



O cliente multi-rede MLDonkey dá acesso a todas as principais redes de compartilhamento de arquivos. Mostraremos aqui como configurá-lo e como controlá-lo com a ajuda do KMLDonkey, um cliente para o KDE.

POR MIRKO ALBRECHT

Trocando arquivos com o MLDonkey e o KMLDonkey

# Bom pra burro

O compartilhamento de arquivos em redes ponto-a-ponto (*peer-to-peer* ou *p2p*, em inglês) é um dos assuntos mais controversos de nosso tempo. Usuários que colocam toda sua coleção de CDs e DVDs em seus discos rígidos e convidam outros usuários para baixá-los estão literalmente metendo a mão nos bolsos das indústrias de música e de filmes; porém, quando as plataformas de compartilhamento são usadas legalmente, são um excelente modo de tornar os arquivos disponíveis para o mundo sem necessidade de pagar por espaço remoto na web.

Alguns dos principais exemplos de redes de compartilhamento de arquivos são Morpheus, Gnutella, BitTorrent e Donkey/eMule. Os clientes atuais, que podem lidar com múltiplas redes ponto-a-ponto, oferecem ao usuário um meio de acesso mais prático. Um desses clientes é o MLDonkey, disponível para Linux, Windows e Mac OS.

## O núcleo

O MLDonkey roda como um núcleo (*core*, em inglês), parecido com um *daemon* em segundo plano. Algumas distribuições incluem o núcleo e um *front-end* gráfico que controla o cliente em modo texto. Outras, como o SUSE Linux 9.2,

simplesmente incluem a interface gráfica do KDE KMLDonkey. O Debian Sarge possui ambas ao alcance de um `apt-get`.

A experiência ensina que é melhor compilar sua própria versão do que usar o programa fornecido por seu distribuidor. O mais bravo entre os bravos vai desejar compilar o código-fonte mais recente para manter-se em condições de igualdade com o processo de desenvolvimento do MLDonkey, que dá saltos bastante dramáticos e por vezes se abala. Os desenvolvedores dessa tecnologia de compartilhamento de arquivos estão constantemente aperfeiçoando os protocolos e as configurações internas. Também é sensato manter-se em dia com as mudanças em cada uma das redes ponto-a-ponto, que usam seus próprios protocolos.

Dito isso, a maioria dos usuários vai preferir uma versão estável; repetimos, a experiência mostra que algumas atualizações não são nada perfeitas. Por isso, se você tiver uma versão estável do MLDonkey, a ordem é ficar longe de atualizações duvidosas que podem facilmente afetar a funcionalidade de seu núcleo.

## Instalando binários

Se você prefere evitar ter de compilar o código, pode baixar os pacotes binários da homepage oficial em [1]. A página do

Spiralvoice [2], outra excelente fonte de downloads, tem um atualizador automático e um compilador de pacotes para o MLDonkey. O site tem uma grande variedade de versões, incluindo todos patches necessários. Quando esta revista foi impressa, a versão oficial mais recente era o MLDonkey 2.5.28 e a mais nova versão não-oficial era a 2.5.29.

Os fóruns do MLDonkey (como o em [3]) vão informá-lo de que a versão que tem a reputação de máxima estabilidade em operações 24x7 e a melhor taxa de download é o núcleo 2.5.16t. O “t” no final do número da versão identifica o lançamento como uma versão corrigida. O próprio Spiralvoice afirma que essa versão é “bastante estável” e se refere à “natureza experimental” da versão atual [2]. Testei uma porção de versões e tudo o que posso dizer é: concordo.

Para instalar o MLDonkey, descompacte o pacote em um diretório digitando o comando `tar xvzf mldonkey-2.5.x.tar.gz`. Dentro do diretório criado durante a descompactação existe um arquivo chamado *mlnet*, que pode ser executado diretamente com o comando `./mlnet`.

Em geral, as versões pré-compiladas que o Spiralvoice oferece funcionam bastante bem. Essas versões têm suporte modular para as principais redes ponto-

a-ponto. O código pode ser mais atulhado, mas isso não deverá afetar um computador moderno. Além disso, você pode usar o front-end gráfico para desabilitar todos os módulos de que você não precisa num estágio posterior.

