

Análise dos adaptadores USB para redes sem fio

Rádio USB



Os badulaques eletrônicos ficam cada vez menores – e os adaptadores para redes sem fio não são uma exceção. O primeiro chaveiro USB para WLAN chegou às lojas em 2003. Este artigo mostra o que devemos observar quando compramos um dispositivo WLAN USB e como colocar essas belezinhas para funcionar.

POR MARCEL HILZINGER

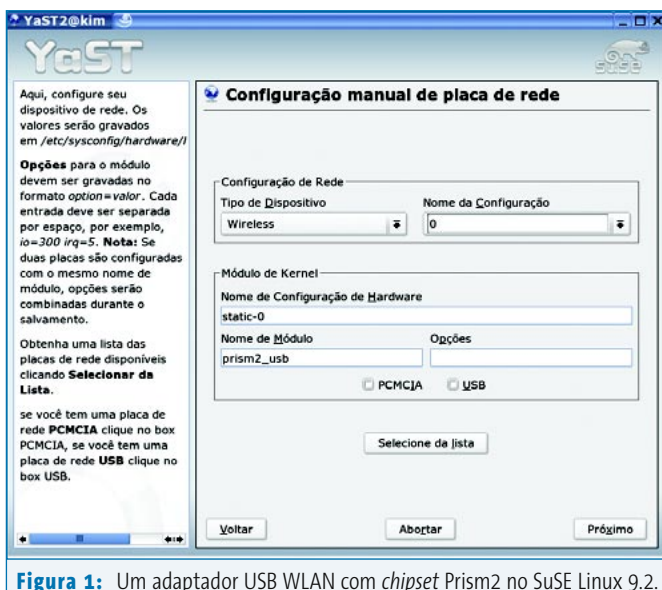


Figura 1: Um adaptador USB WLAN com chipset Prism2 no SuSE Linux 9.2.

O ano novo se foi, e aqui no hemisfério norte faz frio demais para que nos sentemos no jardim para trabalhar com nossos laptops. Entretanto, os adaptadores para rede sem fio não são de uso exclusivo dos amantes do ar livre. Em nossas casas, no trabalho, no aeroporto ou na biblioteca pública podemos nos servir de pontos de acesso (*hotspots*) sem a necessidade de cabos. Os “chaveiros” USB que acomodam uma interface de rede completa não são maiores do que os chaveiros comuns de memória que vimos na edição passada, não ocupando muito espaço no bolso da camisa ou na bolsa.

É preciso tomar cuidado com algumas “pegadinhas” na hora de comprar um dispositivo desse tipo. O Linux não reconhece a maioria dos novos dispositivos de 54 Mbit/s; mesmo os modelos mais antigos podem lhe trazer dores de cabeça.

Mantendo-se do lado certo

Como a explosão das redes sem fio ainda não atingiu o apogeu, novos fabricantes e fornecedores de chaveiros USB para WLAN pipocam aqui e ali todos os dias. Dito isso, os únicos adaptadores que realmente funcionam no Linux sem muito ranger de dentes operam a 11 Mbit/s e usam o chipset *Prism2*. Dois exemplos desses adaptadores são o Netgear MA111 e o D-Link DWL-122 (ver [tabela 1](#)). Uma lista com os dispositivos reconhecidos pelo projeto `Linux-wlan-ng` pode ser obtida em [\[1\]](#).

O SuSE Linux 9.0 ou mais recente oferece a forma mais fácil de operar esses dois adaptadores. Depois de inseridos na porta USB, chame o módulo de configuração de placas de rede do YaST, escolha *Other (not detected)* e depois clique em *Configure...* No *Device type*, ative a opção *Wireless* e clique em *Select from List*. Seu adaptador vai funcionar com o último item do menu: *Wireless device using kernel module prism2_usb.o*. Para confirmar, clique em *OK* (veja a [figura 1](#)). Depois basta ir até *Wireless Settings* e configurar o ESSID e o modo de operação.

Para usar um ponto de acesso público (por exemplo, no aeroporto) ou seu roteador WLAN em casa, use a configuração padrão (*Managed*). Basta então informar o ESSID e clicar em *OK*. Com isso, voltamos à janela *Manual network card configuration*. Se seu ponto de acesso é também o servidor DHCP (por sinal, o caso mais comum), simplesmente clique em *Next* e depois em *Finish*. Pronto! Já pode começar a navegar. Se você não possui um servidor de DHCP, entretanto, será preciso configurar um endereço IP estático. Consulte o responsável pelo seu ponto de acesso ou dê uma olhada na documentação do roteador WLAN para mais informações.

