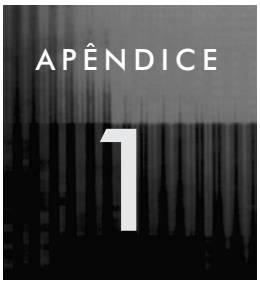


1 Entendendo Ethernet

1.1 Introdução

A imensa maioria das redes locais no mundo utiliza a tecnologia Ethernet. Iniciamos a breve apresentação dessa tecnologia perguntando: qual é o problema que a Ethernet tenta resolver? Em 1973, quando Bob Metcalfe da Xerox inventou a Ethernet, sua motivação era conectar estações de trabalho avançadas entre si e com impressoras laser em alta velocidade. Nessa época, conexões entre computadores eram possíveis, mas utilizando apenas baixas taxas de transmissão. A revolução proporcionada pela invenção de Metcalfe foi permitir a comunicação em alta velocidade, a um custo relativamente baixo. Por "alta velocidade", entende-se uma taxa de transmissão maior que 1 Mbps, considerada alta na época. Hoje, é corriqueiro utilizar Ethernet a velocidades de 100 Mbps e 1 Gbps (1000 Mbps) e a versão com velocidade de 10 Gbps está no forno.

Por outro lado, o atributo de alta velocidade da tecnologia impôs uma restrição de alcance: só seria possível conectar equipamentos que estivessem distantes um do outro de, no máximo, algumas centenas de metros. Nascia a *Rede Local*.

1.1.1 A evolução da Ethernet

Embora os princípios de operação sejam os mesmos, existem muitas variações da tecnologia Ethernet. Três necessidades levaram à existência dessas variações: *maior velocidade*, *melhor facilidade de uso* e *padronização*. Analisaremos mais detalhadamente as variações da tecnologia Ethernet numa seção posterior; por enquanto, expliquemos as forças que motivaram o aparecimento das variações.

Três Mbps (a velocidade original da Ethernet de Metcalfe) era considerada uma velocidade alta em 1973, mas o crescimento do uso de redes fez com que velocida-

des cada vez maiores fossem necessárias para carregar o tráfego de aplicações modernas. Hoje em dia, usuários navegam na Internet, participam de videoconferências, recebem fluxos de áudio etc., necessitando de maior velocidade nas redes locais. Um outro exemplo demonstra claramente a necessidade de redes locais de maior velocidade: a tecnologia de discos mudou muito de 1973 para cá. Há trinta anos, o acesso a um disco remoto teria como gargalo a velocidade do próprio disco. Hoje, os discos são muito mais rápidos e o acesso a um disco remoto satura uma rede de 10 Mbps, velocidade ainda largamente utilizada em redes locais. Ao longo das últimas três décadas, a tecnologia Ethernet tem sido oferecida em velocidades de 1Mbps, 3 Mbps, 10 Mbps, 100 Mbps e 1 Gbps (1000 Mbps). Parcialmente em conseqüência do aumento de velocidade, várias mídias físicas têm sido utilizadas para carregar tráfego Ethernet: cabos coaxiais grossos, cabos coaxiais finos, vários tipos de pares trançados e fibra óticas.

A segunda força que impeliu mudanças na tecnologia Ethernet diz respeito à *facilidade de uso e custo*. O Ethernet original usava um cabo coaxial grosso com duas desvantagens: primeiro, era de difícil instalação; segundo, o cabo coaxial corria de equipamento em equipamento, fazendo com que qualquer interrupção no cabo resultasse na parada total da rede inteira. Para mitigar esses efeitos, outras mídias foram utilizadas. Primeiro o cabo coaxial fino, facilmente dobrável, resolveu a questão de instalação. Segundo, para resolver a questão de interrupção na mídia, criou-se um equipamento chamado repetidor, ao qual todos os equipamentos se conectariam através de um par trançado. Dessa forma, a interrupção de sinal em um par trançado só afetaria a comunicação com um equipamento e não na rede inteira. Para possibilitar velocidades mais altas, várias "categorias" de pares trançados podem ser utilizadas. Finalmente, a mídia de fibra ótica foi introduzida para permitir maiores distâncias, maiores velocidades e devido à sua imunidade a ruídos.

A terceira força motriz da mudança na tecnologia Ethernet é a *padronização*. A Xerox patenteou Ethernet em 1978. Porém, para promover seu uso mais massificado, a Xerox se aliou à DEC e à Intel em 1980 para formar o padrão Ethernet de 10 Mbps chamado "DIX". Num esforço paralelo, o IEEE estava, na mesma época, padronizando tecnologias de redes locais e metropolitanas sob o número 802. A tecnologia Ethernet foi padronizada em 1985 pelo IEEE sob padrão número 802.3. Infelizmente, o padrão DIX e o padrão IEEE 802.3 são ligeiramente diferentes. Com o tempo, outros padrões foram criados para acomodar velocidades crescentes e mídias variadas. Numa seção posterior, discutiremos a questão das variações do Ethernet mais detalhadamente.

1.2 Princípios de operação

Esta seção descreve alguns princípios de operação da tecnologia Ethernet. Tais princípios serão de valia para o administrador de rede na sua tarefa diária de gerenciar redes locais.

1.2.1 Endereçamento

Num nível alto de abstração, pode-se dizer que Ethernet oferece a comunicação entre equipamentos de uma mesma rede física sem uso de conexões e com serviços unicast (um quadro vai para um destino único), multicast (um quadro vai para múltiplos destinos) e difusão (um quadro vai para todos os destinos).

Para concretizar esses serviços, cada entidade participante na comunicação possui um endereço único chamado "endereço MAC" ou "endereço físico". O significado da abreviação "MAC" será explicado adiante. Os endereços MAC possuem 48 bits e são únicos por construção; isso significa que, quando um fabricante constrói uma placa de rede Ethernet, ela recebe um endereço único determinado por hardware. O conjunto de endereços MAC forma um espaço plano (*flat address space*). Em outras palavras, não se utilizam subcampos do endereço para determinar a localização geográfica ou para ajudar no encaminhamento da informação. A placa de rede de endereço 00-07-95-03-FA-89 (em hexadecinal) poderia estar no Brasil, enquanto a placa com o próximo endereço 00-07-95-03-FA-8A poderia estar em outra rede local, na China.

Embora o assunto não tenha a ver com a tecnologia Ethernet em si, é útil lembrar que, no momento em que duas estações conectadas à rede querem conversar, a fonte conhece o endereço IP da estação destino mas não seu endereço MAC; o mapeamento de endereços IP para endereços MAC é feito com o protocolo ARP (*Address Resolution Protocol*).

A Ethernet permite que quadros sejam enviados para endereços especiais. O endereço FF-FF-FF-FF-FF-FF é o endereço de difusão: o quadro enviado para tal endereço é recebido por todas as estações. Além do mais, cada interface de rede (placa de rede) pode ser configurada para receber quadros pertencentes a um grupo multicast. Os endereços multicast iniciam com um bit igual a 1.

1.2.2 O quadro Ethernet

Embora haja uma pequena diferença na organização do quadro entre o padrão DIX e o padrão IEEE 802.3, o protocolo IP utiliza o quadro IEEE 802.3 de uma forma compatível com o padrão DIX.[1] O quadro do protocolo Ethernet é mostrado na Figura A1-1.

[1] No sentido de prover um protocolo de enlace comum para várias tecnologias de rede local, o padrão IEEE 802 separou a camada de enlace em duas subcamadas: Media Access Control (MAC) e Logical Link Control (LLC). O protocolo LLC é independente de tecnologia e há vários protocolos MAC, um para cada tecnologia englobada pelo padrão. Uma delas é Ethernet (IEEE 802.3). Essa separação não ocorre no padrão DIX. O protocolo IP utiliza o Ethernet IEEE 802.3 sem o uso de LLC, o que o torna compatível com DIX.

64 bits	48 bits	48 bits	16 bits	46 a 1500 bytes	32 bits
Preâmbulo	Endereço destino	Endereço de origem	Tipo	Dados	Sequência de verificação de quadro

FIGURA A1-1: O quadro Ethernet.

O conhecimento da organização do quadro não é importante para o administrador de rede. Nosso objetivo é de destacar os seguintes pontos:

- O quadro contém os endereços físicos (MAC) das estações de origem e de destino do quadro. Portanto, Ethernet estabelece comunicação quadro-a-quadro, sem necessidade de estabelecer conexões prévias entre as estações.
- Ao transportar pacotes IPv4, o campo "Tipo" receberá o valor hexadecimal 0x0800; para IPv6, o tipo é 0x86DD; para ARP, é 0x0806. Outros valores possíveis podem ser verificados em *http://www.iana.org/assignments/ethernet-numbers*.
- O tamanho mínimo do quadro (sem incluir o preâmbulo) é de 64 bytes e o tamanho máximo é de 1518 bytes.
- O quadro possui um campo de verificação (chamado *Frame Check Sequence* - FCS ou *Cyclic Redundancy Check* - CRC), que permite que a estação destino detecte erros na transmissão. Ethernet utiliza um protocolo (MAC) do tipo "melhor esforço", o que significa que, embora detecte erros, sua recuperação é deixada para protocolos de níveis superiores (TCP, por exemplo).

1.2.3 O Protocolo MAC

Conceitualmente, uma rede Ethernet simples consiste de um barramento único que todas as estações querem acessar para realizar suas transmissões.[2] Como esse meio é único e compartilhado, apenas uma estação pode transmitir (a comunicação é *half-duplex*). Deve, portanto, haver uma forma de organizar os acessos tal que cada estação possa, eventualmente, transmitir um quadro. O protocolo que realiza esse controle chama-se *Media Access Control* (MAC).

Ethernet usa um mecanismo bastante simples para realizar o acesso; esse mecanismo recebeu o nome de *Carrier-Sense Multiple Access with Collision Detection* (CSMA-CD) ou Acesso Múltiplo Usando Detecção de Portadora e Detecção de Colisão e funciona da seguinte maneira: quando uma estação[3] quer transmitir informação no meio compartilhado, ela espera até verificar que a portadora do meio está

[2] Discutiremos redes Ethernet mais complexas mais adiante neste capítulo.
[3] Na realidade, é a placa de rede, ou *Network Interface Card* – NIC, da estação que implementa o protocolo MAC.

ausente; isso indica que ninguém está transmitindo neste momento. Ela então inicia sua transmissão. Como outra estação pode ter tomado a mesma decisão, é possível que haja uma *colisão*, em que as transmissões interferem uma com a outra, embaralhando a informação no meio. Cada estação detecta a colisão e pára de transmitir. As estações esperam, então, um certo tempo aleatório antes de tentar novamente. Eventualmente, cada estação poderá transmitir sem interferência das demais.

É importante observar que colisões são eventos absolutamente normais numa rede Ethernet, embora um *excesso* de colisões possa diminuir sensivelmente o desempenho da rede.

1.2.4 Ethernet full-duplex

O protocolo CSMA-CD descrito na seção anterior permite acesso múltiplo ao meio, resultando em comunicação *half-duplex*: não há transmissões simultâneas no meio. Sob certas circunstâncias, é possível operar em modo *full-duplex*, com duas estações transmitindo simultaneamente. Isso é possível sempre que a configuração da rede permitir que *no máximo duas* fontes possam transmitir no meio ao mesmo tempo. A Figura A1-2 mostra situações como isso pode ser assegurado.

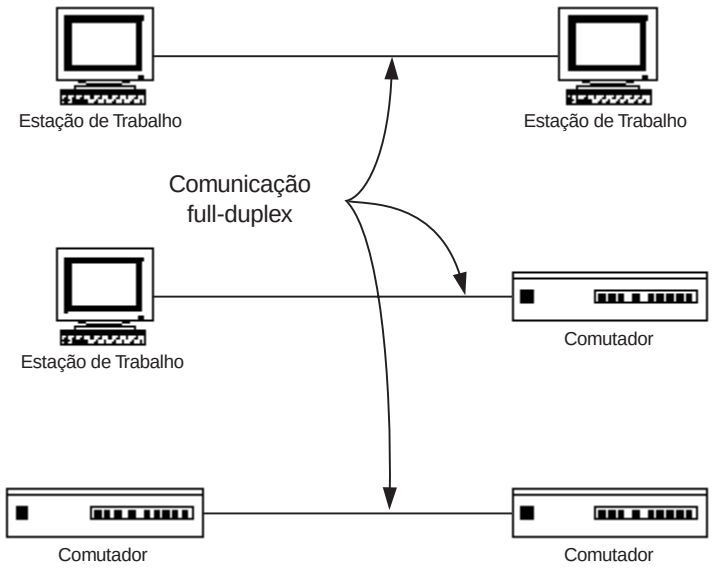


FIGURA A1-2: Comunicação full-duplex.

Nos três casos exibidos na figura, o enlace estabelece uma comunicação ponto-a-ponto entre equipamentos. Observe que nenhum repetidor pode estar envolvido, já que este permitiria que mais de uma estação transmitisse no meio compartilhado.

A comunicação *full-duplex* dobra a capacidade do enlace. Por exemplo, um enlace que, no modo *half-duplex*, possui capacidade total de 100 Mbps tem capacidade de 100 Mbps em cada sentido se operando no modo *full-duplex*.

Raciocinando um pouco, concluímos imediatamente que, já que não há acesso múltiplo a um enlace *full-duplex*, não há necessidade de usar o mecanismo CSMA-CD. De fato, o modo *full-duplex* de operação desabilita o mecanismo CSMA-CD do Ethernet. Os equipamentos envolvidos podem transmitir quando querem, sem detectar a portadora, nem verificar colisões. Na realidade, colisões nunca ocorrem em modo *full-duplex*.

Terminamos a discussão do modo *full*-duplex com um alerta: não basta ter comunicação ponto-a-ponto para que o modo *full-duplex* seja habilitado. Ambos os lados devem ser configurados para esse modo de operação, seja através de um procedimento manual ou através de autonegociação. Não se pode misturar os modos *half-duplex* e *full-duplex*, um em cada lado do enlace. Isso resultaria em erros de vários tipos, incluindo mais colisões, colisões tardias[4] e erros de CRC.

1.2.5 Autonegociação

O aparecimento da tecnologia Fast Ethernet de 100 Mbps em 1995 aumentou não só a velocidade dos enlaces Ethernet, mas também as dores de cabeça provenientes de misturas das tecnologias de 10 Mbps e 100 Mbps, usando mídias compatíveis. É possível, por exemplo, usar o mesmo par trançado para o tráfego 10 Mbps ou 100 Mbps. O mesmo ocorre com fibras óticas. Um outro complicador é a existência de placas de rede 10/100 e 10/100/1000 e equipamentos de interconexão que podem funcionar tanto a 10 Mbps quanto a 100 Mbps ou 1000 Mbps. Portanto, ao conectar um equipamento a um repetidor ou comutador Ethernet, é necessário casar a velocidade de operação e o modo de operação (*half-duplex* ou *full-duplex*). Esse casamento pode ser feito manualmente ou automaticamente através do *Mecanismo de AutoNegociação*, introduzido em 1995.

Quando ambos os lados de um enlace possuem suporte à autonegociação, eles escolhem a combinação de parâmetros que dará melhor desempenho. Isto é, a maior velocidade possível é escolhida (10 Mbps, 100 Mbps ou 1000Mbps) e o modo *full-duplex* é escolhido, caso seja suportado por ambos os lados.

A autonegociação se torna importante quando lembramos que uma instalação de rede freqüentemente tem muitos equipamentos antigos e novos, e que equipamentos podem ser desconectados de uma porta de um equipamento e conectados a outra porta com uma certa freqüência. Se essas tarefas envolvessem sempre uma reconfiguração manual de velocidade e modo de operação, a vida do administrador de rede ficaria muito pior!

---

[4] Uma "colisão tardia" (*late collision*) ocorre depois do maior tempo possível em que poderia ocorrer de acordo com as especificações do padrão Ethernet.

Devido à existência de hardware antigo, ocorrem casos em que um lado, digamos o lado A, oferece suporte à autonegociação enquanto o outro lado, digamos o lado B, não tem tal suporte. Nesse caso, A perceberá que B não está fazendo autonegociação e passará a fazer *detecção paralela*. Neste mecanismo, A descobre a velocidade de B e obrigatoriamente escolhe o modo de operação *half-duplex*. Dois problemas associados à detecção paralela ocorrem na prática:

1. O lado B não oferece suporte à autonegociação, mas foi manualmente configurado em modo *full-duplex*; nesse caso, o lado A escolherá *half-duplex* e a comunicação não ocorrerá de forma satisfatória;
2. O lado A implementa a autonegociação, mas não implementa o padrão corretamente e escolhe o modo de operação *full-duplex*; a solução aqui é de atualizar a versão do driver da placa de rede ou do software do equipamento de interconexão.

Observe que a autonegociação só existe para mídias de par trançado e para Gigabit Ethernet com fibra ótica. Não há opção de autonegociação para Ethernet em fibra ótica em velocidades de 10 Mbps e 100 Mbps. O motivo é que tais equipamentos utilizam feixes de luz de comprimento de onda diferentes, não sendo possível realizar a autonegociação.

1.3 Padrão IEEE de tecnologias Ethernet

Vários sistemas de mídia têm sido padronizados pelo IEEE ao longo dos anos. Nem todos são usados na prática, mas pelo menos seis padrões diferentes são importantes numa rede típica. Os padrões variam de acordo com:

- A velocidade da transmissão;
- O tipo de codificação do sinal utilizado;
- O tipo de mídia utilizado;
- O tipo de conector utilizado.

As velocidades em uso hoje incluem:

- 10 Mbps (Ethernet original);
- 100 Mbps (Fast Ethernet);
- 1000 Mbps ou 1 Gbps (Gigabit Ethernet).

As codificações do sinal incluem:

- Manchester;
- 4B/5B;
- 8B6T;

- 5B/6B;
- 8B/10B;
- 4D-PAM5.

As mídias incluem:

- Cabos coaxiais (raramente utilizados hoje, embora fossem comuns no início da vida da tecnologia Ethernet);
- Pares trançados de várias categorias, as mais comuns sendo Cat 3 (*Voice Grade* – ou "feitos para voz"), Cat 5 e Cat5e. É o tipo de mídia mais comumente utilizado para chegar ao *desktop*;
- Fibras óticas monomodo ou multímodo, mais comumente utilizadas na espinha dorsal (*backbone*) da rede.

Os conectores mais utilizados são:

- RJ-45, para par trançado;
- ST e SC, para fibras óticas.

Podemos agora descrever brevemente os tipos mais comuns de tecnologias Ethernet padronizadas pelo IEEE.

1.3.1 10BASE-5 e 10BASE-2

Essas são as 2 tecnologias originais Ethernet, utilizando cabo coaxial. Funcionam a 10 Mbps e são consideradas tecnologias obsoletas. Não têm suporte ao modo *full-duplex*. Todas as outras tecnologias descritas a seguir permitem operação em modo *full-duplex*.

1.3.2 10BASE-T

Essa é a tecnologia que popularizou o Ethernet. Utiliza velocidade de 10 Mbps e 2 pares de fios trançados de categoria 3, embora cabos de categoria 5 sejam mais largamente utilizados hoje em dia. Os cabos têm comprimento máximo de 100 metros.

1.3.3 10BASE-FL

Operando a 10 Mbps, o enlace é de fibra ótica multi-modo. É uma extensão de um padrão mais antigo chamado *Fiber Optic Inter-Repeater Link* (FOIRL). A fibra pode ter até 2000 metros de comprimento.

1.3.4 100BASE-TX

Fast Ethernet mais comumente empregado, com velocidade de 100 Mbps usando 2 pares de fios trançados de alta qualidade (categoria 5 ou melhor). O cabo está limitado a 100 metros, sem uso de repetidor.

1.3.5 100BASE-FX

Fast Ethernet utilizando fibras óticas multi-modo. A fibra pode ter até 2000 metros de comprimento.

1.3.6 1000BASE-T

Gigabit Ethernet, funciona a 1000 Mbps (1 Gbps). Utiliza 4 pares de fios trançados de categoria 5 ou melhor, com comprimento máximo de 100 metros.

1.3.7 1000BASE-X

Gigabit Ethernet utilizando fibra ótica. Largamente utilizando em *backbones* de redes de campus. A fibra pode ter até 220 metros de comprimento, se for multi-modo e até 5000 metros, se for mono-modo.

1.3.8 10 Gb Ethernet (10GEA)

Em maio de 2002 foi realizada a maior demonstração de interoperabilidade de uma rede 10 Gigabit Ethernet. Equipamentos de vários fabricantes participaram da demonstração. A rede 10Gb Ethernet de 200 Km incluiu 4 dos 7 tipos especificados em *drafts* pela força tarefa IEEE 802.3ae: 10GBASE-LR, 10GBASE-ER, 10GBASE-SR e 10GBASE-LW. Leia mais sobre 10Gb Ethernet em [10GEA].

2 Bibliografia

[GUIA-ETHERNET]	Spurgeon, C. E. Ethernet – *O Guia Definitivo*. Campus, 2000.
[10GEA]	10 Gigabit Ethernet Alliance. Em: `http://www.10gea.org/`

APÊNDICE

2

2 Algumas palavras sobre ferramentas de gerência

2.1 Introdução e motivação

As ferramentas de gerência são o nosso braço direito (às vezes, o esquerdo também) no dia-a-dia de nossas atividades de gerência. São elas que nos ajudam a detectar problemas quando eles ocorrem, ou antes mesmo de ocorrerem (gerência de rede proativa). Gerenciar uma rede sem o auxílio de instrumentação adequada é uma tarefa bastante árdua e que muito provavelmente não oferecerá uma boa qualidade de gerência. Gerenciar uma rede sem ferramenta alguma, ou com ferramentas inadequadas – que, por exemplo, não nos dêem uma boa visão dos principais elementos da rede – é o mesmo que ir para a guerra cego e sem armas.

Tomemos novamente a nossa analogia entre a Gerência de Redes e a Medicina. Na Medicina, podemos dar vários exemplos de que quando o médico não está bem instrumentado fica bem mais difícil dar o diagnóstico ou prever doenças futuras. Imagine, por exemplo, um paciente que chega no médico com febre alta, sintoma característico de infecções. Sem um hemograma (exame de sangue) o médico não poderia descobrir que tipo de infecção está se manifestando no paciente: se virótica ou bacteriana. Sem saber qual o diagnóstico, o médico não poderia tratá-lo. Um outro exemplo: uma mulher vai ao ginecologista para realizar exames de rotina. Ela está com câncer de colo do útero em fase inicial e nenhum sintoma se manifestou ainda. Se o médico não possuir um colposcópio, a detecção desta doença não poderá ser realizada. Mais tarde, apenas quando os sintomas se manifestarem e o câncer já estiver em um estado mais avançado será possível sua detecção.

Com a gerência temos uma situação bastante semelhante. Quando não estamos bem instrumentados, não somos capazes de descobrir problemas e por conseqüência, não seremos capazes de solucioná-los. Isso nos afastará substancialmente de nosso objetivo, que é manter o bom funcionamento da rede.

Existem ferramentas de gerência para todos os gostos e finalidades. Com ferramentas mais simples de gerência, que vêm no próprio sistema operacional de rede, podemos realizar uma gerência ad hoc. Esse tipo de gerência tem seus problemas, como veremos na Seção 2.2.

Plataformas de gerência oferecem aplicações de monitoração e controle da rede mais sofisticadas, possibilitando, portanto, a gerência de grandes redes mais facilmente. Na Seção 2.3 falaremos um pouco mais de plataformas de gerência.

Com o advento da Internet e a proliferação de serviços Web, aplicações de gerência de redes baseadas em Web (que acessamos através de um navegador) estão sendo cada vez mais bem aceitas. Falaremos sobre ferramentas de gerência baseadas em Web na Seção 2.4.

Além de todas essas ferramentas, existem os analisadores de protocolos, dos quais falaremos na Seção 2.5, que nos permitem bater um raio-X do tráfego que percorre uma rede.

2.2 Ferramentas mais simples

Chamamos aqui de ferramentas mais simples as ferramentas que não nos dão uma visão geral da rede, mas que muitas vezes nos ajudam a descobrir características mais internas de determinados elementos da rede. Essas ferramentas são geralmente oferecidas junto com o sistema operacional de rede dos próprios hospedeiros.

Exemplos de ferramentas mais simples são `traceroute` (tracert para máquinas Windows), `ping`, `route`, `netstat`, `ifconfig` (essas duas últimas são ferramentas Unix-like) e `ipconfig` (Windows).

Em muitos momentos, necessitaremos dessas ferramentas. Mas elas sozinhas não são suficientes para realizar bem a tarefa de gerência. Em geral, utilizamos essas ferramentas como ferramentas de apoio depois de termos descoberto que um problema existe.

Com `traceroute`, podemos descobrir onde o problema está localizado. Mas certamente seria muito mais rápido descobrir o problema com o auxílio de uma estação de gerência onde o mapa da rede é apresentado e alarmes são gerados automaticamente quando limiares ou mudanças de estado operacional são detectados.

Quando possuímos apenas essa instrumentação mais simples, dizemos que utilizamos a técnica da porta aberta. Esse é um tipo de gerência totalmente reativo (em que se reage a problemas, que não podem ser previstos). Esse tipo de gerência não tem escala. Não é possível gerenciar uma rede com milhares de elementos dessa forma.

2.3 Plataformas de gerência

Uma estação de gerência é normalmente construída usando uma plataforma de gerência. Para entender o que é uma plataforma de gerência, temos de entender que o mundo gerência é complexo e o software que executa numa estação de gerência não

é uma aplicação única e monolítica. A solução de gerência é montada modularmente usando várias aplicações, à semelhança do que ocorre num hospedeiro em que várias aplicações são instaladas para formar o conjunto de software disponível. Portanto, podemos comparar a plataforma de gerência ao "sistema operacional de gerência". É o software básico de gerência que oferece serviços que várias aplicações individuais de gerência gostariam de receber (tais como realizar sondagens). A plataforma de gerência permite que aplicações individuais de gerência possam se "plugar" nela para formar uma solução de gerência completa. Pode-se dizer, portanto, que uma plataforma de gerência é uma solução incompleta de gerência (ou um framework de gerência) que as aplicações de gerência completam.

Sobre as plataformas são construídas diversas aplicações de gerência, usadas pelos operadores da rede. Costuma-se dizer que as aplicações plugam-se na plataforma. Vemos na Figura A2-1, uma plataforma, com aplicações que acessam as informações dos elementos gerenciados coletadas e armazenadas pela plataforma.

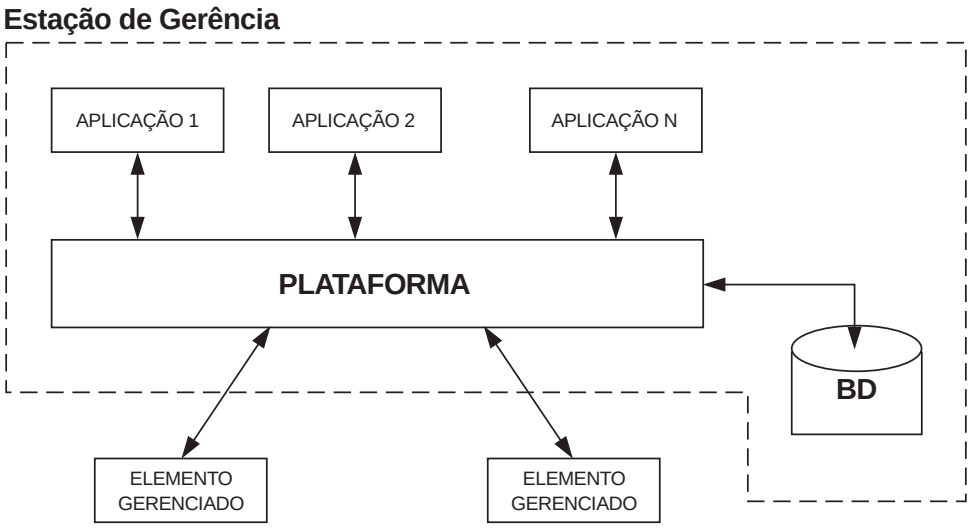


FIGURA A2-1: Arquitetura geral de um Sistema de Gerência de Redes.

Exemplos de plataformas populares de gerência incluem: OpenView da Hewlett Packard, Tivoli da IBM, Spectrum da Aprisma e CA-Unicenter da Computer Associates. Exemplos de aplicações de gerência incluem Netclarity da Lanquest (gerência de desempenho), Alarm Manager da Aprisma (gerência de falhas), AssetView da Hewlett Packard (gerência de bens), CiscoWorks da Cisco (gerência de configuração) etc.

2.3.1 OpenNMS

OpenNMS é um projeto de código aberto dedicado à criação de uma plataforma de gerência de rede. Cada vez mais as empresas dependem de serviços oferecidos atra-

vés da rede, como por exemplo e-mail e Web. A maioria das soluções de gerência patenteadas é muito cara e complexa, além de muitas vezes não se adaptar agilmente às mudanças de tecnologia e de requisitos de gerência dos usuários. A idéia é aproveitar o poder de software de código aberto para desenvolver produtos de gerência de redes poderosos, estáveis e escaláveis. Além de oferecer a gerência tradicional usando o protocolo de gerência SNMP, OpenNMS tem a habilidade de monitorar os serviços oferecidos pela rede, podendo ser usado para gerar relatórios de nível de serviço e notificações de problemas. Mais informações sobre o OpenNMS podem ser encontradas em [OpenNMS].

2.4 Aplicações de gerência baseadas em Web

As aplicações de gerência que permitem aos administradores da rede monitorá-la e controlá-la a partir de um navegador são bastante atrativas. Dentre as vantagems das ferramentas de gerência baseadas em Web encontram-se [3Com-WBM]:

- Possibilidade de monitorar e controlar os elementos da rede usando qualquer navegador em qualquer nó da rede. Antes, quando falávamos de aplicações de gerência *standalone*, os gerentes de rede só podiam usar a ferramenta em máquinas onde ela estivesse instalada e devidamente configurada;
- A interface gráfica da Web já é bem conhecida e as operações realizadas em um navegador também, não sendo necessários gastos com treinamento de pessoal. Além disso a interface gráfica é muito rica, sendo bastante superior a interfaces de linha de comando obtidas através de telnet, por exemplo;
- Usar a Web para distribuir as informações sobre a operação da rede tem se mostrado uma tarefa eficaz. Por exemplo, em um determinado endereço da Intranet poderíamos disponibilizar informações para os usuários sobre o estado da rede e atualizações que eles precisam realizar. Isso evitaria, dentre outras perturbações, ligações em demasia ao *help desk*.

Além disso, não precisamos usar ferramentas diferentes para gerenciar os diversos elementos da infra-estrutura de TI, inclusive os serviços.

Alguns padrões para gerência baseada em Web estão surgindo. Dentre eles encontra-se o WBEM (Web-Based Enterprise Management). O objetivo principal do WBEM é alcançar a gerência unificada de todos os sistemas e redes de uma organização. Para maiores informações sobre WBEM, visite [WBEM_Initiative] e [WBEM_Services]

Costumamos usar, dentre outras ferramentas de gerência, uma ferramenta bastante simples, mas que nos oferece as informações que precisamos em tempo real: o WebManager [Sauvé]. Ela usa o protocolo SNMP para obter informações de gerência dos agentes. Algumas de suas características são:

- Nos mostra o mapa da rede que gerenciamos e sobre cada um dos elementos gerenciados indica seu estado através de um esquema de cores convencional (cor verde para elementos que estão *up*, amarelo para indicar problemas de desempenho e vermelho para indicar falta de conectividade ao equipamento);
- Mostra gráficos do dia atual e dos últimos x dias, sendo x configurável. Sempre escolhemos x como 7, assim podemos ver os dados de toda a semana e conhecer bem o baseline da rede. Considere, por exemplo, que em todos os dias úteis, o tráfego fica sempre em torno de 7 Mbps. Se em plena quarta-feira o tráfego cai para 1 Mbps já suspeitamos que algo errado está ocorrendo.

O WebManager não nos permite controlar os elementos da rede, apenas monitorá-los, mas ele oferece todas as informações de que precisamos para rapidamente localizar e muitas vezes diagnosticar problemas. Além disso, essa ferramenta não oferece ainda o serviço de descobrimento automático de topologia, tornando o trabalho de descrever os elementos da rede bastante enfadonho.

Para saber mais sobre gerência de redes baseada em Web leia [3Com-WBM].

2.5 Analisadores de protocolos

Já falamos bastante sobre analisadores de protocolos no Procedimento 9. Nesta seção, queremos apenas deixar claro que analisadores de protocolos são ferramentas auxiliares da gerência de redes.

2.6 Referências bibliográficas

[WBEM_INITIATIVE]	Web-Based Enterprise Management (WBEM) Initiative. Em: `www.dmtf.org/standards/standard_wbem.php.`
[WBEM_SERVICES]	WBEM Services. Em: `http://wbemservices.sourceforge.net/`
[3COM_WBM]	Web-Based Management: The New Paradigm for Network Management. Em: `http://www.3com.de/pdf/50062701.pdf`
[SAUVÉ]	Sauvé, J.P. WebManager: A Web-Based Network Management Application. Latin American Network Operations and Management Symposium, pp. 335-346, dezembro 1999.
[OPENNMS]	Página oficial do projeto OpenNMS. Em: `http://www.opennms.org/`

APÊNDICE

3

3 Introdução ao SNMP

3.1 Introdução e histórico

Até o início da década de 1980, redes de computadores eram baseadas em arquiteturas e protocolos patenteados, a exemplo de *System Network Architecture* (SNA) da IBM e DECNET da *Digital Equipment Corporation*. Já no final da década de 1980, redes interconectadas baseadas na arquitetura e protocolos TCP/IP estavam em franca ascensão. Porém, do ponto de vista da gerência de tais redes, a situação ainda favorecia arquiteturas proprietárias, devido à inexistência de soluções de gerência de redes TCP/IP.

O termo "gerência" engloba as seguintes atividades típicas:

- **Gerência de configuração**: inclui saber quais elementos fazem parte da rede, quais são suas relações uns com os outros, que parâmetros de configuração cada um deve assumir etc.
- **Gerência de falhas**: diz respeito à detecção de eventos anormais, o diagnóstico de problemas que levaram a esses eventos, acompanhamento e solução dos problemas.
- **Gerência de desempenho**: responsável pela monitoração de indicadores de desempenho, solução de problemas de desempenho, planejamento de capacidade etc.
- **Gerência de segurança**: responsável pela proteção dos recursos disponíveis na rede.
- **Gerência de contabilidade**: responsável pela contabilização e verificação de limites da utilização de recursos da rede, com a divisão de contas feita por usuários ou grupos de usuários.

Neste livro, nosso foco se mantém nos primeiro três aspectos de gerência.

O crescimento de redes TCP/IP ao longo da década de 1980 aumentou consideravelmente as dificuldades de gerência. A demora no aparecimento de soluções abertas baseadas no modelo OSI fez com que um grupo de engenheiros decidisse elaborar uma solução temporária baseada num novo protocolo: *Simple Network Management Protocol* (SNMP).[1]

A simplicidade do SNMP facilitou sua inclusão em equipamentos de interconexão. No final da década de 1990, a solução SNMP já era tão difundida que se estabelecera como padrão de gerência de redes de computadores. Hoje, praticamente todos os equipamentos de interconexão dão suporte a SNMP, bem como muitos outros dispositivos (nobreaks, modems etc.), e sistemas de software (servidores Web, sistemas de bancos de dados etc.).

Neste capítulo, discutimos a arquitetura do mundo SNMP e detalhamos o protocolo SNMP, além da parte mais importante para o gerente de rede: a informação de gerência, presente na forma de *Management Information Bases* (MIBs).

3.2 Arquitetura do mundo SNMP

Qualquer solução de gerência, seja ela baseada no SNMP ou não, exibe uma arquitetura básica comum. Esta arquitetura tem quatro componentes:

- Vários elementos gerenciados;
- Uma ou mais estações de gerência;
- Informação de gerência;
- Um protocolo de gerência.

Passamos a discutir cada componente no resto do presente capítulo.

3.2.1 Elementos gerenciados

Os elementos gerenciados constituem os componentes da rede que precisam operar adequadamente para que a rede ofereça os serviços para os quais foi projetada. Exemplos de elementos gerenciados incluem:

- Hardware: equipamentos de interconexão, enlaces de comunicação, hospedeiros, nobreaks, modems, impressoras etc.
- Software: sistemas operacionais, servidores de bancos de dados, servidores Web, servidores de mail etc.

[1] Formalmente, o mundo SNMP chama-se "Internet-Standard Network Management Framework". Seguindo uma tendência mundial, preferimos usar o termo mais simples "mundo SNMP".

Elementos gerenciados devem possuir um software especial para permitir que sejam gerenciados remotamente. Esse software chama-se "agente".

3.2.2 Estações de gerência

As estações de gerência são hospedeiros munidos de software necessário para gerenciar a rede. Para facilitar a vida dos especialistas em gerência, as estações de gerência são normalmente centralizadas; aliás, é muito freqüente que haja uma única estação de gerência. Só se recorre a várias estações de gerência quando a escala da rede impede que seja gerenciada por uma única estação. Usaremos a expressão "estação de gerência" no singular.

O software presente na estação de gerência que conversa diretamente com os agentes nos elementos gerenciados é chamado de "gerente".

A Estação de gerência pode obter informação de gerência presente nos elementos gerenciados através de uma sondagem[2] regular dos agentes ou até mesmo recebendo informação enviada diretamente pelos agentes; a estação também pode alterar o estado de elementos gerenciados remotos. Adicionalmente, a estação de gerência possui uma interface com o usuário especialmente projetada para facilitar a gerência da rede.

3.2.3 Informação de gerência

As conversas entre gerente e agentes envolvem informação de gerência. Essa informação define os dados que podem ser referenciados em conversas gerente-agentes. Exemplos incluem: informação de erro de transmissão e recepção em enlaces de comunicação, status de um enlace de comunicação, temperatura de um roteador, tensão de entrada de um equipamento nobreak etc.

3.2.4 Protocolo de gerência

O gerente e os agentes trocam informação de gerência usando um protocolo de gerência. O protocolo inclui operações de monitoramento (leitura de informação de gerência) e operações de controle (alteração de informação de gerência presente no elemento gerenciado). Um exemplo de operação de monitoramento ocorre quando o gerente pergunta a um roteador: "Qual é a quantidade de erros ocorrendo no fluxo de entrada na interface número 17?" Um exemplo de uma operação de controle ocorre quando o gerente diz ao roteador: "Desligue sua interface número 17."

---

[2] A palavra *sondagem* é freqüentemente empregada no inglês (*poll*) no mundo da gerência. Efetuar sondagens de elementos gerenciados chama-se *polling*, em inglês.

3.2.5 Plataformas de gerência

Um Sistema de Gerência de Redes é normalmente composto por uma plataforma de gerência e aplicações construídas sobre esta. Elas oferecem funções e serviços básicos de gerência que são comuns a muitas aplicações de gerência. Leia mais sobre plataformas de gerência no Apêndice 2.

3.2.6 O mundo SNMP

O mundo SNMP está organizado conforme a arquitetura explicada anteriormente. Porém, difere de soluções alternativas devido ao chamado "axioma fundamental": o impacto de adicionar gerência de rede aos elementos gerenciados deve ser mínimo, refletindo um menor denominador comum . O resultado é que a solução básica de gerência e o protocolo SNMP são muito simples. A complexidade está nas poucas estações de gerência e não nos milhares de elementos gerenciados. A simplicidade dos agentes SNMP foi fundamental na enorme difusão dessa solução ao longo dos anos 1990.

O mundo SNMP (ou, mais formalmente, "Internet-Standard Network Management Framework") está baseado em três documentos:

- **Structure of Management Information** (SMI). Definido pela RFC 1155, a SMI define essencialmente, a forma pela qual a informação gerenciada é definida.
- **Management Information Base** (MIB) principal. Definida na RFC 1156, a MIB principal do mundo SNMP (chamada MIB-2) define as variáveis de gerência que todo elemento gerenciado deve ter, independentemente de sua função particular. Outras MIBs foram posteriormente definidas para fins particulares, tais como MIB de interfaces Ethernet, MIB de nobreaks, MIB de repetidores etc.
- **Simple Network Management Protocol** (SNMP). Definido pela RFC 1157, é o protocolo usado entre gerente e agente para a gerência, principalmente trocando valores de variáveis de gerência.

