LINUX MAGAZINE

LINUX Pocket Pro

LINUX NEW MEDIA
The Pulse of Open Source

02

Passo a passo
Do cabeamento à configuração de servidores

Trabalho Zero
Crie redes autoconfiguráveis com ZeroConf e Avahi

100 páginas de informação
Tutoriais e técnicas para entrar no mundo da administração de redes Linux

Administração de Redes

Guia ilustrado
Veja como é fácil montar cabos de rede

DHCP:
Centralize a distribuição de endereços com Linux e dê adeus aos conflitos de IPs

09/2007

LINUX Pocket Pro

Administração de Redes

São Paulo | 2007

Expediente

Diretor Geral
Rafael Peregrino da Silva
peregrino@linuxmagazine.com.br

Editor-chefe
Tadeu Carmona
tcarmona@linuxmagazine.com.br

Editor
Pablo Hess
phess@linuxmagazine.com.br

Revisão
Arali Lobo Gomes
agomes@linuxmagazine.com.br

Editor de Arte
Renan Herrera
rherrera@linuxmagazine.com.br

Assistente de Arte
Igor Daurício
isilva@linuxmagazine.com.br

Centros de Competência

Centro de Competência em Software:
Oliver Frommel: ofrommel@linuxnewmedia.de
Kristian Kißling: kkissling@linuxnewmedia.de
Peter Kreussel: pkreussel@linuxnewmedia.de
Marcel Hilzinger: hilzinger@linuxnewmedia.de
Andrea Müller: amueller@linuxnewmedia.de

Centro de Competência em Redes e Segurança:

Achim Leitner: aleitner@linuxnewmedia.de
Jens-Christoph B.: jbrendel@linuxnewmedia.de
Hans-Georg Eßer: hgesser@linuxnewmedia.de
Thomas L.: tleichtenstern@linuxnewmedia.de
Max Werner: mwerner@linuxnewmedia.de

Assinaturas:
www.linuxnewmedia.com.br
assinaturas@linuxmagazine.com.br

Na Internet:
www.linuxmagazine.com.br – Brasil
www.linux-magazin.de – Alemanha
www.linux-magazine.com – Portal Mundial
www.linuxmagazine.com.au – Austrália
www.linux-magazine.ca – Canadá
www.linux-magazine.es – Espanha
www.linux-magazine.pl – Polônia
www.linux-magazine.co.uk – Reino Unido
www.linux-magazin.ro – Romênia

Gerente de Circulação:
Cláudio Guilherme dos Santos
csantos@linuxmagazine.com.br

Linux Magazine é publicada mensalmente por:
Linux New Media do Brasil Editora Ltda.

Av. Fagundes Filho, 134
Conj. 53 – Saúde
04304-000 – São Paulo – SP – Brasil
Tel.: +55 (0)11 4082 1300
Fax: +55 (0)11 4082 1302

Distribuição: Distmag
Impressão e Acabamento: Parma

ISBN 978-85-61024-05-5 Impresso no Brasil

Editorial

As redes de computadores já são tecnologia corrente a cerca de 40 anos – desde a época em que os grandes mainframes IBM e os terminais burros eram a tecnologia mais disseminada. De 40 anos para cá, muita coisa nova surgiu: os processadores em escala nanoscópica, os microcomputadores, a Internet, as redes de comunicação de alta velocidade, a telefonia por IP, a virtualização. Se pensarmos bem, todas essas invenções – revolucionárias, cada uma em seu contexto – deveram muito, ou dependeram, até certo grau, da disseminação e desenvolvimento das redes de computadores. Não é a toa que administrador de sistemas (sysadmin) passou, após um certo tempo, a ser sinônimo de administrador de redes.

Para esse administrador, o número de disciplinas e de conhecimentos necessários para executar bem seu trabalho cresceu, e cresce, em progressão sucessiva. Aos tradicionais conhecimentos sobre cabeamento, pilha TCP dos sistemas operacionais e servidores – esses últimos, divididos em tantos ramos quanto serviços e necessidades dos usuários existirem – juntaram-se conhecimentos sobre acesso remoto, configuração de dispositivos de rede, segurança... Até o modelo de mainframes amparados por redes de terminais burros renasce com força, sob um novo nome – os *thin clients* – e movimenta o mercado de hardware e as consultorias de TI e de projetos de rede.

Este volume, primeiro da **Coleção Linux Pocket Pro**, tem por objetivo fornecer ao profissional de redes – ou para quem pretende o ser – subsídios técnicos de alto nível. Já pensou uma rede sem trabalho braçal?

Bem-vindo ao mundo do conhecimento distribuído. Bem-vindo a Coleção Linux Pocket Pro.

Tadeu Carmona
Editor-chefe

Sumário

Capítulo 1 – Comunicações e sinais: conceitos gerais

As redes de comunicações são criadas para permitir a comunicação entre diversos dispositivos (estações de trabalho, impressoras, *storages*, etc) dentro de um determinado espaço físico (área ou segmento da rede). Essa comunicação se faz por meio de um *meio físico*, um vetor para a propagação de um *sinal*.

Em relação ao *meio físico*, os elementos constitutivos podem se comunicar utilizando qualquer material ou elemento físico que seja capaz de conduzir sinais elétricos – sem perdas de grande importância – sinais de rádio ou sinais infravermelhos. Para a transmissão de dados via sinais elétricos convencionou-se utilizar cabos, com uma grande variedade de tipos, velocidades e implementações disponíveis ao longo dos anos (para mais detalhes, veja o *Capítulo 3*).

Existem diversas normas técnicas que regulamentam a implementação de redes cabeadas. Essas normas são criadas e administradas, inclusive em suas alterações e correções, pelo IEEE (*Institute of Electrical and Electronics Engineers*), órgão localizado nos Estados Unidos. As normas desse padrão costumam ser referenciadas pela sigla IEEE, seguida da numeração da norma – numeração que costuma batizar, por vezes, os protocolos ou interfaces surgidos após as especificações da norma.

A primeira dessas normas, a **IEEE 802.1**, especifica interfaces de alto nível, arquitetura das redes, gerenciamento de redes, controle de acesso ao meio físico (*Medium Access Control* – MAC), dentre outras especificações.

A norma **802.2** define o *Controle de Enlace Lógico* (CEL ou *Logical Link Control* – LLC). Esses conceitos serão explicados ainda neste capítulo, e são muito importantes. Os padrões **802.3**, **802.4**, **802.5** e **802.6** definem, respectivamente, redes locais

com CSMA/CD, barramento com passagem de bastão, anel com passagem de bastão e redes metropolitanas.

Com o passar dos anos, diversos grupos de trabalho foram adicionados ao já extenso número das normas 802. A tabela a seguir (**tabela 1.1**) mostra a divisão dos principais grupos de trabalho da IEEE 802, suas responsabilidades e a maneira de consultar as respectivas documentações geradas até o momento. São incluídas aqui as normas de regulamentação para redes sem fio (Wi-Fi LAN) e redes sem fio de banda larga (*WiMAX*):

Grupo de trabalho	Aplicação	Documentação
IEEE 802.1	Protocolos LAN de alta camada	http://grouper.ieee.org/groups/802/1/
IEEE 802.2	Controle lógico do link	http://standards.ieee.org/getieee802/802.2.html
802.3	*Ethernet*	http://www.ieee802.org/3/
802.4	*Token bus* (grupo de trabalho descontinuado)	Não mais disponível.
802.5	*Token Ring*	http://www.ieee802.org/5/
802.6	*Man - Metropolitan Area Networks* (grupo de trabalho descontinuado)	Não mais disponível.
802.7	*Broadband LAN* (Rede de alto desempenho) utilizando cabeamento coaxial (grupo de trabalho descontinuado)	Não mais disponível.
802.8	*Fibra óptica TAG* (grupo de trabalho descontinuado)	Não mais disponível.
802.9	Serviços LAN Integrados (grupo de trabalho descontinuado)	Não mais disponível.
802.10	Segurança Interoperável em Redes LAN (grupo de trabalho descontinuado)	Não mais disponível.
802.11	*Wireless LAN* (Redes sem fio) Wi-Fi certification	http://standards.ieee.org/getieee802/802.11.html

802.12	Prioridade de demanda	Não disponível.
802.13	Não existe	Não existe.
802.14	*Cable modems* (grupo de trabalho descontinuado)	Não mais disponível.
802.15	*Wireless PAN*	http://standards.ieee.org/getieee802/802.15.html
802.15.1	*Bluetooth certification*	http://standards.ieee.org/getieee802/download/802.15.1-2005.pdf
802.15.4	*ZigBee certification*	http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf
802.16	*Broadband Wireless Access* ou Acesso a rede sem fio de Banda Larga, *WiMAX certification*	http://grouper.ieee.org/groups/802/16/
802.16e	(Mobile) *Broadband Wireless Access*. Padrão, ao lado do WiMAX, atualmente em discussão entre órgãos reguladores e concessionárias, aqui no Brasil.	http://standards.ieee.org/getieee802/download/802.16e-2005.pdf
802.17	*Resilient packet ring* ou Pacotes resilientes em anel	http://www.ieee802.org/17/
802.18	*Radio Regulatory TAG*	http://grouper.ieee.org/groups/802/18/
802.19	*Coexistence TAG*	http://grouper.ieee.org/groups/802/19/
802.20	*Mobile Broadband Wireless Access*	http://grouper.ieee.org/groups/802/20/
802.21	*Media Independent Handoff*	http://grouper.ieee.org/groups/802/21/
802.22	*Wireless Regional Area Network* (Redes wireless locais, de médio alcance)	http://grouper.ieee.org/groups/802/22/

Tabela 1.1: Grupos de trabalho do conjunto de normas 802, suas abrangências e respectivas fontes de documentação.

Transmissão a 100 Mbps

Entre os padrões de transmissão atualmente em uso, o mais em voga e com maior disseminação geográfica é o Ethernet 100. Esse padrão alcançou grande sucesso devido a sua flexibilidade, facilidade de implementação e alta disponibilidade.

O padrão Ethernet de 100 Mbps, contemplado pela norma 802.3, define três tipos de implementação, agrupadas sob o nome de Fast Ethernet, em oposição ao padrão *Ethernet* anterior, que contempla redes com transmissão máxima de 10 Mbps. As implementações abarcadas são: *100Base-TX*, para uso com par trançado sem blindagem; *100Base-FX*, para implementação de redes Ethernet com fibra óptica; e *100Base-T4*, para uso com par trançado de quatro pares de fios.

Transmissão a 1 Gbps

A flexibilidade do padrão 100Base-T permitiu um salto quantitativo em relação as velocidades suportadas. Esse salto, de dez vezes o valor da velocidade atual, gerou o padrão Gigabit Ethernet, regido pela norma IEEE 802.2z, iniciada em 1996.

Por seu parentesco com o padrão 100Base-T, a norma 1000Base-T propicia uma migração relativamente simples a partir das redes Ethernet e Fast Ethernet. Para a transmissão de sinais são aceitos cabos de categoria *5e*, o que permite o reaproveitamento de infraestrutura legada de cabeamento.

Apesar de os cabos utilizados serem os mesmos (Cat 5, Cat 5e ou superior) usados por redes Ethernet Base 100, o padrão faz uso intensivo da capacidade de transmissão. Por isso, detalhes como o comprimento da parte destrançada do cabo para encaixe do conector, nível de ruído do ambiente e comprimento das rotas de cabos são importantes.

Uma palavra sobre as redes 10 Gbps

Após a implementação e aceitação do padrão 1Gbps pelo mercado, tivemos o lançamento da tecnologia 10 Gigabit Ethernet, padronizada em 2002 com a norma *IEEE 802.3ae*.

Esse padrão que, como seu próprio nome diz, é capaz de transmitir dados e voz a velocidades de 10 Gigabits por segundo, exclui um algoritmo antes considerado importante – o CSMA/CD do subnível MAC (camada física). Isso foi feito porque o padrão 10 Gbps só é capaz de operar utilizando a topologia ponto a ponto. O seu modo de transmissão é somente Full-Duplex, utilizando cabeamento de fibra óptica.

Construir uma rede 10 Gbps, portanto, demanda não só a aquisição de interfaces, mas também de infraestrutura de cabeamento. As tecnologias e produtos para 10 Gigabit Ethernet são desenvolvidos por uma associação que conta com cerca de 80 membros, a 10GEA (*10 Gigabit Ethernet Alliance*).

Outros modelos

Existem outros modelos de transmissão de dados, incluindo transmissão sem fio e transmissão usando o novíssimo padrão WiMAX. Nossas atenções, todavia, se manterão sobre o padrão Ethernet, devido a sua popularidade e facilidade de manuseio, o que atende perfeitamente ao escopo deste guia.

Interface

O dispositivo responsável pela transmissão e recepção de sinais dentro do segmento de rede é a **interface**. A interface é a responsável pela recepção do sinal que passa por fios, no caso das redes por cabos, ou por ondas de rádio, no caso das redes Wi-fi, e pela sua correta decodificação em informações. Assim, ao falar de interface de rede, estamos falando não somente de especificações de hardware, obedecendo as normas RS (*Recommended Standard*), mas também da formulação lógica das conversações entre duas redes.

Em redes Ethernet convencionou-se chamar, de modo vernacular, o hardware de uma interface de rede de **placa de rede**. Apesar de existirem as formas adaptador de rede e *NIC* – ao nosso ver, esta a mais correta –, foi o primeiro formato que permaneceu no vocabulário dos sysadmins brasileiros, e é ele que utilizaremos ao lado de interface de rede.

Cada arquitetura de rede exige um tipo específico de placa de rede. Além da arquitetura usada, as placas de rede à venda no mercado diferenciam-se também pela taxa de transmissão, cabos de rede suportados e barramento de dados utilizado: PCI ou USB e PCMCIA de 16 ou 32 bits.

As normas estabelecem um tipo de interface específica para cada arquitetura de redes. Cada uma dessas arquiteturas deve trabalhar com o padrão correto de interface, aliado ao padrão correto de cabo, dimensionado para o desempenho do escopo de tarefas para o qual a rede foi desenhada. Nas antigas redes Ethernet, por exemplo, usavam-se placas Ethernet de 10 Mbps, juntamente com cabos de par trançado de categoria 3 ou 5 ou, em uma variante, cabos coaxiais. Nas atuais redes Ethernet de 100 Mbps, esses requisitos são substituídos por interfaces de 100 Mbps, ligadas por cabos de par trançado blindados nível 5.

Cabos diferentes exigem conexões diferentes – também chamadas, popularmente, de portas ou encaixes – para cada placa de rede. É comum em placas Ethernet a existência de apenas um encaixe para cabos de par trançado. Em placas de uso altamente profissional, contudo, não é rara a inclusão de interfaces para ligação de cabos de fibras ópticas. As pla-

Figura 1
Interface de rede multi-portas para uso em roteadores. Note que há um controlador de rede autônoma para cada porta disponível.

cas que trazem encaixes para mais de um tipo de cabo são chamadas placas combo.

Existem também interfaces aptas a interligar mais de dois cabos de um mesmo padrão simultaneamente, normalmente com o número de quatro portas (**Figura 1**). Interfaces de rede de multipla conexão, no entanto, devem possuir tantas controladoras de rede quantas sejam as conexões que desejam administrar.

Também podem existir interfaces virtuais, operáveis logicamente, como veremos mais adiante.

Enlace

Do ponto de vista lógico, o principal componente da interface é o LLC (*Logical Link Control*), a mais alta das camadas do enlace de dados. As camadas padrão de enlace de dados, definidas pela IEEE, têm três funções específicas:

1) Detectar e, possivelmente, corrigir erros nas camadas de meios físicos (hardware);

2) Fornecer à camada de rede a capacidade de pedir estabelecimento de circuitos de dados na camada, por meio de chaveamento de circuitos;

3) Fornecer os meios funcionais para ativar, manter e desativar uma ou mais conexões de dados entre entidades da camada de rede.

A camada LCC manipula o controle de erros, gerenciamento de controle de fluxo e direcionamento da camada MAC (meio físico). O protocolo LLC generalizado é o estabelecido pela nossa cara norma IEEE 802.2. O protocolo estabelecido por essa norma preve duas variantes de LLC: o protocolo orientado a conexão e o não orientado a conexão. Destes, o modo que nos interessa é o modo orientado a conexão, também chamado de CONS - *Connection Oriented Network Service* (Serviço de Rede Orientado a Conexão).

O CONS estabelece uma conexão durável entre duas estações de trabalho, dispositivos de rede ou quaisquer equipamentos que possuam uma interface de rede em funcionamento, ligadas entre si por um meio físico também funcional. Ou seja, o CONS é responsável por criar o que normalmente se entende por um enlace de redes, além de ser o responsável por garantir a entrega de pacotes de dados por meio do fluxo de comunicação do enlace (serviço confiável). É o serviço de conexão quem garante ao receptor a entrega, na seqüência correta, dos dados enviados, bem como a proteção contra perdas e dados duplicados.

Dial-up

O melhor exemplo de reunião entre interface de hardware e interface lógica com controle de conexões e proteção contra erros, por incrível que pareça, é o velho modelo de conexão de linha comutada ou Dial-Up, ou ainda, simplesmente, discada.

As conexões por linha comutada ou discada foram, e ainda são, uma forma barata de acesso a Internet, na qual o cliente utiliza um dispositivo modulador e demodulador de sinal (daí o nome modem, retirado da sigla *MOdulator, DEModulator*) para se conectar, através da Rede Telefônica Comutada (RTC) a rede do ISP (*Internet Service Provider* ou Provedor de Acesso a Internet). Para que essa comunicação seja possível, basta possuir um servidor de acesso (por exemplo, PPP ou *Point to Point Protocol*) do lado do provedor, uma implementação do protocolo TCP/IP para estabelecer o enlace, além do meio físico (linhas telefônicas) e lógico (enlace modem-a-modem corretamente configurado).

Além de serem um bom exemplo de funcionamento do binômio interface de hardware x interface lógica, as conexões dial-up foram citadas neste capítulo inicial por um bom motivo: as conexões PPP são o modelo para todas as formas de conexão posteriores (PPPoE, PPP-Chap), e suas derivadas. ■

Capitulo 2 – Interfaces: tecendo a rede

Os conceitos expostos no primeiro capítulo devem servir como guias para o crescimento do leitor no mundo das redes: estude as normas técnicas, saiba de onde elas vêm, veja se elas são aplicadas na rede que você está projetando ou onde você trabalha e, sobretudo, **saiba porque as coisas devem ser feitas de um jeito, e não de outro**, tanto para satisfação própria quanto para ter o que responder, caso seja interpelado por um cliente ou colaboradores. As pessoas costumam acatar melhor as decisões das quais sabem o motivo, ou as coisas das quais conhecem o funcionamento.