Para simplesmente conectar dois PCs diretamente, use o modo *ad-Hoc*. Nesse caso, é preciso também selecionar um canal de comunicação. O projeto `Linux-wlan-ng` usa o canal 6 por padrão – se estiver em dúvida, deixe como está. Na caixa de diálogo onde vc informou o ESSID, clique em *Expert settings* para selecionar o canal desejado. Os botões *OK* e *Next* levam à janela em que se pode configurar o endereço de rede. Escolha *Static address setup* e informe o IP. Podemos, por exemplo, usar 192.168.1.1 em uma máquina e 192.168.1.2 na outra.

Um último clique em *Next* deixará seu adaptador USB pronto para a ação. Em algumas distribuições, o dispositivo tem que ser configurado “à unha”. O [quadro 1](#) mostra como.

Quadro 1: Configurando um adaptador Prism2_usb

Para configurar um adaptador USB WLAN com o chipset Prism2, é preciso instalar o `linux-wlan-ng` [2] e as `wireless-tools` [3]. Embora a maioria das distribuições use dois pacotes separados para esses programas, o `linux-wlan-ng` é parte das `wireless-tools` no SuSE, não sendo necessário instalá-lo. O SuSE tem ainda uma abordagem diferente para configurar adaptadores USB baseados no chip Prism2, portanto aconselhamos aos usuários dessa distribuição que se atenham ao YaST.

Para compilar as `wireless-tools`, descompacte o arquivo `tar.gz`, mude para o diretório recém-criado e digite `make`. Depois, mude para o usuário `root` (com o comando `su`) e digite `make install` para copiar os arquivos para `/usr/local`. Para instalar em um diretório diferente (`/usr`, por exemplo), edite o `makefile`. O pacote `linux-wlan-ng` é instalado do mesmo jeito. Nesse caso, o `make config` abre um programa de instalação que permite escolher os drivers necessários. Para o chaveiro USB, diga `não` para as primeiras três perguntas e responda `sim` para `Build Prism2.5 USB (_usb) driver? (y/n)`. Para todas as outras perguntas, simplesmente pressione a tecla **Enter**.

Com os fontes do kernel instalados e corretamente configurados, o `make` pode iniciar o processo de compilação e o `make install` (sempre como `root`) instalará o programa.

Depois de tudo instalado, inclua a linha

```
alias wlan0 prism2_usb
```

no fim de seu arquivo `/etc/modprobe.conf` (ou `/etc/modules.conf` se seu kernel for da série 2.4). Depois, digite `modprobe prism2_usb` para carregar o módulo na memória. O comando `iwconfig` deve mostrar a interface `wlan0` que representa seu chaveiro (ver **quadro 2**).

Para ativar o adaptador, é preciso editar dois arquivos sob `/etc/wlan/`. Para permitir que duas placas wireless rodando em modo *ad-hoc* possam se enxergar mutuamente, ambas precisam de um número IP exclusivo e um identificador de rede (o chamado ESSID) também único. Digite um valor para o `SSID_wlan0` em seu `/etc/wlan/wlan.conf` para resolver o problema. Esse ajuste está na linha 80, aproximadamente. Por exemplo

```
SSID_wlan0="zedocaixao"
```

atribui o nome *zedocaixao* para esse adaptador.

Na etapa seguinte, renomeie o arquivo `/etc/wlan/wlancfg-DEFAULT` para refletir o nome da interface (em nosso exemplo, seria `wlancfg-zedocaixao`) e mude a linha 31 de `IS_ADHOC=n` para `IS_ADHOC=y`, ativando assim o modo *ad-hoc*. Para iniciar o serviço de rede sem fio, digite `/etc/init.d/wlan start`. Se tudo correr bem, será possível associar um endereço IP ao adaptador com o comando `ifconfig wlan0 192.168.1.1 netmask 255.255.255.0`. Para testar, "pingue" a outra máquina (depois de configurada, obviamente) com o comando:

```
ping 192.168.1.2
```

Para usar um ponto de acesso, é preciso escolher o modo *Managed* em vez do *Ad-hoc*. Em outras palavras, restaure a linha alterada no arquivo `/etc/wlan/wlancfg-zedocaixao` novamente para `IS_ADHOC=n`. Como o ponto de acesso, estando em modo *managed*, atribui automaticamente o canal a ser usado, também é necessário ativar a varredura de canais para que o WLAN possa escolher um canal vago: atribua, no arquivo `/etc/wlan/wlan.conf`, o valor `Y` à variável `WLAN_SCAN`. A alteração fica assim:

```
WLAN_SCAN=y
```

Se você sabe quais canais seu ponto de acesso usa, apague os demais da lista de canais (`ChannelList`). Isso acelera a varredura.