3.2.6.1Sondagens e traps

No mundo SNMP, é normalmente o gerente que inicia a conversa com os agentes, numa operação chamada de sondagem (*polling*). Sondagens são consultas periódicas feitas pelo gerente aos agentes de todos os elementos gerenciados, pedindo que informem o valor de algumas variáveis de gerência mais críticas. É assim que a estação de gerência descobre a existência de eventos anormais nos elementos gerenciados. A comunicação iniciada pelos agentes também é possível, embora mais rara. O agente pode enviar um *trap* à estação de gerência para notificar algum evento anormal, sem esperar pela próxima operação de sondagem.

3.2.6.2 O modelo de leitura-escrita do SNMP

O protocolo SNMP funciona de acordo com um modelo operacional extremamente simples chamado "modelo de leitura escrita" ou "modelo *fetch-store*". A informação de gerência mantida pelos agentes consiste de variáveis com valores. O protocolo permite ler o valor de cada variável ou alterar seu valor. Há, portanto, duas operações básicas no protocolo SNMP: GET, para ler o valor de uma variável de gerência e SET, para alterar seu valor.[3] Embora o protocolo seja muito simples, ele é bastante poderoso em virtude do fato de que ações especiais são efeitos colaterais de operações de escrita.

Por exemplo, há uma variável de gerência chamada `ifAdminStatus` que determina o estado desejado para um enlace de comunicação. Se o valor "1" for escrito nessa variável num determinado agente e para uma determinada interface de comunicação, isso significa que desejamos que o enlace esteja "up" ou "em funcionamento". Ao escrever o valor "2" – correspondendo a "down" – nessa variável, o enlace é desabilitado.

3.3 Informação de gerência: objetos, instâncias e MIBs

As variáveis de gerência que podem ser manipuladas pelo protocolo SNMP formam uma base de dados virtual acessível ao agente de um elemento gerenciado. Devido ao grande número de variáveis de gerência (existem vários milhares de variáveis de gerência no mundo SNMP), o espaço de nomes dessas variáveis está estruturado hierarquicamente, como mostra a Figura A3-1.

Nessa árvore, alguns órgãos de padronização internacional (ISO, CCITT) têm seu espaço logo abaixo da raiz. Cada objeto[4] da árvore possui um rótulo com uma descrição textual e um número. Por exemplo, há um objeto chamado "mgmt" com número 2, abaixo de um objeto com rótulo "internet" cujo número é 1. A árvore mostrada na Figura A3-1 exibe apenas o topo da árvore de todas as variáveis de gerência do mundo SNMP.

Os objetos da árvore que não são folhas agrupam um conjunto de objetos relacionados. Os objetos descrevem a informação mantida nos agentes. Uma *instância* de um objeto (também chamada de *variável*) é o que realmente é manipulado pelo protocolo SNMP. Para entender a diferença entre objeto e instância de objeto, devemos observar que, no mundo SNMP, objetos podem ser simples (ou escalares) ou podem fazer parte de uma linha de uma tabela.

---

[3] Veremos todas as operações do protocolo numa seção posterior deste capítulo.

[4] A palavra "objeto" não está sendo empregada aqui no sentido de orientação a objeto, conhecido conceito da ciência da computação. Um objeto é um nodo da árvore identificável por um nome.

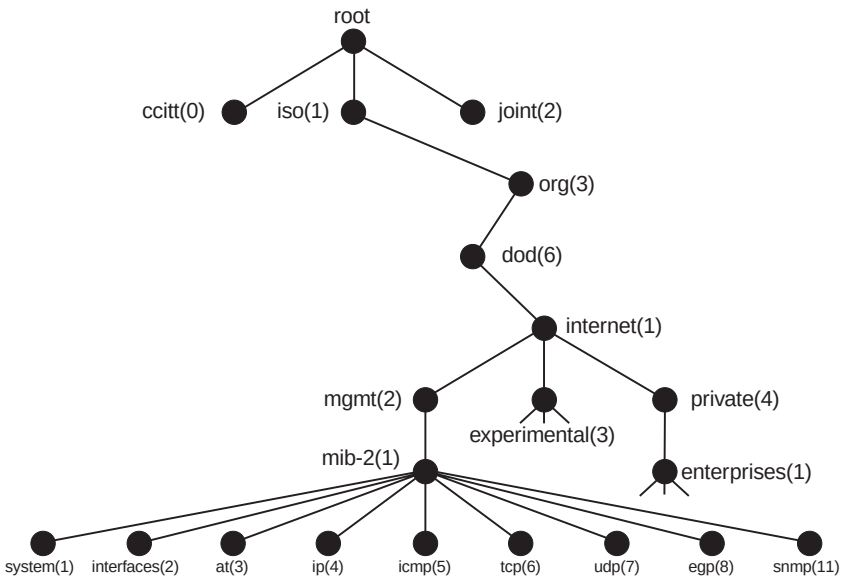


FIGURA A3-1: A estrutura hierárquica das variáveis de gerência

3.3.1 Objetos simples

Vejamos primeiro os objetos simples. Um objeto é identificado unicamente através do caminho percorrido da raiz da árvore até o objeto em questão. Por exemplo, o objeto com *Object Identifier* (ou OID) igual a

```
iso.org.dod.internet.mgmt.mib-2.system.sysDescr
```

identifica um objeto simples que possui uma única instância. A instância tem o nome:

```
iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0
```

Nesse caso, a instância (única) do objeto representa a descrição do elemento gerenciado. Um valor para esta variável poderia ser "IBM 8271 EtherStreamer Switch". O nome numérico 1.3.6.1.2.1.1.1.0 é equivalente ao nome simbólico dado anteriormente.

3.3.2 Objetos em tabelas

Em muitos casos, um determinado objeto possui várias instâncias. Considere, por exemplo, o objeto:

```
iso.org.dod.internet.mgmt.mib-2.interfaces.iftable.ifEntry.ifInOctets
```

Esse objeto informa o número de bytes que foram recebidos numa determinada interface de rede do elemento gerenciado em questão. Porém, como o elemento ge-

renciado possui várias interfaces de rede, deve existir uma instância desse objeto para cada interface. Há vários outros objetos semelhantes que dão informação sobre interfaces de rede. Podemos, portanto, pensar numa tabela, em que as colunas são os objetos (`ifInOctets`, além de outros) e as linhas são as instâncias desses objetos, uma linha para cada interface. No mundo SNMP, só existem tabelas bidimensionais contendo objetos simples.

Para identificar uma instância única de uma variável em tabela, devemos especificar a coluna descendo na árvore de objetos. Já demos um exemplo de um tal nome de objeto (`ifInOctets`). Agora, para especificar a instância particular desejada, devemos especificar a linha desejada. Linhas de tabelas são identificadas unicamente através de uma (ou mais) coluna da tabela que tenham conteúdo único. Chamamos essa coluna de "chave" da tabela.

Examinemos um exemplo. A tabela de interfaces de rede de um elemento gerenciado se chama:

```
iso.org.dod.internet.mgmt.mib-2.interfaces.iftable
```

Cada linha se chama ifentry e contém vários objetos, entre os quais `ifIndex`, `ifDescr`, `ifInOctets` e assim por diante. A chave da tabela é a coluna `ifIndex` que representa o índice da interface. Esse índice pode assumir qualquer valor desde que seja único. Vamos supor que uma interface particular tenha `ifIndex` igual a 17. Então, o valor da coluna `ifInOctets` para essa interface seria

```
iso.org.dod.internet.mgmt.mib-2.interfaces.iftable.ifentry.ifInOctets.17
```

e seu valor poderia ser, digamos 134631, o que significa que, desde um certo momento do passado, 134.631 bytes entraram nessa interface de rede.

Em outras tabelas, é possível que várias colunas sirvam de chave.

3.3.3 Management Information Bases – MIBs

O conjunto de objetos contidos na informação de gerência de um agente é chamado *Management Information Base* ou MIB. Uma MIB normalmente consiste de vários módulos MIB. Um "módulo MIB" é um agrupamento de objetos relacionados. Por exemplo, há uma MIB padrão (chamada MIB-2) que todos os agentes devem suportar, independentemente do tipo do elemento gerenciado. Há também módulos MIB específicos para determinados tipos de elementos gerenciados. Assim, temos um módulo MIB especial para repetidores, outro para roteadores, outro específico para interfaces Ethernet, outro ainda para interfaces ATM e assim por diante. Portanto, um elemento gerenciado específico poderá, na sua MIB, ter o módulo MIB de roteadores, de interfaces Ethernet, além do módulo padrão MIB-2.

O mundo da gerência SNMP criou várias dezenas de módulos MIBs desde o início da década de 1990. Alguns são padronizados (ver adiante), enquanto outros são ditos patenteados. Um módulo MIB patenteado é elaborado pelo fabricante de um

elemento gerenciado particular para armazenar variáveis de gerência consideradas importantes, mas que não fazem parte dos padrões. Existe, por exemplo, módulos MIB especiais da Cisco presentes nos seus equipamentos. Na árvore de objetos, módulos MIB patenteados estão localizados embaixo de `iso.org.dod.internet.private.enterprises`.

3.3.4 A MIB-2

Todos os elementos gerenciados devem dar suporte a um módulo MIB básico chamado formalmente RFC1213-MIB ou, mais popularmente, MIB-2. A raiz do módulo MIB-2 é

```
iso.org.dod.internet.mgmt.mib-2
```

e os filhos imediatos de MIB-2 podem ser vistos na Figura A3-1. Esses filhos são chamados de grupos. Resumimos a informação mais importante presente na MIB-2 abaixo:

- Grupo system
  - Descrição do dispositivo (`sysDescr`)
  - Nome do dispositivo (`sysName`)
  - Há quanto tempo o agente está no ar (`sysUpTime`)
  - Localização física do dispositivo (`sysLocation`)
  - Pessoa responsável pelo elemento (`sysContact`)
- Grupo `interfaces`
  - Quantidade de interfaces (`ifNumber`)
  - A tabela de interfaces (`ifTable`)
    - Descrição da interface (`ifDescr`)
    - Tipo da interface (`ifType`)
    - Velocidade de transmissão (`ifSpeed`)
    - Endereço físico do meio (`ifPhysAddress`)
    - Contador de bytes na entrada (`ifInOctets`)
    - Contador de bytes na saída (`ifOutOctets`)
    - Contador de erros na entrada (`ifInErrors`)
    - Contador de erros na saída (`ifOutErrors`)

- Grupo `ip`
  - Vários contadores, endereços etc.
  - O mapa de tradução de endereços IP para endereços físicos
  - A tabela de roteamento do elemento gerenciado
- Grupo `icmp`
  - Vários contadores de mensagens enviadas e recebidas
  - Contadores por tipo de mensagem ICMP
  - Contador de erros etc.
- Grupo `tcp`
  - Identificador do algoritmo de retransmissão (`tcpRtoAlgorithm`)
  - Número máximo de conexões simultâneas permitidas (`tcpMaxConn`)
  - Número de segmentos enviados e recebidos além de vários outros contadores
  - A tabela de conexões abertas, indicando o estado da conexão, o endereço fonte (IP e porta) e o endereço destino (IP e porta)
- Grupo `udp`
  - Contador de datagramas destinados a portas desconhecidas (`udpNoPorts`)
  - Contadores de datagramas entrando (`udpInDatagrams`)
  - Contadores de datagramas saindo (`udpOutDatagrams`)
  - Vários outros contadores
- Grupo `snmp`
  - Várias informações (contadores etc.) sobre o protocolo SNMP

Como se pode observar, os objetos descritos brevemente são de vários tipos. Há Strings (`sysDescr`, por exemplo), inteiros (`ifNumber`, por exemplo) e contadores (`ifInOctets`, por exemplo), entre outros tipos. O tipo contador (`COUNTER`) merece especial atenção. Um valor de um contador é um valor absoluto contendo o número de eventos (digamos, chegada de um byte de entrada) que ocorreram desde um certo tempo no passado. Por si só, esse valor nada significa por dois motivos: 1) não sabemos o momento do início da contagem e, mais importante, 2) o contador tem 32 bits e pode sofrer *wrap-around* quando atinge o valor máximo de 4294967295. Depois desse valor, o contador volta a zero. Portanto, a forma correta de interpretar um contador é ler seu valor, esperar um certo tempo e ler o contador novamente. A diferença entre os dois valores pode ser usado para calcular a taxa de mudança no valor do contador. Por exemplo, num certo momento, o contador `ifInOctets` tem

valor 123456. Quinze minutos depois, o valor é 654377. Podemos calcular que a taxa média de tráfego de entrada no enlace em questão é:

Taxa de tráfego de entrada = (654377-123456) / (15 * 60) = 590 bytes/segundo

3.3.4.1Um walk num agente

Fazer um "walk" num agente significa varrer toda a árvore de variáveis de gerência de um agente e imprimir o valor de cada variável. O comando "snmpwalk", presente em máquinas UNIX ou Linux, pode ser usado para realizar a operação. Será instrutivo para o leitor examinar a saída desse comando, dado em uma estação de trabalho (um firewall com duas interfaces de rede). Eis a saída, com alguns trechos cortados:

```
system.sysDescr.0 = "Sun SPARCstation Solaris2.
   CheckPointFireWall-1 Version 2.1"
system.sysObjectID.0 = OID: enterprises.1919.1.1
system.sysUpTime.0 = Timeticks: (836503909) 96 days, 19:37:19
system.sysContact.0 = "José Joaquim da Silva Xavier"
system.sysName.0 = "fw.inconfidencia.mineira.br"
system.sysLocation.0 = "Sala 202"
system.sysServices.0 = 72
interfaces.ifNumber.0 = 3
interfaces.ifTable.ifEntry.ifIndex.1 = 1
interfaces.ifTable.ifEntry.ifIndex.2 = 2
interfaces.ifTable.ifEntry.ifIndex.3 = 3
interfaces.ifTable.ifEntry.ifDescr.1 = "lo0" Hex: 6C 6F 30
interfaces.ifTable.ifEntry.ifDescr.2 = "le0" Hex: 6C 65 30
interfaces.ifTable.ifEntry.ifDescr.3 = "le1" Hex: 6C 65 31
interfaces.ifTable.ifEntry.ifType.1 = softwareLoopback(24)
interfaces.ifTable.ifEntry.ifType.2 = ethernet-csmacd(6)
interfaces.ifTable.ifEntry.ifType.3 = ethernet-csmacd(6)
interfaces.ifTable.ifEntry.ifMtu.1 = 8232
interfaces.ifTable.ifEntry.ifMtu.2 = 1500
interfaces.ifTable.ifEntry.ifMtu.3 = 1500
interfaces.ifTable.ifEntry.ifSpeed.1 = Gauge: 10000000
interfaces.ifTable.ifEntry.ifSpeed.2 = Gauge: 10000000
interfaces.ifTable.ifEntry.ifSpeed.3 = Gauge: 10000000
interfaces.ifTable.ifEntry.ifPhysAddress.1 = ""
interfaces.ifTable.ifEntry.ifPhysAddress.2 = Hex: 08 00 20 7E 88 2B
interfaces.ifTable.ifEntry.ifPhysAddress.3 = Hex: 08 00 20 7E 88 2B
interfaces.ifTable.ifEntry.ifAdminStatus.1 = up(1)
```

```
interfaces.ifTable.ifEntry.ifAdminStatus.2 = up(1)
interfaces.ifTable.ifEntry.ifAdminStatus.3 = up(1)
interfaces.ifTable.ifEntry.ifOperStatus.1 = up(1)
interfaces.ifTable.ifEntry.ifOperStatus.2 = up(1)
interfaces.ifTable.ifEntry.ifOperStatus.3 = up(1)
interfaces.ifTable.ifEntry.ifLastChange.1 = Timeticks: (0) 0:00:00
interfaces.ifTable.ifEntry.ifLastChange.2 = Timeticks: (0) 0:00:00
interfaces.ifTable.ifEntry.ifLastChange.3 = Timeticks: (0) 0:00:00
interfaces.ifTable.ifEntry.ifInOctets.1 = 610783
interfaces.ifTable.ifEntry.ifInOctets.2 = 99903685
interfaces.ifTable.ifEntry.ifInOctets.3 = 94029823
interfaces.ifTable.ifEntry.ifInUcastPkts.1 = 0
interfaces.ifTable.ifEntry.ifInUcastPkts.2 = 0
interfaces.ifTable.ifEntry.ifInUcastPkts.3 = 0
interfaces.ifTable.ifEntry.ifInNUcastPkts.1 = 0
interfaces.ifTable.ifEntry.ifInNUcastPkts.2 = 0
interfaces.ifTable.ifEntry.ifInNUcastPkts.3 = 0
interfaces.ifTable.ifEntry.ifInDiscards.1 = 0
interfaces.ifTable.ifEntry.ifInDiscards.2 = 0
interfaces.ifTable.ifEntry.ifInDiscards.3 = 0
interfaces.ifTable.ifEntry.ifInErrors.1 = 0
interfaces.ifTable.ifEntry.ifInErrors.2 = 0
interfaces.ifTable.ifEntry.ifInErrors.3 = 0
interfaces.ifTable.ifEntry.ifInUnknownProtos.1 = 0
interfaces.ifTable.ifEntry.ifInUnknownProtos.2 = 0
interfaces.ifTable.ifEntry.ifInUnknownProtos.3 = 0
interfaces.ifTable.ifEntry.ifOutOctets.1 = 610783
interfaces.ifTable.ifEntry.ifOutOctets.2 = 98517639
interfaces.ifTable.ifEntry.ifOutOctets.3 = 88755644
interfaces.ifTable.ifEntry.ifOutUcastPkts.1 = 0
interfaces.ifTable.ifEntry.ifOutUcastPkts.2 = 0
interfaces.ifTable.ifEntry.ifOutUcastPkts.3 = 0
interfaces.ifTable.ifEntry.ifOutNUcastPkts.1 = 0
interfaces.ifTable.ifEntry.ifOutNUcastPkts.2 = 0
interfaces.ifTable.ifEntry.ifOutNUcastPkts.3 = 0
interfaces.ifTable.ifEntry.ifOutDiscards.1 = 0
interfaces.ifTable.ifEntry.ifOutDiscards.2 = 0
interfaces.ifTable.ifEntry.ifOutDiscards.3 = 0
interfaces.ifTable.ifEntry.ifOutErrors.1 = 0
interfaces.ifTable.ifEntry.ifOutErrors.2 = 5422
```

```
interfaces.ifTable.ifEntry.ifOutErrors.3 = 8
interfaces.ifTable.ifEntry.ifOutQLen.1 = Gauge: 0
interfaces.ifTable.ifEntry.ifOutQLen.2 = Gauge: 0
interfaces.ifTable.ifEntry.ifOutQLen.3 = Gauge: 0
interfaces.ifTable.ifEntry.ifSpecific.1 = OID: .ccitt.nullOID
interfaces.ifTable.ifEntry.ifSpecific.2 = OID: .ccitt.nullOID
interfaces.ifTable.ifEntry.ifSpecific.3 = OID: .ccitt.nullOID
ip.ipForwarding.0 = forwarding(1)
ip.ipDefaultTTL.0 = 255
ip.ipInReceives.0 = 189046788
ip.ipInHdrErrors.0 = 241
ip.ipInAddrErrors.0 = 0
ip.ipForwDatagrams.0 = 186087726
ip.ipInUnknownProtos.0 = 0
ip.ipInDiscards.0 = 783
ip.ipInDelivers.0 = 1384691
ip.ipOutRequests.0 = 904804
ip.ipOutDiscards.0 = 0
ip.ipOutNoRoutes.0 = 0
ip.ipReasmTimeout.0 = 60
ip.ipReasmReqds.0 = 1624
ip.ipReasmOKs.0 = 1624
ip.ipReasmFails.0 = 0
ip.ipFragOKs.0 = 4
ip.ipFragFails.0 = 0
ip.ipFragCreates.0 = 22
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.127.0.0.1 = IpAddress: 127.0.0.1
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.200.252.241.2 = IpAddress:
200.252.241.2
ip.ipAddrTable.ipAddrEntry.ipAdEntAddr.200.252.242.53 = IpAddress:
200.252.242.53
ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.127.0.0.1 = 1
ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.200.252.241.2 = 3
ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.200.252.242.53 = 2
ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.127.0.0.1 = IpAddress:
255.0.0.0
ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.200.252.241.2 = IpAddress:
255.255.255.248
ip.ipAddrTable.ipAddrEntry.ipAdEntNetMask.200.252.242.53 = IpAddress:
255.255.255.128
```

```
ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.127.0.0.1 = 0
ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.200.252.241.2 = 1
ip.ipAddrTable.ipAddrEntry.ipAdEntBcastAddr.200.252.242.53 = 1
icmp.icmpInMsgs.0 = 61390
icmp.icmpInErrors.0 = 0
icmp.icmpInDestUnreachs.0 = 27
icmp.icmpInTimeExcds.0 = 57101
icmp.icmpInParmProbs.0 = 0
icmp.icmpInSrcQuenchs.0 = 0
icmp.icmpInRedirects.0 = 0
icmp.icmpInEchos.0 = 4208
icmp.icmpInEchoReps.0 = 54
icmp.icmpInTimestamps.0 = 0
icmp.icmpInTimestampReps.0 = 0
icmp.icmpInAddrMasks.0 = 0
icmp.icmpInAddrMaskReps.0 = 0
icmp.icmpOutMsgs.0 = 182290
icmp.icmpOutErrors.0 = 0
icmp.icmpOutDestUnreachs.0 = 90030
icmp.icmpOutTimeExcds.0 = 87547
icmp.icmpOutParmProbs.0 = 0
icmp.icmpOutSrcQuenchs.0 = 0
icmp.icmpOutRedirects.0 = 505
icmp.icmpOutEchos.0 = 0
icmp.icmpOutEchoReps.0 = 4208
icmp.icmpOutTimestamps.0 = 0
icmp.icmpOutTimestampReps.0 = 0
icmp.icmpOutAddrMasks.0 = 0
icmp.icmpOutAddrMaskReps.0 = 0
tcp.tcpRtoAlgorithm.0 = vanj(4)
tcp.tcpRtoMin.0 = 200
tcp.tcpRtoMax.0 = 60000
tcp.tcpMaxConn.0 = -1
tcp.tcpActiveOpens.0 = 9892
tcp.tcpPassiveOpens.0 = 3575
tcp.tcpAttemptFails.0 = 175
tcp.tcpEstabResets.0 = 55
tcp.tcpCurrEstab.0 = Gauge: 2
tcp.tcpInSegs.0 = 145450
tcp.tcpOutSegs.0 = 108150
```

```
tcp.tcpRetransSegs.0 = 11268
...
tcp.tcpConnTable.tcpConnEntry.tcpConnSta-
te.127.0.0.1.32773.127.0.0.1.33692 = established(5)
...
tcp.tcpConnTable.tcpConnEntry.tcpConnLo-
calAddress.127.0.0.1.32773.127.0.0.1.33692 = IpAddress: 127.0.0.1
...
tcp.tcpConnTable.tcpConnEntry.tcpConnLocal-
Port.127.0.0.1.32773.127.0.0.1.33692 = 32773
...
tcp.tcpConnTable.tcpConnEntry.tcpConn-
RemAddress.127.0.0.1.32773.127.0.0.1.33692 = IpAddress: 127.0.0.1
...
tcp.tcpConnTable.tcpConnEntry.tcpConnRem-
Port.127.0.0.1.32773.127.0.0.1.33692 = 33692
...
udp.udpInDatagrams.0 = 1246911
udp.udpNoPorts.0 = 1246911
udp.udpInErrors.0 = 0
udp.udpOutDatagrams.0 = 638087
```

3.3.4.2 Definição da MIB-2

É importante que o administrador de uma rede aprenda a ler e interpretar o texto de uma MIB. Por exemplo, veja a seguir, o início da definição da MIB-2:

```
RFC1213-MIB DEFINITIONS ::= BEGIN
 -- This MIB module uses the extended OBJECT-TYPE macro as
 -- defined in [14];
 -- MIB-II (same prefix as MIB-I)
 mib-2   OBJECT IDENTIFIER ::= { mgmt 1 }

 -- groups in MIB-II
 system       OBJECT IDENTIFIER ::= { mib-2 1 }
 interfaces   OBJECT IDENTIFIER ::= { mib-2 2 }
 at           OBJECT IDENTIFIER ::= { mib-2 3 }
 ip           OBJECT IDENTIFIER ::= { mib-2 4 }
 icmp         OBJECT IDENTIFIER ::= { mib-2 5 }
 tcp          OBJECT IDENTIFIER ::= { mib-2 6 }
```

```
udp          OBJECT IDENTIFIER ::= { mib-2 7 }
egp          OBJECT IDENTIFIER ::= { mib-2 8 }
transmission OBJECT IDENTIFIER ::= { mib-2 10 }
snmp         OBJECT IDENTIFIER ::= { mib-2 11 }
```

Nesse trecho da MIB-2, podemos ver os valores dos objetos correspondendo aos grupos mib-2 sendo definidos. Por exemplo, `interfaces` é o filho de número 2 abaixo de `mib-2`. O seguinte trecho mostra uma parte da definição do grupo `system`.

```
-- the System group
-- Implementation of the System group is mandatory for all
-- systems. If an agent is not configured to have a value
-- for any of these variables, a string of length 0 is
-- returned.

sysDescr OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..255))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
  "A textual description of the entity. This value
  should include the full name and version
  identification of the system's hardware type,
  software operating-system, and networking
  software. It is mandatory that this only contain
  printable ASCII characters."
 ::= { system 1 }
```

No trecho anterior, SYNTAX define o tipo do objeto, ACCESS informa se pode ser lido e/ou alterado, STATUS não é mais usado e deve ser ignorado. A cláusula DESCRIPTION dá a semântica do objeto e é uma das partes mais importantes da MIB para o administrador de uma rede.

Continuando com a definição da MIB-2, mostramos a seguir um trecho da definição de grupo `interfaces`.

```
-- the Interfaces group

-- Implementation of the Interfaces group is mandatory for
-- all systems.

ifNumber OBJECT-TYPE
 SYNTAX INTEGER
```

```
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
  "The number of network interfaces (regardless of
  their current state) present on this system."
 ::= { interfaces 1 }

-- the Interfaces table

-- The Interfaces table contains information on the entity's
-- interfaces. Each interface is thought of as being
-- attached to a 'subnetwork'. Note that this term should
-- not be confused with 'subnet' which refers to an
-- addressing partitioning scheme used in the Internet suite
-- of protocols.

ifTable OBJECT-TYPE
 SYNTAX SEQUENCE OF IfEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
  "A list of interface entries. The number of
  entries is given by the value of ifNumber."
 ::= { interfaces 2 }

ifEntry OBJECT-TYPE
 SYNTAX IfEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
  "An interface entry containing objects at the
  subnetwork layer and below for a particular
  interface."
 INDEX  { ifIndex }
 ::= { ifTable 1 }

IfEntry ::=
 SEQUENCE {
  ifIndex     INTEGER,
  ifDescr     DisplayString,
```

```
  ifType     INTEGER,
  ifMtu      INTEGER,
  ifSpeed     Gauge,
  ifPhysAddress  PhysAddress,
  ifAdminStatus  INTEGER,
  ifOperStatus  INTEGER,
  ifLastChange  TimeTicks,
  ifInOctets   Counter,
  ifInUcastPkts  Counter,
  ifInNUcastPkts Counter,
  ifInDiscards  Counter,
  ifInErrors   Counter,
  ifInUnknownProtos Counter,
  ifOutOctets   Counter,
  ifOutUcastPkts Counter,
  ifOutNUcastPkts Counter,
  ifOutDiscards  Counter,
  ifOutErrors   Counter,
  ifOutQLen    Gauge,
  ifSpecific   OBJECT IDENTIFIER
 }

ifDescr OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..255))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
  "A textual string containing information about the
  interface. This string should include the name of
  the manufacturer, the product name and the version
  of the hardware interface."
 ::= { ifEntry 2 }

ifInOctets OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
  "The total number of octets received on the
  interface, including framing characters."
```

```
  ::= { ifEntry 10 }
...
END
```

3.4 Outras MIBs de gerência

Ao longo da década de 1990, dezenas de MIBs foram elaboradas para permitir a gerência de uma vasta gama de elementos. O conjunto de tais MIBs constitui um importantíssimo legado do mundo SNMP. Para dar o sabor do tipo de MIB em existência, listamos alguns módulos abaixo, além do seu número de RFC.

- ATM MIB (RFC 2515)
- Frame Relay DTE Interface Type MIB (RFC 2115)
- BGP version 4 MIB (RFC 1657)
- RDBMS MIB (RFC 1697)
- RADIUS Authentication Server MIB (RFC 2619)
- Mail Monitoring MIB (RFC 2789)
- RMON MIB (RFC 1757)
- Uninterruptible Power Supply MIB (RFC 1628)

Uma lista mais completa de MIBs padronizadas pode ser vista no endereço http://www.simple-times.org.

3.5 O protocolo SNMP

Agora que conhecemos um pouco mais como a informação de gerência está estruturada, podemos dar alguns detalhes adicionais sobre o protocolo SNMP. O SNMP existe em 3 versões, embora a versão inicial de número 1 seja a única versão realmente difundida até hoje. As *Protocol Data Units* (PDUs) do protocolo SNMP versão 1 são brevemente descritas a seguir.

- PDU GET. Serve para o gerente enviar a um agente um pedido de leitura de uma variável de gerência. A PDU contém o *Object Identifier* (OID) da instância a ser lida.
- PDU RESPONSE. O agente responde às PDUs GET, GET-NEXT e SET com uma resposta contida numa PDU deste tipo.
- PDU SET. Serve para o gerente enviar a um agente um pedido de alteração de uma variável de gerência.
- PDU GET-NEXT. Em alguns casos, a PDU GET não pode ser usada par ler

uma variável de gerência, porque o OID não é conhecido. Imagine, por exemplo, ler informação da tabela de interfaces `ifTable` discutida anteriormente. Como o gerente não sabe os valores da chave `ifIndex` que correspondem a cada interface de rede, ele é obrigado a usar a PDU GET-NEXT que permite varrer a tabela seqüencialmente sem saber os OIDs exatos contidas nela.

- PDU TRAP. Esta PDU é usada pelo agente para informar eventos extraordinários ao gerente.

3.5.1 A segurança no mundo SNMP

Infelizmente, o SNMP sofre de uma deficiência grave que tem freado bastante seu uso para controlar redes e não apenas monitorá-las. O problema é que a segurança do protocolo se baseia no uso de senhas (chamados *community names* no mundo SNMP). Elementos gerenciados tipicamente oferecem duas visões de sua informação de gerência: a comunidade de leitura e a comunidade de escrita. Cada comunidade no elemento gerenciado é controlada por um *community name*. Ao enviar uma PDU para o agente, o *community name* deve ser informado na PDU. Se o nome estiver correto, o agente aceitará a PDU e realizará a operação pedida.

O problema é que o *community name* trafega em cada PDU SNMP e sem uso de criptografia. Esse fato deixa o protocolo extremamente inseguro e a maioria das empresas desabilita a comunidade de escrita de seus elementos gerenciados de forma a não comprometer em demasia a segurança da rede.

A versão 3 do protocolo SNMP resolve o problema de segurança, mas ainda não é muito empregada no mercado.

3.6 Bibliografia

Mauro, D. R. e K. J. Schmidt. *SNMP Essencial*. Editora Campus, 2001.
Harnedy, S. *Total SNMP*. Prentice Hall, 1998.
Zeltserman, D. A Practical Guide to SNMPv3 and Network Management. Prentice Hall, 1999.

APÊNDICE

4

4 Conhecendo repetidores, comutadores e roteadores

4.1 Introdução

Uma rede de computadores inclui hospedeiros (estações de trabalho e servidores) interconectados. A comunicação entre os hospedeiros é realizada usando *equipamentos de interconexão*, sendo os mais importantes repetidores (*hubs*), comutadores[1] (*switches*) e roteadores. Embora os três tipos de equipamentos sirvam para formar a sub-rede de comunicação, eles têm diferenças importantes que indicam sua utilização em situações particulares. As diferenças entre os tipos de equipamentos ocorrem em várias dimensões:

- Na **escalabilidade**, isto é, no tamanho de sub-rede de comunicação que pode ser realizada com o equipamento: repetidores são usados para montar redes pequenas, comutadores para fazer redes maiores e roteadores para redes ainda maiores.
- No **alcance geográfico** atingível: repetidores e comutadores são usados para redes de campus[2] e roteadores são usados tanto em redes de campus quanto em redes de longo alcance.
- Na **camada de protocolo**[3] onde atuam: repetidores atuam na camada física, comutadores na camada de enlace e roteadores na camada de rede. Veja a Figura A4-1.

[1] Observem que a palavra "comutador" está sendo empregada para indicar apenas "comutadores de tecnologia Ethernet" utilizadas em redes de campus.
[2] Uma rede de campus é uma rede de alta velocidade abrangendo vários prédios localizados numa área geograficamente restrita, e podendo incluir centenas ou milhares de hospedeiros.
[3] Fazemos referência aqui às sete camadas que compõem a arquitetura de redes estabelecida pela ISO no modelo RM-OSI.

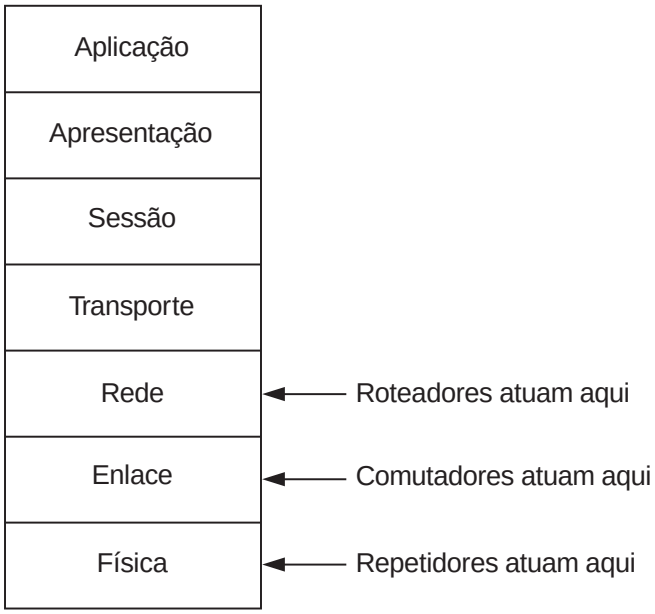


FIGURA A4-1: As camadas do modelo RM-OSI da ISO.

- No **preço**: repetidores são tipicamente mais baratos do que comutadores que são tipicamente mais baratos do que roteadores.
- Na **sofisticação dos serviços** oferecidos: geralmente, os serviços vão crescendo em sofisticação à medida que se passa de repetidor para comutador, e deste para roteador.

No restante deste capítulo veremos as diferenças entre esses três tipos de equipamentos.

4.2 Repetidores

Repetidores Ethernet também são freqüentemente chamados de *concentradores*, *hubs* ou então de *hubs repetidores*. Desde o aparecimento da tecnologia Fast Ethernet a 100 Mbps e do barateamento de comutadores, repetidores têm sido usado cada vez menos. Porém, eles foram largamente utilizados no passado e ainda são ubíquos em redes de campus, principalmente na camada de acesso[4] (conectados diretamente a estações de trabalho) e funcionando a 10 Mbps.

[4] A palavra "camada" aqui não faz referência ao modelo RM-OSI de sete camadas. Trata-se de uma forma comum de organizar redes de campus, usando três camadas: "camada core", "camada de distribuição" e "camada de acesso".

4.2.1 Princípios de operação

Nesta seção, vamos simplificar a discussão e considerar uma rede pequena com um único repetidor. Um repetidor possui várias portas para interconectar equipamentos (ver a Figura A4-2). Qualquer um dos hospedeiros conectados ao repetidor pode se comunicar com outro pelo simples envio de um quadro para o hospedeiro destino. O repetidor recebe o sinal enviado e o retransmite em *cada uma* das outras portas. Simplificando um pouco a realidade, o repetidor é simplesmente um amplificador de sinais: o que é recebido numa porta é amplificado e retransmitido instantaneamente em todas as outras portas. É uma evolução do segmento Ethernet, que usava cabos coaxiais (tecnologias 10BASE-5 e 10BASE-2) para uma solução em que o segmento (o cabo) está logicamente presente dentro do repetidor e cada hospedeiro se conecta individualmente ao segmento com seu próprio cabo, tipicamente usando cabos de pares trançados ou fibras óticas.

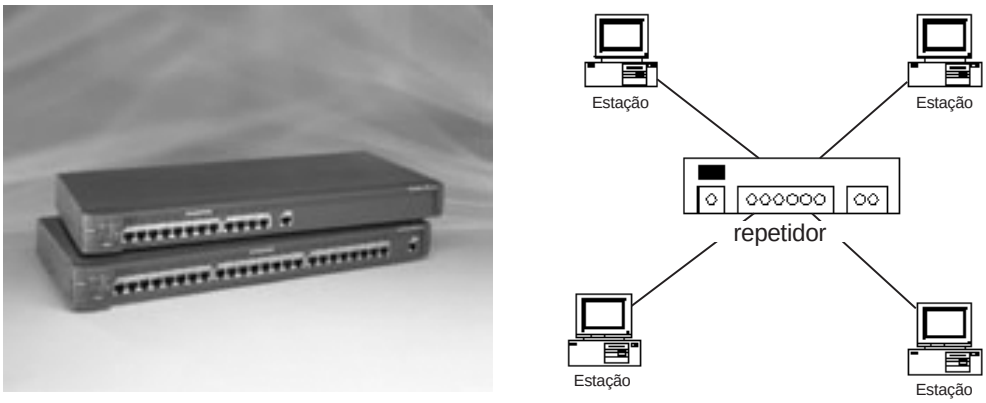


FIGURA A4-2: Um repetidor Cisco Fasthub 400 e visão lógica.

Naturalmente, devido à operação do repetidor, podemos afirmar que:

- O segmento Ethernet é **compartilhado** entre todos os hospedeiros. Se o repetidor for de 10 Mbps, então há uma banda passante total de 10 Mbps para o tráfego total dos hospedeiros;
- Apenas um hospedeiro poderá estar transmitindo de cada vez (a operação é **half-duplex**);
- **Colisões** poderão ocorrer no segmento;
- Todas as portas deverão operar usando a **mesma tecnologia** Ethernet (tipicamente 10BASE-T ou 100BASE-TX).

Chamar o repetidor de "amplificador de sinal" é uma simplificação da realidade. O repetidor ainda realiza as seguintes operações:

- **Regenera** o sinal em cada porta, em vez de apenas amplificá-lo;
- Faz "**imposição de colisão**" para ter certeza de que uma colisão é detectada sem ambigüidade por todos os hospedeiros envolvidos;
- Realiza o **autoparticionamento** de portas, isto é, isola do segmento qualquer porta que esteja causando problemas. Devido à operação do repetidor na camada física, qualquer falha presente numa das portas do repetidor afetará todos os hospedeiros conectados ao repetidor. Um cabo partido, um conector defeituoso, uma placa de rede defeituosa podem causar falhas tais como tempo excessivo de colisão, interferência no sinal etc. O repetidor realiza o autoparticionamento de uma porta onde 30 falhas consecutivas forem detectadas. A porta é restaurada ao segmento assim que um único quadro sem defeito for detectado na porta.

Finalmente, é importante observar que um repetidor é **transparente**. Os hospedeiros não sabem de sua existência, no sentido que nunca o endereçam. Na realidade, um repetidor não possui endereço[5] (nem endereço MAC, nem endereço IP).

4.2.2 Redes com múltiplos repetidores

Um repetidor tem um número limite de portas. O que fazer para construir redes maiores utilizando repetidores? A solução é utilizar vários repetidores, um conectado ao outro de alguma forma. A forma mais simples é utilizar uma das portas do repetidor (na próxima subseção, veremos outras alternativas). Veja a Figura A4-3. Nesta figura, o leitor vai observar que há menção a um *cabo cruzado*. Em repetidores que utilizam pares trançados, o cabo que conecta um hospedeiro a uma porta do repetidor é um cabo paralelo (terminal 1 de um lado do cabo está conectado ao terminal 1 do outro conector, terminal 2 com 2 etc.). Porém, para conectar um repetidor com outro, deve-se utilizar um cabo cruzado para que a comunicação se efetue corretamente. Alternativamente, pode-se utilizar um cabo paralelo desde que se utilize uma porta especial já cruzada (marcada *uplink*) de um dos repetidores.

Esta forma de aumentar o número de hospedeiros utilizando vários repetidores interconectados tem limites:

- Primeiro, a tecnologia empregada (10BASE-T, 100BASE-TX etc) estabelece um limite tanto no número de "saltos de repetidor" que podem existir entre dois hospedeiros quaisquer como também impõe um limite no comprimento dos cabos. Os detalhes variam com a tecnologia empregada; detalhes podem ser vistos em [GUIA-ETHERNET].