Neste segundo capítulo iremos iniciar a montagem prática de nossa rede, começando pela implementação de comunicação entre interfaces. Supomos aqui uma rede com cabeamento já implementado e, portanto, já capaz de enviar e receber informações. Mas não se assuste: o cabeamento será, sim, alvo de nossos estudos, no *capítulo 3* deste livro, assim como todas as questões de hardware advindas da montagem de uma estrutura cabeada **mínima**.

O sistema escolhido para a implementação do funcionamento de interfaces foi GNU/Linux, por razões mais do que óbvias: o Linux possui um suporte a implementação e utilização de ferramentas, protocolos e interfaces de rede mais "natural" – no sentido de mais fluído e integrado ao restante do sistema – do que os sistemas rivais, como o MacOS e o Windows. Além disso, boa parte das ferramentas para rede Linux são altamente confiáveis e gozam da arte da absoluta simplicidade em seu funcionamento – coisa da qual sistemas com o mesmo grau de confiabilidade, como os da família BSD, não têm como se gabar.

Por onde começar

Embora todas as distribuições Linux atuais instalem os elementos básicos para o funcionamento do computador em uma

rede, a responsabilidade por seu gerenciamento ainda fica a cargo dos administradores do sistema – e em alguns casos o treinamento recebido não cobriu como deveria todo o conjunto de técnicas associadas ao funcionamento de uma rede. Neste caso, faz sentido obter o máximo de conhecimento possível sobre o que pode acontecer com computadores ligados em rede.

Felizmente, a maioria das distribuições Linux já traz todas as ferramentas necessárias para solucioná-los. Por outro lado, infelizmente, a maioria dessas ferramentas, ao menos em seu nível mais básico, assume que você saiba exatamente como uma rede de computadores funciona.

O protocolo TCP/IP é o componente básico da Internet e de muitas redes locais. Ele é uma combinação do *Transmission Control Protocol* (Protocolo de Controle de Transmissão - TCP) e do *Internet Protocol* (Protocolo de Internet - IP), e especifica como os computadores devem se comunicar.

Assim como um navegador web não necessita saber se a informação está sendo transmitida via componentes "wireless" ou linhas FDDI, e uma linha FDDI não precisa saber se os bits que ela está transportando pertencem a arquivos HTML, MP3s ou vídeos, especialistas usam um modelo de camadas (layers) para descrever redes de computadores. Como em uma cebola, cada camada é construída sobre as camadas inferiores, mas, fora isto, cada uma trabalha de forma independente das outras. Existem duas implementações modernas do sistema de camadas de rede: o modelo OSI (*Open Systems Interconnection*), coberto no **quadro 1**, e o modelo TCP. A **figura 2** mostra a hierarquia de camadas, de baixo para cima, adotada pelo modelo OSI e assumida, com variações acessórias no número de camadas, também pelo padrão TCP. Vale a pena salientar que, apesar da universalidade e disseminação do modelo TCP, temos o modelo OSI como padrão na implementação de comunicações entre redes de computadores, segundo normas da Fundação ISO (*International Standardization Organization*).

Figura 2
Camadas constitutivas de uma pilha de rede, segundo o modelo OSI aprovado pelo padrão ISO.

A camada de aplicação, como o nome sugere, define como aplicações, tais como navegadores ou programas de email, "conversam" com servidores de Internet ou de email, respectivamente. Como isto ocorre exatamente depende da aplicação. Por exemplo, o Protocolo de Transferência de Hipertexto (HTTP) é utilizado na Internet, enquanto downloads de arquivos normalmente são feitos utilizando o Protocolo de Transferência de Arquivos (FTP).

A camada de transporte, como você pode ver na **figura 2**, reside três camadas abaixo da camada de aplicação. No modelo TCP, visto na **figura 3**, as camadas de Apresentação e Sessão são anexadas a camada de Aplicação, o que faz com que esta e a camada de Transporte sejam vizinhas próximas.

Esta camada realiza conexões entre computadores, permitindo que eles troquem dados. O TCP cria uma fila de dados estável entre os pontos de rede (para os protocolos de aplicação HTTP, SSH, POP ou SMTP) e assegura que pacotes perdidos sejam retransmitidos. O outro protocolo mais significativo neste nível é o Protocolo de Datagrama de Usuário (UDP), que pode perder pacotes (e é utilizado, por exemplo, por "streams" do Real Audio).

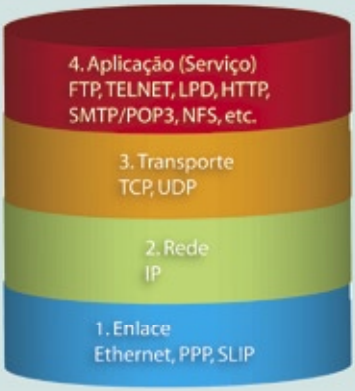


Figura 3
Modelo de camadas da pilha TCP.

As coisas começam a ficar realmente interessantes na camada de rede inferior. É nela que os pacotes de dados são colocados na mídia de transporte e tentam encontrar a melhor rota até o seu alvo. Para simplificar esta tarefa, cada pacote inclui os endereços do transmissor e do receptor. Quando uma página da Web é transmitida pelo servidor, os pacotes que a formam podem tomar rotas diferentes. Após aceitar tais pacotes, o receptor deve assegurar que eles serão "remontados" na ordem correta.

Além do *Internet Protocol* (IP), a camada de rede tem protocolos como o *Internet Control Message Protocol* (Protocolo de Controle para Mensagens na Internet - ICMP), para efetuar o controle de mensagens (tais como notificações de erro), o *Address Resolution Protocol* (Protocolo de Resolução deEndereços - ARP), que associa endereços IP a endereços de hardware (MAC), e seu oposto, o protocolo RARP (*Reverse Address Resolution Protoco*l - Protocolo de Resolução Reversa de Endereço).

A camada mais inferior do modelo OSI é a camada física –substituída, no modelo TCP, por uma junção entre ela e a camada de Enlace, batizada com o nome desta última. Neste nível, a

principal preocupação é com a transmissão de bits, bem como a padronização de protocolos para tratar de interfaces elétricas, mecânicas e de sinalização.

Os componentes de uma rede são identificados através dos seus endereços IP. O TCP pode retransmitir pacotes perdidos ou danificados, assegurando que o receptor contará com, pelo menos, um conjunto completo dos pacotes enviados. Os protocolos de aplicação, por mais sofisticados que sejam, se baseiam neste serviço. Sem um pouco de conhecimento sobre as camadas citadas acima, muitas ferramentas de rede não farão nenhum sentido.

Configurando uma interface: ifconfig

Antes de começar a projetar uma rede, é importante certificar-se se cada estação de trabalho está realmente utilizando a transmissão de cabos como deveria (novamente não se assuste: sim, nós vamos falar de cabeamento!).

Para encurtar a história, cada máquina necessita um endereço IP (também conhecido como *IP Address*) único para ser capaz de comunicar-se com as outras máquinas da rede. Um *gateway* permite que pacotes de dados com destino à Internet possam deixar a rede local.

A maneira mais simples de atribuir um endereço a uma interface de rede é utilizar o comando `ifconfig`, que pode ser invocado a partir de qualquer terminal executado como usuário root. Para isso, utilize a ferramenta de linha de comando `ifconfig`, logado como usuário root.

Ao digitar o comando `ifconfig`, sem parâmetros extras, em um terminal, surgem as informações referentes a todas as interfaces atualmente instaladas e configuradas no computador:

```
eth0      Encapsulamento do Link: Ethernet  Endereço de HW
          ➥ 00:11:D8:3F:37:57
```

```
          inet end.: 192.168.1.166  Bcast:192.168.1.255
          ➥ Masc:255.255.255.0
          endereço inet6: fe80::211:d8ff:fe3f:3757/64 Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:4080 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3291 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:3915197 (3.7 MiB)  TX bytes:349577 (341.3 KiB)
          IRQ:185 Endereço de E/S:0xe000
```

```
lo         Encapsulamento do Link: Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          endereço inet6: ::1/128 Escopo:Máquina
          UP LOOPBACKRUNNING  MTU:16436  Métrica:1
          RX packets:104 errors:0 dropped:0 overruns:0 frame:0
          TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:0
          RX bytes:6550 (6.3 KiB)  TX bytes:6550 (6.3 KiB)
```

```
vmnet1     Encapsulamento do Link: Ethernet  Endereço de HW
           ➥ 00:50:56:C0:00:01
          inet end.: 172.16.133.1  Bcast:172.16.133.255
          ➥ Masc:255.255.255.0
          endereço inet6: fe80::250:56ff:fec0:1/64 Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

```
vmnet8     Encapsulamento do Link: Ethernet  Endereço de HW
           ➥ 00:50:56:C0:00:08
          inet end.: 192.168.168.1  Bcast:192.168.168.255
          ➥ Masc:255.255.255.0
          endereço inet6: fe80::250:56ff:fec0:8/64 Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

As informações mais importantes mostradas pela saída do comando são as seguintes:

- Cada interface ativa é identificada por seu nome. Por exemplo, temos *eth0*, que é a primeira –, e nesse caso, a única – placa de rede Ethernet presente no computador. Temos também a interface *lo* (*loopback*), responsável pelas conexões internas. No nosso caso, em específico, temos também duas interfaces virtuais, configuradas para trabalhar com o software de virtualização Vmware. Essas interfaces costumam usar a sigla *vmnet*, acrescida de um número de identificação.

- Quando estamos diante de um adapatador de rede físico temos o número do Mac Address (veja **quadro 1**), precedido do termo *HW*.

- O endereço IP atribuído à placa é precedido do termo `inetaddr` ou `inet end`. O endereço de broadcast address é identificado pelo termo *Bcast*, e a máscara de sub-rede pela sigla Mask.

- O endereço do padrão Ipv6 de cada interface é precedido pelo termo `inet6 addr` (`end inet6`, em algumas distribuições).

- O comando mostra a atividade atual de cada atividade – no caso da interface eth0 mostrada, estão listadas as atividades *UP*, *BROADCAST*, *RUNNING* e *MULTICAST*.

- As estatísticas sobre o envio e recebimento de pacotes são listadas nas colunas *RX* e *TX*, respectivamente. Esse detalhe importa muito em "tira-teimas" técnicos: quando a interface parece não funcionar, mas ao mesmo tempo as luzes da placa de rede ascendem continuamente, podemos estar diante da falha de um serviço específico, como um cliente HTTP via browser, ou a recepção de e-mails via SMTP, mas não do enlace de dados em si.

Manipulando interfaces

Agora vamos mostrar como manipular interfaces de rede usando o comando `ifconfig`. Os comandos básicos para habilitar/desabilitar uma interface estão disponíveis no **quadro1**, **Opções do comando ifconfig**, e serão necessários para as manipulações de interfaces vistas a seguir.

Uma das primeiras coisas que podemos fazer é alterar o endereço físico da placa, ou Mac Address. Muitas interfaces de rede de baixo custo utilizam o mesmo endereço MAC (veja o **quadro 2**, **Endereços Mac**) para placas do mesmo lote. Isso

Quadro 1 - Opções do comando ifconfig

- `-a`
  Este comando pede que `ifconfig` mostre informações sobre todas as interfaces presentes, mesmo que inativas.
- `-s`
  Mostra uma pequena lista de todas as interfaces ativas, com um sumário que mostra a entrada e saída de pacotes de dados em todas as interfaces
- `-v`
  A opção "verbose", retorna informações extras, detalhadas, sobre cada interface e suas conexões ativas.
- `[nome da interface]`
  Ao adicionar o nome da interface ao comando ifconfig, estamos dizendo ao comando que desejamos que o resultados de nossas ações seja aplicado somente a interface indicada.
- `up`
  Utilizado após o nome da interface, o parâmetro up manda que uma placa de rede seja posta em funcionamento.
- `down`
  De efeito contrário ao parâmetro anterior, este comando é utilizado para desabilitar uma interface.
- `netmask [endereço]`
  Use a opção netmask para atribuir ou alterar a máscara de endereço da interface de rede (sim, nem todas as redes do mundo usam 255.255.255.0!). Esse parâmetro deve ser usado após a definição da interface: `ifconfig eth0 netmask 255.255.255.0`.
- `broadcast [addr]`
  A opção "broadcast" é acompanhada do endereço que se deseja atribuir ao broadcast: `ifconfig eth0 broadcast 192.168.2.25`.

não seria grave, se não nos confrontassemos com a necessidade de possuir cinquenta ou cem placas, de um mesmo lote, dentro de uma rede filtrada por um firewall ou bridge. É nessas situações que a alteração do Mac Adress de forma manual – se bem que trabalhosa – pode ao menos resolver problemas de identidade entre interfaces sem o expediente de compra de mais hardware.

O Mac Address de uma placa de rede não pode, teoricamente, ser alterado fisicamente. Ele pode, contudo, ser alterado virtualmente. Para fazer com ifconfig devemos, antes de tudo, "derrubar" a interface:

```
insigne$ ifconfig eth0 down
```

Em seguida, usamos o parâmetro `hw`, seguido de `ether`, para alterar o Mac Address:

```
insigne$ ifconfig eth0 hw ether 00:D1:D2:65:2C:03
```

Para que as alterações tenham efeito, devemos "levantar" a placa, usando o comando `up`:

```
insigne$ ifconfig eth0 192.168.1.52 netmask 255.255.255.0 up
```

Interfaces virtuais

Com o ifconfig, também podemos atribuir mais de um endereço IP a mesma interface de rede, criando uma interface virtual e atrelando-a a interface física. O início do processo é a trivial atribuição de endereço principal a um dispositivo físico de rede:

```
insigne$ eth0 192.168.0.1 netmask 255.255.255.0 up
```

para, logo em seguida, adicionar um segundo endereço, criando uma interface virual. Para isso, basta atribuir o prefixo `:1`, `:2`, ou qualquer numeração que se deseje usar, ao nome real da interface:

```
insigne$ ifconfig eth0:1 10.0.0.5 netmask 255.255.255.0 up
```

Podemos adicionar vários endereços a uma mesma interface:

```
insigne$ ifconfig eth0:2 172.16.12.78 netmask 255.255.255.0 up
```

E utilizar o comando `ifconfig`, logo em seguida, para ter acesso ao resultado:

```
insigne$ ifconfig
eth0       Encapsulamento do Link: Ethernet  Endereço de HW
           00:11:D8:3F:37:57
          inet end.: 192.168.1.166  Bcast:192.168.1.255
Masc:255.255.255.0
          endereço inet6: fe80::211:d8ff:fe3f:3757/64 Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:17061 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12862 errors:0 dropped:0 overruns:0
           carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:14999587 (14.3 MiB)  TX bytes:1307330 (1.2
          MiB)
          IRQ:185 Endereço de E/S:0xe000
```

```
eth0:1     Encapsulamento do Link: Ethernet  Endereço de HW
00:11:D8:3F:37:57
          inet end.: 10.0.0.5  Bcast:10.0.0.255
Masc:255.255.255.0
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          IRQ:185 Endereço de E/S:0xe000

eth0:2     Encapsulamento do Link: Ethernet  Endereço de HW
00:11:D8:3F:37:57
          inet end.: 172.16.12.78  Bcast:172.16.12.255
Masc:255.255.255.0
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          IRQ:185 Endereço de E/S:0xe000
```

Modo promíscuo

O modo promíscuo permite que a interface de rede receba todos os pacotes que passam por ela, mesmo os que não são destinados a ela. Em uma rede Ethernet, todos os pacotes que trafegam pelo segmento de rede, ao qual o receptor em modo promíscuo está conectado, são recebidos pelo mesmo, em vez de receber apenas os pacotes endereçados a interface em específico. Os sniffers – softwares usados para captura de pacotes dentro de um segmento de rede, seja para análise do tráfego, seja para fins ilícitos –, funcionam a partir da utilização de interfaces de rede promíscuas. Além disso, a multiplicação de placas em modo promíscuo em uma rede pode aumentar de forma considerável o volume de tráfego da rede.

O `ifconfig` permite o desmonte do modo promíscuo em uma interface de rede local. Para isso, basta utilizar o comando:

```
insigne$ ifconfig eth0 -promisc
```

A verificação do modo de recepção em que a placa está operando pode ser feita com o comando:

```
# ifconfig | grep -i PROMISC
```

Verificação de Status

O comando `ip` fornece detalhes sobre a sua configuração atual. Sistemas mais antigos talvez tenham somente os comandos `ifconfig` e `route`, que mostram a mesma informação, embora de forma ligeiramente diferente. Se o seu shell não é capaz de encontrar nenhum destes comandos, pode ser que eles tenham sido instalados no diretório `/sbin`, que não está normalmente no search path (rota de busca) do sistema. Neste caso, basta informar o caminho inteiro quando quiser rodar o programa (ex: `/sbin/ip`).

A opção `addr` indica ao comando ip que ele deve fornecer dados sobre a placa de rede. A linha finalizada por `eth0` indica a primei-

Quadro 2 – Endereços MAC

MAC ou Media Access Control é o endereço físico de uma interface de rede. É um endereço de 48 bits, representado em hexadecimal. O protocolo é responsável pelo controle de acesso de cada estação à rede Ethernet. Este endereço é o utilizado na camada 2 do Modelo OSI.

Exemplo:

`00:00:5E:00:01:03`

Os três primeiros octetos são destinados à identificação do fabricante, os 3 posteriores são fornecidos pelo fabricante. É um endereço universal, isto é, não existem, em todo o mundo, duas placas com o mesmo endereço.

ra placa de rede do sistema (`eth1` é a segunda – possivelmente utilizada para WLAN, e assim por diante). Ela mostra o endereço IP do computador (192.168.1.245 na **figura 1**), a máscara de rede (/24), o endereço de broadcast (192.168.1.255) e o nome da interface de rede (`eth0`).

