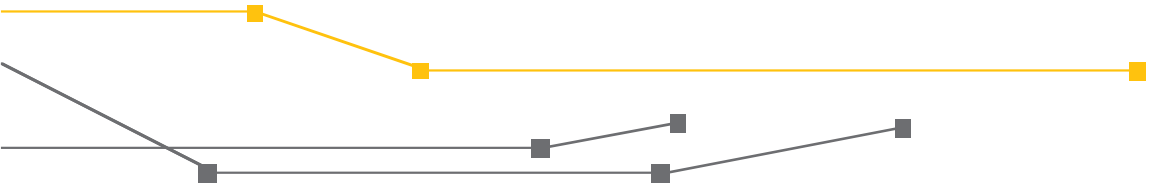
Técnicas de transição IPv4/IPv6
NAT64

RNP
Ministério da Cultura
Ministério da Saúde
Ministério da Educação
Ministério da Ciência, Tecnologia e Inovação
GOVERNO FEDERAL BRASIL PÁTRIA EDUCADORA



Versão	Data	Responsável	Modificação
1.0	18/10/2016	Guilherme B. Ladvocat	Versão inicial
1.1	20/10/2016	Ari Frazão	Revisão do texto
1.2	28/12/2016	Guilherme B. Ladvocat	Inclusão do tópico “2.2. Solução *open source* de uma rede wireless puramente IPv6 com NAT64”

1. Introdução

A escassez de endereços IPv4, juntamente com a crescente demanda de endereços para conectar usuários e máquinas, têm trazido empecilhos para a operação e a expansão de infraestruturas de redes de todos os portes.

Implantar puramente o protocolo IPv6 em paralelo com o IPv4 (*dual-stack*) não garante a continuidade de expansão, visto que o número de endereços IPv4 disponíveis é infinitamente menor se comparado com a oferta do IPv6. Neste sentido, uma rede puramente IPv6 comunicando-se com redes locais e globais neste mesmo protocolo seria o melhor dos cenários. Porém, até que isto se torne realidade, um *middleware* deve intermediar a comunicação entre os protocolos.

A técnica de transição IPv4/IPv6 adotada neste projeto foi o NAT64, definida pela RFC 6146. Neste projeto, a Gerência de Operações da RNP testou e homologou a tecnologia em um ambiente de testes provido pela Universidade Federal de Pernambuco. Um segundo laboratório foi montado no âmbito de criar uma solução *open source* desta arquitetura, possibilitando uma implementação sem custos com software. Uma série de testes de compatibilidade com esta tecnologia foram realizados, utilizando os sistemas operacionais mais comuns no mercado.

No fim, tem-se que esta técnica de transição mostrou-se funcional para uma gama de aplicações, podendo, porém, apresentar limitações quando o protocolo IPv6 não é suportado nativamente em camadas superiores à de rede.

2. NAT64

A técnica de transição NAT64, que utiliza unicamente endereços IPv6 nos hosts clientes e possibilita a comunicação com destinos em IPv6 e IPv4, tem-se mostrado bastante popular em listas de discussões e RFCs. Um sub-protocolo crucial neste processo é o DNS64, que integra a solução, possibilitando a descoberta de endereços remotos IPv4, manipulando-os num formato em que o protocolo IPv6 entenda.

O exemplo abaixo ilustra o processo.

Inicialmente, quando um host configurado apenas com endereço IPv6 deseja comunicar-se com um servidor que só suporta o protocolo IPv4, esta comunicação é iniciada por meio de dois principais processos, sendo o primeiro uma resolução DNS64, e o segundo uma tradução de cabeçalhos IP pelo *gateway* NAT64.

Na figura 2.1, onde este processo é ilustrado, o host envia um pedido de resolução sobre domínio h2.example.com ao servidor DNS64. O servidor, por sua vez, envia a resposta com o endereço IPv4 resolvido, mas de forma convertida para IPv6 (64:ff9b::c00:201). Desta maneira, o host cliente entende, de forma transparente, que irá se comunicar com um servidor que possui endereçamento IPv6, encaminhando os pacotes ao *gateway* que está configurado para suportar NAT64. Este *gateway* entende que, quando

um destino cujos 32 bits iniciais possui os caracteres ‘64:ff9b’, terá, em seus 32 bits finais, o endereço IPv4 convertido em hexadecimal. Assim, o *gateway* converte o cabeçalho IPv6 para IPv4 com base no endereço destino original transformando-o em um destino IPv4 (hexadecimal para decimal) e a origem é dada da mesma forma como no NAT44, com um IPv4 particular a múltiplas conexões.

Desta forma, o *gateway* irá funcionar como intermediador entre os dois protocolos.

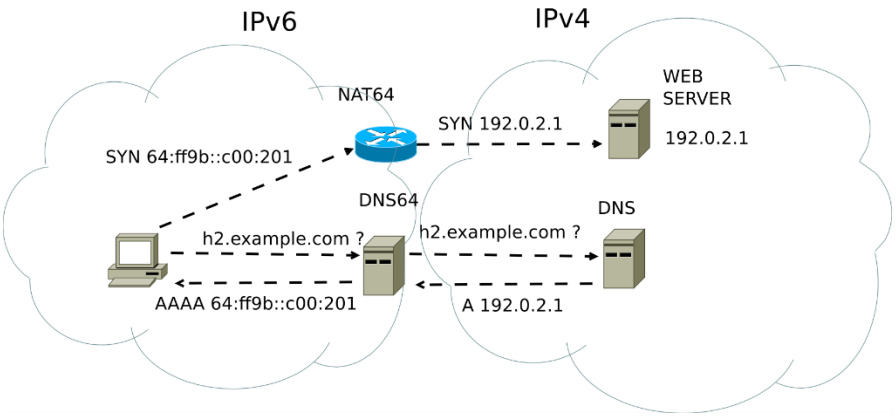


Figura 2.1 – Exemplo genérico sobre o funcionamento do NAT64.

Para que este processo fique claro, o seguinte teste é apresentado, conforme figura 2.2.

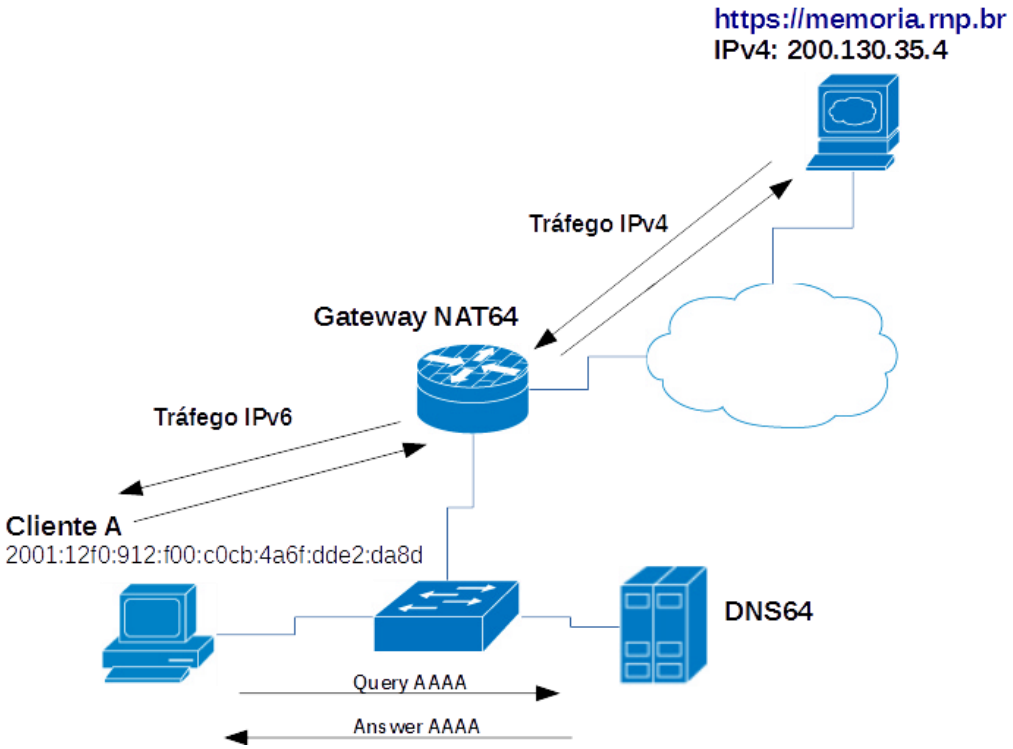


Figura 2.2 – Topologia do laboratório com uma rede LAN puramente IPv6.

1- O cliente A, cujo endereço IPv6 é o 2001:12f0:912:f00:c0cb:4a6f:dde2:da8d, deseja se comunicar com o website https://memoria.rnp.br, cujo endereço IPv4 é o 200.130.35.4.
2- O processo de resolução se inicia quando o cliente A envia uma solicitação ao servidor DNS64, demandando o endereço IP do referido website, ilustrado na captura da figura 2.3.
3- O servidor DNS64 o responde, conforme figura 2.4, com o endereço 64:ff9b::c882:2304. Note que os 32 bits finais desta resposta correspondem ao endereço do website (200.130.35.4) no formato hexadecimal.
4- O cliente A, então, envia pacotes ao seu *default gateway*, para que este faça a tradução entre os protocolos, conforme figura 2.5.
5- O *gateway* faz a tradução IPv6/IPv4 no sentido cliente -> servidor e a tradução IPv4/IPv6 no sentido servidor -> cliente.
6- Conforme ilustrado na figura 2.6, o cliente A recebe a resposta após esta tradução de forma transparente, como se tivesse, de fato, feito uma comunicação nativa com um destino IPv6.

