4

A subcamada de controle de acesso ao meio

Muitos protocolos de comunicação da camada de enlace, que estudamos no Capítulo 3, contam com um meio de comunicação por broadcast para transmitir dados. Qualquer um daqueles protocolos requer mecanismos adicionais para permitir que vários transmissores compartilhem o meio de transmissão de forma eficiente e justa.

Em qualquer rede de broadcast, a questão fundamental é determinar quem tem direito de usar o canal quando há uma disputa por ele. Para tornar essa questão mais clara, considere uma chamada de conferência, na qual seis pessoas, em seis telefones, estão conectadas entre si, de forma que cada uma pode ouvir e falar com todas as outras. É muito provável que, quando uma delas parar de falar, duas ou mais comecem a falar ao mesmo tempo, causando confusão. Em uma reunião face a face, a confusão é evitada por meios externos. Por exemplo, as pessoas levantam a mão para pedir permissão para falar. Quando apenas um canal está disponível, determinar quem deve ser o próximo a falar é muito mais difícil. Existem vários protocolos destinados a solucionar o problema, e eles formam o conteúdo deste capítulo. Na literatura, os canais de broadcast às vezes são referidos como **canais de multiacesso** ou **canais de acesso aleatório**.

Os protocolos usados para determinar quem será o próximo em um canal de multiacesso pertencem a uma subcamada da camada de enlace de dados, chamada **MAC** (**Medium Access Control**). Ela é especialmente importante em LANs, particularmente nas sem fio, pois o wireless é naturalmente um canal de broadcast. Alguns aspectos de uma WAN (p. ex., uma interconexão direta) utilizam enlaces ponto a ponto; outros (p. ex., uma rede de acesso compartilhada em um ISP de cabo) são compartilhados e também contam com a camada MAC para facilitar o compartilhamento. Como os canais de multiacesso têm uma relação muito íntima com as LANs, neste capítulo trataremos das LANs em geral, bem como de algumas questões que não fazem parte estritamente da subcamada MAC, mas o assunto principal será o controle do canal.

Tecnicamente, a subcamada MAC é a parte inferior da camada de enlace de dados e, portanto, deveríamos tê-la estudado antes de analisar todos os protocolos ponto a ponto apresentados no Capítulo 3. No entanto, para a maioria das pessoas, é mais fácil entender os protocolos que envolvem várias partes após protocolos de duas partes serem bem compreendidos. Por essa razão, nos desviamos um pouco da ordem de apresentação estritamente de baixo para cima.

4.1 O PROBLEMA DA ALOCAÇÃO DE CANAIS

O tema central deste capítulo é definir como alocar um único canal de broadcast entre usuários concorrentes. O canal poderia ser uma parte do espectro sem fio em uma região geográfica, ou um fio isolado ou fibra óptica ao qual vários nós são conectados. Isso não importa. Nos dois casos, o canal conecta cada usuário a todos os outros e qualquer usuário que faz uso completo do canal interfere na utilização que os outros também fazem dele.

Analisaremos primeiro as limitações dos esquemas de alocação estáticos para o tráfego em rajada. Depois, mostraremos as principais premissas usadas para modelar os esquemas dinâmicos que examinaremos nas próximas seções.

4.1.1 Alocação estática de canais

A maneira tradicional de alocar um único canal, como um tronco telefônico, entre vários usuários concorrentes é dividir sua capacidade usando um dos esquemas de multiplexação que descrevemos na Seção 2.4.4, como FDM (Frequency Division Multiplexing). Se existem N usuários, a largura de banda é dividida em N partes do mesmo tamanho, e a cada usuário será atribuída uma parte. Como cada usuário tem uma banda de frequência particular, não há interferência entre eles. Quando existe apenas um número pequeno e constante de usuários, cada um dos quais com um fluxo constante ou uma carga de tráfego pesada, essa divisão é um mecanismo de alocação simples e eficiente. Um exemplo de uso sem fio são as estações de rádio FM. Cada estação recebe uma parte da banda de FM e a utiliza, na maior parte do tempo, para transmitir seu sinal.

No entanto, quando o número de transmissores é grande e continuamente variável, ou quando o tráfego ocorre em rajadas, a FDM apresenta alguns problemas. Se o espectro for dividido em N regiões, e menos de N usuários estiverem interessados em estabelecer comunicação no momento, grande parte do valioso espectro será desperdiçada e, se mais de N usuários quiserem se comunicar, alguns deles terão o acesso negado por falta de largura de banda, mesmo que alguns dos usuários aos quais uma banda de frequência foi alocada raramente transmitam ou recebam dados.

Contudo, mesmo supondo que o número de usuários poderia ser, de algum modo, mantido constante em N, a divisão de um único canal disponível em subcanais estáticos revela uma ineficiência inerente. O problema básico é que, quando alguns usuários ficam inativos, sua largura de banda é simplesmente perdida. Eles não estão utilizando essa largura de banda, e ninguém mais pode fazê-lo. Uma alocação estática não é apropriada para a maioria dos sistemas de computadores em que o tráfego de dados ocorre em rajadas (são comuns relações de 1.000:1 entre o tráfego de pico e o tráfego médio). Em consequência disso, a maioria dos canais permanecerá ociosa na maior parte do tempo.

O fraco desempenho da FDM estática pode ser facilmente visto com um simples cálculo da teoria do enfileiramento. Vamos começar com o atraso de tempo médio, T, para um canal com capacidade C bps. Consideramos que os quadros chegam aleatoriamente, com uma taxa de chegada de λ quadros/s. O comprimento de cada quadro varia, com um comprimento médio de $1/\mu$ bits. Com esses parâmetros, a taxa de serviço do canal é μC quadros/s. Pela teoria do enfileiramento, o resultado é

$$T = \frac{1}{\mu C - \lambda}$$

(Para os curiosos, esse resultado é para uma fila "M/M/1". Ele requer que a aleatoriedade dos tempos entre as chegadas e os comprimentos de quadro siga uma distribuição exponencial ou, de modo equivalente, obedeça a uma série de Poisson.)

Em nosso exemplo, se C é 100 Mbps, o comprimento do quadro médio, $1/\mu$, é 10.000 bits e a taxa de chegada de quadros, λ, é 5.000 quadros/s, então $T = 200$ μs. Observe que, se ignorarmos o atraso de enfileiramento e simplesmente perguntarmos quanto tempo é necessário para enviar um quadro de 10.000 bits em uma rede de 100 Mbps, obteremos a resposta (incorreta) de 100 μs. Esse resultado só é válido quando não há disputa pelo canal.

Agora, vamos dividir o único canal em N subcanais independentes, cada um com capacidade de C/N bps. A taxa média de entrada em cada um dos subcanais será, agora, λ/N. Ao recalcularmos T, obteremos:

$$T_N = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT$$

O atraso médio para o canal dividido é N vezes pior do que seria se todos os quadros estivessem, de alguma forma mágica, distribuídos de maneira ordenada em uma grande fila central. Esse mesmo resultado explica por que um banco cheio de caixas eletrônicos funciona melhor com uma única fila alimentando todas as máquinas do que uma fila separada à frente de cada máquina.

Os mesmos argumentos que se aplicam à FDM também se aplicam a outras formas de dividir o canal estaticamente. Se usássemos a multiplexação por divisão de tempo, ou TDM (Time Division Multiplexing), e alocássemos cada usuário a cada n-ésimo slot de tempo, e ainda se um usuário não usar o slot alocado, este será simplesmente desperdiçado. O mesmo é válido se dividirmos as redes fisicamente. Usando mais uma vez nosso exemplo anterior, se substituíssemos a rede de 100 Mbps por dez redes de 10 Mbps e fizéssemos a alocação estática de cada usuário a uma delas, o atraso médio saltaria de 200 μs para 2 ms.

Como nenhum dos métodos estáticos tradicionais de alocação de canais funciona bem com um tráfego em rajada, agora trataremos dos métodos dinâmicos.

4.1.2 Premissas para a alocação dinâmica de canais

Antes de começarmos a descrever o primeiro dos muitos métodos de alocação de canais a serem discutidos neste capítulo, vale a pena formular cuidadosamente o problema da alocação. Existem cinco premissas fundamentais subjacentes a todo trabalho realizado nessa área, que serão descritas a seguir.

1. **Tráfego independente**. O modelo consiste em N **estações** independentes (computadores, telefones), cada qual com um programa ou usuário que gera quadros para transmissão. O número esperado de quadros gerados em um intervalo de duração Δt é $\lambda\Delta t$, onde λ é

uma constante (a taxa de chegada de novos quadros). Uma vez gerado um quadro, a estação é bloqueada e nada faz até que o quadro tenha sido transmitido com êxito.

2. **Premissa de canal único**. Um único canal está disponível para todas as comunicações – todas as estações podem transmitir e receber por ele. As estações são consideradas igualmente capazes, embora os protocolos possam atribuir diferentes papéis (p. ex., prioridades) a elas.
3. **Colisões observáveis**. Se dois quadros são transmitidos simultaneamente, eles se sobrepõem no tempo, e o sinal resultante é adulterado. Esse evento é denominado **colisão**. Todas as estações podem detectar colisões. Um quadro que tenha sofrido colisão terá de ser retransmitido posteriormente. Não há outros erros além dos gerados por colisões.
4. **Tempo contínuo ou segmentado (slotted)**. O tempo pode ser considerado contínuo, caso em que a transmissão do quadro pode começar a qualquer instante. Como alternativa, o tempo pode ser segmentado ou dividido em intervalos discretos (slots). As transmissões de quadros sempre começam no início de um slot. Um slot pode conter 0, 1 ou mais quadros, correspondentes a um slot ocioso, a uma transmissão bem-sucedida ou a uma colisão, respectivamente.
5. **Detecção de portadora (carrier sense) ou sem detecção de portadora**. Com a premissa de detecção de portadora, as estações conseguem detectar se o canal está sendo usado antes de tentar utilizá-lo. Se for detectado que o canal está ocupado, nenhuma estação tentará usá-lo até que ele fique livre. Se não houver detecção de portadora, as estações não conseguem detectar o canal antes de tentar utilizá-lo. Elas simplesmente vão em frente e transmitem. Somente mais tarde conseguem determinar se a transmissão foi ou não bem-sucedida.

Ainda é necessário discutir essas premissas um pouco mais. A primeira diz que as chegadas de quadro são independentes, entre estações ou em uma estação específica, e que eles são gerados de modo imprevisível, mas a uma taxa constante. Na realidade, essa premissa não é um modelo de tráfego de rede particularmente bom, pois sabemos que os pacotes chegam em rajadas em intervalos escalonados de tempo (Paxson e Floyd, 1995). Pesquisas recentes confirmam que o padrão ainda existe (Fontugne et al. 2017). Apesar disso, **modelos de Poisson**, como normalmente são chamados, são úteis em parte porque são matematicamente tratáveis. Eles nos ajudam a analisar protocolos para entender, em linhas gerais, como o desempenho muda ao longo de um intervalo de operação e como ele se compara com outros projetos.

A premissa de canal único é o núcleo do modelo. Não existem formas externas de comunicação. As estações não podem levantar as mãos para solicitar que o professor lhes permita falar, de modo que precisaremos de soluções melhores.

As três premissas restantes dependem da engenharia do sistema, e diremos quais delas são mantidas quando examinarmos um protocolo em particular.

A premissa de colisão também é básica. As estações precisam de um modo para detectar colisões se tiverem de transmitir quadros em vez de deixar que se percam. Para canais com fio, o hardware do nó pode ser projetado para detectar colisões quando elas ocorrerem. As estações podem, então, terminar suas transmissões prematuramente para evitar desperdiçar capacidade. Essa detecção é muito mais difícil para canais sem fio, de modo que as colisões normalmente são deduzidas após o fato, pela falta de um quadro de confirmação esperado. Também é possível que alguns quadros envolvidos em uma colisão sejam recebidos com sucesso, dependendo dos detalhes dos sinais e do hardware de recepção. Contudo, essa situação não é o caso comum, de modo que iremos supor que todos os quadros envolvidos em uma colisão são perdidos. Também veremos protocolos projetados para impedir colisões em primeiro lugar.

O objetivo das duas premissas alternativas em relação ao tempo é que o tempo segmentado pode ser usado para melhorar o desempenho. Todavia, é preciso que as estações sigam um relógio mestre ou sincronizem suas ações entre si para dividir o tempo em intervalos distintos. Logo, ele nem sempre está disponível. Discutiremos e analisaremos os sistemas com os dois tipos de tempo. Para determinado sistema, somente um deles é válido.

Da mesma forma, uma rede pode ter a detecção de portadora ou não. Em geral, as redes com fio têm detecção de portadora. No entanto, as redes sem fio não podem usá-la de forma efetiva, porque nem toda estação está dentro do alcance de rádio das outras. De modo semelhante, a detecção de portadora não estará disponível em outros ambientes nos quais uma estação não pode se comunicar diretamente com outras estações, por exemplo, um modem a cabo no qual as estações precisam se comunicar pelo headend do cabo. Observe que a palavra "portadora" (*carrier*), nesse sentido, se refere a um sinal pelo canal, e não tem nenhuma relação com as operadoras comuns (p. ex., as empresas de telefonia) que remontam à época do Pony Express.

Para evitar qualquer mal-entendido, vale a pena observar que nenhum protocolo de multiacesso garante entrega confiável. Mesmo na ausência de colisões, o receptor pode ter copiado parte do quadro incorretamente por diversos motivos. Outras partes da camada de enlace ou de camadas superiores oferecem confiabilidade.

4.2 PROTOCOLOS DE ACESSO MÚLTIPLO

Existem muitos algoritmos conhecidos para alocar um canal de acesso múltiplo. Nas seções a seguir, estudaremos uma pequena amostra dos mais interessantes e apresentaremos alguns exemplos práticos de sua utilização.

4.2.1 ALOHA

A história do nosso primeiro protocolo de acesso múltiplo, ou MAC, começa no Havaí primitivo, no início da década de 1970. Nesse caso, "primitivo" pode ser interpretado como "não tendo um sistema telefônico funcional". Isso não tornava a vida mais agradável para o pesquisador Norman Abramson e seus colegas da University of Hawaii, que estavam tentando conectar usuários nas ilhas remotas ao computador principal em Honolulu. Esticar seus próprios cabos sob o Oceano Pacífico estava fora de cogitação e, portanto, eles procuravam uma solução diferente.

A solução encontrada usava rádios de curta distância, com cada terminal de usuário compartilhando a mesma frequência upstream para enviar quadros ao computador central. Isso incluía um método simples e elegante para resolver o problema de alocação de canal. Seu trabalho foi ampliado por vários pesquisadores desde então (Schwartz e Abramson, 2009). Embora o trabalho de Abramson, denominado sistema ALOHA, usasse a radiofrequência terrestre, a ideia básica é aplicável a qualquer sistema em que usuários sem nenhuma coordenação estão competindo pelo uso de um único canal compartilhado.

Descreveremos aqui duas versões do ALOHA: original e slotted (segmentado). Elas diferem quanto ao fato de o tempo ser contínuo, como na versão original, ou dividido em slots discretos em que todos os quadros devem se encaixar.

ALOHA original

A ideia básica de um sistema ALOHA é simples: permitir que os usuários transmitam sempre que tiverem dados para enviar. Naturalmente, haverá colisões, e os quadros que colidirem serão danificados. Os transmissores precisam, de alguma maneira, descobrir se isso acontece. No sistema ALOHA, após cada estação ter transmitido seu quadro para o computador central, este retransmite o quadro para todas as estações. Desse modo, uma estação transmissora pode escutar por broadcast a partir do hub para ver se seu quadro passou. Em outros sistemas, como nas LANs com fio, o transmissor precisa ser capaz de escutar colisões enquanto transmite.

Se o quadro foi destruído, o transmissor apenas espera um período aleatório e o envia novamente. O tempo de espera deve ser aleatório, caso contrário os mesmos quadros continuarão a colidir repetidas vezes, de forma inflexível. Os sistemas em que vários usuários compartilham um canal comum de forma que possa gerar conflitos em geral são conhecidos como **sistemas de disputa**.

A Figura 4.1 mostra um esboço da geração de quadros em um sistema ALOHA. Os quadros foram criados com o mesmo comprimento porque o throughput dos sistemas ALOHA é maximizado quando o comprimento dos quadros é uniforme em vez de variável.

Sempre que dois quadros tentarem ocupar o canal ao mesmo tempo, haverá uma colisão (como pode ser visto na Figura 4.1) e ambos serão danificados. Se o primeiro bit de um novo quadro se sobrepuser apenas ao último bit de um quadro quase terminado, os dois quadros serão totalmente destruídos (ou seja, terão checksums incorretos) e terão de ser retransmitidos posteriormente. O checksum não consegue (e não deve) fazer distinção entre uma perda total e uma perda parcial. Quadro com erro é quadro com erro, sem distinções.

Uma questão interessante é: qual é a eficiência de um canal ALOHA? Em outras palavras, que fração de todos os quadros transmitidos escapa de colisões nessas

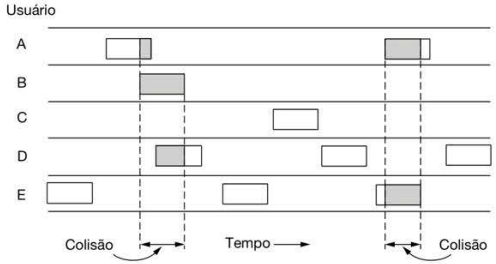


Figura 4.1 No ALOHA original, os quadros são transmitidos em tempos totalmente arbitrários.

circunstâncias tão caóticas? Vamos considerar primeiro um conjunto infinito de usuários interativos em seus terminais (estações). O usuário sempre se encontra em um de dois estados: digitação ou espera. Inicialmente, todos os usuários estão no estado de digitação. Quando uma linha é conectada, o usuário para de digitar e espera uma resposta. Então, a estação transmite um quadro contendo a linha e verifica o canal para saber se a transmissão foi bem-sucedida. Em caso afirmativo, o usuário vê a resposta e volta a digitar. Caso contrário, ele continua a esperar e o quadro é retransmitido continuamente até ser enviado com êxito.

O "tempo de quadro" representa o período necessário para transmitir o quadro padrão de comprimento fixo (i.e., o comprimento do quadro dividido pela taxa de bits). Nesse ponto, supomos que os novos quadros sejam gerados pelas estações de acordo com uma distribuição de Poisson, com uma média de N quadros por tempo de quadro. (A premissa de população infinita é necessária para garantir que N não diminuirá à medida que os usuários forem bloqueados). Se $N > 1$, a comunidade de usuários está gerando quadros em uma taxa superior à capacidade do canal, e praticamente todos os quadros sofrerão colisões. Para um throughput razoável, esperaríamos $0 < N < 1$.