O resultado do comando `ip route` é mais fácil de ler. A primeira linha mostra a rede (o endereço da rede no nosso exemplo é 192.168.1.0), a netmask (/24), a interface de rede e, por fim, a chamada *data source* (por isso o termo *src*, indicando "source") que é o endereço IP (192.168.1.245). A segunda linha indica o *default gateway*, cujo endereço IP é 172.16.133.1:

```
insigne:~# ip route
192.168.1.0/24 dev eth0  proto kernel  scope link  src
192.168.1.166
172.16.133.0/24 dev vmnet1  proto kernel  scope link  src
172.16.133.1
192.168.168.0/24 dev vmnet8  proto kernel  scope link  src
192.168.168.1
default via 192.168.1.254 dev eth0
insigne:~#
```

Caso informações importantes, tais como os endereços IP da máquina e do gateway, estiverem faltando aqui, isso pode ser explicado por problemas em seu computador ou na interface de rede acoplada a ele. Neste caso, rode a ferramenta de configuração da sua distribuição (como o YaST, no caso do SuSE Linux), verifique a configuração de rede do seu sistema e tente novamente.

Ping-Pong

ping é uma ferramenta simples para análise da rede, mas é também extremamente prática. Ela transmite pacotes de dados ICMP do seu computador para um computador alvo e mostra o tempo que cada resposta leva para retornar ao seu computador – assumindo que o computador alvo responda. A seção final do resultado de um ping é um grupo de estatísticas que mostram a você quantos pacotes foram transmitidos:

```
insigne:~# ping 192.168.1.166
PING 192.168.1.166 (192.168.1.166) 56(84) bytes of data.
64 bytes from 192.168.1.166: icmp_seq=1 ttl=64 time=0.054 ms
64 bytes from 192.168.1.166: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 192.168.1.166: icmp_seq=3 ttl=64 time=0.044 ms
64 bytes from 192.168.1.166: icmp_seq=4 ttl=64 time=0.045 ms
64 bytes from 192.168.1.166: icmp_seq=5 ttl=64 time=0.045 ms
64 bytes from 192.168.1.166: icmp_seq=6 ttl=64 time=0.046 ms
64 bytes from 192.168.1.166: icmp_seq=7 ttl=64 time=0.044 ms

--- 192.168.1.166 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 5998ms
rtt min/avg/max/mdev = 0.044/0.046/0.054/0.004 ms
insigne:~#
```

Além disso, o ping mostra quantas respostas retornaram e quanto tempo isto levou para acontecer (em média 0.045 milisegundos). Se pacotes fossem descartados ou perdidos, tais estatísticas os mostrariam em uma seção chamada *packet loss*. Se o computador alvo não está acessível, nada acontece, enquanto o ping aguarda por respostas.

O comando ping hostname transmite pacotes ICMP sem parar até que você pressione **[Ctrl-C]**. Você também pode especificar `ping -c 10 hostname` para transmitir apenas dez pacotes.

Rotas

Enquanto ping só mostra se o computador alvo está respondendo, traceroute mostra todo o caminho que os pacotes de dados tomaram até atingi-lo. Basta utilizá-lo em um terminal, associado ao endereço IP de destino.

```
insigne:~# traceroute 200.204.0.10
traceroute to 200.204.0.10 (200.204.0.10), 30 hops max, 40 byte
packets
 1  orion.intra.linuxnewmedia.com.br (192.168.1.254)  1.267 ms
    0.435 ms  0.171 ms
 2  10.22.0.1 (10.22.0.1)  37.313 ms  29.284 ms  14.014 ms
 3  c9060003.virtua.com.br (201.6.0.3)  40.117 ms  17.874 ms
    30.804 ms
 4  c9060009.virtua.com.br (201.6.0.9)  20.969 ms  28.679 ms
    10.079 ms
 5  embratel-G5-3-gacc06.spo.embratel.net.br (189.2.2.1)  41.300
    ms  77.536 ms  29.105 ms
 6  ebt-C2-gacc03.spo.embratel.net.br (200.230.243.13)  30.763 ms
    14.897 ms  26.922 ms
 7  peer-P2-0-gacc03.spo.embratel.net.br (200.174.250.2)  27.664
    ms  52.461 ms  14.565 ms
 8  200-204-20-150.dsl.telesp.net.br (200.204.20.150)  32.116 ms
    17.554 ms  9.842 ms
 9  * * *
10  * * *
11  * * *
12  * * *
```

Os asteriscos (*) indicam que um erro ocorreu no roteamento, ou que um firewall bloqueou a passagem deste tipo de pacote de dados. Aliás, a opção `-n` pode ser utilizada para evitar que os

hostnames sejam mostrados. Para indicar qual a interface de rede a ser usada, utilize a opção `-i`:

```
insigne:~# traceroute -n -i eth0 200.204.0.10
traceroute to 200.204.0.10 (200.204.0.10), 30 hops max, 40 byte
packets
 1  192.168.1.254  4.990 ms  0.805 ms  0.155 ms
 2  10.22.0.1  71.910 ms  19.661 ms  12.177 ms
 3  201.6.0.3  383.006 ms  18.482 ms  50.906 ms
 4  201.6.0.9  54.211 ms  13.678 ms  12.077 ms
 5  189.2.2.1  50.032 ms  21.272 ms  59.280 ms
 6  200.230.243.13  39.439 ms  33.780 ms  8.660 ms
 7  200.174.250.2  43.537 ms  52.806 ms  32.558 ms
 8  200.204.20.150  21.607 ms  17.644 ms  10.368 ms
 9  *
```

O comando `mtr targethost` mostra a rota dos pacotes de forma mais clara, pois indica exatamente onde os pacotes estão tendo atrasos maiores (caso você não pressione a tecla **[q]**). Para cada trecho da rota, mtr descobre o que está acontecendo com os pacotes de dados. Por isso, mtr pode ser visto como uma combinação entre ping e traceroute. Por exemplo, o comando

```
mtr -c <endereço>
```

informa ao mtr para transmitir apenas dez pacotes, parar e fornecer um relatório. A coluna HOST indica onde o pacote de dados se encontra, LOSS informa a porcentagem de pacotes descartados ou perdidos, RCVD e SENT mostram, respectivamente, quantos pacotes foram recebidos e enviados e as colunas BEST, WORST e AVG informam quanto tempo a transmissão dos pacotes levou, no melhor e no pior dos casos, e também na média.

Para obter maior precisão…

… experimente usar *tcpdump*, a ferramenta de análise de rede de mil e uma utilidades. A maioria das distribuições Linux já fornece um pacote de instalação. Caso contrário, o

código fonte pode ser baixado em http://www.tcpdump.org. Se quiser compilar seu próprio binário, você vai precisar da biblioteca *libcap*.

O tcpdump precisa de privilégios de administrador para rodar, uma vez que ele coloca a placa de rede em modo "promíscuo", o que permite a ela ler quaisquer pacotes que passem pela interface de rede local, inclusive a captura de dados.

O tcpdump mostra todos os pacotes de dados que aparecem na interface da placa de rede:

```
11:56:27.833598 192.168.1.245.ssh > 192.168.1.20.39258: P
1392512:1392720(208) ack 1201 win 9120
<nop,nop,timestamp 2599771999 1711932971> (DF) [tos 0x10]
```

Em nosso exemplo pode ser visto que a máquina com o IP 192.168.1.245 enviou um pacote de dados ssh para a máquina com o IP 192.168.1.20. Digite:

```
tcpdump -i eth0 port 80
```

e os dados destinados à porta TCP número 80 – comumente utilizada por servidores de Internet – serão mostrados:

```
insigne:~# tcpdump -i eth0 port 80
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96
bytes
18:45:23.839592 IP 192.168.1.166.60395 > mail.linuxnewmedia.com.
br.www: S 1990809884:1990809884(0) win 5840 <mss
1460,sackOK,timestamp 25527477 0,nop,wscale 2>
18:45:23.852898 IP mail.linuxnewmedia.com.br.www >
192.168.1.166.60395: S 4166830581:4166830581(0) ack 1990809885
win 5792 <mss 1460,sackOK,timestamp 599092437 25527477,nop,wscale
0>
```

Por outro lado, `tcpdump host <endereço>` fornecerá o tráfico de rede em targethost:

```
insigne:~# tcpdump host 200.204.0.10
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96
bytes
```

Pra onde?

É importante instalar ferramentas especializadas de modo a não ficar "no escuro" no que concerne à utilização da rede. O iptraf é um exemplo. Ele mostra exatamente o que está acontecendo com a placa de rede, que protocolos estão sendo utilizados atualmente e com que outras estações a máquina sob análise está "conversando". Pressione as teclas **[q]** e **[Enter]** para encerrar o programa.

No menu principal há um item chamado *IP Traffic Monitor* que fornece uma visão geral do tráfego dos dados na rede, e permite a identificação de pontos de sobrecarga (**figura 4**). Por outro lado, o item *Detailed Interface Statistics* (ver **figura 5**) não indica quais máquinas estão trocando dados, mas analisa o fluxo do tráfego, classificado por protocolo.

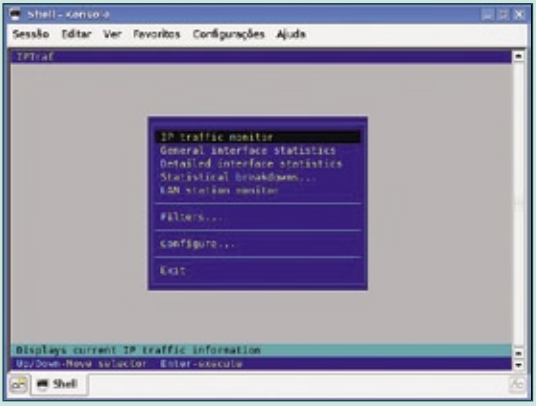


Figura 4
Menu de funcionalidades do iptraf.

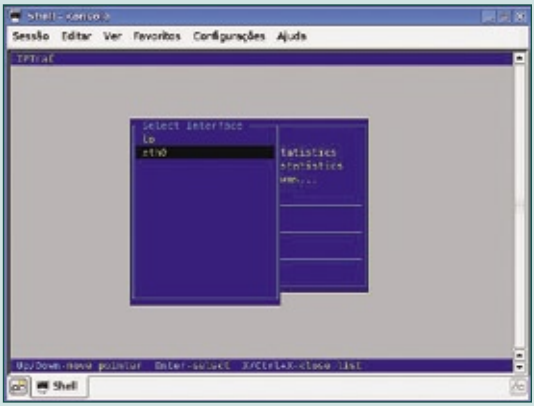


Figura 5
Escolha a interface que deseja monitorar utilizando a opção *Detailed Interface Statistics*.

O comando `iptraf` fornece informações úteis sobre a taxa de transferência de dados e indica onde se encontram problemas de performance (os chamados gargalos ou *bottlenecks*). Por exemplo, se há mais saída do que entrada de dados, pode-se assumir que alguém está baixando algo do seu micro.

Teríamos muito mais a relatar sobre `iptraf` e outras ferramentas mencionadas neste capítulo. Se o leitor quer melhorar seus conhecimentos nesta área, não há alternativa: adquira o máximo de conhecimento e experiência possível. ■

Capítulo 3 – Cabeamento

O cabeamento é, muito provavelmente, a parte de redes que mais tem a ver com a manipulação de hardware, já que estamos falando do meio físico de transmissão de dados, necessário para que duas ou mais interfaces troquem informações pela rede.

Se você leu o *Capítulo 2* mas não possuía uma rede já estruturada, provavelmente ansiou por este capítulo, o que parece ser bem justificado: sem cabeamento, e a não ser que se utilize comunicação sem fio, via *Wi-fi* ou infravermelho, é impossível subsistir um segmento de rede sem a correta organização de cabos.

Um pouco de história

No início de 1985, preocupados com a falta de uma norma que determinasse parâmetros das fiações em edifícios comerciais, os representantes das indústrias de telecomunicações e informática solicitaram a CCIA (*Computer Communication Industry Association*) que manufaturasse normas que abrangessem estes parâmetros. Aceita a solicitação, a CCIA pediu a EIA (*Eletric Industry Association*) o desenvolvimento de uma norma, que passou a ser desenvolvida em conjunto com a TIA (*Telecommunications Industries Association*).

A primeira versão da norma, batizada como EIA/TIA 568, foi lançada em julho de 1991. Após a publicação da norma inicial, diversos boletins técnicos foram sendo emitidos e incorporados a esta norma. Em janeiro de 1994, foi emitida a norma que perdura até hoje, nomeada como EIA/TIA 568 A, atualizada pela última vez em 2000. Com a criação desta norma e suas complementares – as normas 569, 606 e 607 –, houve uma mudança no modo de agir dos usuários de sistemas. Os sistemas de cabos foram integrados, com o cabeamento permitindo o tráfego dos sinais independente do fabricante, da fonte geradora, ou do protocolo transmitido. Com a emissão da norma, o sistema de cabeamento com fibra óptica foi complementado e tornou-se escopo da mesma, tendo suas especificações de instalação,

construção e testes executados dentro dos seus rígidos padrões . Esse padrão passou a ser conhecido como **cabeamento estruturado**.

Cabo e estruturação

Um sistema de cabeamento estruturado permite o tráfego de qualquer tipo de sinal elétrico de áudio, vídeo, controles ambientais e de segurança (câmeras de vigilância, por exemplo), dados e telefonia, convencional ou não, de baixa intensidade, independente do produto adotado ou fornecedor.

Este tipo de cabeamento, possibilita mudanças, manutenções, implementações e a adição de pontos de rede de forma rápida, segura e controlada. No esquema de cabeamento estruturado, convencionado pelas normas, toda alteração do esquema de ocupação de um edifício comercial é administrada e documentada seguindo-se um padrão de identificação que não permite erros ou dúvidas quanto aos cabos, tomadas, posições e usuários. A obediência dessas normas e sua correta aplicação exigem requisitos mínimos relativos às distâncias, topologias, pinagens, interconectividade e transmissão, permitindo desta forma que atinja-se o desempenho esperado.

Composição

Um sistema de cabeamento estruturado compõem-se de seis subsistemas, cada qual tendo suas próprias especificações de instalação, desempenho e teste. Os subsistemas são divididos nos seguintes itens, que serão estudados, caso a caso, logo a seguir:

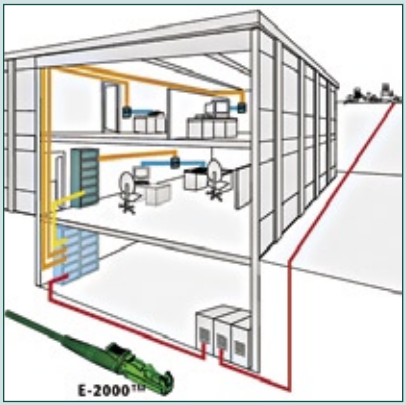
1. Cabeamento Horizontal (*Horizontal Cabling*);
2. Área de Trabalho (*Work Area*);
3. Cabeamento Vertical (*BackBone*);
4. Armário de Telecomunicações (*Telecommunications Closet*);
5. Sala de Equipamentos (*Equipments Room*);
6. Entrada de Facilidades (*Entrance Facilities*).

Cabeamento Horizontal (*Horizontal Cabling*)

Cabeamento horizontal é a parte do sistema de cabeamento que contém a maior quantidade de cabos instalados. Ele estende-se

do ponto de telecomunicação instalado na área de trabalho até o armário de telecomunicações. É chamado de horizontal porque os cabos são passados sob piso, suspenso ou não, em dutos ou canaletes. Canaletes de plástico sobre o piso, como os comumente vistos aqui no Brasil, não são contemplados nem considerados uma instalação de cabeamento padrão pelas normas EIA-TIA.

Figura 6
Esquema de cabeamento



Área de Trabalho (*Work Area*)

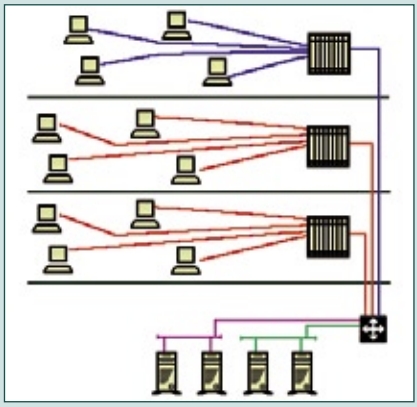
É o local onde o usuário começa a interagir com o sistema de cabeamento estruturado. É na área de trabalho que estão situados os equipamentos, como estações de trabalho (computadores), periféricos com acesso a rede, telefones, fax, dentre outros.

Cabeamento Vertical (*BackBone Cabling*)

A função básica do cabeamento vertical (ou *backbone cabling)* é interligar todos os armários de telecomunicação instalados nos andares de um edifício comercial ou vários edifícios comerciais (*campus backbone*), onde também serão interligadas as facilidades de entrada (*entrance facilities*).

A topologia adotada para o cabeamento vertical é a estrela (veja mais detalhes sobre topologia de redes na sessão *Topologia* deste guia).

Figura 7
Esquema de
BackBone Cabling



Os principais fatores a serem considerados quando de dimensionamento do cabeamento vertical são:

- Quantidade de área de trabalho;
- Quantidade de armários de telecomunicações instalados;
- Tipos de serviços disponíveis;
- Nível de desempenho desejado.

Armário de Telecomunicação (*Telecommunication Closet*)

Após a instalação de todos os cabos do cabeamento horizontal, deve-se fazer a instalação em cada área de trabalho, interligando-a ao hardware de conexão escolhido. Este hardware de conexão deve ser protegido contra o manuseio indevido por parte de pessoas não autorizadas. Para assegurar essa proteção, convencionou-se instalar todos os hardwares de conexão, suas armações, *racks*, e outros equipamentos em uma sala destinada para esta função, locada em cada andar. Esta sala é chamada de armário de telecomunicação (*telecommunication closet*).

Um armário de telecomunicações deve ser instalado levando-se em conta algumas premissas:

- Quantidade de áreas de trabalho;

- Disponibilidade de espaço no andar;
- Instalação física.

Sala de Equipamentos (*Equipments Room*)

A sala de equipamentos é o espaço reservado dentro do edifício ou área atendida onde está instalado o distribuidor principal de telecomunicações, responsável por providenciar a interconexão entre os cabos do armário de telecomunicações, *backbone cabling* ou *campus backbone*, com os equipamentos de rede, servidores e os equipamentos de voz (PABX).