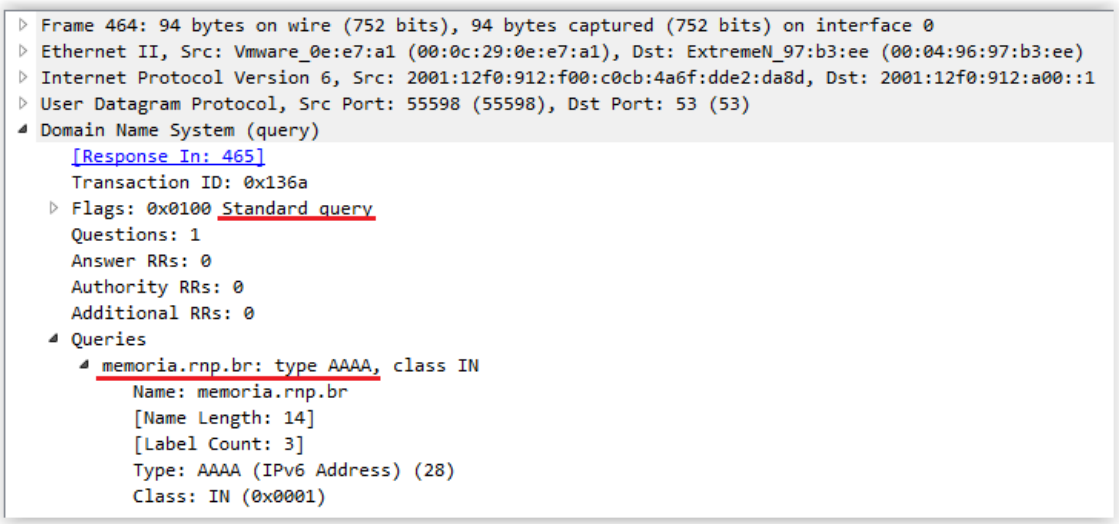


Figura 2.3 – Captura de *query* DNS AAAA ao servidor DNS64.

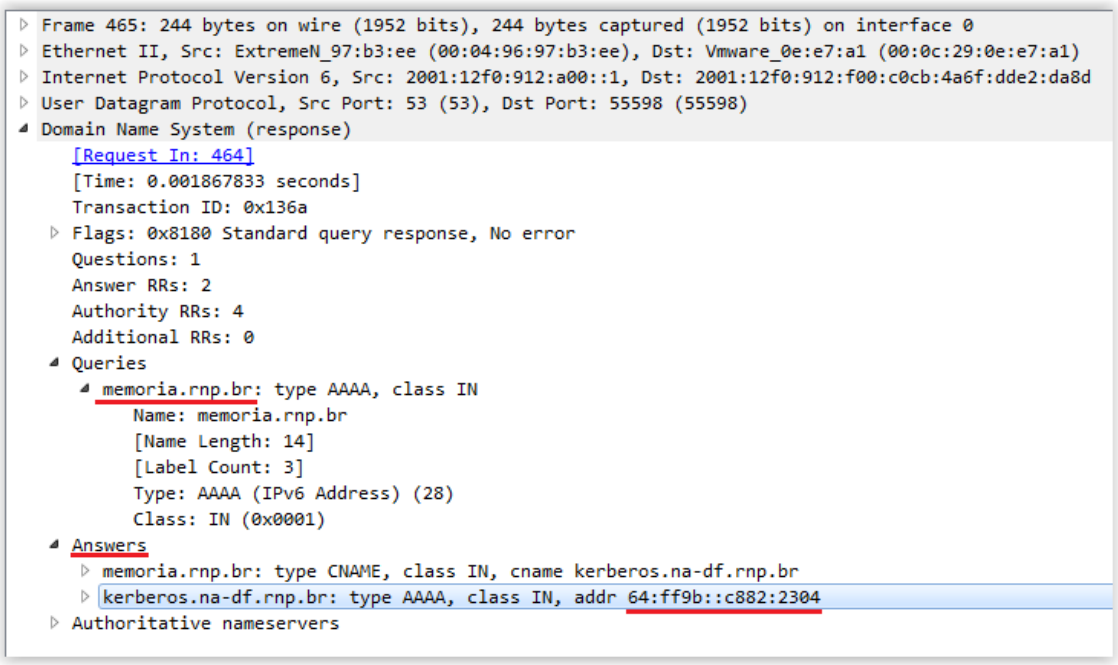


Figura 2.4 – Captura da resposta do servidor DNS64 com o endereço IPv4 convertido para IPv6.

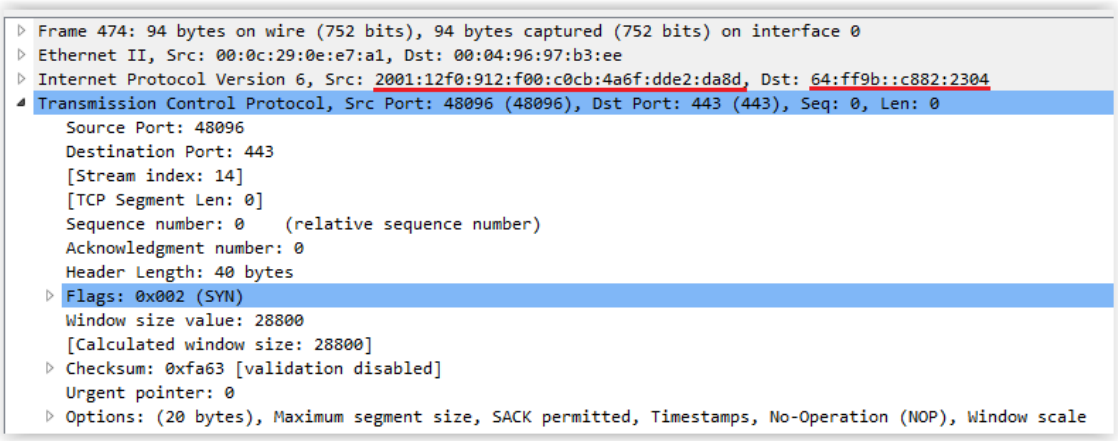


Figura 2.5 – Captura do primeiro pacote de dados enviado ao servidor web com endereço destino resolvido pelo servidor DNS64.

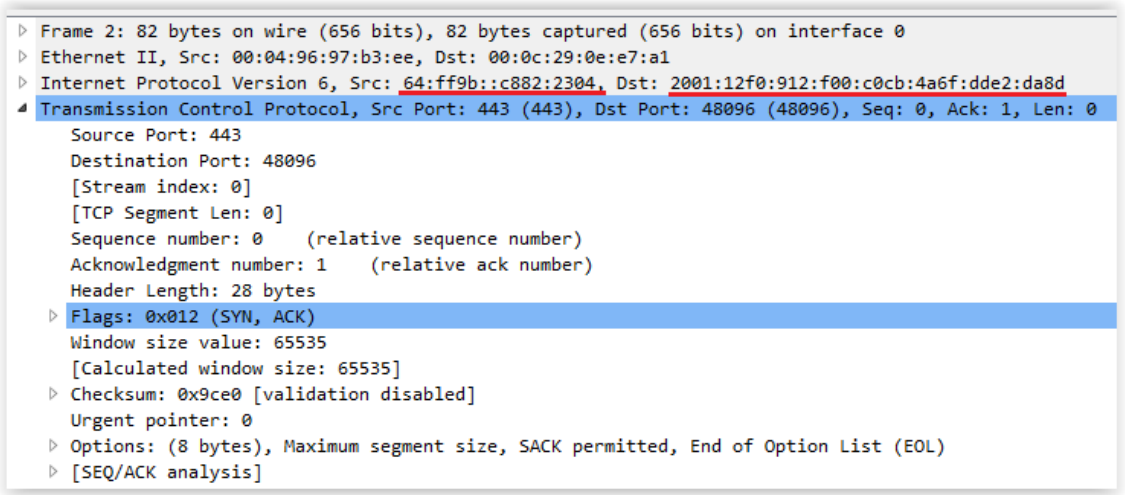


Figura 2.6 – Resposta do servidor web com cabeçalhos IP convertidos.

2.1. Laboratório de NAT64 da Universidade Federal de Pernambuco

No âmbito deste projeto, o Núcleo de Tecnologia da Informação da Universidade Federal de Pernambuco implementou um laboratório com a técnica de transição NAT64. A topologia construída é a mesma da Figura 2.2.

As configurações necessárias para o funcionamento desta técnica de transição são: *Gateway* NAT64, DNS64 e conectividade IPv6 entre os equipamentos nesta mesma LAN. No laboratório, o Firewall Palo Alto PA-5050 foi utilizado como *gateway* responsável pela função do NAT64. As configurações deste equipamento são descritas na figura 2.7.

			Original Packet						Translated Packet	
	Name	Tags	Source Zone	Destination Zone	Desti... Inter...	Source Address	Destination Address	Service	Source Translation	Destination Translation
1	NAT_64_UFPE_POP	none	UFPE	POP-RNP	any	2001:12f0:912::/48	64:ff9b::/96	any	dynamic-ip-and-port ethernet1/24 200.133.31.2/29	none
2	NAT_64_UFPE_DMZ	none	UFPE	DMZ-UFPE	any	2001:12f0:912::/48	64:ff9b::/96	any	dynamic-ip-and-port ethernet1/21 150.161.100.254/29	none
3	NAT_64_UFPE_GER...	none	UFPE	GERENCIA	any	2001:12f0:912::/48	64:ff9b::/96	any	dynamic-ip-and-port ethernet1/21 150.161.100.254/29	none

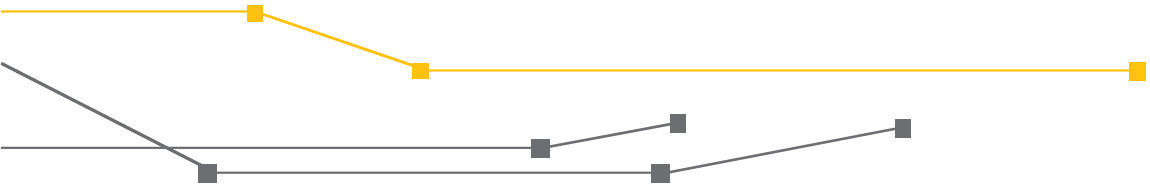
Figura 2.7 – Configuração das políticas de NAT64 no Firewall Palo Alto PA-5050.

Os campos das políticas de NAT64 são:

Name: nome da política de NAT64.

Tags: etiqueta descritiva (não obrigatório).

Source Zone: zona de origem, no caso do laboratório, a zona que representa a LAN.



Destination Zone: zona de destino, no caso do laboratório, a zona que representa a WAN.

Destination Interface: interface destino. Pode ser definida a interface de saída para o provedor com trânsito IPv4 ou simplesmente deixar qualquer (*any*) interface e o pacote será roteado segundo as políticas de roteamento do equipamento.

Source Address: endereço de origem. Definido pelo prefixo IPv6 alocado à LAN do laboratório.