## Dando à luz um burro

Compilar o MLDonkey a partir dos fontes resulta em uma versão do programa que combina melhor com seu ambiente de hardware e aplicativos preferidos. Repetimos: *não é necessário compilar a partir dos fontes*, você pode instalar o binário de sua distribuição se quiser.

Opõe pela versão mencionada anteriormente, `mldonkey-2.5.16`, com o patch `patch_pack16t`. Ambos os arquivos estão disponíveis para download em [2] e basta seguir as etapas normais para descompactá-los. Mova os conteúdos do pacote do patch para o diretório que você criou ao descompactar o arquivo tar principal, `/mldonkey-2.5.16`. Vá então para esse diretório e digite `patch -p0 < patch_pack16t` para instalar o patch.

Será preciso configurar o arquivo `makefile` no mesmo diretório. Decidimos restringir o número de redes que o MLDonkey pode acessar, liberando apenas BitTorrent, Fasttrack e eDonkey2000. Não usaremos a interface gráfica GTK por enquanto. O comando `configure` que permite fazer isso é:

```
./configure --disable-gui 2
--enable-batch 2
--disable-opennap2
--disable-audiogalaxy 2
--disable-gnutella 2
--disable-gnutella2 2
--disable-soulseek 2
--disable-openft 2
--disable-cymes 2
--enable-pthread
```

Se você tiver preferências diferentes, pode mudar `--disable` para `--enable` conforme necessário. Por exemplo, se você só precisar acessar a rede eDonkey2000, basta chamar `./configure --disable-multinet`, sem esquecer de especificar suas outras preferências.

Este artigo se concentrará nas redes eDonkey, BitTorrent e Fasttrack (Kzazaa), que são as mais populares e difundidas entre a comunidade que compartilha arquivos. Mesmo a simples menção de

todas as redes ponto-a-ponto e suas peculiaridades tomariam mais espaço do que temos para este artigo. Mas se você estiver interessado em aprender mais, sempre pode conferir as FAQs (*Perguntas Feitas Frequentemente*) do MLDonkey em [4]. Isso ajuda muito com os diversos módulos de rede implementados pelo MLDonkey.

Para compilar o MLDonkey, você precisa da ferramenta *Objective Caml* (essa é a linguagem de programação em que o programa foi escrito) versão 3.08 e da *LablGTK* 1.2.7. É possível usar o utilitário `wget` para baixar os pacotes necessários da internet durante a fase `./configure` e compilar os pacotes. Porém, a aplicação do `patch_pack16t` significa que você precisa de versões mais novas de ambas as ferramentas. Para desabilitar a instalação automática delas, remova a opção `--enable-batch` do comando `configure`. De qualquer forma, se você optar por desabilitar a interface gráfica (`--disable-gui`), não precisará atualizar o *LablGTK*.

Para compilar o *OCaml* e o *LablGTK* antes de começar, primeiro baixe os fontes de [5] e [6] e descompacte-os. A compilação do *Objective Caml* é um pouco diferente da típica compilação em três fases:

```
./configure
make world opt opt.opt
make install
```

A sintaxe do *LablGTK* também se afasta das normas. Na verdade, são precisos quatro passos:

```
make configure
make
make opt
make install
```

Em ambos os casos, você precisa de privilégios de administrador para rodar o `make install`. Se você tiver o SUSE Linux 9.2, não precisa passar pelo aborrecimento de compilar essas duas ferramentas, já que as distribuições as incluem em versões recentes. Você pode rodar o *YaST* para instalar as ferramentas, se é que ainda não o fez.

Após passar por essa barreira, você deve conseguir rodar `configure` no diretório `/mldonkey-2.5.16` sem erros. Se você tiver algum problema ao compilar o MLDonkey, pode deixar o patch de fora ou usar a variante `patch_pack16r` no lugar do `16t`. Os

comandos `make depend` e `make` fornecem o executável; não há `make install`. Para rodar o MLDonkey como usuário não-privilegiado, basta rodar o núcleo `mlnet`, que deve ter entre 3 e 3,5 MBytes.