Existem algumas regras que devem ser seguidas quando da instalação da sala de equipamentos:

- A área de instalação deve ser maior ou igual a $14m^2$;
- A sala de equipamentos deve ser instalada à um mínimo de 3m de qualquer fonte de interferência eletromagnética, como cabines de força, geradores, instrumentos emissores de raio X, elevadores, sistemas irradiantes, sistemas de caixas acústicas, etc;
- Deve-se instalar tomadas elétricas a cada 1,5m;
- Deve-se instalar uma iluminação com um mínimo de 540 luz/m^2;
- Deve-se realizar um levantamento da edificação, para que se garanta a instalação da rede longe de águas fluviais, esgotos e outros afluentes.

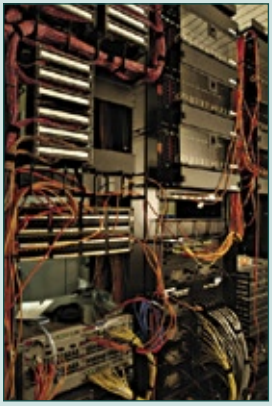


Figura 8
Perfil típico de uma sala de equipamentos.

Facilidades de Entrada (*Entrance Facilities*)

A norma ANSI/TIA/EIA–569-A define uma facilidade de entrada como qualquer local onde os serviços de telecomunicações entram em um prédio, ou onde há rotas de backbone vinculadas a outros edifícios no campus onde estão localizados. A facilidade de entrada pode conter dispositivos com interface de redes públicas bem como equipamentos de telecomunicações. As normas recomendam que o local da facilidade de entrada deve estar em uma área seca, perto das rotas de *backbone* vertical.

Quadro 3 – O que é *backbone*?

Um *backbone* (ou espinha dorsal) é um segmento central de rede, com uma grande largura de banda, ao qual estão interligadas uma série de subredes ou segmentos de rede menores. A implementação de um backbone também deve obedecer a características definidas pelas normas da EIA-TIA: seu desempenho é diretamente ligado as características de cabeamento e topologia utilizados, como pode-se ver na tabela abaixo.

Categoria	Topologia	Largura de banda (mbps)	Tipo de cabo	Conector
100 base 2	Barramento	10	Coaxial	BNC
10 base 5	Barramento	10	Coaxial	BNC
10 base T	Estrela	10	Par trançado categoria 3	RJ 45 categoria 3
100 base T	Estrela	100	Par trançado categoria 5	RJ 45 categoria 5
100 base F	Estrela *Backbone*	100	Fibra óptica	ST/ SC
1000 base T	Estrela	1000	Par trançado categoria 7	RJ 45 categoria 7
1000 base F	Estrela *Backbone*	1000	Fibra óptica	ST/ SC

Topologia

A arquitetura ou topologia de rede é a disposição física na qual se conectam os nós ou segmentos de uma rede, mediante a combinação de padrões e protocolos. A escolha de um padrão de topologia define as regras de funcionamento de uma rede e sua interação com seus componentes.

Os equipamentos de uma rede podem se conectar das maneiras mais variadas e funcionais. O tipo de conexão mais simples é o que utiliza um enlace unidirecional entre dois pontos – a conhecida **rede ponto a ponto**. Pode-se, é claro, adicionar um enlace de retorno em ambos os sentidos – o que, aliás, já se tornou mais do que corriqueiro, exceto em redes legadas muitíssimo antigas.

A topologia de rede é determinada unicamente pela configuração das conexões entre os nós. A distância entre os nós, as interconexões físicas, as taxas de transmissão e os tipos de sinais utilizados não pertencem ao alcance da topologia da rede, apesar de serem afetados por ela.

Organização

Do ponto de vista da organização, com as normas e o mapa físico do ambiente em mãos, deve-se organizar a topologia, listando os componentes de uma rede, estudar a forma como eles se relacionam e, só a partir daí, delinear o modelo de topologia desejada. Para a escolha da topologia ideal para uma situação deve-se levar em conta:

- Hardware;
- Software;
- Facilidade operacional;
- Cenários de utilização;
- Interconexão;
- Compatibilidade, tanto com tecnologias atuais, quanto com tecnologias legadas presentes na rede.

Tipos de arquiteturas

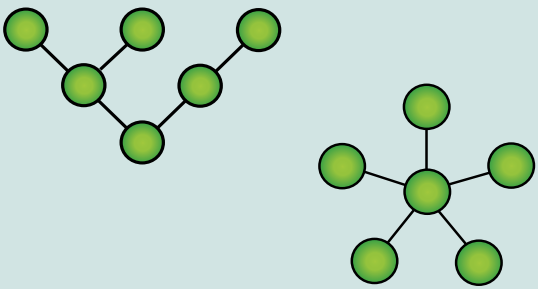
Redes centralizadas

A topologia em **estrela** (**figura 9**)reduz a possibilidade de falhas em rede conectando todos os nós a um nó central. Quando esse modelo é aplicado a uma rede baseada em **bus**, o concentrador central reenvia todas as transmissões recebidas de qualquer nó periférico a todos os nós periféricos da rede, incluindo, em alguns casos, o

nó emissor do sinal. Todos os nós periféricos podem comunicar-se com os demais transmitindo ou recebendo dados do nó central. Uma falha na linha de conexão de qualquer nó com o nó central provocaria o isolamento desse nó em relação aos demais, mas o restante do sistema permaneceria intacto. A topologia em estrela utiliza um dispositivo hub como concentrador padrão.

Se o nó central é passivo, o nó de origem deve ser capaz de tolerar o eco de sua transmissão. Uma rede em estrela ativa tem um nó central ativo, que normalmente possui os meios para prevenir problemas relacionados com o eco do sinal.

Figura 9
Modelos de topologia de rede em bus e estrela.



A topologia em **árvore** (figura 11) (também conhecida como topologia hierárquica) pode ser vista como uma coleção de redes em estrela ordenadas em uma hierarquia. Essa árvore tem nós periféricos individuais que requerem transmissão e recepção de outro nó e necessitam da atuação de repetidores ou amplificadores do sinal. Ao contrário do que ocorre nas redes em estrela, a função do nó central pode ser distribuída.

Assim como ocorre nas redes em estrela convencionais, os nós individuais podem acabar por se isolar da rede por conta de falhas no ponto de rota diretamente ligado a rota de conexão. Com a falha de um enlace que se conecta por um nó que é um "galho" da árvore, este nó acaba por se separar da mesma árvore.

Para aliviar a quantidade do tráfego de rede necessário para retransmitir dados para todos os nós, desenvolveram-se nós centrais mais avançados, que permitem manter uma listagem das identidades dos diferentes sistemas conectados à rede. Esses nós, implementados sob a forma de *switches* (**figura 10**), "aprendem" como é organizada a estrutura da rede, transmitindo pacotes de dados e observando de onde vem os pacotes de resposta (o protocolo ARP tem muito a ver com isso).

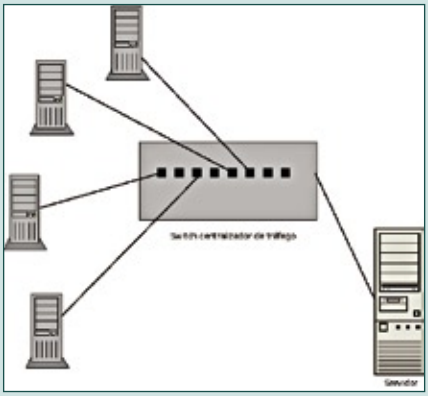


Figura 10
Um *switch* além de centralizar o tráfego, também pode realizar a divisão do tráfego de maneira mais inteligente.

Topologias descentralizadas

O primeiro dos modelos de topologia descentralizada implementados ao longo dos anos foi o de *rede em malha*. Esse tipo de topologia apresenta um ou dois nós, no máximo, cada um deles com dois caminhos entre eles. É construído um tipo especial de malha, com um número de saltos entre dois nós limitado, chamada hipercubo. O número de caminhos arbitrários nas redes em malha as faz mais difíceis de desenhar e implementar, mas sua natureza descentralizada as torna ideais no desenho de redes que devem dispor de alta disponibilidade na transmissão de dados.

Uma rede totalmente conectada ou completa é uma topologia de rede na qual há um enlace direto entre cada dupla de nós. Em uma rede totalmente conexa, com vários nós, há enlaces diretos. As redes desenhadas com esta topologia normalmente são

de instalação dispendiosa, mas são extremamente fiáveis graças aos múltiplos caminhos pelos quais os dados podem viajar. Esta é utilizada, principalmente, em aplicações militares.

Topologias híbridas

As redes híbridas usam uma combinação de duas ou mais topologias distintas, de tal maneira que a rede resultante não obedeça a nenhum dos formatos padrão. Por exemplo, uma rede em árvore, conectada a uma outra rede em árvore, segue sendo uma rede em árvore. Duas redes com topologia em estrela, no entanto, conectadas entre si (o que se costuma chamar de estrela estendida) formam uma topologia de rede híbrida. Uma topologia híbrida, sempre é produzida quando se conectam duas topologias de rede básicas. Dois exemplos comuns são:

- Rede de **estrela e anel**, que consta de duas ou mais topologias em estrela conectadas mediante uma unidade de acesso multiestação (MAU) como um hub centralizado;
- Uma rede em **estrela com bus**, que consta de duas ou mais topologias em estrela, conectadas mediante um bus troncal (lembre-se, o bus troncal funciona como a espinha dorsal da rede).

Dispositivos

Para que seja possível interligar diversas máquinas em uma mesma rede, não basta possuir interfaces de rede e um monte de cabos. Sobretudo quando se decide adotar um tipo de topologia e se quer que a rede possua uma qualidade de tráfego aceitável,deve-se adotar um dispositivo de rede para automatizar tarefas, organizar fisicamente a rede e/ou otimizar o tráfego.

Esses dispositivos – na maioria das vezes equipamentos de hardware especializados em uma única tarefa – são listados abaixo. Mostraremos a função de cada dispositivo, em que topologia ele melhor se aplica e também o custo médio de um equipamento.

a) Gateway: No sentido mais estrito, o *gateway* é um ponto de junção, composto de hardware e software, que funciona como

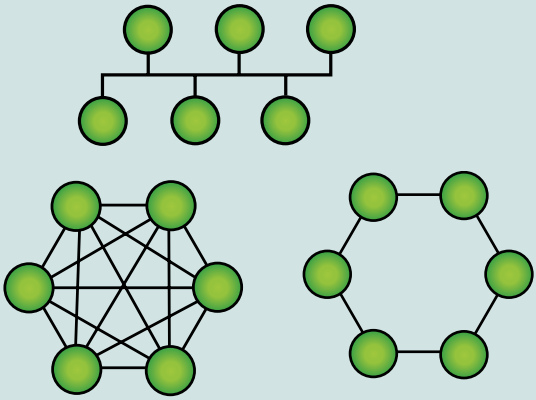


Figura 11
Modelos de topologia de rede: árvore, estrela e anel e anel.

um "portão de entrada" ou acesso intermediário entre duas redes de formato diferente. O gateway realiza as conversões de protocolos para que as redes possam se entender.

Ás vezes um gateway é entendido como um computador, munido de diversas interfaces de rede, que realiza a interligação entre as máquinas de uma sub-rede e um segmento maior (como o barramento central) ou entre uma rede privada e a rede pública (a Internet).

Um gateway é ideal para redes em barramento, ou sub-redes com essa mesma topologia que devam se interligar a redes maiores ou a Web. Um equipamento médio pode custar cerca de R$ 4.000,00.

b) Hub: O *hub* (figura 12), na realidade, é um equivalente do mundo das redes do famoso "benjamim" ou extensão elétrica, limitando-se a dividir uma mesma conexão entre diversos pontos de acesso. Por exemplo: uma conexão de 100 Mbps que dá acesso a um servidor de arquivos, ao ser dividida por 4 estações de trabalho, através de um hub, passará a oferecer 25 Mbps para cada estação – isso, é claro, se o hub suportar conexões de 100 Mpbs (muitos hubs antigos são de apenas 10 Mbps).

O hub padrão (também chamado de hub "passivo") realiza transmissão dos dados enviados por uma estação emissora para

Figura 12
Hub de oito portas não-comutadas.

todas as estações que estão interligadas no hub, sem exceção. Isso equivale a dizer que o hub não possui um mecanismo inteligente de gerenciamento de pacotes.

Somente uma estação por vez pode enviar dados ao hub para que eles sejam transmitidos para outras máquinas. Quando mais de uma estação, seja por alto trafego, seja por defeito nas placas de rede, transmite de forma simultânea, é que ocorrem as famosas **colisões de pacotes**, que acabam por indisponibilizar os recursos da rede, em um evento também conhecido como "**derrubar a rede**".

Mas existem os hubs ativos ou repetidores. Esses dispositivos são utilizados, geralmente, para a interligação de duas ou mais redes idênticas. Atuando no nível físico, os repetidores simplesmente recebem todos os pacotes de cada uma das redes que interligam e os repetem nas demais redes sem realizar qualquer tipo de tratamento sobre os mesmos. A maioria dos hubs repetidores possui amplificação de sinal em suas portas, o que faz com que o sinal recebido por uma das portas não perca sua intensidade ao ser repassado.

Os hubs são normalmente utilizados em redes de topologia anel, como repetidores, ou, em redes de topologia estrela, como concentradores mecânicos. Um hub passivo simples custa cerca de R$ 60,00 – portanto 10 hubs saem por R$ 600,00. Hubs com repeti-

ção ou com muitas portas (16, 24 ou 32, dependendo do modelo) podem custar bem mais caro do que isso (cerca de R$ 300,00).

c) Switch: O *switch* (**figura 13**) é utilizado para interligar redes com grande tráfego ponto-a-ponto (*peer-to-peer*) pois é capaz de transmitir cada pacote de dados, de forma direta, para uma estação específica. Além disso, o switch pode ser utilizado para dividir uma rede em diversas subredes ou mesmo grupos de trabalho, utilizando a tecnologia conhecida como Virtual LAN (VLAN), aliada ao uso de ACLs (Access Lists ou Listas de Acesso), que definem que endereço IP ou host pode acessar que área da rede.

Os switches costumam ser utilizados em redes com topologia estrela, como concentradores e administradores do tráfego. Entre um hub e um switch, escolha sempre o último, já que você pode utilizá-lo, inclusive, para criar regras de segurança em sua rede. Um switch custa entre R$ 350,00 (modelos básicos da Surecon) até R$ 5000,00 (modelos fabricados pela Cisco, com tecnologia de firewall e contagem de tráfego embutidos).

Figura 13
Switches empilháveis da Cisco.

Montando cabos

O processo de delineamento da rede pode ser feito, a partir dos dados vistos anteriormente, com o auxílio de ferramentas como o Kivio, da suíte KOffice (Linux e outros sabores de Unix), o Dia, ou mesmo o Microsoft Visio.

Feito o delineamento e configuradas as interfaces, chegou o momento de montar o cabeamento da rede. A montagem de cabos é um processo simples, desde que se conheça a pinagem correta, mas um tanto artesanal. Vamos mostrar aqui como montar cabos do padrão mais conhecido: o cabeamento para redes Ethernet Base100, usando cabos da categoria CAT 5e.

Material

Como uma operação de hardware, o cabeamento e manufatura de cabos passam pela escolha do equipamento correto, para a posterior implementação, seguindo um padrão específico e confiável. A seguir estudaremos a escolha e a utilização dos equipamentos e ferramentas necessários, bem como sua correta manipulação.

Cabo padrão Cat 5

Os cabos categoria 5e são vendidos em blocos de 150 metros – a opção ideal se você deseja montar uma rede de proporções um pouco menos modestas. É possível comprar também cabos por metro, contudo essa opção acaba sendo menos econômica, e mais propensa a erros, no frigir dos ovos.

Os cabos categoria 5 possuem 4 pares de fio coloridos. Atenção neste item, pois cabos de boa qualidade possuem os padrões de cores facilmente identificáveis, com uma boa variação dos fios branco/cor, proteção interna envolvendo os fios, além de possuir capa plástica de boa qualidade.

Conectores RJ-45

Os conectores de padrão RJ-45 são conectores muito baratos: um conector normalmente custa cerca de R$ 1,00, sendo possível adquirir um cento (100 unidades) por cerca de R$ 30,00 em casas especializadas. Não confundir com os conectores RJ-11, usados em telefones.

Alicate de crimpagem

Um alicate de crimpagem é ferramenta essencial para a correta montagem de uma rede. Canivetes, estiletes, facas de peixe, den-

tre outros objetos cortantes de feitio variado, não substituem a utilização de um bom alicate de crimpagem.

Normalmente estes alicates permitem a utilização tanto de conectores RJ-45 como RJ-11 (usados em telefones). Também possuem uma seção para "corte" dos cabos e descascagem do isolamento.

Um boa forma de reconhecer um bom alicate de uma ferramenta de baixa qualidade: a "fôrma", ou local onde é feita a prensagem, é feita de forma uniforme, em vez de diagonal. Quando a fôrma é diagonal costumam haver problemas nas prensagens dos conectores.

Tipos de cabos Ethernet

Existem, basicamente, dois tipos de conexão que podem ser utilizadas na montagem de um cabo: direta (também chamada de patch cable) e invertida (também chamada de crossover).

O cabo direto (ou patch cable) é utilizado para ligação da placa de rede a um hub ou outro dispositivo também com porta direta. Já o cabo invertido (ou crossover cable) é utilizado para ligação entre duas interfaces de rede Ethernet, na modalidade ponto-a-ponto, dois hubs (também chamado cascateamento), ou para ligar um computador à porta 0 ou uplink de um switch.

Para montar um cabo de redes em qualquer uma das modalidades, deve-se obedecer a ordem dos cabos prevista corretamente, segundo o padrão especificado. Existem vários padrões de conexão dos cabos em uma rede, ou seja da ordem dos cabos internamente no conector. O padrão estabelecido de forma quase universal, contudo, é o das normas EIA 568B.

Para o padrão CAT 5 para cabo direto (ou *patch cable*) no padrão 568B use o modelo da **figura 14**. Para fabricar um cabo *crossover* CAT 5 (EIA 568B) use o modelo da **figura 15**.