Destination Address: endereço de destino. Definido com o endereço reservado 64:ff9b::/96, que caracteriza um destino traduzido no escopo do NAT64.

Service: serviços ou protocolos que serão permitidos nesta política. Neste caso, qualquer (*any*) protocolo foi a opção adotada.

Source Translation: origem da tradução. Foram definidos a interface de origem (LAN) e um endereço IPv4 público para compor o campo "*source address*" do cabeçalho IPv4, após a tradução.

Destination Translation: tradução de destino. Faz a tradução de um destino, porém foi necessário nesta solução.

Servidor DNS64

A configuração do servidor DNS é bem simples. O software utilizado foi o Bind9, cuja configuração é exposta abaixo:

Configuração do Bind9 no arquivo "named.conf.local":

```
forwarders {
      2001:4860:4860::8888; \\Encaminhador Consulta DNS
};
view "dns64" {
   dns64      64:ff9b::/96 {
      clients { any; };
     exclude { 64:ff9b::/96; ::ffff:0000:0000/96; };
     suffix ::;
   };
}; //end view dns64
```

Documentação oficial do BIND: https://www.isc.org/bind-9-11-arm/

Caso não seja de interesse do gestor de TI configurar um servidor DNS64, pode-se utilizar um servidor público, como o que é ofertado pelo Google:

https://developers.google.com/speed/public-dns/docs/dns64

Para saber detalhes acerca da implementação do NAT64 em sistemas Linux, deve-se acessar:

https://www.jool.mx

http://www.litech.org/tayga/

2.2. Solução *open source* de uma rede *wireless* puramente IPv6 com NAT64

Neste projeto, uma solução *open source* foi testada, onde uma rede *wireless* puramente IPv6, juntamente com a técnica de transição NAT64, foram implementadas. Esta arquitetura foi montada com um Raspberry Pi, funcionando como *gateway* NAT64, uma placa *wireless*, um DNS64 público e diversos softwares com funções específicas. Assim a conectividade puramente IPv6 pôde ser ofertada a dispositivos sem fio. A Figura 2.8 ilustra a topologia.

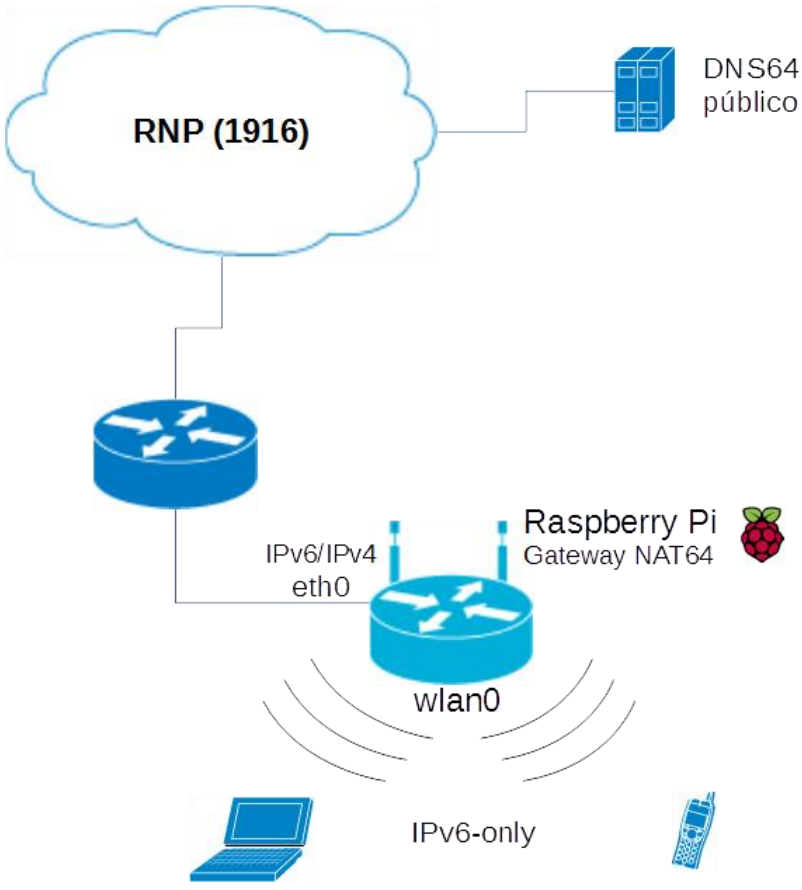
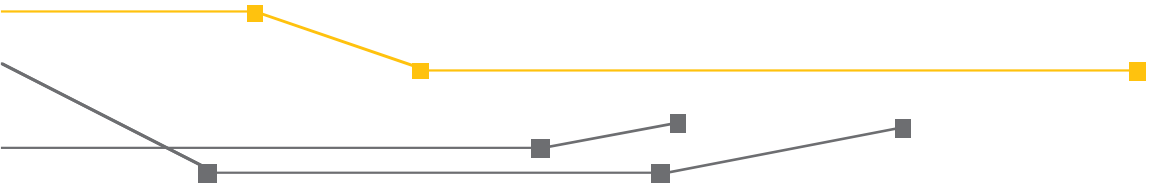


Figura 2.8 - Topologia da rede *wireless* puramente IPv6.



Abaixo seguem as instruções de como replicar o ambiente citado anteriormente.

Componentes utilizados:

Hardware:
- Raspberry Pi modelo B;
- Placa Wireless Edimax USB (driver rtl8192cu);
- Cartão SD 8GB Classe 10;
- Cabo de alimentação USB;
- Cabo UTP CAT-5.

Software:
- Sistema operacional Raspbian "2016-11-25-raspbian-jessie" (Debian);
- Hostapd – daemon com função de controlador wireless;
- RADVD – daemon responsável pelo encaminhamento de mensagens de Router Advertisement;
- ISC-DHCP-SERVER – funciona como um DHCPv6 Stateless;
- Jool – daemon responsável pelas funções de NAT64.

Configuração das interfaces de rede

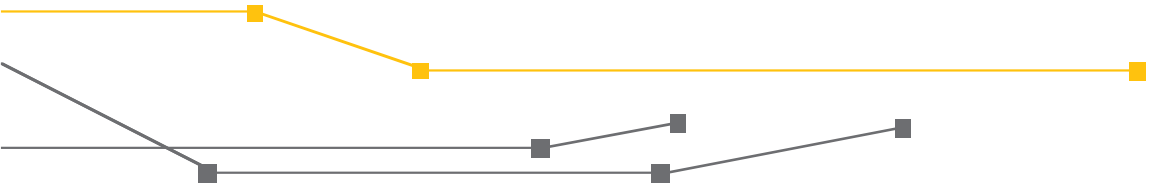
Inicialmente, é necessário configurar as interfaces diretamente conectadas ao Raspberry Pi.

A interface que possui conectividade com a internet é a eth0. Esta será configurada com endereçamento IPv4/IPv6 (Pilha dupla / *Dual-stack*). O endereço IPv4 será utilizado como IP de origem nos cabeçalhos traduzidos no escopo do NAT64.

A interface de LAN, que proverá conectividade puramente IPv6 a usuários, tanto para destinos IPv4 quanto IPv6, é a interface wlan0 (interface *wireless*). Na interface de LAN só haverá endereçamento IPv6, formando uma rede puramente IPv6 (IPv6-*only*).

Parâmetros das interfaces do Raspberry Pi:

```
root@raspberrypi:/home/pi# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:d7:37:75
          inet addr:200.143.193.139  Bcast:200.143.193.191  Mask:255.255.255.192
          inet6 addr: 2001:12f0:41c::1/64 Scope:Global
          inet6 addr: fe80::ba27:ebff:fed7:3775/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12894 errors:0 dropped:28 overruns:0 frame:0
          TX packets:2538 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1750842 (1.6 MiB)  TX bytes:354251 (345.9 KiB)

wlan0     Link encap:Ethernet  HWaddr 74:da:38:0e:2e:15
          inet addr:169.254.126.114  Bcast:169.254.255.255  Mask:255.255.0.0
          inet6 addr: 2001:12f0:41c:100::1916/64 Scope:Global
          inet6 addr: fe80::1e8d:1ce8:c1e8:3c8d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3074 errors:0 dropped:20 overruns:0 frame:0
```

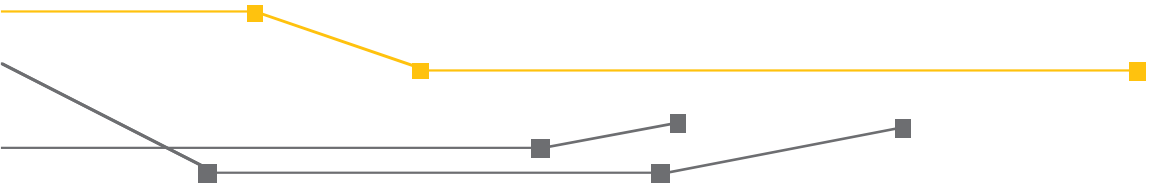


```
        TX packets:2552 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:422873 (412.9 KiB)  TX bytes:1441192 (1.3 MiB)


pi@raspberrypi:~ $ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address 200.143.193.139
   netmask 26
   gateway 200.143.193.135
   dns-nameservers 8.8.8.8

iface eth0 inet6 static
   address 2001:12f0:41c::1
   netmask 64
   gateway 2001:12f0:41c::2

allow-hotplug wlan0
iface wlan0 inet6 static
   address 2001:12f0:41c:100::1916
   netmask 64
   dns-nameservers 2001:4860:4860::6464 2001:4860:4860::64
   #wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Atualização do sistema

Para atualizar os pacotes, bibliotecas e o kernel, aplique os seguintes comandos:

```
pi@raspberrypi:~ $sudo -s
root@raspberrypi:/home/pi# apt-get update
root@raspberrypi:/home/pi# rpi-update
root@raspberrypi:/home/pi# reboot

root@raspberrypi:/home/pi# apt-get upgrade
root@raspberrypi:/home/pi# apt-get install raspberrypi-kernel-headers
root@raspberrypi:/home/pi# apt-get install build-essential


pi@raspberrypi:~ $ uname -a
Linux raspberrypi 4.4.38+ #938 Thu Dec 15 15:17:54 GMT 2016 armv6l GNU/Linux
```



Instalação do Jool

O software escolhido para suportar as funcionalidades do NAT64 foi o Jool. Este foi desenvolvido pelo NIC do México (http://nicmexico.mx/) e pode ser compilado em sistemas Linux. Para mais informações acesse: https://www.jool.mx

Download do Jool - https://www.jool.mx/en/download.html

```
root@raspberrypi:~/Downloads# wget https://github.com/NICMx/releases/raw/master/Jool/Jool-3.5.2.zip
```

Descompacte-o:

```
root@raspberrypi:~/Downloads# unzip Jool-3.5.2.zip
```

Instale a aplicação dkms:

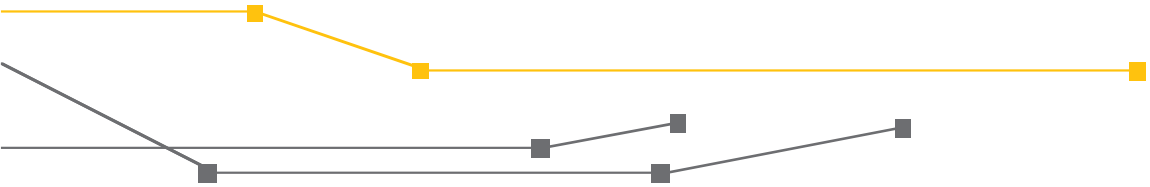
```
root@raspberrypi:~/Downloads# apt-get install dkms
```

Para finalizar, instale o pacote Jool, fora da pasta extraída, e inicie o processo. Observação: a instalação pode levar um bom tempo.