Para rodar o MLDonkey a partir de qualquer lugar em seu sistema, você pode copiar o `mlnet` para seu diretório global de binários digitando `cp mlnet /usr/local/bin`. Obviamente, você precisa de poderes de administrador (*root*) para fazê-lo. Em seguida torne o arquivo executável por todos os usuários digitando `chmod a+x /usr/local/bin/mlnet` ou atribua os privilégios de usuário apropriados de acordo com seu sistema.

## Iniciando o núcleo

Agora, qualquer usuário pode iniciar o núcleo do MLDonkey numa janela de terminal digitando `mlnet`. Se você não pôs o núcleo em seu `path`, vá para o diretório em que se encontra o binário `mlnet` e digite `./mlnet`.

Na primeira vez em que você chama o programa, ele cria um punhado de arquivos e diretórios em sua pasta pessoal. Para evitar essa bagunça e manter os arquivos num lugar central, ajuste uma variável de ambiente. Para isso, abra o arquivo `.bashrc` em seu diretório pessoal e acrescente uma linha com o comando `export MLDONKEY_DIR="/caminho/para/diretorio_mldonkey"` (ex: `/home/usuario/mldonkey`), certificando-se de ter criado o diretório antes disso. Feche a sessão do console e inicie-a novamente para aplicar as mudanças.

Ao digitar `mlnet`, as seguintes linhas devem aparecer em sua tela, além de um bom punhado de outras informações:

```
Welcome to MLDonkey
SAVING SHARED FILES AND SOURCES2
on localtime: 12/ 9, 17:07:33
SAVED
Options correctly saved
Core started on localtime: 2
12/ 9, 17:07:33
Disabling output to console, 2
to enable: stdout true
```

Você pode pressionar `[Ctrl]+[C]` para sair do programa; O MLDonkey também desliga quando a janela em que o núcleo foi iniciado é fechada. Para acabar com a necessidade de manter uma janela aberta ao rodar o `mlnet`, pode-se

executar o programa com o parâmetro `-daemon`. Dessa forma, o núcleo vai rodar como um daemon em segundo plano; os usuários podem até sair da sessão (“deslogar”) sem que o `mldnet` seja desativado. Para interromper o processo em segundo plano, digite `killall mldnet`.

## Gerenciando o núcleo

Agora que o núcleo do *MLDonkey* está rodando em segundo plano, há muitos modos de acessar o `daemon`. Se você prefere uma abordagem espartana, pode usar até o *Telnet*. Digite `telnet localhost 4000` para acessar a interface do *MLDonkey* em modo texto.

Digitar um ponto de interrogação no prompt do *Telnet* seguido do comando `s termo_a_buscar` iniciará uma busca, o comando `vr` exibirá os resultados e `d numero_resultado` iniciará um download. O comando `q` sai da interface. Digitar dois pontos de interrogação exibe uma ajuda mais detalhada; é sensato dar uma conferida nisso se sua intenção é continuar gerenciando o *MLDonkey* através da interface *Telnet*. Como alternativa, você pode abrir a interface web do *MLDonkey* digitando o endereço <http://localhost:4080> em seu navegador.

Se esses dois métodos não forem de seu agrado, você pode optar por um dos muitos *front-ends* com interface gráfica. A versão pré-compilada tem uma útil interface escrita em GTK, como mencionado anteriormente. Ela é bem-estruturada, intuitiva e deixa muito pouco a desejar. Usuários do Gnome provavelmente preferirão essa interface.

Se você preferir gerenciar o *MLDonkey* com seu navegador web, dê uma olhada no *phpEselGui*, uma útil interface escrita em PHP. Outra interface popular, *Zuul*, infelizmente não é mais compatível com as versões atuais do *MLDonkey*.