[5] A única exceção a essa regra ocorre quando o repetidor é gerenciável, em cujo caso ele precisa ser endereçado pela estação de gerência que deseja obter informação a respeito do repetidor. Mesmo assim, ele continua invisível às estações a ele conectadas.

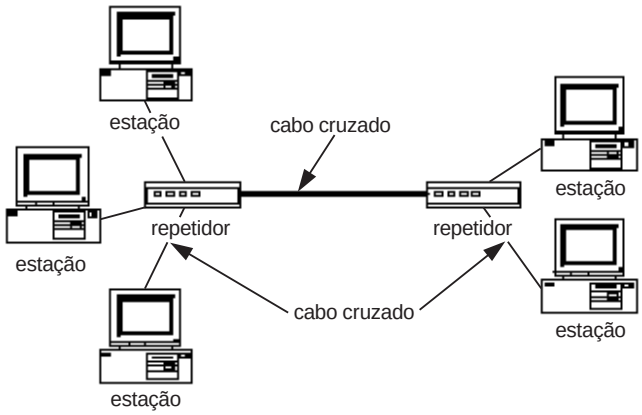


FIGURA A4-3: Rede com 2 repetidores.

- Segundo, toda a rede assim formada consiste de um **único segmento compartilhado**. Portanto, a banda passante disponível (digamos 10 Mbps) não é aumentada e deve ser compartilhada entre mais hospedeiros, o que aumenta as chances de saturação do segmento.
- Terceiro, ainda como conseqüência do segmento único, a rede inteira assim formada consiste de um **único domínio de colisão**. Qualquer hospedeiro, localizado em qualquer um dos repetidores, poderá ter sua transmissão colidindo com a transmissão de qualquer outro hospedeiro. O incremento de hospedeiros causará, portanto, um incremento na taxa de colisões.

4.2.3 Características especiais de repetidores

Repetidores não nascem iguais. Entre as características que diferenciam os vários modelos disponíveis, podemos citar:

- O número de portas disponíveis (tipicamente 12 ou 24);
- A tecnologia empregada nas portas (tipicamente 10BASE-T e/ou 100BASE-TX);
- Modelo chassi (expansão via módulos; emprega uma única fonte de alimentação para todos os módulos);
- Modelo empilhável (repetidores individuais, cada um com sua própria fonte de alimentação conectados através de um conector especial de expansão; os repetidores assim conectados agem como único repetidor lógico, valendo um único "salto de repetidor");
- Suporte ao *autosense* (para estabelecer a velocidade 10 Mbps ou 100 Mbps);
- Porta *uplink* com fibra óptica;

- Suporte à gerência via protocolo SNMP, podendo oferecer suporte às seguintes *Management Information Bases* (MIBs): MIB-II, Repeater MIB, Ethernet MIB, RMON MIB;
- Servidor Web embutido para facilitar a gerência através de um browser.

4.2.4 Conclusão

Repetidores estão caindo em desuso, devido principalmente ao barateamento de comutadores, mais vantajosos em termos de desempenho. Podemos dizer que hoje (em 2003), repetidores raramente são empregados em *novas* redes de campus, a não ser em redes muito pequenas com baixo tráfego agregado. Há, por outro lado, uma quantidade enorme de repetidores em uso em redes mais antigas.

4.3 Comutadores

Nesta seção, examinamos um segundo tipo de equipamento de interconexão: o comutador Ethernet, ou simplesmente comutador. É o equipamento mais freqüentemente empregado para montar redes de campus de alta velocidade.

4.3.1 Por que outro tipo de equipamento?

Os repetidores que descrevemos na última seção têm a grande vantagem da simplicidade e do baixo custo. Se utilizássemos apenas repetidores para formar redes, conectando-os entre si[6], o que ocorreria? Seria possível? Três restrições, já mencionadas, fazem com que uma rede não possa ser montada apenas com repetidores, para atingir qualquer tamanho:

- A tecnologia Ethernet utiliza o protocolo CSMA-CD (ver Apêndice 1) e esta forma de acesso ao meio impõe **restrições sobre o tempo máximo** que o sinal elétrico pode levar entre duas extremidades da rede. Portanto, há um limite físico, geográfico, digamos, para a extensão de uma rede feita de repetidores.
- Uma rede montada apenas com repetidores forma um **único segmento compartilhado que satura** com a adição de mais fontes de tráfego. Por exemplo, numa rede de repetidores de 10 Mbps, há uma banda passante *total* de 10 Mbps para todos os hospedeiros compartilharem. Nos anos 1980, antes da Internet e, principalmente, da Web, era possível atender a centenas de usuários com apenas esta banda passante. Porém, novos tipos de tráfego (imagens

[6] Sem formar laços, pois isto causaria uma pane geral na rede devido à transmissão em laço dos dados entre os repetidores.

da Web, videoconferência, voz sobre IP etc.) requerem muito mais banda passante, saturando assim uma rede de 10 Mbps com poucas dezenas de usuários. A tecnologia Fast Ethernet (que opera a 100 Mbps) alivia a situação, mas não a resolve. Simplesmente empurra o limite em que a saturação ocorrerá.

- Uma rede montada apenas com repetidores forma um **único domínio de colisão**. Com um grande acréscimo no número de hospedeiros querendo conversar no meio compartilhado, a probabilidade de colisão aumenta até inviabilizar o uso da rede.

Precisamos, portanto, de um novo tipo de equipamento de interconexão que não sofra dos problemas mencionados e que permita fazer redes de campus maiores, abrangendo centenas ou milhares de hospedeiros. Tais equipamentos deverão obedecer aos seguintes requisitos básicos:

- Devem oferecer **banda passante dedicada** em cada porta para evitar a saturação dos enlaces;
- Tal como o repetidor, o comutador deve ser transparente: os hospedeiros não sabem de sua existência, no sentido que nunca o endereçam.[7]

Este equipamento chama-se *comutador*[8] (vide Figura 4-4) e descrevemos seu princípio de operação na próxima seção.

FIGURA A4-4: O comutador Catalyst 3550 da Cisco.

[7] Mais uma vez, tal como no caso do repetidor, para poder gerenciar um comutador, ele deve ser endereçável. Porém, ele só será endereçado pelas estações de gerência e continua transparente aos demais hospedeiros da rede.

[8] Observe que o termo "comutador" também se aplica a outras tecnologias tais como ATM. Consideramos aqui apenas comutadores com tecnologia Ethernet. Este equipamento é freqüentemente chamado "Ethernet switch" ("switches", no plural).

4.3.2 Princípios de operação

O princípio básico no qual o comutador se apóia é impedir que um sinal recebido numa porta seja imediatamente retransmitido em outra porta. O comutador recebe o sinal numa porta, reconhece o quadro, armazena-o internamente, escolhe a porta de destino do quadro e o reencaminha para esta porta. É importante observar que o reenvio do quadro recebido na porta de saída ocorrerá em momento distinto da recepção do quadro na porta original. Portanto, podemos ver que o comutador é um equipamento do tipo armazenamento-e-reenvio (*store-and-forward*).

Neste tipo de equipamento, há três atividades distintas ocorrendo: recepção, escolha do destino e reenvio. Essas atividades ocorrem em momentos distintos, o que implica que outra camada de protocolos, além da camada física, entrou em ação. De fato, o comutador age na camada de enlace (camada 2). Aliás, a palavra "comutador" indica que a ação de escolha do destino é realizada na camada de enlace. Sabemos que uma camada adicional entrou em ação porque a escolha do destino (comutação) necessitou examinar o quadro para descobrir o endereço de destino. Os próprios conceitos de "quadro" e "endereço de destino" não existem na camada física e estão entre as principais diferenças entre as camadas física e de enlace.

4.3.3 O algoritmo de comutação

Há ainda uma mágica a ser explicada: de que forma o comutador sabe para qual porta destino encaminhar o quadro? (Vide Figura A4-5). A única informação disponível sobre o destino é o endereço MAC, que não contém informação de roteamento (ou encaminhamento). Já que queremos que o comutador seja transparente, estamos proibidos de configurá-lo previamente para informar que endereços MAC são atingidos em que portas.

O algoritmo de encaminhamento usa uma tabela de encaminhamento mantida no comutador. A tabela de encaminhamento tem uma linha para cada endereço MAC conhecido e mantém a porta a ser utilizada para alcançar o endereço associado. Por exemplo, na Figura A4-6, o equipamento com endereço MAC F é alcançado através da porta 6. Inicialmente, a tabela de encaminhamento está vazia. Usando uma técnica que explicaremos logo, o comutador vai aprendendo onde estão os equipamentos com determinados endereços MAC e preenche a tabela de encaminhamento. Num determinado momento, o comutador poderá conhecer a informação relativa a alguns endereços MAC, mas não a todos. Podemos agora descrever o algoritmo de encaminhamento:

Ao receber um quadro destinado ao endereço MAC X, o comutador encaminha o quadro à porta associada a esse endereço na tabela de encaminhamento, caso exista. Quando não existe, o quadro é encaminhado em *todas* as portas, menos naquela na qual chegou.

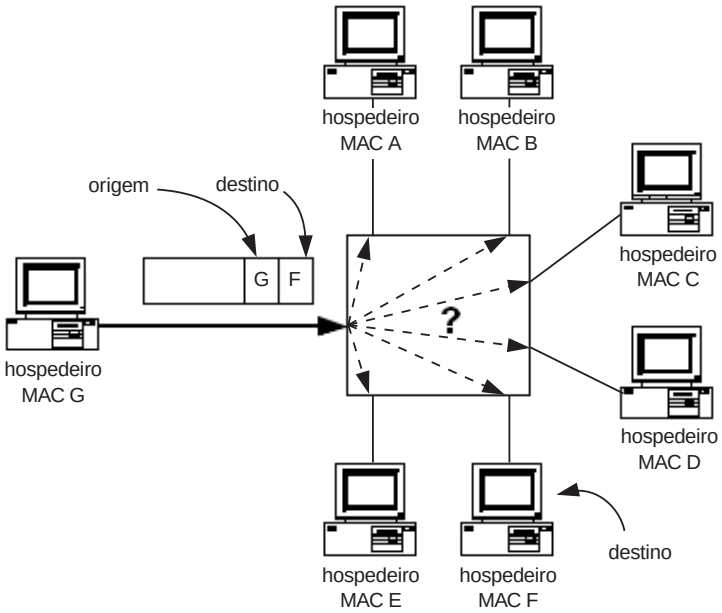


FIGURA A4-5: Decisão de encaminhamento de quadro.

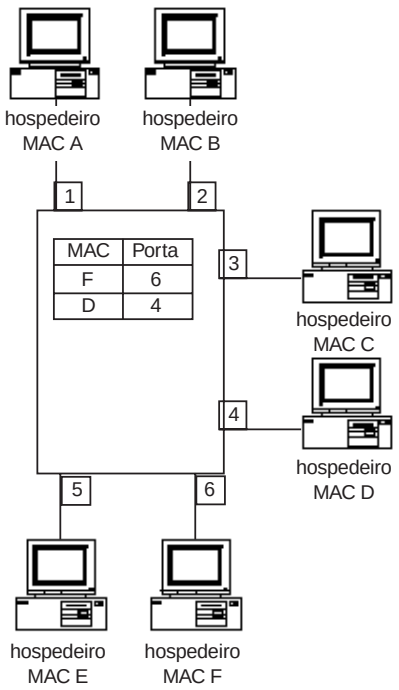


FIGURA A4-6: A tabela de encaminhamento de um comutador.

A técnica de enviar o quadro em todas as portas menos pela qual chegou, chama-se "inundação" (*flooding*, em inglês).

Resta-nos explicar como o comutador aprende as correspondências que ele mantém na tabela de encaminhamento. A técnica se chama "aprendizado para trás" (*backward learning*) e é bastante simples: ao receber um quadro na porta P com endereço origem E_o e endereço destino E_d, o comutador, embora talvez não saiba como alcançar o destino E_d, sabe que a *origem* E_o é alcançada pela porta P (já que o quadro veio de lá). Portanto, o comutador coloca o par (E_o, P) na tabela de encaminhamento. Num momento futuro, quando chegar um quadro endereçado a E_o, o comutador o encaminhará diretamente à porta P sem usar uma enchente.

A Figura A4-7 explica melhor a técnica de aprendizagem para trás. Para complicar um pouco mais, temos dois comutadores envolvidos. O hospedeiro com endereço MAC F quer enviar um quadro para o hospedeiro com MAC G. Na parte a da Figura, o primeiro comutador (à esquerda) usa uma enchente para alcançar G, já que não possui G na tabela de encaminhamento. O quadro chega ao segundo comutador através das portas 3 e 7. O segundo comutador também tem tabela de encaminhamento vazia e usa uma enchente para alcançar G. Na parte b da figura, podemos ver que ambos os comutadores aprenderam como chegar ao endereço MAC F através de "aprendizagem para trás". Na parte c da figura, o hospedeiro G responde com um quadro para F. Como o endereço MAC F está na tabela de ambos os comutadores, nenhum deles precisa usar inundação e o quadro é comutado diretamente para a porta correta (a porta 7 para o comutador da direita e a porta 6 para o comutador da esquerda). Finalmente, na parte d da figura, ambos os comutadores aprenderam como chegar ao hospedeiro com MAC G.

Os algoritmos descritos (de encaminhamento e de aprendizagem) apareceram num tipo de equipamento chamado *ponte* (bridge). Na realidade, os comutadores Ethernet objetos de nossa discussão são pontes com múltiplas portas. A palavra *ponte* não é mais usada corriqueiramente e emprega-se a palavra *comutador*.

Duas observações encerram esta subseção: primeiro, vale notar que quadros de difusão e de multicast serão sempre encaminhados por inundação pelos comutadores. Segundo, os dois algoritmos do comutador são sempre *implementados em* hardware para permitir que o comutador tenha desempenho suficiente para comutar os quadros que chegarem nas várias portas na velocidade nominal das portas. Dizemos, então, que o comutador pode operar em *wire speed*.

4.3.4 Conseqüências do modo de operação de um comutador

Nesta seção, examinamos algumas das conseqüências do modo de operação de um comutador explicado na seção anterior.

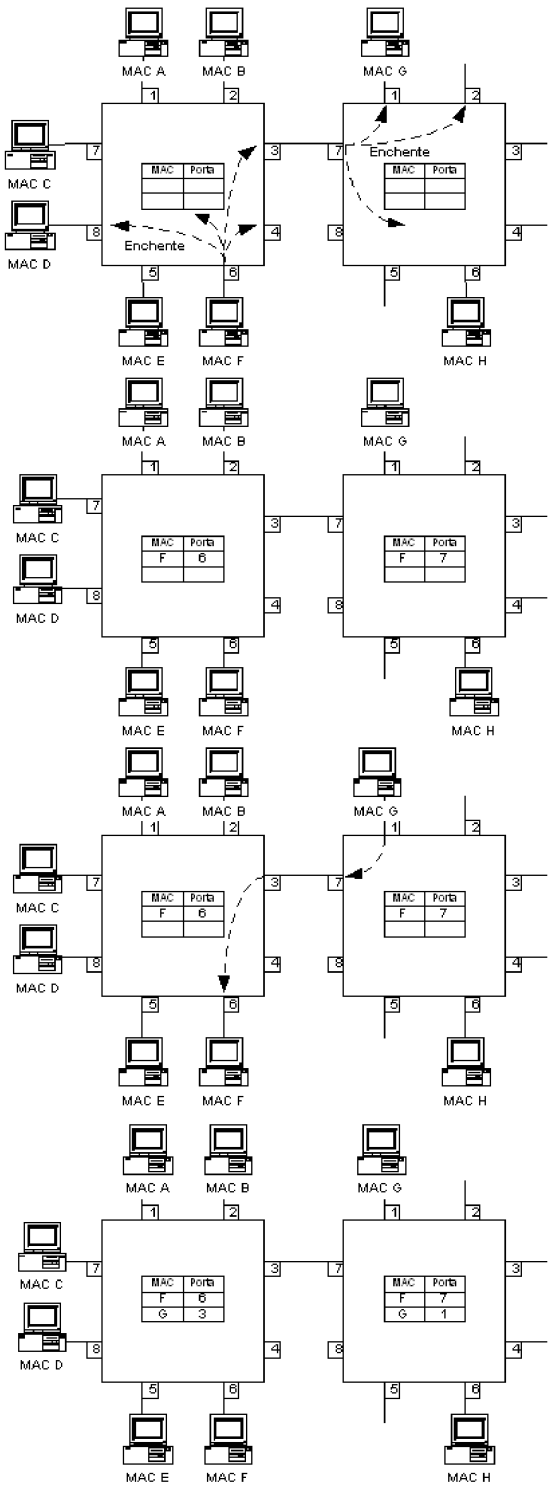


FIGURA A4-7: Aprendizagem de encaminhamento.

Comunicação paralela. O fato de o comutador ser um equipamento do tipo armazenamento-e-reenvio significa que várias comunicações paralelas poderão estar ocorrendo simultaneamente nas várias portas do equipamento. Na Figura A4-8, podemos ver 3 quadros diferentes sendo transmitidos em portas do comutador (portas 1, 2, 4, 5 e 7).

Comunicação full-duplex. Imagine que, numa porta de um comutador, esteja conectado um hospedeiro apenas. Nesta porta, apenas dois equipamentos podem transmitir: o comutador ou o hospedeiro. Portanto, há possibilidade de utilizar o meio de comunicação em modo *full-duplex*, permitindo que haja transmissão nos dois sentidos ao mesmo tempo. Para que isso ocorra, a porta do comutador deve ser configurada para comunicação *full-duplex*, o que pode ser feito manualmente ou através de autonegociação entre o comutador e o hospedeiro. A comunicação *full-duplex* também pode ocorrer se houver um outro comutador conectado à porta do comutador original, ou um roteador; enfim, qualquer equipamento que não seja um repetidor pode ser usado com comunicação *full-duplex*. Porém, quando um repetidor estiver conectado a uma porta de um comutador, esta porta deverá funcionar em modo *half-duplex*.

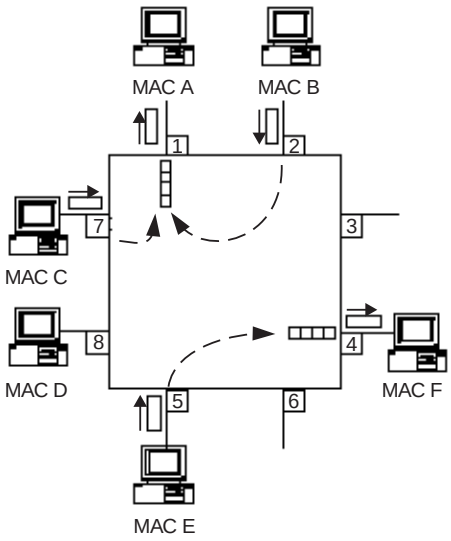


FIGURA A4-8: Transmissões simultâneas num comutador.

Domínios de colisão. Devido à operação de armazenamento-e-reenvio, cada porta de um comutador é um domínio de colisão independente (ver Figura A4-9). Onde há um repetidor, a comunicação deve ocorrer no modo *half-duplex*, e colisões poderão ocorrer. Caso não haja repetidor conectado a uma porta, a comunicação nessa porta poderá ocorrer no modo *full-duplex*, e não haverá colisões no segmento. Se, na Figura A4-9, o comutador fosse um repetidor, existiria um único domínio de colisão envolvendo todos os equipamentos mostrados na figura.

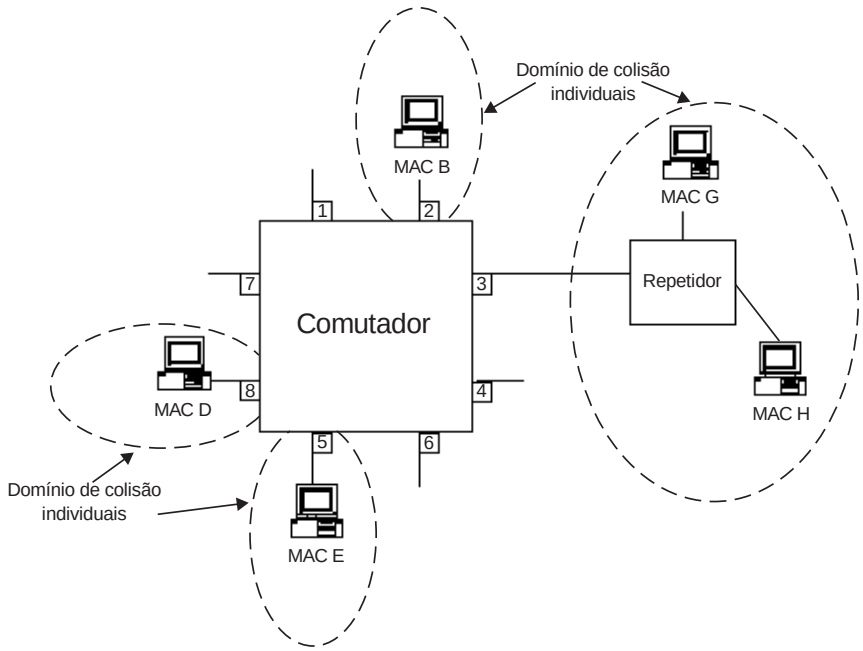


FIGURA A4-9: Domínios de colisão num comutador.

Portas com velocidades e tecnologias diferentes. Já que a retransmissão de uma porta para outra não é imediata, as portas de um comutador não precisam necessariamente operar na mesma velocidade. Podemos ter, por exemplo, um comutador com algumas portas operando a 10 Mbps, outras a 100 Mbps e outras ainda a 1 Gbps. Contraste essa situação com o repetidor no qual todas as portas devem necessariamente operar na mesma velocidade. Obviamente, uma conseqüência disso é que as portas do comutador podem ser de tecnologias diferentes, misturando, por exemplo, 10BASE-T com 100BASE-TX, ou ainda portas com pares trançados e outras com fibra óptica.

Formação de filas. Examinando a Figura A4-8 novamente, o leitor poderá observar que duas transmissões estão sendo feitas para o hospedeiro "MAC A": elas vêm das portas 7 e 2 do comutador. Não há como realizar o envio *simultâneo* dos quadros provenientes dessas para a porta de saída (porta 1). Portanto, um dos quadros deverá ficar em *fila*, esperando sua vez. A formação de filas é característica de qualquer equipamento do tipo armazenamento-e-reenvio. A formação de filas ocorre por vários motivos:

- O tráfego chega de forma aleatória e simultaneamente em várias portas e pode se dirigir a uma mesma porta de saída;
- As portas podem funcionar em velocidades diferentes; se uma porta veloz estiver recebendo tráfego indo para uma porta mais lenta, poderá haver formação de fila.

A formação de filas é normal e não implica defeito de funcionamento. Há, porém, duas conseqüências da formação de filas: um aumento de atraso na informação que pode, com grandes filas, ser sentido pelo usuário; e uma possibilidade de perda de informação, já que o comutador possui memória limitada para armazenar os quadros em fila. Um quadro recebido corretamente por um comutador e não reenviado por falta de memória para colocá-lo em fila é chamado de *quadro descartado*.

Domínio de difusão único. Lembre-se de que quadros de difusão e de multicast são sempre encaminhados por inundação por um comutador. Como conseqüência, uma rede formada com um ou mais comutadores e repetidores forma um *único domínio de difusão*. Um quadro de difusão irá para todos os equipamentos que possam ser alcançados, passando por repetidores e comutadores. Diz-se também – e isso significa a mesma coisa – que "um comutador não pára a difusão".

Tratamento de laços. Laços envolvendo comutadores podem causar um problema. Na Figura A4-10, o hospedeiro "MAC A" manda um quadro para "MAC H". Num primeiro momento (ver o número 1 num pequeno círculo), o quadro é enviado para o comutador na porta 1. Supondo que o comutador utilize uma inundação para encaminhar o quadro, este irá para todas as portas, incluindo 3 e 4, ambas conectadas a um segundo comutador (formando um laço). O segundo comutador recebe um quadro na porta 7 e, digamos, também utiliza inundação para encaminhá-lo. Portanto, este quadro irá para a porta 6 (como desejado) e para a porta 8, voltando para o comutador original. O mesmo ocorre com o quadro recebido na porta 8 do segundo comutador. Não é difícil verificar que o comutador original usará inundação novamente para encaminhar esses quadros e que um laço infinito de quadros resulta. Este comportamento deve ser evitado a todo custo. A primeira solução consiste em nunca formar laços envolvendo comutadores. Entretanto, essa não é uma boa solução, pois, como mostraremos logo a seguir, é altamente desejável utilizar laços em redes. A segunda solução é de "cortar os laços" automaticamente, desabilitando o encaminhamento em certas portas do comutador. Isso é feito pelo protocolo de árvore de cobertura (*spanning tree protocol*). Através desse protocolo, os comutadores conversam entre si e definem que portas de que comutadores deverão deixar de encaminhar quadros, evitando assim os laços. É extremamente importante, portanto, que o protocolo de árvore de cobertura esteja habilitado em todos os comutadores da rede. Detalhes sobre esse protocolo podem ser vistos no Apêndice 6.

Mas, perguntará o leitor atento, se o protocolo de árvore de cobertura remove os laços, de que serve colocar laços desde o início? A resposta é que precisamos de laços para obter redundância. Redes estão ficando mais importantes para as empresas e é importante que elas continuem funcionando apesar da existência de problemas na rede. Uma das formas mais utilizadas de incluir redundância numa rede é de formar laços, introduzindo assim *rotas alternativas* para que quadros cheguem ao destino desejado. Por exemplo, a Figura A4-11 mostra como uma rede de campus é

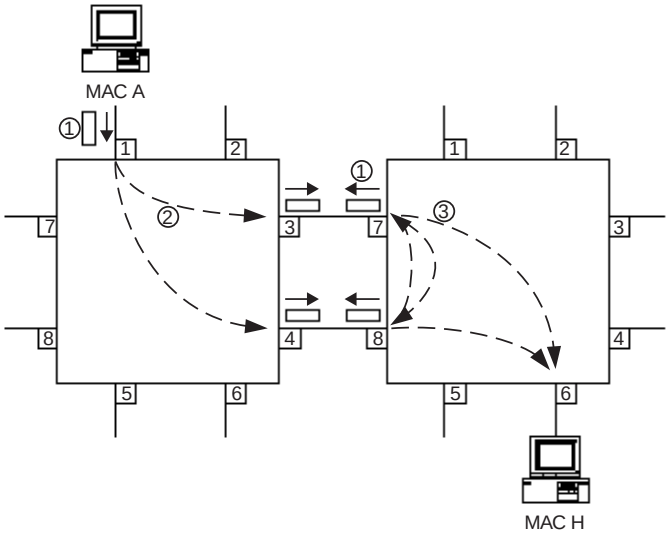


FIGURA A4-10: Um laço envolvendo comutadores.

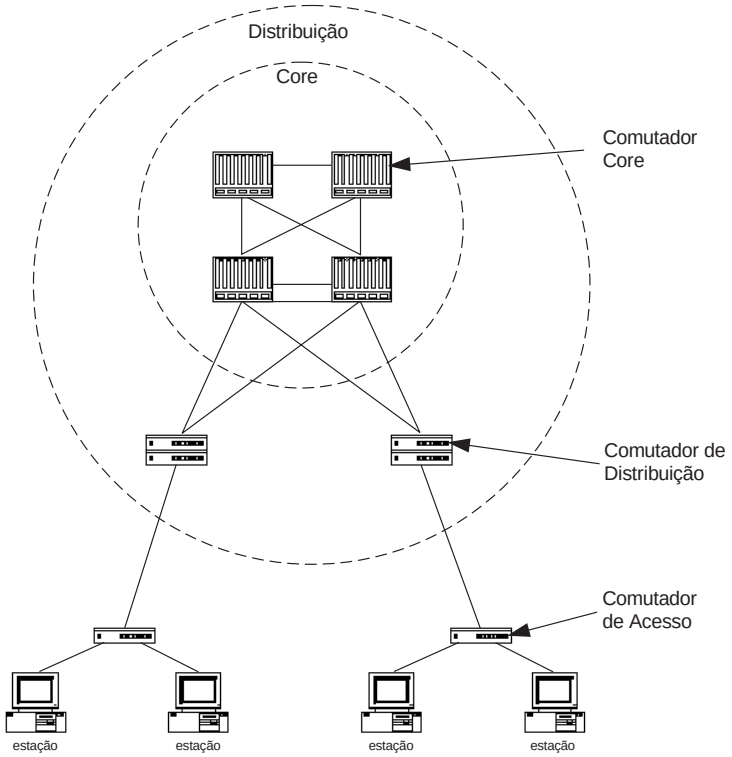


FIGURA A4-11: Redundância nos enlaces de uma rede de campus.

tipicamente projetada: vários caminhos redundantes existem nas camadas *core* e de distribuição. Nessa figura, o protocolo de árvore de cobertura eliminará os laços, desabilitando o encaminhamento de quadros em certas portas. Porém, ao detectar a perda de um enlace, o protocolo refaz todos os cálculos e escolhe desabilitar portas diferentes, de forma a manter a comunicação entre todos os comutadores, mas sem a presença de laços.

4.3.5 Características especiais de comutadores

Comutadores diferem muito entre si. A primeira grande diferença é decorrente do uso que se faz da palavra "comutador". Inicialmente, os comutadores eram tais como descritos nesta seção. Com tempo, os comutadores adquiriram funções de roteamento e passaram a ser chamados "comutadores de nível 3". Também apareceram "comutadores de nível 4 e 7". Para minimizar a confusão, esta seção fala apenas de comutadores "feijão-com-arroz", que também chamamos "comutadores de nível 2". Discutimos apenas esses comutadores aqui. A próxima seção abordará comutadores avançados, de níveis 3 e superiores.

Enumeramos, a seguir, algumas características que diferenciam os vários modelos disponíveis:

- O número de portas disponíveis (de 8 a várias centenas de portas);
- A tecnologia empregada nas portas (tipicamente, 10/100BaseTX, 100BaseFX com fibra monomodo ou multimodo, Gigabit Ethernet (SX LX/LH, e ZX), 10 Gigabit Ethernet (LR, EX4));
- A capacidade variada de comutação, podendo alcançar centenas de Gbps e centenas de Mpps (pacotes por segundo);
- O modelo chassi (expansão via módulos; emprega uma única fonte de alimentação para todos os módulos);
- Alta disponibilidade através de componentes redundantes e *hot-swappable*, tais como fontes de alimentação, ventiladores, relógios, *uplinks*, hardware de comutação (*switching fabric*), módulos de interfaces;
- Suporte à gerência via protocolo SNMP, podendo oferecer suporte às seguintes *Management Information Bases* (MIBs): MIB-II, Ethernet MIB, RMON MIB, High Capacity RMON, Switch MIB, SMON MIB;
- Outras opções de gerência tais como suporte a Switched Port Analyzer (SPAN);
- Servidor Web embutido para facilitar a gerência através de um browser.

4.3.6 Conclusão

Comutadores estão entre os equipamentos de interconexão mais utilizados para projetar uma rede de campus moderna. Seu alto desempenho e baixo custo fazem com que tenham superado completamente os repetidores, mesmo na camada de acesso de uma rede de campus.

4.4 Roteadores

Os roteadores são o último tipo de equipamento que discutiremos. Já que roteadores podem ser bastante complexos, só resumiremos os pontos principais nesta seção. O leitor poderá consultar a referência [Cisco-Internetworking] para obter detalhes adicionais. Além do mais, como o próprio nome diz, um roteador se ocupa basicamente da operação de *roteamento*. Detalhes sobre o roteamento no mundo TCP/IP podem ser obtidos no Apêndice 8.

4.4.1 Por que um outro tipo de equipamento?

Iniciamos a discussão da mesma forma que fizemos na discussão de comutadores: com uma pergunta: que tal usar apenas comutadores para fazer redes? Eles têm a vantagem da simplicidade (são transparentes) e da velocidade. Seria tão bom se pudéssemos usá-los exclusivamente para montar uma rede. Não funciona por vários motivos:

- **Falta de escala**. Os comutadores Ethernet discutidos na última seção utilizam dois mecanismos que não possuem escala. O primeiro é o uso de inundações para encaminhar quadros. Uma inundação vai para *todos* os hospedeiros. Seria impossível usar essa técnica para atingir um hospedeiro na China[9], digamos. Os canais de comunicação seriam completamente engargalados com excesso de tráfego. A regra básica é: manter inundações em domínios restritos. O segundo mecanismo utilizado pelos comutadores que não se adapta bem ao crescimento da rede é o tratamento de quadros de difusão e de multicast. Lembre-se de que o comutador encaminha tais quadros em todas as portas. Mais uma vez, temos um mecanismo com pouca escalabilidade: é impossível fazer difusão para *todos* os hospedeiros de uma rede. Alguém tem de parar a difusão para que ela não chegue na China. Aliás, dos dois mecanismos mencionados, a difusão é ainda pior, já que ela entregará um quadro a cada um dos hospedeiros da rede. A inundação faz com que o quadro seja visto por todas as placas de rede dos hospedeiros, mas ele só será aceito pela placa de rede endereçada.
- **Falta de interfaces**. Embora não tenhamos mencionado muitos detalhes sobre as interfaces de rede presentes em comutadores, normalmente eles não

[9] Se algum leitor estiver lendo este livro na China, substitua o país por "Brasil". ☺

possuem interfaces para tecnologias de longo alcance, tais como T1/E1, E3/T3, OC-12/ATM, OC-3/ATM etc.

- **Falta de controle**. O comutador não permite estabelecer controle administrativo sobre qual tráfego pode ser encaminhado para onde.[10] Nas empresas, é importante ter controle sobre o encaminhamento da informação, tanto para dizer qual tráfego vai onde, como para dizer qual tráfego deve simplesmente ser rejeitado.

Precisamos, portanto, de um novo tipo de equipamento de interconexão que não sofra dos problemas mencionados e que permita fazer redes maiores, abrangendo muitos milhares de hospedeiros. As redes assim projetadas poderão ser de campus ou de longo alcance (WAN). Tais equipamentos deverão obedecer aos seguintes requisitos básicos:

- O equipamento não pode usar a técnica de inundação ou outra técnica que não possua escala na sua operação;
- O equipamento não deve propagar quadros de difusão de camada de enlace, isto é, quadros enviados para o endereço MAC de difusão (FF-FF-FF-FF-FF-FF);
- O equipamento pode disponibilizar interfaces que permitam montar redes de longo alcance;
- O equipamento poderá prover serviços que permitam estabelecer controle administrativo sobre o encaminhamento de tráfego;
- Embora não possa ser chamado de "requisito", aceitamos que o equipamento *não* seja transparente e que seja endereçado explicitamente pelos outros equipamentos da rede.

Este equipamento chama-se *roteador* (ver Figura A4-12) e descrevemos seu princípio de operação na próxima seção.

FIGURA A4-12: O roteador modelo 10000 da Cisco.

[10] Lembre que estamos falando de comutador de nível 2. Comutadores de níveis 3 e superior são essencialmente roteadores e serão discutidos adiante.

4.4.2 Princípios de operação

Em dois aspectos fundamentais, o roteador é semelhante ao comutador: ele é um equipamento do tipo armazenamento-e-reenvio e ele possui uma tabela de encaminhamento – chamada de "tabela de roteamento". As grandes diferenças são: a) o que o roteador faz quando não acha o destino na tabela; e b) como a tabela é preenchida. Vejamos alguns detalhes agora.

Antes de explicar os detalhes sobre a operação de roteamento, é importante entender que *uma nova camada está entrando em ação*. O repetidor atua na camada física e o comutador atua na camada de enlaces. Para poder eliminar os defeitos de escala dos comutadores, o roteador faz a informação "subir" mais uma camada: a camada de rede. É na camada de rede que se localiza a tabela de roteamento e onde ocorre a decisão de encaminhamento tomada pelo roteador. A existência dessa camada faz com que o roteador possa parar (ou delimitar) as inundações e os quadros de difusão que causaram problemas de escala nos comutadores.

Um segundo ponto importante a ser destacado é que, diferentemente do repetidor e do comutador, *o roteador não é um equipamento transparente*. Se um hospedeiro A quiser se comunicar com um hospedeiro B, e se a conexão entre os dois passar por um roteador R, então o hospedeiro A deverá encaminhar a informação endereçada para R, pedindo que este a faça chegar a B. O roteador R, por sua vez, poderá encaminhar a informação para um segundo roteador S, endereçando-o explicitamente. Eventualmente, um último roteador poderá encaminhar a informação ao hospedeiro B diretamente. Vê-se, portanto, que o hospedeiro A sabe da existência do roteador R e é por esse motivo que dizemos que R não é transparente.

Uma das conseqüências da não transparência do roteador é que este deve ser endereçado pelos hospedeiros (na realidade, todos os hospedeiros e roteadores devem ser endereçados). Que esquema de endereçamento deve ser usado? Uma possibilidade seria usar o endereço MAC para esse fim. A idéia não funciona por dois motivos:

- Nem toda tecnologia empregada na camada de enlace possui endereçamento. Endereços MAC existem para Ethernet, por exemplo, mas não para enlaces E1;
- Os endereços MAC formam um espaço de endereçamento que chamamos de *plano*, o que significa que não há hierarquia útil nos endereços. Porém, para fazer roteamento baseado em endereços, é extremamente útil ter um esquema de endereçamento hierárquico. Por exemplo, sabemos que os números de telefone +55-83-333-1404 e +55-83-333-1405 devem estar no mesmo bairro da mesma cidade, o que facilita o roteamento. Diz-se que números de telefone formam um espaço hierárquico. Por outro lado, endereços MAC não formam um espaço hierárquico. O endereço 12-34-56-78-90-AB pode pertencer a uma placa de rede no Brasil e o próximo endereço – 12-34-56-78-90-AC – poderia estar na China. Rotear assim seria muito difícil.

Um novo espaço de endereçamento, específico para a camada de rede, é necessário. Por exemplo, no mundo TCP/IP, o "endereço IP" é usado para esse fim. Um exemplo de um endereço IP é 192.168.0.1. Detalhes sobre o endereçamento IP podem ser vistos no Apêndice 7.

Podemos agora descrever brevemente o algoritmo de roteamento. Na camada de rede, cada hospedeiro e roteador possui uma tabela de roteamento que indica o próximo endereço no caminho para cada destino possível. O hospedeiro A deseja enviar um pacote para o hospedeiro B. O hospedeiro A consulta sua tabela de roteamento e localiza o endereço B, ou pelo menos a região que deve conter o endereço B. A tabela de roteamento indicará o endereço R do roteador ao qual se deve encaminhar o pacote. O hospedeiro A deve necessariamente poder enviar um quadro para o roteador R usando apenas a camada de enlace; isto é, A e R devem pertencer à mesma rede física. Ao receber o quadro, o roteador R faz "subir" seu conteúdo para a camada de rede onde a decisão de roteamento (como alcançar "B"?) é tomada; como resultado, o pacote seguirá, de roteador em roteador, até chegar ao endereço destino, B.

Resta saber como a tabela de roteamento é preenchida. Os truques usados pelos comutadores não funcionam aqui (por não possuírem escala). O preenchimento pode ser manual ou pode ser baseado em protocolos especiais de roteamento. Esses últimos são protocolos que roteadores usam para conversar entre si, trocando informação de roteamento para achar as "melhores" rotas. Detalhes podem ser vistos no Apêndice 8 e em [Comer].

4.4.3 Comutadores de nível 3

Pensemos um pouco sobre a forma de montar uma rede de campus. Redes de campus precisam de comutadores para interligar os equipamentos. Se a rede for pequena, ela é tipicamente montada usando exclusivamente comutadores tais como os vimos numa seção anterior. Por outro lado, redes de campus maiores (com várias centenas ou milhares de hospedeiros) não podem ser construídas apenas com os comutadores que vimos, pois todos os equipamentos participariam de um mesmo domínio de difusão, afetando em demasia o desempenho de todos os equipamentos da rede. Portanto, uma rede de campus grande precisa utilizar roteadores para delimitar os domínios de difusão. Observe que tais roteadores tipicamente não possuiriam interfaces para tecnologias de longo alcance, já que são usados para delimitar o tráfego de difusão e não para alcançar lugares distantes. Talvez seja melhor dizer que precisamos da *função de roteamento*.