```
root@raspberrypi:~/Downloads# ls
Jool-3.5.2  Jool-3.5.2.zip  radvd-2.15  radvd-2.15.tar.gz

root@raspberrypi:~/Downloads# dkms install Jool-3.5.2

root@raspberrypi:~/Downloads# /sbin/modprobe jool pool6=64:ff9b::/96
```

Para iniciar o processo do Jool durante o boot, adicione o comando "/sbin/modprobe jool pool6=64:ff9b::/96" antes da string "exit 0" no arquivo /etc/rc.local.

```
pi@raspberrypi:~ $ cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
```



```
fi

#Inicia JOOL durante o boot

/sbin/modprobe jool pool6=64:ff9b::/96

exit 0
```

Instalação e configuração do *daemon* HOSTAPD

Para conectividade *wireless*, entre os hosts usuários e o *gateway* (Raspberry Pi), o software hostapd será configurado.

Instale o software através do seguinte comando:

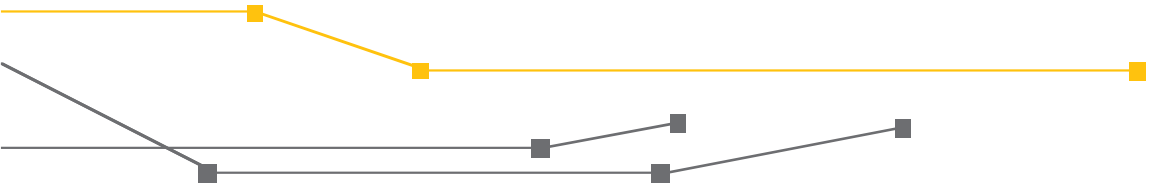
```
root@raspberrypi:~/Downloads#  apt-get install hostapd
```

A seguinte configuração determina os parâmetros da rede *wireless*. As variáveis abaixo podem ser customizadas para atender cenários específicos.

```
pi@raspberrypi:~ $ cat /etc/hostapd/hostapd.conf
interface=wlan0  #nome da interface wireless
driver=nl80211
ssid=IPv6_only                  #nome da rede wifi
hw_mode=g                       #modo de operação da interface
channel=6                       #canal da rede wifi
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=ipv6_rnp    #senha da rede wifi
wpa_key_mgmt=WPA-PSK
#wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Inclua a linha de configuração " DAEMON_CONF="/etc/hostapd/hostapd.conf" " no arquivo /etc/default/hostapd para localizar a configuração realizada no bloco anterior.

```
root@raspberrypi:/home/pi# cat /etc/default/hostapd

# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"

# Additional daemon options to be appended to hostapd command:-
#       -d   show more debug messages (-dd for even more)
```



```
#      -K   include key data in debug messages
#      -t   include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automatically
# configured by the init.d script and must not be added to DAEMON_OPTS.
#
#DAEMON_OPTS=""
```

Instalando RADVD 2.15, que tem suporte ao RDNSS

O método escolhido para delegação de endereços IPv6 a usuários ocorre através do processo radvd, que envia mensagens de *Router Advertisement* com basicamente duas informações:

1 - Prefixo de rede a ser utilizado no processo de SLAAC (https://en.wikipedia.org/wiki/IPv6#Stateless_address_autoconfiguration_.28SLAAC.29)

2 - Servidores DNS64 recursivos, que compõe a solução do NAT64. Neste caso foram utilizados os servidores públicos de DNS64 do Google (https://developers.google.com/speed/public-dns/docs/dns64)

```
root@raspberrypi:/home/pi# apt-get install bison flex
```

Baixe o radvd diretamente do website http://www.litech.org/radvd/, pois com o "apt-get install radvd" a versão instalada não suporta a função de RDNSS.

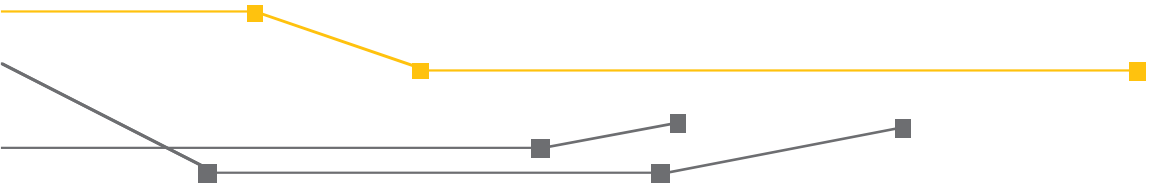
```
root@raspberrypi:/home/pi# wget http://www.litech.org/radvd/dist/radvd-2.15.tar.gz

root@raspberrypi:/home/pi# tar -vzxf radvd-2.15.tar.gz
```

Entre na pasta do radvd e rode:

```
root@raspberrypi:/home/pi# ./configure --prefix=/usr/local --sysconfdir=/etc --mandir=/usr/share/man

root@raspberrypi:/home/pi# make

root@raspberrypi:/home/pi# make install
```

Verifique a versão instalada:

```
root@raspberrypi:/home/pi/Downloads/radvd-2.15# radvd -v
Version: 2.15

Compiled in settings:
  default config file        "/etc/radvd.conf"
  default pidfile            "/var/run/radvd.pid"
  default logfile            "/var/log/radvd.log"
  default syslog facility     24
```



Configure o arquivo /etc/radvd.conf com os parâmetros desejados.

Prefix será o prefixo alocado a interface de LAN (wlan0) para que os hosts consigam realizar a autoconfiguração de suas interfaces de rede.

O bloco de RDNSS pode ser copiado caso queira utilizar os servidores públicos de DNS64 do Google, ou substitua os endereços para os servidores DNS64 desejados.

```
root@raspberrypi:/home/pi# cat /etc/radvd.conf
interface wlan0 {
      AdvManagedFlag on;
      AdvSendAdvert on;
      MinRtrAdvInterval 3;
      MaxRtrAdvInterval 60;
      route ::/0 {
      };
      prefix 2001:12f0:41c:100::/64 {
            AdvOnLink on;
            AdvAutonomous on;
            AdvRouterAddr on;
      };
      RDNSS 2001:4860:4860::6464 2001:4860:4860::64 {
            AdvRDNSSLifetime 30;
      };
};
```

Stateless DHCPv6 - Para suporte a DNS

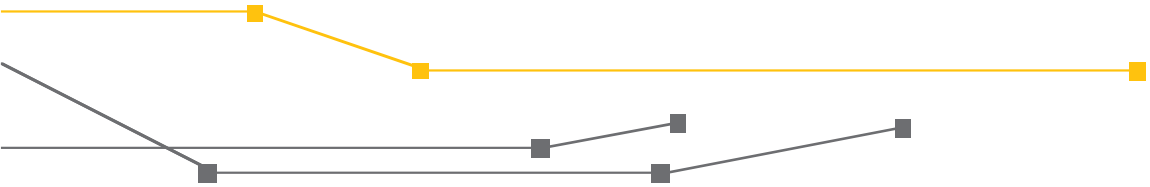
Dentre vários sistemas operacionais testados (Android, Apple, Linux), somente o Microsoft Windows (versões 7 e 10) não suporta a RFC6106 (RDNSS). Logo, faz-se necessário a entrega de servidores DNS através de outro método, sendo este o Stateless DHCPv6.

Instale o software isc-dhcp-server:

```
root@raspberrypi:/home/pi# apt-get install isc-dhcp-server
```

Configure o arquivo /etc/default/isc-dhcp-server, conforme abaixo:

```
root@raspberrypi:/home/pi# cat /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
DHCPD_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
DHCPD_PID=/var/run/dhcpd6.pid
```



```
# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
OPTIONS="-6"

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="wlan0"
```

Configure o arquivo /etc/dhcp/dhcpd6.conf com a subnet de LAN e adicione o range de prefixos alocáveis a usuários e os servidores DNS64.

```
root@raspberrypi:/home/pi# cat  /etc/dhcp/dhcpd6.conf
ddns-update-style none;
default-lease-time 86400;
max-lease-time 86400;
authoritative;

subnet6 2001:12f0:41c:100::/64
{
   range6 2001:12f0:41c:100::100 2001:12f0:41c:100::300;
   option dhcp6.name-servers 2001:4860:4860::6464, 2001:4860:4860::64;
   option dhcp6.domain-search "rnp.br";
}
```

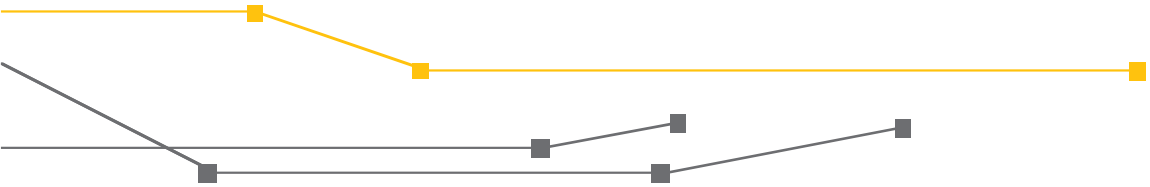
Crie o arquivo /var/lib/dhcp/dhcpd6.leases :

```
root@raspberrypi:/home/pi# touch /var/lib/dhcp/dhcpd6.leases

root@raspberrypi:/home/pi# systemctl start isc-dhcp-server
root@raspberrypi:/home/pi# systemctl status isc-dhcp-server
```