Caçando problemas

Se o adaptador não conseguir se conectar na primeira tentativa, digite `iwconfig` para verificar se o sistema realmente detectou seu adaptador. O **quadro 2** mostra alguns exemplos do que pode sair disso. Se o comando não mostrar um dispositivo de rede sem fio, digite `/sbin/lsmmod | grep prism` para verificar se o módulo `prism2_usb` está mesmo carregado. Se não estiver, digite `modprobe prism2_usb` para carregá-lo manualmente.

As mensagens emitidas pelo kernel quando plugamos dispositivos USB também são bastante úteis. Para consultá-las,

torne-se `root` e digite `tail -f /var/log/messages` em um terminal. A opção `-f` coloca o comando `tail` em modo interativo, permitindo que vejamos as mensagens na hora em que surgem. Tente plugar e desplugar o chaveiro e veja o que acontece. As distribuições com o sistema *hotplug* ativado (como o Fedora, o Mandrake e o SuSE) devem carregar o módulo automaticamente a cada vez que o chaveiro é inserido na porta USB.

Também é uma boa pedida verificar as regras do firewall. O SuSE Linux 9.2 e posteriores criam regras de firewall automaticamente sempre que um dispositivo

de rede for ativado. Será preciso mexer nessas regras para que o novo dispositivo funcione. Para os testes iniciais, desabilite o firewall – é a maneira mais fácil e rápida de comprovar se o problema é com ele.

Velocidade total

O único dispositivo de alta velocidade (54 Mbit/s) que funciona com Linux no momento é o D-Link DWL-G122. Infelizmente, um driver oficial ainda não havia sido lançado até a data de publicação desta matéria. No laboratório da Linux Magazine, testamos uma versão beta do driver com o kernel 2.6.7. ➔

Tabela 1: Adaptadores USB WLAN testados

Nome	Fabricante	Chip	Módulo	Velocidade	Driver para Linux	Ponto de acesso	Ad-Hoc	Preço aproximado
MA111	Netgear	Prism2	prism2_usb	11 Mbit/s	✓	✓	✓	R\$ 215,00
DWL-122	D-Link	Prism2	prism2_usb	11 Mbit/s	✓	✓	✓	R\$ 240,00
DWL-G122	D-Link	Ralink	rt2570	54 Mbit/s	✓ ¹	✓	✗	R\$ 500,00
WL-U356	Planet	ZyDAS ZD1211	zd1211	54 Mbit/s	✓	✗	✗	R\$ 253,00 ²
WG111	Netgear	Prism Frisbee	?	54 Mbit/s	✗	✗	✗	R\$ 105,00 ²
C54RU	Conceptronic	Intersil/Prism	?	54 Mbit/s	✗	✗	✗	R\$ 103,00 ²
SMC2862W-G	SMC Networks	Prism Frisbee	prism54u	54 Mbit/s	✓	✗	✗	R\$ 105,00 ²

¹ Não há drivers oficiais para Linux.

² Produtos não disponíveis no mercado brasileiro; preços aproximados, baseados no preço médio no mercado norte-americano, com cotação do dólar em 25/02/05.

É preciso compilar o driver digitando `make -C /caminho/para/os/fontes_do_kernel SUBDIRS=$PWD modules`. Depois, trabalhando como `root`, carregue o módulo na memória com o comando `insmod rt2570.ko`. Por fim, habilite o dispositivo WLAN com `ifconfig rausb0 up`. Será preciso configurar um endereço IP para a interface, que pode ser estático ou, se você possuir um servidor de DHCP, dinâmico. De acordo com a D-Link, o driver oficial para o kernel 2.6 já deve estar disponível no FTP da empresa no momento da publicação desta matéria. Infelizmente, a D-Link não pretende lançar uma versão para o kernel 2.4. Esperemos que os problemas em modo *ad-hoc* já estejam resolvidos no driver oficial.