Ao longo dos anos 1990, comutadores foram evoluindo até integrarem a função de roteamento também. Embora tais equipamentos sejam roteadores, é mais freqüente chamá-los de *comutadores de nível 3*. Por que diferenciá-los assim? Primeiro, esses equipamentos normalmente não possuem interfaces para tecnologias de longo alcance: são feitos para rotear numa rede de campus. Em segundo lugar, eles continuam sendo comutadores ao mesmo tempo em que são roteadores. Como assim? Vamos explicar: o comutador de nível 3 permite agrupar portas em *redes virtu-*

ais (ou VLANs).[11] Cada rede virtual é um domínio de difusão. Portanto, podemos considerar uma VLAN como sendo um domínio de difusão configurável. É, portanto, possível limitar os domínios de difusão para que não alcancem todos os hospedeiros presentes na rede, mas apenas aqueles presentes na VLAN. Isso é muito bonito e útil, mas surge a questão: como enviar informação entre hospedeiros de VLANs *diferentes*? A resposta é utilizar roteamento; o comutador, portanto, deve também servir de roteador, agindo na camada 3, para passar informação de uma VLAN para outra. Detalhes adicionais sobre VLANs e comutadores de nível 3 podem ser vistos em [Cisco-Internetworking] e no Apêndice 10.

Para deixar as coisas um pouco mais complexas, devemos mencionar que comutadores evoluíram ainda mais e, hoje, permitem fazer seu trabalho de encaminhamento não apenas baseados no endereço MAC (comutador de nível 2) ou no endereço de rede (comutador de nível 3), mas também no endereço de transporte (comutador de nível 4) e no endereço de aplicação (comutador de nível 7). A utilidade destes últimos é basicamente a de realizar balanceamento de carga em redes muito carregadas.

4.4.4 Características especiais de roteadores

Roteadores são equipamentos extremamente variáveis se considerarmos a funcionalidade adicional que pode ser agregada ao equipamento. Mencionaremos apenas algumas aqui:

- Roteadores possuem freqüentemente uma grande variedade de interfaces LAN (rede local) e WAN (para longo alcance);
- Roteadores, normalmente, dão suporte a uma variedade de protocolos de roteamento, tais como *Border Gateway Protocol* (BGP), *Intermediate System-to-Intermediate System* (IS-IS), *Open Shortest Path First* (OSPF) etc.;
- Roteadores, normalmente, dão suporte a diversos protocolos de multicast, tais como PIM, IGMP, CGMP, DVMRP etc.;
- Roteadores são freqüentemente multiprotocolo (conseguem rotear tráfego de vários protocolos de rede, tais como IP, IPX etc.);
- Roteadores freqüentemente incluem um filtro de pacotes (ou até um firewall completo), permitindo estabelecer regras de segurança sobre que tipo de tráfego pode ser roteado pelo equipamento;
- Devido a sua criticalidade, roteadores freqüentemente possuem recursos especiais para aumentar a disponibilidade da rede ou melhorar a manutenibilidade. Exemplos de tais recursos incluem suporte ao *Hot Standby Routing Pro-*

[11] Muitos comutadores permitem criar VLANs por vários critérios e não apenas por portas.

tocol (HSRP), módulos redundantes (módulos de processamento, fontes de alimentação, ventiladores etc.) e *hot swappable*;

- Roteadores podem ter recursos especiais para priorizar o tráfego roteado ou, de forma geral, dividir os recursos disponíveis, principalmente a banda passante dos enlaces, entre várias classes de tráfego. Podemos chamar esses recursos de "Orientados à Qualidade de Serviço". Esses recursos incluem *Weighted Fair Queuing* (WFQ) e roteamento baseado em políticas administrativas (*Policy Based Routing*);
- Suporte a uma vasta gama de recursos e protocolos adicionais tais como: MPLS, VPN, tunelamento GRE;
- Suporte à gerência via protocolo SNMP.

4.4.5 Conclusão

A característica-chave que permite que roteadores tenham escala, podendo ser utilizados para montar redes muito grandes é a *não transparência*. Portanto, qualquer rede de certo tamanho empregará roteadores, freqüentemente muitos deles. Roteadores são utilizados em redes corporativas onde:

- Há necessidade de acessar um local remoto usando um enlace de longa distância
- Há necessidade de conter quadros de difusão; isso ocorre normalmente na camada distribuição de uma rede de campus;
- Há necessidade de policiar tráfego, tipicamente na camada de distribuição de uma rede de campus ou no ponto em que a rede corporativa se conecta a redes externas.

4.5 Bibliografia

[GUIA-ETHERNET]	Spurgeon, C. E. Ethernet – *O Guia Definitivo*. Editora Campus, 2000.
[CISCO-INTERNETWORKING]	Internetworking Technologies Handbook, 3ª edição, Cisco Press, 2000.
[COMER]	Comer, Douglas. *Internetworking with TCP/IP: Principles, Protocols, and Architectures*. Prentice Hall, 4ª edição.

APÊNDICE

5

5 Documentação de rede

5.1 Introdução

Um ponto muito importante e freqüentemente relegado a segundo plano, é a documentação de uma rede de comunicação.

Particularmente em redes mais antigas, não projetadas de acordo com as normas de cabeamento estruturado, nem sempre é fácil encontrar informações em quantidade e qualidade suficientes para que se possa substituir o administrador da rede sem sentir um frio na espinha.

Sim, isso seria o ideal. Poder substituir seu administrador de rede, sem ter de passar semanas – ou meses – com medo de que a rede possa entrar em colapso e parar de funcionar de uma hora para outra.

Com a aderência cada vez maior às normas do cabeamento estruturado, esse problema tende a diminuir, na medida em que a própria normatização prescreve um padrão de documentação de rede – ao menos para o cabeamento.

Mas somente a documentação do cabeamento não é suficiente. O que dizer dos usuários da rede? Quantos são? O que esperam da rede? Que serviços são oferecidos? Quais serviços devem estar disponíveis 24 horas por dia, 7 dias na semana? Qual o desempenho que se espera desses serviços?

Essas, e tantas outras perguntas, deveriam ser respondidas por uma boa documentação de rede. Obviamente, não é simples elaborar e, principalmente, manter atualizada tal documentação.

Diversas proposições existem para tentar definir o que, onde e como deve-se documentar em relação à uma rede.

Esse pequeno apêndice é mais uma proposição. Ele não pretende ser completo e nem conclusivo. É uma composição de diversas proposições já analisadas e/ou usadas pelos autores, particularmente a proposição mostrada em [OPPENHEIMER] com um pouco da experiência adquirida pelos mesmos ao longo dos anos.

5.2 Roteiro para documentação da rede

O roteiro proposto aqui objetiva tornar mais metódico o procedimento de documentação de uma rede. Não necessariamente todos os seus passos precisam ser seguidos – embora isso seja o recomendado.

O importante é que exista algum tipo de documentação da rede, de modo que o administrador e seus auxiliarem possam recuperar a funcionalidade da rede no menor tempo possível quando da ocorrência de algum problema na mesma.

Em outras palavras, além, obviamente, das ferramentas de monitoração e gerência, uma boa documentação é fator fundamental para a minimização de *downtime* da rede.

5.2.1 Identificação das necessidades e objetivos do cliente da rede

Como início de uma documentação de rede, é importante ter uma descrição do que o cliente necessita e quais são seus objetivos. A palavra cliente aqui é usada no sentido mais amplo, identificando a própria corporação.

O ideal é que a rede seja encarada como uma ferramenta para ajudar a corporação a atingir seus objetivos de negócio e, para tanto, a corporação espera que a rede atenda bem suas necessidades.

É importante destacar os objetivos e restrições do negócio, os objetivos e restrições técnicos, e caracterizar o tráfego projetado para a rede, incluindo principais fluxos (de onde vem e para onde vão os fluxos de dados – aplicações da intranet, aplicações da extranet, uso da Internet, tráfego entre matriz e filiais), carga – agregação de fluxos e requisitos de QoS (*Quality of Service*).

Requisitos técnicos tais como escalabilidade, disponibilidade, desempenho, segurança, gerenciabilidade, usabilidade, adaptabilidade e custo-benefício, devem ser descritos.

O escopo da rede, ou seja, sua abrangência – física e de serviços – dentro da corporação deve ser bem definido. O ideal seria destacar bem os locais e serviços que devem e não devem ser atendidos e oferecidos pela rede.

Restrições arquiteturais e ambientais que podem afetar a implementação e/ou expansão da rede devem ser descritas.

Além disso, devemos descrever a comunidade de usuários – possivelmente dividida em classes – com suas necessidades de serviços, e as aplicações, com seus atributos e necessidades específicas.

Requisitos de treinamento e de suporte devem estar definidos.

5.2.2 Projeto lógico

No projeto lógico busca-se documentar a organização lógica da rede. Por organização lógica costuma-se entender:

- A topologia lógica da rede;
- Uma descrição dos protocolos de nível 2 (comutação) e nível 3 (roteamento), incluindo qualquer recomendação sobre o uso desses protocolos;
- Um esquema de endereçamento e atribuição de nomes;
- Um esquema de roteamento;
- Os mecanismos e produtos recomendados para a segurança, incluindo um resumo de políticas de segurança e procedimentos associados (um plano completo de segurança pode ser incluído como apêndice);
- Recomendações sobre arquitetura e produtos para a gerência;
- Explicações sobre o porquê de várias decisões tomadas, relacionando as decisões aos objetivos do cliente.

É importante incluir esquemas e desenhos no projeto lógico que facilitem sua compreensão. Veremos na próxima subseção um exemplo completo para elucidar melhor as recomendações descritas.

5.2.2.1 Topologia Lógica

Consideremos uma rede com a topologia indicada na Figura A5-1.

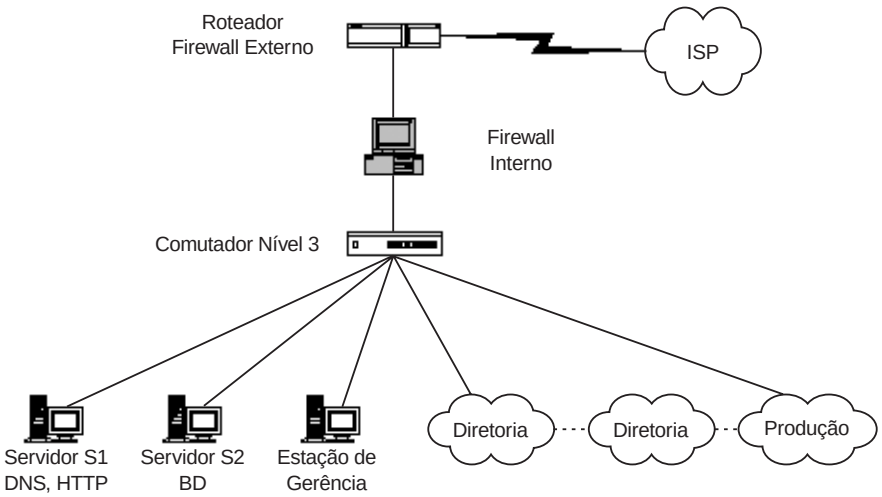


FIGURA A5-1: Topologia lógica.

Observe que a topologia lógica fornece uma visão geral da organização da rede sem, contudo, especificar qualquer informação relativa a cabeamento, tecnologias de transmissão usadas, disposição física de cabos e equipamentos etc.

5.2.2.2 Protocolos Níveis 2 e 3

Nessa proposição, procurou-se definir uma forma de interconexão dos diversos setores da empresa através de comutadores nível 2 e um comutador nível 3 (também conhecido como *switch-router*[1]), dotado de capacidade de roteamento e de filtragem de pacotes. Procura-se fornecer uma redundância de acesso de qualquer setor da empresa até os servidores por meio de comutação nível 2, com o uso do protocolo de Spanning Tree (SPT).[2]

A rede toda utilizará a arquitetura TCP/IP; o comutador nível 3 com capacidade de filtragem de pacotes permitirá a utilização de redes virtuais (*Virtual LANs* – VLANS) e a realização de um controle de tráfego mais apurado entre os diversos setores da empresa.

Como a implementação da capacidade de redundância é baseada em protocolo nível 2 (SPT), o roteamento utilizado pode ser o estático.

5.2.2.3 Esquema de Endereçamento e Atribuição de Nomes

Para essa rede, um esquema de endereçamento possível seria o indicado na Tabela A5-1.

TABELA A5-1: Esquema de endereçamento.

Local/Setor	Ender. Rede e Máscara	Faixa de Endereços	Ender. de Difusão
Zona Desmilitarizada	201.202.203.0 255.255.255.0	201.202.203.1 a 201.202.203.254	201.202.203.255
Servidores (VLAN 0)	192.168.0.0 255.255.255.0	192.168.0.1 a 192.168.0.254	192.168.0.255
Diretoria (VLAN 1)	192.168.1.0 255.255.255.0	192.168.1.1 a 192.168.1.254	192.168.1.255
Gerência (VLAN 2)	192.168.2.0 255.255.255.0	192.168.2.1 a 192.168.2.254	192.168.2.255
Produção (VLAN 3)	192.168.3.0 255.255.255.0	192.168.3.1 a 192.168.3.254	192.168.3.255

Para atribuir nomes aos elementos da rede, deve-se adotar alguma convenção simples que seja utilizada de forma consistente o tempo todo.

[1] Leia mais sobre equipamentos de rede no Apêndice 4.
[2] Leia mais sobre este protocolo no Apêndice 6.

Para servidores, por exemplo, pode-se usar Sv-X[-Y], onde X indica o serviço oferecido pelo servidor e [.Y] indica, opcionalmente, um número de índice. Exemplos: Sv-DNS-1, Sv-DNS-2, Sv-MAIL.

No caso de servidores que acumulam diversos serviços, pode-se adotar uma convenção mais simples com Sv-N, onde N indica um índice seqüencial. Exemplos: Sv-1, Sv-2.

Para estações clientes e as respectivas tomadas nas áreas de trabalho, pode-se usar ppSSS-ee-tt, onde pp indica o pavimento, SSS indica a sala, ee indica o espelho e tt indica a tomada no espelho. Exemplos: 01S10-01-02, máquina/tomada 02, do primeiro espelho da sala 10 do primeiro pavimento.

Para roteadores, pode-se usar Rt-N, onde N indica um índice seqüencial. Exemplos: Rt-1, Rt-2.

Para comutadores, pode-se usar CnX-Y-Z, onde X indica o nível do comutador – 2 ou 3, Y indica o pavimento e Z indica um índice seqüencial. Exemplos: Cn2-2-1, Cn2-2-2, Cn3-3-1.

Para Armário de Telecomunicação, Painel de Manobra (*patch-panel*) e tomada em Painel de Manobra, pode-se usar ppA-qq-tt, onde pp indica o pavimento, A indica um armário no pavimento, qq indica o painel de manobra numerado de cima para baixo no armário, iniciando em 01 e tt indica a tomada no painel de manobra. Exemplos: 03C, armário C do pavimento 03; 03C-02, painel de manobra 02, do armário 03C; 91D-04-02, tomada 02 do painel de manobra 91D-04.

Cabos de conexão cruzada e de área de trabalho devem ser identificados em ambas as pontas com números inteiros sequenciais.

5.2.2.4 Esquema de Roteamento

Considerando o roteamento estático e o uso de redes virtuais, o esquema de roteamento a ser usado torna-se simples. Basicamente:

- Cada servidor tem como rota default o Comutador nível 3, na sua respectiva VLAN;
- Cada cliente da Diretoria, Gerência e Produção tem como rota default o Comutador nível 3, na sua respectiva VLAN;
- O comutador nível 3 tem como rota default o Firewall Interno;
- O firewall Interno tem como rota default o Roteador/Firewall Externo;
- O firewall Interno deve estar configurado para realizar NAT (*Network Address Translation*) ou ser um servidor Proxy.

5.2.2.5 Mecanismos e Produtos de Segurança

Como norma de segurança básica, adotou-se o uso de dois níveis de Firewall (Externo e Interno), com implementação de endereçamento privativo para a rede

interna. O Firewall Interno deve ser, idealmente, configurado para realização de serviço Proxy, sobre o qual serão definidos os serviços que podem ou não ser utilizados pelos usuários da rede.

Os procedimentos padrão de identificação e autenticação de usuários, bem como a concessão de direitos de utilização de serviços para cada usuário (ou classe de usuários), devem ser descritos no documento de "Política de Segurança" elaborado pela empresa.

5.2.2.6 Recomendações sobre Arquitetura e Produtos de Gerência

Considerando a necessidade de se manter a rede operando em sua capacidade plena, com a maior qualidade de serviço possível, embora a mesma não seja uma rede de grande porte, adota-se uma arquitetura de gerência de rede centralizada, padrão SNMP (*Simple Network Management Protocol*), com a adoção da ferramenta de gerência de rede XYZ, em sua versão KLM, desenvolvida pela empresa RST.

Tal ferramenta executa sobre uma plataforma baseada em microcomputador compatível com IBM PC, com memória mínima de 256 MBytes e área de armazenamento em disco mínimo de 40 Gbytes.

5.2.3 Projeto físico

No projeto físico costuma-se documentar a organização física da rede. Por organização física costuma-se entender:

- A topologia física da rede, destacando pontos de interconexão, centros de fiação etc.;
- A especificação das tecnologias de cabeamento e de transmissão utilizadas, com justificativas para cada escolha;
- A especificação dos equipamentos utilizados – máquinas clientes, máquinas servidoras, máquinas de armazenamento de dados (*data stores*), máquinas de backup (*backup*), dispositivos de interconexão (concentradores, comutadores, roteadores etc.) – com justificativas para cada escolha;
- A escolha do provedor de acesso à Internet e a forma de conexão ao mesmo;
- Os custos de manutenção mensal (ou anual) de equipamentos e serviços.

Vejamos um exemplo completo para elucidar melhor as recomendações acima.

5.2.3.1 Topologia Física

A rede está implantada em um prédio de 3 pavimentos, com a seguinte ocupação:

Pavimento	Ocupação	Quantidade de Pontos
3	Diretoria	16
2	Gerência	20
1	Produção	20

Cada pavimento dispõe de um armário de telecomunicação que funciona como centro de fiação para o pavimento e local de instalação dos equipamentos de interconexão da rede.

No pavimento 3/Diretoria, localiza-se a sala de equipamentos que contém os servidores da rede.

A Figura A5-2 mostra a topologia física da rede.

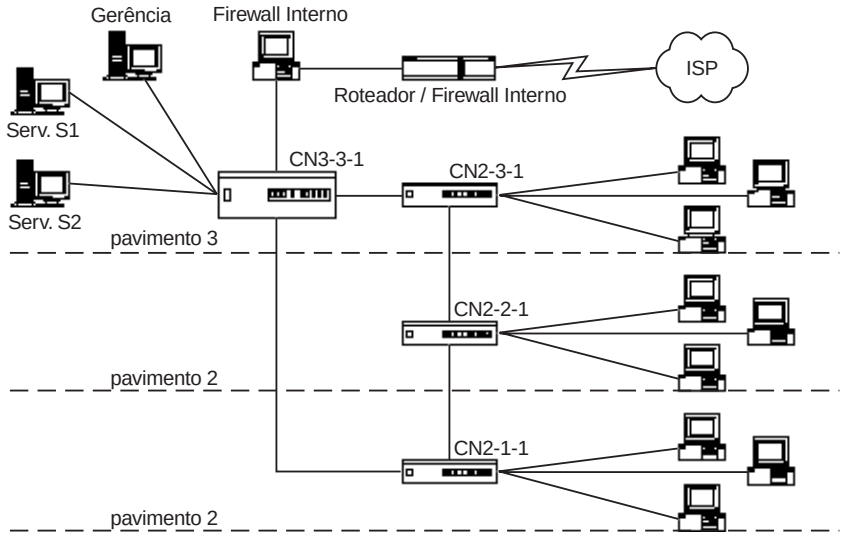


FIGURA A5-2: Topologia física.

5.2.3.2 Tecnologias de Cabeamento e Transmissão

Considerando a adoção das normas de cabeamento estruturado e a abrangência da rede para sua implantação nos 3 pavimentos, adotou-se um cabeamento vertical em fibra ótica monomodo, com conectorização padrão SC para suporte de tráfego a 1000 Mbits por segundo (Mbps).

Para o cabeamento horizontal, adotou-se cabo de pares trançados, categoria 5e, com conectorização padrão RJ45/568A, com suporte para tráfego a 100 Mbps.

Para a conexão ao ISP, usamos o serviço Frame-Relay oferecido pela operadora de telefonia local, com velocidade nominal de 512 Mbps e CIR (*Commited Insertion Rate*) de 256 Mbps, que apresenta a melhor relação custo/benefício para a empresa.

5.2.3.3 Especificação de Equipamentos Utilizados

Os equipamentos seguintes são utilizados na rede. Em cada caso, são indicados marca e modelo do produto, versão do hardware e software, características básicas, nome e endereço da assistência técnica, identificação na rede (de acordo com as regras de nomes adotada).

Equipamento	Hardware/ Software	Características Básicas	SNMP	RMON	Assistência Técnica	ID. na Rede
Roteador XYZ 9104	4.1/3.3	2 WAN E1 4 LAN 10TX	x	x	XYZ	Rt-1 ou Fw-ext
Firewall Interno Micro P4, 256 MB RAM, 80 GB HD, 2 Eth 10/100 TX	---/ Linux 2.2.24		x		ACME	Fw-int
Comutador N3 XYZ 9274	3.3/2.4	8 LAN 1000	x	x	XYZ	Cn3-3-1
Comutador N2 XYZ	4.3/1.2	2 LAN 1000 16 LAN 10/100	x	x	XYZ	Cn2-1-1 Cn2-2-1 Cn2-3-1
Servidor Micro P4, 512 MB RAM, 160 GB HD, 1 Eth 10/100 TX	---/ Linux 2.2.4		x		ACME	Sv-1
Servidor Micro P4, 512 MB RAM, 160 GB HD, 1 Eth 10/100 TX	---/ Linux 2.2.4		x		ACME	Sv-2
Estação de Gerência Micro P4, 256 MB RAM, 80 GB HD, 1 Eth 10/100 TX	---/ Linux 2.2.4		x		ACME	Sv-3 ou Gerencia
Cliente Micro P4, 256 MB RAM, 40 GB HD, 1 Eth 10/100 TX	---/ Linux 2.2.4		x		ACME	Padrão da empresa

XYZ
Praça da Paz, 232 – Centro
Fone/Fax: (83) 123.4567
Contato: Sr. Pedro

ACME
Av. Mar de Rosas, 485 – Centro
Fone/Fax: (83) 765.4321
Contado: Elza

5.2.3.4 Provedor de Acesso à Internet

Informações sobre os provedores de acesso à Internet utilizados devem também ser documentadas. Para este exemplo, poderíamos documentar que a empresa Sdrufs&Sdrufs está sendo usada como provedor de acesso à Internet, com um enlace com tecnologia Frame-relay em 512/256 Kbps.

5.2.3.5 Custo de manutenção

Os custos de manutenção mensal da rede são divididos em custos de provimento de serviço Internet, custos de contrato de manutenção preventiva e custeio geral (pequenas despesas diversas), de acordo com a Tabela 5.2 abaixo.

TABELA A5-2: Custo de Manutenção

Item	Custo mensal (R$)
Serviço Internet	1.500,00
Manutenção preventiva	10.000,00
Total	11.500,00

5.2.4 Configuração de Equipamentos

Uma parte importante da gerência de uma rede é a configuração dos equipamentos que a compõe.

Com ou sem o uso de uma ferramenta de gerência, é de suma importância que haja documentação sobre a configuração dos equipamentos da rede.

Essa documentação pode ser feita na forma de roteiros impressos (menos indicado) ou de procedimentos parametrizáveis que possam se executados a partir de uma estação de trabalho.

Para diversos equipamentos de interconexão, existem programas de configuração que permitem o salvamento da configuração de um equipamento em um arquivo em disco para posterior reconfiguração do equipamento no caso de problemas de perda de configuração.

Servidores e estações clientes devem ter seus procedimentos de instalação e configuração bem definidos e impressos, para que possam ser localizados e usados rapidamente.

5.2.5 Documentação adicional

Além dos itens já elencados, é importantes anexar à documentação da rede os seguintes elementos:

- Planta baixa de infra-estrutura, indicando as dimensões de tubulação e/ou eletrocalhas utilizadas;
- Planta baixa com o encaminhamento dos cabos, indicando o número de cabos UTP e/ou fibra por segmento da tubulação;
- Relatório dos testes de certificação de todos os pontos instalados;
- Relatório de testes dos segmentos de fibra óptica;
- Layout dos Armários de Telecomunicações;
- Mapa de interconexão dos componentes ativos e passivos, isto é, lista de todas as tomadas RJ45 de cada painel de conexão e das portas dos equipamentos;
- Termos de garantia dos elementos ativos e passivos da rede.

As plantas baixa dos prédios com o projeto de rede, deverão ser fornecidas, idealmente, em formato apropriado (por exemplo, AUTOCAD), obedecendo às seguintes convenções:

- Nível 0 – edificação e arquitetura com legenda, contendo escala do desenho, identificação da unidade, nome do prédio, pavimento, nome do projetista e data de execução;
- Nível 1 – tubulação preexistente e a construída;
- Nível 2 – cabeamento UTP;
- Nível 3 – cabeamento óptico;
- Nível 4 – componentes ativos tais como estações de trabalho, estações servidoras, concentradores (*hubs*), comutadores (*switchs*), roteadores etc.;
- Nível 5 – componentes passivos, armários de telecomunicação, painéis de manobra e pontos de telecomunicações;
- Nível 6 – identificação de salas e observações;
- Nível 7 – móveis ou outros objetos.

Os termos de garantia obtidos ao final da implantação de uma rede, devem descrever claramente os limites e a duração da garantia para cada componente do sistema instalado. Mesmo que o prestador de serviço tenha contratado outros terceiros, a garantia final será dada e mantida pelo prestador. Os requisitos mínimos de garantia recomendados para cada componente são:

- Equipamentos ativos: 1 ano após a instalação (idealmente 3 anos para equipamentos de interconexão);
- Cabos e componentes acessórios: 5 anos contra defeitos de fabricação;
- Infra-estrutura: 3 anos contra ferrugem e garantia de resistência mecânica;
- Funcionalidade e desempenho: 5 anos.

Declaração de desempenho assegurado para as aplicações para as quais a rede física foi proposta, com indicação de possíveis restrições para outras aplicações ou para as aplicações introduzidas no futuro pelos principais organismos internacionais (IEEE, TIA/EIA, ISO/IEC, ATM FORUM etc.), também deve ser fornecida.

5.3 Referências bibliográficas

[OPPENHEIMER]	Oppenheimer, P. *Top-Down Network Design*. Cisco Press. 1998.

5.3.1 Leitura recomendada

Maggiora, P. L. D., Elliot, C. E., Pavone Jr, R. L., Phelps, K. J., Thompson, J. M. *Performance and Fault Management*. Cisco Press. 2000.

APÊNDICE

6

6 Conhecendo o protocolo árvore de cobertura

6.1 Introdução

Para que possamos entender a importância do Protocolo Árvore de Cobertura (*Spanning Tree Protocol*), precisamos entender como os comutadores funcionam. Comutadores utilizam um mecanismo chamado *backward learning* (aprendizagem pela origem) para descobrir para que porta enviar um quadro. Esse mecanismo funciona basicamente da seguinte forma: sempre que o comutador recebe um quadro por uma determinada porta, ele verifica o endereço MAC da origem do quadro e aprende que dados destinados a essa máquina devem ser enviados por essa porta. Assim, o comutador sabe através de que portas as máquinas que já "falaram" na rede são alcançáveis. Quando um comutador não sabe para que porta transmitir um quadro, ele o envia para todas as portas, exceto para aquela através da qual o quadro foi recebido (inundação). Veja mais detalhes sobre o funcionamento dos comutadores no Apêndice 4.

Em muitas organizações, para aumentar a confiabilidade da rede, comutadores são ligados através de enlaces redundantes, formando um laço. A Figura A6-1 apresenta uma topologia com um laço entre comutadores.

FIGURA A6-1: Topologia com laço entre comutadores.

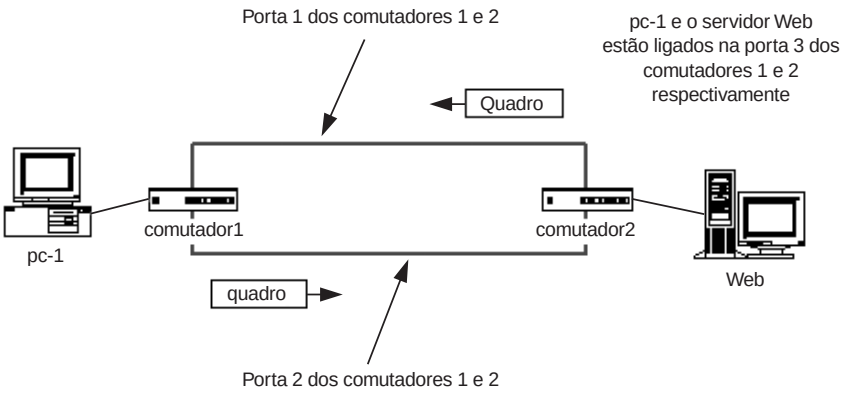


FIGURA A6-2: Duplicação de quadros entre comutadores em paralelo.

Quando temos uma topologia com laço entre comutadores, é imprescindível que o Protocolo Árvore de Cobertura (PAC) seja habilitado em todos os comutadores que participam do laço. Levando em consideração a rede da Figura A6-1, vejamos o que acontece se o PAC não estiver habilitado quando a máquina pc-1 deseja enviar um quadro o servidor Web:

O comutador 1 receberá um quadro (inicialmente chamado quadro A). Ele lê o endereço destino do quadro e sabe que deve enviá-lo através de sua porta 2 (conectada ao comutador 2).

↓

O quadro é recebido pelo comutador2. Nesse momento, comutador2 descobre por que porta enviar quadros destinados a pc-1. Suponha que o comutador2 não sabe por qual de suas portas ele deve enviar o quadro recebido de forma que ele chegue no destino. Então, o comutador2 envia o quadro para todas as suas portas, exceto para a porta através da qual ele chegou. Veja Figura A6-2.

↓

O servidor Web, ligado a comutador2, já recebeu o quadro. No entanto, o comutador1 – o primeiro comutador a enviar o quadro em questão – o receberá novamente, e dessa vez, através de sua porta 1 (que também está conectada a comutador2).

↓

Ao receber o novo quadro (já duplicado – quadro2 da Figura A6-2), comutador1 modificará sua tabela de endereços, pois agora ele pensa que pc-1 está conectada à porta 1 e não mais à porta 3. Comutador1 propagará o quadro recebido para todas as suas portas, exceto a porta através do qual o quadro foi recebido. Quando o quadro duplicado chegar novamente ao comutador2, o mesmo ocorrerá. O servidor Web receberá várias cópias idênticas do mesmo quadro e as tabelas de endereçamento dos comutadores ficarão instáveis.

Todos os quadros transmitidos pelos comutadores 1 e 2 são idênticos ao quadro originalmente enviado por pc-1. Esse processo será repetido indefinidamente, gerando quadros idênticos e propagando-os, até que a largura de banda fique saturada, tornando impossível a comunicação.

6.2 Como o PAC funciona?

A solução para o problema apresentado na seção anterior é simples. Os comutadores devem se comunicar entre si e concordar em desabilitar certos enlaces redundantes, de forma que os enlaces ativos que os interconectam representem uma topologia em árvore, onde laços não existem. Os comutadores realizam suas tarefas como se os enlaces de redundância não existissem, sendo, portanto, impossível a ocorrência de transmissão infinita de quadros entre comutadores. Se um enlace que estava ativo se tornar indisponível, o PAC reconfigura a rede reativando enlaces antes bloqueados.

Os comutadores nos quais o PAC está habilitado realizam a tarefa de descobrir uma árvore de cobertura entre eles e bloquear portas de redundância. Chamamos a árvore final de árvore de cobertura, pois a partir de qualquer comutador da rede, quaisquer outros comutadores que antes eram acessíveis, continuarão sendo após a desabilitação de algumas portas, porém sem a presença de laços. Em outras palavras, se antes do PAC ser habilitado, o comutador A podia alcançar os comutadores B, C e D, após a habilitação do PAC o comutador A continua podendo se comunicar com os mesmos comutadores B, C e D.

Todos os comutadores onde o PAC está habilitado se comunicam através de mensagens que chamamos *bridge protocol data units* (BPDUs). Falaremos mais sobre as BPDUs na Seção 3. Por enquanto, basta saber que elas existem e é através delas que os comutadores se comunicam.

Para construir a árvore de cobertura, a primeira tarefa a ser realizada é eleger um comutador para ser a raiz da árvore de cobertura. O objetivo do protocolo árvore de cobertura é detectar o menor caminho entre esse comutador raiz e os demais comutadores da rede. Os caminhos alternativos de maior custo são bloqueados. Dessa forma, não existirão laços na rede. É interessante que o comutador raiz seja estável, de preferência veloz e o mais central possível, uma vez que grande parte do tráfego entre os demais comutadores passará por ele.

Cada comutador tem um identificador único utilizado para a eleição da raiz. Uma porção desse identificador é fixa, a outra porção pode ser configurada pelo administrador da rede. A porção fixa é, em geral, o menor endereço MAC de todas as portas do comutador. A porção configurável é uma prioridade, quanto menor o número, maior a prioridade do comutador.

Cada porta dos comutadores também tem um identificador único, que é, em geral, o endereço MAC da porta. Além disso, cada porta dos comutadores é associada a um custo de caminho, que representa o custo de se transmitir um quadro atra-

vés da porta em questão. Os custos de caminho das portas vêm configurados com um valor default, mas este pode ser mudado pelo administrador da rede.

O comutador com menor identificador é escolhido para ser a raiz. Se você deixar todos os comutadores com a mesma prioridade, o comutador que possuir uma porta com o menor endereço MAC dentre todos será a raiz.

Vejamos, através de um exemplo, como o PAC funciona. Considere a Figura A6-3. Cada comutador tem conexão direta com os demais. Como a árvore de cobertura entre os comutadores é formada e mantida?

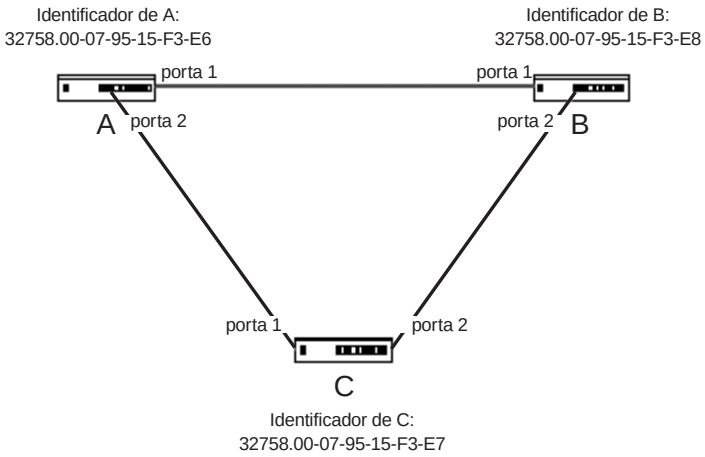


FIGURA A6-3: Comutadores A, B e C formam um laço.

Quando iniciados, todos os comutadores pensam que são a raiz da árvore de cobertura. Suponha que o comutador C foi o mais rápido e ele será o primeiro a enviar uma BPDU de configuração aos demais. Nessa mensagem, o identificador da raiz será o identificador de C: 32768.00-07-95-15-F3-E7. A e B recebem a mensagem e, ao processá-la, B percebe que não é a raiz, pois o identificador recebido é menor que seu identificador. B, então passa a achar que C é a raiz. O comutador A continua achando que ele é a raiz, pois seu identificador é menor que o identificador da raiz recebido de C. Nesse momento, B e C acham que C é a raiz enquanto comutador A acha que ele próprio é a raiz.

Suponha que agora comutador A vai enviar uma mensagem de configuração. Nesta mensagem, ele anunciará a si próprio como a raiz da árvore de cobertura. Quando B e C recebem a mensagem de A, eles percebem que o identificador de A é menor que o deles, e então todos os comutadores passam a concordar que A é a raiz da árvore de cobertura.

Chega então o momento de se escolherem as portas raízes. Cada um dos comutadores, exceto a raiz, deve selecionar uma porta raiz. A porta raiz de um comutador é a porta mais próxima do comutador raiz. Essa porta é escolhida com base no custo do caminho associado a cada porta, sendo considerado o custo cumulativo de todos

os enlaces intermediários para se chegar de uma porta à raiz. As portas 1 dos comutadores B e C são eleitas as portas raízes.

Para cada segmento entre dois comutadores, escolhe-se uma porta designada. As portas designadas de um segmento oferecem o menor custo de caminho entre a raiz e as máquinas deste segmento. Para a eleição das portas designadas de cada segmento, considere que todos os enlaces entre A, B e C têm a mesma velocidade e o mesmo custo, suponha, 19.

No segmento que liga os comutadores A e C a porta 2 do comutador A é escolhida, pois ela está diretamente conectada ao comutador raiz. Pelo mesmo motivo, a porta 1 do comutador A é escolhida como a porta raiz do segmento que liga os comutadores A e B. No segmento que liga B e C, o custo de se chegar à raiz é o mesmo para a porta 2 de B e para a porta 2 de C. O critério de desempate é o identificador dos comutadores envolvidos. Como o identificador de C é menor que o de B, a porta 2 de C é escolhida como a porta raiz para este segmento.

Todas as portas escolhidas como raízes ou como portas designadas ficam ativas, enquanto as demais são bloqueadas. A porta 2 do comutador B será, portanto, desativada para que laços lógicos não ocorram na rede. Veja o resultado na Figura A6-4. A animação de todo este processo pode ser encontrada em [CISCO-STP-TROUBLE].

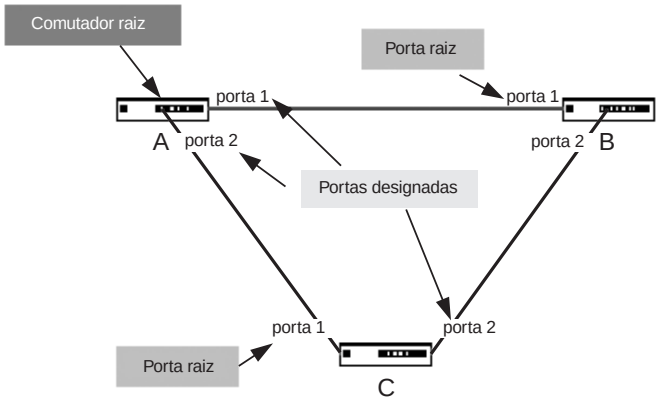


FIGURA A6-4: Formação da árvore de cobertura.

Depois da formação da árvore de cobertura, os comutadores continuam enviando BPDUs de configuração para os comutadores vizinhos em intervalos regulares. As BPDUs de configuração descem na árvore de cobertura a partir das portas designadas. As portas 1 e 2 do comutador A, por exemplo, enviam, de tempos em tempos, BPDUs de configuração, que são recebidos pelos comutadores B e C. Essas BPDUs de configuração contêm informações sobre a topologia da árvore de cobertura. Na seção seguinte veremos um pouco mais de detalhes sobre as BPDUs de configuração.

Se um comutador parar de receber BPDUs de configuração de um vizinho por um intervalo de tempo mais longo que o normal, ele inferirá que ocorreu uma mudança de topologia e a árvore de cobertura deverá ser recalculada.

6.3 Formato das BPDUs

Como já vimos, os comutadores calculam a árvore de cobertura através de troca de mensagens. Essas mensagens chamam-se BPDUs (*bridge protocol data units*). Existem dois tipos de BPDUs: BPDUs de configuração e BPDUs de mudança de topologia.

A estrutura de uma BPDU de configuração é mostrada na Figura A6-5. Alguns campos não precisam ser explicados, pois seus próprios nomes já descrevem muito bem as informações que carregam. Vamos nos concentrar um pouco mais em descrever os campos que definem valores de tempo, que são mais importantes para o contexto da gerência.

O campo **message age** especifica o tempo decorrido desde o envio da mensagem pelas portas designadas no comutador raiz que serviu de base para essa mensagem. Anteriormente, falamos que as mensagens de configuração descem na árvore de cobertura a partir da raiz. Quando os comutadores vizinhos da raiz recebem uma BPDU de configuração da raiz, eles o processam e enviam mensagens de configuração através de suas portas designadas para comutadores vizinhos. E assim, as mensagens de configuração se propagam na rede comutada. Cada comutador tem configurado um tempo para a idade máxima da mensagem. O valor desse temporizador também é indicado na BPDU de configuração no campo **max age**. Quando um comutador recebe uma BPDU com uma idade superior à idade configurada no comutador, a BPDU é descartada.