Habilitar roteamento IPv6

Para habilitar o encaminhamento de pacotes (roteamento) IPv6 pelo Raspberry Pi, insira a seguinte linha de configuração "net.ipv6.conf.all.forwarding=1" no arquivo /etc/sysctl.conf.

```
pi@raspberrypi:~ $ sudo cat /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

##############################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
```

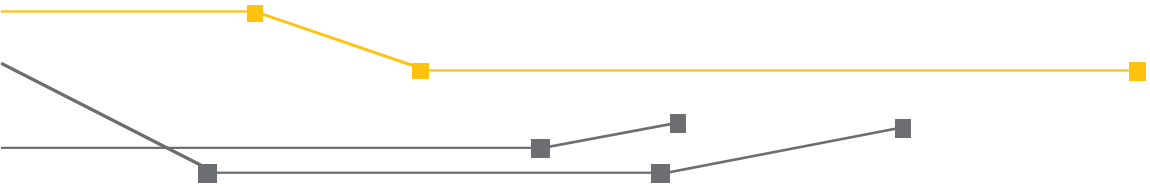


```
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#
```

Habilitar processos aos próximos *boots*

Para que o sistema inicialize os processos necessários durante o *boot*, insira os seguintes comandos:

```
pi@raspberrypi:~ $ sudo update-rc.d hostapd enable

pi@raspberrypi:~ $ sudo update-rc.d radvd enable

pi@raspberrypi:~ $ sudo update-rc.d isc-dhcp-server enable
```



Reinicie o sistema novamente:

```
pi@raspberrypi:~ $ sudo reboot
```

2.3. Homologação da usabilidade do NAT64

Sistemas operacionais testados:

- Windows 7 (SP1);
- Windows 10
- Linux Mint e Ubuntu - Kernel 4.4.0 (a distribuição não implica nos resultados e sim a versão do Kernel).
- Iphone (4s, 5s e 6s)
- Android (5.1)

Nos testes efetuados, a maioria das aplicações dependentes de conectividade funcionou com a técnica de transição NAT64 sem dificuldades. Porém, o aplicativo Skype, muito usual para troca de mensagens e chamadas de voz e vídeo, apresentou problemas em seu software cliente impossibilitando o seu uso. Já a sua plataforma web funcionou apenas para mensagens de texto.

Abaixo, segue um resumo das aplicações testadas. No Anexo I, são exibidas algumas capturas de tela com as evidências dos testes realizados.

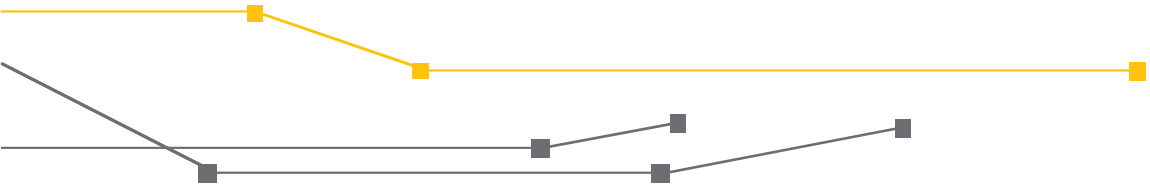
Aplicações que funcionaram:

- Teamviewer
- Mconf
- Filesender@rnp
- NTP
- FTP
- Telnet
- SSH
- HTTP
- HTTPS
- IMAP
- Java
- Ping
- Traceroute
- NMAP
- Aplicativos de internet banking (Itaú, BB)
- Whatsapp, Facebook, Instagram, Twitter
- Google Maps, Gmail
- Dropbox
- Evernote
- Youtube

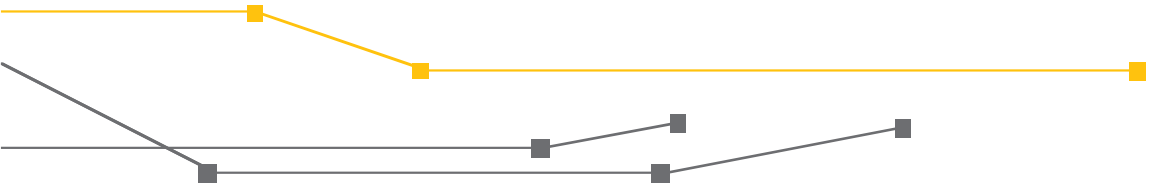
Aplicação que não funcionou:

- Skype

Observação: Para algumas aplicações onde não é suportada a inserção de um endereço IPv6 em um campo de entrada, pode-se utilizar um nome DNS para tal finalidade. Quando uma aplicação suporta IPv6 em campos de entrada, mas o endereço não está mapeado para um nome DNS, a



conversão do endereço IPv4 para o formato NAT64 (64:ff9b::<endereço IPv4 em hexadecimal>) pode ser necessária.



3. Conclusão

O NAT64 como técnica de transição entre os protocolos IPv4 e IPv6 mostrou-se funcional dentre uma gama de aplicações demandantes por conectividade. Como toda tecnologia, o NAT64 impõe limitações em determinados cenários, onde a presença do IPv4 está enraizada, muitas vezes no código fonte de um software.

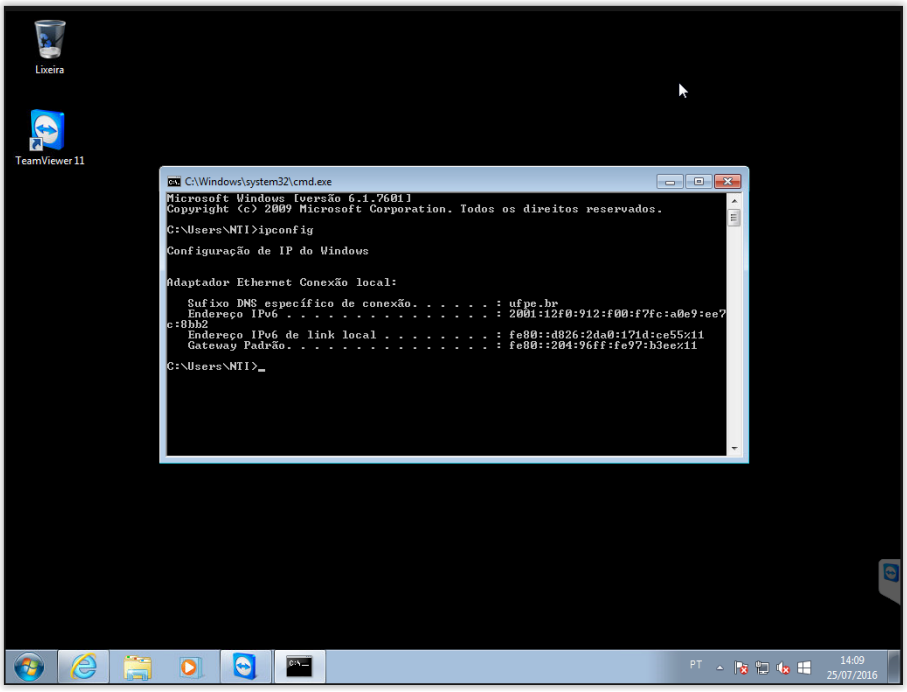
Uma excelente aplicação desta técnica dá-se em redes de grande densidade de usuários, como redes sem fio (wifi), onde não é possível a utilização de endereços válidos para cada usuário final. Ambientes controlados, como redes locais virtuais (VLANs), onde se sabe que a conectividade não será impactada por alguma limitação do protocolo, também são ótimos ambientes para sua implementação.

Os movimentos em direção à implementação do protocolo IPv6 são incontestáveis. Logo, de uma maneira ou outra, esta e outras técnicas existentes deverão ser estudadas, testadas e implementadas, possibilitando a continuidade da expansão e operação de ambientes de TIC.

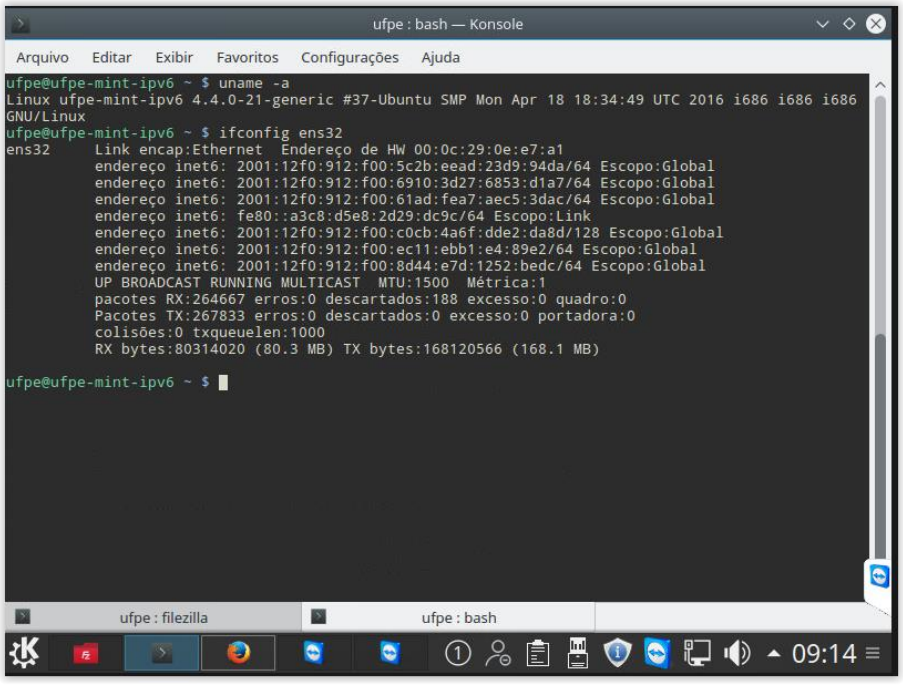
Anexo I – Capturas de telas

Configuração de rede:

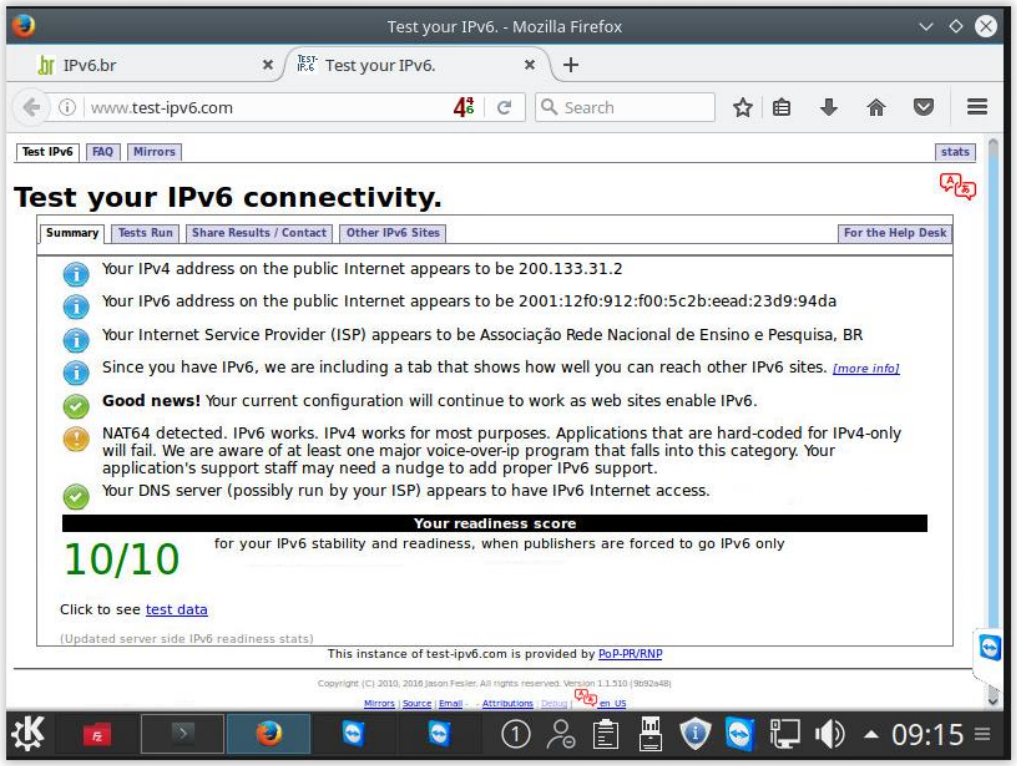
Windows 7:



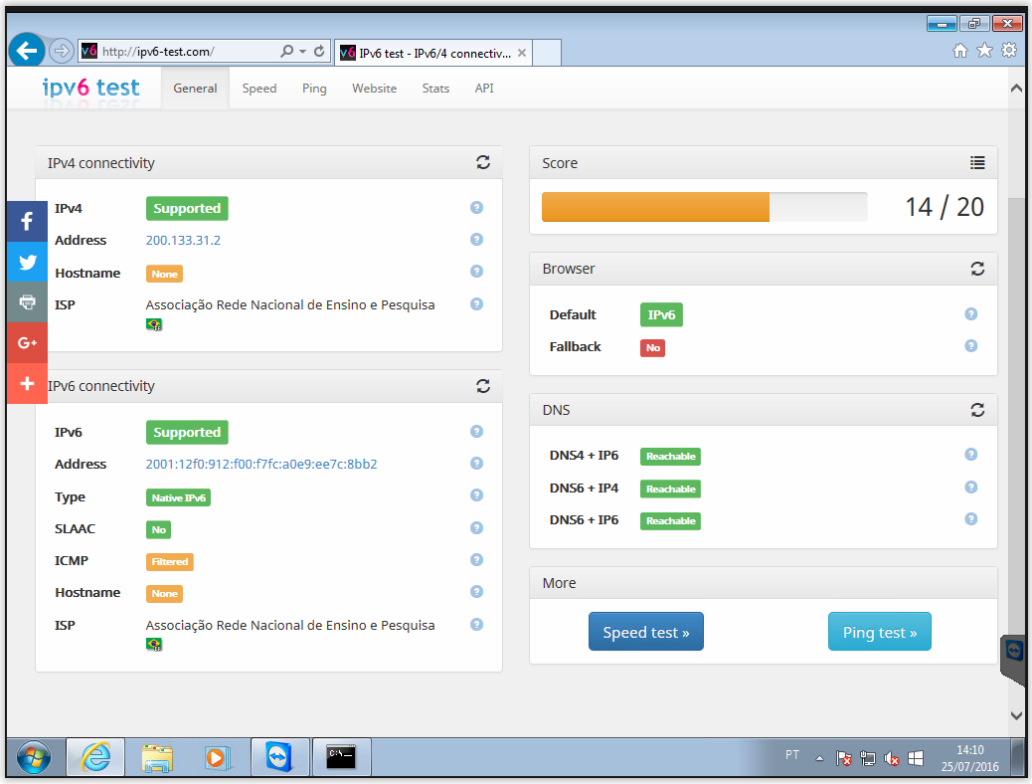
Linux Mint:



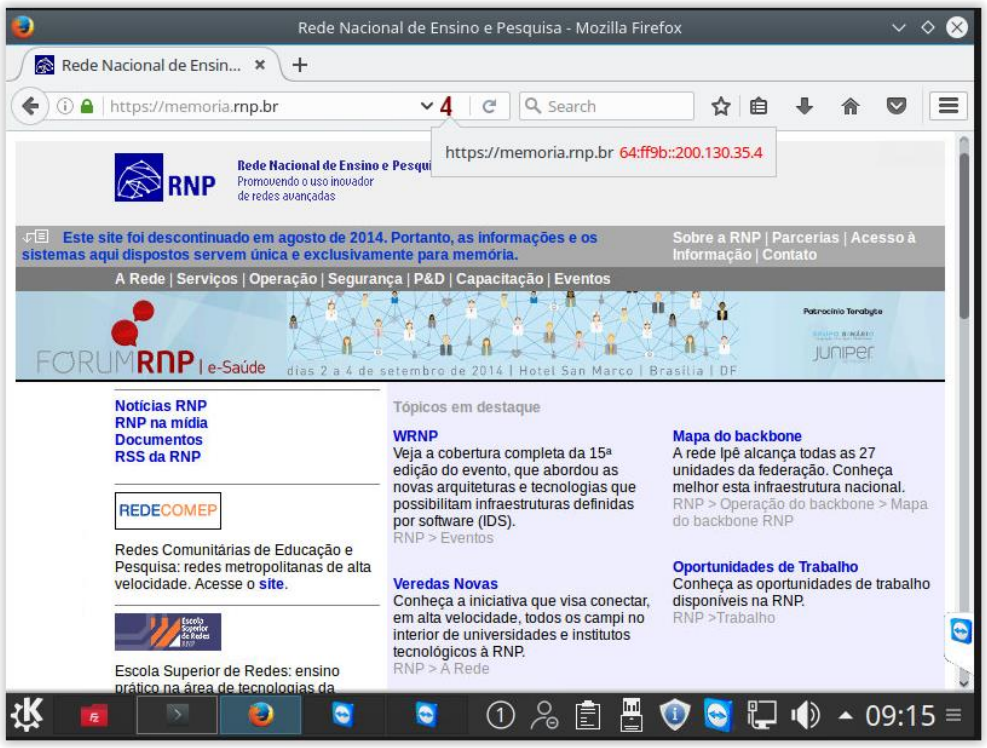
www.test-ipv6.com (Linux):



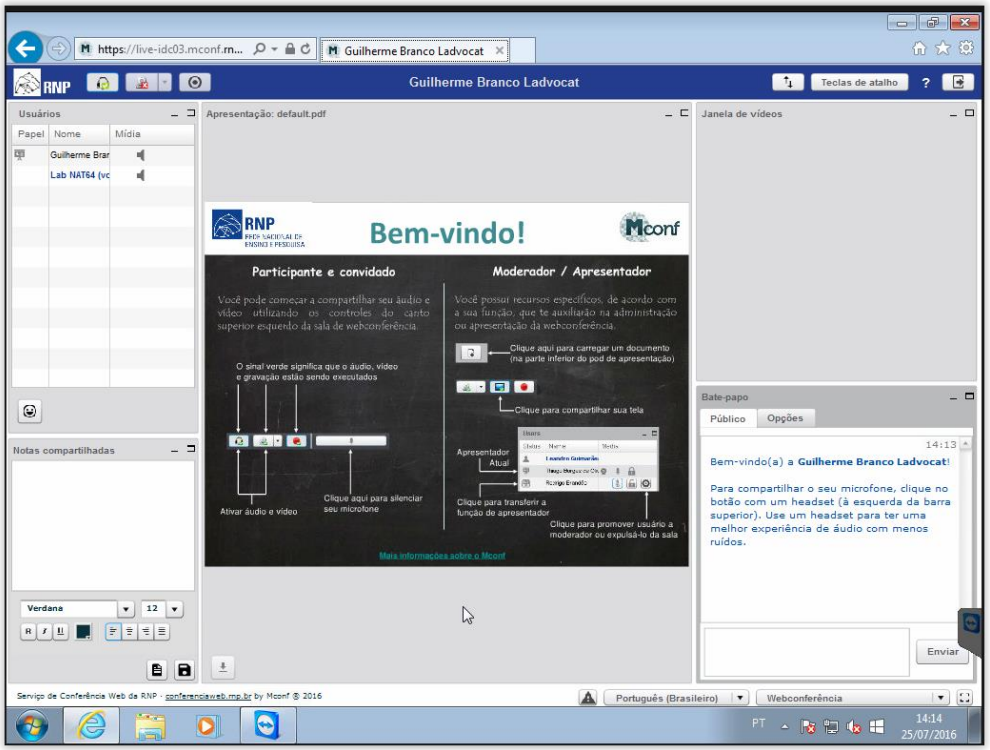
http://ipv6-test.com (Windows):