Experimental

Os desenvolvedores e “fuçadores” do kernel talvez gostem do adaptador *WL-U356*, da Planet, e do *SMC2862W-G* da SMC Networks. Há um driver para Linux no site oficial da empresa [4] para o adaptador da Planet, que compila tanto no kernel 2.4 quanto no 2.6. Depois de compilar o driver com o comando `make`, carregue o módulo `zd1211` manualmente na memória (como `root`, obviamente) e habilite a interface com `ifconfig eth1 up`. O comando `iwconfig` deve, agora, mostrar a interface. Em nosso laboratório, isso foi tudo o que conseguimos: o dispositivo está lá, mas qualquer tentativa de fazê-lo falar com o ponto de acesso mostrou-se infrutífera. O problema não pôde ser resolvido porque o comando `iwconfig` não nos deixou configurar nenhum dos parâmetros da placa, nem mesmo o `ESSID`. A Linux Magazine tentou entrar em contato com o desenvolvedor do driver, mas não obteve nenhuma resposta até a publicação desta matéria.

O adaptador da SMC é outra tarefa digna de Hércules. Os drivers para o dispositivo, baseado no chip Prism Frisbee, ainda estão em desenvolvimento [5], bem como o *firmware*. Depois de descompactar os fontes siga a mesma receita que usamos para a *DWL-G122*:

```
make -C /path/to/your/kernel/sources \
SUBDIRS=$PWD modules
insmod prism54u.ko
```

Emita os comandos como usuário `root`. Assim como no adaptador da Planet, empacamos depois de uma pequena vitória. O adaptador é detectado ao se carregar o módulo `prism54u`, mas fica mudo.

Quadro 2: Exemplos do iwconfig

sem criptografia em modo *ad-hoc*

```
suse92:/home/marcel # iwconfig
lo        no wireless extensions.
eth0      no wireless extensions.
wlan0     IEEE 802.11-DS  ESSID:"zedocaixao"  Nickname:"suse92"
          Mode:Ad-Hoc  Frequency:2.442 GHz  Cell: 02:0F:DA:B5:C1:3E
          Bit Rate:2 Mb/s  Tx-Power:2346 dBm
          Retry min limit:8  RTS thr:off  Fragment thr:off
          Encryption key:off
          Link Quality=0/92  Signal level=-100 dBm  Noise level=-100 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0
sit0      no wireless extensions.
```

com criptografia em modo *managed*

```
client:/home/otto # iwconfig
lo        no wireless extensions.
eth0      no wireless extensions.
wlan0     IEEE 802.11-b  ESSID:"zedocaixao"  Nickname:"wireless"
          Mode:Managed  Frequency:2.462 GHz  Access Point: 00:09:5B:2D:B8:98
          Bit Rate:11 Mb/s  Tx-Power:2346 dBm
          Retry min limit:8  RTS thr:off  Fragment thr:off
          Encryption key:3BFA-1658-F26F-65BB-1015-0222-49  Security mode:open
          Link Quality=92/92  Signal level=-40 dBm  Noise level=-95 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0
sit0      no wireless extensions.
```

Fechando os buracos

Não há nada de errado em usar uma rede sem fio não-criptografada para fins de teste. Entretanto, em ambiente de produção, uma chave WEP é imperativa. Para configurar adaptadores com o chipset Prims2, edite o arquivo `/etc/wlan/wlancfg-ESSID`, sendo `ESSID` o identificador da interface. Use a variável `AuthType` para especificar se o `linux-wlan-ng` usará – ou não – uma *chave pública*. Entretanto, como esse método é mais propenso a sofrer ataques, talvez você queira usar o modo `Open`. A versão `0.2.1-pre22` das `wireless-tools` causou alguma confusão em nossos testes. Mesmo modificando a variável, o `iwconfig` insistia em mostrar `Security mode:restricted` em vez de `Open`. Testes com as chaves públicas e o modo `Open` demonstraram, entretanto, que o chaveiro USB não estava simplesmente ignorando a variável – portanto, o problema não era com o hardware.

Na seção `#WEP` também há algumas alterações a fazer, como mostra o **quadro 3**. As primeiras duas variáveis

permitem especificar se o cliente usará a criptografia do hardware (valor `false`) ou a baseada no driver do `linux-wlan-ng`. Como afirma a documentação, o software é muito mais rápido que o hardware nesse caso, portanto altere essa variável para `true`. A linha `dot11PrivacyInvoked=true` ativa a criptografia.

Na próxima linha, especifique a chave de criptografia a ser usada pelo programa. Um máximo de quatro chaves criptográficas podem ser associadas com outras redes sem fio. Para evitar que a máquina aceite dados não criptografados vindos de outros computadores, ajuste a variável `dot11ExcludeUnencrypted` para `true`; mas atenção: esse arranjo só funciona com chaves compartilhadas.