Além dos novos quadros, as estações também geram retransmissões de quadros que sofreram colisões anteriormente. Vamos supor ainda que os quadros antigos e novos combinados também sejam uma distribuição de Poisson, com média G por tempo de quadro. Evidentemente, $G \geq N$. Em situações de carga baixa (ou seja, $N \approx 0$), ocorrerão poucas colisões e, portanto, haverá poucas retransmissões. Por conseguinte, $G \approx N$. Em situações de carga alta, ocorrerão várias colisões e, portanto, $G > N$. Para qualquer carga, o throughput S é simplesmente a carga oferecida, G, multiplicada pela probabilidade P_0 de uma transmissão ser bem-sucedida – isto é, $S = GP_0$, onde P_0 é a probabilidade de um quadro não sofrer colisão.

Um quadro não sofrerá colisão se nenhum outro for enviado dentro de um tempo de quadro a partir de seu início, como mostra a Figura 4.2. Em que condições o quadro sombreado chegará sem erros? Seja t o tempo necessário para enviar um quadro. Se qualquer outro usuário tiver gerado um quadro no intervalo entre t_0 e $t_0 + t$, o final desse quadro colidirá com o início do quadro sombreado. Na verdade, esse quadro já estava condenado antes de o primeiro bit ser transmitido; porém, como no ALOHA original uma estação não escuta o canal antes de transmitir, não há como saber se já havia outro quadro a caminho. Da mesma forma, qualquer outro quadro iniciado entre $t_0 + t$ e $t_0 + 2t$ colidirá com o final do quadro sombreado.

A probabilidade de k quadros serem gerados durante determinado tempo de quadro, no qual G quadros são esperados, é obtida pela distribuição de Poisson

$$\Pr[k] = \frac{G^k \, e^{-G}}{k!} \tag{4.1}$$

e, portanto, a probabilidade de zero quadros é simplesmente e^{-G}. Em um intervalo com duração de dois tempos de quadro, o número médio de quadros gerados é $2G$. A probabilidade de nenhum outro quadro ser iniciado durante todo o período de vulnerabilidade é, portanto, indicada por $P_0 = e^{-2G}$. Usando $S = GP_0$, obtemos:

$$S = Ge^{-2G}$$

A Figura 4.3 mostra a relação entre o tráfego oferecido e o throughput. O throughput máximo ocorre em $G = 0{,}5$, com $S = 1/2e$, o que corresponde a aproximadamente 0,184. Em outras palavras, o melhor que podemos esperar é uma utilização do canal de 18%. Esse resultado não é muito animador, mas, com todas as pessoas transmitindo à vontade, dificilmente poderíamos esperar uma taxa de 100% de êxito.

Slotted ALOHA

Logo depois que o ALOHA entrou em cena, Roberts (1972) publicou um método para duplicar a capacidade de um

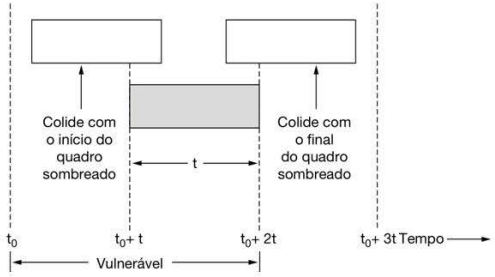


Figura 4.2 Período de vulnerabilidade do quadro sombreado.

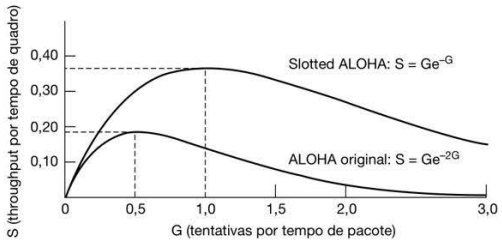


Figura 4.3 Throughput em comparação com o tráfego oferecido para sistemas ALOHA.

sistema ALOHA. Sua proposta era dividir o tempo em intervalos discretos, chamados **slots**, com cada intervalo correspondendo a um quadro. Esse método exige que os usuários concordem em relação às fronteiras dos slots. Uma forma de alcançar a sincronização entre os usuários seria ter uma estação especial que emitisse um sinal sonoro no início de cada intervalo, como um relógio.

No método de Roberts, que passou a ser conhecido como **slotted ALOHA** – em contraste com o **ALOHA original** de Abramson –, um computador não tem permissão para transmitir sempre que o usuário digita uma linha. Em vez disso, é necessário esperar o início do próximo slot. Consequentemente, o ALOHA original contínuo transforma-se em um sistema discreto. O período de vulnerabilidade está agora reduzido à metade. Para verificar isso, examine a Figura 4.2 e imagine as colisões que agora são possíveis. A probabilidade de não haver outro tráfego durante o mesmo slot de nosso quadro de teste é e^{-G}, o que nos leva a:

$$S = Ge^{-G}$$

Como podemos ver na Figura 4.3, a taxa máxima do slotted ALOHA é $G = 1$, com um throughput $S = 1/e$, ou aproximadamente 0,368, o dobro do ALOHA original. Se o sistema estiver funcionando a uma taxa de $G = 1$, a probabilidade de um slot vazio será 0,368 (pela Equação 4.1). O melhor que podemos esperar com a utilização de um slotted ALOHA é 37% de slots vazios, 37% de sucessos e 26% de colisões. O funcionamento em valores superiores de G reduz o número de slots vazios, mas aumenta exponencialmente o número de colisões. Para ver como ocorre esse rápido crescimento de colisões com G, considere a transmissão de um quadro de teste. A probabilidade de ele evitar uma colisão é de e^{-G}, que é a probabilidade de todos os outros usuários estarem inativos nesse slot. A probabilidade de uma colisão é, então, apenas $1 - e^{-G}$. A probabilidade de uma transmissão exigir exatamente k tentativas (ou seja, $k - 1$ colisões seguidas por uma transmissão bem-sucedida) é

$$P_k = e^{-G}(1 - e^{-G})^{k-1}$$

O número esperado de transmissões, E, por cada linha digitada em um terminal é, portanto,

$$E = \sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} ke^{-G}(1 - e^{-G})^{k-1} = e^{G}$$

Como resultado da dependência exponencial de E em relação a G, pequenos aumentos na carga do canal podem reduzir drasticamente seu desempenho.

O slotted ALOHA é importante por uma razão que, a princípio, talvez não seja óbvia. Ele foi criado na década de 1970, foi usado em alguns sistemas experimentais e depois foi quase esquecido (exceto por autores de livro excêntricos, que gostavam dele). Quando foi criado o acesso à Internet por cabo, surgiu o problema de como alocar um canal compartilhado entre vários usuários concorrentes, e o slotted ALOHA foi resgatado para salvar a situação, com um punhado de outras ideias misturadas. Com frequência, protocolos perfeitamente válidos caem em desuso por razões políticas (p. ex., quando alguma grande empresa deseja que todas as outras sigam seu modelo) ou em virtude de tendências tecnológicas em constante mudança. Então, anos depois, alguém inteligente percebe que um protocolo descartado muito antes resolve seu problema atual. Por essa razão, neste capítulo estudaremos diversos protocolos elegantes que não são muito utilizados hoje, mas que poderiam facilmente ser empregados em aplicações futuras, desde que projetistas de redes em número suficiente tivessem conhecimento deles. É claro que também estudaremos muitos protocolos usados atualmente.

4.2.2 Protocolos de acesso múltiplo com detecção de portadora

Com o slotted ALOHA, a melhor utilização de canal que é possível conseguir é $1/e$. Isso não surpreende, pois, com as estações transmitindo à vontade, sem prestar atenção ao que as outras estão fazendo, é provável que ocorram muitas colisões. Contudo, em LANs, as estações podem detectar o que outras estão fazendo e, então, adaptar seu

comportamento de acordo com essa situação, podendo atingir uma utilização melhor que $1/e$. Nesta seção, estudaremos alguns protocolos que melhoram o desempenho da rede.

Os protocolos nos quais as estações escutam uma portadora (i.e., uma transmissão) e funcionam de acordo com ela são denominados **protocolos com detecção de portadora**. Muitos deles têm sido propostos e já há muito tempo foram analisados com detalhes. Por exemplo, consulte Kleinrock e Tobagi (1975). A seguir, mencionaremos algumas versões dos protocolos com detecção de portadora.

CSMA persistente e não persistente

O primeiro protocolo com detecção de portadora que estudaremos aqui denomina-se **CSMA (Carrier Sense Multiple Access) 1-persistente**. Esse é um nome extenso para indicar o esquema CSMA mais simples. Quando uma estação tem dados a transmitir, primeiro ela escuta o canal para ver se mais alguém está transmitindo no momento. Se o canal estiver desocupado, as estações enviam seus dados. Caso contrário, a estação espera até que ele fique desocupado e então transmite um quadro. Se ocorrer uma colisão, a estação espera um intervalo de tempo aleatório e começa tudo de novo. Esse protocolo é denominado 1-persistente porque a estação transmite com probabilidade 1 sempre que encontra o canal desocupado.

Você poderia esperar que esse esquema evitasse colisões, exceto no caso raro de transmissões simultâneas, mas, na verdade, isso não acontece (a situação é muito pior do que esta). Se duas estações estão prontas no meio da transmissão de uma terceira estação, ambas esperarão educadamente até que a transmissão termine e, depois, ambas começarão a transmitir simultaneamente, resultando em uma colisão. Se elas não fossem tão impacientes, haveria menos colisões.

De modo mais sutil, o atraso de propagação tem um efeito importante sobre as colisões. Há uma chance de que, logo após uma estação começar a transmitir, outra estação fique pronta para transmitir e escutar o canal. Se o sinal da primeira estação ainda não tiver atingido a segunda, esta detectará um canal desocupado e também começará a transmitir, resultando em uma colisão. Essa probabilidade depende do número de quadros que cabem no canal, ou o **produto largura de banda-atraso** do canal. Se apenas uma pequena fração do quadro couber no canal, o que é o caso na maioria das LANs, uma vez que o atraso de propagação é pequeno, o risco de uma colisão acontecer é pequeno. Quanto maior o produto largura de banda-atraso, maior será a importância desse efeito e pior será o desempenho do protocolo.

Mesmo assim, esse protocolo tem um desempenho bem melhor que o ALOHA original, pois ambas as estações respeitam a transmissão e desistem de interferir no quadro de uma terceira estação, de modo que ele atravessa sem problemas. Exatamente o mesmo se aplica ao slotted ALOHA.

Um segundo protocolo com detecção de portadora é o **CSMA não persistente**. Nesse protocolo, é feita uma tentativa consciente de ser menos ávido que no protocolo anterior. Como antes, uma estação escuta o canal quando deseja enviar um quadro e, se ninguém mais estiver transmitindo, inicia a transmissão imediatamente. No entanto, se o canal já estiver sendo utilizado, a estação não permanecerá escutando continuamente a fim de se apoderar de imediato do canal após detectar o fim da transmissão anterior. Em vez disso, a estação aguardará durante um intervalo aleatório e, em seguida, repetirá o algoritmo. Consequentemente, esse algoritmo leva a uma melhor utilização do canal, e a atrasos maiores do que no CSMA 1-persistente.

O último protocolo é o **CSMA p-persistente**. Ele se aplica a canais segmentados (slotted) e funciona da forma apresentada a seguir. Quando está pronta para transmitir, a estação escuta o canal. Se ele estiver desocupado, a estação transmite com uma probabilidade p. Com uma probabilidade $q = 1 - p$, haverá um adiamento até o próximo slot. Se este também estiver desocupado, haverá uma transmissão ou um novo adiamento, com probabilidades p e q. Esse processo se repete até o quadro ser transmitido ou até que outra estação tenha iniciado uma transmissão. Neste último caso, ela age como se tivesse ocorrido uma colisão (ou seja, aguarda durante um intervalo aleatório e reinicia a transmissão). Se inicialmente detectar que o canal está ocupado, a estação espera pelo próximo slot e aplica o algoritmo anterior. O IEEE 802.11 usa uma melhoria do CSMA p-persistente, que discutiremos na Seção 4.4.

A Figura 4.4 mostra o throughput calculado em comparação com o tráfego oferecido para os três protocolos, bem como para o ALOHA original e o slotted ALOHA.

CSMA com detecção de colisões

Os protocolos CSMA persistentes e não persistentes são um avanço claro em relação ao ALOHA, pois garantem que nenhuma estação começará a transmitir enquanto o canal estiver ocupado. Contudo, se duas estações perceberem que o canal está desocupado e começarem a transmitir simultaneamente, seus sinais ainda causarão colisão. Outro avanço é que as estações podem detectar a colisão rapidamente e interromper a transmissão de forma abrupta (em vez de completá-la), pois não têm como reparar a situação. Essa estratégia economiza tempo e largura de banda.

Esse protocolo, conhecido como **CSMA/CD (CSMA with Collision Detection)**, é a base da LAN Ethernet clássica; assim, vale a pena dedicarmos algum tempo a examiná-lo em detalhes. É importante observar que a detecção de colisão é um processo analógico. O hardware da estação deve escutar o canal enquanto está transmitindo. Se o sinal que ela lê de volta for diferente do sinal que está enviando, ela saberá que está havendo uma colisão. As implicações

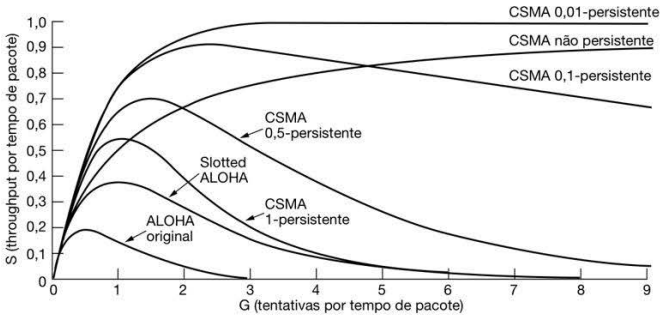


Figura 4.4 Comparação entre a utilização do canal e a carga de vários protocolos de acesso aleatório.

são que um sinal recebido não deverá ser pequeno em comparação com o sinal transmitido (o que é difícil para as redes sem fio, pois os sinais recebidos podem ser um milhão de vezes mais fracos do que os sinais transmitidos) e que a modulação deve ser escolhida para permitir que as colisões sejam detectadas (p. ex., uma colisão de dois sinais de 0 volt pode muito bem ser impossível de detectar).

O CSMA/CD e vários outros protocolos de LANs utilizam o modelo conceitual apresentado na Figura 4.5. No ponto marcado com t_0, uma estação terminou a transmissão de seu quadro. Qualquer outra estação que tenha um quadro a ser enviado pode transmiti-lo. Se duas ou mais estações decidirem transmitir simultaneamente, haverá uma colisão. Se uma estação detecta uma colisão, ela cancela sua transmissão, espera um intervalo aleatório e, em seguida, tenta novamente (supondo que nenhuma outra estação tenha começado a transmitir nesse ínterim). Dessa forma, nosso modelo de CSMA/CD consistirá em períodos alternados de disputa e de transmissão, com a ocorrência de períodos de inatividade quando todas as estações estiverem em repouso (p. ex., por falta de trabalho).

Agora, vamos analisar mais de perto os detalhes do algoritmo de disputa. Suponha que duas estações comecem uma transmissão no instante exato t_0. Quanto tempo elas levarão para perceber que houve uma colisão? A resposta a essa pergunta é essencial para determinar a duração do intervalo de disputa e, portanto, quais serão o atraso e o throughput.

O tempo mínimo para a detecção de uma colisão é apenas o tempo que o sinal leva para se propagar de uma estação até a outra. Com base nessa informação, você poderia pensar que uma estação que não detectasse uma colisão durante um intervalo igual ao tempo de propagação em todo o cabo, após ter iniciado sua transmissão, teria certeza de haver se apoderado do cabo. Com o termo "apoderado", queremos dizer que todas as outras estações sabem da transmissão e não interferirão. Essa conclusão está incorreta.

Considere a pior hipótese possível a seguir. Seja τ o tempo de propagação de um sinal entre as duas estações mais distantes. Em t_0, uma estação começa a transmitir. Em $t_0 + \tau - \varepsilon$, um instante antes de o sinal chegar à estação mais distante, essa estação também começa a transmitir. É claro que ela detecta a colisão quase instantaneamente e para, mas o pequeno ruído causado pela colisão não retorna à estação original até o período $2\tau - \varepsilon$. Em outras palavras, no pior cenário, uma estação só poderá ter certeza de ter se apoderado do canal após transmitir durante o período 2τ sem detectar uma colisão.

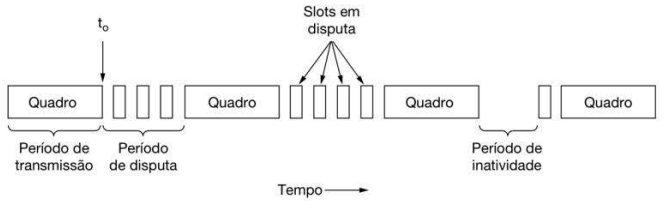


Figura 4.5 O CSMA/CD pode estar em um destes três estados: transmissão, disputa ou inatividade.

Compreendendo isso, podemos pensar na disputa do CSMA/CD como um sistema slotted ALOHA, com uma largura de slot igual a 2τ. Em um cabo coaxial de 1 km de comprimento, $\tau \approx 5\mu s$. A diferença para CSMA/CD em comparação com o slotted ALOHA é que os slots em que apenas uma estação transmite (ou seja, em que o canal é apoderado) são acompanhados pelo restante de um quadro. Essa diferença melhorará bastante o desempenho se o tempo do quadro for muito maior que o tempo de propagação.

4.2.3 Protocolos livres de colisão

Embora as colisões não ocorram com o CSMA/CD depois que uma estação captura o canal sem ambiguidade, elas ainda podem ocorrer durante o período de disputa. Essas colisões afetam de modo adverso o desempenho do sistema, em especial quando o produto largura de banda-atraso é grande, por exemplo, quando o cabo é longo (ou seja, quando τ é grande) e os quadros são curtos. As colisões não só reduzem a largura de banda, mas também tornam variável o tempo para transmitir um quadro, o que não é bom para um tráfego em tempo real, como VoIP. Além disso, o CSMA/CD não é aplicável de maneira universal.

Nesta seção, examinaremos alguns protocolos que resolvem a disputa pelo canal sem a ocorrência de colisões, nem mesmo durante o período de disputa. A maioria desses protocolos não é usada atualmente em sistemas importantes, mas, em um campo que muda rapidamente, a existência de alguns protocolos com excelentes propriedades disponíveis para sistemas futuros frequentemente é algo bom.

Nos protocolos que descreveremos, vamos supor que existam exatamente N estações, cada uma programada com um endereço exclusivo de 0 até $N - 1$. O fato de que algumas estações talvez possam estar inativas durante parte do tempo não tem importância. Também supomos que o atraso de propagação é desprezível. A pergunta básica permanece: que estação terá a posse do canal após uma transmissão bem-sucedida? Continuaremos a utilizar o modelo mostrado na Figura 4.5 com seus slots discretos de disputa.