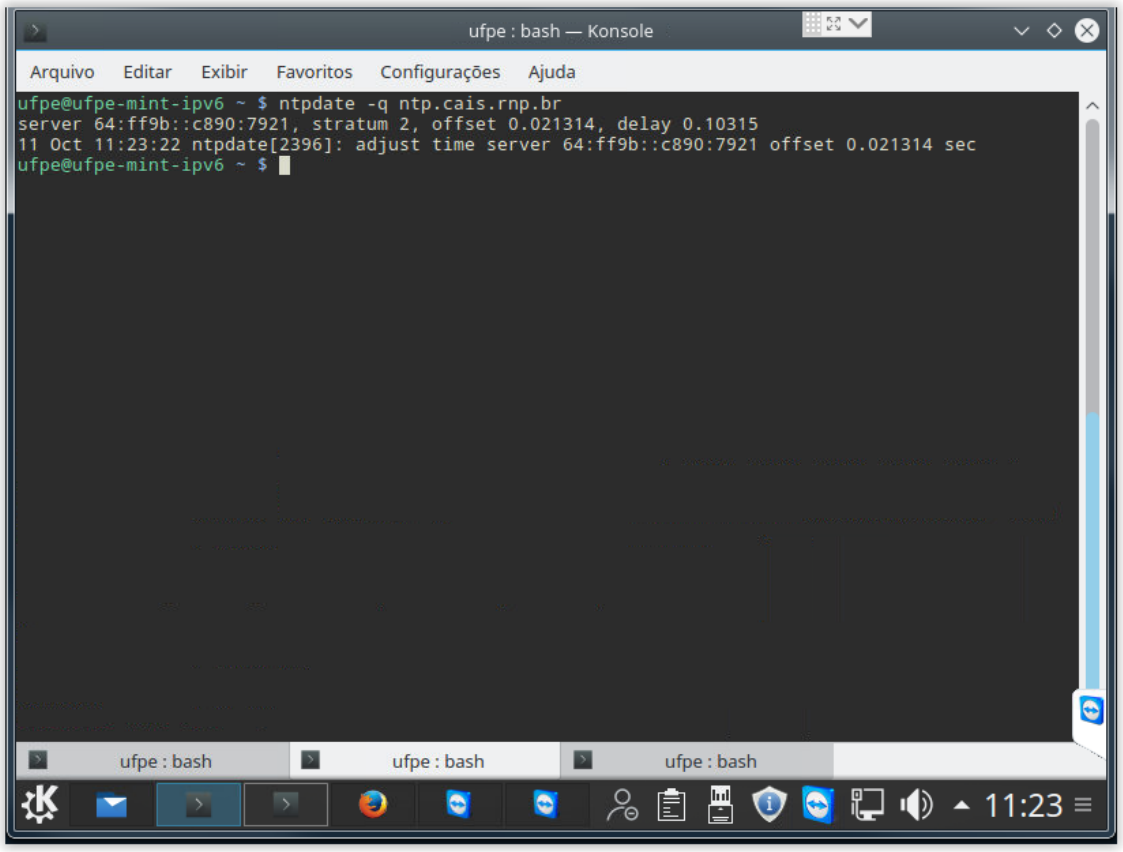
HTTPS:



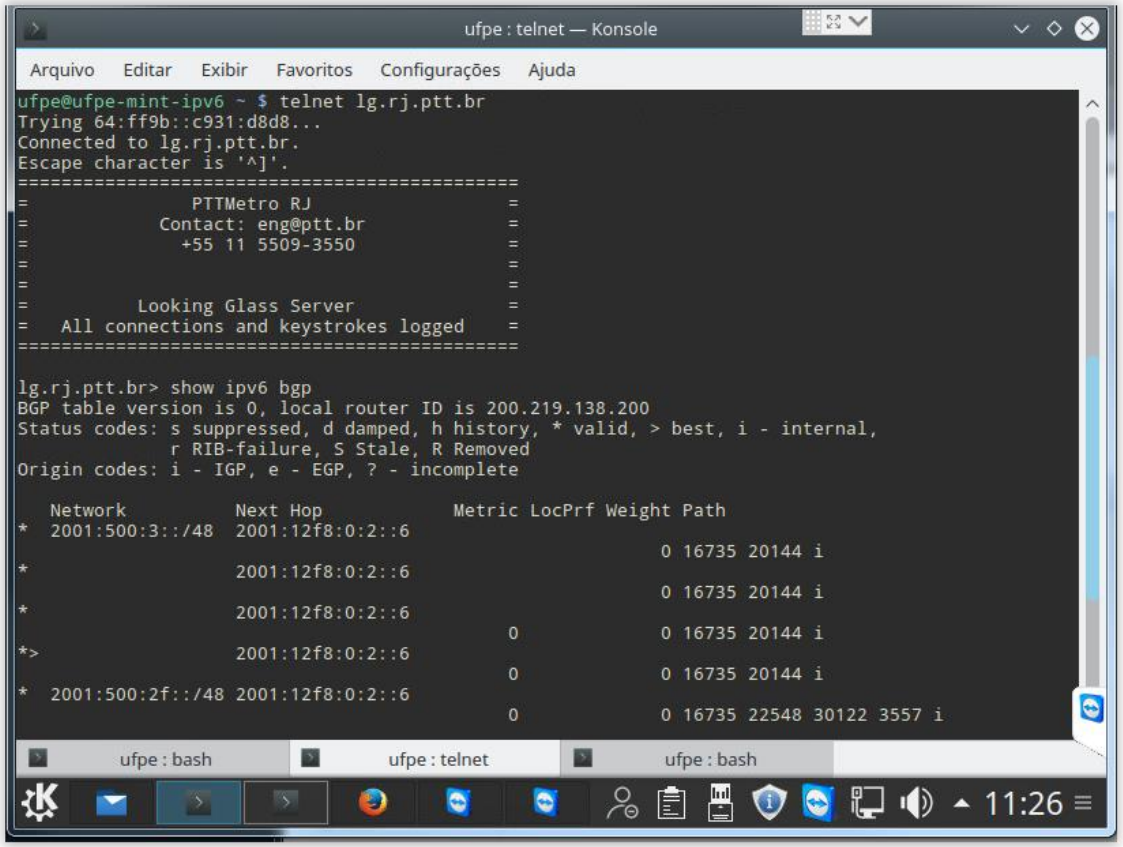
Mconf (HTTPS):



NTP:


ufpe : bash — Konsole
Arquivo Editar Exibir Favoritos Configurações Ajuda
ufpe@ufpe-mint-ipv6 ~ $ ntpdate -q ntp.cais.rnp.br
server 64:ff9b::c890:7921, stratum 2, offset 0.021314, delay 0.10315
11 Oct 11:23:22 ntpdate[2396]: adjust time server 64:ff9b::c890:7921 offset 0.021314 sec
ufpe@ufpe-mint-ipv6 ~ $
ufpe : bash
ufpe : bash
ufpe : bash
11:23

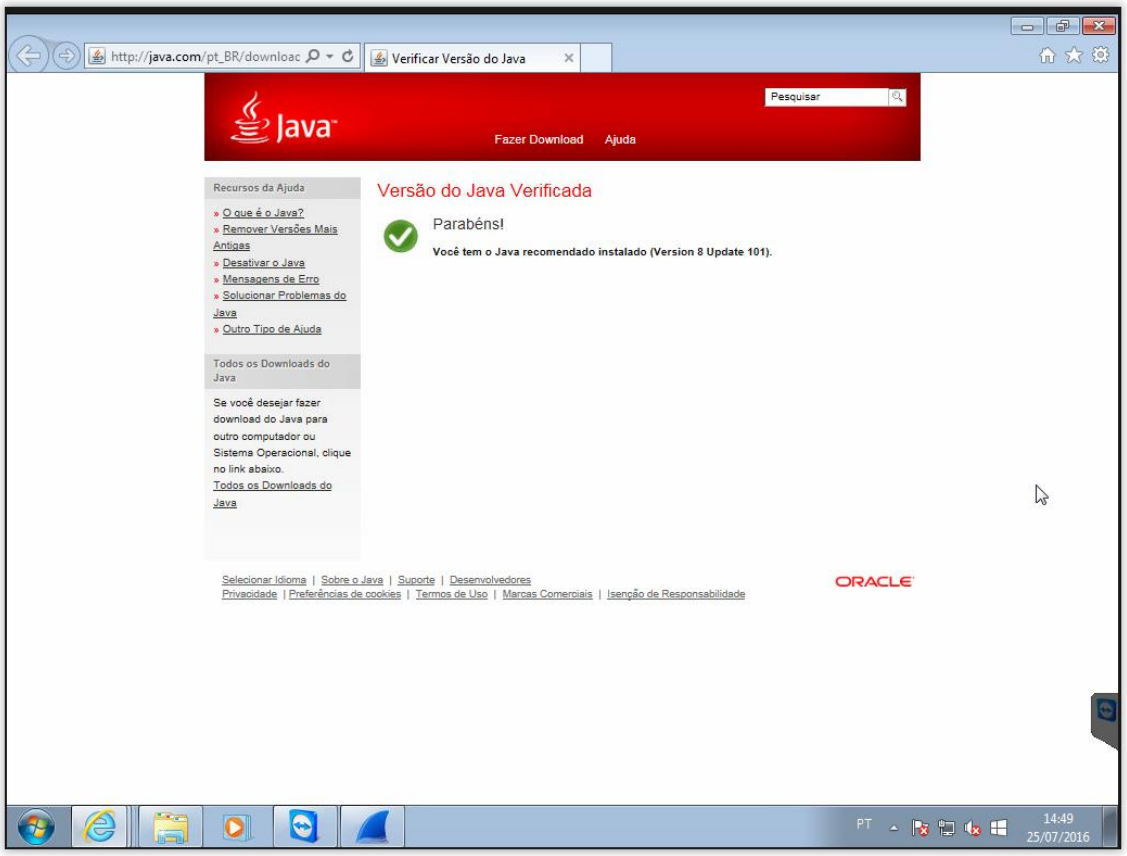
Telnet:


ufpe : telnet — Konsole
Arquivo Editar Exibir Favoritos Configurações Ajuda
ufpe@ufpe-mint-ipv6 ~ $ telnet lg.rj.ptt.br
Trying 64:ff9b::c931:d8d8...
Connected to lg.rj.ptt.br.
Escape character is '^]'.
PTTMetro RJ
Contact: eng@ptt.br
+55 11 5509-3550
Looking Glass Server
All connections and keystrokes logged
lg.rj.ptt.br> show ipv6 bgp
BGP table version is 0, local router ID is 200.219.138.200
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
* 2001:500:3::/48 2001:12f8:0:2::6 0 16735 20144 i
* 2001:12f8:0:2::6 0 16735 20144 i
* 2001:12f8:0:2::6 0 16735 20144 i
*> 2001:12f8:0:2::6 0 16735 20144 i
* 2001:500:2f::/48 2001:12f8:0:2::6 0 16735 22548 30122 3557 i
ufpe : bash
ufpe : telnet
ufpe : bash
11:26