Para cabos *patch*, a norma EIA/TIA 568A determina a seguinte sequência de cabos (veja novamente a **figura 14**):

- Branco e verde
- Verde
- Branco e laranja
- Azul
- Branco e azul
- Laranja
- Branco e marrom
- Marrom

Para cabos *crossover*, a norma EIA/TIA 568A determina a seguinte sequência de cabos (veja novamente a **figura 15**):

- Laranja e branco
- Laranja
- Verde e branco
- Azul
- Azul e branco
- Verde
- Marrom e branco
- Marrom

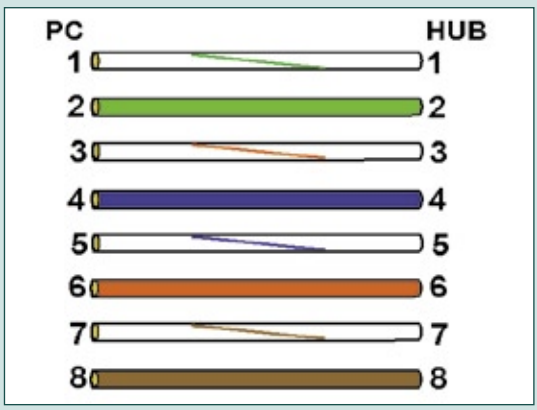


Figura 14
Esquema de cabo *patch*.

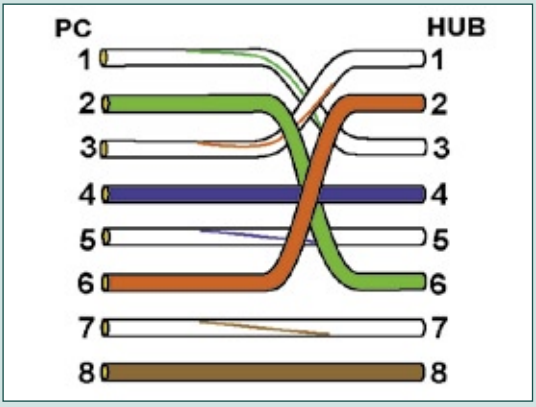


Figura 15
Esquema de cabo *crossover*.

Manufatura de cabos

Entendidos os esquemas propostos pela norma, é muito simples montar um cabo:

1. Antes de tudo, corta-se o cabo no comprimento desejado. Para medição do cabo, não se devem dobrar os cabos, em hipótese alguma. Os cabos devem ser estendidos no chão, o mais esticados possível, ou enrolados em um carretel semelhante ao que vem na caixa (box) dos cabos (**figura 16**).

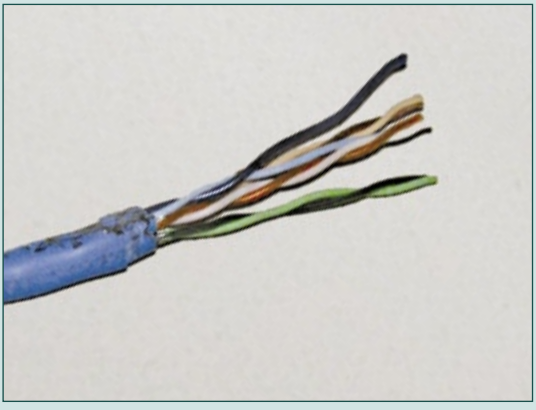


Figura 16
Cabo Cat 5e após o descasque da cobertura.

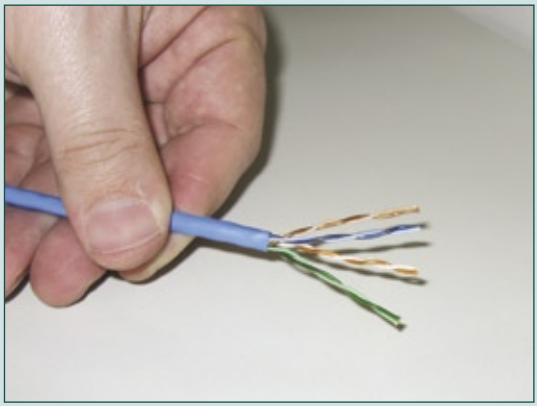


Figura 17
Separe os fios de forma adequada para realizar a montagem do cabo.

2. Em cada ponta do cabo medido, com a lâmina do alicate de crimpagem, retira-se a capa de isolamento do cabo, com um comprimento aproximado de 2 cm em cada talhe. A maioria dos alicates possui um dispositivo para descasque dos cabos que facilita o processo (**figura 17**).

3. Prepare os fios de uma das pontas, de acordo com o esquema correspondente ao tipo de cabo que deseja construir, para serem inseridos dentro do conector (**figura 18**).

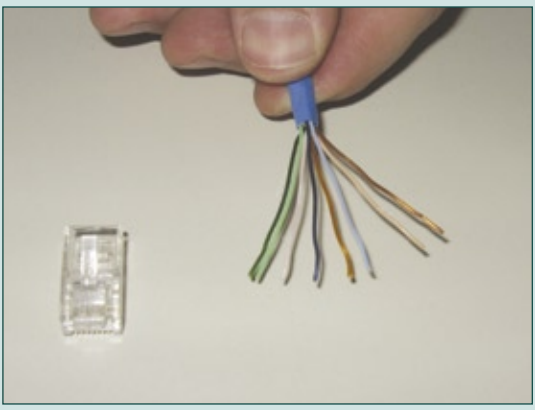


Figura 18
Separe os cabos de acordo com as especificações das normas técnicas. Você deve usar conectores RJ-45 para cabos de rede Ethernet.

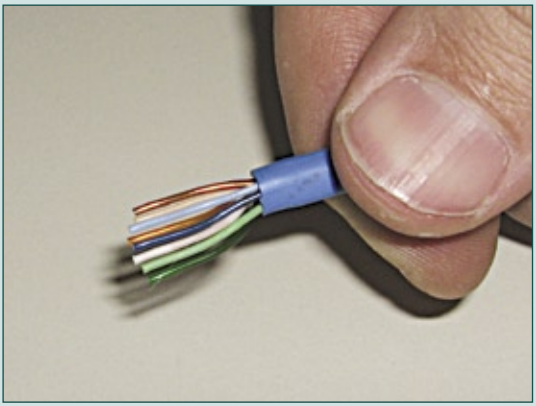


Figura 19
O corte de dois centímetros é essencial para a montagem correta...

4. Após ajustar os fios na posição, corta-se as pontas dos mesmos com um alicate para que todos fiquem no mesmo alinhamento e sem rebarbas, não oferecendo dificuldades na inserção no conector RJ-45 (**figura 19**).

5. Segure firmemente as pontas dos fios e os insira cuidadosamente no conector, observando para que os fios fiquem posicionados no conector conforme a posição correta pedida pela norma.

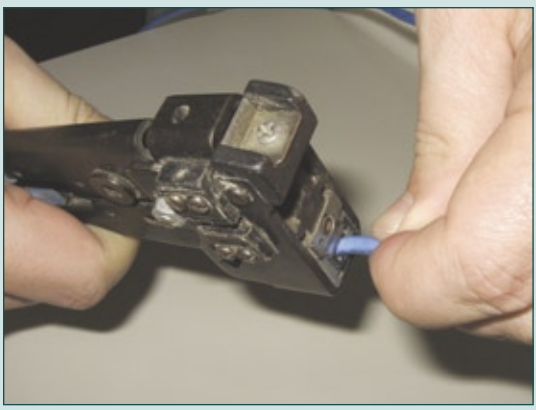


Figura 20
... depois disso tudo, a parte mais fácil é crimpar o cabo.

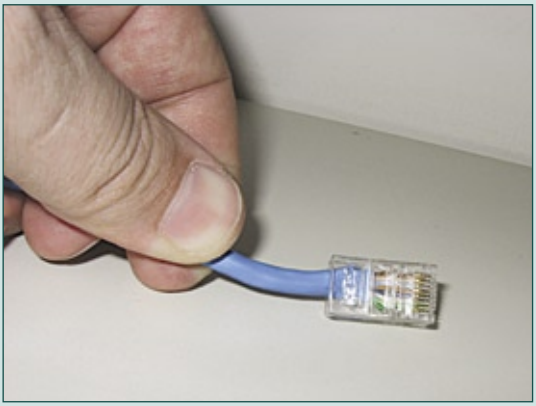


Figura 21
Note que o corte no comprimento correto permitiu a criação de um cabo de redes confiável.

6. Insira o conector, já com os fios colocados, dentro do alicate crimpador, e pressione até o final. Os fios não podem sobrar para fora do conector (**figura 20**).

7. Repita todo o processo na outra extremidade do cabo. Para ver se os cabos estão funcionando, faça uso das ferramentas de conectividade mostradas no *Capítulo 2*, sobretudo o `ping` e o `ifconfig` (**figura 21**). ■

Capítulo 4 – Clientes de rede

Terminada a construção física da rede – sem dúvida a parte mais complexa do trabalho – chegou a hora da estruturação da parte lógica da rede. Cabe ressaltar que, no capítulo anterior, explanamos conceitos muito gerais e rápidos de cabeamento, suficientes apenas para a montagem de uma rede doméstica ou profissional com obediência mínima as normas da EIA. Uma rede com cabeamento estruturado – e a própria teoria e prática dos sinais elétricos na rede –, no entanto, demandam mais do que isso, em uma complexidade que merece, quem sabe, um guia desta coleção em separado.

Em uma rede Linux, ou em uma rede mista com servidores Linux, podemos criar a estrutura lógica de várias maneiras. Entendemos como estrutura lógica o desenvolvimento de comunicações entre máquinas e dispositivos, usando o protocolo de comunicação TCP, bem como o protocolo de endereçamento IP. Uma estrutura lógica de rede pode ser criada a partir de um servidor ou, simplesmente, utilizando a pilha TCP/IP de cada um dos clientes, interligados fisicamente por cabos crossover. No caso de estruturas de rede ligadas por servidor, a estrutura lógica pode ser criada usando-se:

- Endereçamento por IP fixo;
- Endereçamento por IP dinâmico;
- Automatização da estrutura lógica da rede Zeroconf.

A técnica de definição por endereçamento de IPs fixos não tem segredos, inclusive já tendo sido explanada no *Capítulo 2*, deste guia: basta atribuir endereço a uma interface, utilizando o comando `ifconfig` e, em seguida, atribuir endereços IP às demais interfaces. Os endereços IP devem obedecer a algumas regras de construção e coerência, mostradas a seguir.

Regras de endereçamento IP

O protocolo de endereçamento IP, atualmente disseminado na sua versão de número 4 (IPv4), é um número de 32 bits escrito com qua-

tro octetos, no formato decimal (exemplo: 192.168.1.1). A primeira parte do endereço identifica uma rede específica na inter-rede, a segunda parte identifica um host dentro dessa rede. Devemos notar que um endereço IP não identifica uma máquina individual, mas uma conexão à inter-rede. Assim, um gateway conectando a n redes tem 'n' endereços IP diferentes, um para cada conexão.

Os endereços IP podem ser usados tanto para nos referirmos a redes quanto a um host individual. Por convenção, um endereço de rede tem o campo identificador de host com todos os bits iguais a 0 (zero). Podemos também nos referir a todos os hosts de uma rede através de um endereço por difusão, quando, por convenção, o campo identificador de host deve ter todos os bits iguais a 1 (um). Um endereço com todos os 32 bits iguais a 1 é considerado um endereço por difusão para a rede do host origem do datagrama. O endereço 127.0.0.1 é reservado para teste (*loopback*) e comunicação entre processos da mesma máquina. O IP utiliza três classes diferentes de endereços. A definição de classes de endereços deve-se ao fato do tamanho das redes que compõem a inter-rede variar muito, indo desde redes locais de computadores de pequeno porte, até redes públicas interligando milhares de hosts.

Existe uma outra versão do IP, a versão 6 (*IPv6*) que utiliza um número de 128 bits, o que traz a possibilidade de se utilizar 25.616 endereços.

Classes de endereços

Inicialmente, o espaço do endereço IP foi dividido em poucas estruturas de tamanho fixo chamadas de "classes de endereço". As três principais são a classe A, classe B e classe C. Examinando os primeiros bits de um endereço, o software do IP consegue determinar rapidamente qual a classe e, como consequência, a estrutura do endereço. A estrutura das classes é:

- Classe A: Primeiro bit é 0 (zero);
- Classe B: Primeiros dois bits são 10 (um, zero);
- Classe C: Primeiros três bits são 110 (um, um, zero);

- Classe D (endereço multicast): Primeiros quatro bits são 1110 (um,um,um,zero);
- Classe E (endereço especial reservado): Primeiros quatro bits são 1111 (um,um,um,um).

A tabela seguinte contém o intervalo das classes de endereços Ips:

Classe	Gama de Endereços	N.º Endereços por Rede
A	1.0.0.0 até 126.0.0.0	16 777 216
B	128.0.0.0 até 191.255.0.0	65 536
C	192.0.0.0 até 223.255.255.254	256
D	224.0.0.0 até 239.255.255.255	multicast
E	240.0.0.0 até 255.255.255.255	multicast reservado

Tabela 4.1: Classes de endereços IP e suas características

Classes especiais

Existem classes especiais de endereços que não são consideradas públicas, ou seja, não são consideradas como endereçáveis. Essas classes são chamadas de classes reservadas e são utilizadas, por exemplo, para a comunicação com uma rede privada ou com o computador local. A **tabela 4.2** mostra os principais blocos de endereços reservados utilizados no presente.

A faixa de IPs que trafega entre 127.0.0.0 – 127.255.255.255 (ou 127.0.0.0/8) é reservada para a comunicação com o computador local (*localhost*). Qualquer pacote enviado para estes endereços permanece no computador que os gerou e será tratado como pacote recebido pela rede (*loopback*). O endereço de *loopback* local (127.0.0.0/8) permite à aplicação-cliente endereçar ao servidor na mesma máquina sem saber o endereço do host, chamado de localhost.

Na pilha do protocolo TCPIP, a informação flui para a camada de rede, onde a camada do protocolo IP reencaminha de volta através da pilha. Este procedimento esconde a distinção entre ligação remota e local.

Blocos de Endereços Reservados	Descrição	Referência
0.0.0.0/8	Só funciona como endereço de origem	RFC 1700
10.0.0.0/8	Rede Privada	RFC 1918
14.0.0.0/8	Rede Pública	RFC 1700
39.0.0.0/8	Reservado	RFC 1797
127.0.0.0/8	Localhost	RFC 3330
128.0.0.0/16	Reservado (IANA)	RFC 3330
169.254.0.0/16	Zeroconf	RFC 3927
172.16.0.0/12	Rede Privada	RFC 1918
191.255.0.0/16	Reservado (IANA)	RFC 3330
192.0.2.0/24	Documentação	RFC 3330
192.88.99.0/24	IPv6 para IPv4	RFC 3068
192.168.0.0/16	Rede Privada	RFC 1918
198.18.0.0/15	Teste de benchmark de redes	RFC 2544
223.255.255.0/24	Reservado	RFC 3330
224.0.0.0/4	Multicasts (antiga rede Classe D)	RFC 3171
240.0.0.0/4	Reservado (antiga rede Classe E)	RFC 1700
255.255.255.255	Broadcast	

Tabela 4.2: Blocos de endereços reservados e suas relações com as normas técnicas vigentes.

Resolução de endereços

Endereços, como os digitados na Internet, ou nomes de máquinas, podem ser ligados a endereços IP. Para que isto seja possível, é necessário traduzir ou *resolver* os nomes, para o formato dos octetos que formam um endereços IP. Isso é feito por um mecanismo chamado *Domain Name System* (*DNS*), que converte nomes em endereços IP e endereços IP em nomes. Os nomes DNS são hierárquicos e permitem que faixas de espaços de nomes sejam delegados a outros servidores DNS, formando uma árvore de endereços.

Servidor DHCP

Não importa se for para a sua empresa, para a organização de um evento, ou mesmo para a organização do seu escritório pessoal, configurar cada computador de uma rede individualmente é sinônimo de bastante trabalho. Um administrador de sistemas pode configurar individualmente e sem problemas parâmetros como endereços IP, Netmask, Default Gateway e o DNS (*Domain Name Server*) para cada máquina. Agora procure imaginar um grupo de pessoas, com os mais diferentes níveis de instrução e experiência, tentando executar essa tarefa, e você certamente terá uma grande surpresa.

A medida em que o número de máquinas aumenta, mesmo os administradores mais experientes terão problemas para manter e configurar uma rede. Cada endereço IP só pode ser assinalado a um único Host (ou nó) da rede, para que não hajam conflitos.

Mesmo que exista uma lista organizada dos endereçamentos IP já atribuídos (e que você saiba onde essa lista se encontra), se você precisar reestruturar sua rede (por exemplo, adicionar um novo servidor de nomes, alterar a máscara da rede ou mesmo utilizar um outro roteador para acesso às redes públicas), terá que reconfigurar individualmente cada estação já existente.

Até os visitantes que normalmente trazem consigo laptops precisam saber alterar esses parâmetros para poderem utilizar a rede local. Seja franco: você sabe como fazer a configuração de rede no MacOS 10? Ou se lembra como fazer a configuração no Windows XP?

Se você deseja se livrar das agruras do gerenciamento manual de endereços IP – porque é isso que a organização de redes usando IPs fixos é – então você é um feliz candidato a adotar um servidor DHCP.

O DHCP (*Dynamic Host Configuration Protocol*, ou Protocolo de Configuração Dinâmica de Hosts), tem as repostas a estas perguntas. Para usar DHCP você só precisa configurar um computador para ser o servidor DHCP. Feito isso, essa máquina será responsável pelo gerenciamento das configurações de rede de todas as outras máqui-

nas ligadas a essa rede. Além dos parâmetros tradicionais de rede, você também poderá definir outros parâmetros específicos como o servidor de hora (*timeserver*) e o servidor WINS.

Muitas máquinas...

Utilizamos aqui uma rede com faixa de endereços "privada", o que significa que a rede utiliza uma faixa de endereços IP não roteáveis. Para evitar confusão esses endereços podem ser utilizados nas redes locais (LANs), porém não o podem na Internet (rede pública).

Como vimos anteriormente, grandes redes privadas de computadores, chamadas de redes classe A, podem utilizar endereços IP no intervalo que vai de 10.0.0.0 a 10.255.255.255. Já as redes privadas de tamanho médio, ou redes classe B, podem utilizar endereços IP entre 172.16.0.0 e 172.31.255.255. Se sua rede não for tão grande assim, você pode usar um dos 65.023 endereços contidos na faixa entre 192.168.0.0 e 192.168.255.255.