O protocolo bit-map

Em nosso primeiro protocolo livre de colisão, o **método básico bit-map**, cada período de disputa consiste em exatamente N slots. Se tiver um quadro para transmitir, a estação 0 envia um bit 1 durante o slot número zero. Nenhuma outra estação poderá transmitir durante esse slot. Independentemente do que a estação 0 fizer, a estação 1 terá a oportunidade de transmitir um bit 1 durante o slot 1, mas apenas se tiver um quadro na fila para ser enviado. Em geral, é possível que a estação j informe que tem um quadro para transmitir inserindo um bit 1 no slot j. Depois que todos os N slots tiverem passado, cada estação terá total conhecimento de quais estações desejam transmitir. Nesse ponto, elas começam a transmitir em ordem numérica (ver Figura 4.6).

Como todas as estações concordam sobre quem será a próxima a transmitir, nunca haverá colisões. Após a última estação ter transmitido seu quadro, um evento que todas as estações podem monitorar com facilidade, inicia-se outro período de disputa de N bits. Se uma estação ficar pronta logo após seu slot de bits ter passado, ela não conseguirá transmitir e precisará permanecer inativa até que todas as demais tenham tido a chance de transmitir e o bit-map tenha voltado a passar por ela.

Protocolos como esse, nos quais o desejo de transmitir é difundido antes de ocorrer a transmissão real, são chamados de **protocolos de reserva**, pois eles reservam a posse do canal com antecedência e impedem colisões. Vamos analisar rapidamente o desempenho desse protocolo. Para facilitar, mediremos o tempo em unidades do slot de bits de disputa, com os quadros de dados consistindo em d unidades de tempo.

Em condições de carga baixa, o bit-map simplesmente será repetido várias vezes, por falta de quadros de dados. Considere a situação do ponto de vista de uma estação com numeração baixa, como 0 ou 1. Normalmente, quando ela fica pronta para enviar, o slot "atual" estará em algum ponto no meio do bit-map. Em média, a estação terá de esperar $N/2$ slots para que a varredura atual seja concluída e mais N slots completos até que a varredura seguinte se encerre, para poder começar a transmitir.

As estações que estiverem aguardando e tiverem números mais altos obterão resultados melhores. Em geral, elas só precisarão esperar pela metade de uma varredura ($N/2$ slots de bits) antes de iniciar a transmissão. As estações com numeração mais alta raramente precisam esperar pela próxima varredura. Como as estações de numeração baixa precisam esperar em média 1,5 N slots e as de numeração alta precisam esperar em média 0,5 N slot, a média para todas as estações é N slots.

Figura 4.6 O protocolo básico bit-map.

É fácil calcular a eficiência do canal com carga baixa. O overhead por quadro é de N bits, e o volume de dados é de d bits, o que resulta em uma eficiência igual a $d/(d + N)$.

Sob carga alta, quando todas as estações têm algo a enviar o tempo todo, o período de disputa de N bits é dividido proporcionalmente entre N quadros, produzindo um overhead de apenas 1 bit por quadro, ou uma eficiência igual a $d/(d + 1)$. O atraso médio para um quadro é equivalente à soma do tempo de espera na fila dentro da estação, mais um adicional de $(N - 1)d + N$, uma vez que ele alcança o início de sua fila interna. Esse intervalo é o tempo necessário para esperar até que todas as outras estações tenham sua vez para enviar um quadro e outro bit-map.

Passagem de tokens

A essência do protocolo bit-map é que ele permite que cada estação transmita um quadro por vez, em uma ordem predefinida. Outra forma de realizar a mesma coisa é passar uma pequena mensagem, chamada **token** ou **sinal**, de uma estação para a seguinte, na mesma ordem predefinida. O token representa a permissão para enviar. Se uma estação tem um quadro na fila para transmissão quando recebe o token, ela pode enviar esse quadro antes de passar o token para a próxima estação. Se ela não tiver um quadro na fila, ela simplesmente passará o token.

Em um protocolo que utiliza a topologia de anel de tokens (**token ring**), esta é usada para definir a ordem em que as estações transmitem. As estações são conectadas às seguintes formando um anel único. A passagem do token para a estação seguinte consiste simplesmente em receber o token em uma direção e transmiti-lo em outra, como vemos na Figura 4.7. Os quadros também são transmitidos na direção do token. Desse modo, eles circularão em torno do anel e alcançarão qualquer estação que seja o destino. Contudo, para impedir que o quadro circule indefinidamente (assim como o próprio token), alguma estação precisa removê-lo do anel. Essa estação pode ser a que enviou o quadro originalmente, depois que ele passou por um ciclo completo, ou a estação de destino do quadro.

Observe que não precisamos de um anel físico para implementar a passagem de tokens. Em vez disso, o canal que conecta as estações poderia ser um único e longo barramento (cabo). Em seguida, cada estação usa o barramento para enviar o token para a próxima estação em uma sequência predefinida. A posse do token permite que uma estação use o barramento para enviar um quadro. Esse protocolo é chamado de **barramento de tokens** (ou **token bus**). Ele é definido no IEEE 8024, um padrão que fracassou ao ponto de o IEEE retirá-lo. Os padrões nem sempre são eternos.

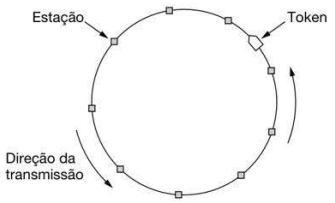


Figura 4.7 Token ring.

O desempenho da passagem de tokens é semelhante ao do protocolo bit-map, embora os slots de disputa e os quadros de um ciclo agora estejam embaralhados. Depois de enviar um quadro, cada estação precisa esperar que todas as N estações (incluindo ela mesma) transmitam o token aos seus vizinhos e as outras $N - 1$ estações transmitam um quadro, se tiverem um. Uma diferença sutil é que, como todas as posições no ciclo são equivalentes, não existe parcialidade para estações com numeração baixa ou alta. Para o token ring, cada estação também está apenas transmitindo o token, enquanto sua estação vizinha anterior no protocolo realiza o passo seguinte. Cada token não precisa se propagar para todas as estações antes que o protocolo avance para o passo seguinte.

Os token rings surgiram como protocolos MAC com certa consistência. Um antigo protocolo token ring (chamado "Token Ring" e padronizado como IEEE 802.5) era popular na década de 1980 como uma alternativa à Ethernet clássica. Na década de 1990, um token ring muito mais rápido, chamado **FDDI** (**Fiber Distributed Data Interface**), foi extinto pela Ethernet comutada. Na década de 2000, um token ring chamado **RPR** (**Resilient Packet Ring**) foi definido como IEEE 802.17 para padronizar a mistura de anéis metropolitanos em uso pelos ISPs. Ainda veremos o que a década de 2020 nos oferecerá.

Contagem regressiva binária

Um problema com o protocolo básico bit-map, e também com a passagem de tokens, é que o overhead é de 1 bit por estação e, portanto, ele não se adapta muito bem a redes com milhares de estações. Podemos fazer melhor que isso usando endereços binários de estações com um canal que, de alguma forma, combine transmissões. Uma estação que queira usar o canal transmite seu endereço como uma sequência de bits binários, começando com o bit de alta ordem. Supomos que todos os endereços têm o mesmo número de bits. Os bits de cada posição de endereço das diferentes estações passam juntos por uma operação OR booleana pelo canal quando são enviados ao mesmo tempo. Chamaremos esse protocolo de **contagem regressiva binária**. Ele foi usado no Datakit (Fraser, 1983). Esse protocolo pressupõe implicitamente que os atrasos de transmissão são desprezíveis, de forma que todas as estações detectam bits declarados quase instantaneamente.

Para evitar conflitos, é necessário que seja aplicada uma regra de arbitragem: assim que uma estação percebe

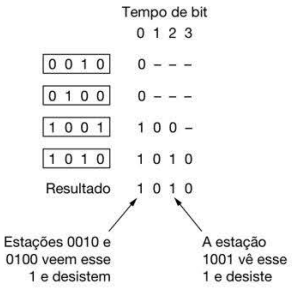


Figura 4.8 Protocolo de contagem regressiva binária. Um traço significa inatividade.

que um bit de alta ordem que em seu endereço era 0 foi sobrescrito por um bit 1, ela desiste. Por exemplo, se as estações 0010, 0100, 1001 e 1010 estiverem todas tentando acessar o canal, no primeiro período de um bit, as estações transmitirão 0, 0, 1 e 1, respectivamente. Esses valores passarão pela operação OR para formar um valor 1. As estações 0010 e 0100 veem o valor 1 e sabem que uma estação de numeração mais alta está disputando o canal e, portanto, desistem da tentativa na rodada atual. As estações 1001 e 1010 prosseguem.

O próximo bit é 0, e ambas as estações continuam a transmissão. O próximo bit é 1 e, portanto, a estação 1001 desiste. A vencedora é a estação 1010, pois tem o endereço mais alto. Após vencer a disputa, é provável que agora ela possa transmitir um quadro, após o qual terá início outro ciclo de disputa. A Figura 4.8 ilustra esse protocolo. Ele tem a propriedade de dar às estações com numeração mais alta uma prioridade maior do que a concedida às estações de numeração mais baixa, o que pode ser bom ou ruim, dependendo do contexto.

Com esse método, a eficiência do canal é $d/(d + \log_2 N)$. No entanto, se o formato do quadro tiver sido corretamente escolhido, de forma que o endereço do transmissor seja o primeiro campo do quadro, mesmo esses $\log_2 N$ bits não serão desperdiçados, e a eficiência será de 100%.

A contagem regressiva binária é um exemplo de protocolo simples, elegante e eficiente, que está esperando o momento de ser redescoberto. Esperamos que algum dia ele encontre um novo lar.

4.2.4 Protocolos de disputa limitada

Vimos até agora duas estratégias básicas para a aquisição de canais em uma rede de broadcast: protocolos de disputa, como no CSMA, e protocolos livres de colisão. Cada estratégia é classificada de acordo com seu desempenho em relação a duas medidas importantes: o atraso em carga baixa e a eficiência de canal em carga alta. Em condições de carga leve, a disputa (ou seja, o ALOHA original ou o slotted ALOHA) é preferível, em virtude de seu baixo índice de atraso (pois as colisões são raras). À medida que a carga aumenta, a disputa torna-se cada vez menos interessante, pois o overhead associado à arbitragem do canal torna-se maior. O oposto também é verdadeiro em relação aos protocolos livres de colisão. Em carga baixa, eles têm um alto índice de atraso, mas, à medida que a carga aumenta, a eficiência do canal melhora (pois os overheads são fixos).

Obviamente, seria bom se pudéssemos combinar as melhores propriedades dos protocolos de disputa e dos protocolos livres de colisão, chegando a um novo protocolo que usaria não só a disputa em cargas baixas, para proporcionar um baixo índice de atraso, mas também a técnica livre de colisão em carga alta, para oferecer uma boa eficiência do canal. Esses protocolos, que chamaremos de **protocolos de disputa limitada**, de fato existem, e concluirão nosso estudo sobre redes com detecção de portadora.

Até agora, os únicos protocolos de disputa que estudamos são simétricos. Ou seja, cada estação tenta acessar o canal com alguma probabilidade, p, com todas as estações usando o mesmo p. É interessante observar que o desempenho geral do sistema às vezes pode ser melhorado com o uso de um protocolo que atribua probabilidades distintas a diferentes estações.

Antes de examinarmos os protocolos assimétricos, faremos uma pequena revisão do desempenho do caso simétrico. Suponha que k estações estejam disputando o acesso a um canal. Cada uma tem a probabilidade p de transmitir durante cada slot. A probabilidade de alguma estação acessar o canal com sucesso durante determinado slot é a probabilidade de que qualquer estação transmita, com probabilidade p, e todas as outras $k - 1$ estações adiem, cada uma com probabilidade $1 - p$. Esse valor é $kp(1 - p)^{k-1}$. Para encontrar o valor ideal de p, diferenciamos em relação a p, definimos o resultado como zero e resolvemos a equação para p. Ao fazer isso, descobrimos que o melhor valor de p é $1/k$. Ao substituirmos $p = 1/k$, obteremos:

$$\Pr[\text{sucesso com } p \text{ ideal}] = \left(\frac{k-1}{k}\right)^{k-1}$$

Essa probabilidade está representada na Figura 4.9. Para um pequeno número de estações, as chances de sucesso são boas, mas, tão logo o número de estações alcance até mesmo cinco, a probabilidade cai até um número próximo de seu valor assintótico, $1/e$.

Pela Figura 4.9, fica evidente que a probabilidade de alguma estação adquirir o canal só pode ser aumentada diminuindo-se o volume de competição. Os protocolos de disputa limitada fazem exatamente isso. Primeiro, eles dividem as estações em grupos (não necessariamente

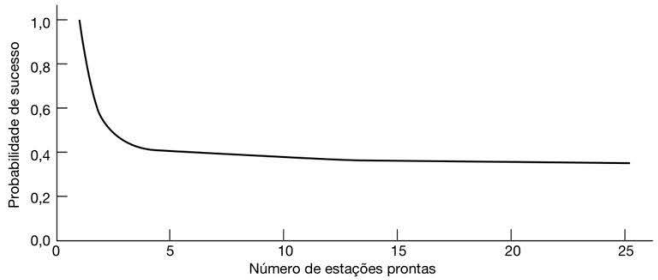


Figura 4.9 Probabilidade de aquisição de um canal de disputa simétrico.

disjuntos). Apenas os membros do grupo 0 podem disputar o slot 0. Se um deles obtiver êxito, adquirirá o canal e transmitirá seu quadro. Se um slot permanecer inativo ou se ocorrer uma colisão, os membros do grupo 1 disputarão o slot 1, etc. Fazendo-se uma divisão apropriada das estações em grupos, o volume de disputa por cada slot pode ser reduzido e, assim, a operação de cada slot ficará próxima à extremidade esquerda da Figura 4.9.

O truque é a maneira de atribuir estações a slots. Antes de analisarmos o caso geral, vamos considerar algumas situações especiais. Em um extremo, cada grupo tem apenas um membro. Essa atribuição garante que nunca ocorrerão colisões, pois existirá, no máximo, uma estação disputando qualquer slot dado. Já vimos esse tipo de protocolo antes (p. ex., a contagem regressiva binária). A próxima situação especial é atribuir duas estações por grupo. A probabilidade de ambas tentarem transmitir durante um slot é p^2, que, para um p pequeno, é desprezível. À medida que mais e mais estações são atribuídas ao mesmo slot, a probabilidade de colisão aumenta, mas diminui a extensão da varredura de bit-map necessária para que todas tenham uma chance. A situação-limite consiste em um único grupo que contém todas as estações (ou seja, o slotted ALOHA). O que precisamos é de uma forma de atribuir dinamicamente estações a slots, com várias estações por slot quando a carga for baixa, e poucas estações (ou apenas uma) por slot quando a carga for alta.

O protocolo adaptativo tree-walk

Uma maneira particularmente simples de fazer as atribuições necessárias consiste em usar o algoritmo desenvolvido pelo exército norte-americano para testar a incidência de sífilis em soldados durante a Segunda Guerra Mundial (Dorfman, 1943). Em resumo, o exército extraiu uma amostra de sangue de N soldados. Uma parte de cada amostra foi colocada em um único tubo de teste. Essa amostra misturada foi submetida a teste para detectar anticorpos. Se nenhum anticorpo fosse encontrado, todos os soldados do grupo eram considerados saudáveis. Se houvesse anticorpos, duas novas amostras misturadas eram preparadas, uma dos soldados numerados de 1 a $N/2$ e outra com o sangue dos demais soldados. O processo era repetido recursivamente até que os soldados infectados fossem identificados.

Para a versão computacional desse algoritmo (Capetanakis, 1979), é conveniente imaginar as estações como as folhas de uma árvore binária, conforme ilustra a Figura 4.10. No primeiro slot de disputa que segue uma transmissão de quadro bem-sucedida, o slot 0, todas as estações têm permissão para tentar acessar o canal. Se uma delas conseguir, muito bem. Se ocorrer uma colisão, durante o slot 1 apenas as estações que estiverem sob o nó 2 da árvore poderão disputar o canal. Se uma delas se apoderar dele, o slot seguinte ao quadro ficará reservado para as estações do nó 3. Por sua vez, se duas ou mais estações no nó 2 quiserem transmitir, haverá uma colisão durante o slot 1 e, nesse caso, será a vez do nó 4 durante o slot 2.

Basicamente, se ocorrer uma colisão durante o slot 0, toda a árvore será pesquisada, primeiro em profundidade, a fim de localizar todas as estações prontas para transmissão. Cada slot de bits é associado a algum nó específico da árvore. Se houver uma colisão, a pesquisa continuará

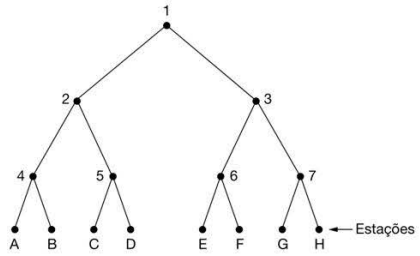


Figura 4.10 A árvore para oito estações.

recursivamente com os filhos localizados à esquerda e à direita desse nó. Se um slot de bits estiver inativo ou se houver apenas uma estação transmitindo nesse slot, a pesquisa de seu nó poderá ser encerrada, pois todas as estações prontas terão sido localizadas. (Se houvesse mais de uma, uma colisão teria ocorrido.)

Quando a carga do sistema está muito pesada, quase não vale a pena o esforço de dedicar o slot 0 ao nó 1, pois esse procedimento só faz sentido na eventualidade improvável de que exatamente uma estação tenha um quadro a ser transmitido. Assim, alguém poderia argumentar que os nós 2 e 3 também deveriam ser ignorados, pela mesma razão. Em termos mais gerais, em que nível da árvore a pesquisa deve ter início? É claro que, quanto maior for a carga, mais baixo na árvore o ponto de início da pesquisa deve estar. Por ora, vamos supor que cada estação tenha uma boa estimativa do número q de estações prontas, por exemplo, com base no monitoramento de tráfego mais recente.

Para prosseguir, vamos numerar os níveis da árvore a partir do topo, com o nó 1 da Figura 4.10 no nível 0, os nós 2 e 3 no nível 1, e assim por diante.. Observe que cada nó do nível i tem uma fração 2^{-i} das estações que se encontram abaixo dele. Se as q estações prontas estiverem uniformemente distribuídas, o número esperado dessas estações abaixo de um nó específico do nível i será apenas $2^{-i}\,q$. Intuitivamente, seria de se esperar que o nível ideal para iniciar a pesquisa na árvore fosse aquele no qual o número médio de estações em disputa por slot fosse igual a 1, isto é, o nível em que $2^{-i}\,q = 1$. Resolvendo essa equação, descobrimos que $i = \log_2 q$.