Em cada comutador, precisamos configurar 3 temporizadores: o **max age**, já citado, o **hello time** e o **forward delay**. O **hello time** indica de quanto em quanto tempo o comutador que está transmitindo a BPDU está configurado para enviar BPDUs de configuração quando ele for a raiz. O campo **hello time** traz o valor do temporizador **hello time** configurado na raiz atual da árvore de cobertura.

O temporizador **forward delay** diz respeito ao tempo que um comutador espera antes de desbloquear uma porta. Para que laços temporários não ocorram na rede, uma porta não passa do estado "bloqueada" para o estafo "ativa" diretamente. Primeiro, a porta fica num estágio de escuta durante o tempo estabelecido pelo temporizador **forward delay**. Caso a porta ainda deva continuar ativa, ela entra no estado de aprendizagem e fica nesse estado durante o tempo estabelecido pelo temporizador **forward delay**. Depois de toda essa espera, se a porta não foi bloqueada, ela entra no estado ativo. Na próxima seção, falamos um pouco sobre os estados das portas. Nas sugestões de tratamento do problema na Seção **Problema com Árvore de Cobertura**, você encontra uma tabela com os valores indicados para cada um desses temporizadores.

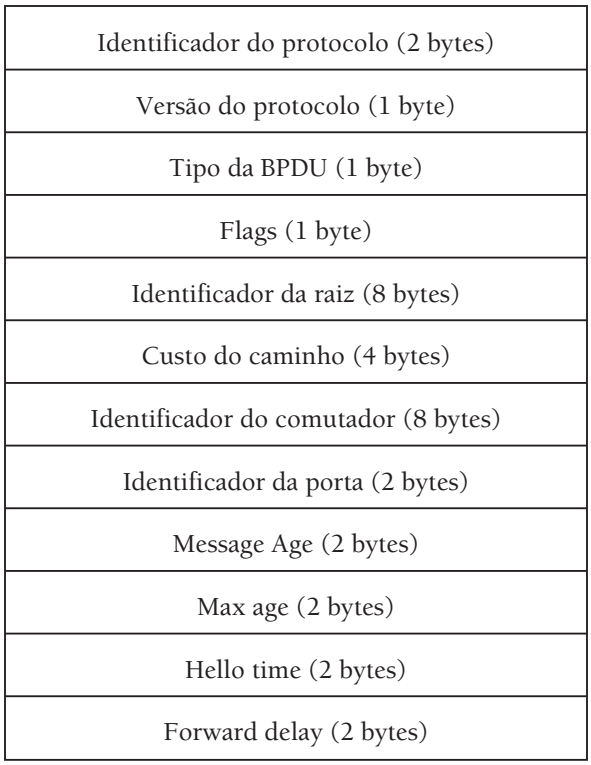


FIGURA A6-5: Estrutura de uma BPDU de configuração.

Normalmente, um comutador recebe BPDUs de configuração de um outro comutador que está mais próximo da raiz que ele, isto é, está acima dele na árvore de cobertura. Uma BPDU de configuração nunca é enviada no sentido das folhas da árvore para a raiz. Para que um comutador possa enviar informações para comutadores acima dele na árvore um novo tipo de BPDU foi introduzido: **BPDUs de notificações de mudança de topologia** (NMT). As BPDUs de mudança de topologia possuem apenas os 3 primeiros campos de uma BPDU de configuração.

Quando um comutador precisa notificar outros sobre uma mudança de topologia observada, ele começa a enviar BPDUs de NMTs em sua porta raiz. Ao receber uma NMT o comutador responde com uma BPDU de configuração com o valor do campo flag alterado para indicar que essa é uma BPDU de reconhecimento. O comutador que recebeu a notificação também gera outra notificação e a envia através de sua porta raiz. A mensagem é transmitida de comutador a comutador, até que chega na raiz.

Ao ser notificada de uma mudança, a raiz passa um certo tempo transmitindo mensagens de configuração com o campo **flag**, indicando que houve uma mudança de topologia. Os comutadores que recebem BPDUs da raiz também propagam essas mensagens para baixo até chegarem nas folhas da árvore, e todos os comutadores ficam cientes de uma mudança de topologia. A árvore de cobertura será então recal-

culada e durante este intervalo o tempo de envelhecimento das tabelas de endereços dos comutadores é reduzido para **forward delay**. Veja mais informações sobre mudanças de topologia em [CISCO-TC].

6.4 Estado das portas

Para evitar a criação de laços temporários, uma porta não passa do estado "bloqueada" para o estado "ativa" diretamente. Nesta seção, veremos quais os estados possíveis para as portas dos comutadores e como se dá a transição de um estado para outro.

As portas de comutadores que estão com PAC habilitado sempre estão em um dos seguintes estados: bloqueada, estado de escuta, estado de aprendizagem, ativa ou desabilitada. Inicialmente, as portas são bloqueadas. Independente do estado em que esteja, uma porta pode ser desabilitada. Estando no estado bloqueada, uma porta pode passar para o estado de escuta. Apenas do estado de escuta uma porta pode passar para o estado de aprendizagem. Enfim, estando no estado de aprendizagem uma porta pode passar para o estado ativa. A Figura A6-6 mostra um diagrama de estados que ilustra as regras de passagem de estado apresentadas.

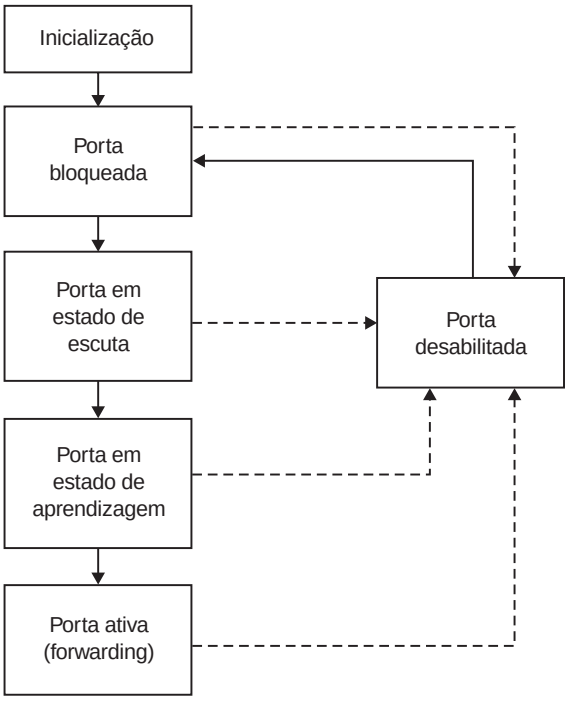


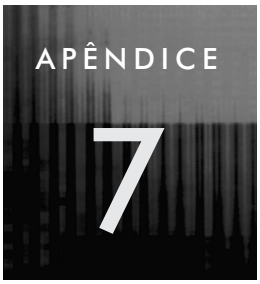
FIGURA A6-6: Estado das portas no PAC.

Quando uma porta vai mudando do estado de bloqueio em direção ao estado ativo, ela vai, aos poucos, processando ou transmitindo mensagens que antes eram

descartadas. Por exemplo, portas no estado de escuta não transmitem BPDUs nem quaisquer outros quadros, exceto quadros com informações de gerência. No estado posterior, elas começam a transmitir BPDUs, mas ainda se negam a repassar ou receber outros tipos de quadros. Veja mais informações sobre a transição dos estados das portas em [CISCO-STP].

6.5 Referências bibliográficas

[CISCO-STP]	Understanding Spanning-Tree Protocol. Em: `http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/sw_ntman/cwsimain/cwsi2/cwsiug2/vlan2/stpapp.htm`
[CISCO-STP-TROUBLE]	Troubleshooting Spanning-Tree Protocol and Related Design Considerations. Em: `http://www.cisco.com/warp/public/473/16.html`
[CISCO-STP-CONF]	Configuring Spanning Tree. Em: `http://www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_5_2/config/spantree.htm`
[CISCO-TC]	Understanding Spanning-Tree Protocol Topology Changes. Em: `http://www.cisco.com/warp/public/473/17.html`
[IEEE802.1D]	Media Access Control (MAC) Bridges. IEEE Padrão 802.1d, 1998.



7 Introdução ao TCP/IP

7.1 Histórico

A arquitetura TCP/IP (ou Internet) teve seu início quando a Agência de Projetos Avançados de Pesquisa em Defesa (Defense Advanced Research Projects Agency – DARPA) dos Estados Unidos da América (EUA) iniciou, por volta de 1969, um projeto de pesquisa para criar uma rede experimental de comutação de pacotes – ARPANET – que deveria ter como suas principais características: eficiência, confiabilidade e comunicação de dados independente de fornecedores.

Devido ao grande sucesso dessa rede, em poucos anos a ARPANET deixa de ser de uso experimental e passa a ter uso operacional. Seu desenvolvimento continua, e inicia-se a definição de um conjunto de protocolos que vão ter dois dos principais adotados como nome para a própria arquitetura: TCP/IP.

Por volta de 1983, o TCP/IP torna-se um padrão na ARPANET e a DARPA propõe a separação da rede em dois blocos distintos:

- ARPANET, usada pela comunidade civil que havia participado direta ou indiretamente no desenvolvimendo do projeto.
- MILNET, usada pela comunidade militar.

À união, ou interconexão, dessas duas redes deu-se o nome de Internet (oriundo de *Internetworking*).

Em 1985, a Fundação Nacional de Ciência (Nacional Science Foundation – NSF) dos EUA promove expansão da Internet para a comunidade científica americana, criando a NSFNET. De 1986 a 1992, a NSF disponibiliza acesso Internet para comunidade científica fora dos Estados Unidos, chegando ao Brasil em 1989.

A partir de 1990, o Ministério de Ciências e Tecnologia (MCT), através do projeto Rede Nacional de Pesquisa (RNP), começa a implantar no Brasil uma rede de comunicação de dados baseada na arquitetura TCP/IP interligando as diversas instituições de ensino e pesquisa, particularmente, as instituições federais de ensino superior.

A partir de 1993, a arquitetura TCP/IP torna-se um padrão *de facto* para a interconexão de redes de computadores de diferentes tecnologias, passando a ser usada para os mais variados fins em todo o mundo.

Hoje, usa-se intensivamente a Internet, e a chamada Internet 2 é articulada e implementada em diversos países, inclusive no Brasil. Ela tem por objetivos a implementação de comunicação em alta velocidade (155/622 Mbps em grosso, 256, 1024, 2048 Mbps no varejo) para possibilitar a disseminação dos mais diversos tipos de informação, particularmente aquelas baseadas em multimídia que integram texto, imagens, áudio e vídeo.

7.2 Características básicas

A arquitetura TCP/IP tem um conjunto de características básicas que a ajudaram a tornar-se um padrão mundial:

- Possui um padrão de protocolos aberto, não associado a nenhum tipo específico de computador ou sistema operacional;
- É independente de hardware específico para acesso ao meio físico de transmissão. Redes TCP/IP podem ser implementadas com tecnologia Ethernet, Token-ring, linha comutada de comunicação de dados – LCCD (linha telefônica), linha privativa de comunicação de dados – LPCD, enlaces Frame-relay e qualquer outro tipo de meio de transmissão;
- Possui um esquema de endereçamento simples e universal – o endereço IP – que permite a identificação única de um elemento da rede (na rede local, ou no planeta);
- Possui protocolos de alto nível padronizados para disponibilização universal e consistente de serviços aos usuários;
- Possui uma documentação ampla acessível na própria Internet sob a forma de "Request for Comments" – RFC's, que não sofrem do rigor imposto aos relatórios técnicos formais. As RFC's contêm as últimas versões das especificações de todos os protocolos TCP/IP padrões;
- A interconexão de duas ou mais redes com arquitetura TCP/IP é feita por meio de roteadores, que nada mais são que computadores especializados – podendo mesmo ser um microcomputador comum – que possuem duas ou mais interfaces de comunicação.

A Figura A7-1 ilustra essa forma de interconexão.

O esquema de endereçamento universal e o uso de roteadores como elementos de interconexão permitem a criação de uma redes virtual envolvendo todo o planeta, conforme ilustra a Figura A7-2.

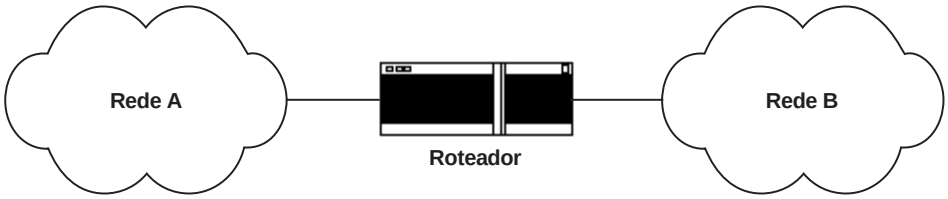


FIGURA A7-1: Interconexão de redes.

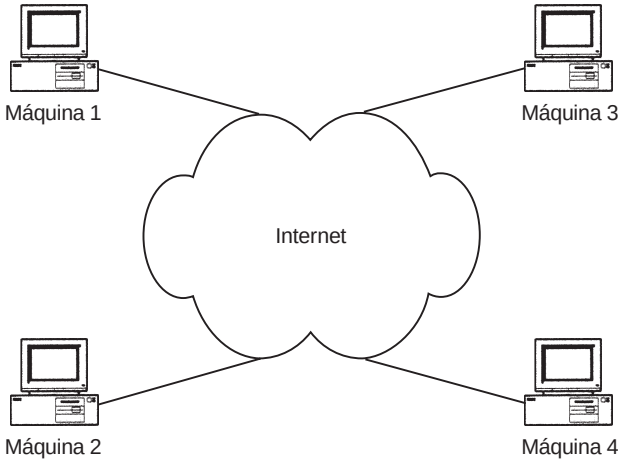


FIGURA A7-2: Rede virtual única.

Na realidade, a rede se apresenta como um conjunto de redes com diversas tecnologias de transmissão e topologias, interligadas por meio de roteadores conforme mostra a Figura A7-3.

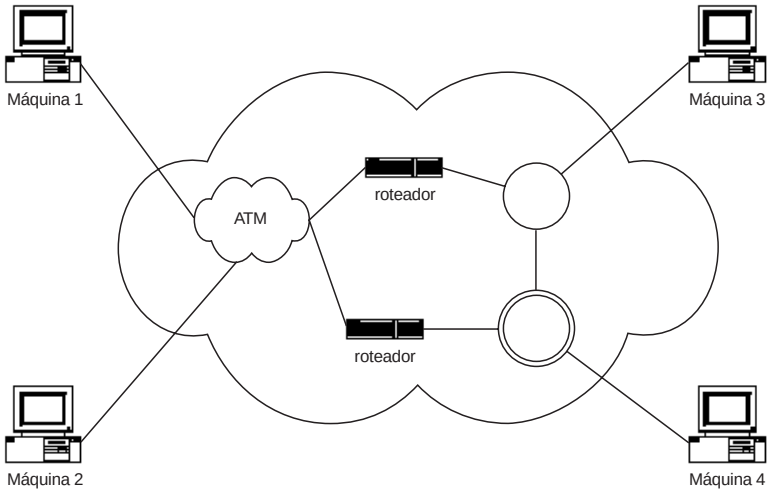


FIGURA A7-3: Rede real.

7.3 Arquitetura

A arquitetura TCP/IP – Internet é organizada em quatro camadas – ou níveis – conceituais, construídas sobre uma quinta camada que não faz parte do modelo – chamada intra-rede – como mostra a Figura A7-4.

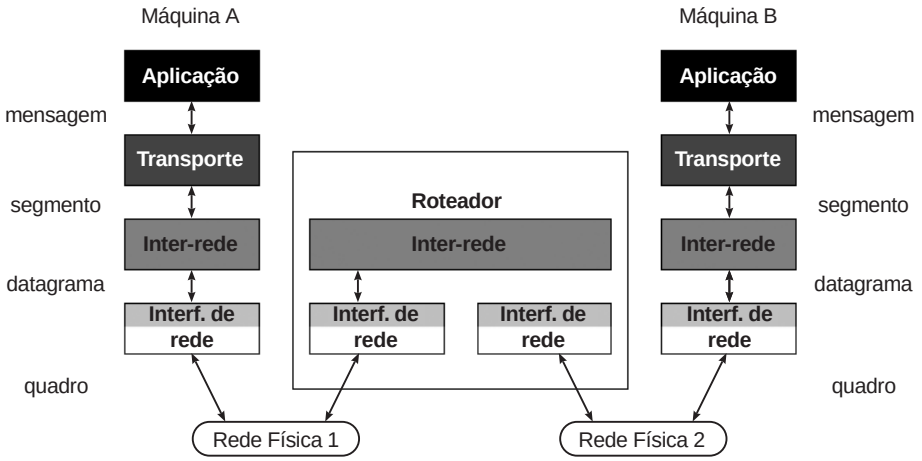


FIGURA A7-4: Arquitetura TCP/IP – Internet.

7.3.1 Camada de aplicação

A camada de aplicação é a parte da arquitetura que oferece serviços de rede aos usuários finais. Faz parte dela, por exemplo, os navegadores WEB tais como Internet Explorer, Netscape e Konqueror – no lado cliente, e os servidores WEB, tais como Internet Information Server e Apache – no lado servidor. A comunicação entre o lado cliente e o lado servidor se dá por meio do protocolo de aplicação chamado *Hipertext Text Transport Protocol – HTTP*.

Diversos outros protocolos servem de base para a implementação de outros serviços. Dentre os mais importantes, citamos:

- SMTP – *Simple Mail Transport Protocol*: fornece o serviço de correio eletrônico;
- TELNET – *Terminal Network Protocol*: fornece o serviço de terminal remoto através da rede;
- FTP – *File Transfer Protocol*: fornece o serviço de transferência de arquivos;
- DNS – *Domain Name Service*: fornece o serviço de mapeamento de nomes em números IP (e vice-versa);
- NFS – *Network File System*: fornece o serviço de compartilhamento de sistemas de arquivos através da rede;

- SNMP – *Simple Network Management Protocol*: fornece o serviço de gerenciamento de equipamentos de forma remota.

7.3.2 Camada de transporte

A camada de transporte é a parte da arquitetura que oferece o serviço de comunicação fim-a-fim (computador-a-computador) confiável ou não aos desenvolvedores de aplicações. Suas funções compreendem:

- Montagem de segmentos para a camada de rede a partir de mensagens vindas da camada de aplicação – fragmentação;
- Montagem de mensagens para a camada de aplicação a partir de segmentos vindos da camada de rede – desfragmentação;
- Transmissão de dados sem conexão e não confiável, através do protocolo *User Datragram Protocol* – UDP, ou com conexão e confiável, através do protocolo *Transport Control Protocol* – TCP;
- Retransmissão temporizada, seqüenciamento de segmentos (máximo de 64 KB cada) e controle de fluxo;
- Armazenamento temporário de segmentos na recepção e transmissão;
- Identificação de processos na origem e no destino.

7.3.2.1 Protocolo UDP

O protocolo UDP oferece serviço de transporte de dados com as seguintes características:

- Baseado em seqüências de bytes estruturados – *datagram*;
- Baseado em serviço sem conexão;
- Menor sobrecarga na rede;
- Não confiável – os dados transmitidos na origem podem não ser entregues no destino e a aplicação transmissora não é avisada disso;
- Sem controle de seqüência – a mensagem na origem tem obrigatoriamente de caber em um único segmento de transporte (64 Kbytes);
- Bom para pequenas quantidades de dados a transmitir;
- Bom para aplicações do tipo consulta/resposta;
- Bom para aplicações que têm seus próprios mecanismos de entrega confiável.

A Figura A7-5 mostra o formato de um segmento UDP, ou datagrama UDP.

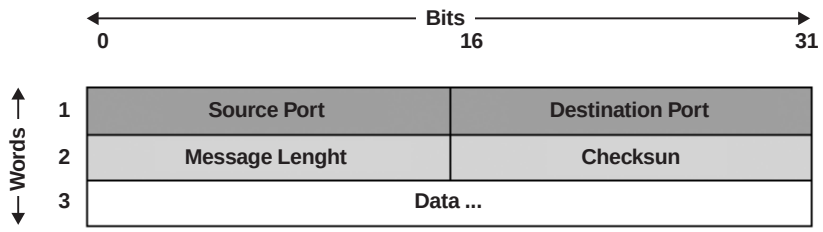


FIGURA A7-5: Datagrama UDP.

Os elementos componentes do datagrama são:

- **Source/Destination Port:** porta de origem/destino (identificam os processos envolvidos na comunicação);
- **Message Lenght:** tamanho do segmento;
- **Checksun:** verificação de erro;
- **Data:** início dos dados

7.3.2.2 Protocolo TCP

O protocolo TCP oferece serviço de transporte de dados com as seguintes características:

- Baseado em seqüências de bytes não-estruturados – *streams*;
- Baseado em serviço com conexão;
- Maior sobrecarga na rede;
- Confiável – se os dados transmitidos na origem não puderem ser entregues no destino, a aplicação transmissora será avisada disso;
- Necessário para maiores quantidades de dados a transmitir;
- Necessário para aplicação do tipo interativa;
- Necessário para aplicação que não tem seu próprio mecanismo de entrega confiável.

A Figura A7-6 mostra o formato de um segmento TCP, ou pacote TCP.
Os elementos componentes do pacote são:

- **Source/Destination Port:** porta de origem/destino (identificam os processos envolvidos na conexão);
- **Sequence Number:** número de seqüência do pacote dentro da seqüência de pacotes da mensagem;

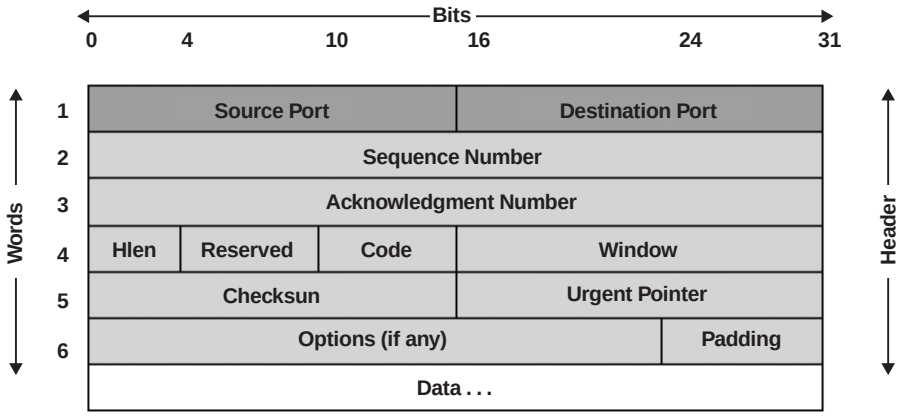


FIGURA A7-6: Pacote TCP.

- **Acknowledgment Number:** número do próximo pacote esperado no destino;
- **Hlen:** tamanho do cabeçalho;
- **Reserved:** reservado;
- **Code bits:** finalidade do pacote:
  - URG: campo *Urgent Pointer* é válido;
  - ACK: campo *Ack Number* é válido;
  - PSH: segmento requer um *push*;
  - RST: *reset* a conexão;
  - SYN: sincronize números de seqüência – abra conexão;
  - FIN: origem terminou sua seqüência de bytes – feche conexão
- **Window:** tamanho da janela deslizante (número de pacotes enviados em seqüência antes de receber reconhecimento);
- **Checksun:** verificação de erro no pacote;
- **Urgent Pointer:** posição onde os dados urgentes se encontram no pacote;
- **Options:** tamanho (opcional) máximo de segmento (“*Maximum Segment Size*”);
- **Padding:** preenchimento;
- **Data:** dados do segmento.

7.3.3 Camada de inter-rede (ou somente rede)

A camada de inter-rede fornece o serviço de entrega de datagramas UDP ou pacotes TCP entre uma máquina origem e uma máquina destino. Como oferece um serviço

não orientado a conexão, diz-se que a camada inter-rede oferece um serviço baseado em datagrama. No caso, datagrama IP – Internet Protocol. O protocolo IP é considerado o "tijolo" de construção da Internet.

As funções da camada inter-rede compreendem:

- Montagem de datagramas IP a partir de segmentos recebidos da camada de transporte – fragmentação de segmentos;
- Montagem de segmentos para a camada de transporte a partir de datagramas IP recebidos da camada de acesso ao meio físico – desfragmentação de segmentos;
- Transmissão de dados sem conexão e não confiável – modo datagrama;
- Identificação de máquina de origem e máquina de destino;
- Encaminhamento ou roteamento de datagramas IP através da rede lógica;
- Integração de diversas redes físicas formando uma única rede lógica.

A Figura A7-7 mostra o formato de um datagrama IP.

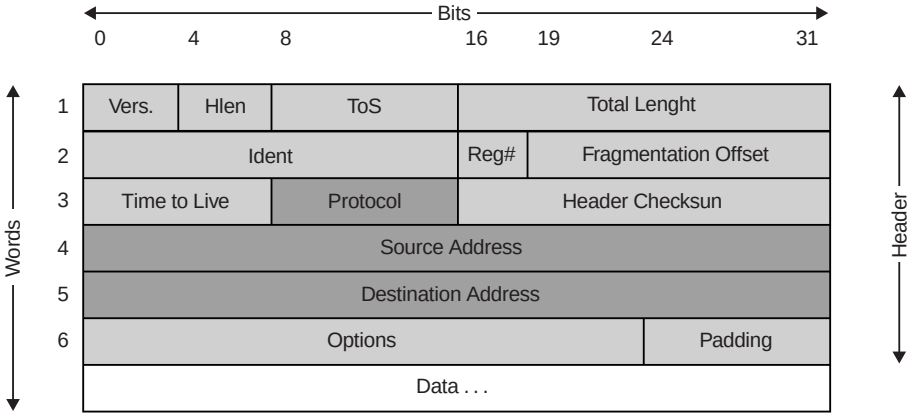


FIGURA A7-7: Datagrama IP.

Os elementos componentes do datagrama são:

- **Version:** versão do IP (normalmente 4, mas diversas organizações já começam a usar a versão 6);
- **Hlen:** tamanho do cabeçalho do datagrama;
- **ToS:** tipo do serviço (precedência normal/controle, baixo retardo, alta eficiência, alta confiabilidade), sem garantia de cumprimento;
- **Total Length:** tamanho total do datagrama (máximo de 64 Kbytes);
- **Ident:** identificação do datagrama (único para cada datagrama);

- **Flags:** controle de fragmentação (habilita/desabilita fragmentação, marca de fim do datagrama original);
- **Fragmentation Offset:** deslocamento do fragmento;
- **TTL:** tempo de vida do datagrama (inicia em N, decrementa a cada passagem por um roteador; chegando em 0 (zero), o datagrama é descartado e é gerada uma mensagem de erro);
- **Protocol:** protocolo de nível superior (TCP, UDP);
- **Header checkksun:** verificação de integridade;
- **Source Address:** endereço origem (máquina emissora);
- **Dest Address:** endereço destino (máquina receptora);
- **Options:** opções de teste e depuração:
  - **Record Route Option:** o datagrama guarda endereços de roteadores intermediários por onde passou;
  - **Source Route Option:** o sistema origem define rota que um datagrama deve seguir;
  - **Timestamp Option:** o datagrama guarda informação sobre data e hora que chegou aos roteadores intermediários;
- **Padding:** preenchimento;
- **Data:** dados transportados.

7.3.4 Camada de interface de rede

A camada de interface de rede fornece o serviço de entrega de dados entre máquinas diretamente conectadas entre si. De modo geral, desempenha as funções de enlace de dados e de acesso físico do padrão RM-OSI.

As funções da camada interface de rede compreendem:

- Receber datagramas IP da camada inter-rede e transmiti-los através da tecnologia de rede física disponível;
- Fazer o encapsulamento de datagramas IP em quadros da rede física disponível (eventualmente com fragmentação);
- Fazer o mapeamento de endereços lógicos Internet em endereços físicos de equipamentos na rede.

Em geral, esta camada é implementada através de gerentes ("drivers") de dispositivos que podem ser disponibilizados pelo próprio fabricante da rede física disponível. Isso permite a implantação do TCP/IP sobre qualquer hardware de rede ou

sistema de comunicação de dados (Ethernet, Token Ring, FDDI, X25, Frame Relay, ATM, linhas seriais etc.). Em linhas seriais é comum o uso dos protocolos SLIP, PPP ou outro protocolo patenteado.

7.4 Mapeamento de endereços lógicos Internet em endereços físicos de rede

Endereços IP são lógicos (só existem em software). Para usar a rede física, é necessário que se faça o mapeamento do endereço lógico no endereço físico. Como se faz isso? Usando o protocolo ARP (*Address Resolution Protocol*). Através de uma mensagem de difusão (broadcast), uma máquina pergunta na rede qual é o endereço físico que correspondente a um determinado endereço IP. Somente a máquina referenciada responde à solicitação. Esse padrão de comportamento é ilustrado na Figura A7-8.

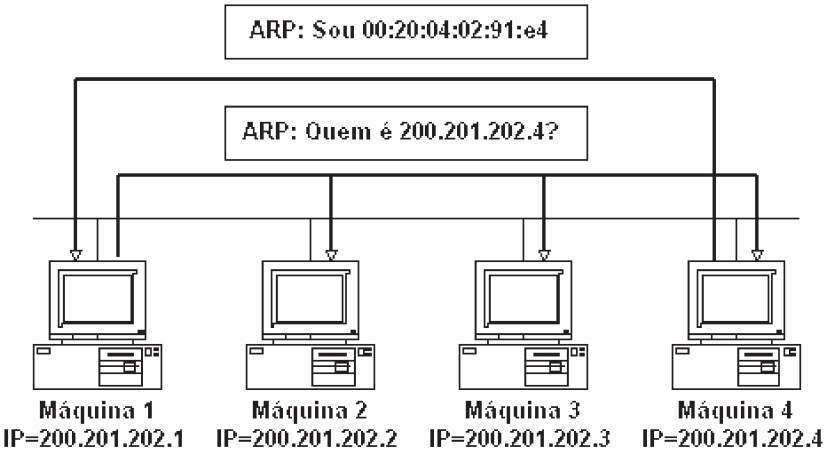


FIGURA A7-8: Consulta/resposta ARP.

Esse esquema funciona eficientemente porque há uma tabela de mapeamento ARP em cada máquina da rede. A máquina que faz uma pergunta na rede já informa seu endereço IP e seu endereço físico; as máquinas que ouvem a pergunta armazenam essa informação nas suas tabelas ARP.

7.5 Encapsulamento de dados

A Figura A7-9 resume o esquema de encapsulamento de dados da arquitetura TCP/IP.

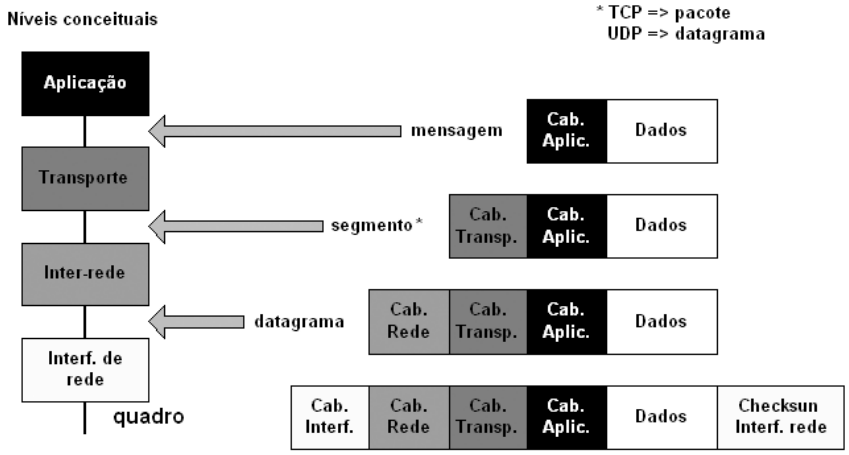


FIGURA A7-9: Ecapsulamento de dados TCP/IP.

7.6 Arquitetura TCP/IP *versus* arquitetura RM-OSI

A Figura A7-10 mostra as equivalências entre a arquitetura TCP/IP e arquitetura RM-OSI.

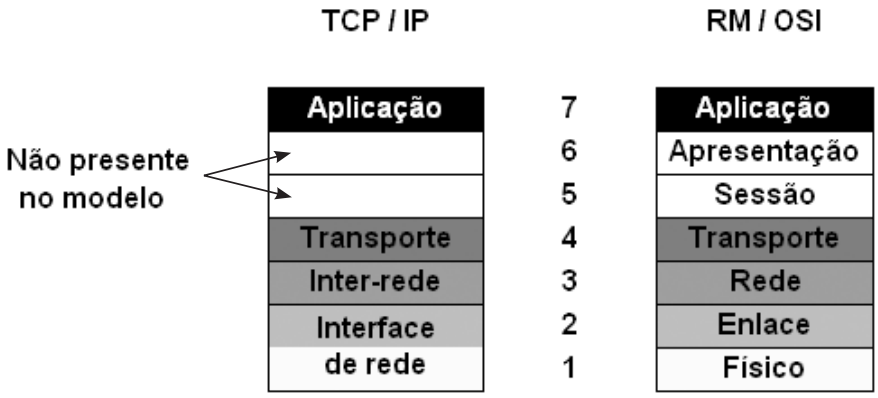


FIGURA A7-10: TCP/IP versus RM-OSI.

Na arquitetura TCP/IP, as funcionalidades das camadas de sessão e apresentação do modelo de referência OSI são incorporadas na camada de aplicação e transporte.

7.7 Endereçamento

A arquitetura TCP/IP define um esquema de endereçamento universal – endereço IP – que deve:

- Identificar unicamente uma rede na Internet;
- Identificar unicamente cada máquina de uma rede.

Um endereço IP compõe-se de uma quadra de números naturais na faixa de 0 (zero) a 255 – um byte, normalmente representado por

número . número . número . número

Exemplos de endereços IP são:

- 100 . 101 . 102 . 103
- 150 . 165 . 166 . 0
- 200 . 201 . 203 . 255

Os endereços IP são divididos em 5 classes: A, B, C, D e E, conforme mostrado a seguir.

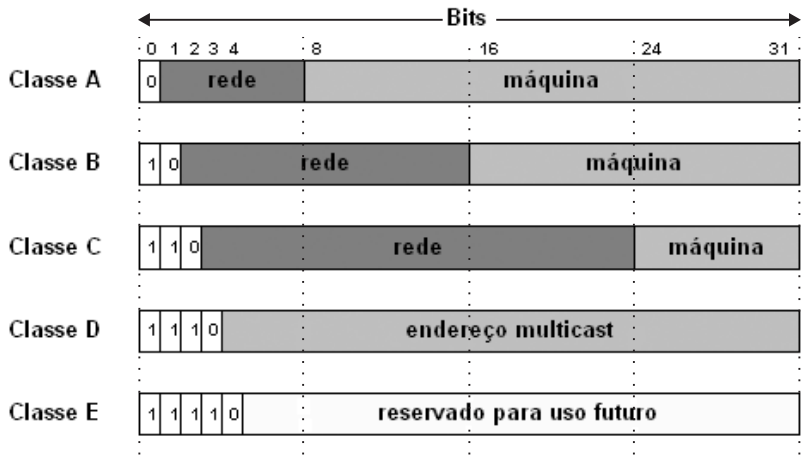


FIGURA A7-11: Classes de endereços IP.

Essa classificação fornece os limites de endereçamento apresentados na Tabela A7-1.

TABELA A7-1: Limites de endereços de cada classe.

Classe	Menor Endereço	Maior Endereço
A	1 . 0 . 0 . 0	126 . 255 . 255 . 255
B	128 . 0 . 0 . 0	191 . 255 . 255 . 255
C	192 . 0 . 0 . 0	223 . 255 . 255 . 255
D	224 . 0 . 0 . 0	239 . 255 . 255 . 255
E	240 . 0 . 0 . 0	247 . 255 . 255 . 255

Observa-se que alguns endereços são reservados.

7.7.1 Endereço de loopback

Um dos endereços reservados é o endereço 127.0.0.0 da classe A. Ele é usado para testes do TCP/IP e para comunicação interprocessos em uma máquina local. Quando um desenvolvedor de aplicações escreve seu sistema cliente-servidor, uma forma de testá-lo é ativar o lado servidor em uma máquina A e ativar o lado cliente em uma máquina B para verificar a correta comunicação entre os dois lados.

Uma forma mais confortável e menos dispendiosa de realizar esses testes é ativar o lado servidor e o lado cliente em uma única máquina. Para tanto, basta que cada porção do sistema faça referência ao endereço IP 127.0.0.1, dado que ele, na implementação da arquitetura TCP/IP, é uma referência explícita à máquina onde o programa está sendo executado.

Quando uma aplicação usa o endereço de *loopback* como destino, o software do protocolo TCP/IP devolve os dados sem gerar tráfego na rede. É uma forma simples de fazer com que um cliente local fale com o servidor local correspondente, sem que se tenha de alterar o programa cliente e/ou o programa servidor.

Do ponto de vista do programador de aplicações, seu sistema funciona sempre do mesmo jeito, não importando se está ou não usando a rede de comunicação.

7.7.2 Máscara de rede – *netmask*

A máscara de rede é um padrão de endereço IP que serve para extrair a identificação de rede de um endereço IP através de uma operação simples de AND binário.

Exemplo:

Endereço IP:	200	.	237	.	190	.	21	
								AND
Máscara de rede:	255	.	255	.	255	.	0	
								=
	200	.	237	.	190	.	0	

Para obter o endereço de máquina, faz-se uma operação binária AND com o complemento de 2 da máscara de rede.

Endereço IP:	200	.	237	.	190	.	21	
								AND
Máscara de rede:	0	.	0	.	0	.	255	
								=
	0	.	0	.	0	.	21	

7.7.3 Endereço de difusão – broadcast

O endereço de difusão serve para endereçar simultaneamente todas as máquinas de uma rede. Ele é formado colocando-se todos os bits da parte de endereçamento de máquina de um endereço IP com valor 1.

Exemplo:

Endereço IP	Endereço de difusão
200 . 237 . 190 . 21	200 . 237 . 190 . 255
150 . 165 . 166 . 21	150 . 165 . 255 . 255
26 . 27 . 28 . 21	26 . 255 . 255 . 255

Exemplo final:

Endereço IP	Máscara de rede	Endereço de rede	Endereço de máquina	Endereço de difusão
200.4.2.91	255.255.255.0	200.4.2.0	0.0.0.91	200.4.2.255
150.165.7.8	255.255.0.0	150.165.0.0	0.0.7.8	150.165.255.255
26.27.28.21	255.0.0.0	26.0.0.0	0.27.28.21	26.255.255.255

A Figura A7-12 mostra um exemplo de atribuição de endereços IP às máquinas de uma rede local.

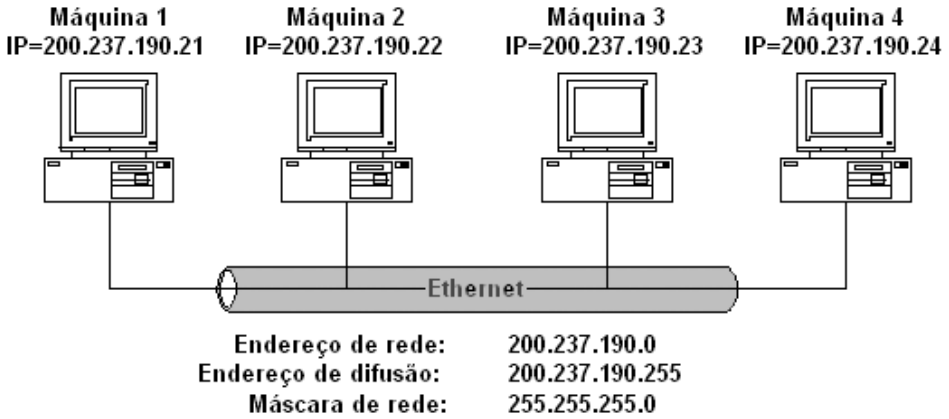


FIGURA A7-12: Endereçamento de rede TCP/IP.

7.7.4 Sub-redes

O esquema de endereçamento da arquitetura TCP/IP pode ser mudado localmente, a critério do administrador de rede, usando bits de endereçamento de máquina como um adicional para endereçamento de rede.

