DSLAM, a experiência pode (e deve ser reproduzível, sempre).



Qual foi o legado da experiência?

Poder reaproveitar hardware obsoleto, pois conexões via cobre com ADSL podem cobrir distância de até 5KM e para usos pontuais, é uma solução a ser considerada, ainda que possa ser polêmica para alguns especialistas.

E principalmente, a resiliência do mpd, que demonstra a importância do mpd no suporte a diferentes aplicações de rede, garantindo conectividade e estabilidade de forma simples, com ampla documentação.



E por incrível que pareça....

Testamos também em ambiente dial-up, mas utilizando o ppp, provendo com o FreeBSD a pilha dual-stack, e mesmo com tantas vantagens, ele é mais lento, principalmente pelo overhead de cabeçalho e dos mecanismos de transição. O IPv4 possui cabeçalhos fixos de 20 bytes, o IPv6, de 40. Em banda larga, isso é insignificante, mas em conexões estreitas, é um desafio.



Referências Bibliográficas

Handbook FreeBSD – <https://freebsd.org>

Mpd – <https://mpd.sourceforge.net/doc5/mpd.html>

BSDSul – <https://bsdsul.com.br>

Rtadvd- <https://man.freebsd.org/cgi/man.cgi?rtadvd>

NIC.Br – <https://nic.br>



Muito obrigado !

Mauro Paes Corrêa

Fundador do BSDSul
<https://bsdsul.com.br>

CEO – GNX
<https://gnx.com.br>

Linkedin – <https://linkedin.com/in/mauopaescorrea>
E-Mail: mauro@gnx.com.br