Foram descobertas diversas melhorias no algoritmo básico, as quais são abordadas em detalhes por Bertsekas e Gallager (1992), mas os pesquisadores ainda estão trabalhando nisso (De Marco e Kowalski, 2017). Por exemplo, considere a hipótese em que as estações G e H são as únicas que estão esperando para transmitir. No nó 1, ocorrerá uma colisão, de modo que 2 será testado e descoberto como nó inativo. É inútil testar o nó 3, pois é certo que haverá colisão (sabemos que duas ou mais estações abaixo de 1 estão prontas e que nenhuma delas está abaixo de 2; portanto, todas devem estar abaixo de 3). A sondagem do nó 3 pode ser ignorada, e o nó 6 será testado em seguida. Quando essa sondagem também não produzir nenhum resultado, 7 poderá ser ignorado e o nó G poderá ser testado em seguida.

4.2.5 Protocolos de LANs sem fio

Um sistema de notebooks que se comunicam por rádio pode ser considerado uma LAN sem fio, como discutimos na Seção 1.4.3. Essa LAN é um exemplo de canal de broadcast. Ela também tem propriedades um pouco diferentes das que caracterizam as LANs com fio, o que leva a diferentes protocolos MAC. Nesta seção, analisaremos alguns deles. Na Seção 4.4, examinaremos a rede 802.11 (WiFi) em detalhes.

Uma configuração comum para uma LAN sem fio é um edifício comercial com pontos de acesso (PAs) estrategicamente posicionados. Todos os PAs são interconectados com o uso de cobre ou fibra, para melhorar a conectividade com as estações que falam com eles. Se a potência de transmissão dos PAs e dos notebooks for ajustada para um alcance de dezenas de metros, as salas vizinhas se tornarão uma única célula e o edifício inteiro passará a ser um grande sistema celular, como os que estudamos no Capítulo 2, exceto que cada célula só tem um canal, compartilhado por todas as estações em sua célula, incluindo o PA. Em geral, ela oferece larguras de banda de megabits/s até gigabits/s. Teoricamente, o IEEE 802.11ac pode atingir 7 Gbps; porém, na prática, ele é muito mais lento.

Já notamos que os sistemas sem fio normalmente não podem detectar uma colisão enquanto ela está ocorrendo. O sinal recebido em uma estação pode ser curto, talvez um milhão de vezes mais fraco que o sinal que está sendo transmitido. Encontrá-lo é como procurar uma onda no oceano. Em vez disso, as confirmações são usadas para descobrir colisões e outros erros após o fato.

Há uma diferença ainda mais importante entre as LANs sem fio e as convencionais. Uma estação em uma LAN sem fio pode não ser capaz de transmitir quadros ou recebê-los de todas as estações, em decorrência do alcance de rádio limitado das estações. Nas LANs com fio, quando uma estação envia um quadro, todas as outras estações o recebem. A ausência dessa propriedade nas LANs sem fio causa uma série de complicações.

Vamos simplificar supondo que cada transmissor de rádio tenha algum alcance fixo, representado por uma região de cobertura circular dentro da qual outra estação possa detectar e receber a transmissão da estação. É importante observar que, na prática, as regiões de cobertura não são tão regulares, pois a propagação dos sinais de rádio depende do ambiente. As paredes e outros obstáculos que atenuam e refletem sinais podem fazer o alcance ser bastante diferente em diversas direções. Mas um modelo circular simples servirá aos nossos propósitos.

Uma abordagem simples para o uso de uma LAN sem fio seria tentar o CSMA: basta escutar outras transmissões e transmitir apenas se ninguém mais estiver usando o canal. O problema é que esse protocolo realmente não é uma boa maneira de pensar nas redes sem fio, pois o que importa para a recepção é a interferência no receptor, e não no transmissor. Para ver a natureza do problema, considere a Figura 4.11, na qual ilustramos quatro estações sem fio. Para os nossos propósitos, não importa quais são PAs e quais são notebooks. O alcance do rádio é tal que A e B estão dentro do alcance e podem interferir um no outro. C também pode interferir em B e D, mas não em A.

Considere primeiro o que acontece quando A está transmitindo para B, como mostra a Figura 4.11(a). Se A transmitir e depois C imediatamente detectar o meio físico,

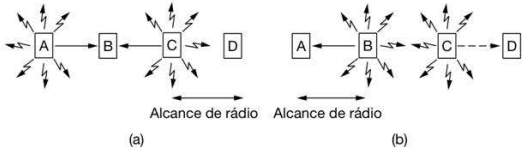


Figura 4.11 Uma LAN sem fio. (a) A e C são terminais ocultos ao transmitir para B. (b) B e C são terminais expostos ao transmitir para A e D.

ele não ouvirá A, pois essa estação está fora do alcance e, portanto, C concluirá incorretamente que pode transmitir para B. Se começar a transmitir, C interferirá em B, removendo o quadro de A. (Consideramos aqui que nenhum esquema tipo CDMA é usado para oferecer múltiplos canais, de modo que as colisões inutilizam o sinal e destroem os dois quadros.) Queremos um protocolo MAC que impeça esse tipo de colisão, pois isso desperdiça largura de banda. O problema de uma estação não conseguir detectar uma provável concorrente pelo meio físico, porque a estação concorrente está muito longe, é denominado **problema da estação oculta**.

Agora, vamos considerar uma situação diferente: B está transmitindo para A ao mesmo tempo que C deseja transmitir para D, como mostra a Figura 4.11(b). Se detectar o meio físico, C ouvirá uma transmissão em andamento e concluirá incorretamente que não pode transmitir para D (como mostra a linha tracejada). Na verdade, essa transmissão só geraria uma recepção de má qualidade na zona entre B e C, em que nenhum dos receptores desejados está localizado. Queremos um protocolo MAC que impeça esse tipo de adiamento, pois ele desperdiça largura de banda. Essa situação é chamada de **problema da estação exposta**.

O problema é que, antes de iniciar uma transmissão, a estação realmente deseja saber se há ou não atividade de rádio em torno do receptor. O CSMA apenas informa a ela se há ou não atividade na estação que detecta a portadora. Com um fio, todos os sinais se propagam para todas as estações e, portanto, não existe distinção. Todavia, somente uma transmissão pode ocorrer de cada vez em qualquer parte do sistema. Em um sistema baseado em ondas de rádio de pequeno alcance, várias transmissões podem ocorrer simultaneamente, se todas tiverem destinos diferentes e eles estiverem fora do alcance uns dos outros. Queremos que essa concorrência aconteça quando a célula se tornar cada vez maior, da mesma forma que pessoas em uma festa não devem esperar que todos na sala fiquem em silêncio antes de começar a falar; várias conversas podem ocorrer ao mesmo tempo em uma sala grande, desde que elas não sejam dirigidas para o mesmo local.

Um protocolo antigo e influente, que trata desses problemas em LANs sem fio, é o **acesso múltiplo com prevenção de colisão**, ou **MACA (Multiple Access with Collision Avoidance)** (Karn, 1990 e Garcia-Luna-Aceves, 2017). A ideia básica consiste em fazer o transmissor estimular o receptor a liberar um quadro curto como saída, de modo que as estações vizinhas possam detectar essa transmissão e evitar transmitir enquanto o quadro de dados (grande) estiver sendo recebido. Essa técnica é usada no lugar da detecção de portadora.

O MACA é ilustrado na Figura 4.12. Vamos analisar agora como A envia um quadro para B. A inicia a transmissão enviando um quadro de **solicitação para envio**, ou **RTS (Request to Send)**, para B, como mostra a Figura 4.12(a).

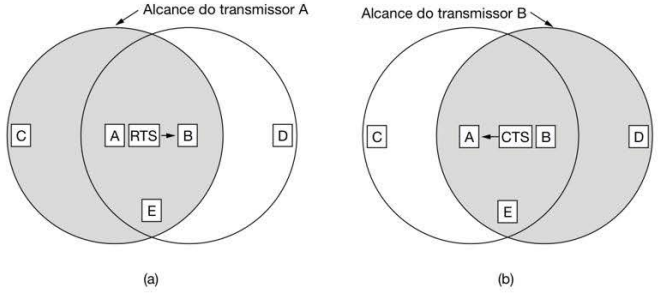


Figura 4.12 O protocolo MACA. (a) A está enviando um quadro RTS para B. (b) B está respondendo com um quadro CTS para A.

Esse quadro curto (30 bytes) contém o comprimento do quadro de dados que possivelmente será enviado em seguida. Depois disso, *B* responde com um quadro de **liberação para envio**, ou **CTS (Clear to Send)**, como mostra a Figura 4.12(b). O quadro CTS contém o tamanho dos dados (copiado do quadro RTS). Após o recebimento do quadro CTS, *A* inicia a transmissão.

Agora, vejamos como reagem as estações que estão ouvindo ambos os quadros. Qualquer estação que esteja ouvindo o quadro RTS está próxima de *A* e deve permanecer inativa por tempo suficiente para que o CTS seja transmitido de volta para *A*, sem conflito. Qualquer estação que esteja ouvindo o CTS está próxima de *B* e deve permanecer inativa durante a transmissão de dados que está a caminho, cujo tamanho pode ser verificado pelo exame do quadro CTS.

Na Figura 4.12, *C* está dentro do alcance de *A*, mas não no alcance de *B*. Portanto, essa estação pode detectar o RTS de *A*, mas não o CTS de *B*. Desde que não interfira no CTS, a estação é livre para transmitir enquanto o quadro de dados está sendo enviado. Ao contrário, *D* está dentro do alcance de *B*, mas não de *A*. Ela não detecta o RTS, mas sim o CTS. Ao detectá-lo, ela recebe a indicação de que está perto de uma estação que está prestes a receber um quadro e, portanto, adia a transmissão até o momento em que o envio desse quadro provavelmente esteja concluído. A estação *E* detecta as duas mensagens de controle e, como *D*, deve permanecer inativa até que a transmissão do quadro de dados seja concluída.

Apesar dessas precauções, ainda pode haver colisões. Por exemplo, *B* e *C* poderiam enviar quadros RTS para *A* ao mesmo tempo. Haverá uma colisão entre esses quadros e eles se perderão. No caso de uma colisão, um transmissor que não obtiver êxito (ou seja, o que não detectar um CTS no intervalo esperado) aguardará durante um intervalo aleatório e tentará novamente mais tarde.

4.3 ETHERNET

Agora, concluímos nossa discussão resumida sobre protocolos de alocação de canais e, portanto, é hora de analisar como esses princípios se aplicam a sistemas reais. Muitos dos projetos para redes pessoais, locais e metropolitanas foram padronizados com o nome IEEE 802. Alguns desses padrões sobreviveram, mas muitos não, como vimos na Figura 1.38. Algumas pessoas que acreditam em reencarnação creem que Charles Darwin retornou como membro da associação de padrões do IEEE com a finalidade de eliminar os menos capazes. Os mais importantes entre os sobreviventes são os padrões 802.3 (Ethernet) e 802.11 (LAN sem fio). O Bluetooth (PAN sem fio) é bastante utilizado, mas agora foi padronizado fora do 802.15.

Começaremos nosso estudo dos sistemas reais com a Ethernet, provavelmente o tipo de rede de computação mais utilizado no mundo. Existem dois tipos de Ethernet: **Ethernet clássica**, que resolve o problema de acesso múltiplo por meio de técnicas que estudamos neste capítulo, e **Ethernet comutada**, em que dispositivos chamados **switches** são usados para conectar diferentes computadores. É importante observar que, embora ambas sejam chamadas Ethernet, elas são muito diferentes. A Ethernet clássica é a forma original, que atuava em velocidades de 3 a 10 Mbps. A Ethernet comutada é a evolução da Ethernet, e trabalha em velocidades de 100, 1.000, 10.000, 40.000 ou 100.000 Mbps, ao que chamamos Fast Ethernet, gigabit Ethernet, 10 gigabit Ethernet, 40 gigabit Ethernet ou 100 gigabit Ethernet. Na prática, somente a Ethernet comutada é usada atualmente.

Discutiremos essas formas históricas da Ethernet em ordem cronológica, mostrando como elas se desenvolveram. Como Ethernet e IEEE 802.3 são idênticos, exceto por uma pequena diferença (que discutiremos em breve), muitas pessoas usam os termos "Ethernet" e "IEEE 802.3" para indicar a mesma coisa. Também faremos isso aqui. Para obter mais informações sobre Ethernet, consulte Spurgeon e Zimmerman (2014).

4.3.1 Camada física da Ethernet clássica

A história da Ethernet começa mais ou menos na época da ALOHA, quando um aluno chamado Bob Metcalfe conseguiu seu título de bacharel no MIT e depois "subiu o rio" para obter seu título de Ph.D. em Harvard. Durante seus estudos, conheceu o trabalho de Abramson sobre ALOHA. Ele ficou tão interessado que, depois de se formar em Harvard, decidiu passar o verão no Havaí trabalhando com Abramson, antes de iniciar seu trabalho no Xerox PARC (Palo Alto Research Center). Quando chegou ao PARC, viu que os pesquisadores de lá haviam projetado e montado o que mais tarde seriam chamados computadores pessoais. Mas as máquinas eram isoladas. Usando seu conhecimento do trabalho de Abramson, Metcalfe, com seu colega David Boggs, projetou e implementou a primeira rede local (Metcalfe e Boggs, 1976). Ele usou um único cabo coaxial grosso e conseguiu trabalhar a 3 Mbps.

Metcalfe e Boggs chamaram o sistema de **Ethernet**, fazendo referência ao *éter transmissor de luz* (do inglês *luminiferous ether*), através do qual se acreditava que a radiação eletromagnética se propagava. (Quando o físico britânico do século XIX James Clerk Maxwell descobriu que a radiação eletromagnética poderia ser descrita por uma equação de onda, os cientistas acharam que o espaço deveria estar repleto de algum meio etéreo em que a radiação estava se propagando. Somente depois do famoso experimento de Michelson-Morley, em 1887, é que os físicos descobriram que a radiação eletromagnética podia se propagar no vácuo.)

A rede Ethernet da Xerox foi tão bem-sucedida que DEC, Intel e Xerox chegaram a um padrão em 1978 para

uma Ethernet de 10 Mbps, chamado **padrão DIX**. Com uma pequena mudança, o padrão DIX tornou-se o padrão IEEE 802.3 em 1983. Infelizmente para a Xerox, ela já tinha um histórico de criar invenções originais (como o computador pessoal) e não conseguir comercializá-las, história contada em *Fumbling the Future* [Tateando o futuro], de Smith e Alexander (1988). Quando a Xerox mostrou pouco interesse em fazer algo com a Ethernet além de ajudar a padronizá-la, Metcalfe formou sua própria empresa, a 3Com, para vender adaptadores Ethernet para PCs. Ele vendeu muitos milhões deles.

A Ethernet clássica percorria o prédio como um cabo longo único, ao qual todos os computadores eram conectados. Essa arquitetura é mostrada na Figura 4.13. A primeira variedade, popularmente conhecida como **thick Ethernet**, era semelhante a uma mangueira amarela de jardim, com marcações a cada 2,5 m, mostrando onde conectar os computadores. (O padrão 802.3 não *exigia* realmente que o cabo fosse amarelo, mas *sugeria* isso.) Ela foi acompanhada pela **thin Ethernet**, que encurvava com mais facilidade e fazia conexões usando conectores BNC padrão da indústria. A thin Ethernet era muito mais barata e fácil de instalar, mas só podia ter 185 m por segmento (em vez dos 500 m da thick Ethernet), cada um dos quais podendo lidar com apenas 30 máquinas (em vez de 100).

Cada versão da Ethernet tem um comprimento máximo de cabo por segmento (ou seja, comprimento não amplificado) sobre o qual o sinal será propagado. Para permitir redes maiores, vários cabos podem ser conectados por **repetidores**. Um repetidor é um dispositivo da camada física que recebe, amplifica (ou seja, regenera) e retransmite sinais nas duas direções. Em relação ao software, diversos segmentos de cabo conectados por repetidores não são diferentes de um único cabo (exceto por um pequeno atraso introduzido pelos repetidores).

Por um a um desses cabos, a informação era enviada usando a codificação Manchester que estudamos na Seção 2.4.3. Uma Ethernet poderia conter vários segmentos de cabo e vários repetidores, mas dois transceptores não poderiam estar mais de 2,5 km afastados um do outro e nenhum caminho entre dois transceptores quaisquer poderia atravessar mais de quatro repetidores. O motivo para essa restrição foi para que o protocolo MAC, que examinaremos em seguida, funcionasse corretamente.

4.3.2 O protocolo da subcamada MAC Ethernet clássica

O formato usado para transmitir quadros é mostrado na Figura 4.14. Cada quadro começa com um *Preâmbulo* de 8 bytes, cada um contendo o padrão de bits 10101010 (com exceção do último byte, em que os dois últimos bits são 11). Esse último byte é chamado de delimitador de *Início de quadro* para o 802.3. A codificação Manchester desse padrão produz uma onda quadrada de 10 MHz por 6,4 μs, a fim de permitir a sincronização entre o clock do receptor e o clock do transmissor. Os dois últimos bits 1 dizem ao receptor que o restante do quadro está para começar.

Em seguida, o quadro contém dois endereços, um para o destino e um para a origem. Cada um deles possui 6 bytes de extensão. O primeiro bit transmitido do endereço de destino é 0 para endereços comuns e 1 para endereços de grupos. Estes permitem que diversas estações escutem um único endereço. Quando um quadro é enviado para um

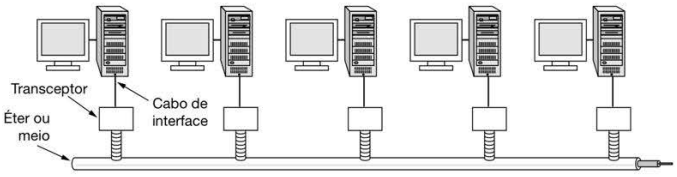


Figura 4.13 Arquitetura da Ethernet clássica.

Figura 4.14 Formato dos quadros. (a) Ethernet (DIX). (b) IEEE 802.3

endereço de grupo, todas as estações do grupo o recebem. A transmissão para um grupo de estações é chamada de **multicasting**. O endereço que consiste em todos os bits 1 é reservado para **broadcasting**. Um quadro contendo todos os bits 1 no campo de destino é aceito por todas as estações da rede. O multicasting é mais seletivo, mas envolve o gerenciamento de grupos para definir quais estações pertencem ao grupo. Por sua vez, o broadcasting não diferencia entre estação alguma e, por isso, não requer qualquer gerenciamento de grupos.

Uma característica interessante dos endereços de origem da estação é que eles são globalmente exclusivos, atribuídos de forma centralizada pelo IEEE para garantir que duas estações em qualquer lugar do mundo nunca tenham o mesmo endereço. A ideia é que qualquer estação possa endereçar de forma exclusiva qualquer outra estação simplesmente informando o número de 48 bits correto. Para fazer isso, os três primeiros bytes do campo de endereço são usados para um **identificador exclusivo da organização**, ou **OUI** (**Organizationally Unique Identifier**). Os valores para esse campo são atribuídos diretamente pelo IEEE e indicam o fabricante. Os fabricantes recebem blocos de 2^{24} endereços. O fabricante atribui os três últimos bytes do endereço e programa o endereço completo na NIC antes que ela seja vendida.