Para tanto, deve-se definir uma máscara de rede "não-padrão" que permita extrair os endereços de rede e de máquina corretamente.

Por exemplo, o administrador da classe B 150.160 (que comporta aproximadamente 256 x 256 máquinas) pode "tirar" 8 bits do endereço de máquina e "acrescentar" 8 bits no endereço de rede, passando a dispor das redes:

150 . 160 . 1 . 0
150 . 160 . 2 . 0
. . .
150 . 160 . 254 . 0

Agora, cada sub-rede dispõe de endereços de máquina variando de 1 a 254. A máscara de rede usada passaria a ser 255 . 255 . 255 . 0.

Algo semelhante pode ser feito com a classe C 200.237.190.0, "tirando" 3 bits do endereço de máquina, "colocando-os" no endereço de rede, como apresentado na Tabela A7-2.

TABELA A7-2: Exemplo de endereços de sub-rede.

Endereço de rede	Máquina inicial	Máquina final	Endereço de difusão
200.237.190.0	1	30	200.237.190.31
200.237.190.32	33	62	200.237.190.63
200.237.190.64	65	94	200.237.190.95
200.237.190.96	97	126	200.237.190.127
200.237.190.128	129	158	200.237.190.159
200.237.190.160	161	190	200.237.190.191
200.237.190.192	193	222	200.237.190.223
200.237.190.224	225	254	200.237.190.255

A máscara de rede usada passaria a ser 255.255.255.224, de acordo com o ilustrado na Figura A7-13.

A seguir, temos a utilização de sub-redes em duas redes locais.

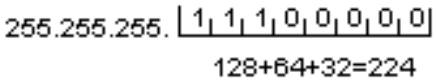


FIGURA A7-13: Máscara de sub-rede (exemplo).

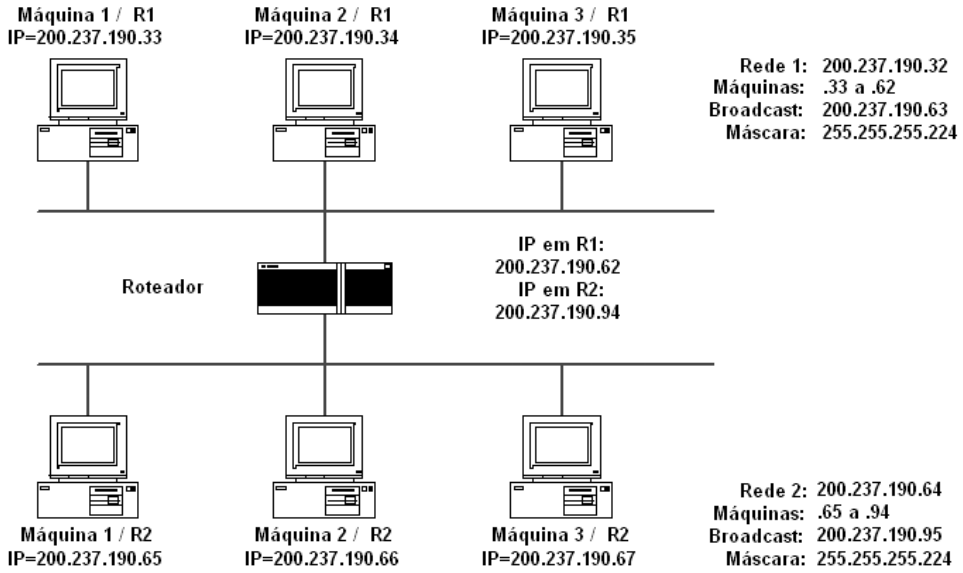


FIGURA A7-14: Uso de sub-rede.

Observa-se que um endereço IP deve ser atribuído a cada interface de comunicação de um equipamento ligado em rede. Na figura anterior, o roteador está conectado em ambas as redes R1 e R2, tendo em cada uma um endereço.

7.8 Bibliografia

Comer, D. *Internetworking with TCP/IP: Principles, Protocols, and Architectures*. Volume 1. 4a edição. Prentice Hall, 2000.

APÊNDICE

8

8 Rotear é preciso

8.1 Conceituação

Roteamento é o mecanismo através do qual duas máquinas em comunicação "encontram" e utilizam o melhor caminho através de uma rede. O processo envolve:

- Determinar que caminhos estão disponíveis;
- Selecionar o "melhor" caminho para uma finalidade particular;
- Usar o caminho para chegar aos outros sistemas;
- Ajustar o formato dos dados às tecnologias de transporte disponíveis (Maximum Transmission Unit – MTU).

Na arquitetura TCP/IP, o roteamento é baseado no endereçamento IP, particularmente, na parte de identificação de rede de um endereço IP. Toda a tarefa é desenvolvida na camada Inter-rede da pilha de protocolos TCP/IP. A Figura A8-1 ilustra esse esquema.

8.2 Mecanismos de entrega de dados

Quando uma máquina destino encontra-se na mesma rede física da máquina origem, faz-se uma ENTREGA DIRETA de dados. Nesse caso, o mapeamento do endereço lógico (IP) para o endereço físico (Ethernet, Token-ring) é feito via protocolo ARP, seguido da entrega dos dados. A Figura A8-2 ilustra esse mecanismo.

Quando uma máquina destino não se encontra na mesma rede física da máquina origem, faz-se uma ENTREGA INDIRETA de dados. Nesse caso, os dados são enviados para o roteador mais próximo, e assim sucessivamente até atingirem a máquina destino. A Figura A8.3 ilustra esse mecanismo.

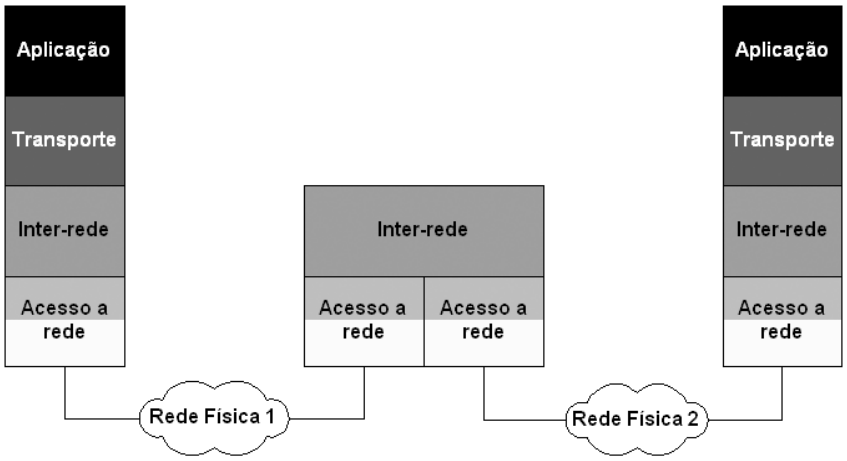


FIGURA A8-1: Roteamento com IP.

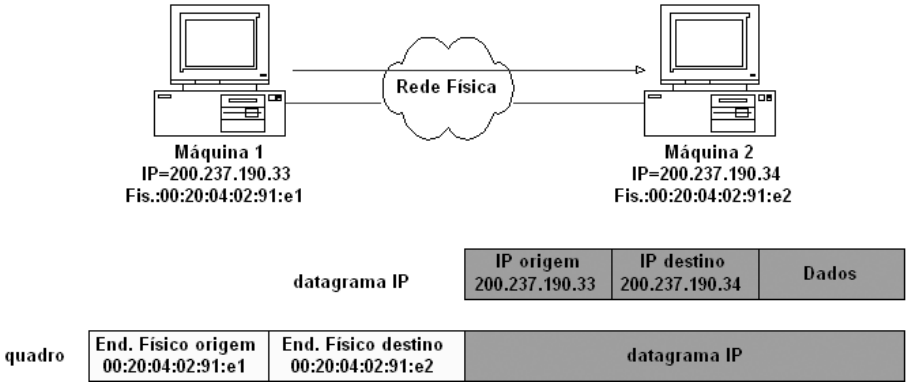


FIGURA A8-2: Entrega direta de dados.

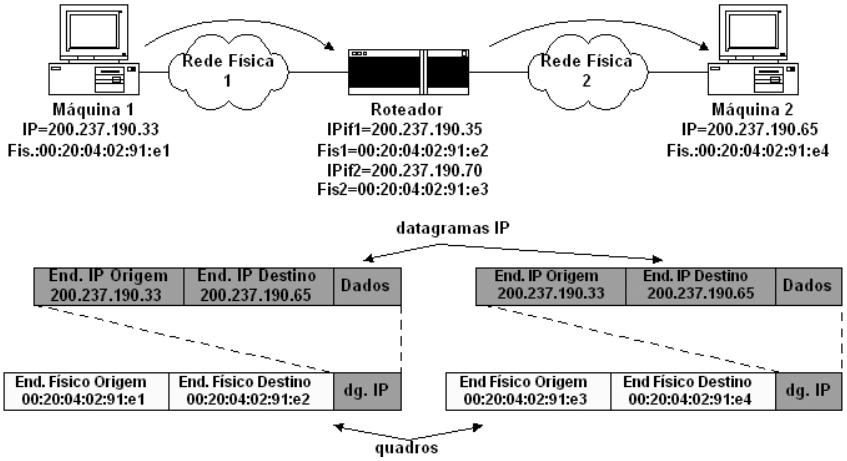


FIGURA A8-3: Entrega indireta de dados.

8.3 Máscara de rede e roteamento

Para saber como entregar um datagrama, a máquina/roteador origem precisa saber se a máquina/roteador destino pertence a uma rede diretamente conectada a ela ou não.

Isso é feito através de uma operação de AND binário do endereço IP próprio e do endereço IP destino, com a máscara de rede. Se a identificação de rede do endereço IP local for igual à identificação de rede do endereço IP destino, origem e destino estão na mesma rede física.

A Figura A8.4 mostra um exemplo em que três máquinas distintas são consideradas como pertencentes à mesma rede.

Máscara de Rede (255.255.255.224)

Endereço IP	Byte 3 e 2	Byte 1 e 0 (3 bits)	Byte 0(5 bits)	Endereço de Rede
200.237.190.33	11001000 11101101	10111110 001	00001	200.237.190.32
200.237.190.43	11001000 11101101	10111110 001	01011	200.237.190.32
200.237.190.53	11001000 11101101	10111110 001	10101	200.237.190.32

FIGURA A8-4: Determinando endereço de rede.

8.4 Tabela de roteamento

Cada máquina/roteador da rede precisa dispor de informações sobre a(s) rede(s) a(s) qual(is) está conectada. Tais informações permitem à máquina/roteador fazer a entrega de dados como visto anteriormente. A esse conjunto de informações dá-se o nome de **Tabela de Roteamento**.

A tabela de roteamento deve guardar informações sobre que conexões estão disponíveis para se atingir uma determinada rede e alguma indicação de performance ou custo do uso de uma dada conexão.

Antes de enviar um datagrama, uma máquina/roteador precisa consultar a tabela de roteamento para decidir por qual conexão de rede enviá-lo. Obtida a resposta, a máquina faz a entrega do datagrama de forma direta (destino em rede diretamente conectada) ou através de um roteador (destino não em rede diretamente conectada).

A Figura A8-5 ilustra o conceito de tabela de roteamento.

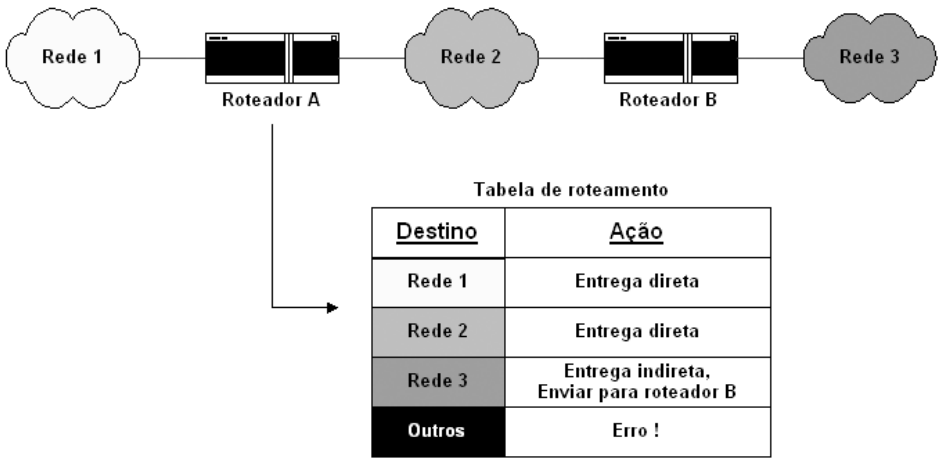


FIGURA A8-5: Exemplo de tabela de roteamento (esquema).

8.4.1 Entradas na tabela de roteamento

As entradas da tabela de roteamento fornecem informações sobre roteamento para redes lógicas; cada entrada tem tipicamente a forma:

Endereço IP da rede destino (D)	Máscara de rede (M)	Endereço IP do roteador (R)

Cada entrada especifica uma rede destino, a máscara de rede e o próximo roteador a ser usado para se chegar à rede destino. Para redes diretamente conectadas, o endereço IP do roteador destino é o endereço da interface de conexão à rede. Algumas entradas podem especificar, ocasionalmente, o endereço IP de uma máquina destino.

É comum a existência de uma entrada para a rede destino "default", cujo roteador indicado deve receber o datagrama, cujo endereço destino não pertença a nenhuma das redes destino registradas na tabela. A rede default é normalmente indicada como rede 0.0.0.0 com máscara 0.0.0.0.

8.5 Algoritmo de roteamento

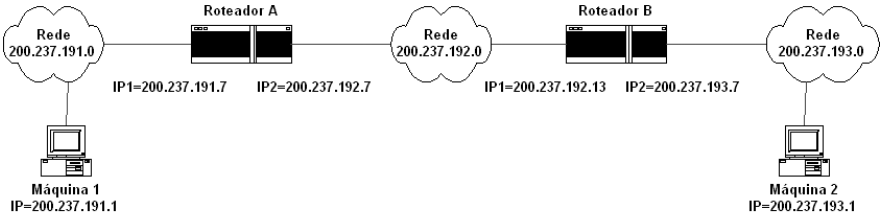
Dada a tabela de roteamento e um datagrama a ser encaminhado (roteado), o algoritmo de roteamento realiza, basicamente, as seguintes atividades:

- Extrai o endereço IP destino (IP-dest) do datagrama;
- Para cada entrada i da tabela de roteamento (Di, Mi, Ri):
  - Calcula o endereço IP da rede destino (IPR-dest) fazendo IPR-dest = IP-dest AND Mi;

- Se IPR-dest = Di, encaminha o datagrama para o roteador Ri;

- Se não encontrar nenhuma alternativa para encaminhamento do datagrama, declara "Erro de Roteamento".

A Figura A8.6 mostra as tabelas de roteamento das máquinas 1 e 2 e dos roteadores A e B de uma rede com arquitetura TCP/IP, não ligada à Internet.



Máquina 1

Rede	Máscara	Roteador
200.237.191.0	255.255.255.0	200.237.191.1
0.0.0.0	0.0.0.0	200.237.191.7

Máquina 2

Rede	Máscara	Roteador
200.237.193.0	255.255.255.0	200.237.193.1
0.0.0.0	0.0.0.0	200.237.193.7

Roteador A

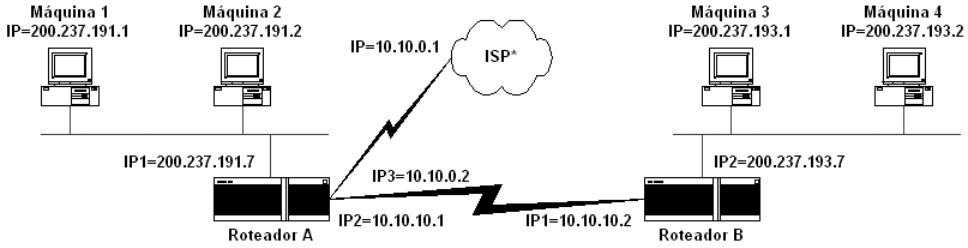
Rede	Máscara	Roteador
200.237.191.0	255.255.255.0	200.237.191.7
200.237.192.0	255.255.255.0	200.237.192.7
200.237.193.0	255.255.255.0	200.237.192.13

Roteador B

Rede	Máscara	Roteador
200.237.191.0	255.255.255.0	200.237.192.7
200.237.192.0	255.255.255.0	200.237.192.13
200.237.193.0	255.255.255.0	200.237.193.7

FIGURA A8-6: Tabela de roteamento (exemplo 1).

A Figura A8.7 a seguir mostra as tabelas de roteamento das máquinas 1 e 3 e dos roteadores A e B de uma rede com arquitetura TCP/IP, ligada à Internet.



Máquina 1

Rede	Máscara	Roteador
200.237.191.0	255.255.255.0	200.237.191.1
0.0.0.0	0.0.0.0	200.237.191.7

Máquina 3

Rede	Máscara	Roteador
200.237.193.0	255.255.255.0	200.237.193.1
0.0.0.0	0.0.0.0	200.237.193.7

Roteador A

Rede	Máscara	Roteador
200.237.191.0	255.255.255.0	200.237.191.7
200.237.193.0	255.255.255.0	10.10.10.2
0.0.0.0	0.0.0.0	10.10.0.1

Roteador B

Rede	Máscara	Roteador
200.237.191.0	255.255.255.0	10.10.10.1
200.237.193.0	255.255.255.0	200.237.193.7
0.0.0.0	0.0.0.0	10.10.10.1

* ISP - Internet Service Provider

FIGURA A8-7: Tabela de roteamento (exemplo 2).

8.6 Entendendo o Routing Information Protocol – RIP

Como vimos na seção anterior, cada nó de uma rede com arquitetura TCP/IP precisa dispor de uma tabela de roteamento para poder encaminhar corretamente datagramas IP através da rede.

A construção e manutenção dessas tabelas de roteamento também é conhecida como modelo ou algoritmo de roteamento.

Um algoritmo de roteamento pode ser:

- **Não adaptativo**, quando as decisões de roteamento são definidas antecipadamente (pelo gerente da rede, por exemplo) e colocadas nos roteadores quando estes são ligados – **roteamento estático**;
- **Adaptativo**, quando as decisões de roteamento são [re]definidas continuamente, de acordo com a estrutura da rede (topologia, carga etc.) – **roteamento dinâmico**.

Nesse último caso, o algoritmo de roteamento pode ser do tipo **Vetor-Distância** ou **Estado-do-Enlace** cujas características são resumidas na Tabela A8-1.

TABELA A8-1: Classificação de algoritmos adaptativos.

VETOR DISTÂNCIA	ESTADO DE ENLACE
Em cada nó da rede, o algoritmo mantém uma tabela com a menor a distância para cada destino	Em cada nó da rede, o algoritmo mantém uma tabela com o melhor caminho para cada destino
Distância é medida pela quantidade de roteadores que o datagrama IP tem de atravessar (*hop count*) até o destino	O melhor caminho é definido com base em informações da rede (velocidade, atraso, taxa de ocupação de enlaces etc.)
Roteadores vizinhos trocam informações a respeito de suas tabelas de roteamento (tabela inteira)	Roteadores vizinhos trocam informações a respeito dos seus enlaces (somente o que for alterado desde a última troca de informação)

8.6.1 Roteamento dinâmico com rip

Cada máquina/roteador mantém uma tabela de roteamento onde é indicada a "menor distância" conhecida até cada rede destino e que conexão usar para chegar lá. Essa "menor distância" é a chamada **métrica** do roteamento dinâmico, e, no caso do RIP, é definida como sendo a quantidade de nós intermediários que um datagrama tem de atravessar até chegar ao seu destino.

Periodicamente, cada máquina/roteador envia sua tabela de roteamento para todos os seus vizinhos. Ao receber a tabela de roteamento de um vizinho, uma máquina/roteador compara a tabela recebida com a sua própria tabela de roteamento, fazendo uma atualização da sua tabela para mantê-la atualizada em relação a todas as redes acessíveis e suas respectivas métricas de acesso.

Normalmente, uma máquina/roteador exporta sua tabela de roteamento a cada 30 segundos.

Uma máquina/roteador que deixa de receber informações de um vizinho por 60 segundos passa a considerar que as redes destino que ele aprendeu a atingir através desse vizinho estão inacessíveis no momento. Ele faz isso atribuindo métrica 16 a essas entradas em sua tabela de rota.

Se a máquina/roteador não receber informações desse vizinho por 90 segundos, passa a considerar que as redes destino não existem mais. Ele faz isso eliminando as entradas correspondentes de sua tabela de rotas.

8.6.2 Exemplo completo

Para melhorar nossa compreensão, vamos ver um exemplo completo. Vamos considerar que uma empresa deseja interconectar suas 4 redes locais através de linhas de comunicação privativas (LPCDs) de acordo com a Figura A8-8.

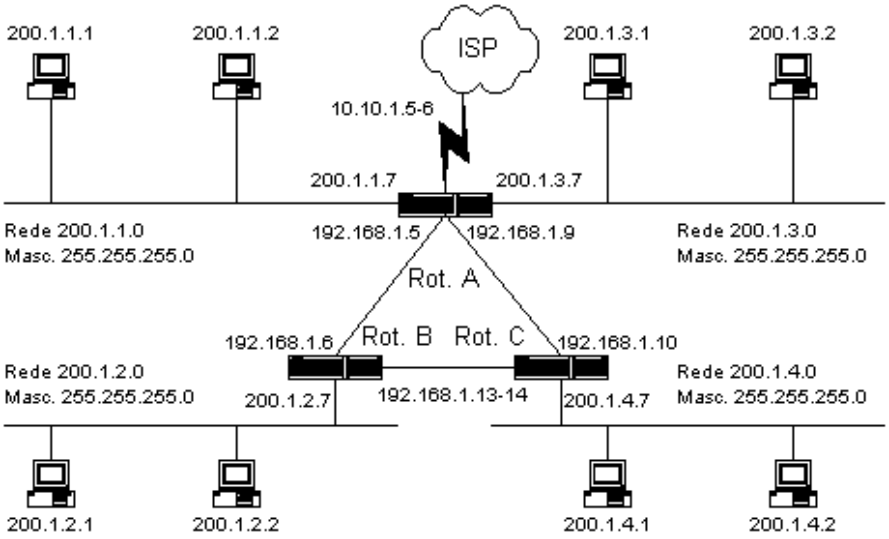


FIGURA A8-8: Rede usada no exemplo.

Para realizar tal comunicação, a empresa pretende usar três roteadores (A, B e C), interligados em anel para prover redundância de acesso entre as redes. Dessa forma, haverá dois caminhos disponíveis entre quaisquer duas redes locais.

Como seria impraticável a utilização de roteamento estático nesse caso, a empresa optou pela utilização do roteamento dinâmico usando o protocolo RIP. Para tanto, ela ativou o serviço de roteamento dinâmico com protocolo RIP em cada um dos seus roteadores (A, B e C).

Os endereços de rede, máscara e difusão usados em cada segmento de rede são indicados na Tabela A8-2.

TABELA A8-2: Endereços utilizados.

Segmento	Rede	Máscara	Endereços	Difusão
ISP - Empresa	10.10.10.4	255.255.255.252	.5 e .6	10.10.10.7
Rot. A – Rot. B	192.168.1.4	255.255.255.252	.5 e .6	192.168.1.7
Rot. A – Rot. C	192.168.1.8	255.255.255.252	.9 e .10	192.168.1.11
Rot. B – Rot. C	192.168.1.12	255.255.255.252	.13 e .14	192.168.1.15
Rede 1	200.1.1.0	255.255.255.0	.1 a .254	200.1.1.255
Rede 2	200.1.2.0	255.255.255.0	.1 a .254	200.1.2.255
Rede 3	200.1.3.0	255.255.255.0	.1 a .254	200.1.3.255
Rede 4	200.1.4.0	255.255.255.0	.1 a .254	200.1.4.255

Logo após a ligação dos roteadores, cada um deles vai conhecer as redes às quais estiver diretamente conectado. As tabelas de roteamento de cada um deles seriam semelhantes à ilustração a seguir.

Roteador A

Rede destino	Máscara	Roteador/Interface	Métrica
200.1.1.0	255.255.255.0	200.1.1.7	0
200.1.3.0	255.255.255.0	200.1.3.7	0
192.168.1.4	255.255.255.252	192.168.1.5	0
192.168.1.8	255.255.255.252	192.168.1.9	0
10.10.10.4	255.255.255.252	10.10.10.6	0

Roteador B

Rede destino	Máscara	Roteador/Interface	Métrica
200.1.2.0	255.255.255.0	200.1.2.7	0
192.168.1.4	255.255.255.252	192.168.1.6	0
192.168.1.12	255.255.255.252	192.168.1.13	0

Roteador C

Rede destino	Máscara	Roteador/Interface	Métrica
200.1.4.0	255.255.255.0	200.1.4.7	0
192.168.1.8	255.255.255.252	192.168.1.10	0
192.168.1.12	255.255.255.252	192.168.1.14	0

Após algumas poucas trocas de tabelas de roteamento entre os roteadores vizinhos, o chamado tempo de convergência, todos os roteadores teriam suas tabelas de roteamento atualizadas com informações a respeito de todas as redes acessíveis. A tabela de roteamento do roteador B, por exemplo, seria como ilustrado a seguir.

Roteador B

Rede destino	Máscara	Roteador/Interface	Métrica
200.1.1.0	255.255.255.0	192.168.1.5	1
200.1.2.0	255.255.255.0	200.1.2.7	0
200.1.3.0	255.255.255.0	192.168.1.5	1
200.1.4.0	255.255.255.0	192.168.1.14	1
192.168.1.4	255.255.255.252	192.168.1.6	0
192.168.1.8	255.255.255.252	192.168.1.5	1
192.168.1.12	255.255.255.252	192.168.1.13	0
10.10.10.4	255.255.255.252	192.168.1.5	1
0.0.0.0	255.255.255.252	192.168.1.5	2

É interessante observar que nos enlaces ponto a ponto, como o estabelecido entre os roteadores A e B, foram usados endereços IP de subrede. Em particular, nes-

ses enlaces foram usados endereços privativos na faixa 10.0.0.0 (pelo provedor de acesso Internet) e na faixa 192.162.x.y. Essa é uma prática comum para economizar endereços IP não privativos.

Nas máquinas dos usuários, o roteamento utilizado será o estático, com cada máquina tendo como rota default o roteador que dá acesso à rede local onde a máquina se localiza. A tabela a seguir ilustra a tabela de roteamento da máquina 2 da rede 200.1.3.0.

Rede destino	Máscara	Roteador/Interface
200.1.3.0	255.255.255.0	200.1.3.2
0.0.0.0	0.0.0.0	200.1.3.7

O protocolo de roteamento dinâmico RIP tem sua aplicação limitada dadas suas duas grandes deficiências:

- Sua noção de métrica é baseada somente na quantidade de nós intermediários que um datagrama IP tem de atravessar para atingir uma dada rede destino. Tal métrica não leva em consideração as características dos enlaces de rede que o datagrama terá de percorrer, particularmente a velocidade, o atraso as taxas de congestionamento e de erro desses enlaces;
- Sua pequena abrangência, dado que a métrica 16 é considerada infinita em termos de roteamento, ou seja, para o RIP, uma métrica 16 indica que a rede destino é inacessível.

Apesar dessas deficiências, o roteamento dinâmico com RIP ainda é usado em diversas redes mundo afora.

Um substituto bem mais elaborado, que supre as deficiências anteriores é o protocolo OSPF (Open Shortest Path First) que vem paulatinamente substituindo o RIP.

8.7 Bibliografia

Comer, D. *Internetworking with TCP/IP: Principles, Protocols, and Architectures*. Volume 1. 4ª edição. Prentice Hall, 2000.

9 Introdução ao DHCP

9.1 Motivação

Para que uma máquina possa participar de uma rede TCP/IP, é necessário que seu software TCP/IP esteja corretamente configurado. Configurar o software TCP/IP (a rede) em uma máquina significa basicamente[1] definir endereço IP da máquina, máscara de rede e endereços IP do roteador default e do servidor de nomes a serem utilizados.

Imagine uma rede TCP/IP com cinco máquinas clientes e um servidor. Seria fácil para João, o administrador dessa rede, configurar manualmente o software TCP/IP em cada uma das suas seis máquinas.

Agora imagine que essa rede está crescendo rapidamente, e passou a conter 50 máquinas. Cada vez que uma nova máquina é inserida na rede, João é chamado, pois só ele tem o controle de quais endereços IPs já foram atribuídos. João precisa ser cauteloso para evitar duplicação de endereços IPs.

Suponha agora que a rede da qual estamos falando tenha 200 máquinas clientes e meia dúzia de servidores. Dois outros profissionais foram contratados para ajudar João. Os três têm autoridade para configurar o software TCP/IP das máquinas. Para que não haja IPs duplicados na rede, eles precisam manter as informações sobre a atribuição de IPs atualizada. Mas agora existem 3 pessoas modificando essas informações.

Manter a configuração manual do software TCP/IP em redes grandes é algo impraticável: requer muito tempo dos administradores, além de ser uma prática bastante vulnerável a erros.

Para complicar ainda mais a situação, o presidente da empresa está incentivando os funcionários a levarem um notebook pessoal para o trabalho. Assim, eles podem continuar em casa o que estavam fazendo na empresa. João vai precisar gastar

[1] Esta é a configuração básica. Mas configurações adicionais podem ser feitas, como, por exemplo, informar o nome de domínio da máquina e quem é o servidor WINS.

um endereço IP para cada notebook? E o que pode ser ainda pior: quando o usuário do notebook for para uma outra sub-rede[2], João ou outro membro da equipe terá de ser chamado para reconfigurar o software TCP/IP do notebook.

A solução para todos os problemas de João e de sua equipe é simplesmente passar a utilizar o serviço DHCP. O DHCP – *Dynamic Host Configuration Protocol* – é um protocolo de comunicação que permite que administradores de rede gerenciem de forma central e automática a atribuição das configurações do software TCP/IP das máquinas de uma rede. O serviço DHCP permite que os administradores supervisionem a distribuição dos endereços IP de um ponto central (a partir do servidor DHCP), e quando as máquinas forem levadas para outro local, serão passadas as configurações adequadas.

9.2 Visão geral do DHCP

O serviço DHCP apresenta, como a maioria dos serviços das redes TCP/IP, uma arquitetura cliente/servidor. Nessa arquitetura, as máquinas que precisam ser configuradas são chamadas de clientes DHCP e as máquinas que oferecem as configurações são chamadas de servidores DHCP. Para que uma máquina tenha suas configurações de rede obtidas dinamicamente, basta que ela seja definida como um cliente DHCP. O serviço DHCP está associado à porta UDP 67 do lado servidor e UDP 68 do lado cliente.

Suponha que um novo cliente DHCP será inserido na rede. Quando a máquina é ligada, ela envia para todas as outras máquinas da mesma rede física onde ela está a seguinte pergunta: "Meu endereço MAC é M, eu preciso me comunicar na rede, quem poderá informar minhas configurações?". Quando o servidor DHCP ouve essa pergunta ele responde: "Que tal o endereço IP X?". Então o cliente diz: "Acho que esse endereço está legal! E quais são minhas outras configurações?". Finalmente o servidor responde: "Diante disso, seu endereço IP será X, seu roteador default será R, sua máscara de rede será Z e o servidor de nomes que lhe serve é N. Eu lhe concedo essas configurações por T dias."

Esse foi apenas um exemplo superficial e personificado de como se dá a negociação entre o servidor e o cliente DHCP. A conversa entre eles pode tomar rumos diferentes do apresentado. O servidor DHCP pode, por exemplo, não responder ao cliente por não possuir mais endereços disponíveis. Ou, devido a um erro de configuração[3], o servidor pode oferecer ao cliente um endereço que já existe na rede. Alguns clientes, ao receber um endereço do servidor, fazem uma verificação e percebem quando o servidor lhes passou um IP duplicado, negando o endereço oferecido. Na seção a seguir veremos mais detalhes de como funciona o protocolo DHCP.

---

[2] Em outra sub-rede as configurações de rede são outras.

[3] Alguém configurou manualmente uma máquina com um endereço IP que estava no conjunto de endereços a serem oferecidos pelo servidor.

DHCP e BOOTP

O DHCP é uma extensão de um protocolo mais antigo, chamado BOOTP (Bootstrap Protocol). O BOOTP, assim como o DHCP, é capaz de configurar todo o software TCP/IP de um cliente quando é iniciado. A equipe de gerência da rede precisa configurar o protocolo BOOTP informando associações pré-fixadas de endereços físicos com endereços lógicos (IP, por exemplo). Cada vez que uma nova máquina participar da rede, os administradores precisam acrescentar o endereço físico dessa máquina e seu respectivo IP na configuração do BOOTP. Com o crescimento e maior dinamismo das redes, o BOOTP passou a não ser escalável e apropriado como mecanismo de configuração dinâmica da rede em máquinas clientes.

O DHCP, como extensão do BOOTP, é capaz de oferecer todas as funcionalidades do BOOTP e outras novas. Com o DHCP, novas máquinas podem "entrar" na rede sem que os administradores tenham de modificar quaisquer configurações. O DHCP herdou do BOOTP a mesma porta de transporte (UDP 67 para servidores e 68 para clientes) e o mesmo formato de mensagens.

9.3 Como o DHCP funciona?

O DHCP usa o conceito de *aluguel*. O endereço IP oferecido a um cliente é válido durante um certo período de tempo. Chamamos esse período de **tempo de concessão**. O tempo de concessão é um valor configurado por nós, administradores do serviço DHCP. Geralmente, usamos tempos de configuração maiores (de dezenas de meses, por exemplo) quando a quantidade de endereços IP é sempre maior que a quantidade de máquinas ativas na rede e as máquinas não são móveis. Em casos onde máquinas entram e saem da rede com freqüência, costumamos escolher um tempo de concessão menor, de algumas unidades de horas. No exemplo da seção anterior o tempo de concessão era T dias.

Além do tempo de concessão, configuramos também no servidor as **faixas de endereços** que ele poderá oferecer aos clientes DHCP e a máscara de rede. João, por exemplo, poderia configurar o seu servidor para oferecer aos clientes a faixa de endereços que vai de 192.168.1.1 a 192.168.1.200 e a máscara de rede 255.255.255.0. As configurações de rede adicionais também são inseridas pelo administrador: roteador default e servidores de nomes, por exemplo.

A Figura A9-1, baseada em [DHCP-CISCO], mostra, ainda de forma simplificada, como o serviço DHCP funciona.

Quando um cliente é ligado/reiniciado, ele envia um quadro de difusão contendo uma mensagem DHCPDISCOVER. Se ele já foi servido por um servidor DHCP, pode ter armazenado localmente o último endereço IP recebido e o tempo de concessão. Nesse caso, ele pode sugerir na mensagem DHCPDISCOVER o mesmo endereço IP que lhe fora concedido outrora. Com isso, quando o número de endere-

ços é suficiente, é possível que clientes DHCP possam permanecer com o mesmo endereço IP durante muito tempo.

Se o servidor estiver no mesmo domínio de difusão que o cliente, ele imediatamente responderá com uma mensagem DHCPOFFER.[4] É possível que existam vários servidores e que todos respondam ao cliente. Na mensagem DHCPOFFER, o servidor oferece um endereço IP disponível. O cliente responde ao servidor com uma mensagem DHCPREQUEST. Com essa mensagem, o cliente solicita formalmente o endereço anteriormente oferecido pelo servidor. Se o cliente receber mensagens DHCPOFFER de mais de um servidor, ele terá de selecionar o servidor que utilizará. Em geral, o cliente escolhe quem primeiro lhe responde.

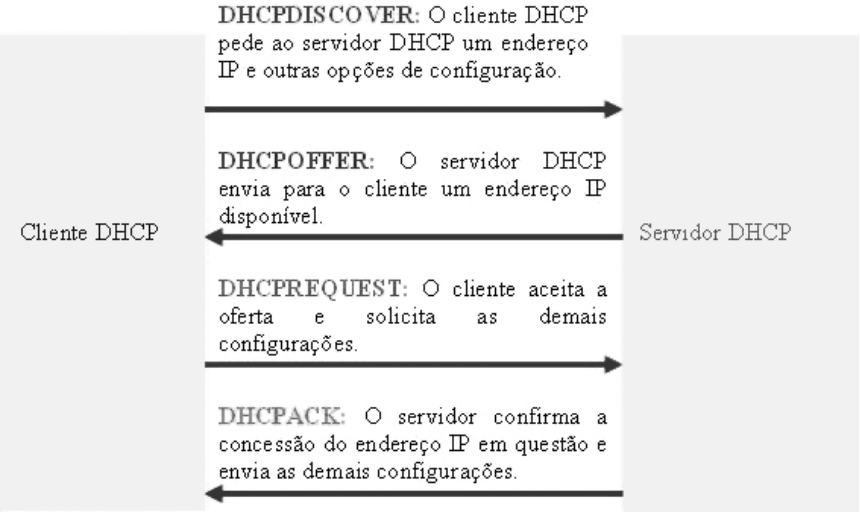


FIGURA A9-1: Diagrama de uma sessão DHCP típica.

Na mensagem DHCPREQUEST, o cliente informa explicitamente o endereço IP prestes a lhe ser concedido. Quando o servidor selecionado recebe a mensagem DHCPREQUEST, ele responde com uma mensagem DHCPACK ou uma mensagem DHCPNAK. No primeiro caso, o servidor está confirmando que concederá ao cliente o endereço IP em questão e informa ainda todas as outras configurações de rede associadas ao escopo[5] em questão. O servidor retornará uma mensagem DHCPNAK quando não for capaz de atender à requisição do cliente. Por exemplo, o endereço de rede em questão foi concedido a outro cliente. Nesse caso, o cliente reinicia o processo.

Os clientes DHCP disponíveis nos sistemas operacionais mais novos checam o endereço recebido para assegurar que não é um IP já atribuído a outra máquina. Se o

[4] Se o servidor estiver em outro domínio de difusão será necessária a utilização de um agente de repasse DHCP – veja Seção 9.6.
[5] Veja Seção 9.5.

cliente detectar que o endereço IP em questão já está configurado em outra máquina, ele envia para o servidor uma mensagem DHCPDECLINE e reinicia o processo.

Antes de ser desligado, o cliente DHCP envia para o servidor uma mensagem DHCPRELEASE para liberar seu endereço IP.

Mas o que acontece quando o tempo de concessão (T) expira? Em condições normais, esse tempo não chega a expirar. Sempre que se passa a metade do tempo de concessão, o cliente DHCP fala com o servidor para renovar seu aluguel. No tempo T/2 o cliente envia diretamente para o servidor uma mensagem DHCPREQUEST com o seu endereço IP. O servidor então responde ao cliente com uma mensagem DHCPACK que contém um novo tempo de concessão. Dessa forma, o cliente readquire o direito de continuar utilizando o endereço IP que lhe foi passado por mais tempo. Caso o cliente não consiga falar com o servidor em T/2, ele tentará novamente renovar seu aluguel mais tarde. Veja mais detalhes em [RFC2131, COMER].

9.4 Mensagens DHCP

As mensagens DHCP têm um formato próprio, com campos bem conhecidos. A Figura A9-2 apresenta o formato de uma mensagem DHCP. Entre parênteses indicamos o tamanho (em bytes) de cada campo. Na RFC 2131 [RFC2131], onde o protocolo DHCP é definido, você encontrará o significado de cada um de seus campos. Trataremos aqui apenas de alguns campos, que serão mencionados em seções futuras ou que consideramos mais interessantes.

<table>
<tr><td>op (1)</td><td>Htype (1)</td><td>hlen (1)</td><td>hops (1)</td></tr>
<tr><td colspan="4">xid(4)</td></tr>
<tr><td colspan="2">secs (2)</td><td colspan="2">flags (2)</td></tr>
<tr><td colspan="4">ciaddr (4)</td></tr>
<tr><td colspan="4">yiaddr (4)</td></tr>
<tr><td colspan="4">siaddr (4)</td></tr>
<tr><td colspan="4">giaddr (4)</td></tr>
<tr><td colspan="4">chaddr (16)</td></tr>
<tr><td colspan="4">sname (64)</td></tr>
<tr><td colspan="4">file (128)</td></tr>
<tr><td colspan="4">options (variável)</td></tr>
</table>

FIGURA A9-2: Formato das mensagens DHCP.