A segunda seção indica o mecanismo de geração de chaves que o `linux-wlan-ng` vai usar. Por padrão o `make install` instala a ferramenta `nwepgen`. O subdiretório `add-ons` abaixo do diretório dos fontes do `linux-wlan-ng` também possui as ferramentas `lwepgen` e `keygen`, que você pode compilar (digite `make` para isso) e

Quadro 3: Configurando a criptografia WEP

```
#=====WEP=====
# [Habilita/Desabilita] o WEP. Os valores só são relevantes
# se a variável PrivacyInvoked é ajustada para o valor "true"
lnxreq_hostWPEncrypt=true      # true|false
lnxreq_hostWEPDecrypt=true     # true|false
dot11PrivacyInvoked=true       # true|false
dot11WEPDefaultKeyID=0        # 0|1|2|3
dot11ExcludeUnencrypted=true   # true|false, em AP significa que o WEP é obrigatório.
# Se PRIV_GENSTR não estiver vazio, use PRIV_GENTSTR para gerar
# chaves (não é obrigatório, mas é bem conveniente).
# O diretório add-ons/ no código fonte contém outros geradores.
PRIV_GENERATOR=/sbin/nwepgen   # nwepgen, Compatível com o Nessus
PRIV_KEY128=true               # Tamanho em bits da chave a ser gerada
PRIV_GENSTR=""
# Ou configure as chaves explicitamente.
# Use genstr ou chaves, *nunca* ambos ao mesmo tempo.
dot11WEPDefaultKey0=3B:FA:16:58:F2:6F:65:BB:10:15:02:22:49
dot11WEPDefaultKey1=          # xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
dot11WEPDefaultKey2=          # exemplo: 01:20:03:40:05 ou
dot11WEPDefaultKey3=          # 01:02:03:04:05:06:07:08:09:0a:0b:0c:0d
```

copiar para `/sbin/` ou `/usr/local/sbin/`. Ambas as ferramentas podem gerar chaves WEP; entretanto, não são compatíveis com todos os dispositivos.

Se você pretende usar uma dessas ferramentas, deve fornecer uma senha em `PRIV_GENSTR`, que será usada como semente para a geração da chave. Entretanto, o mais comum é criar uma senha no ponto de acesso e simplesmente digitar a chave resultante em `dot11WEPDefaultKey0`. Observe que é preciso digitar a chave como mostrado no exemplo, não em grupos de quatro dígitos separados por pontos, vírgulas ou traços.

Finalmente, `PRIV_KEY128` permite escolher entre chaves de 64 ou 128 bits. Apesar de mais moderno e seguro, alguns dispositivos não reconhecem as chaves de 128 bits. Todos os dispositivos testados para este artigo, entretanto, reconheceram a criptografia de 128 bits sem problemas.

Novamente, o SuSE Linux retira das costas do usuário o peso dessa configuração toda. Basta, simplesmente, selecionar o tipo de chave que deve ser usada (`Passphrase`, `ASCII` ou `Hexadecimal`) e o tipo de chave criptográfica na caixa de texto apropriada. As ferramentas que mencionamos anteriormente também podem ajudar, caso o usuário queira usar uma chave hexadecimal. No SuSE Linux, essas ferramentas fazem parte do pacote `wireless-tools`.

Logout

Apesar dos fabricantes de chaveiros WLAN possuírem uma gama muito grande de produtos no mercado – muitos deles com taxas de transferência de até 54 Mbit/s – a maioria dos usuários de Linux ainda estão presos às já relativamente obsoletas interfaces de 11 Mbits/s baseadas no chip Prism2. Essa velocidade é mais que suficiente para navegar na Internet, mesmo numa conexão DSL de alta velocidade, mas numa rede local pode atrapalhar bastante o fluxo de dados. Não perca as esperanças: o DWL-G122 da D-Link é um adaptador USB WLAN que funciona a 54 Mbit/s no Linux, tendo drivers bastante bons. Fique antenado no site dos fabricantes (e na sua Linux Magazine Brasil) para mais notícias sobre o assunto. ■

INFORMAÇÕES

- | | |
|-----|---|
| [1] | <p>Suporte a redes sem fio no Linux:
 http://www.linux-wlan.org/docs/wlan_adapters.html.gz</p> |
| [2] | <p>Projeto Linux-wlan-ng:
 http://www.linux-wlan.org</p> |
| [3] | <p>Wireless-tools:
 http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html</p> |
| [4] | <p>Driver para o adaptador Planet:
 http://www.planet.com.tw/download.php</p> |
| [5] | <p>Driver para o chipset Prism54 USB:
 http://jbnote.free.fr/prism54usb/</p> |