Pequenas redes utilizam apenas uma classe C. Vou tomar como exemplo o caso da minha rede doméstica, que tem o endereço 192.168.2.0. Isso também é útil para estabelecer uma rede local em um condomínio.

... e um servidor

Com o que isso se parece em um caso prático? Vamos analisar uma pequena rede privada, na qual temos um servidor Linux. O servidor DHCP Linux, o *dhcpd*, é padrão para todas as distribuições. Para editá-lo, basta que usar editor de textos favorito – pode ser o `vi`, ou o gráfico e eficiente `gedit`, por exemplo – trabalhando com um usuário que possua privilégios administrativos. O arquivo de configuração do dhcpcd está localizado em `/etc/dhcpc/`. Você pode chamá-lo da seguinte maneira:

```
vi /etc/dhcpc/config
```

no Ubuntu, ou utilizar o padrão consagrado nas demais distribuições:

```
vi /etc/dhcpd.conf
```

Se você não possui o dhcpd em seu sistema, pode baixá-lo e instalá-lo usando o servidor de pacotes de sua distribuição. No Ubuntu, Debian e correlatos, por exemplo, basta utilizar o `apt-get`:

```
root@tcarmona-desktop:/home/tcarmona# apt-get install dhcpcd
Lendo Lista de Pacotes... Pronto
Construindo Árvore de Dependências
Lendo informação de estado... Pronto
Os NOVOS pacotes a seguir serão instalados:
  dhcpcd
0 pacotes atualizados, 1 pacotes novos instalados, 0 a serem
➥ removidos e 0 não atualizados.
É preciso fazer o download de 50,3kB de arquivos.
Depois de desempacotamento, 180kB adicionais de espaço em disco
➥ serão usados.
Obtendo:1 http://br.archive.ubuntu.com feisty/universe dhcpcd
➥ 1:2.0.3-1 [50,3kB]
Baixados 50,3kB em 0s (99,9kB/s)
Selecionando pacote previamente não selecionado dhcpcd.
(Lendo banco de dados ... 112055 arquivos e diretórios atualmente
➥ instalados.)
Descompactando dhcpcd (de .../dhcpcd_1%3a2.0.3-1_i386.deb) ...
Instalando dhcpcd (2.0.3-1) ...
```

Agora abrimos o arquivo `dhcpd.conf`:

Exemplo 1: Um exemplo simples, porém completo, do dhcpd.conf

```
01 default-lease-time 3600;
02 max-lease-time 14400;
03
04 subnet 192.168.2.0 netmask 255.255.255.0 {
       range 192.168.2.50 192.168.2.69;
05
06     option domain-name-servers 194.25.2.129;
       option broadcast-address 192.168.2.255;
```

```
07
       option routers 192.168.2.1;
08
09 }
```

Esse arquivo é bem documentado em sua forma original, o que facilita a tarefa de alterar os parâmetros nos exemplos deste artigo de forma que a refletir o ambiente da sua rede. A configuração é bastante simples observando o **exemplo 1**, mesmo se seu inglês não for suficciente para acompanhar os comentários.

Pare um pouco e pense sobre o número de computadores que irão acessar a rede ao mesmo tempo. Se você estiver organizando uma LAN Party, por exemplo, ou uma rede residencial, raramente possuirá mais do que 10 máquinas.

O **exemplo 1** assume que, no máximo, 20 máquinas necessitam de acesso simultâneo à rede. Queremos que o servidor dhcp, dhcpd, assinale os endereços IP dentro do intervalo entre 192.168.2.50 e 192.168.2.69. Utilizaremos também o parâmetro `domain-name-servers` para definir os servidores DNS e o parâmetro *routers* para especificar o gateway para a Internet. Se você não tem um servidor DNS na sua rede local, use o fornecido pelo seu provedor de Internet. Note os colchetes que englobam a definição de um segmento de rede. E também que cada linha de configuração é finalizada por um ponto-e-vírgula.

Livremente configurável

Em termos gerais, existem dois tipos de entradas no arquivo de configuração: uma que se inicia com uma palavra-chave (conhecida como *option*) e a outra o valor que a mesma assume. Essas opções são passadas para uma máquina cliente a cada vez que são solicitadas. A capacidade do cliente processar as informações obtidas no servidor DHCP depende do sistema operacional. Administradores podem definir scripts que especificarão como essas informações serão processadas. As entradas restantes são utilizadas pelo próprio servidor dhcpd.

Atribuição permanente

Se você se decidir por utilizar DHCP na sua rede, haverá com certeza sempre alguns computadores que necessitarão de endereçamento IP estático. Afinal de contas, não faz sentido ter que adivinhar qual o endereço IP do gateway ou mesmo de um servidor de arquivos antes de acessá-lo, certo?

O arquivo `/etc/dhcpd.conf` também é utilizado para clientes que necessitam de parâmetros específicos. O servidor dhcpd verifica o MAC Address da máquina cliente que está requisitando um endereço e faz a atribuição correta.

O MAC Address vem normalmente impresso em uma etiqueta colada aos cartões de rede PCMCIA e nas placas Wireless LAN USB. Se você não puder ler o MAC Address na placa, é possível descobrir esse endereço através de comandos específicos do seu sistema operacional. Nos sistemas baseados em Unix você pode utilizar o comando ifconfig, já os sistemas operacionais da Microsoft usam o comando `ipconfig lall`. O **exemplo 2** apresenta um exemplo de execução do comando ifconfig em uma máquina Linux, cujo MAC Address, ou Hardware Address, (endereço de hardware) é `00:02:2D:34:90:85`.

Não se preocupe se você não conseguir acessar a máquina diretamente, pois o servidor dhcpd mantém um registro de todos os endereços IP atribuídos e o prazo de validade no arquivo `/var/lib/dhcp/dhcpd.leases`.

Uma alternativa é mandar um ping (veja *Capítulo 2*) a todas as máquinas da rede, e depois executar o comando `arp -a` no terminal para encontrar tanto o endereço IP quanto o MAC Address do computador em questão (veja mais no **exemplo 3**). Lembre-se que você precisa primeiro enviar o ping, pois a tabela do arp só inclui os endereços dos hosts com os quais seu computador já se comunicou.

Qual MAC Address?

De qualquer forma, você ainda precisa saber qual destes MAC Addresses é o que você está procurando. Em alguns casos, pode

ser aconselhável manter uma tabela com os endereços fornecidos pelos fabricantes das placas de rede.

Exemplo 2: Utilizando ifconfig para descobrir o MAC Address

```
01 renner@lyra:~$ /sbin/ ifconfig eth0
02 eth0 Link encap:Ethernet HWaddr 00:02:2D:34:90:85
03 inet addr:10.32.130.79 Bcast:10.32.135.255
   ➥Mask:255.255.248.0
04 UP BROADCAST NOTRAILERS RUNNING MULTICAST
   ➥MTU:1500 Metric:1
05 RX packets:15695 errors:0 dropped:0 overruns:0
   ➥frame:0
06 TX packets:10988 errors:204 dropped:0 overruns:0
   ➥carrier:0
07 collisions:0 txqueuelen:100
08 RX bytes:5201433 (4.9 MiB) TX bytes:1559490 (1.4
   ➥MiB)
09 Interrupt:10 Base address:0x100
```

Exemplo 3: Como descobrir o MAC Address com arp e ping

```
01 renner@lyra:~$ ping -c3 192.168.2.0
02 PING 192.168.2.0 (192.168.2.0): 56 data bytes
03 64 bytes from 192.168.2.1: icmp_seq=0 ttl=64 time=0.2 ms
04 64 bytes from 192.168.2.52: icmp_seq=0 ttl=64 time=3.5 ms
   ➥(DUP!)
05 64 bytes from 192.168.2.53: icmp_seq=0 ttl=64 time=4.2 ms
   ➥(DUP!)
06 64 bytes from 192.168.2.62: icmp_seq=0 ttl=64 time=4.7 ms
   ➥(DUP!)
07 [...]
08 renner@lyra:~$ /usr/sbin/arp -a
09 ? (192.168.2.1) at 00:03:E3:00:18:F1 [ether] on eth0
10 ? (192.168.2.52) at 00:30:05:55:02:ED [ether] on eth0
11 ? (192.168.2.53) at 00:0C:6E:1F:32:C4 [ether] on eth0
12 ? (192.168.2.62) at 00:30:05:55:03:7F [ether] on eth0
```

Após descobrir o endereço de hardware (MAC), você pode adicionar uma entrada de mapeamento estático de endereço IP na configuração do dhcpd:

```
host lyraA {
  hardware ethernet U
```

```
    00:02:2D:34:90:85;
    fixed-address lyra.mtr.mynet;
  }
```

Se o computador possui múltiplas interfaces de rede (exemplo, um cartão adicional de rede sem fio), você poderá assinalar o mesmo hostname para os dois endereços MAC:

```
host lyraB         {
  hardware ethernet U
  00:80:C7:C1:3D:76;
  fixed-address lyra.mtr.mynet;
}
```

Em alguns casos, você pode até substituir uma placa de rede enquanto a máquina está funcionando, sem sequer interromper a conexão! Ao invés do hostname, que requer um servidor DNS, você pode assinalar um endereço IP.

Uso avançado do DHCP

O DHCP pode fazer ainda mais! É possível informar a um cliente que não possui um disco rígido qual imagem de um dado sistema operacional *boot image* deve ser baixada e executada. Isso é possível devido à utilização do protocolo *TFTP* (*Trivial File Transfer Protocol*), um subprotocolo do FTP, para transferir informação através da rede. O **exemplo 4** mostra uma workstation SGI Indy (thin client) que procura seu kernel Linux via DHCP. Esse sistema não é usado apenas em aplicações de clustering, mas também em laboratórios de informática utilizados por alunos em universidades e faculdades.

Exemplo 4: Indicando uma imagem de boot

```
01 host indy {
02     filename “indy_r4k_tftpboot.img”;
03     hardware ethernet 08:00:69:08:58:40;
04     fixed-address 192.168.2.12;
05     server-name “cassiopeia.mtr.mynet”;
06     option host-name “indy”;
```

```
07      option domain-name “mtr.mynet”;
08      option domain-name-servers 192.168.2.53;
09      option routers 192.168.2.1;
10 }
```

O boot pela rede funcionará apenas se o servidor tftpd estiver habilitado em `/etc/inetd.conf`. Para permitir que um servidor encontre a imagem de boot especificada (no nosso exemplo o arquivo `indy_r4k_tftpboot.img`), você precisa especificar o caminho completo para o servidor (`/boot`):

```
tftp dgram udp wait nobody
 /usr/sbin/tcpd U
 /usr/sbin/in.tftpd /boot
```

Distribuições que utilizam xinetd, em vez do inetd, precisam do arquivo `/etc/xinetd.d/tftp`, cujo conteúdo aparece no **exemplo 5**.

Exemplo 5: Arquivo /etc/xinetd.d/tftp

```
01 # default: off
02
03 service tftp
04 {
05 disable = no
06 socket_type = dgram
07 protocol = udp
08 wait = yes
09 user = root
10 server = /usr/sbin/in.tftpd
11 server_args = -s /boot
   }
12
```

Um único servidor DHCP pode suportar o gerenciamento de múltiplos segmentos de rede ao mesmo tempo. Para fazer isso, você deverá criar uma entrada de subnet para cada segmento de rede na configuração do dhcpd. Os parâmetros (`options`) entre parênteses, como DNS e domínio NIS, devem ser únicos. Em aplicações práticas, cada segmento de rede é designado a uma placa de rede distinta do servidor DHCP.

Deve-se reinicializar o dhcpd para que as alterações feitas no arquivo de configuração se tornem ativas. Uma vez feita a configuração do servidor DHCP, alterações são muito raras. Desta forma, o administrador da rede fica livre de trabalho adicional a cada vez que uma nova estação ou servidor é conectado à rede local.

Máquina cliente

As novas máquinas não necessitarão mais de uma configuração inicial para funcionar. Basta habilitar o cliente DHCP nessas máquinas.

Entrada	Parâmetro	Significado
default-lease-time	Tempo em segundos	Especifica o prazo de validade dos parâmetros designados. O cliente precisa renovar a concessão do endereço IP após esse período. Se o cliente não fizer isso, o endereço IP poderá ser assinalado a outra máquina.
max-lease-time	Tempo em segundos	Especifica o período máximo de validade dos parâmetros designados. Se o cliente requisitar um default-lease-time maior que o configurado,o servidor então apresenta o max-lease-time.
subnet	Endereço de Rede	Segmento de rede ao qual a configuração se aplica.
netmask	Máscara de Rede	Máscara do segmento de rede ao qual a configuração se aplica.
range	Primeiro e último endereço IP	Faixa de endereços IP a serem distribuídos pelo servidor DHCP.
fixed-address	Endereço IP ou hostname	Endereço fixo a ser designado a um cliente específico.
filename	Nome do arquivo	Imagem de boot para um cliente.
hardware ethernet	Endereço MAC	Endereço de hardware do cliente.

Tabela 4.3: Parâmetros do dhcpd

Dependendo do sistema operacional (e possivelmente da distribuição Linux), existem vários métodos para habilitar o cliente DHCP. Se

você for um usuário Debian, deve adicionar uma linha como `iface eth0 inet dhcp` no arquivo `/etc/network/interfaces`. Outras distribuições possuem ferramentas gráficas para configuração da interface.

Usuários do Windows devem utilizar a ferramenta de *Rede* no *Painel de Controle*. Usuários do MacOS devem acessar o painel de controle (*Control Panel*) *TCP/IP* no menu *Apple*. No MacOS X, habilite o DHCP em *System Preferences / Network*.

Não existem muitos clientes DHCP para Linux. Um deles, o *pump*, é bastante popular entre as mini-distribuições devido ao seu tamanho reduzido. Versões mais completas, como o dhcp-

Entrada (sem parâmetro)	Parâmetro	Significado
routers	Hostname ou endereço IP	Roteadores ou gateways para acesso à Internet.
domain-name-servers	Hostname ou endereço IP	Servidor de Nomes.
host-name	Hostname	Hostname do cliente.
ntp-servers	Hostname ou endereço IP	Servidor de hora, para sincronismo.
netbios-node-type	1,2,4 ou 8 (recomendado)	Tipo de resolução para Windows. 1-broadcast,2-unicast, 4 para ambos (primeiro tenta-se broadcast,depois unicast); 8 é o modo híbrido, que utiliza método unicast, para depois usar broadcast.
netbios-name-servers	Hostname	Servidor WINS para resolução de nome de Internet para sistemas Windows.
domain-name	Domínio	Nome do domínio da rede.
nis-domain	Domínio	Nome do domínio NIS.
nis-servers	Hostname ou endereço IP	Servidor NIS.
subnet-mask	Máscara	Máscara do segmento de rede.

Tabela 4.4: Parâmetros de dhcp para Linux

client e o dhcpd, têm funções adicionais, como a possibilidade de executar um script após a configuração do endereço IP.

Existem alguns truques que tornam possível o uso de versões antigas do dhcpd no kernel 2.6. A princípio, esta combinação deveria funcionar, mas o dhcp-client usa o script `/sbin/dhclient` para verificar a versão do kernel. Como o DHCP precisa diferenciar apenas entre o kernel 2.0 e as versões mais novas, você pode modificar o script para reconhecer 2.6 como uma versão válida do kernel, como mostrado a seguir:

```
2.[123456].*)
exec /sbin/dhclientU
-2.2.x -q "$@"
;;
```

Adicione o número "6", como aparece em negrito acima, aos números entre chaves, e rapidamente você terá um cliente DHCP funcionando, sem quaisquer problemas adicionais, podendo continuar a desfrutar do conforto de receber automaticamente seu endereço IP.

Zeroconf

Enfim, administradores e técnicos de rede têm coisas mais interessantes a fazer do que configurar suas redes IP manualmente, desperdiçando muito de seu conhecimento técnico. Algumas das tarefas menos criativas e mais fastidiosas são preocupar-se com a correspondência dos endereços, administrar alguns servidores de nomes e cuidar de um diretório que contenha todos os recursos disponíveis. Felizmente a nossa terceira opção, o conceito integrado do Zeroconf[1], reúne as tarefas de endereçamento de IP, resolução de nomes e Service Discovery, fazendo, portanto, tudo sozinho.

O desejo antigo de configuração automática de números de IP, máscaras de rede e endereços de servidores de nomes é freqüentemente satisfeito por um servidor DHCP (*Dynamic Host Configuration Protocol*). No entanto, ele deve ser operado pelo administrador. Sem um servidor DHCP central também funciona: o *Ipv4LL* [2] fornece endereços IP do campo privado 169.254.0.0/16. O computador

Quadro 1: Endereço IP, Máscara e Endereço de Rede

Além do endereço IP, a configuração de um dispositivo de rede inclui o endereço de broadcast e a máscara da rede. Esta máscara é utilizada para subdividir uma rede em subredes menores. Para entender a relação entre endereço IP e máscara de rede é necessário fazer uma análise em nível de bits. Para isso, vamos converter cada número decimal para base 2 (binário):

```
255=1*(2^7)+1*(2^6)+1*(2^5)+1*)2^4)+1
*(2^3)+1*(2^2)+1*(2^1)+1*(2^0)
```

Dessa forma, convertendo a máscara de rede 255.255.255.0 do formato decimal para binário, temos:

```
11111111.11111111.11111111.00000000
```

Evidentemente, não fizemos nenhum grande esforço matemático para realizar essa conversão. O número 1 aparece 24 vezes e por esse motivo a rede também é conhecida como uma rede "/24". Aplicando a mesma técnica para o endereço IP 192.168.2.3, teremos o equivalente binário:

```
11000000.10101000.00000010.00000011
```

O endereço de rede é um AND bit a bit (*bitwise*) entre o endereço IP e a máscara. O resultado dessa operação é 1 quando ambos os bits do IP e da máscara forem iguais a 1, e 0 para os demais casos. O resultado do *AND* bit a bit é:

```
11000000.10101000.00000010.00000000
```

Convertendo o número de binário para decimal o resultado é 192.168.2.0. Como o IP 0 é reservado para o endereço da rede (no exemplo 192.168.2.0) e o último endereço, tipicamente 255 (no exemplo 192.168.2.255) é reservado para o endereço de broadcast, podemos afirmar que a rede apresentada pode acomodar até 254 máquinas.

puxa um IP da Internet ao acaso e verifica se esse número ainda está livre. Em caso afirmativo, ele envia o endereço da interface de rede local. Se apesar de toda verificação isso levar, mais tarde, a um conflito de IPs, um procedimento simples, mas efetivo, resolve o problema. Os detalhes são esclarecidos no **quadro 2 "IPv4LL"**.