Quando um cliente DHCP já recebeu um determinado endereço IP do servidor, esse endereço fica armazenado em uma cache, mesmo depois de ter sido liberado pelo cliente. Dessa forma, quando o cliente novamente participar da rede, solicita ao servidor o mesmo endereço anteriormente concedido a ele. Quando o cliente DHCP lembra-se de seu endereço anteriormente concedido[6], ele preenche o campo *ciaddr* com esse endereço. O servidor DHCP informa o endereço IP oferecido ao cliente através do campo *yiaddr*.

Quando uma mensagem DHCP é recebida para retransmissão por um agente de repasse, o campo *giaddr* é alterado. O agente de repasse preenche esse campo com seu endereço IP. Caso a mensagem seja repassada diretamente do cliente para o servidor, o campo *giaddr* tem valor zero.

Através do campo opções, o servidor comunica ao cliente as informações necessárias para que ele participe da rede. Em geral, cada opção define uma informação. Por exemplo, a opção *subnet mask* indica a máscara de rede que o cliente deve usar. Na Figura A9-3 encontra-se uma mensagem DHCPACK capturada com o Sniffer Pro v3.5 da Network Associates. Nessa figura, destacamos o endereço IP que o servidor envia para o cliente e algumas opções de configuração.

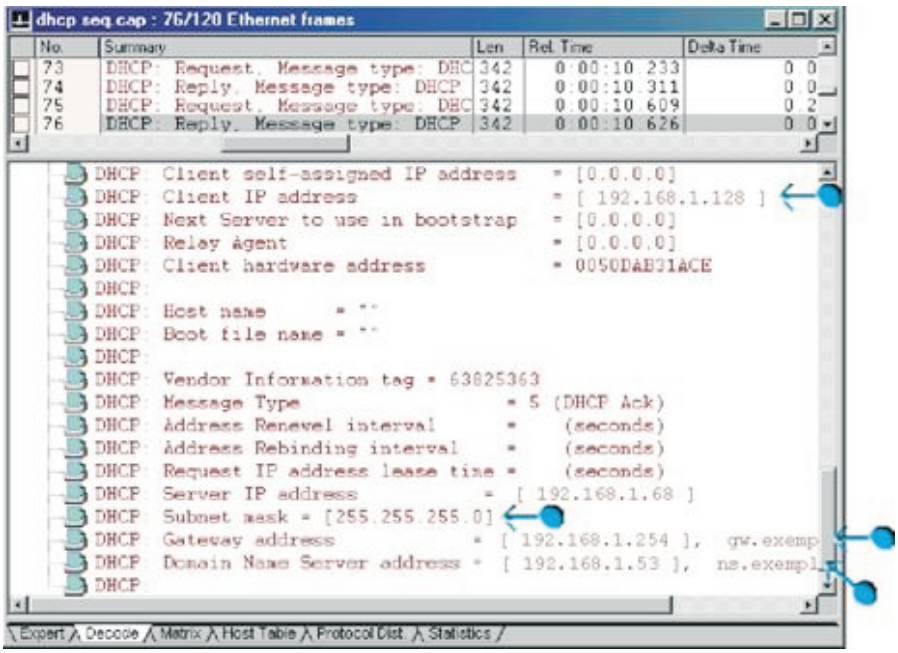


FIGURA A9-3: Mensagem DHCPACK capturada com Sniffer da network Associates.

[6] Isto pode ser após ter liberado sua concessão, como falamos anteriormente, ou ao tentar renová-la.

9.5 Escopos DHCP

O serviço DHCP, assim como outros serviços da rede, depende do envio de quadros de difusão. Então você poderia se perguntar: deve existir um servidor DHCP para cada sub-rede? A resposta é: não, graças à possibilidade de usar um agente de repasse DHCP (ou agente de repasse BOOTP, como é chamado na especificação do protocolo [RFC2131]). Mas para que um servidor DHCP possa servir a várias sub-redes distintas é necessário que ele esteja devidamente configurado para tal. Apenas na próxima seção veremos como agentes de repasse funcionam em conjunto com vários escopos de forma que seja possível existir apenas um servidor DHCP oferecendo configurações TCP/IP para clientes em diferentes sub-redes.

Máquinas em sub-redes diferentes devem ter prefixos de rede diferentes. Veja a rede apresentada na Figura A9-4. A máquina dm1 deve ter prefixo de rede 192.168.1.0 e máscara de rede 255.255.255.0, enquanto a máquina df1 deve ter prefixo 192.168.2.0 e máscara 255.255.255.0. Para o servidor DHCP dessa figura, devemos configurar no mínimo dois escopos: um para as máquinas da sub-rede do Departamento de Marketing e outro para as máquinas do Departamento de Finanças.

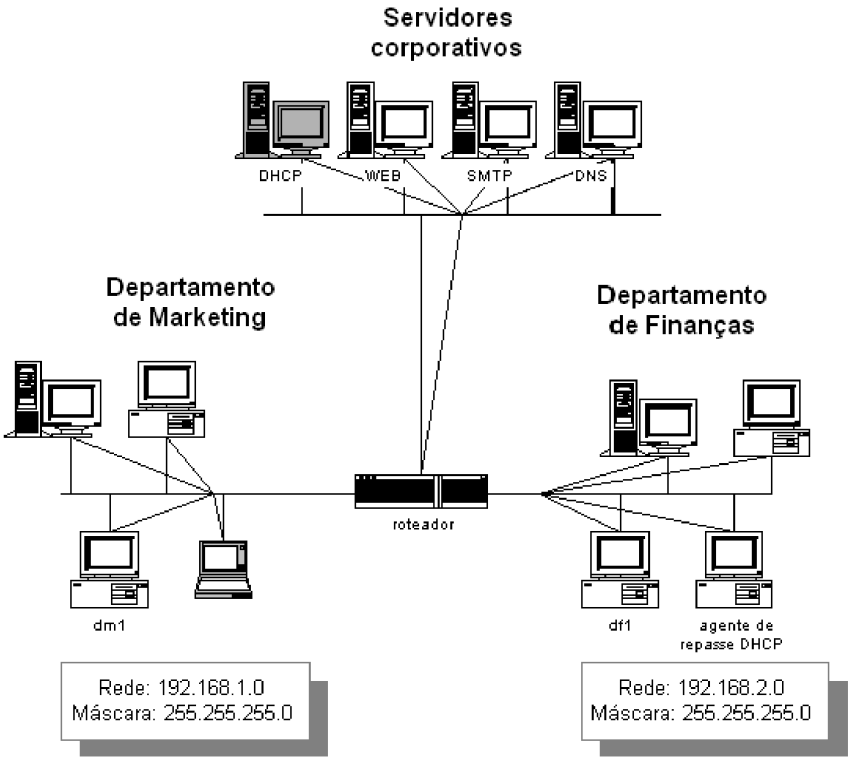


FIGURA A9-4: Um único servidor DHCP para várias sub-redes.

Um escopo DHCP é uma coleção de informações de configuração TCP/IP que definem os parâmetros que serão usados para configurar todos os clientes DHCP de uma determinada sub-rede. Deve existir no servidor um escopo definido para cada sub-rede onde existem clientes DHCP deste servidor. Os escopos definem as seguintes informações:

- **Faixa de endereços IP** a serem atribuídos aos clientes. Geralmente, definimos o primeiro e o último endereço desta faixa. Além disso, é possível excluir desta faixa alguns endereços que já estão reservados para máquinas com IP fixo (servidores e roteadores, por exemplo). Esses endereços excluídos, apesar de fazerem parte da faixa de endereços IP configurada, não são oferecidos a clientes DHCP;
- **Máscara de rede**. Esta máscara será passada aos clientes e serve para permitir a identificação da rede lógica à qual os endereços IP da faixa de endereços pertencem;
- **Tempo de concessão**. É o período de tempo durante o qual um endereço é cedido para um cliente DHCP;
- **Outras opções de configuração** tais como endereço do roteador default e dos servidores de nomes dos clientes DHCP do escopo.

9.6 O agente de repasse

Um agente de repasse DHCP é um hospedeiro ou roteador que repassa mensagens DHCP de clientes para servidores DHCP. É muito simples configurar um agente de repasse. Só é preciso saber o(s) endereço(s) IP(s) do(s) servidor(es) DHCP para o qual as mensagens dos clientes DHCP serão repassadas.

Quando um agente de repasse recebe uma mensagem DHCPDISCOVER, ele a retransmite para os servidores DHCP que conhece. Antes de repassar a mensagem, ele coloca seu endereço no campo *giaddr* da mensagem.

Quando o servidor DHCP recebe a mensagem, ele verifica o campo *giaddr*. Quando esse campo é zero, o servidor DHCP sabe que a máquina que enviou o DHCPDISCOVER está na mesma rede local que ele. Então ele envia o DHCPOFFER para o endereço de difusão, uma vez que a máquina não possui ainda endereço IP configurado. Caso o campo *giaddr* não seja zero, o servidor DHCP envia o DHCPOFFER para o endereço definido nesse campo, que é o endereço do agente de repasse.

Considerando a Figura A9-4, vejamos a seguir como o agente de repasse DHCP funciona:

1. O cliente df1 envia uma mensagem DHCPDISCOVER para todas as máquinas da sub-rede do Departamento de Finanças. Essa mensagem destina-se à porta UDP 67 das máquinas da sub-rede;
2. O agente de repasse, assim como as demais máquinas da rede, recebe essa mensagem. No entanto, apenas o agente de repasse processará a mensagem.

Ele a examinará, preencherá o campo *giaddr* com o endereço da interface na qual recebeu a solicitação DHCP e a repassará para os servidores DHCP que conhece;

3. Quando o servidor DHCP recebe uma mensagem DHCPDISCOVER, ele observa o valor do campo *giaddr*. Quando esse campo tem valor diferente de zero, o servidor sabe que a mensagem lhe foi repassada por um agente de repasse;
4. Caso o servidor DHCP tenha muitos escopos configurados, o próprio endereço do campo *giaddr* guiará o servidor para que ele escolha o escopo correto. Por exemplo, se o endereço do agente de repasse da Figura A9-4 for 192.168.2.222, o campo *giaddr* da mensagem recebida pelo servidor será 192.168.2.222. O servidor DHCP procurará por um escopo que contenha o endereço do agente de repasse;
5. Uma vez encontrado o escopo adequado, o servidor envia uma mensagem DHCPOFFER diretamente para o agente de repasse, que a repassa para o cliente DHCP. O endereço desse cliente ainda não foi definido, então essa resposta é repassada para o cliente através de um quadro de difusão;
6. De forma semelhante, mensagens DHCPREQUEST são repassadas de clientes para servidores e DHCPACK e DHCPNAK de servidores para clientes.

Lembre-se de que, para que tudo isto funcione, é importante que o endereço das interfaces do agente de repasse a partir das quais requisições DHCP podem ser recebidas e as respectivas máscaras de rede estejam na faixa de endereços configurada no escopo DHCP do servidor.

9.7 Referências

[COMER]	Comer, Douglas. *Internetworking with TCP/IP: Principles, Protocols, and Architectures*. Prentice Hall, 4a edição.
[DHCP-CISCO]	*Knowledge: Understanding DHCP*. Em `http://www.cisco.com/warp/public/779/smbiz/service/knowledge/tcpip/dhcp.htm`
[RFC2131]	Droms, R. *Dynamic Host Configuration Protocol*. Março, 1997. Em `http://www.ietf.org/rfc/rfc2131.txt?number=2131`

APÊNDICE

10

Neste apêndice mostraremos o que é e como funciona o serviço de nomes de domínio. Infelizmente, informações sobre características mais avançadas não serão encontradas aqui. Para mais informações sobre o serviço de nomes, leia [DNS&BIND, DNS&WIN2000, DNS&WINNT].

10 O serviço de nomes

10.1 Motivação e introdução

Como já sabemos, todas as máquinas da Internet são identificadas por um endereço IP único. Mas já imaginou como seria referenciar essas máquinas com base unicamente em seu endereço IP? Para visitar a página Web da sua loja de CDs predileta você teria de decorar o endereço IP do servidor Web desta loja. Teria de decorar o IP do banco, do servidor de correio eletrônico e de tantos outros servidores que nos prestam serviços. Numa rede do tamanho da Internet é impraticável identificar cada máquina apenas por seu endereço IP. Numa LAN com algumas dezenas de máquinas, isso já poderia se tornar um complicador.

Para solucionar esse problema existem os nomes de domínio. Em vez de identificarmos as máquinas única e exclusivamente via seus endereços IPs, – que são numéricos – podemos identificá-las também através de seus nomes de domínio. A existência dos nomes de domínio não exclui a necessidade dos endereços IPs. Os nomes de domínio existem principalmente pelo fato de nós, seres humanos, nos sentirmos mais confortáveis em memorizar nomes que números.

Quando tentamos estabelecer conexão com uma máquina através de seu nome, a primeira providência a ser tomada é a tradução do nome no IP da máquina. Vejamos um exemplo: suponha que você queira visitar a página sediada por www.exemplo.com.br. Você abre o seu navegador e digita o nome do servidor Web. O navegador precisará estabelecer uma conexão com o servidor www.exemplo.com.br. Mas, para tal, o endereço IP de www.exemplo.com.br precisa ser des-

coberto. Quem realiza o mapeamento de nomes de domínio em endereços IPs são servidores que chamamos de servidores de nomes, ou servidores DNS. Um servidor de nomes conhecido deve ser consultado sobre qual o endereço IP corresponde ao nome www.exemplo.com.br e esse servidor retornará o endereço IP correto para que, então, o cliente Web possa estabelecer uma conexão com o servidor Web.

O serviço de nomes de domínio é composto por um grandioso banco de dados distribuído por todo o mundo que mantém nomes e endereços IPs das máquinas registradas na Internet. Esses bancos de dados são tipicamente sistemas Unix-like que executam o software Berkeley Internet Name Domain (BIND).[1]

O serviço de nomes é responsável por dois tipos de mapeamentos diferentes: o mapeamento de um nome de domínio em um endereço IP, que chamamos de resolução direta e o mapeamento de um endereço IP em um nome de domínio, que chamamos de resolução reversa. A resolução direta é bem mais utilizada, pois é mais comum conhecermos o nome de domínio de uma máquina que o seu IP, mas a resolução reversa é importante também. Por questões de segurança, muitos servidores, ao receberem uma solicitação de estabelecimento de conexão realizam uma consulta reversa de nomes para descobrir o nome do cliente e em seguida, com o nome descoberto, realiza uma consulta direta. A conexão só é estabelecida se as duas consultas oferecerem resultados compatíveis. Na Seção 10.4 mostramos como uma consulta direta é realizada e na Seção 10.6 falaremos mais sobre a resolução reversa de nomes.

Uma idéia simplista do projeto do serviço de nomes seria manter em cada servidor de nomes tabelas com informações sobre o mapeamento nome de domínio ⇔ IP. Obviamente, nas proporções alcançadas pela Internet (com milhões de máquinas conectadas), esta solução não seria escalável. Então imaginou-se um ambiente colaborativo, onde cada servidor de nomes é responsável por manter sob sua responsabilidade informações sobre uma certa parte dos mapeamentos. Assim, todos colaboram entre si para que quaisquer nomes (caso existam) possam ser mapeados.

A idéia é muito elegante: organizar a Internet em Domínios Administrativos e criar um banco de dados hierárquico distribuído, onde exista um ou mais servidores em cada nível da hierarquia, responsáveis por fornecer informações sobre nomes que se situam abaixo desse ponto da hierarquia (apenas um nível). Para entendermos o serviço de nomes é imprescindível que entendamos como essa hierarquia de nomes (que também chamamos espaço de endereçamento) está organizada. Isso será feito na próxima seção.

---

[1] BIND é ainda hoje a implementação mais utilizada do serviço de nomes. No entanto, o aumento da utilização de máquinas Windows como servidores vem causando o uso mais freqüente da implementação Microsoft do serviço de nomes.

10.2 O espaço de nomes de domínio

A estrutura hierárquica de nomes de domínios é apresentada na Figura A10-1. A raiz da árvore é representada por uma cadeia de caracteres vazia (""). Muitas implementações do DNS representam a raiz por um ponto ("."). Todos os outros nós da árvore têm um rótulo, que é uma cadeia de caracteres que pode ter até 63 caracteres. A raiz não sabe quem são todos os nós da árvore, mas conhece todos os seus filhos imediatos. Para sabermos o nome absoluto de um nó basta caminhar na árvore na direção folhas → raiz, separando os rótulos encontrados no caminho por pontos ("."). Por exemplo, o nome absoluto do nó campus é campus.com.br.

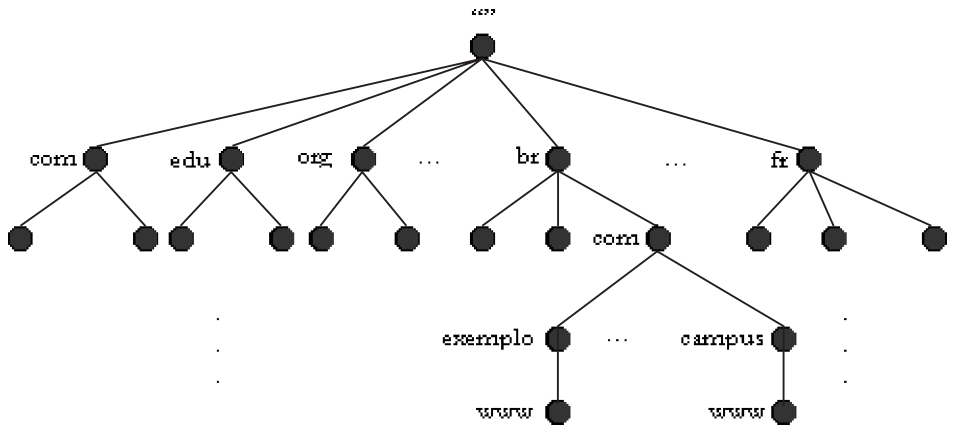


FIGURA A10-1: A estrutura hierárquica do espaço de nomes de domínio.

É comum chamarmos nomes absolutos de FQDN, que é a sigla de *Fully Qualified Domain Name*. FQDNs são nomes de nós da árvore construídos a partir da raiz. Existem também nomes relativos, escritos em relação a algum nó da árvore que não seja a raiz. Observando a figura podemos dizer que o nome relativo de exemplo com relação a br é exemplo.com. Os nós irmãos devem ter rótulos diferentes. Isso garante que todos os nomes absolutos de todos os nós sejam únicos.

Um domínio é simplesmente uma subárvore do espaço de nomes de domínio. O domínio br, por exemplo, é toda a subárvore para a qual o nó br é a raiz. Um domínio pode conter outros domínios (que são comumente chamados subdomínios ou zonas). Por exemplo, campus.com.br faz parte dos domínios br e com.br. As folhas da árvore de um domínio representam nomes de máquinas deste domínio. Nós intermediários também podem representar, além do nome do domínio, nomes de máquinas. Por exemplo, campus.com.br pode ser o FQDN do domínio, mas pode também ser um nome atribuído às máquinas servidoras Web da Editora Campus.

É comum classificarmos um domínio por níveis de acordo com sua posição no espaço de nomes de domínio. Chamamos os filhos da raiz de domínios de alto nível (*top level domains*) ou domínios de primeiro nível. Os domínios edu, com e br, por

exemplo, são domínios de primeiro nível. Os filhos dos domínios de alto nível são domínios de segundo nível e assim por diante. Campus.com.br, por exemplo, é um domínio de terceiro nível.

Os domínios de alto nível podem ser genéricos ou representantes de países. Dentre os domínios genéricos encontramos edu (de instituições educacionais), com (comercial) e gov (governo federal dos Estados Unidos). Br e fr são exemplos de domínios de países.

10.3 Servidores de nomes e resolvedores

Servidores de nomes armazenam informações sobre alguma parte do espaço de nomes de domínio (uma ou mais zonas), que são obtidas a partir de um arquivo local ou de um outro servidor de nomes. Esses servidores são máquinas que conhecem um pouco da estrutura hierárquica de nomes de domínio e podem, portanto, ajudar na descoberta de um mapeamento nome ⇔ IP ou IP ⇔ nome.

Os servidores de nomes executam uma implementação do DNS. A mais utilizada atualmente é o BIND. Nessa implementação, o processo responsável pela resolução de nomes chama-se `named`. Na implementação Microsoft o processo servidor de nomes chama-se `dns.exe`.

Para que um nome de domínio exista na Internet é necessário que:

1. O domínio seja devidamente cadastrado no registro apropriado. Todos os domínios na Internet com extensão .br, por exemplo, são registrados, exclusivamente, no Registro .br acessado através do endereço http://registro.br;

2. Pelo menos uma máquina seja configurada para ser o servidor de nomes primário do domínio e outra para ser o servidor de nomes secundário.[2]

Um servidor de nomes de um domínio não precisa necessariamente conhecer todas as máquinas de um domínio. Ele pode simplesmente conhecer quem são os servidores de seus subdomínios e, estes, sim, respondem por máquinas dos subdomínios.

Existem três tipos de servidores de nomes: primários, secundários e servidores só de armazenamento, sendo os dois primeiros os mais utilizados. Os servidores de nomes primários lêem dados de configuração a partir de arquivos locais. Já os servidores secundários obtêm seus dados de configuração a partir de outro servidor de nomes (em geral um servidor primário, mas não obrigatoriamente). Quando um servidor secundário obtém dados sobre uma zona (ou domínio) a partir de um servidor principal, dizemos que ocorreu uma transferência de zona. É no servidor de nomes primário que realizamos toda a configuração diretamente. Os servidores de nomes secundários são espelhos dos servidores de nomes primários. Assim, só precisamos realizar modificações em um servidor – o primário.

[2] Caso este requisito não esteja sendo atendido, você nem poderá cadastrar o seu domínio no registro.

É claro que uma pequena minoria das máquinas da Internet são servidoras de nomes. Então você poderia perguntar: e como conseguimos descobrir o IP relacionado a um determinado nome? A resposta é simples: as máquinas que não são servidoras de nomes são clientes DNS. O lado cliente DNS é chamado resolvedor (*resolver*). A única configuração necessária para o resolvedor é indicar pelo menos um endereço IP de um servidor DNS que aceite ser o servidor de nomes deste cliente.[3] O resolvedor se comunica com o servidor DNS que ele conhece para descobrir mapeamentos nome ⇔ IP e IP ⇔ nome. O resolvedor é responsável pelas seguintes tarefas:

- Consultar um servidor de nomes;
- Interpretar as respostas recebidas do servidor;
- Retornar a informação descoberta para o programa que a solicitou (pode ser, por exemplo, o seu navegador).

10.4 Como o DNS funciona?

Suponha que você está usando a máquina pc-1.exemplo.com.br. O resolvedor dessa máquina conhece o servidor de nomes ns.exemplo.com.br, cujo IP é 192.168.1.53. Você quer visitar a página da Editora Campus e digita em seu navegador o nome www.campus.com.br. Quando você confirma que realmente deseja visitar o endereço digitado (teclando *Enter*, por exemplo) o navegador solicita ao resolvedor o endereço IP de www.campus.com.br. O resolvedor se comunica com ns.exemplo.com.br. Esse servidor de nomes não sabe realizar esse mapeamento para o resolvedor imediatamente, pois ele tem apenas informações sobre seus filhos imediatos e www.campus.com.br não é um deles. O que faz ns.exemplo.com.br? Recorre à raiz da árvore. Todos os servidores de nomes devem conhecer pelo menos uma raiz, caso contrário, ele só poderá decidir sobre nomes que estão um nível abaixo dele.

O servidor ns.exemplo.com.br pergunta à raiz qual o IP de www.campus.com.br. Pela mesma razão que ns.exemplo.com.br não soube responder essa questão imediatamente, a raiz também não o saberá. A raiz informa a ns.exemplo[4] que não sabe quem é www.campus.com.br, mas sabe quem é ns.br[5], seu filho imediato, que está mais próximo na árvore do nome procurado. Ao receber essa resposta, ns.exemplo pergunta a ns.br: qual o IP de www.campus.com.br? Mais uma vez, a resposta não poderá ser dada imediatamente. Mas ns.br informa a ns.exemplo que sabe quem é ns.com.br. Assim, ns.exemplo pergunta a ns.com.br qual é o IP correspondente ao nome www.campus.com.br. O servidor ns.com.br não sabe a resposta, mas indica

[3] Veremos no fim deste apêndice que por questões de segurança podemos configurar o servidor para aceitar realizar consultas apenas para clientes que pertençam a uma certa faixa de endereços IP.
[4] Chamaremos ns.exemplo.com.br de ns.exemplo daqui por diante.
[5] Usamos ns para nomear uma máquina que é servidor de nomes de um domínio. ns.br, por exemplo, é servidor de nomes do domínio br.

quem é ns.campus.com.br. Quando ns.exemplo pergunta a ns.campus.com.br qual o IP de www.campus.com.br, ns.campus.com.br responde que o IP é 200.219.182.70. O servidor ns.campus.com.br sabe essa resposta, pois www.campus.com.br é filho imediato de campus.com.br no espaço de nomes de domínio. A Figura A10-2 ilustra esta pesquisa.

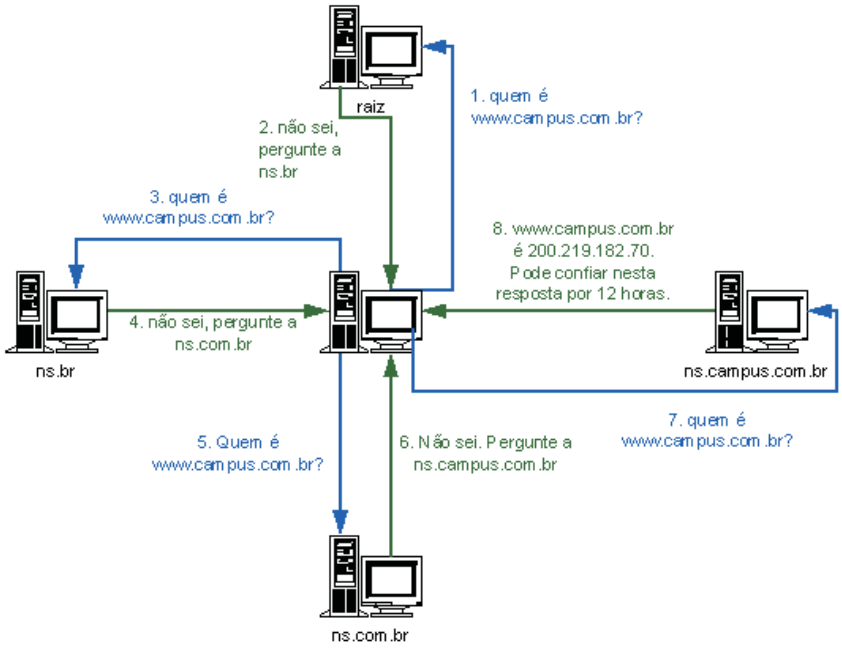


FIGURA A10-2: Resolução de www.campus.com.br na Internet.

Ao receber a resposta de ns.campus.com.br, ns.exemplo vai armazená-la em uma cache local por 12 horas. Assim, nas próximas 12 horas, se algum outro cliente DNS de ns.exemplo quiser visitar www.campus.com.br, não será mais preciso consultar servidor algum.

Chamamos esse modo de resolução apresentado de recursivo. No modo recursivo, um servidor, ao receber uma consulta, contata quantos servidores forem necessário para descobrir o mapeamento desejado (ou descobrir que o nome consultado não existe). Existe um outro modo de resolução, chamado iterativo, em que o servidor, ao receber uma consulta, responde com a melhor resposta já conhecida para quem solicitou a consulta, seja esta uma resposta final ou o endereço de um outro servidor. Note que ns.exemplo.com.br realizou uma pesquisa no modo recursivo, pois ele recebeu do resolvedor uma consulta recursiva. Ao consultar outros servidores com o objetivo de resolver essa consulta, ns.exemplo enviou consultas iterativas. Por essa razão, os demais servidores que participaram do processo apenas informaram um outro servidor mais próximo ou, finalmente, a resolução do nome.

10.5 Configuração dos servidores de nomes

Ao contrário dos resolvedores, os servidores de nomes precisam ter uma configuração um pouco mais elaborada. Primeiramente, ele precisa conhecer pelo menos uma raiz. Na árvore apresentada na Figura A10-1, encontramos apenas uma raiz, mas, na realidade, para que o serviço não tenha um ponto único de falha e tenha escala existem várias raízes. As raízes são consultadas quase sempre que um resolvedor pergunta algo a um servidor. Elas, portanto, participam diretamente de quase todas as resoluções de nomes, sendo peças fundamentais do serviço de nomes de domínio.

A primeira configuração de um servidor DNS é, portanto, ensiná-lo como se comunicar com as raízes. Na implementação BIND, é comum chamarmos de named.root o arquivo que traz informações sobre as raízes. Não precisamos decorar o endereço das raízes. Quando obtemos o BIND, esse arquivo já vem pronto para nós. Além disso, é fácil obtê-lo na Internet.

Além de conhecer as raízes, precisamos configurar no servidor quem são todos os seus filhos imediatos. Suponha que você esteja configurando o servidor de nomes do domínio exemplo.com.br. Você precisa configurar no servidor quem são todos os filhos imediatos de exemplo.com.br. Dentre esses filhos podemos encontrar máquinas finais e outros domínios (subdomínios). Considere que o domínio exemplo.com.br tem 5 máquinas clientes (pc-1 a pc-5), dois servidores de nomes, um servidor Web e outro de correio eletrônico. Além dessas máquinas, existem dois subdomínios: livros e cds. Na Figura A10-3 apresentamos um arquivo chamado named.zone através do qual ensinamos ao servidor de nomes em questão quem são seus filhos imediatos.

```
;; Arquivo de configuração de zona de exemplo.com.br
$TTL 43200

@   IN   SOA   ns.exemplo.com.br.   root.exemplo.com.br.  (
                    200201231   ; Serial
                    8h          ; Refresh - 8H
                    2h          ; Retry - 2H
                    2w          ; Expire - 2 Sem.
                    2h)         ; Minimum TTL - 2H

        IN   NS   ns.exemplo.com.br.
        IN   A    192.168.1.53
        IN   MX   10    192.168.1.25
        IN   NS   ns2.exemplo.com.br.
```

FIGURA A10-3: Arquivo named.zone do domínio exemplo.com.br.

```
pc-1      IN   A    192.168.1.1
          IN   MX   10   mail
pc-2      IN   A    192.168.1.2
          IN   MX   10   mail
pc-3      IN   A    192.168.1.3
          IN   MX   10   mail
pc-4      IN   A    192.168.1.4
          IN   MX   10   mail
pc-5      IN   A    192.168.1.5
          IN   MX   10   mail

mail      IN   A    192.168.1.25
          IN   MX   10   mail

ns        IN   A    192.168.1.53
          IN   MX   10   mail
ns2       IN   A    192.168.1.54
          IN   MX   10   mail

www       IN   A    192.168.1.80
          IN   MX   10   mail

;;   domínios abaixo de exemplo.com.br

livros       IN   NS   ns.livros.exemplo.com.br.
ns.livros    IN   A    192.168.10.53
cds          IN   NS   ns.cds.exemplo.com.br.
ns.cds       IN   A    192.168.20.53
```

FIGURA A10-3: Continuação

Vamos analisar rapidamente esse arquivo. Primeiramente, é importante ressaltar que ele é usado apenas para resolução direta de nomes (dado um nome, queremos descobrir o IP correspondente). Mais adiante falamos mais detalhadamente sobre a resolução reversa.

Na seção anterior, ns.campus.com.br disse a ns.exemplo.com.br para confiar em sua resposta por apenas 12 horas. A este tempo chamamos de TTL (*time to live*) default. O TTL default é configurado através da diretiva `$TTL`. Quando ns.exemplo oferece a outro servidor uma informação sobre seus filhos, esse outro servidor a armazena em uma cache. Mas essa entrada da cache só é válida por no máximo o TTL default oferecido. Assim, se houver alguma modificação na configuração de

ns.exemplo relacionada aos dados que foram configurados, o servidor que armazenou informações na cache só perceberá após no máximo o TTL default ter decorrido. Por essa razão, não é interessante que o TTL default seja muito grande – de mais de um dia, por exemplo.

No registro SOA configuramos:

- **O número de série do arquivo**: este número de série vale como um controle de versão. Um servidor secundário só transfere os dados do servidor principal caso o número de série do arquivo no servidor principal seja maior que o número de série em seu arquivo correspondente;
- **Tempo de renovação (refresh)**: indica de quanto em quanto tempo o servidor precisa se comunicar com o servidor principal em busca de atualizações. Esse tempo só é realmente utilizado por servidores secundários;
- **Tempo de tentar novamente (retry)**: é possível que o secundário não consiga se comunicar com o servidor principal quando necessário. Assim, ele tentará estabelecer comunicação novamente em busca de atualizações após o tempo de *retry* configurado. Esse tempo também só é utilizado por servidores secundários;
- **Tempo de expiração**: caso o servidor secundário não consiga falar com o principal por um tempo maior ou igual ao de expiração, o servidor secundário deixar de responder pelo domínio desatualizado. Mais uma vez, este é um parâmetro usado apenas pelos servidores secundários;
- **TTL mínimo**: é o TTL default para as respostas negativas (quando o servidor responde que o nome pesquisado não existe).

Esse cabeçalho repete-se em todos os arquivos de configuração do DNS, exceto no named.root.

Depois do cabeçalho, damos informações sobre os servidores de nomes. Quem são os servidores de nomes deste domínio (registros `IN NS`), seus endereços IP (registros `IN A`) e seus servidores de correio eletrônico (registros `IN MX`).

Enfim, chegou a hora de dizer quem são os filhos imediatos de exemplo.com.br. Isso é feito através de registros de endereço (`IN A`). A seguinte linha indica que pc-2.exemplo.com.br corresponde ao endereço IP 192.168.1.2:

```
pc-2   IN   A   192.168.1.2
```

Note que após cada máquina existe um registro de correio eletrônico (`IN MX`). Esse registro indica quem é o servidor de correio eletrônico da máquina. Assim, se quisermos enviar um e-mail para maria@pc-2.exemplo.com.br, ns.exemplo saberá indicar quem é o servidor de correio eletrônico que deverá ser contatado. Veremos mais detalhes sobre isso no Apêndice 11.

No fim do arquivo, configuramos dois subdomínios usando o registro `IN NS`. Nesse caso, ns.exemplo está repassando a responsabilidade de responder sobre as zo-

nas livros e cds para ns.livros.exemplo.com.br e ns.cds.exemplo.com.br, respectivamente. Assim, quando ns.exemplo receber alguma pesquisa sobre máquinas do domínio livros.exemplo.com.br e cds.exemplo.com.br, ele informará o endereço do servidor de nomes que realmente possa responder as consultas em questão.

10.6 Como mapear IPs em nomes

No início deste apêndice, dissemos que o DNS é capaz de mapear nomes em IPs e vice-versa. Mas, até o momento, falamos apenas no mapeamento nome → IP, que chamamos mapeamento direto. E o mapeamento reverso, como é feito?

Para realizar essa tarefa, o domínio in-addr.arpa foi criado no espaço de nomes de domínio. Abaixo desse domínio, os números são os rótulos dos nós da árvore. Esse domínio é amplo o bastante para abrigar todos os endereços IPs da Internet. Veja uma ilustração na Figura A10-4.

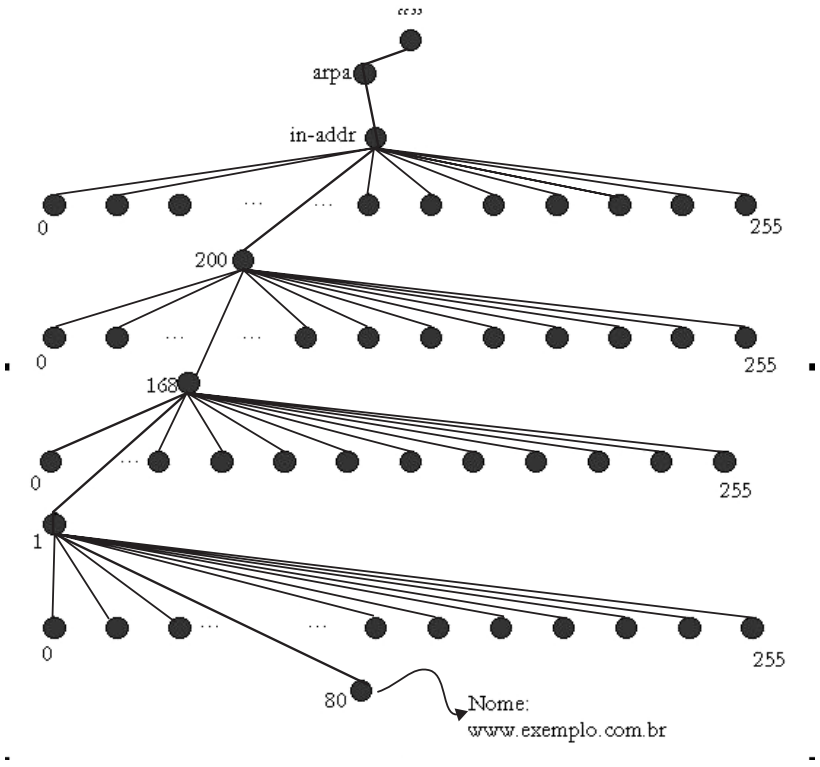


FIGURA A10-4: O domínio in-addr.arpa.

As raízes guardam informações sobre servidores que podem responder a endereços IP existentes. Assim, as pesquisas são realizadas de forma semelhante às pesquisas diretas, no entanto, a raiz já indica um servidor que tenha autoridade para

responder sobre o endereço consultado. Em outras palavras, a raiz sabe informações sobre o que está abaixo de in-addr.arpa, apesar de não serem seus filhos imediatos. Para descobrir o nome correspondente a 200.168.1.80 pergunta-se à raiz quem é 80.1.168.200.in-addr.arpa? A raiz diz que não sabe, mas sabe quem responde por 1.168.200.in-addr.arpa. Esse servidor é, então, consultado e retorna a resposta correta: www.exemplo.com.br.

Quando um servidor recebe uma consulta de mapeamento reverso, ele busca por informações configuradas através do registro `IN PTR`. No BIND um outro arquivo, que costumamos chamar named.rev, traz informações para realização de mapeamento reverso. Veja na Figura A10-5 um exemplo deste arquivo.

```
;; Arquivo de configuração do mapeamento reverso de exemplo.com.br
$TTL 43200
@   IN   SOA   ns.exemplo.com.br. root.exemplo.com.br. (
                    200201231   ; Serial
                    8h          ; Refresh - 8H
                    2h          ; Retry - 2H
                    2w          ; Expire - 2 Sem.
                    2h)         ; Minimum TTL - 2H

          IN   NS   ns.exemplo.com.br.

1         IN   PTR   pc-1.exemplo.com.br.
2         IN   PTR   pc-2.exemplo.com.br.
3         IN   PTR   pc-3.exemplo.com.br.
4         IN   PTR   pc-4.exemplo.com.br.
5         IN   PTR   pc-5.exemplo.com.br.

25        IN   PTR   mail.exemplo.com.br.

53        IN   PTR   ns.exemplo.com.br.
54        IN   PTR   ns2.exemplo.com.br.
80        IN   PTR   www.exemplo.com.br.
```

FIGURA A10-5: Exemplo de arquivo de configuração de mapeamento reverso.

10.7 O arquivo de configurações gerais

Além dos arquivos de configuração já mencionados, na implementação BIND existe um arquivo de configuração geral do servidor de nomes. Este arquivo é o /etc/named.conf. No named.conf, dentre outras informações, dizemos as zonas (ou domí-

nios) sobre as quais o servidor de nomes tem autoridade para responder. Nesse arquivo, ligamos os domínios aos arquivos de configuração. Veja na Figura A10-6 um exemplo desse arquivo.

```
#Arquivo de configuração de ns.exemplo.com.br
options {
    directory "/var/named";
    pid-file "/var/named/named.pid";
};
zone "." {
    type hint;
    file "named.ca";
};
zone "exemplo.com.br"{
    type master;
    file "named.zone";
};
zone "1.168.192.in-addr.arpa"{
    type master;
    file "named.rev";
};
zone "0.0.127.in-addr.arpa"{
    type master;
    file "named.local";
};
```

FIGURA A10-6: O arquivo named.conf - exemplo simples.

Esse arquivo diz exatamente que domínios são representados pelo servidor de nomes e onde estão os arquivos de configuração de cada um deles. Existem muitas outras configurações de mais alto nível. Para mais informações, veja [DNS&BIND].