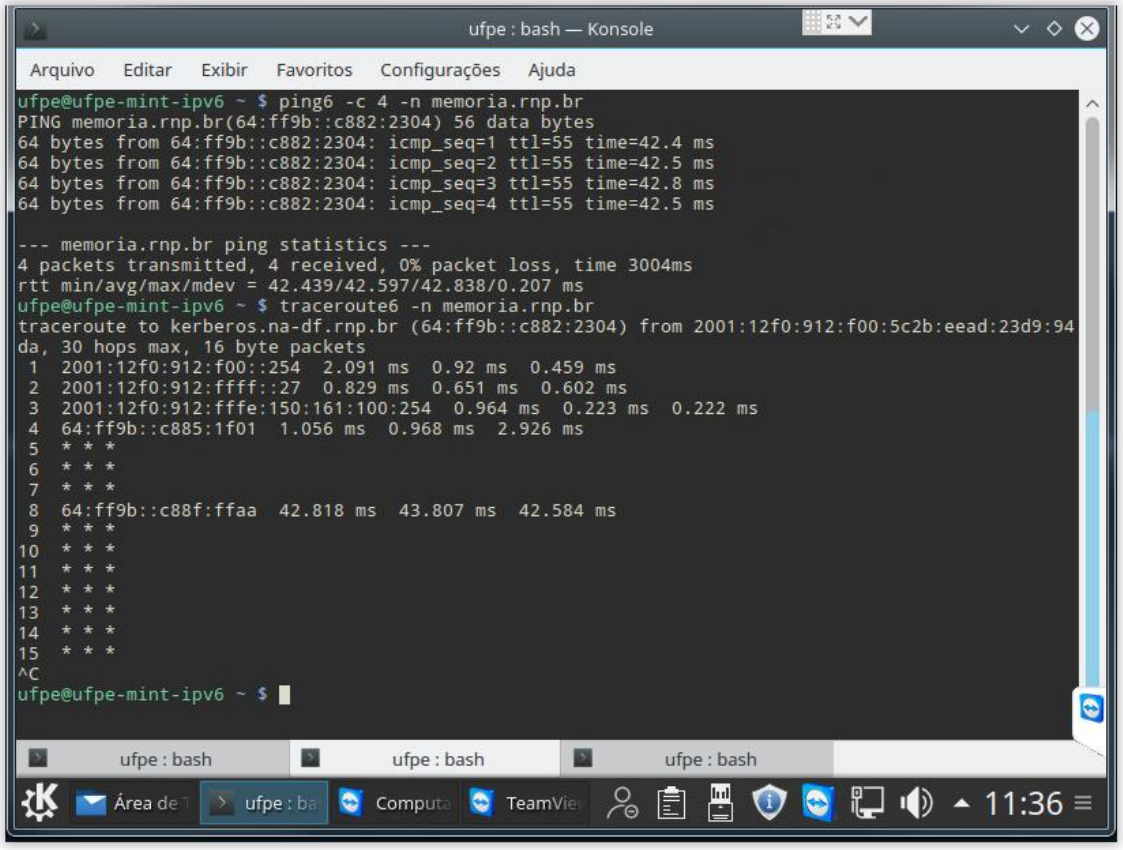
HTTP:


http://www.speedtest.net/
Speedtest.net by Ookla - T...
OOKLA SPEEDTEST PINGTEST AWARDS
The Global Standard in Internet Metrics
SPEEDTEST
ADVERTISE BECOME A HOST MY RESULTS SUPPORT SETTINGS LOGIN CREATE ACCOUNT
Get Google Chrome
Fast, simple & secure web browser for all your devices. Download now!
PING 2 ms
DOWNLOAD SPEED 93.92 Mbps
UPLOAD SPEED 94.64 Mbps
NEW SERVER
TEST AGAIN
SHARE THIS RESULT
Get Google Chrome
A fast, secure, and free browser for all your devices. Download now!
Are you on RNP?
Take our Broadband Internet Survey!
Measure the quality of your connection.
BEGIN TEST
200.133.31.2
RNP
Rate Your ISP
Carpina
Hosted by PIERNET TELECOM
15 MEGA
POR APENAS
R$64,90 /MÊS
PEÇA JÁ
15 MEGA POR R$/MÊS
WI-FI E ANTIVÍRUS GRÁ
ASSINE JÁ
OU LIGUE 3003-7303
10 Gbps IP Transit $2000/month
IPv6+IPv4 and BGP for Your Network or Internet Company!
Learn More
Gmail for Work
Seja mais profissional com o e-mail personalizado.
Teste Grátis
https://www.googleadservices.com/pagead/aclk?sa=L&ai=C3-ePWFCWV5HhF4-3fNvku7AJ...
PT
14:46
25/07/2016

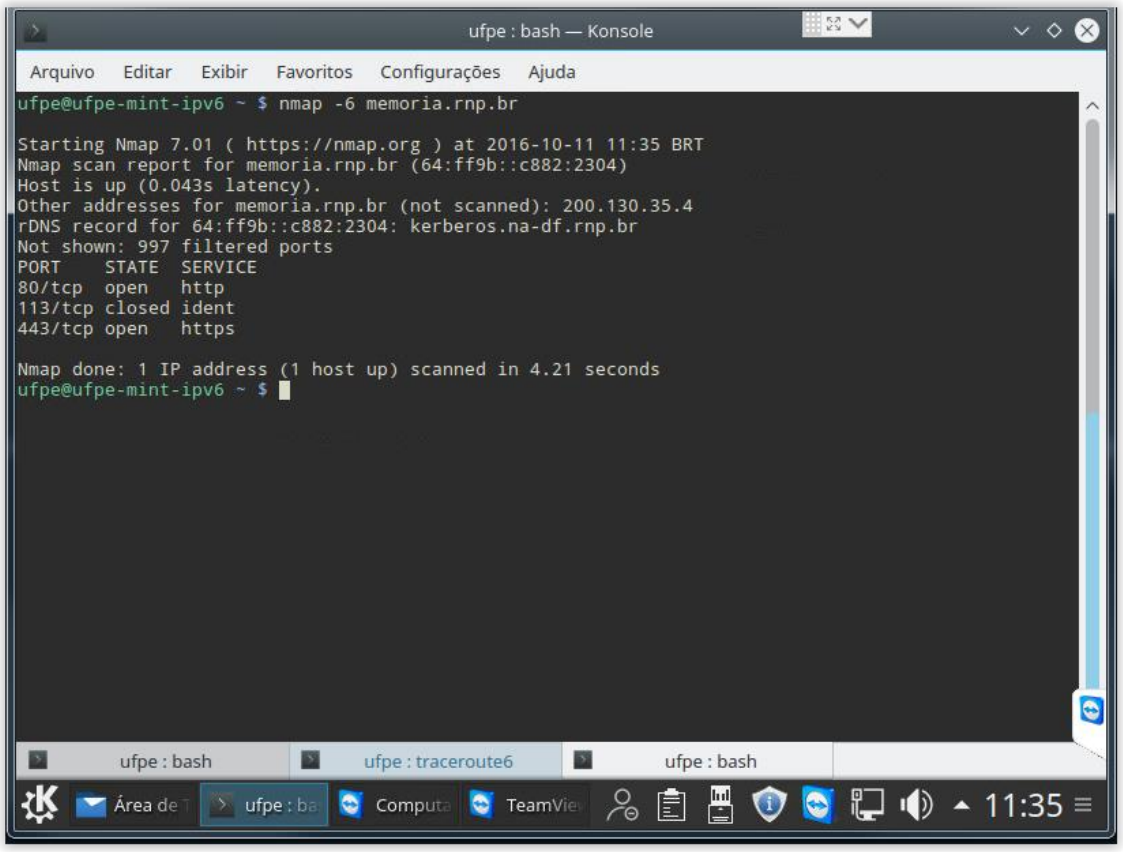
Java:


http://java.com/pt_BR/downloac
Verificar Versão do Java
Java
Pesquisar
Fazer Download
Ajuda
Recursos da Ajuda
» O que é o Java?
» Remover Versões Mais Antigas
» Desativar o Java
» Mensagens de Erro
» Solucionar Problemas do Java
» Outro Tipo de Ajuda
Todos os Downloads do Java
Se você desejar fazer download do Java para outro computador ou Sistema Operacional, clique no link abaixo.
Todos os Downloads do Java
Versão do Java Verificada
Parabéns!
Você tem o Java recomendado instalado (Version 8 Update 101).
Selecionar Idioma | Sobre o Java | Suporte | Desenvolvedores
Privacidade | Preferências de cookies | Termos de Uso | Marcas Comerciais | Isenção de Responsabilidade
ORACLE
PT
14:49
25/07/2016

Ping e Traceroute:



NMAP:


ufpe : bash — Konsole
Arquivo Editar Exibir Favoritos Configurações Ajuda
ufpe@ufpe-mint-ipv6 ~ $ nmap -6 memoria.rnp.br
Starting Nmap 7.01 (https://nmap.org) at 2016-10-11 11:35 BRT
Nmap scan report for memoria.rnp.br (64:ff9b::c882:2304)
Host is up (0.043s latency).
Other addresses for memoria.rnp.br (not scanned): 200.130.35.4
rDNS record for 64:ff9b::c882:2304: kerberos.na-df.rnp.br
Not shown: 997 filtered ports
PORT STATE SERVICE
80/tcp open http
113/tcp closed ident
443/tcp open https
Nmap done: 1 IP address (1 host up) scanned in 4.21 seconds
ufpe@ufpe-mint-ipv6 ~ $
ufpe : bash
ufpe : traceroute6
ufpe : bash
11:35

RNP
Ministério da **Cultura**
Ministério da **Saúde**
Ministério da **Educação**
Ministério da **Ciência, Tecnologia e Inovação**
GOVERNO FEDERAL BRASIL PÁTRIA EDUCADORA