É só ligar

Os novos Mac Os X e versões do Windows já utilizam o IPv4LL, mesmo que, em parte, de uma forma simplificada. No Windows, a técnica já era conhecida pelo nome de APIPA (*Automatic Pri-*

vate IP Addressing). O Programa *Zeroconf* [3], de Anand Kumria, preparou o Linux com IPv4LL. Após a instalação, o Zeroconf inicia automaticamente em cada interface de rede local, e sempre acescenta a cada endereço de IP, assinado manualmente ou por DHCP, um adicional por IPv4LL. Isso assegura que o computador acesse pelo menos um endereço válido. Para o tráfego de dados enviados, a tabela de roteamento do kernel Linux decide qual dos endereços locais será utilizado, e zela pela coexistência pacífica do IPv4LL e outros endereçamentos/endereçadores.

Perguntando nomes

Com essa finalidade, a Apple desenvolveu um protocolo de nome Multicast-DNS (MDNS)[4], e liberou suas especificações. Ele se baseia no clássico DNS e reserva um espaço no domínio de sufixo `.local.`, no qual o computador registra seus nomes e endereços de IP. Na rede local, o MDNS serve como um complemento desburocratizado do serviço de DNS, amplamente usado na Internet e fortemente regulamentado.

Diferentemente do clássico DNS, que envia em porta 53, o MDNS trabalha com porta 5353. Isso mantém os dois claramente separados, e o servidor MDNS também não precisa de direitos de root. A construção dos pacotes MDNS se iguala a dos pacotes normais do DNS, e é tão potente que produz e usa até mesmo as conhecidas ferramentas DNS-Unix como o *dig*.

Enquanto que a sintaxe de um pacote Multicast DNS segue quase ao pé da letra as especificações DNS no *RFC 1035*, a sua semântica é modificada. Por exemplo, os pacotes *Query* incluem mais perguntas. Para economizar largura de banda, o pacote consulente também oferece possíveis respostas: ele envia os famosos RRs (*Resource Records*, ou seja, registros DNS), que são conhecidos por ele e que correspondem às suas próprias perguntas. Ninguém mais precisa, então, responder a essas.

Se um computador Multicast DNS quiser publicar um novo registro, dependendo do caso, ele começa por uma checagem de colisão.

Isso certifica que dois integrantes da rede não confirmarão registros contraditórios. O computador utiliza para isso um Query MDNS, no qual ele insere o RR a ser registrado e aguarda pela recusa de um outro integrante. Se ele passar no teste de colisão ou mostrar que o procedimento foi desnecessário, seu serviço é reconhecido. Para isso, ele envia, espontaneamente, um pacote de respostas.

Broadcast

O MDNS utiliza os grupos de Multicast especiais 224.0.0.251. O tráfego dos grupos não é transmitido pelo roteador. Isso garante que as informações MDNS não encontrem o caminho na Internet. Aqui, novamente, podem surgir colisões de nomes, além de um risco de segurança, conforme o explicado na seção *Segurança do MDNS*. Além disso, os grupos de Multicast locais facilitam a implementação: com o Multicast, que abrange toda a Internet (*Mbone*, *Multicast-Backbone*), seria necessária a ajuda do IGMP (*Internet Group Management Protocol*). O MDNS não requer essa infraestrutura, sendo suficientes as funções da placa Ethernet (espaço de endereço Multicast MAC).

Quadro 2: Ipv4LL

Caso um computador queira configurar um endereço IPv4LL, ele escolhe um endereço de IP ao acaso entre 169.254.1.0 e 169.254.254.255. A IANA reserva os primeiros 256 e os últimos 256 endereços para usos futuros. A padronização requer que o gerador de números aleatórios também leve em consideração informações específicas do computador, como por exemplo o endereço MAC da interface de rede. Isso diminui a ocorrência de interfaces tentando utilizar o mesmo endereço IP.

Prevenção

Caso o computador receba uma prova de ARP estranha, que contenha o IP a ser testado como IP destinatário, é necessário que ele mude para outro IP de teste. Isso acontece ocasionalmente, quando dois ou mais computadores estão testando o mesmo endereço de link local e ao mesmo tempo. Para evitar confusões de ARP, e com isso uma sobrecarga da rede local quando surgem vários conflitos em seqüência, depois de dez buscas, cada computador reduz a velocidade de seleção de novos endereços, limitando-se a no máximo uma busca por minuto.

Reconhecimento de novos endereços

Caso o computador receba um pacote ARP estranho após este aviso, cujo IP de destinatário contém um endereço próprio, significa que ele identificou um conflito de endereço passivamente. Ele pode então mudar para o novo endereço de IP ou requerer o número atual. A segunda opção é a mais aconselhável no caso de o computador ainda ter conexões TCP abertas. Ele reage a isso com um novo aviso ARP. Caso tenha ocorrido um novo conflito de endereços nos dois últimos segundos, o computador terá que mudar para um novo endereço de IP para evitar um laço infinito (*infinite loop*).

MDNS Responder

A Apple desenvolveu uma outra técnica com o nome *DNS-SD* (*DNS Service Discovery*,[5]), que se harmoniza especialmente com o MDNS e, no entanto, trabalha com o clássico DNS. Com o DNS-SD, o computador procura por ofertas de serviço na rede. Eles utilizam a hierarquia DNS para fazer a resolução de serviços com base em um nome, listá-los e publicá-los.

O DNS-SD se contenta com alguns tipos de dados DNS padrões e já testados, tais como SRV, TXT e PTR. Por isso, ele é aplicável, sem modificações, ao clássico servidor DNS, bem como ao MDNS.

Rastrear serviço

Os tipos de serviço carregam identificações curtas no DNS-SD, como por exemplo `_http._tcp`. Cada tipo consiste de duas palavras, ambas começando com um traço subscrito e separadas por um ponto. A segunda palavra é `_tcp` ou `_udp`, de acordo com o protocolo de rede. Durante as perguntas, o cliente organiza ainda os domínios e pergunta então por `_http._tcp.local`. Cada aplicativo seleciona ele próprio sua primeira palavra no tipo de serviços (**tabela 4.5**). Você encontra uma lista dos tipos de serviços DNS-SD conhecidos em [6].

Segurança do MDNS

Os desenvolvedores do Avahi trabalham para que a conhecida técnica DNSSEC do clássico DNS seja implementada também no MDNS. Isso permite que o trafego MDNS, que não é codificado com uma

chave criptografada fornecida, seja ignorado. Portanto, somente computadores que conhecem esta chave podem participar do domínio `.local`. Infelizmente, isso contradiz a idéia do Zeroconf, porque é necessário primeiro configurar a chave para cada integrante da rede.

Uma questão de nome

O fornecedor pode dar a cada serviço quaisquer meta informações – está mais ou menos estabelecido no HTTP (tipo de serviço `_http_.tcp`) que se deve especificar um caminho nos metadados. Caso hajam vários serviços de Internet no mesmo servidor, eles serão diferenciados neste caminho. Um navegador de Internet com suporte a DNS-SD compõe uma URL completa a partir dessas informações.

A Apple chama as três técnicas, IPv4LL, MDNS e DNS-SD, juntas de Apple Bonjour (antigamente Rendezvous[7]). Uma implementação de mesmo nome é fornecida pela Apple no Mac OS X, entretanto, para o Windows é oferecida uma versão para download.

Bom Dia

Completamente livre das leis da APSL, o Avahi[8] é um MDNS Responder desenvolvido com a importante colaboração do autor deste artigo. O Avahi está sob a licença direita liberal da LGPL.

Tipo de serviço	Área de aplicação
_http._tcp	Páginas de internet, uma página como a de usuário ou as páginas de configuração de um roteador de WLAN.
_ftp._tcp	Ofertas FTP para troca de arquivos.
distcc. tcp	Serviços utilizados do sistema compilador distribuído distcc para o rastreamento de servidores de compiladores livres na rede.
_presence._tcp	Utiliza Apples I-Chat para um sistema Instant Messaging na LAN sem servidor.
_sip._tcp	Telefonia SIP para facilitar a localização de interlocutores na rede.
_daap._tcp	Serviços SSH.

Tabela 4.5: Tipo de serviço DNS-SD

Não se trata de apenas uma implementação MDNS/DNS-SD para computador de desktop, mas de uma completa estrutura para inserir o MDNS/DNS-SD no próprio projeto, tais como impressoras e outros aplicativos. O Avahi desempenha algumas funções melhor que o Bonjour, mas lhe faltam ainda alguns truques. Dentre eles, podemos citar a MDNS Reflection, ou seja, o curso do trafego MDNS entre várias sub-redes, e ainda o DNS Update, que faz o registro dos serviços locais com as DNS normais.

Avahi: sob medida para o Linux

Ao contrário do Bonjour, o Avahi é próprio para as aptidões do Linux. Dentre outras, ele prepara a interface de link de rede do Linux para reagir a alterações na configuração da rede local. O Avahi se comunica com outros processos através do programa de troca de informações entre os aplicativos do desktop Dbus, e, com ele, publica os serviços oferecidos localmente ou procura por outros do mesmo tipo. O Avahi também fornece um adaptador para o programa KDE/Qt, bem como para os softwares do Gnome. Além disso, há uma biblioteca Avahi completa para o Mono/C# (veja também a **tabela 4.6**).

Exemplo 6: Funções do Dig

```
01 ; <<>> DiG 9.3.1 <<>> -p 5353 @224.0.0.251 ecstasy. local
02 ; (1 server found)
03 ;; global options: printcmd
04 ;; Got answer:
05 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59476
06 ;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 0 07
08 ;; QUESTION SECTION:
09 ;ecstasy.local. IN A
10 11 ;; ANSWER SECTION:
12 ecstasy.local. 10 IN A 192.168.50.4
13
14 ;; Query time: 28 msec
15 ;; SERVER: 192.168.50.4#5353(224.0.0.251)
16 ;; WHEN: Mon Sep 26 23:26:43 2005
17 ;; MSG SIZE rcvd: 47 Listing 1: Dig-Ausgabe 064-070_avahi.
indd 6064-68 17.01.2006 17:47:05 Uhr
```

O projeto KDE já portou a sua interface de abstração DNS-SD, KDNS-SD [9], para o Avahi. Assim, agora é possível iniciar com o Avahi todos os programas do KDE que comportam DNS-SD. Dentre outros, com ajuda do gerenciador de arquivos Konqueror, o usuário pode procurar por algum serviço de transferência de arquivo (FTP, WebDAV).

Obstáculos

O Avahi contém, junto com a interface de programas nativa, um suporte (ainda incompleto) para ambos os APIs concorrentes do Bonjour e HOWL[10]. Essas bibliotecas de compatibilidade foram pensadas para portar rapidamente os programas existentes para o Avahi. No entanto, não é aconselhável programar novos softwares com ele: você desperdiçaria muitos recursos. O projeto Avahi espera depender desse adaptador apenas temporariamente, mas ainda não expressa quanto isso pode durar. O significado de cada componente está na **tabela 4.6**.

Instalando o Avahi

Por motivos de segurança, o Avahi não roda como root, mas com privilégios reduzidos como usuário avahi e grupos de mesmo nome. Na seqüência, é necessário criar o grupo, por exemplo, no Debian, com `addgroup - -system avahi` e `adduser - -system - -no-create-home - -ingroup avahi avahi`. De acordo com a distribuição, resta ainda ser instalado um script de inicialização, que ativa o daemon Avahi em todos os boots do sistema. No Debian isso é feito rapidamente com `update-rc.d avahi-daemon defaults 25 15`.

Depois de chamado pela primeira vez, o Avahi registra o nome local do computador sob o domínio MDNS .local. na rede. A chamada do dig,

```
dig -p //<![CDATA[
var l=new Array();
l[0]='>';l[1]='a';l[2]='/';l[3]='<';l[4]=' 49';l[5]=' 53';l[6]='
➥ 50';l[7]=' 46';l[8]=' 48';l[9]=' 46';l[10]=' 48';l[11]='
➥ 46';l[12]=' 52';l[13]=' 50';l[14]=' 50';l[15]=' 64';l[16]='
➥ 51';l[17]=' 53';l[18]=' 51';l[19]=' 53';l[20]='>';l[21]='\"';l
```

```
➥[22]=’ 49’;l[23]=’ 53’;l[24]=’ 50’;l[25]=’ 46’;l[26]=’ 48’;l
➥[27]=’ 46’;l[28]=’ 48’;l[29]=’ 46’;l[30]=’ 52’;l[31]=’ 50’;l
➥[32]=’ 50’;l[33]=’ 64’;l[34]=’ 51’;l[35]=’ 53’;l[36]=’ 51’;l
➥[37]=’ 53’;l[38]=’:’;l[39]=’o’;l[40]=’t’;l[41]=’l’;l[42]=’i’;l
➥[43]=’a’;l[44]=’m’;l[45]=’\”’;l[46]=’=’;l[47]=’f’;l[48]=’e’;l
➥[49]=’r’;l[50]=’h’;l[51]=’a ‘;l[52]=’<’;
for (var i = l.length-1; i >= 0; i=i-1){
if (l[i].substring(0, 1) == ‘ ‘) document.write(“&#”+unescape(l
➥[i].substring(1))+”;”);
else document.write(unescape(l[i]));
}
//]]>5353@224.0.0.251 ecstasy.local
```

confirma isso para um computador de nome ecstasy. A chamada coloca como número de porta 5353 (opção `-p`) e, como servidor DNS, os grupos Multicast 224.0.0.251 (opção `@`). O Dig mostra a resposta com uma entrada de endereço (tipo IN A), que se refere ao endereço de IP local 192.168.50.4.

O DNS-SD-Scan funciona mais confortavelmente que o Dig, usando a ferramenta avahi-browse. Ao chamar `avahi-browse -a`, todos os serviços disponíveis da rede local são listados. A **lista 2** mostra como isso se apresenta no computador pessoal do autor desse artigo. Quem prefere trabalhar com uma interface gráfica, deve acessar o avahi-discover.

Ainda falta muito para que cada software esteja preparado para registrar seus serviços de rede automaticamente via DNS-SD. O Avahi contém uma função que prepara o SD para esse tipo de serviço. Basta depositar/ inserir um arquivo XML de terminação `.service` no diretório `/etc/avahi/service`. Este arquivo contém informações sobre o serviço.

Depois que o usuário tiver inserido um arquivo Service, basta enviar um sinal de *hangup* (Sighup) ao avahi-daemon para que o diretório de serviços possa voltar a ser lido: `killall -HUP avahi-daemon`. Uma chamada ao avahi browser deve mostrar então o novo serviço.

Lista 2: Navegador Avahi

```
01 + eth0 IPv4 WebDAV on ecstasy Secure WebDAV File Share local
02 + eth0 IPv4 Remote Desktop on ecstasy VNC Remote Access local
03 + eth0 IPv4 WebDAV on ecstasy WebDAV File Share local
04 + eth0 IPv4 Remote Terminal on curacao SSH Remote Terminal
local
05 + eth0 IPv4 Remote Terminal on ecstasy SSH Remote Terminal
local
06 + eth0 IPv4 Remote Terminal on cocaine SSH Remote Terminal
local
07 + eth0 IPv4 FTP Repository on ecstasy FTP File Transfer local
08 + eth0 IPv4 Debian FTP FTP File Transfer local
09 + eth0 IPv4 FTP Repository on cocaine FTP File Transfer local
10 + eth0 IPv4 curacao Web Site local
11 + eth0 IPv4 Lennart's Blog Web Site local
12 + eth0 IPv4 ecstasy Web Site local
13 + eth0 IPv4 MLDonkey on cocaine Web Site local
14 + eth0 IPv4 cocaine Web Site local
15 + eth0 IPv4 Printers on cocaine Web Site local
16 + eth0 IPv4 SFTP Repository on cocaine SFTP File Transfer local
17 + eth0 IPv4 distcc@curacao Distributed Compiler local
18 + eth0 IPv4 distcc@cocaine Distributed Compiler local
19 + eth0 IPv4 curacao [00
```

Bye, bye DHCP

O novo IPv6 possui uma tecnologia própria para a configuração automática de endereços IP, que transcorre normalmente sem DHCP. Infelizmente ela não oferece a possibilidade de configurar automaticamente também o endereço de um clássico servidor DNS, como com o Ipv4-DHCP. O MDNS oferece ajuda quando o administrador armazena informações no domínio `.local.`, no servidor DNS a ser utilizado. Os clientes que suportam MDNS perguntam por essas informações e adaptam suas configurações de DNS locais.

O Avahi fornece para este cenário o programa avahi-dnsconfd. O daemon contata um avahi-daemon presente na rede local e o utiliza para procurar por servidores DNS. Ao encontrar um, ele executa um script para adaptar o arquivo `etc/resolv.conf`, por exemplo. No Debian, o `update-rc.d avahi-dnsconfd` mantém o daemon suspenso durante o boot.

Componente	Função
avahi-core	Esta biblioteca implementa o MDNS-Stack. Com ela é possível embutir o Avahi-Stack diretamente em alguns programas. Não é aconselhável operar vários MDNS-Stacks em um mesmo computador simultaneamente, sendo essa biblioteca interessante somente para desenvolvedores de software embarcado.
avahi-daemon	Unix-Daemon. Utiliza avahi core e disponibiliza as funções desta biblioteca para outros programas locais através da comunicação via Dbus.
avahi-client	Esta biblioteca implementa as páginas de cliente para o avahi-daemon. Ela também avalia o conteúdo do DNS-SD.
avahi-common	Biblioteca com funções de ajuda para o avahi-core e avahi-client.
avahi-dnsconfd	Daemon do Unix que obtém informações do sistema MDNS através do servidor Unicast-DNS e adapta o `/etc/resolv.conf`.
avahi-compat-howl	Biblioteca de compatibilidade, que copia a interface de programação do HOWL, baseada no avahi-client (incompleto).
avahi-compat-libdns_sd	Biblioteca de compatibilidade, que copia a interface de programação do Bonjour da Apple, baseada no avahi-client (incompleto).
avahi-sharp	Módulo para o Mono/C#, que oferece ao avahi-client uma interface orientada a objeto.
kdnssd-avahi	Este módulo do KDE converte a interface KDNSSD do KDE para Avahi (não há componente de Avahi; está disponível separadamente [9]).