10.8 Dicas de segurança

O serviço de nomes é um dos mais importantes. Quando ele não está funcionando, muitos outros serviços também ficam indisponíveis, pois são acessados através do nome do servidor. Você precisa dedicar um pouco mais de atenção aos servidores DNS de sua organização para que a probabilidade de uma interrupção desse serviço seja bem pequena.

A primeira dica é: acompanhe sempre as novidades sobre a implementação de seu servidor de nomes. Quando surgirem atualizações, faça-as o mais rapidamente

possível. Procure se inscrever em listas de segurança relacionadas à sua implementação DNS. Se seu servidor DNS é uma implementação BIND, por exemplo, inscreva-se na lista de segurança do BIND em *http://www.isc.org/services/public/lists/bind-lists.html* para estar sempre atualizado. Novas versões do servidor sempre trazem correções de erros e vulnerabilidades. Continuar com o servidor antigo significa ter um servidor vulnerável, sujeito a ataques.

Caso você esteja usando a implementação BIND, pode usar o arquivo named.conf para tornar o seu servidor de nomes mais seguro:

1. Não permita que qualquer servidor DNS faça transferências de suas zonas. Configure nos seus servidores DNS principais quais os servidores que podem fazer transferência de zona. Na implementação BIND versões 8 e 9, você pode adicionar uma opção global (para todas as zonas configuradas) ou uma opção para cada zona que indique a quem a transferência de zona é permitida:[6]

```
//modo global
options {
      allow-transfer {<DNS1>; <DNS2>; …; <dnsn>; };
};
//para uma zona específica
zone  "exemplo.com.br" {
      type master;
      file "named.zone";
      allow-transfer {<DNS1>; <DNS2>; …; <dnsn>; };
};
```

   Onde: <DNS1>, <DNS>, … <dnsn> são endereços IP dos servidores aos quais a transferência de zona é permitida.

   Nos servidores escravos, proíba completamente a transferência de zonas.

```
//para uma zona específica
zone  "exemplo.com.br" {
      type slave;
      file "bak.exemplo.zone";
      allow-transfer { none; };
};
```

   É importante que apenas servidores realmente autorizados possam realizar transferências de zonas de seu domínio. Quando essa restrição não existe,

[6] Em servidores BIND mais antigos, use a diretiva `xfrnets` para especificar as máquinas que podem fazer transferência de zonas.

atacantes podem realizar uma transferência de zona para obter informações mais íntimas sobre as máquinas e servidores de seu domínio. Com essas informações, ele pode descobrir rapidamente quem são os servidores importantes, descobrir vulnerabilidades e iniciar um ataque. Por exemplo, ele pode descobrir quem é o servidor de correio eletrônico, executar `telnet` para a porta 25 dele e descobrir qual a implementação e versão do servidor SMTP. Depois, ele vai atrás de vulnerabilidades dessa versão do servidor SMTP… Seu servidor, muito provavelmente, será invadido.

3. Não tenha apenas um servidor de nomes para realizar todas as funções do serviço de nomes. Pode-se dizer que o servidor de nomes tem duas funções distintas: resolver nomes para outros servidores de nomes e resolver nomes para clientes DNS. A arquitetura apresentada na Figura 7-2 do Capítulo 7 é um exemplo de como separar estas funcionalidades. Ao separar estas funções torna-se mais fácil proteger os servidores. Por exemplo [DNS&BIND]:

   a. Nos servidores externos, proíba que eles façam consultas solicitadas por clientes DNS. Permita apenas que eles respondam a perguntas iterativas de outros servidores DNS;

   ```
   options {
      recursion no;
   };
   ```

   b. Nos servidores internos, configure quais as faixas de endereços dos clientes DNS;

   ```
   options {
      allow-query {192.168.1/24; };
   };
   ```

10.9 Bibliografia

Albitz, P. Liu, C. *DNS* e *BIND*. Editora Campus, 2001.

APÊNDICE

11

O serviço de correio eletrônico, juntamente com o serviço Web, compõe o time dos serviços mais utilizados na grande maioria das organizações. Esse serviço tem várias similaridades com o serviço de correio normal, mas uma grande diferença: as "cartas" eletrônicas não demoram dias para chegar no destino, independente de qual seja. Neste apêndice exploraremos um pouco esse serviço tão indispensável, seja no trabalho ou na nossa vida pessoal.

11 O serviço de correio eletrônico

11.1 Conceituação

O serviço de correio eletrônico permite a troca de informações (mensagens, documentos etc.) de forma rápida e conveniente entre dois ou mais usuários da Internet. Ele provê comunicação entre dois pontos distintos na rede, mesmo que o destino não esteja ativo no momento do envio da informação.

Na Figura A11-1 mostramos os elementos conceituais de um sistema de correio eletrônico [COMER]. Quando um usuário envia uma mensagem, o sistema a coloca em uma área de armazenamento privativa (que costumamos chamar de fila de mail) junto com a identificação do emissor e receptor, máquina-destino e momento em que a mensagem entrou na fila. O sistema, então, inicia a transferência da mensagem para a máquina remota como uma atividade em background, permitindo ao remetente prosseguir com outras atividades em sua máquina.

O processo que transfere a mensagem em background se torna um cliente. A sua primeira tarefa é utilizar o sistema de nomes de domínio para descobrir o endereço IP da máquina remota para a qual a mensagem deverá ser enviada. Lembram-se dos registros `IN MX` que mostramos no Apêndice 10? São eles que indicam qual máquina é responsável pelo serviço de nomes. O servidor de nomes do domínio a que o destinatário da mensagem pertence será consultado para tal. Após descobrir o endereço IP do servidor de correio eletrônico remoto, o processo de envio de e-mail tentará estabelecer uma conexão TCP com ele. Se a conexão for estabele-

cida, a mensagem é transferida para o servidor remoto, que a armazenará numa área de *spool*. Caso a conexão TCP não possa ser estabelecida, a mensagem ficará na fila, que é processada pelo processo emissor de tempos em tempos (normalmente a cada 30 minutos) e também quando novas mensagens chegam na fila. Quando uma mensagem fica na fila sem que o processo emissor consiga enviá-la por um tempo longo (geralmente 3 ou 4 dias), o processo entregador de mensagens desiste de entregar essa mensagem e envia uma notificação ao remetente da mensagem.

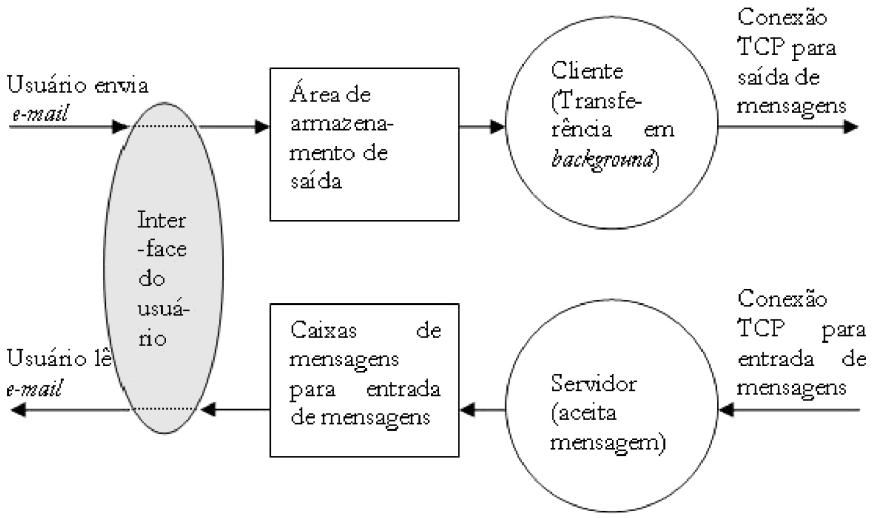


FIGURA A11-1: Componentes conceituais de um sistema de correio eletrônico [Comer].

11.2 Um pouco mais de detalhes

Na seção anterior já apresentamos de forma simplificada o sistema de correio eletrônico. Nesta seção, vamos explicar esse sistema com um pouco mais de detalhes técnicos. A Figura A11-2 é uma representação mais minuciosa dos componentes do sistema de correio eletrônico.

Quando um usuário deseja enviar uma mensagem, ele utiliza um programa que serve de interface entre ele e o sistema de correio eletrônico. As interfaces com o usuário mais comuns no ambiente UNIX são `mail`, `elm`, `pine` e `mailtool`. No ambiente Windows, as interfaces com o usuário mais comuns são Microsoft Outlook Express, Netscape Netmessenger e Eudora. Esses programas estabelecem uma conexão TCP com o servidor de correio eletrônico (um soquete TCP, porta 25). No momento do envio de mensagens a interface com o usuário fala com o servidor de correio o protocolo SMTP (*Simple Mail Transfer Protocol*). O servidor de correio (que representamos na Figura A11-2 como o agente de transporte servidor) mais comum no ambiente UNIX é o programa `sendmail`. Existem muitos outros bons agen-

tes de transporte UNIX, tais como `qmail` e `Postfix`. O agente de transporte mais comum no ambiente Windows é o programa Internet Mail Exchanger da Microsoft.

Após receber a mensagem para envio, o agente de transporte servidor coloca-a na fila de mensagens de saída, conforme explicamos na seção anterior. Em background, o agente de transporte conecta-se (como se fosse um cliente) a um outro agente de transporte: o agente de transporte da máquina remota para a qual a mensagem deve ser entregue. Neste ponto, um agente de transporte só deve aceitar transmitir mensagens caso elas tenham sido enviadas por clientes da organização.

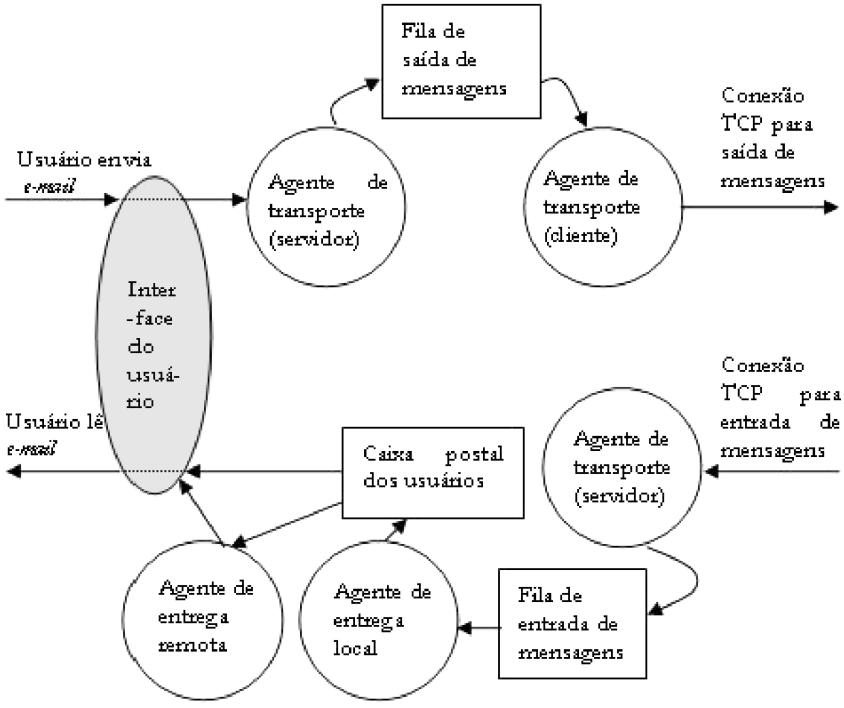


FIGURA A11-2: Esquema conceitual mais detalhado do serviço de correio eletrônico.

Quando o agente de transporte da máquina remota recebe a mensagem, ele a coloca numa fila de entrada. Essa fila é processada de tempos em tempos por um agente de entrega local. Esse agente tem a responsabilidade de retirar a mensagem da fila de entrada e colocá-la na caixa postal do usuário para a qual a mensagem está endereçada. O agente de entrega local varia de acordo com o sistema operacional, sendo comum no ambiente UNIX o `procmail`.

O agente de entrega local põe as mensagens do usuário em um diretório da máquina servidora. Caso os usuários usem uma interface com o usuário na própria máquina servidora (usar `pine` no próprio servidor, por exemplo), não é necessário um agente de entrega remota. No entanto, é bem mais comum atualmente que os

usuários usem uma máquina cliente. Nesse caso, um agente de entrega remota entrará em ação. Esse agente tipicamente transfere as mensagens da caixa postal do usuário no servidor para a máquina cliente do usuário. O agente de entrega remota mais comum é o POP3 (*Post Office Protocol*), existindo também o IMAP (*Internet Mail Access Protocol*).

Como podemos observar, o serviço de correio eletrônico é bastante semelhante ao serviço de correio postal, podendo ser feitas, inclusive, várias analogias entre elementos atuantes no correio postal e no correio eletrônico. Em ambos, temos:

- Nome e endereço do remetente;
- Nome e endereço do destinatário;
- Agente de coleta/exibição de correspondência;
- Agente de despacho e recepção de correspondência;
- Agente de entrega de correspondência.

11.3 Endereços eletrônicos e cabeçalhos

Todos nós já conhecemos o formato padrão de um endereço eletrônico: usuário@domínio ou usuário@máquina.domínio. A primeira forma é preferível porque abstrai o nome da máquina do endereço do usuário (a máquina pode deixar de existir, mas o endereço do usuário continua válido). Mas como decifrar o cabeçalho das mensagens? Podemos descobrir informações importantes analisando-o. Observe o cabeçalho a seguir.

```
From leitor@suacasa.com.br Sat Feb 6 10:15:30 2000
Return-Path: leitor@suacasa.com.br
Received: from mail.suacasa.com.br (200.241.192.254)
by mail.editora.com.br (8.9.5/8.9.5) id AA00123; Sat 6 feb 2000 10:15:00 BSB
Received: from pc1.suacasa.com.br (200.241.192.1)
by mail.suacasa.com.br (8.9.5/8.9.5) id AA00123; Sat 6 feb 2000 09:15:15 BSB
Date: Sat, 6 feb 2000 09:15:10 BSB
From: leitor@suacasa.com.br (Maria)
Message-Id: <200002070915.AA00123@suacasa.com.br>
Subject: curiosidade!
To: duvidas@editora.com.br

QUANDO SAI A SEGUNDA EDIÇÃO DO LIVRO???
```

Nessa mensagem, alguém utilizando o endereço eletrônico leitor@suacasa.com.br enviou uma mensagem para duvidas@editora.com.br. O remetente da mensagem utilizou, para enviar a mensagem, a máquina (200.241.192.1). O agente de

transporte que recebeu a mensagem para repasse foi suacasa.com.br. Esse agente de transporte transmitiu a mensagem para mail.editora.br.

Esse exemplo que apresentamos mostra um cabeçalho bem simples. Ao observar o cabeçalho, é possível identificarmos a máquina cliente utilizada para enviar a mensagem, os servidores de correio eletrônico envolvidos no transporte da mensagem, os momentos exatos em que a mensagem foi enviada pelo remetente, recebida pelo servidor do lado do rementente e, mais tarde, recebida pelo servidor do lado destinatário.

Note que foi possível resolver o endereço da máquina cliente e o nome do servidor mostrou-se do mesmo domínio que o endereço eletrônico do remetente. Em muitos casos (desconfie destes), não é possível resolver o nome do servidor ou da máquina cliente e o endereço eletrônico do remetente pertence a um domínio inexistente ou que possui outro servidor de correio eletrônico.

11.4 Algumas opções de configuração do sendmail

Constantemente, são descobertas vulnerabilidades no `sendmail` e uma nova versão do servidor é disponibilizada. Isso significa que os usuários do `sendmail` precisam estar sempre alertas para as atualizações. Caso contrário, estarão executando uma versão com vulnerabilidades, sujeita a ataques. Apesar disso, o servidor de correio eletrônico (agente de transporte) mais utilizado no mundo atualmente (em 2003) ainda é o `sendmail`. Sendo assim, resolvemos, nesta seção, oferecer algumas dicas de configuração do `sendmail` que podem ser úteis. Mais informações sobre o `sendmail` e suas opções de configuração são encontradas em [SENDMAIL].

Opções de configuração são configuradas no arquivo /etc/mail/sendmail.cf. Elas informam ao `sendmail` muitos parâmetros importantes. Por exemplo, indicam a localização de arquivos-chave, tempos de expiração e definem como o `sendmail` deve agir. As opções devem ser usadas para "sintonizar" o seu `sendmail` de modo que se comporte de acordo com as suas necessidades particulares. O comando `O` é usado para definir opções do `sendmail`.

11.4.1 O diretório da fila

O sendmail armazena em uma fila todos os e-mails a serem transmitidos. O sendmail confia que nessa fila existem apenas e-mails verdadeiros (que realmente foram transmitidos por clientes de correio eletrônico) inseridos na fila pelo superusuário do sistema (root). Portanto, é imprescindível que nesse diretório apenas o `root` tenha permissão de escrita e leitura. Caso contrário, qualquer usuário pode modificar a fila, o que não é bom para você, para sua empresa e para seus usuários. Para ver os e-mails que estão na fila, usamos o comando `mailq`. Para usar esse comando, é necessário estar como o usuário `root`.

No arquivo de configuração do sendmail, a opção QueueDirectory é usada para indicar o diretório onde as mensagens serão enfileiradas. Por default, o diretório é /var/spool/mqueue. A seguinte linha do sendmail.cf configura essa opção:

```
O QueueDirectory= /var/spool/mqueue
```

Exceto se você tiver razões muito fortes para tal, não modifique o diretório da fila de mensagens.

11.4.2 Limitando o tempo de espera na fila

Existem várias opções de tempo de expiração em fila, mas falaremos aqui apenas das duas que consideramos mais críticas. Uma mensagem geralmente fica na fila porque não pôde ser entregue pelo servidor ao ser recebida (o servidor remoto estava fora do ar, por exemplo). A fila será processada de tempos em tempos. Cada vez que ela é processada, há a possibilidade de o servidor conseguir transmiti-la. Quando isso ocorre, é interessante que:

- O remetente da mensagem seja informado de que ela ainda não pôde ser transmitida após um certo tempo;
- uma mensagem não possa ficar indefinidamente na fila; isso poderia causar falta de espaço em disco;
- quando uma mensagem for excluída da fila, o remetente deve ser informado que sua mensagem não chegou no destinatário.

Configuramos esse comportamento do sendmail através de duas opções: Timeout.Queuewarn e Timeout.Queuereturn. O tempo estabelecido por Timeout.Queuewarn indica quanto tempo, depois de a mensagem estar na fila, o remetente é avisado que ela ainda não foi transmitida. Em geral, configuramos essa opção para algumas unidades de horas. Quatro horas nos parece um valor bastante razoável.

A opção Timeout.Queuereturn indica quanto tempo, depois de estar na fila, uma mensagem é excluída da fila e o seu remetente informado de tal fato. Em geral, essa opção é configurada com algumas unidades de dias. Cinco dias é para nós bastante razoável.

Ao estabelecer essas opções é importante levar em consideração a pior interrupção de rede que você possa imaginar em sua empresa. Se, por exemplo, tipicamente as interrupções de rede nunca ultrapassam um dia, cinco dias para Queuereturn pode ser um exagero. Por outro lado, se todos os equipamentos de rede são desligados no fim de semana, configurar essa opção com dois dias pode ser perigoso.

11.4.3 Permissão de arquivos temporários

Em muitos momentos, o sendmail precisa criar arquivos temporários (arquivos no diretório fila, por exemplo). As permissões que são dadas a esses arquivos são deter-

minadas pela opção `TempFileMode`. O valor dessa opção pode variar entre 0600 (arquivos lidos apenas por `sendmail`) e 0666 (arquivos que podem ser lidos por todos). Por questões de segurança, aconselhamos usar o modo de permissão mais restritivo: 0600.

11.4.4 Tamanho máximo das mensagens

É interessante restringir o tamanho máximo das mensagens aceitas para transmissão pelo `sendmail`. Caso esse limite não seja estabelecido, mensagens muito grandes podem encher rapidamente o(s) disco(s) do servidor. A opção `MaxMessageSize` deve ser usada para limitar o tamanho das mensagens. Costumamos limitar o tamanho das mensagens em 3MB, mas cabe a você definir um tamanho adequado para sua organização. Por default, o `sendmail` não estabelece limites para o tamanho das mensagens recebidas.

11.5 Referências bibliográficas

[COMER]	Comer, D. *Internetworking with TCP/IP: Principles, Protocols, and Architectures*. Volume 1. 4ª edição. Prentice Hall, 2000.
[SENDMAIL]	Costales, B. Allman, E. *Sendmail*. 2ª Edição. O'Reilly. Janeiro, 1997.

APÊNDICE

12

12 Introdução a VLANs

Como vimos no Apêndice 4, cada porta de um comutador[1] define um domínio único de colisões. No entanto, o mesmo não ocorre com a difusão. Todas as portas de um comutador participam de um mesmo domínio de difusão. Dessa forma, ao criamos uma rede utilizando comutadores com o equipamento de interconexão de mais alto nível, quando uma estação envia um quadro de difusão, todas as outras o recebem e são obrigadas a processá-lo.

Se o número de máquinas crescer muito, o domínio único de difusão poderá prejudicar o desempenho das máquinas da rede, que terão suas atividades interrompidas constantemente para processar quadros de difusão. Há alguns anos, a solução típica seria introduzir roteadores na rede, que são barreiras naturais de quadros de difusão. Atualmente, existe uma outra solução: configurar redes virtuais nos comutadores.

As redes virtuais (virtual LANs – VLANs) são configuradas em comutadores e têm a funcionalidade de deter o tráfego de difusão. Quando um quadro de difusão é recebido por um comutador onde VLANs estão configuradas, esse quadro é repassado apenas para as portas que pertencem à rede virtual à qual a origem do quadro pertence, e não para todas as outras portas do comutador, como originalmente.

VLANs possibilitam a segmentação da rede não baseada em cabeamento físico. Assim, usuários em ambientes físicos distintos, mas que fazem parte do mesmo grupo de trabalho (e se comunicam freqüentemente entre si, além de acessarem os mesmos servidores), podem participar da mesma VLAN e, portanto, do mesmo domínio de difusão, podendo haver entre eles comunicação direta. Os usuários podem atingir os servidores sem precisar passar por roteadores. Basta que usuários e servidores façam parte da mesma VLAN. Como para alguns tipos de VLANS – como

[1] Aqui estamos referenciando comutadores nível 2.

veremos adiante – é possível que uma máquina participe de mais de uma VLAN, os servidores poderiam fazer parte de todas as VLANs – ou pelo menos de todas as VLANs que abrigam seus clientes – e, assim, eles seriam acessados pelos clientes sem que o tráfego precisasse passar por roteadores.

A melhor definição para VLANs que conhecemos é: **VLANs são domínios de difusão configuráveis** por algum critério. Esses critérios podem ser vários. Este apêndice trata superficialmente de alguns tipos básicos de VLANs. Falaremos também do roteamento entre VLANs.

12.1 Tipos de VLANs

Como já mencionamos na seção anterior, vários são os critérios que podem ser utilizados para se definir VLANs em comutadores. Iniciaremos esta seção tratando de critérios que consideramos mais utilizados atualmente e finalizamos apresentando alguns outros critérios.

12.1.1 VLANs por porta

Este é o tipo de VLANs mais utilizado e o mais simples de configurar também. Por outro lado, este é o critério menos flexível e poderoso. Mais adiante veremos o porquê. Ao utilizar esse tipo de VLAN, simplesmente dizemos quais portas do comutador pertencem a quais VLANs. Vejamos um exemplo para que esse conceito fique mais claro.

Suponha que você acabou de comprar um comutador com 24 portas e gostaria de conectar nesse comutador máquinas do Departamento de Marketing e máquina do Departamento de Vendas. Normalmente, a configuração default de um comutador é que exista uma única VLAN (a VLAN 1) definida por portas, da qual todas as portas do comutador participam. Mas, você não gostaria que os quadros de difusão de um departamento fossem vistos pelas máquinas do outro. Você pode, então, configurar o comutador para que as portas 1-12 pertençam à VLAN 1 (do Departamento de Marketing) e as demais portas pertençam à VLAN 2 (do Departamento de Vendas). Com isso, quando uma máquina do Departamento de marketing enviar um quadro de difusão, ele seria visto apenas pelas máquinas deste departamento. Veja o resultado na Figura A12-1.

Inicialmente, uma VLAN podia abranger apenas um comutador, como exemplificamos na Figura 12-1. No entanto, implementações de VLANs mais novas nos permitem configurar VLANs que abrangem vários comutadores. Vejamos mais um exemplo: você tem 2 comutadores de 12 portas (comutador1 e comutador2). Usuários do Departamento de Marketing e de Vendas precisam ser conectados em ambos os comutadores. Você pode criar uma VLAN 1 que acolhe as portas 1-3 e 6-8 do comutador1 e 5-12 do comutador2, onde usuários do Departamento de Marketing serão conectados. As demais portas de ambos os comutadores participariam da

VLAN 2 e nelas serão conectados os usuários do Departamento de Vendas. Este cenário é ilustrado na Figura A12-2.

Quando VLANs abrangem múltiplos comutadores, é necessário que eles troquem informações entre si. Os comutadores 1 e 2 estão ligados entre si através de um enlace que chamamos de tronco, no qual passam dados de ambas as VLANs.

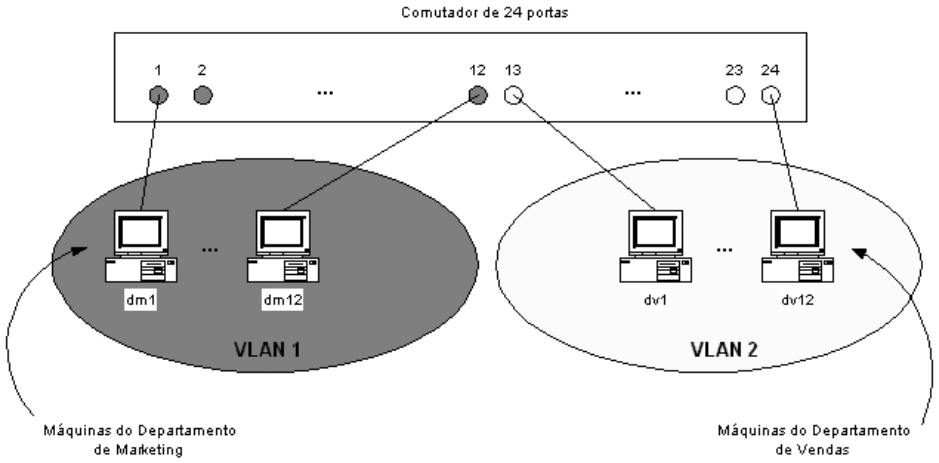


FIGURA A12-1: VLANs definidas por portas.

Quando definimos VLANs por portas, limitamos uma porta a participar de apenas uma VLAN. Em outras palavras, ao definir VLANs por portas, não podemos configurar uma porta para participar de mais de uma VLAN. Essa é uma limitação desse tipo de VLAN mais simples de ser configurado e mantido. Nesse ambiente de VLANs por portas, se tivermos o intuito de conectar um repetidor com várias máquinas a uma porta do comutador, todas as máquinas participarão de uma mesma VLAN.

Portanto, se existirem equipamentos (servidores corporativos e roteadores, por exemplo) que necessitem participar de mais de uma VLAN, outro tipo de VLAN deverá ser definido. Comutadores mais baratos oferecem apenas esse tipo de VLAN.

VLANs por porta nos permitem a segmentação da rede de acordo com o cabeamento físico. Esse tipo de segmentação é pouco flexível, dificultando operações de Acréscimos, Movimentações, Mudanças (AMM) de equipamentos. Se um usuário tem IP fixo e precisa se locomover com sua máquina, um novo IP precisará ser configurado a cada mudança. E pior: suponha que a máquina desse usuário, lotado no Departamento de Marketing, seja um cliente DHCP. Ele poderá se locomover na empresa como desejar, sem necessidade de reconfigurações. No entanto, cada vez que ele mudar de local, ele passará a participar de uma sub-rede lógica diferente (terá endereço IP de uma sub-rede que não é a de Marketing) e poderá não ter acesso a certos serviços que só são permitidos aos usuários do Departamento de Marketing. Configurar VLANs por porta é bastante simples. No entanto, esse tipo de VLAN não oferece a flexibilidade necessária para facilitar AMMs.

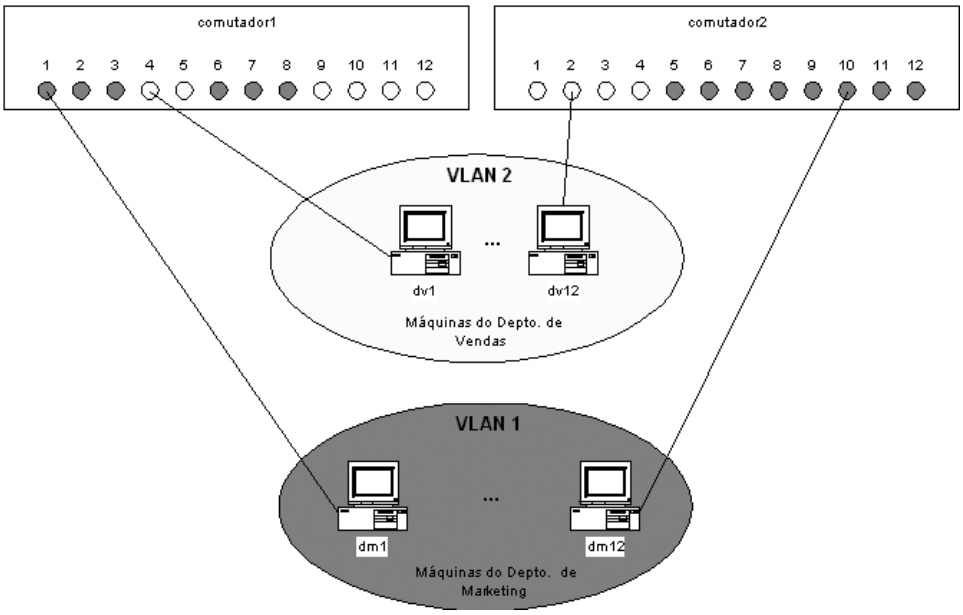


FIGURA A12-2: VLANs definidas por grupos de portas que abrangem múltiplos comutadores.

12.1.2 VLANs por MAC

Podemos, também, definir VLANs por endereço MAC. Em vez de dizer quais portas participam de quais VLANs, dizemos quais endereços MAC participam de quais VLANs. Esse tipo de VLAN é bastante trabalhoso de configurar e, muitas vezes, de manter. Inicialmente, devemos inserir cada máquina em pelo menos uma VLAN. Mais tarde, cada vez que uma nova máquina for adicionada na rede, é necessário inserir seu MAC na VLAN correta.

VLANs baseadas em endereço MAC permitem-nos mover estações de trabalho para uma localização física diferente na rede e, ainda assim, a estação continuará pertencendo à mesma VLAN original. Dessa forma, VLANs definidas por endereço MAC podem ser pensadas com um tipo de VLAN baseada em usuário – já que ela segue o usuário aonde ele for.

Veja um exemplo de VLAN definida por MAC na Figura 12-3. Note que existem usuários com computadores portáteis (00:ab:43:90:30:9e e 09:34:eb:44:12:5e). Não importa em que comutadores eles se conectam dentro da rede corporativa, eles sempre farão parte da mesma VLAN.

12.1.3 VLANs por endereço lógico

Um terceiro tipo de VLAN leva em consideração o endereço de rede dos equipamentos. Essas VLANs são configuradas exatamente como roteadores e segmentação física. Cada segmento físico recebe seu endereço de sub-rede.

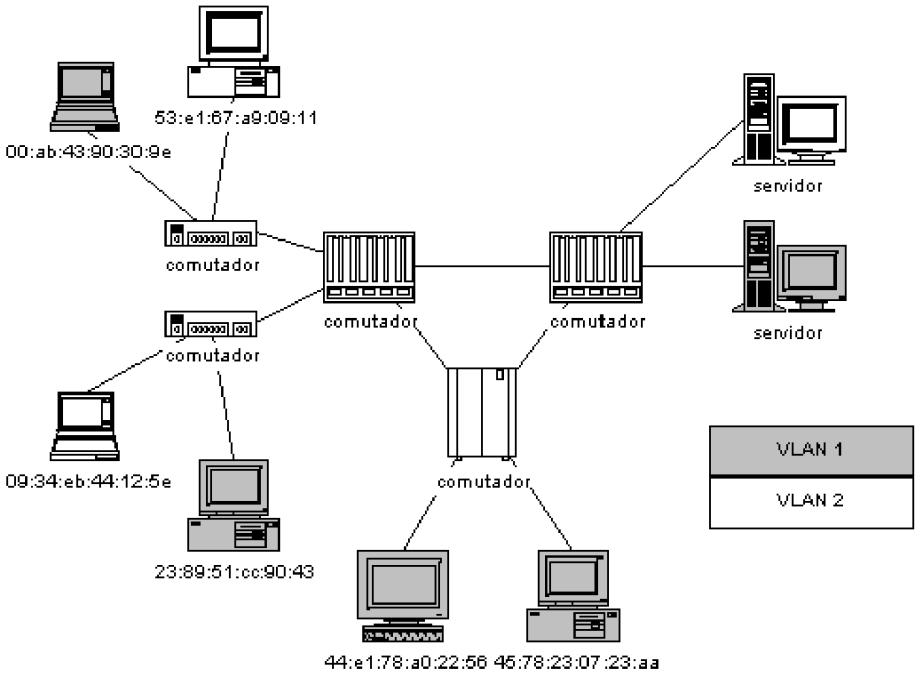


FIGURA A12-3: VLANs definidas por endereço MAC.

Por exemplo, podemos configurar em um comutador duas VLANs: a VLAN 1, que abrange equipamentos cujo endereço IP pertencem à sub-rede 192.168.1.0/24 e a VLAN 2, que abrange as máquinas que participam da sub-rede 192.168.2.0/24. Veja um exemplo de VLAN definida por endereço lógico na Figura A12-4.

Uma grande vantagem desse tipo de VLAN é que usuários podem se locomover livremente e continuarem a participar da sua VLAN original. Além disso, não precisam ser trocadas informações sobre VLANs entre comutadores diminuindo o custo de se manter estas VLANs. A grande desvantagem é que os comutadores terão de abrir e analisar pacotes de rede, o que, obviamente, é uma tarefa muito mais cara do que a tarefa original de apenas analisar quadros (nível 2).

É importante ressaltar que, apesar de o comutador analisar informações da camada de rede, nenhuma decisão de roteamento está sendo tomada. Isto é, o comutador não está agindo necessariamente como um roteador. Como veremos adiante, qualquer que seja o tipo de VLAN definida, a função de roteamento é necessária para que membros de VLANs distintas possam se comunicar[2] e muitos comutadores – os chamados comutadores de nível 3 – sejam capazes de realizar o roteamento entre VLANs.

[2] Falaremos mais sobre roteamento entre VLANs na Seção 12.3.

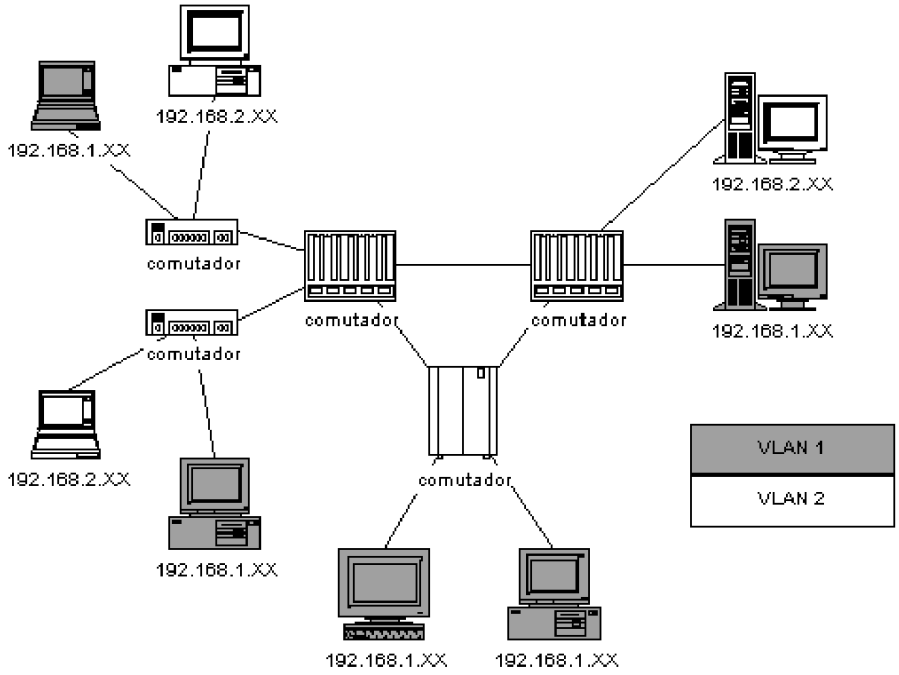


FIGURA A12-4: VLAN baseada em endereços lógicos.

12.2 Outros tipos de VLANs

Além dos três tipos de VLANs já citados, existem outros. Dentre eles encontramos:

- **VLANs baseadas em protocolos** → com este tipo de VLAN podemos separar máquinas que conversam protocolos de rede diferentes em VLANs distintas. Fazemos, então, a separação de NetBeui, DECNET, IP, IPX etc. Estas VLANs são extremamente simples de configurar;
- **VLAN baseada em endereços de multicast** → este tipo de VLAN é criado dinamicamente pela escuta de pacotes de IGMP (*Internet Group Management Protocol*);
- **VLANS baseadas em políticas gerais** → são VLANs formadas pela combinação de quaisquer informações que apareçam no quadro (endereço MAC, endereço de nível 3, tipo de protocolo etc.). Este tipo de VLAN é muito flexível;
- **VLANS baseadas em autenticação de usuários** → com este tipo de VLAN poderíamos formar, por exemplo, uma VLAN com todos os usuários do Departamento de Finanças. Mesmo que um usuário saia de seu departamento e use uma máquina em outro, ao autenticar-se (efetuar logon na rede) ele faria parte da VLAN de seu departamento. Este tipo de VLAN usa um servidor de autenticação que depois autoriza a entrada desse equipamento na VLAN.

12.3 Roteamento entre VLANs

Quando duas máquinas participam de VLANs diferentes, mesmo que estejam conectadas ao mesmo comutador, elas não podem mais realizar uma entrega direta de pacotes entre elas. A razão é bem simples. Antes, quando ambas as máquinas faziam parte da mesma VLAN, elas participavam do mesmo domínio de difusão. Assim, uma requisição ARP (ver Apêndice 7), que é um quadro de difusão, sempre atingia a outra máquina. Mas, quando elas são separadas em VLANs distintas, uma requisição ARP de uma máquina não atinge mais a outra máquina, pois elas não mais compartilham o mesmo domínio de difusão. Sendo assim, para cruzar VLANs, temos de usar roteamento.

O **próprio comutador pode fazer o roteamento** de nível 3. Os chamados comutadores nível 3 ou *brouters* incluem roteador embutido para os protocolos IP e IPX. Uma outra saída é usar **roteadores externos de apenas um braço** (*one-arm routers*). Estes roteadores só são usados por causa de suas tabelas de roteamento e não por causa de suas múltiplas interfaces como seria o caso normal para cruzar segmentos físicos.

O fato de muitos comutadores serem capazes de realizar o roteamento em nível de rede não significa que podemos aposentar os roteadores. Eles ainda são muito utilizados por várias razões. Dentre elas, citamos: (a) aproveitar bons roteadores existentes, (b) roteadores são mais conhecidos pelos administradores de redes, facilitando a configuração e a operação dos mesmos, (c) roteadores possuem portas de rede de longa distância – comutadores nível 3 normalmente só possuem portas de rede local, (d) roteadores dão suporte a múltiplos protocolos de rede (IP, X25, IPX etc.) e de roteamento dinâmico (RIP I/II, OSPF, BGP etc.) – comutadores nível 3 normalmente somente dão suporte a poucos protocolos de rede (IP e IPX) e de roteamento dinâmico (RIP I/II e OSPF).

12.4 Bibliografia

[VLAN-REPORT]	The Virtual LAN Technology Report. `http://www.3com.com/other/pdfs/solutions/en_US/20037401.pdf`
[INTEL_VLANS]	Virtual LANs: Flexible network segmentation for high-speed LANs `http://www.intel.com/network/connectivity/resources/doc_library/tech_brief/virtual_lans.pdf`
[CISCO-VLAN-TRUNKS]	Configuring Ethernet VLAN Trunks. `http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sft_6_1/configgd/e_trunk.htm`