## KMLDonkey

Já há algum tempo existe um *front-end* do *MLDonkey* para o ambiente gráfico KDE, o *KMLDonkey*, que tornou-se um programa realmente útil durante os últimos meses. Você pode adicionar o *KMLDonkey* ao painel do KDE e atribuir a ele o tratamento dos links *ed2k* (links para downloads de arquivos via *eDonkey*). O *KMLDonkey* também tem uma interface logicamente estruturada, auto-explicativa na maior parte.

Usuários do SUSE descobrirão uma versão atual dos binários do *KMLDonkey* para o SUSE Linux 9.1 e 9.2 na página do Packman [7]. A versão oferecida com a atual distribuição SUSE 9.2 é o velho *kml-donkey-0.9.1*. Usuários do Debian encontram pacotes pré-compilados em seu repositório APT favorito.

A versão mais atual do *KMLDonkey* quando esta revista foi à gráfica, o *kml-donkey-0.10*, está disponível para download em [8]. Compilar o aplicativo a partir dos fontes é rápido e indolor. Para fazê-lo, baixe o pacote `.tar` com o código e em seguida compile e instale o programa com os seguintes comandos:

```
tar jxvf kml-donkey-0.10.tar.bz2
cd kml-donkey-0.10
./configure --prefix=$KDEDIR
make
su -c "make install"<I>
```

O diretório do KDE (`$KDEDIR`) no SUSE Linux é `/opt/kde3`, por isso o comando seria `./configure --prefix=/opt/kde3`. O link para o programa recém-compilado fica no item *Internet* do menu do KDE. Você pode pressionar **[Alt]+[F2]** e digitar `kml-donkey` para chamar o aplicativo diretamente, supondo-se que o núcleo do *MLDonkey* já esteja rodando.

## Configuração

Após iniciar o *KMLDonkey*, a primeira coisa a fazer é configurar a interface do programa na última aba, *Terminal (Console)*. Na parte de baixo da aba está o terminal, que pode ser usado para se definir uma senha de acesso ao núcleo – até agora, qualquer um poderia ter acessado seu núcleo usando *Telnet* ou *HTML*. Por exemplo, para configurar como senha a palavra `53Nh4`, digite o seguinte no prompt do terminal: `add_user admin 53Nh4`

Será preciso fazer mais alguns ajustes no menu *Configurações | Configurar Conexão (Settings | Configure Connection)* e *Configurações | Configurar KMLDonkey (Settings | Configure MLDonkey)*. No primeiro submenu, digite a senha e as credenciais do usuário que você acaba de estabelecer. Também se pode configurar o *modo de início* do *KMLDonkey* para rodar o núcleo no momento em que é executado. Isso lhe poupará o trabalho de iniciar o núcleo manualmente, como descrito acima.

O segundo menu traz as opções das diversas redes de compartilhamento de arquivos reconhecidas pelo *MLDonkeyv*. A área *Redes (Networks)* permite habilitar e configurar módulos individuais. Você pode atribuir um valor `false` a uma rede para evitar usá-la. Se precisar habilitar uma rede, como a *eDonkey2000*, selecione `true`.

## eDonkey 2000

O *KMLDonkey* já vem com configurações padrão muito úteis para a maioria dos protocolos de rede ponto-a-ponto. Entretanto, os valores para a rede *eDonkey2000* precisam receber alguma atenção extra. Vá ao item `force_high_id` no painel da direita em *Redes | Donkey (Network | Donkey)* e ajuste o valor para `true`. Isso manda o cliente deixar de lado as conexões com servidores que só suportam IDs baixos e procurar os que permitam IDs altos.

Um ID baixo não vai levar você muito longe numa rede de compartilhamento de arquivos. O ID baixo significa mais trabalho administrativo para o servidor *eDonkey*, já que as portas específicas dele, `4661/tcp`, `4662/tcp` e `4665/udp`, estão bloqueadas por um firewall ou não estão sendo roteadas por um roteador. Você precisa liberar essas portas em seu firewall pessoal (ou seja, aquele instalado no próprio PC do usuário) para que seu núcleo possa conseguir um ID alto. No *SUSE-Firewall2*, adicione essas portas em *Services | Other services (Serviços | Serviços Adicionais, figura 1)*. Para outras distribuições consulte a documentação apropriada. Se o seu sistema *MLDonkey* estiver se escondendo por trás de um roteador – que, muitas vezes, age como firewall – saiba que muitos dispositivos têm uma conveniente interface web que permitirá redirecionar conexões por essas portas para sua máquina (o item é tipicamente chamado *Virtual Servers, Port Redirection* ou algo semelhante). Confira a documentação de seu roteador para saber mais.