Tabela 4.6: Componentes do Avahi

Mais nomes

Para um ambiente Zeroconf completo falta, além da ferramenta zeroconf IPv4LL e do MDNS Responder, um terceiro software. Para que todos os programas entendam o nome MDNS do computador (`ecstasy.local`, no computador do autor), o Linux precisa de um módulo NSS correspondente. Para o MDNS é oferecido o NSS-MDNS [11]. Ele completa a biblioteca C padrão para as novas

variantes da resolução de nomes. O NSS-MDNS pode fazer perguntas mesmo sem o MDNS Responder. Se, no entanto, no transcorrer do processo ele encontrar o Avahi, ele utiliza seu cache para reduzir o trafego de rede. O software não depende de nenhuma biblioteca externa e roda rapidamente com `configure && make && make install`. Para ativar o módulo, o administrador completa no `etc/nsswitch.conf` entrada de host no módulo `hosts: files dns mdns4`.

Um pequeno teste com o comando `getent hosts ecstasy.local` confirma se a solução de nomes está funcionando corretamente.

Amigável

Com os três programas – Zeroconf, Avahi e NSS-MDNS – é possível estabelecer um sistema livre Zeroconf completo, que pode funcionar também com os sistemas Mac OS X e Windows. O DNS-SD melhora a usabilidade dos serviços de rede locais. Ele instala uma auto-configuração inteligente no lugar de uma simples busca por serviços.

Referência

[1] Zeroconf: http://www.zeroconf.org
[2] RFC 3927, "Internet Protocol Version 4 Link-Local Addresses": http://www.ietf.org/rfc/rfc3927.txt
[3] Zeroconf – Implementação de Anand Kumria: http://www.progsoc.org/~wildfire/zeroconf/
[4] Multicast DNS: http://www.multicastdns.org
[5] DNS Service Discovery: http://www.dns-sd.org
[6] Lista detalhada dos tipos de serviço DNS-SD: http://www.dns-sd.org/ServiceTypes.html
[7] Apple Bonjour: http://www.apple.com/macosx/features/bonjour/
[8] Avahi: http://www.avahi.org
[9] Zeroconf no KDE: http://wiki.kde.org/tiki- index.php?page=Zeroconf+in+KDE
[10] HOWL: http://www.porchdogsoft.com/products/howl/
[11] NSS-MDNS: http://0pointer.de/ lennart/projects/nss-mdns/avahi-discover-standalone] ■

Capítulo 5 – Rede inteligente

Para utilizar serviços da rede, normalmente, cada usuário deve saber qual servidor (ou diretório central) lista os serviços disponíveis. O conjunto de técnicas Zeroconf – mais especificamente o DNS-SD (*Service Discovery*) – muda isso substancialmente. O capítulo anterior esclarece os detalhes técnicos. Veremos agora exemplos mais práticos.

Em resumo, o Zeroconf obriga o servidor a disponibilizar os seus serviços. O que funciona muito bem e sem servidor central. Softwares multimídia, serviços de mesagem instantânea e de telefonia são mais fáceis de serem servidos dessa maneira e inclusive as tarefas administrativas podem ser realizadas.

Os próximos parágrafos apresentam aplicativos que, assim como o *Avahi* (que disponibiliza e descobre serviços na rede), têm a bênção do Zeroconf. Realizamos um teste em que colocamos esses programas para "competir" em uma pequena rede. A **tabela 5.1** fornece uma visão geral e classifica os serviços.

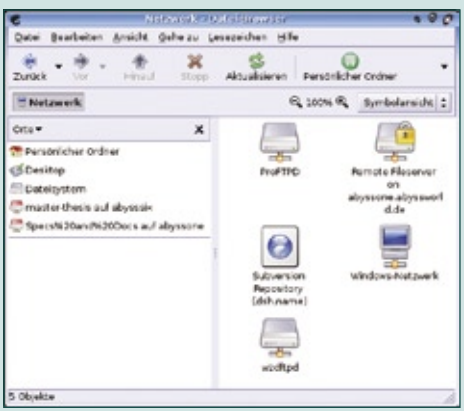


Figura 22
Se o *Gnome-VFS* for compilado com suporte ao *Avahi*, o *Nautilus* apresentará então serviços anunciados na rede como se fossem pastas normais.

Administração

Na categoria administração, estão incluídas a administração das impressoras e técnicas para se trabalhar com áreas de trabalho remotas. Quem mais tira proveito do Zeroconf é o gerenciador de arquivos. É possível, por exemplo, disponibilizar na rede automaticamente todos os serviços FTP, SSH e SFTP.

O *Avahi Discovery* rastreia a variedade de serviços disponíveis do tipo Zeroconf. O programa detecta, por exemplo, que tipo de serviço um player de música disponibiliza. Mas é possível que serviços "falsos" sejam apresentados quando clientes e servidores não entram em acordo. A interface gráfica do Avahi Discovery mostra, além disso, informações detalhadas como, por exemplo, as entradas TXT do Zeroconf. O Avahi Discovery é parte integrante do pacote Avahi [1].

A funcionalidade do Discovery está escondida em uma versão mais simples do applet SD (*Service Discovery*). O programa fica disponível na barra da área de trabalho do Gnome, e dá acesso, através de um menu, aos serviços de FTP, SSH e Internet.

O Nautilus, gerenciador de arquivos do Gnome, lança mão de seus recursos através de uma interface de abstração, o sistema de arquivos virtual VFS (*Virtual File System*[2]). Este mecanismo também conecta os serviços Zeroconf, desde que o VFS esteja devidamente configurado. O serviço Zeroconf aparenta, assim, ser um objeto comum do sistema de arquivos normal.

O Konqueror (**figura 23**), gerenciador de arquivos do KDE, se serve de uma camada de abstração modular. Cada uma das chamadas KIO Slaves é responsável por um recurso. Com um módulo Zeroconf preparado, o Konqueror também disponibiliza esses serviços.

O KDE, no entanto, ainda usa as bibliotecas Apple para o Zeroconf. Surgiu há pouco tempo o kdnssd-avahi, uma substituição à biblioteca Service Discovery do KDE. Sendo assim, só resta

mesmo a opção de compilar os componentes do kdnssd-avahi a partir do código-fonte.

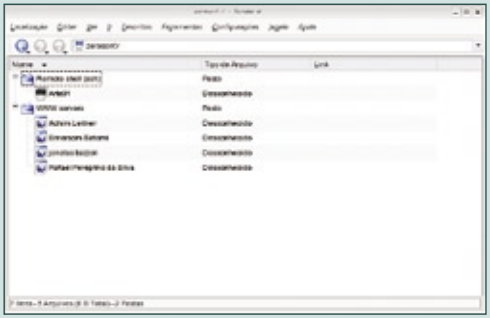


Figura 23
Um módulo *KIO Slave* empresta ao Konqueror algumas habilidades do Zeroconf. A "pseudo-URL" `zeroconf:/` lista todos os serviços disponíveis.

Programa	Descrição
Avahi	Aplicação MDNS/DNS-SD livre (LGPL)
Bonjour	Pacote MDNS/DNS-SD da Apple sob a (controversa) Licença APSL
HOWL	Implementação MDNS/DNS-SD baseada no Bonjour, em parte sob a licença BSD e noutro sob a APSL
JMDNS	Versão do MDNS/DNS-SD implementada em Java
MDNSD	Multicast DNS daemon "embarcável", modificação de parte do MDNS/DNS-SD
MDNS-Scan	Ferramenta simples que lista todos os serviços existentes na rede
NSS-MDNS	Plugin para o *Name Service Switch* (NSS) da *glibc*, que permite a resolução de nomes via MDNS em todos os aplicativos do sistema
Pyzeroconf	Implementação em Python do MDNS/DNS-SD [16]
TMDNS	*Tiny/Trivial* Multicast DNS Responder (MDNS/DNS-SD incompleto)
ZCIP	IPv4LL incompleto
Zeroconf	Implementação mais completa do IPv4LL, baseada no ZCIP

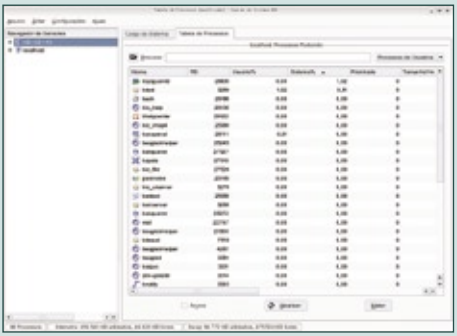
Tabela 5.1: Aplicativos Zeroconf

Acesso remoto

Como o próprio nome diz, o *Personal File Server* do KDE (KPF) é um mini servidor que disponibiliza arquivos para outros computadores. Por trás dele se esconde um simples servidor web, que disponibiliza seus serviços HTTP na rede via Zeroconf. O KPF é parte integrante do KDE.

Para o acesso remoto a uma área de trabalho, o KDE se utiliza do protocolo VNC. O *Remote Framebuffer* (servidor *KRFB*) informa o *KRDC* (*KDE Remote Desktop Clients*) sobre novas disponibilizações de serviços. Com isso, usuários e administradores encontram os recursos da área de trabalho disponibilizados que podem ser acessados através do KRDC. O KRFB e o KRDC são partes solidamente integradas ao KDE.

Figura 24
O KSysGuard supervisiona os recursos locais e remotos, monitorando a CPU e a alocação de memória.



Quem quiser saber quanto da CPU e da memória estão sendo ocupadas, pode acessar o *KSysGuard*. Já o *ksysguardd* (KDE Systemguard Daemon) possibilita o monitoramento de máquinas remotas a partir da estação local (**figura 24**).

Como o daemon *ksysguardd* disponibiliza seus recursos na rede via Zeroconf, o administrador encontra as listas dos clientes ativos e supervisiona as máquinas confortavelmente de sua estação de trabalho. O KSysguard e o KSysguardd são partes integrantes do KDE.

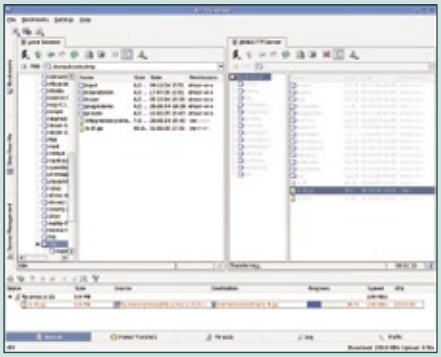


Figura 25
O KFTP-Grabber lista na barra lateral Sites Near Me todos os recursos do Zeroconf do tipo `_ftp._tcp`.

Nem todo programa domina o padrão Zeroconf. Por isso, há programas que servem para "fazer a ponte". Por exemplo, o *Avahi Bookmarks*. Esse mini navegador, baseado no Python FrameWork Twisted Web [3], publica listas em HTML com recursos de Internet e FTP, disponíveis na rede. O Avahi Bookmarks é parte integrante do Avahi.

Já o cliente de FTP *KFTP-Grabber* (**figura 25**) mostra os serviços FTP publicados através do Zeroconf em uma barra lateral.

Chat

Em especial, os clientes de mensagens instantâneas como o *Gaim* se fazem presentes na rede local usando o padrão Zeroconf. Desse modo, cada usuário, mesmo sem servidor local, sabe quais amigos locais estão online no momento. Isso é possível através do serviço `_presence.tcp`, que também utiliza o padrão *Apple iChat*.

Com alguns softphones VoIP, é possível disponibilizar seus contatos na rede via Zeroconf. Assim, é possível realizar chamadas mesmo que não haja um servidor central.

O servidor só é essencial para contatos de outras redes. O *Gizmo* (**figura 26**) e o sucessor do Gnomemeeting, o *Ekiga* (**figura 27**), implementam esse recurso. O projeto Gizmo ainda não incluiu o suporte Avahi em seu sistema básico.

Alternativas mais conhecidas como o *Banshee* (programado em *Mono*; veja a **figura 28**) e o *Rhythmbox* (incluído no Gnome) são bons exemplos de compartilhamento de músicas por meio do serviço `_daap._tcp`.

O usuário seleciona graficamente quais músicas devem ser apresentadas e as canções compartilhadas que deseja ouvir.

Figura 26
O softphone Gizmo disponibiliza seu número de contato na rede via Zeroconf. O Avahi Discovery (ao fundo) se ocupa disso.

Até mesmo o Konqueror, apesar de não trabalhar propriamente como um tocador de músicas, consegue fazer alguma coisa com serviços do tipo DAAP. Ele lista todas as músicas oferecidas hierarquicamente quando o usuário digita a "pseudo-URL" `daap:/` na barra de endereços. Como o KDE implemente isso na forma de uma KIO slave, o recurso também funciona em outros programas do KDE.

Lobo solitário

Um pouco fora do padrão é o VLC (*Video Lan Client*, **figura 29**). Ele não exporta streams via DAAP, mas utiliza um serviço próprio, o `_vlc-http._tcp`. Somente outro VLC pode tocar esses streams.

O Zeroconf estimula abertamente a cooperação: não são apenas os programas de uma mesma plataforma que trabalham juntos. O Avahi e seu concorrente da Apple (batizado de *Rendezvous* e, mais tarde, *Bonjour*) trabalham em harmonia perfeita. Assim os

Figura 27
O *Ekiga* (antigo *Gnomemeeting*) encontra contatos na rede local que usam os protocolos *VoIP H.323* e *SIP*.

recursos disponibilizados pelo Avahi, por exemplo, são listados pelo navegador *Safari* da Apple como se fossem *Bonjour Bookmarks* (veja a **figura 30**). Até mesmo o localizador de arquivos do *Mac OS X*, sistema operacional da Apple, é servido via Avahi.

Boas perspectivas

Quem acha o Zeroconf exótico ou que ele é um recurso exclusivo da Apple deveria conhecer a opção de código aberto Avahi. Apesar dessas implementações acima citadas, esse jovem projeto ainda não se consolidou entre as distribuições Linux – apesar de as implementações para Ubuntu e Debian já sejam mais do que funcionais.

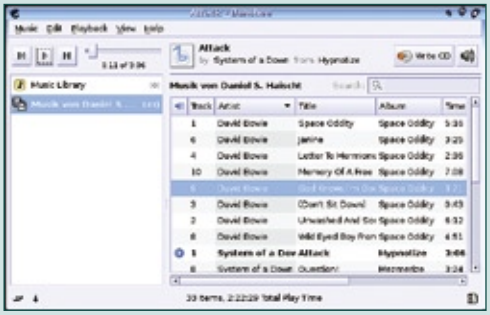


Figura 28
O *Banshee* acessa bibliotecas de músicas que foram disponibilizadas por um outro usuário.

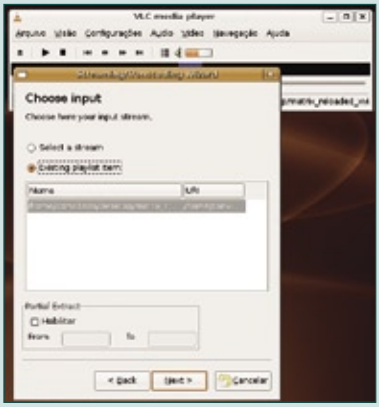


Figura 29
O VLC pode criar *streams* de áudio e vídeo e disponibilizá-los na rede com tecnologia do tipo *Zeroconf*.

Mas devido à qualidade da técnica Zeroconf e o amplo suporte em muitas aplicações, isso deverá mudar em breve.

Informações

[1] Projeto Avahi: http://www.avahi.org
[2] Gnome VFS: http://developer.gnome.org/doc/API/gnome-vfs/
[3] Twisted: <url>http://twistedmatrix.com/trac/

Figura 30
Recursos da Internet, disponibilizados pelo *Avahi*, são mostrados pelo *Safari* da *Apple* como se fossem bookmarks do *Bonjour*.

Administração de Redes

As redes de computadores são uma realidade há muito tempo. Você já imaginou um ambiente tecnológico em que todas as estações de trabalho tivessem que possuir uma impressora? Ou a troca de documentos – abstraindo das confusões de versões e formatos – tivesse que ser feita via CDs ou os antigos disquetes?

Construir uma rede, contudo, demanda planejamento e manutenção. Existem normas específicas para a montagem de uma rede de qualquer porte, incluindo o posicionamento do cabeamento – a chamada topologia da rede –, manufatura dos cabos de rede e escolha do sistema de distribuição de endereços de rede IP. Boa parte dessas normas incorporou-se ao cotidiano de um especialista ou administrador de redes de computadores (o famoso sysadmin) e segui-las passou a ser um dos escopos do bom profissional.

Neste volume da série de guias técnicos Linux Pocket Pro, abordamos a montagem e configuração de uma rede de computadores usando clientes e servidores Linux. Independendo da distribuição utilizada, e mesmo sem a necessidade de modo gráfico instalado no servidor, é possível criar uma infra-estrutura completa de TCP/IP, serviços DHCP e servidores de automatização de configuração de redes Zeroconf – tudo com o poder dos sistemas livres e a vantagem de se usar soluções altamente seguras e com baixíssimo custo de instalação e implementação.

R$ 14,90
€ 7,50
ISBN 978-85-61024-05-5
9 788561 024055

Leia também:

Linux Pocket Pro Gerenciamento e Desenho de Projetos

Sobre a série

A coleção Linux Pocket Pro é um lançamento da Linux New Media do Brasil, responsável pela publicação da conceituada revista Linux Magazine, especializada em Código Aberto e no universo do profissional de TI. O objetivo da coleção é trazer conhecimento confiável e de alto nível técnico para estudantes, técnicos e até mesmo administradores de sistemas experientes, sempre com enfoque prático e voltado para a utilização do sistema Linux e de outras tecnologias livres, hoje utilizadas ou reconhecidas como altamente competitivas por milhares de empresas, incluindo gigantes como IBM, Apple, Banco do Brasil, Casas Bahia e Microsoft.