Em seguida, vem o campo *Tipo* ou *Tamanho*, dependendo se o quadro é Ethernet ou IEEE 802.3. A Ethernet usa um campo *Tipo* para informar ao receptor o que fazer com o quadro. Vários protocolos da camada de rede podem estar em uso ao mesmo tempo na mesma máquina; assim, quando chega um quadro Ethernet, o sistema operacional tem de saber a qual deles deve entregar o quadro. O campo *Tipo* especifica que processo deve receber o quadro. Por exemplo, um código tipo 0x0800 significa que os dados contêm um pacote IPv4.

O IEEE 802.3, em sua sabedoria, decidiu que esse campo transportaria o tamanho do quadro, pois o tamanho Ethernet era determinado examinando o interior dos dados – uma violação do uso de camadas, se é que isso existiu. Naturalmente, isso significava que não havia como o receptor descobrir o que fazer com um quadro que chegava. Esse problema foi tratado pelo acréscimo de outro cabeçalho para o protocolo de controle lógico do enlace dentro dos dados, que veremos mais adiante. Ele usa 8 bytes para transportar os 2 bytes de informação do tipo de protocolo.

Infelizmente, quando o 802.3 foi publicado, já havia tanto hardware e software para a Ethernet DIX em uso que poucos fabricantes e usuários tiveram interesse em modificar os campos de *Tipo* e *Tamanho*. Em 1997, o IEEE jogou a toalha e disse que as duas maneiras poderiam ser usadas. Felizmente, todos os campos de *Tipo* em uso antes de 1997 tinham valores maiores que 1500, que ficou bem estabelecido como o tamanho máximo dos dados. Agora, a regra é que qualquer número menor ou igual a 0x600 (1536) pode ser interpretado como *Tamanho*, ao passo que qualquer número maior que 0x600 pode ser interpretado como *Tipo*. Agora o IEEE pode afirmar que todos estão usando seu padrão e todos os outros podem continuar fazendo o que já faziam (não se preocupar com o LLC [logical link control]) sem se sentir culpados por isso. Isso é o que acontece quando a política (industrial) encontra a tecnologia.

Depois, vêm os dados, com até 1.500 bytes. Esse limite foi escolhido de forma um tanto arbitrária na época em que o padrão Ethernet foi esculpido, principalmente com base no fato de que um transceptor precisa ter RAM suficiente para guardar um quadro inteiro e, em 1978, a RAM tinha um custo muito alto. Um limite superior maior significaria mais RAM e, consequentemente, um transceptor mais caro.

Além de haver um comprimento máximo de quadro, também existe um comprimento mínimo. Embora um campo de dados de 0 bytes às vezes seja útil, ele causa um problema. Quando um transceptor detecta uma colisão, ele trunca o quadro atual, o que significa que bits perdidos e fragmentos de quadros aparecem a todo instante no cabo. Para tornar mais fácil a distinção entre quadros válidos e lixo, o padrão Ethernet exige que os quadros válidos tenham pelo menos 64 bytes de extensão, do endereço de destino até o campo de checksum, incluindo ambos. Se a parte de dados de um quadro for menor que 46 bytes, o campo *Preenchimento* será usado para preencher o quadro até o tamanho mínimo.

Outra (e mais importante) razão para a existência de um quadro de comprimento mínimo é impedir que uma estação conclua a transmissão de um quadro curto antes de o primeiro bit ter atingido a outra extremidade do cabo, em que ele poderá colidir com outro quadro. Esse problema é ilustrado na Figura 4.15. No tempo 0, a estação *A*, localizada em uma extremidade da rede, envia um quadro. Vamos chamar de τ o tempo de propagação que ele leva para atingir a outra extremidade. Momentos antes de o quadro chegar à outra extremidade (ou seja, no tempo $\tau - \varepsilon$), a estação mais distante, *B*, inicia a transmissão. Quando detecta que está recebendo mais potência do que está transmitindo, *B* sabe que ocorreu uma colisão, interrompe a transmissão e gera uma rajada de sinal ruidoso de 48 bits para avisar as demais estações. Em outras palavras, ela bloqueia o éter (meio) para ter certeza de que o transmissor não ignorará a colisão. Aproximadamente no tempo 2τ, o transmissor detecta a rajada de ruídos e também cancela sua transmissão. Depois, ele espera por um tempo aleatório antes de tentar novamente.

Se uma estação tentar transmitir um quadro muito curto, é concebível que ocorra uma colisão, mas a transmissão será concluída antes que a rajada de sinal ruidoso retorne no instante 2τ. Então, o transmissor concluirá incorretamente que o quadro foi enviado com êxito. Para evitar que essa situação ocorra, a transmissão de todos os quadros deve

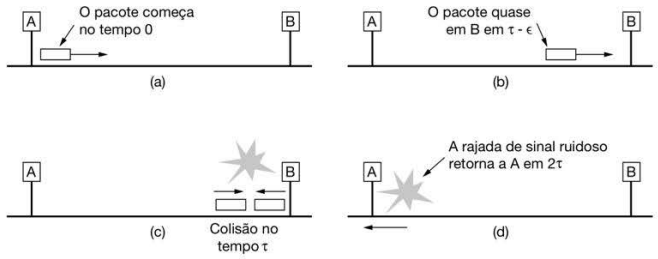


Figura 4.15 A detecção de colisão pode demorar até o tempo 2τ.

demorar mais de 2τ para transmitir, de forma que a transmissão ainda esteja acontecendo quando a rajada de sinal ruidoso voltar ao transmissor. Para uma LAN de 10 Mbps com um comprimento máximo de 2.500 m e quatro repetidores (de acordo com a especificação 802.3), o tempo de ida e volta (incluindo o tempo de propagação pelos quatro repetidores) foi calculado em quase 50 μs na pior das hipóteses. Portanto, o quadro mínimo deve demorar pelo menos esse tempo para ser transmitido. A 10 Mbps, um bit demora 100 ns, e assim 500 bits é o menor tamanho de quadro que oferece a garantia de funcionar. Para acrescentar certa margem de segurança, esse número foi arredondado para 512 bits ou 64 bytes.

O último campo Ethernet é o *Checksum*. Ele é um CRC de 32 bits do tipo que estudamos na Seção 3.2. De fato, é definido exatamente pelo polinômio gerador que mostramos lá, que apareceu para PPP, ADSL e outros enlaces também. Esse CRC é um código de detecção de erro usado para determinar se os bits do quadro foram recebidos corretamente. Ele simplesmente realiza a detecção de erros, com o quadro sendo descartado se algum erro for detectado.

CSMA/CD com backoff exponencial binário

A Ethernet clássica utiliza o algoritmo CSMA/CD 1-persistente que estudamos na Seção 4.2. Esse descritor só significa que as estações sentem o meio quando elas têm um quadro para transmitir e o enviam assim que o meio se torna desocupado. Elas monitoram o canal em busca de colisões enquanto transmitem. Se houver uma colisão, elas cancelam a transmissão com um curto sinal de interferência e retransmitem após um intervalo aleatório.

Vejamos agora como é determinado o intervalo aleatório quando ocorre uma colisão, pois esse é um método novo. O modelo ainda é o da Figura 4.5. Depois de uma colisão, o tempo é dividido em slots discretos, cujo comprimento é igual ao pior tempo de propagação da viagem de ida e volta no éter (2τ). Para acomodar o caminho mais longo permitido pelo padrão Ethernet, o tempo de duração do slot foi definido como 512 períodos de duração de um bit, ou 51,2 μs.

Depois da primeira colisão, cada estação espera 0 ou 1 tempo de slot aleatoriamente antes de tentar novamente. Se duas estações colidirem e selecionarem o mesmo número aleatório, elas colidirão novamente. Depois da segunda colisão, cada uma seleciona ao acaso 0, 1, 2 ou 3 e aguarda durante esse número de tempos de slot. Se ocorrer uma terceira colisão (cuja probabilidade é de 0,25), na próxima vez o número de slots que a estação deverá esperar será escolhido ao acaso no intervalo de 0 a $2^3 - 1$.

Em geral, depois de i colisões, é escolhido um número aleatório entre 0 e $2^i - 1$, e esse número de slots será ignorado. Entretanto, após terem sido alcançadas dez colisões, o intervalo de randomização será congelado em um máximo de 1.023 slots. Depois de 16 colisões, o controlador desiste e informa o erro ao computador. Qualquer recuperação adicional caberá às camadas superiores.

Esse algoritmo, chamado **backoff exponencial binário**, foi escolhido para se adaptar dinamicamente ao número de estações que estão tentando transmitir. Se o intervalo de escolha do número aleatório para todas as colisões fosse 1023, o risco de duas estações colidirem uma segunda vez seria desprezível, mas o tempo de espera médio depois de uma colisão seria de centenas de períodos de slot, ocasionando um atraso significativo. Em contrapartida, se cada estação sempre esperasse entre 0 ou 1 slot, e se 100 estações tentassem transmitir ao mesmo tempo, elas colidiriam repetidas vezes até que 99 delas escolhessem 1 e a estação restante escolhesse 0. Isso poderia levar anos. Aumentando-se exponencialmente o intervalo de tempo aleatoriamente, à medida que ocorre um número cada vez maior de colisões consecutivas, o algoritmo assegura um baixo atraso quando apenas algumas estações colidem, mas também garante que a colisão seja resolvida em um período de tempo razoável quando muitas estações colidirem. A restrição de recuo a 1023 impede que o limite aumente demais.

Se não houver colisão, o transmissor considera que o quadro provavelmente foi entregue com êxito. Ou seja,

nem CSMA/CD nem Ethernet oferecem confirmações. Essa escolha é apropriada para canais com fio e de fibra óptica, que têm baixas taxas de erro. Quaisquer erros que ocorram devem então ser detectados pelo CRC e recuperados pelas camadas mais altas. Para canais sem fio, que têm mais erros, veremos que as confirmações realmente são utilizadas.

4.3.3 Desempenho da Ethernet

Agora, vamos examinar rapidamente o desempenho da Ethernet sob condições de carga alta e constante, ou seja, k estações sempre prontas a transmitir. Uma análise completa do algoritmo de backoff exponencial binário é muito complicada. Em vez disso, seguiremos Metcalfe e Boggs (1976) e vamos supor uma probabilidade de retransmissão constante em cada slot. Se cada estação transmitir durante um slot de disputa com probabilidade p, a probabilidade A de que alguma estação tome posse do canal existente nesse slot será:

$$A = kp(1-p)^{k-1}$$

A é maximizado quando $p = 1/k$, com $A \to 1/e$, à medida que $k \to \infty$. A probabilidade de que o intervalo de disputa tenha exatamente j slots é $A(1-A)^{j-1}$, de forma que o número médio de slots por disputa é dado por

$$\sum_{j=0}^{\infty} jA(1-A)^{j-1} = \frac{1}{A}$$

Como cada slot tem uma duração de 2τ, o intervalo médio de disputa, w, é $2\tau/A$. Supondo-se um valor ideal para p, o número médio de slots de disputa nunca será maior que e; portanto, w é, no máximo, $2\tau e \approx 5{,}4\tau$.

Se o quadro leva em média P segundos para ser transmitido, quando muitas estações tiverem quadros para enviar,

$$\text{Eficiência do canal} = \frac{P}{P + 2\tau/A} \qquad (4.2)$$

Aqui, vemos onde a distância máxima do cabo entre duas estações entra nos números do desempenho. Quanto maior for o cabo, maior será o intervalo de disputa, o que explica por que o padrão Ethernet especifica um comprimento máximo de cabo.

É instrutivo formular a Equação 4.2 em termos do comprimento do quadro, F, da largura de banda da rede, B, do comprimento do cabo, L, e da velocidade de propagação do sinal, c, para o caso ideal de e slots de disputa por quadro. Com $P = F/B$, a Equação 4.2 passa a ser:

$$\text{Eficiência do canal} = \frac{1}{1 + 2BLe/cF} \qquad (4.3)$$

Quando o segundo termo no denominador for grande, a eficiência da rede será baixa. Mais especificamente, aumentar a largura de banda da rede ou a distância (o produto BL) reduz a eficiência para determinado tamanho de quadro. Infelizmente, a maior parte das pesquisas em hardware de rede visa exatamente ao aumento desse produto. As pessoas querem alta largura de banda a longas distâncias (MANs de fibra óptica, p. ex.), o que sugere que o padrão Ethernet implementado dessa maneira talvez não seja o melhor sistema para essas aplicações. Veremos outras formas de implementar a Ethernet na próxima seção.

Na Figura 4.16, a eficiência do canal é representada contra o número de estações prontas para $2\tau = 51{,}2\ \mu$s e uma taxa de dados de 10 Mbps, usando-se a Equação 4.3. Com um tempo por slot de 64 bytes, não surpreende que quadros de 64 bytes não sejam eficientes. Por sua vez, com quadros de 1.024 bytes e um valor assintótico de e slots de 64 bytes por intervalo de disputa, o período de disputa é de 174 bytes e a eficiência é de 85%. Esse resultado é muito melhor do que os 37% de eficiência da slotted ALOHA.

Talvez valha a pena mencionar que houve um grande número de análises teóricas sobre o desempenho da Ethernet (e de outras redes). A maioria dos resultados deve ser considerada com certa cautela (talvez muita cautela) por duas razões. Primeiro, praticamente todos esses trabalhos presumem que o tráfego obedece a uma série de Poisson. Quando os pesquisadores começaram a analisar dados reais, eles descobriram que o tráfego de rede raras vezes é de Poisson, e sim semelhante ou em forma de rajada em um intervalo escalonado de tempo (Paxson e Floyd, 1995; e Fontugne et al., 2017). Isso significa que calcular uma média durante intervalos longos não suaviza o tráfego. Assim como no uso de modelos questionáveis, muitas das análises focam nos casos de desempenho "interessantes" para carga estranhamente alta. Boggs et al. (1988) mostraram, com

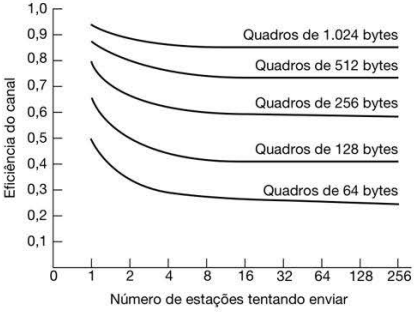


Figura 4.16 Eficiência da Ethernet a 10 Mbps com tempos por slot de 512 bits.

experimentos, que a Ethernet funciona bem na realidade, até mesmo com carga moderadamente alta.

4.3.4 Ethernet comutada

A Ethernet logo começou a evoluir para longe da arquitetura de cabo longo único da Ethernet clássica (o éter). Os problemas associados a encontrar interrupções ou conexões partidas a levaram para um tipo diferente de padrão de fiação, em que cada estação tem um cabo dedicado esticado até um **hub** central. Um hub simplesmente conecta todos os fios eletricamente, como se eles fossem únicos. Essa configuração pode ser vista na Figura 4.17(a).

Os fios eram pares trançados da companhia telefônica, pois a maioria dos prédios de escritórios já estava conectada dessa forma e normalmente havia muita capacidade ociosa disponível. Esse reúso foi um ganho, mas reduziu o tamanho máximo do cabo do hub para 100 m (200 m, se fossem usados pares trançados de alta qualidade da Categoria 5). A inclusão ou remoção de uma estação é mais simples nessa configuração, e uma interrupção de cabo pode ser facilmente detectada. Com a vantagem de elas serem capazes de usar a fiação existente e a facilidade de manutenção, os hubs de par trançado rapidamente se tornaram a forma dominante na topologia Ethernet.

Todavia, os hubs não aumentam a capacidade, pois são logicamente equivalentes ao cabo longo e único da Ethernet clássica. Quando mais e mais estações são acrescentadas, cada estação recebe uma fatia cada vez menor da capacidade fixa. Por fim, a LAN saturará. Uma saída é usar uma velocidade maior, digamos, de 10 Mbps para 100 Mbps, 1 Gbps ou velocidades ainda maiores. Mas, com o crescimento da multimídia e de servidores poderosos, até mesmo a Ethernet de 1 Gbps pode ficar saturada.

Felizmente, existe outra solução para lidar com o aumento da carga: a Ethernet comutada. O núcleo desse sistema é um **switch**, que contém uma placa integrada (ou *backplane*) de alta velocidade que conecta todas as portas, como mostra a Figura 4.17(b). Por fora, um switch se parece com um hub. Ambos são caixas, normalmente com 4 a 48 portas, cada uma contendo um conector RJ-45 padrão para um cabo de par trançado. Cada cabo conecta o switch ou hub a um único computador, como mostra a Figura 4.18. Um switch também tem as mesmas vantagens de um hub. É muito fácil acrescentar ou remover uma nova estação conectando ou desconectando um fio, e é fácil encontrar a maioria das falhas, pois um cabo ou porta com defeito normalmente afetará apenas uma estação. Ainda existe um componente compartilhado que pode falhar – o próprio switch –, mas, se todas as estações perderem conectividade, o pessoal de TI sabe o que fazer para resolver o problema: substituir o switch inteiro.

Dentro do switch, porém, algo muito diferente está acontecendo. Os switches só enviam quadros às portas para as quais esses quadros são destinados. Quando uma porta do switch recebe um quadro Ethernet de uma estação, o switch verifica os endereços Ethernet para saber para qual porta o quadro se destina. Essa etapa requer que o switch descubra quais portas correspondem a quais endereços, um processo que explicaremos na Seção 4.8, quando analisarmos o caso geral dos switches conectados a outros switches. Por enquanto, basta considerar que o switch

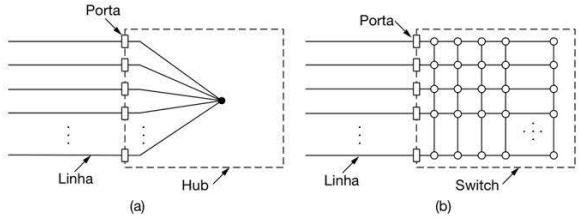


Figura 4.17 (a) Hub. (b) Switch.

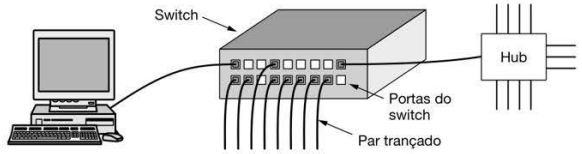


Figura 4.18 Um switch Ethernet.

conhece a porta de destino do quadro. Depois, ele encaminha o quadro por sua placa interna de alta velocidade até a porta de destino. A placa interna normalmente trabalha com muitos Gbps, usando um protocolo próprio que não precisa ser padronizado, pois fica inteiramente oculto dentro do switch. A porta de destino, então, transmite o quadro no fio para que ele alcance a estação intencionada. Nenhuma das outras portas sequer saberá que o quadro existe.