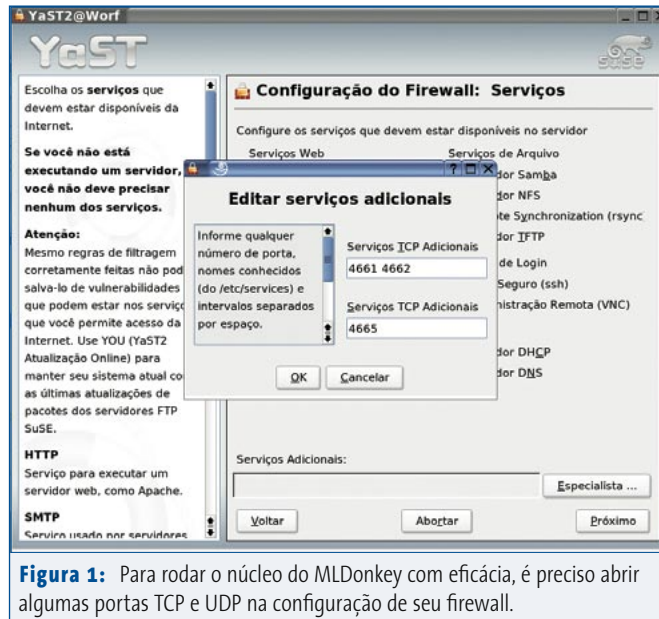
Agora o núcleo deve lhe atribuir um ID alto; confira se isso é verdade usando um comando simples no console do *MLDonkey*: `id`. Se o resultado for seu próprio endereço IP, a atribuição de um ID alto funcionou, o que significa que você não ficará mais no final da lista de clientes.

## Sintonia Fina

Alguns usuários sugerem configurar o valor de `max_connected_servers` para o mais alto possível. Isso significa que alguns arquivos são baixados mais rapidamente e – ao menos no início – as buscas são atendidas mais generosamente. Porém, isso sobrecarrega a rede e alguns usuários e servidores podem não gostar nada disso. Cada vez que você se conecta em um servidor, ocupa uma das “vagas” disponíveis nele – e essas vagas são limitadas. Cada vaga que você ocupar é uma a menos para os outros usuários. Porém, um valor maior do que 3 (o padrão) não vai garantir downloads mais rápidos; por isso é melhor deixá-lo como está.

O item `Bandwidth` permite especificar a largura máxima de banda reservada para upload e download. A `max_hard_upload_rate` não deve ser menor que 5 kbits/s para conexões DSL nem acima de 3 kbit/s para links ISDN. A razão para isso é que escolher uma taxa de upload demasiado alta afetará enormemente sua taxa de download. Por exemplo, se um upload em uma conexão DSL de 1000 kbit/s atinge o limite de upload de 128 kbit/s, sua conexão provavelmente será bloqueada e mesmo o acesso normal à Internet vai ficar muito mais lento. Você precisa ajustar sua taxa de download para refletir seu ambiente; afinal de contas, você conhece sua conexão à Internet melhor do que qualquer outra pessoa. Um valor de 0 aqui significa *ilimitado*. Uma regriinha comum é configurar a velocidade de download como sendo quatro vezes a de upload. Por exemplo, upload a 5 Kb/s e download a 20 Kb/s.