O que acontecerá se duas estações ou portas quiserem transmitir um quadro ao mesmo tempo? Novamente, os switches diferem dos hubs. Em um hub, todas as estações estão no mesmo **domínio de colisão**. Elas precisam usar o algoritmo CSMA/CD para programar suas transmissões. Em um switch, cada porta é seu próprio domínio de colisão independente. No caso comum de um cabo full-duplex, a estação e a porta podem enviar um quadro no cabo ao mesmo tempo, sem se preocupar com outras portas e estações. As colisões agora são impossíveis e o CSMA/CD não é necessário. Contudo, se o cabo for half-duplex, a estação e a porta precisam disputar com CSMA/CD pela transmissão, de forma normal.

Um switch melhora o desempenho em relação a um hub de duas maneiras. Primeiro, como não existem colisões, a capacidade é usada de modo mais eficiente. Segundo, e mais importante, com um switch, vários quadros podem ser enviados simultaneamente (por estações diferentes). Esses quadros alcançarão as portas do switch e trafegarão pela placa integrada do switch para ser enviados nas portas apropriadas. Todavia, como dois quadros podem ser enviados para a mesma porta de saída ao mesmo tempo, o switch precisa ter um buffer, para que possa temporariamente enfileirar um quadro de entrada até que ele possa ser transmitido para a porta de saída. Em geral, essas melhorias dão um grande ganho de desempenho, o que não é possível com um hub. O throughput total do sistema normalmente pode ser aumentado em uma ordem de grandeza, dependendo do número de portas e padrões de tráfego.

A mudança nas portas em que os quadros são enviados também tem benefícios para a segurança. A maioria das interfaces de LAN possui um **modo promíscuo**, em que *todos* os quadros são dados a cada computador, não apenas os endereçados a ele. Com um hub, cada computador conectado pode ver o tráfego enviado entre todos os outros computadores. Espiões adoram esse recurso. Com um switch, o tráfego é encaminhado apenas para as portas às quais ele é destinado. Essa restrição oferece melhor isolamento, de modo que o tráfego não escapará com facilidade nem cairá em mãos erradas. Contudo, é melhor criptografar o tráfego se a segurança realmente for necessária.

Tendo em vista que o switch espera apenas quadros Ethernet padrão em cada porta de entrada, é possível usar algumas dessas portas como concentradoras. Na Figura 4.18, a porta localizada no canto superior direito não está conectada a uma estação isolada, mas a um hub de 12 portas. À medida que chegam ao hub, os quadros disputam a rede Ethernet normalmente, inclusive com colisões e backoff exponencial binário. Os quadros bem-sucedidos são enviados ao switch e são tratados como quaisquer outros quadros recebidos. O switch não sabe que eles tiveram de brigar para chegar lá. Uma vez no switch, eles são enviados para a linha de saída correta pela placa integrada de alta velocidade. Também é possível que o destino correto fosse uma das linhas conectadas ao hub, quando o quadro já foi entregue, de modo que o switch simplesmente o descarta. Os hubs são mais simples e mais baratos que os switches, mas, em decorrência da queda nos preços dos switches, eles estão rapidamente se tornando espécies em extinção. As redes modernas usam Ethernet comutada quase exclusivamente. Apesar disso, ainda existem hubs legados.

4.3.5 Fast Ethernet

Ao mesmo tempo em que os switches estavam se tornando populares, a velocidade da Ethernet de 10 Mbps estava sendo pressionada. A princípio, 10 Mbps parecia ser o paraíso, da mesma forma que os modems a cabo pareciam ser o paraíso para os usuários de modems telefônicos de 56 kbps. Todavia, o que era novidade se dissipou com rapidez. Como uma espécie de corolário da Lei de Parkinson ("O trabalho se expande até preencher o tempo disponível para sua conclusão"), parecia que os dados se expandiam para preencher toda a largura de banda disponível para sua transmissão.

Muitas instalações precisavam de maior largura de banda e tinham diversas LANs de 10 Mbps conectadas por um labirinto de repetidores, hubs e switches, embora às vezes parecesse, para os administradores de redes, que elas estavam conectadas por goma de mascar e tela de arame. Contudo, até mesmo com switches Ethernet, a largura de banda máxima de um único computador era limitada pelo cabo que o conectava à porta do switch.

Foi nesse ambiente que o IEEE reuniu o comitê do 802.3 em 1992, com instruções para produzir uma LAN mais rápida. Uma das propostas era manter o 802.3 exatamente como estava, apenas tornando-o mais rápido. Outra proposta era refazê-lo completamente, para integrar um grande número de novos recursos, como tráfego em tempo real e voz digitalizada, mas manter o antigo nome (por motivos de marketing). Após alguma discussão, o comitê decidiu manter o 802.3 como ele era, simplesmente tornando-o mais rápido. Essa estratégia realizaria o trabalho antes que a tecnologia mudasse, evitando problemas não previstos com um projeto totalmente novo. O novo projeto também seria compatível com as LANs Ethernet existentes. As pessoas que apoiavam a proposta perdedora fizeram o que qualquer pessoa do setor de informática que se preza faria nessas circunstâncias – formaram seu próprio comitê e padronizaram sua LAN assim mesmo (como o padrão 802.12). Esse padrão fracassou por completo.

O trabalho foi feito rapidamente (pelas normas dos comitês de padronização) e o resultado, o 802.3u, foi oficialmente aprovado pelo IEEE em junho de 1995. Tecnicamente, o 802.3u não é um padrão novo, mas um adendo ao padrão 802.3 existente (para enfatizar sua compatibilidade). Essa estratégia é muito utilizada. Como praticamente todos o chamam de **Fast Ethernet**, em vez de 802.3u, faremos o mesmo.

A ideia básica por trás da Fast Ethernet era simples: manter os antigos formatos de quadros, interfaces e regras de procedimentos, e apenas reduzir o tempo de bit de 100 ns para 10 ns. Tecnicamente, teria sido possível copiar a Ethernet clássica de 10 Mbps e continuar a detectar colisões a tempo, pela simples redução do comprimento máximo do cabo a um décimo do comprimento original. Entretanto, as vantagens do cabeamento de par trançado eram tão grandes que a Fast Ethernet se baseou inteiramente nesse projeto. Por isso, todos os sistemas Fast Ethernet usam hubs e switches; porém, cabos multiponto com conectores de pressão ou conectores BNC não são permitidos.

Entretanto, algumas decisões ainda precisavam ser tomadas, sendo que a mais importante dizia respeito aos tipos de fios que seriam aceitos. Um dos concorrentes era o par trançado da Categoria 3. O argumento a favor dele era que todo escritório do mundo ocidental tinha pelo menos quatro pares trançados da Categoria 3 (ou melhor) instalados entre ele e um armário de fiação telefônica a uma distância máxima de 100 m. Às vezes, há dois cabos desse tipo. Desse modo, o uso do par trançado da Categoria 3 tornaria possível conectar computadores desktop usando a Fast Ethernet, sem a necessidade de refazer a fiação do prédio, uma enorme vantagem para muitas empresas.

A principal desvantagem do par trançado da Categoria 3 é sua incapacidade para transportar sinais de 100 Mbps por 100 m, a distância máxima especificada entre o computador e o hub para hubs de 10 Mbps. Por sua vez, a fiação de par trançado da Categoria 5 é capaz de tratar 100 m com facilidade, e a fibra pode ir muito mais longe que isso. Decidiu-se permitir as três possibilidades, como mostra a Figura 4.19, mas incentivar a solução da Categoria 3, para que fosse possível obter a capacidade de transporte adicional necessária.

O esquema de par trançado sem blindagem, ou UTP (Unshielded Twisted Pair), da Categoria 3, formalmente chamado **100Base-T4**, emprega uma velocidade de sinalização de 25 MHz, somente 25% mais rápida do que os 20 MHz da Ethernet padrão. (Lembre-se de que a codificação Manchester, discutida na Seção 2.4.3, requer dois períodos de clock para cada um dos 10 milhões de bits enviados a cada segundo.) Contudo, para atingir a taxa de bits necessária, o 100Base-T4 exige quatro pares trançados. Dos quatro pares, um sempre é para o hub, um sempre é do hub e os outros dois são comutáveis para a direção da transmissão atual. Para conseguir 100 Mbps dos três pares trançados na direção da transmissão, um esquema bastante complicado é usado em cada par trançado. Ele envolve o envio de dígitos ternários com três níveis de tensão. Esse esquema provavelmente não ganhará nenhum prêmio de elegância, e (felizmente) deixaremos de lado os detalhes. Todavia, como a fiação da telefonia padrão há décadas tem quatro pares por cabo, a maioria dos escritórios é capaz de usar a fiação existente. É claro que isso significa abrir mão do telefone do seu escritório, mas esse certamente é um pequeno preço a pagar por um e-mail mais rápido.

O 100Base-T4 foi deixado de lado quando muitos prédios de escritórios tiveram a fiação trocada para o UTP de Categoria 5 para Ethernet **100Base-TX**, que veio para dominar o mercado. Esse projeto é mais simples porque os fios podem lidar com taxas de clock de 125 MHz. Somente dois pares trançados por estação são usados, um para o hub e outro a partir dele. Nem a codificação binária direta (ou seja, NRZ) nem a codificação Manchester são usadas. Em vez disso, é usada a codificação **4B/5B**, que descrevemos na Seção 2.4.3. Quatro bits de dados são codificados como 5 bits de sinal e enviados a 125 MHz para fornecer 100 Mbps. Esse esquema é simples, mas tem transições suficientes para sincronização e usa a largura de banda do fio relativamente bem. O sistema 100Base-TX é full-duplex; as estações podem transmitir a 100 Mbps em um par trançado e recebem em 100 Mbps em outro par trançado ao mesmo tempo.

A última opção, o **100Base-FX**, utiliza dois filamentos de fibra multimodo, um para cada sentido; por isso, ele também é full-duplex, com 100 Mbps em cada sentido. Nessa configuração, a distância entre uma estação e o switch pode ser de até 2 km.

A Fast Ethernet permite a interconexão por hubs ou switches. Para garantir que o algoritmo CSMA/CD

Nome	Cabo	Tam. máx. de segmento	Vantagens
100Base-T4	Par trançado	100 m	Utiliza UTP da Categoria 3
100Base-TX	Par trançado	100 m	Full-duplex a 100 Mbps (UTP Cat. 5)
100Base-FX	Fibra óptica	2.000 m	Full-duplex a 100 Mbps; grandes distâncias

Figura 4.19 O cabeamento Fast Ethernet original.

continue a funcionar, o relacionamento entre o tamanho de quadro mínimo e o tamanho de cabo máximo deve ser mantido enquanto a velocidade da rede sobe de 10 Mbps para 100 Mbps. Assim, ou o comprimento mínimo do quadro de 64 bytes deve aumentar ou o comprimento máximo do cabo de 2.500 m deve diminuir proporcionalmente. A escolha fácil foi que a distância máxima entre duas estações quaisquer fosse diminuída por um fator de 10, pois um hub com cabos de 100 m já está dentro desse novo máximo. Contudo, os cabos 100Base-FX de 2 km são muito longos para aceitar um hub de 100 Mbps com o algoritmo de colisão normal da Ethernet. Esses cabos, em vez disso, precisam ser conectados a um switch e operar em um modo full-duplex para que não haja colisões.

Os usuários rapidamente começaram a implantar a Fast Ethernet, mas eles não quiseram abandonar as placas Ethernet de 10 Mbps nos computadores mais antigos. Por conseguinte, praticamente todos os switches Ethernet podem lidar com uma mistura de estações de 10 Mbps e 100 Mbps. Para facilitar o upgrading, o próprio padrão oferece um mecanismo chamado **autonegociação**, que permite que duas estações negociem automaticamente a velocidade ideal (10 ou 100 Mbps) e o tipo de duplex (half ou full). Isso quase sempre funciona bem, mas pode ocasionar problemas de divergência do duplex quando uma extremidade do enlace autonegocia e a outra não, ficando definida como o modo full-duplex (Shalunov e Carlson, 2005). A maioria dos produtos Ethernet utiliza esse recurso para se configurar.

4.3.6 Gigabit Ethernet

A tinta mal havia secado no padrão Fast Ethernet quando o comitê 802 começou a trabalhar em uma Ethernet ainda mais rápida, prontamente apelidada de **gigabit Ethernet**. O IEEE ratificou a forma mais popular como 802.3ab em 1999. A seguir descreveremos algumas das principais características da gigabit Ethernet. Você pode encontrar mais informações em Spurgeon e Zimmerman (2014).

Os objetivos do comitê para a gigabit Ethernet eram essencialmente os mesmos do comitê para a Fast Ethernet: torná-la 10 vezes mais rápida, mantendo a compatibilidade com todos os padrões Ethernet existentes. Em particular, a gigabit Ethernet tinha de oferecer o serviço de datagrama não confirmado com unicasting e broadcasting, empregar o mesmo esquema de endereçamento de 48 bits já em uso e manter o mesmo formato de quadro, inclusive seus tamanhos mínimo e máximo. O padrão final atendeu a todos esses objetivos.

Também como a Fast Ethernet, todas as configurações da gigabit Ethernet utilizam enlaces ponto a ponto. Na configuração mais simples, ilustrada na Figura 4.20(a), dois computadores estão diretamente conectados um ao outro. Contudo, o caso mais comum consiste em um switch ou um hub conectado a vários computadores e possivelmente a switches ou hubs adicionais, como mostra a Figura 4.20(b). Em ambas as configurações, cada cabo Ethernet tem exatamente dois dispositivos conectados a ele, nem mais nem menos.

Assim como a Fast Ethernet, a gigabit Ethernet admite dois modos de operação: o full-duplex e o half-duplex. O modo "normal" é o full-duplex, que permite tráfego em ambos os sentidos ao mesmo tempo. Ele é usado quando existe um switch central conectado a computadores (ou outros switches) na periferia. Nessa configuração, todas as linhas têm buffers de armazenamento, de forma que cada computador e cada switch são livres para enviar quadros sempre que quiserem. O transmissor não precisa observar o canal para saber se ele está sendo usado por mais alguém, pois a disputa é impossível. Na linha entre um computador e um switch, o computador é o único transmissor possível para o switch naquela linha, e a transmissão terá sucesso ainda que o switch nesse instante esteja transmitindo um quadro para o computador (porque a linha é full-duplex). Tendo em vista que nenhuma disputa é possível, o protocolo CSMA/CD não é usado, e assim o comprimento máximo do cabo é determinado pela intensidade do sinal, não pelo tempo que uma rajada de sinal

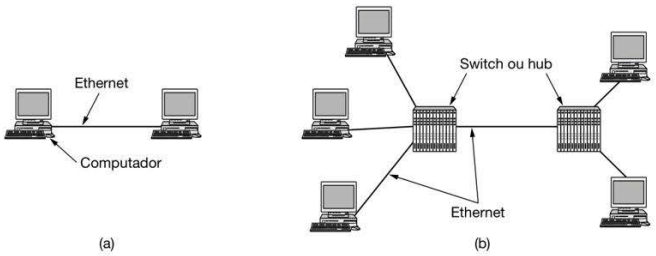


Figura 4.20 (a) Uma Ethernet com duas estações. (b) Uma Ethernet com várias estações.

ruidoso leva para se propagar de volta até o transmissor na pior das hipóteses. Os switches são livres para se misturar e combinar suas velocidades. A autonegociação é admitida, como na Fast Ethernet, mas agora a escolha é entre 10, 100 e 1.000 Mbps.

O outro modo de operação, o half-duplex, é usado quando os computadores estão conectados a um hub, não a um switch. Um hub não armazena os quadros recebidos em buffers. Em vez disso, ele estabelece conexões elétricas internas para todas as linhas, simulando o cabo multiponto usado na Ethernet clássica. Nesse modo, colisões são possíveis e, portanto, é necessário o protocolo CSMA/CD padrão. Tendo em vista que um quadro mínimo de 64 bytes (o mais curto permitido) agora pode ser transmitido 100 vezes mais rápido que na Ethernet clássica, a distância máxima é 100 vezes menor (ou seja, 25 m), a fim de manter a propriedade essencial de que o transmissor ainda transmitirá quando uma rajada de sinal ruidoso voltar a ele, mesmo na pior das hipóteses. Com um cabo de 2.500 m, o transmissor de um quadro de 64 bytes a 1 Gbps terminaria a transmissão bem antes de o quadro sequer ter chegado a percorrer um décimo da distância até a outra extremidade, quanto mais ir até a extremidade e voltar.

Essa restrição de distância foi tão séria que duas características foram acrescentadas ao padrão para aumentar a distância máxima do cabo para 200 m, o que provavelmente é suficiente para a maioria dos escritórios. A primeira característica, chamada **extensão de portadora**, basicamente informa ao hardware para adicionar seu próprio preenchimento ao quadro normal, a fim de estendê-lo para 512 bytes. Considerando que esse preenchimento é adicionado pelo hardware transmissor e removido pelo hardware receptor, o software não tem conhecimento desse fato, o que significa que não é necessária qualquer mudança no software existente. A desvantagem é que o uso de 512 bytes de largura de banda para transmitir 46 bytes de dados do usuário (a carga útil de um quadro de 64 bytes) tem uma eficiência de linha de apenas 9%.

A segunda característica, chamada **rajada de quadros**, permite a um transmissor enviar uma sequência concatenada de vários quadros em uma única transmissão. Se a rajada total tiver menos de 512 bytes, o hardware a preencherá novamente. Se houver quadros suficientes esperando pela transmissão, esse esquema será altamente eficiente e preferível à extensão de portadora.

Com toda franqueza, é difícil imaginar uma organização se envolvendo com as dificuldades de compra e instalação de placas gigabit Ethernet para obter alto desempenho, e depois conectar os computadores a um antigo hub para simular a Ethernet clássica, com todas as suas colisões. As interfaces e os switches da gigabit Ethernet eram muito caros, mas seu preço caiu rapidamente à medida que o volume de vendas aumentou. Ainda assim, a compatibilidade é sagrada na indústria de informática e, então, o comitê é obrigado a aceitá-la. Hoje, a maioria dos computadores vem com uma interface Ethernet capaz de operar a 10, 100 e 1.000 Mbps (e talvez ainda mais), compatível com todas as velocidades.

A gigabit Ethernet admite cabeamento de cobre e de fibra, como mostra a Figura 4.21. A sinalização à velocidade de aproximadamente 1 Gbps requer a codificação e o envio de um bit a cada nanossegundo. Esse truque foi realizado inicialmente com cabos de cobre curtos e blindados (a versão 1000Base-CX) e fibras ópticas. Para estas, dois comprimentos de onda são permitidos e o resultado são duas versões: 0,85 mícron (curto, para 1000Base-SX) e 1,3 mícron (longo, para 1000Base-LX).

A sinalização no comprimento de onda curto pode ser alcançada com LEDs mais baratos. Ela é usada com a fibra multimodo e é útil para conexões dentro de um prédio, pois pode se estender até 500 m para a fibra de 50 micra. A sinalização no comprimento de onda longo exige lasers. Todavia, quando combinado com a fibra de modo único (10 micra), o comprimento do cabo pode ser de até 5 km. Esse limite permite conexões a longa distância entre prédios, como para o backbone de um campus, como em um enlace ponto a ponto dedicado. Outras variações do padrão permitiram enlaces ainda mais longos sobre a fibra de modo único.

Para enviar bits por essas versões da gigabit Ethernet, a codificação **8B/10B**, que descrevemos na Seção 2.4.3, foi emprestada de outra tecnologia de redes, chamada Fibre Channel. Esse esquema codifica 8 bits de dados em palavras de código de 10 bits, que são enviadas pelo fio ou fibra, daí o nome 8B/10B. As palavras de código foram escolhidas de modo que pudessem ser balanceadas (ou seja,

Nome	Cabo	Distância máxima do segmento	Vantagens
1000Base-SX	Fibra óptica	550 m	Fibra multimodo (50, 62,5 micra)
1000Base-LX	Fibra óptica	5.000 m	Modo único (10μ) ou multimodo (50, 62,5μ)
1000Base-CX	2 pares de STP	25 m	Par trançado blindado
1000Base-T	4 pares de UTP	100 m	UTP padrão da Categoria 5

Figura 4.21 O cabeamento da gigabit Ethernet.

tivessem o mesmo número de 0s e 1s) com transições suficientes para a recuperação de clock. O envio dos bits codificados com NRZ requer uma largura de banda de sinalização de 25% a mais do que a necessária para os bits não codificados, uma grande melhoria em relação à expansão de 100% da codificação Manchester.

Contudo, todas essas opções exigiam novos cabos de cobre ou fibra para dar suporte à sinalização mais rápida. Nenhum deles utilizava a grande quantidade de UTP de Categoria 5 que havia sido instalada com a Fast Ethernet. Dentro de um ano, o 1000Base-T surgiu para preencher essa lacuna, e tem sido a forma mais popular de gigabit Ethernet desde então. As pessoas aparentemente não quiseram mudar a fiação de seus prédios.

Uma sinalização mais complicada é necessária para fazer a Ethernet funcionar a 1.000 Mbps sobre fios de Categoria 5. Para começar, todos os quatro pares trançados no cabo são usados, e cada um é usado nas duas direções ao mesmo tempo, usando o processamento de sinal digital para separar os sinais. Pelos fios, um a um, cinco níveis de tensão que transportam 2 bits são usados para sinalizar em 125 Msímbolos/s. O mapeamento para produzir os símbolos a partir dos bits não é simples. Ele envolve embaralhamento e transições, seguidos por um código de correção de erros em que quatro valores são embutidos em cinco níveis de sinal.

Uma velocidade de 1 Gbps é bastante alta. Por exemplo, se um receptor estiver ocupado com alguma outra tarefa, mesmo durante 1 ms, e não esvaziar o buffer de entrada em alguma linha, nesse intervalo poderão se acumular até 1.953 quadros. Além disso, quando um computador em uma gigabit Ethernet estiver transmitindo dados pela linha a um computador em uma Ethernet clássica, serão muito prováveis sobrecargas no buffer. Como consequência dessas duas observações, a gigabit Ethernet admite controle de fluxo. O mecanismo consiste na transmissão de um quadro de controle especial de uma extremidade a outra, informando que a extremidade receptora deve fazer uma pausa durante algum período predeterminado. Para controle de fluxo, são usados quadros PAUSE, contendo o tipo 0x8808. As pausas são dadas em unidades de tempo mínimo por quadro. Para a gigabit Ethernet, a unidade de tempo é 512 ns, permitindo pausas de até 33,6 ms.

Existe mais uma extensão introduzida com a gigabit Ethernet. **Quadros jumbo** permitem que os quadros tenham mais de 1.500 bytes, normalmente até 9 KB. Essa extensão é patenteada. Ela não é reconhecida pelo padrão porque, se for usada, a Ethernet não será mais compatível com versões anteriores, mas, de qualquer forma, a maioria dos vendedores oferece suporte para ela. O raciocínio é que 1.500 bytes representam uma unidade curta nas velocidades de gigabit. Manipulando blocos de informação maiores, a taxa de quadros pode ser diminuída, com o processamento associado a ela, como a interrupção do processador para dizer que um quadro chegou, ou a divisão e recombinação de mensagens que eram muito grandes para caber em um quadro Ethernet.

4.3.7 Ethernet de 10 gigabits

Assim que a gigabit Ethernet foi padronizada, o comitê 802 ficou entediado e quis voltar ao trabalho. O IEEE pediu que se iniciassem os estudos sobre a Ethernet de 10 gigabits. Esse trabalho seguiu aproximadamente os mesmos moldes dos padrões Ethernet anteriores, com os padrões para fibra e cabo de cobre blindado, que apareceram primeiro em 2002 e 2004, seguidos pelo padrão para par trançado de cobre em 2006.

A velocidade de 10 Gbps é verdadeiramente prodigiosa, mil vezes mais rápida que a Ethernet original. Onde ela poderia ser necessária? A resposta é: dentro dos centros de dados e estações, para conectar roteadores de ponta, switches e servidores, bem como em troncos de longa distância e alta largura de banda, entre estações que habilitam redes metropolitanas inteiras com base em Ethernet e fibra. As conexões de longa distância utilizam fibra óptica, enquanto as conexões curtas podem usar cobre ou fibra.

Todas as versões da Ethernet de 10 gigabits admitem apenas operação full-duplex. CSMA/CD não faz mais parte do projeto, e os padrões se concentram em detalhes das camadas físicas que podem trabalhar em velocidades muito altas. Contudo, a compatibilidade ainda é importante, de modo que as interfaces Ethernet de 10 gigabits autonegociam e recuam para a velocidade mais baixa admitida pelas duas extremidades da linha.

Os principais tipos de Ethernet de 10 gigabits são listados na Figura 4.22. A fibra multimodo com comprimento de onda de 0,85μ (curta) e 1,5μ (estendida) é usada para longas distâncias. A 10GBase-ER pode percorrer distâncias de 40 km, o que a torna adequada para aplicações remotas. Todas essas versões enviam um fluxo serial de informações, produzido pelo embaralhamento dos bits de dados, depois a codificação com um código **64B/66B**. Essa codificação tem menos overhead do que uma codificação 8B/10B.

A primeira versão de cobre definida, 10GBase-CX4, usa um cabo com quatro pares de fiação de cobre twinaxial. Cada par usa codificação 8B/10B e trabalha a 3,125 Gsímbolos/s para alcançar 10 Gbps. Essa versão é mais barata do que a fibra e chegou cedo ao mercado, mas ainda não sabemos se em longo prazo vencerá a Ethernet de 10 gigabits sobre a fiação de par trançado mais comum.

A 10GBase-T é uma versão que usa cabos UTP. Embora exija fiação de Categoria 6a, para pequenas distâncias, ela pode usar categorias inferiores (incluindo a Categoria 5) para permitir algum reúso do cabeamento instalado. Não é surpresa que a camada física seja muito complicada para alcançar 10 Gbps sobre par trançado. Só veremos por alto alguns dos detalhes de alto nível. Cada um dos quatro pares

Nome	Cabo	Distância máxima do segmento	Vantagens
10GBase-SR	Fibra óptica	Até 300 m	Fibra multimodo (0,85 μ)
10GBase-LR	Fibra óptica	10 km	Fibra monomodo (1,3 μ)
10GBase-ER	Fibra óptica	40 km	Fibra monomodo (1,5 μ)
10GBase-CX4	4 pares de twinax	15 m	Cobre twinaxial
10GBase-T	4 pares de UTP	100 m	UTP padrão da Categoria 6a

Figura 4.22 O cabeamento da Ethernet de 10 gigabits.

trançados é usado para enviar 2.500 Mbps/s em ambas as direções. Essa velocidade é alcançada usando-se uma taxa de sinalização de 800 Msímbolos/s, com símbolos que usam 16 níveis de tensão. Os símbolos são produzidos embaralhando-se os dados, protegendo-os com o código LDPC (Low Density Parity Check) e codificando ainda mais para correção de erros.

A Ethernet de 10 gigabits ainda está sacudindo o mercado, mas o comitê 802.3 já fez progressos. Ao final de 2007, o IEEE criou um grupo para padronizar a Ethernet operando a 40 Gbps e a 100 Gbps. Essa atualização permitirá que a Ethernet concorra em ambientes de desempenho muito alto, incluindo conexões de longa distância em redes de backbone e conexões curtas, nas placas integradas de equipamentos. O padrão ainda não está concluído, mas já estão disponíveis produtos patenteados.

4.3.8 Ethernet de 40 e 100 gibabits

Depois de terminar de padronizar a Ethernet de 10 gigabits, o comitê 802.11 começou a trabalhar em novos padrões para Ethernet de 40 gigabits/s e 100 gigabits/s. O primeiro é voltado para conexões internas nos data centers, não em escritórios comuns e certamente não em usuários finais. Este último é direcionado ao backbone da Internet e, dessa forma, deve funcionar em trechos de rede óptica de milhares de quilômetros. Um uso possível é uma LAN virtual privada para conectar um data center com um milhão de CPUs a outro data center com um milhão de CPUs.

O primeiro padrão foi 802.3ba, aprovado em 2010, seguido pelo 802.3bj (2014) e 802.3cd (2018). Todos eles definem Ethernet em 40 Gbps e 100 Gbps. Os objetivos do projeto incluíram:

1. Compatibilidade com padrões 802.3 para 1 gigabit/s.
2. Permitir que os tamanhos de quadro mínimo e máximo permanecessem os mesmos.
3. Lidar com taxas de erro de bit de 10^{-12} ou menos.
4. Funcionar bem em redes ópticas.
5. Ter taxas de dados de 40 Gbps ou 100 Gbps.
6. Permitir o uso de fibra monomodo ou multimodo e placas integradas especializadas.

Os novos padrões eliminam o fio de cobre em favor da fibra óptica e placas integradas de alto desempenho (cobre) usadas em data centers que suportam a computação em nuvem. Podem ser usados meia dúzia de esquemas de modulação, incluindo 64B/66B (como 8B/10B, porém com mais bits). Além disso, até 10 pistas paralelas a 10 Gbps cada uma podem ser usadas para chegar a 100 Gbps. As pistas são normalmente bandas de frequência diferentes em uma fibra óptica. A integração em redes ópticas existentes usa a recomendação G.709 da ITU.

Por volta de 2018, um pequeno número de empresas começou a introduzir switches e placas adaptadoras de rede de 100 Gbps. Para quem 100 Gbps não é suficiente, já foi iniciado o trabalho nos padrões para até 400 gigabits/s, também chamados de 400GbE. Os padrões são 802.3cd, 802.3ck, 802.3cm e 802.3cn, se você quiser pesquisá-los. A 400 Gbps, um filme 4K típico (compactado) pode ser baixado por completo em cerca de 2 segundos.

4.3.9 Retrospectiva da Ethernet

A Ethernet existe há mais de 40 anos e não tem concorrentes sérios; portanto, é provável que continue no mercado ainda por muitos anos. Poucas arquiteturas de CPUs, sistemas operacionais ou linguagens de programação seriam capazes de se manter na liderança por quatro décadas, continuando com força. Sem dúvida, a Ethernet fez algo correto. O que foi?

Provavelmente a principal razão para sua longevidade seja o fato de que a Ethernet é simples e flexível. Na prática, simples se traduz como confiável, de baixo custo e de fácil manutenção. Depois que a arquitetura de hub e switch foi adotada, as falhas se tornaram extremamente raras. As pessoas hesitam em substituir algo que funciona bem o tempo todo, em especial quando sabem que uma quantidade terrível de itens da indústria de informática funciona muito mal. Muitas das chamadas "atualizações" são bem piores que as versões substituídas por elas.

Simplicidade também se traduz em economia. A fiação de par trançado tem custo relativamente baixo, assim como os componentes do hardware. Eles começam caros

quando há uma transição, por exemplo, novas NICs ou switches da gigabit Ethernet, mas são apenas acréscimos a uma rede bem estabelecida (não uma substituição dela) e os preços caem rapidamente à medida que o volume de vendas aumenta.

A Ethernet é de fácil manutenção. Não existe nenhum software para instalar (além dos drivers) e não há nenhuma tabela de configuração para gerenciar (e errar). Além disso, a inclusão de novos hosts é simples: basta conectá-los.

Outro ponto importante é que a Ethernet é capaz de interoperar facilmente com o TCP/IP, que se tornou dominante. O IP é um protocolo não orientado a conexões e, portanto, se ajusta perfeitamente à Ethernet, que também é não orientada a conexões. O IP não tem a mesma facilidade para se ajustar a alternativas orientadas a conexão, como o ATM. Essa falta de compatibilidade definitivamente prejudicou as chances de sucesso do ATM.

Por fim, e talvez mais importante, a Ethernet foi capaz de evoluir em certos aspectos cruciais. As velocidades aumentaram várias ordens de grandeza, e os hubs e switches foram introduzidos, mas essas mudanças não exigiram alterações no software e normalmente permitiam que o cabeamento existente fosse reutilizado por um tempo. Quando um vendedor de redes aparece em uma grande instalação e diz: "Tenho esta nova e fantástica rede para você. Basta se desfazer de todo o seu hardware e reescrever todo o seu software", ele tem um problema.

Muitas tecnologias alternativas, que você provavelmente nem sequer ouviu falar, eram mais rápidas que a Ethernet quando foram introduzidas. Assim como o ATM, essa lista inclui o FDDI (Fiber Distributed Data Interface) e o Fibre Channel,* duas LANs ópticas baseadas em anéis. Ambas eram incompatíveis com a Ethernet. Nenhuma delas teve sucesso. Elas eram muito complicadas, o que resultava em chips complexos e altos preços. A lição que deveria ter sido aprendida aqui é que as coisas precisam ser mantidas simples. Por fim, a Ethernet os alcançou em termos de velocidade, geralmente pegando parte de sua tecnologia emprestada, por exemplo, a codificação 4B/5B do FDDI e a codificação 8B/10B do Fibre Channel. Então, ou elas não tinham mais vantagens e desapareceram silenciosamente ou tiveram funções especializadas.

Parece que a Ethernet continuará a se expandir em suas aplicações por algum tempo. A Ethernet de 10 gigabits acabou com as restrições de distância do CSMA/CD. Tem sido realizado muito esforço para a **carrier-grade Ethernet**, ou simplesmente **Carrier-Ethernet**, para permitir que os provedores de rede ofereçam serviços baseados em Ethernet aos seus clientes para redes metropolitanas e a longas distâncias (Hawkins, 2016). Essa aplicação transporta quadros Ethernet a longas distâncias através da fibra e exige melhores recursos de gerenciamento para ajudar as operadoras a oferecer serviços confiáveis e de alta qualidade. As redes com velocidade muito alta, como 100GbE, também estão sendo usadas em placas integradas, conectando componentes em grandes roteadores ou servidores. Esses dois usos são adicionais ao envio de quadros entre computadores em escritórios. O próximo passo é a 400GbE, e esse nem sequer poderá ser o último.

*Ele se chama "Fibre Channel" e não "Fiber Channel", pois o editor do documento era britânico.

4.4 LANS SEM FIO

As LANs sem fio estão cada vez mais populares e um número crescente de casas, escritórios, lanchonetes, bibliotecas, aeroportos, zoológicos e outros lugares públicos está sendo equipado com elas, para conectar computadores, notebooks, tablets e smartphones à Internet. As LANs sem fio também podem ser usadas para permitir que dois ou mais computadores vizinhos se comuniquem sem usar a Internet.

O principal padrão de LAN sem fio é o 802.11. Vimos algumas informações básicas sobre ele na Seção 1.5.3. Agora, vamos examinar mais de perto a tecnologia. Nas próximas seções, estudaremos a pilha de protocolos, as técnicas de transmissão de rádio na camada física, o protocolo da subcamada MAC, a estrutura do quadro e os serviços fornecidos. Para obter mais informações sobre o 802.11, consulte Bing (2017) e Davis (2018). Para conhecer os detalhes mais profundos, consulte os próprios padrões publicados pelo IEEE.

4.4.1 802.11: arquitetura e pilha de protocolos

As redes 802.11 podem ser usadas em dois modos. O modo mais popular é conectar clientes, como laptops e smartphones, a outra rede, como uma intranet da empresa ou a Internet. Esse modo aparece na Figura 4.23(a). No modo de infraestrutura, cada cliente está associado a um **PA** (**ponto de acesso**), que, por sua vez, está conectado a outra rede. O cliente transmite e recebe seus pacotes por meio do PA. Vários PAs podem ser conectados, normalmente por uma rede com fio chamada **sistema de distribuição**, para formar uma rede 802.11 estendida. Nesse caso, os clientes podem enviar quadros aos outros clientes por meio de seus PAs.

O outro modo, mostrado na Figura 4.23(b), é uma **rede ad hoc**. Trata-se de uma coleção de computadores que estão associados de modo que possam enviar quadros diretamente uns aos outros. Não existe PA. Como o acesso à Internet é a principal aplicação para redes sem fio, as redes ad hoc não são muito populares.

Agora, vejamos os protocolos. Todos os protocolos 802, incluindo 802.11 e Ethernet, têm certas características

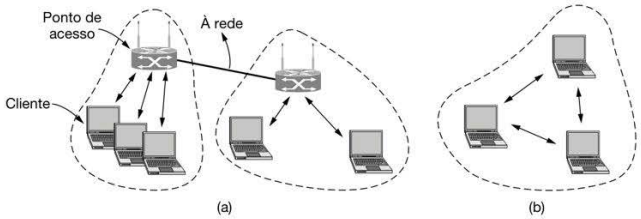


Figura 4.23 Arquitetura 802.11. (a) Modo de infraestrutura. (b) Modo ad hoc.

comuns em sua estrutura. Uma visão parcial da pilha de protocolos do 802.11 para suas principais variantes é dada na Figura 4.24. A pilha é a mesma para clientes e PAs. A camada física corresponde muito bem à camada física do modelo OSI, mas a camada de enlace de dados em todos os protocolos 802 se divide em duas ou mais subcamadas. No 802.11, a subcamada MAC determina como o canal é alocado, isto é, quem terá a oportunidade de transmitir a seguir. Acima dela encontra-se a subcamada LLC, cujo trabalho é ocultar as diferenças entre as diversas variações do 802 e torná-las indistinguíveis no que se refere à camada de rede. Essa poderia ter sido uma responsividade significativa, mas atualmente a LLC é uma camada de cola, que identifica o protocolo (p. ex., IP) que é transportado dentro de um quadro 802.11.

Várias técnicas de transmissão foram acrescentadas à camada física à medida que o 802.11 evoluiu desde o seu aparecimento, em 1997. Duas das técnicas iniciais, infravermelho como nos controles remotos de televisão e salto de frequência na banda de 2,4 GHz, agora não são mais usadas. A terceira técnica inicial, o espectro de dispersão de sequência direta a 1 ou 2 Mbps na banda de 2,4 GHz, foi estendida para trabalhar em velocidades de até 11 Mbps e tornou-se rapidamente um sucesso. Ela agora é conhecida como 802.11b.