Já `Path` permite especificar onde arquivos temporários e downloads serão armazenados. Com isso concluímos as principais configurações; assim, não há nada que o impeça de começar a trocar arquivos pela primeira vez. Claro, o núcleo do *MLDonkey* está cheio de botões e alavancas que você pode experimentar segundo sua curiosidade. Para maio-



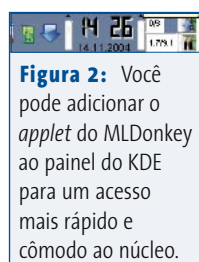
**Figura 1:** Para rodar o núcleo do *MLDonkey* com eficácia, é preciso abrir algumas portas TCP e UDP na configuração de seu firewall.

res informações, confira os fóruns online sobre o *MLDonkey*, por exemplo o que está em [3].

## Downloads com o KMLDonkey

A interface do programa é extremamente intuitiva e distribui as funções do programa em uma coleção de abas. `Servidores` (*Servers*) leva a uma lista dos servidores de compartilhamento de arquivos e permite abrir ou fechar conexões a eles. `Pesquisa` (*Search*) permite buscar arquivos em diversas redes; você pode especificar o tipo do arquivo e restringir o tamanho dele. Também é boa idéia dar uma olhada na aba `Estatísticas` (*Statistics*) de vez em quando. A aba dá informações interessantes sobre o desempenho de download de seu cliente.

Um dos mais interessantes recursos do *KMLDonkey* é que você pode integrar o programa ao seu painel do KDE. Para fazê-lo, clique com o botão direito em uma área vazia do Kicker (a “barra de tarefas” do KDE) e selecione `Adicionar | Aplicativo | MLDonkey Applet` (`Add | Applet | MLDonkey Applet`) para adicionar o programa. Você também pode configurar alguns parâmetros críticos do núcleo aqui (**figura 2**), incluindo as taxas máximas de download e upload. O *applet* mostra as taxas atuais para que você possa ter uma visão geral. Se a taxa de download estiver demasiado baixa para as tarefas do momento, pressione o botão de pausa para



**Figura 2:** Você pode adicionar o *applet* do *MLDonkey* ao painel do KDE para um acesso mais rápido e cômodo ao núcleo.

dizer ao núcleo do *MLDonkey* que aguarde até você clicar em *Continuar* (*Play*) novamente.

Como anteriormente mencionado, o navegador Konqueror agora reconhece a URI `ed2k` como um protocolo válido. Se você tropeçar em um link desse tipo na Internet, o navegador acrescentará o download ao núcleo atual com um clique do mouse.

## Conclusão

Após toda a negociação da fase de instalação, a troca de arquivos numa rede de compartilhamento é muito fácil. Graças a interfaces gráficas bem pensadas como o *KMLDonkey*,

qualquer pessoa pode oferecer arquivos à comunidade e acessar os downloads oferecidos por outras pessoas.

Mesmo assim, o pacote *MLDonkey* poderia ter alguns detalhes um pouco mais bem aparados. Algo que eu gostaria de ver seria uma instalação automática baseada em RPM, que removeria a necessidade de copiar o aplicativo para a posição correta na árvore de diretórios.

E esperamos que a indústria do entretenimento não acabe com a comunidade de compartilhamento com essa sua atual postura de criminalizar qualquer tipo de troca de arquivos. ■

## INFORMAÇÕES

- [1] Homepage do *MLDonkey*: <http://www.mldonkey.org>
- [2] Binários para *MLDonkey* da Spiralvoice: <http://download.berlios.de/pub/mldonkey/spiralvoice/>
- [3] Fórum do *MLDonkey*: <http://mldonkey.berlios.de/modules.php?name=Forums>
- [4] FAQ do *MLDonkey*: <http://www.nongnu.org/mldonkey/faq.html>
- [5] Objective Caml: <http://caml.inria.fr/ocaml/>
- [6] Código-fonte do *lablGTK*: <http://wwwfun.kurims.kyoto-u.ac.jp/soft/olabl/lablgtk.html>
- [7] Binários do *KMLDonkey* para SUSE: <http://packman.links2linux.org/?action=451>
- [8] Código-fonte do *KMLDonkey*: <http://savannah.nongnu.org/download/kmldonkey/kmldonkey-0.10.tar.bz2>