Para dar aos viciados em redes sem fio o aumento de velocidade tão desejado, novas técnicas de transmissão, baseadas no esquema OFDM (Orthogonal Frequency Division Multiplexing), que descrevemos na Seção 2.5.3, foram introduzidas em 1999 e em 2003. A primeira é chamada 802.11a e usa uma banda de frequência diferente, a de 5 GHz. A segunda ficou com 2,4 GHz e compatibilidade. Ela é chamada 802.11g. Ambas oferecem velocidades de até 54 Mbps.

Técnicas de transmissão que usam simultaneamente várias antenas no transmissor e no receptor para aumentar a velocidade foram finalizadas como 802.11n, em outubro de 2009.

Em dezembro de 2013, o IEEE já tinha usado todas as letras do alfabeto, e publicou o próximo padrão como 802.11ac. Como um parêntese, os membros do comitê 802.11 conhecem o alfabeto inteiro, e usam as letras "que faltam", como 802.11r, para pequenas melhorias e acréscimos técnicos (geralmente, para esclarecimentos e reparos de falhas). O 802.11ac opera na banda de 5 GHz, o que significa que os dispositivos mais antigos, que usam apenas a banda de 2,4 GHz, não poderão usá-lo. Os dispositivos móveis mais modernos usam 802.11ac. Mais recentemente, o padrão 802.11ax foi aprovado para ainda mais velocidade.

Agora, vamos examinar cada uma dessas técnicas de transmissão em linhas gerais. Contudo, abordaremos

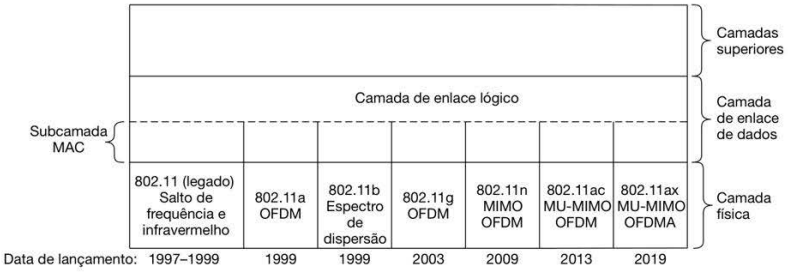


Figura 4.24 Parte da pilha de protocolos do 802.11.

apenas aquelas que estão em uso, pulando os métodos de transmissão 802.11 legados. Tecnicamente, elas pertencem à camada física e deveriam ter sido examinadas no Capítulo 2; porém, como estão estritamente relacionadas às LANs em geral e em particular à LAN 802.11, preferimos tratá-las aqui.

4.4.2 802.11: a camada física

Cada uma das técnicas de transmissão torna possível enviar um quadro MAC de uma estação para outra através do ar. Contudo, elas diferem na tecnologia usada e na velocidade que podem alcançar na prática. Uma descrição detalhada dessas tecnologias está muito além do escopo deste livro, mas algumas palavras sobre cada uma relacionarão as técnicas ao conteúdo abordado no Capítulo 2, fornecendo aos leitores interessados material para pesquisar mais informações em outras fontes.

Todas as técnicas do 802.11 utilizam rádios de curto alcance para transmitir sinais nas bandas de frequência ISM de 2,4 GHz ou 5 GHz. Essas bandas têm a vantagem de não ser licenciadas e, portanto, estar disponíveis gratuitamente a qualquer transmissor que queira cumprir algumas restrições, como a potência irradiada de no máximo 1 W (embora 50 mW seja mais comum para rádios de LAN sem fio). Infelizmente, esse fato também é conhecido pelos fabricantes de aparelhos de abertura automática de garagem, telefones sem fio, fornos de micro-ondas e diversos outros dispositivos, todos competindo com os notebooks e smartphones pelo mesmo espectro. A banda de 2,4 GHz costuma ser mais sobrecarregada do que a de 5 GHz, de modo que esta pode ser melhor para algumas aplicações, embora tenha um alcance mais curto, em virtude da frequência mais alta. Infelizmente, as ondas de rádio mais curtas a 5 GHz não penetram em paredes tão bem quanto as de 2,4 GHz, de modo que 5 GHz não é uma vantagem definitiva.

Todos os métodos de transmissão também definem taxas múltiplas. A ideia é que diferentes taxas podem ser usadas dependendo das condições atuais. Se o sinal sem fio for fraco, uma taxa baixa poderá ser usada. Se o sinal for claro, a taxa mais alta poderá ser usada. Esses ajustes constituem o que chamamos de **adaptação de taxa**. Como as taxas variam por um fator de 10 ou mais, uma boa adaptação de taxa é importante para um bom desempenho. É claro que, pelo fato de ela não ser necessária para a interoperabilidade, os padrões não dizem como a adaptação de taxa deve ser feita.

O primeiro método de transmissão que veremos é o **802.11b**, de espectro de dispersão que admite taxas de 1, 2, 5,5 e 11 Mbps, embora na prática a taxa de operação seja quase sempre 11 Mbps. Isso é semelhante ao sistema CDMA, que examinamos na Seção 2.4.4, exceto que há somente um código de espalhamento compartilhado por todos os usuários. O espalhamento é usado para satisfazer ao requisito da FCC de que a potência deve ser espalhada pela banda ISM. A sequência de espalhamento usada pelo 802.11b é uma **sequência de Barker**. Ela tem como propriedade uma baixa autocorrelação, exceto quando as sequências estão alinhadas, o que permite que um receptor intercepte o início de uma transmissão. Para transmitir em uma taxa de 1 Mbps, a sequência de Barker é usada com a modulação BPSK para enviar 1 bit por 11 chips. Os chips são transmitidos a uma taxa de 11 Mchips/s. Para enviar a 2 Mbps, ela é usada com a modulação QPSK para enviar 2 bits por 11 chips. As taxas mais altas são diferentes, pois usam uma técnica conhecida como **chaveamento de código complementar**, ou **CCK (Complementary Code Keying)**, para construir códigos em vez da sequência de Barker. A taxa de 5,5 Mbps envia 4 bits em cada código de 8 chips, e a taxa de 11 Mbps envia 8 bits em cada código de 8 chips.

Em seguida, chegamos ao **802.11a**, que admite taxas de até 54 Mbps na banda ISM de 5 GHz. Você poderia esperar que o 802.11a viesse antes do 802.11b, mas não foi assim. Embora o grupo 802.11a tenha sido estabelecido primeiro, o padrão 802.11b foi aprovado primeiro e seu produto chegou ao mercado bem antes dos produtos 802.11a, parcialmente em virtude da dificuldade de operar na banda mais alta de 5 GHz.

O método 802.11a é baseado na **multiplexação por divisão ortogonal de frequência**, ou **OFDM (Orthogonal Frequency Division Multiplexing)**, pois a OFDM usa o espectro com eficiência e resiste a degradações do sinal sem fio, como o enfraquecimento por múltiplos caminhos. Os bits são enviados por 52 subportadoras em paralelo, 48 transportando dados e 4 usadas para sincronização. Cada símbolo dura 4 μs e envia 1, 2, 4 ou 6 bits. Os bits são codificados para correção de erros, primeiro com um código de convolução binário, de modo que somente 1/2, 2/3 ou 3/4 dos bits não são redundantes. Com diferentes combinações, o 802.11a pode trabalhar em oito taxas, variando de 6 a 54 Mbps. Essas taxas são significativamente mais rápidas do que as taxas 802.11b, e existe menos interferência na banda de 5 GHz. Contudo, o 802.11b tem um alcance que é cerca de sete vezes maior que o do 802.11a, o que em muitas situações é mais importante.

Mesmo com o alcance maior, o pessoal do 802.11b não tinha intenção de permitir que esse início vencesse o campeonato de velocidade. Felizmente, em maio de 2002, a FCC retirou sua regra, existente havia muito tempo, de exigir que todo equipamento de comunicação sem fio operasse nas bandas ISM nos Estados Unidos para usar o espectro de dispersão, de modo que passou a trabalhar no **802.11g**, que foi aprovado pelo IEEE em 2003. Ele copia os métodos de modulação OFDM do 802.11a, mas opera na banda ISM estreita de 2,4 GHz, com o 802.11b. Ele oferece as mesmas taxas do 802.11a (6 a 54 Mbps) mais, é claro, a compatibilidade com quaisquer dispositivos 802.11b que estejam

nas proximidades. Todas essas diferentes escolhas podem ser confusas para os clientes, de modo que é comum que os produtos ofereçam suporte para 802.11a/b/g em uma única placa de interface de rede.

Não satisfeito em parar aí, o comitê do IEEE começou a trabalhar em uma camada física de alto throughput, chamada **802.11n**. Ela foi ratificada em 2009. Seu objetivo foi um throughput de pelo menos 100 Mbps depois que todos os overheads da rede sem fio fossem removidos. Isso exigia um aumento de velocidade bruto, com um fator de pelo menos quatro. Para isso acontecer, o comitê dobrou os canais de 20 MHz para 40 MHz e reduziu os overheads de enquadramento, permitindo que um grupo de quadros fosse enviado em conjunto. Todavia, o mais significativo é que o 802.11n usa até quatro antenas para transmitir até quatro fluxos de informação ao mesmo tempo. Os sinais dos fluxos interferem no receptor, mas eles podem ser separados usando as técnicas de comunicação de **entrada múltipla, saída múltipla**, ou **MIMO (Multiple Input, Multiple Output)**. O uso de múltiplas antenas oferece um grande aumento de velocidade e, além disso, melhora o alcance e a confiabilidade. MIMO, assim como OFDM, é uma daquelas ideias de comunicação inteligentes que estão mudando os projetos das redes sem fio e das quais, provavelmente, todos nós ouviremos falar muito no futuro. Para obter uma breve introdução às antenas múltiplas no 802.11, consulte Halpein et al. (2010).

Em 2013, o IEEE publicou o padrão 802.11ac. Ele usa canais mais largos (80 MHz e 160 MHz), modulação 256-QAM e **MU-MIMO (MultiUser MIMO)**, com até oito fluxos e outros truques para aumentar a taxa de bits até um máximo teórico de 7 Gbps, embora, na prática, isso nunca tenha sido alcançado. Os dispositivos móveis mais modernos geralmente usam 802.11ac.

Outro padrão 802.11 recente é o **802.11ad**, que opera na banda de 60 GHz (57 a 71 GHz), o que significa que as ondas de rádio são muito curtas: somente 5 mm de extensão. Essas ondas não penetram em paredes ou outros objetos, de modo que o padrão só é útil dentro de uma única sala. Entretanto, essa é uma vantagem e também uma desvantagem. Significa que, não importando o que a pessoa no outro escritório ou apartamento esteja fazendo, não interferirá com o que você está fazendo. A combinação de alta largura de banda e penetração fraca o torna ideal para o streaming de filmes 4K ou 8K não compactados, de uma estação base em uma sala para dispositivos móveis na sala. Uma melhoria neste padrão, aumentando a largura de banda por um fator de quatro, é o padrão **802 11ay**.

Agora chegamos ao **802.11ax**, às vezes conhecido como o padrão de rede **sem fio de alta eficiência**. O nome mais conhecido do consumidor é **WiFi 6** (se você pensou que ficaria com WiFi de 1 a 5, se enganou; os nomes antigos eram baseados nos números dos padrões IEEE e a WiFi Alliance decidiu chamar essa revisão de WiFi 6 porque é a sexta versão do padrão WiFi). Ele permite uma codificação QAM mais eficiente, juntamente com um novo esquema de modulação, OFDMA. Ele pode (a princípio) operar em partes não licenciadas do espectro de até 7 GHz e pode (teoricamente) atingir uma taxa de dados de 11 Gbps. Você pode tentar isso em casa, se quiser, mas a menos que tenha um laboratório de teste perfeitamente projetado, você não obterá 11 Gbps. No entanto, você pode alcançar 1 Gbps.

No 802.11ax OFDMA, um alocador central reserva unidades de recursos de comprimento fixo para cada uma das estações de transmissão, reduzindo assim a disputa em implantações densas. O 802.11ax também oferece suporte para a reutilização do espectro espacial, por meio de uma técnica chamada **coloração**, pela qual um remetente marca o início de sua transmissão de modo a permitir que outros remetentes determinem se pode haver uso simultâneo do espectro. Em algumas circunstâncias, um remetente pode transmitir simultaneamente se reduzir sua potência de modo apropriado.

Além disso, o 802.11ax usa 1024-QAM, que permite que cada símbolo codifique 10 bits, em oposição aos 8 bits/símbolo no 256-QAM, usado pelo 802.11ac. O padrão também oferece suporte a uma programação mais inteligente, por meio de um recurso chamado **tempo de despertar-alvo ou horário de ativação desejado**, que permite que um roteador coloque os dispositivos da casa em programações de transmissão, para minimizar colisões. Esse recurso é provavelmente mais útil em casas inteligentes, onde cada vez mais dispositivos conectados podem precisar enviar "sinais de vida" periodicamente para o roteador doméstico.

4.4.3 802.11: o protocolo da subcamada MAC

Agora, vamos retornar dos domínios da engenharia elétrica para os da ciência da computação. O protocolo da subcamada MAC do 802.11 é bastante diferente do protocolo da Ethernet, em razão da complexidade inerente à comunicação sem fio.

Primeiro, os rádios quase sempre são half-duplex, significando que eles não podem transmitir e escutar rajadas de sinais ruidosos ao mesmo tempo em uma única frequência. O sinal recebido pode facilmente ser um milhão de vezes mais fraco do que o sinal transmitido, de modo que não pode ser detectado ao mesmo tempo. Com a Ethernet, uma estação só precisa esperar até o éter ficar inativo para começar a transmitir. Se não receber de volta uma rajada de sinal ruidoso enquanto transmite os primeiros 64 bytes, é quase certo que o quadro tenha sido entregue corretamente. No caso das LANs sem fio, esse mecanismo de detecção de colisão não funciona.

Em vez disso, o 802.11 tenta evitar colisões com um protocolo chamado **CSMA com prevenção de colisão**, ou **CSMA/CA (CSMA with Collision Avoidance)**.

Ele é conceitualmente semelhante ao CSMA/CD da Ethernet, com detecção de portadora antes de transmitir e o algoritmo de recuo (backoff) exponencial binário após as colisões. Contudo, uma estação que tem um quadro para transmitir começa com um recuo aleatório (exceto no caso em que ela não tenha usado o canal recentemente e o canal esteja inoperante). Ela não espera por uma colisão. O número de slots a recuar é escolhido na faixa de 0 a, digamos, 15, no caso da camada física OFDM. A estação espera até que o canal esteja inoperante, detectando que não existe sinal por um curto período (chamado DIFS, como explicaremos mais adiante), e conta regressivamente os slots inoperantes, interrompendo quando os quadros forem enviados. Ela envia seu quadro quando o contador chega a 0. Se o quadro passar, o destino imediatamente envia uma confirmação curta. A falta de uma confirmação é deduzida como indicativo de erro, seja uma colisão, seja outro erro qualquer. Nesse caso, o transmissor dobra o período de recuo e tenta novamente, continuando com o recuo exponencial, como na Ethernet, até que o quadro tenha sido transmitido com sucesso ou o número máximo de retransmissões tenha sido alcançado.

Uma linha do tempo como exemplo aparece na Figura 4.25. A estação *A* é a primeira a transmitir um quadro. Enquanto *A* está transmitindo, as estações *B* e *C* ficam prontas para enviar. Elas veem que o canal está ocupado e esperam até que ele esteja livre. Pouco depois de *A* receber uma confirmação, o canal é liberado. Contudo, em vez de enviar um quadro imediatamente e colidir, *B* e *C* realizam um recuo. *C* escolhe um recuo pequeno e, assim, transmite primeiro. *B* interrompe sua contagem enquanto detecta que *C* está usando o canal e retoma depois que *C* tiver recebido uma confirmação. *B* logo conclui seu recuo e transmite seu quadro.

Em comparação com a Ethernet, existem duas diferenças principais. Primeiro, iniciar os recuos cedo ajuda a evitar colisões. Essa prevenção vale a pena porque as colisões são dispendiosas, já que o quadro inteiro é transmitido mesmo que ocorra uma colisão. Em segundo lugar, as confirmações são usadas para deduzir colisões, pois estas não podem ser detectadas.

Esse modo de operação é chamado de **função de coordenação distribuída**, ou **DCF** (**Distributed Coordination Function**), pois cada estação atua de modo independente, sem qualquer tipo de controle central. O padrão também inclui um modo de operação opcional, chamado de **função de coordenação de ponto**, ou **PCF** (**Point Coordination Function**), em que o PA controla toda a atividade em sua célula, assim como uma estação-base de celular. Todavia, o PCF não é usado na prática porque normalmente não existe um modo de impedir que as estações em outra rede vizinha transmitam um tráfego simultâneo.

O segundo problema é que o alcance de transmissão de estações distintas pode não ser o mesmo. Com um fio, o sistema é preparado de modo que todas as estações possam escutar umas às outras. Com a complexidade da propagação por ondas de rádio, essa situação não acontece para as estações sem fio. Consequentemente, podem surgir situações como o problema do terminal oculto, mencionado anteriormente e ilustrado novamente na Figura 4.26(a). Como nem todas as estações estão dentro do mesmo alcance de rádio, as transmissões que acontecem em uma parte de uma célula podem não ser recebidas em outra parte da mesma célula. Nesse exemplo, a estação *C* está transmitindo para a estação *B*. Se *A* detectar o canal, ela não escutará nada e concluirá incorretamente que agora pode começar a transmitir para *B*. Essa decisão leva a uma colisão.

A situação inversa é o problema do terminal exposto, ilustrado na Figura 4.26(b). Aqui, *B* deseja enviar para *C* e, portanto, escuta o canal. Quando ele detecta uma transmissão, conclui incorretamente que não pode transmitir para *C*, embora *A* de fato possa estar transmitindo para *D* (não mostrado). Essa decisão desperdiça uma oportunidade de transmissão.

Para reduzir ambiguidades sobre qual estação está transmitindo, o 802.11 define a detecção do canal de modo físico e virtual. A detecção física simplesmente verifica o

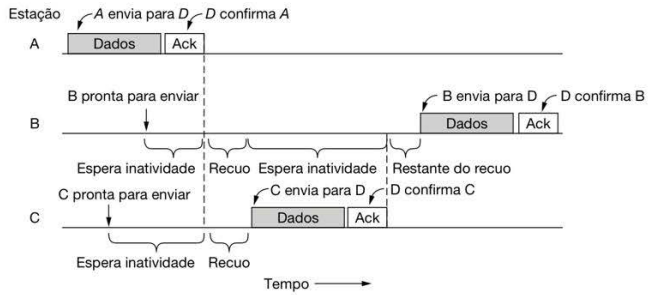


Figura 4.25 Transmitindo um quadro com CSMA/CA.

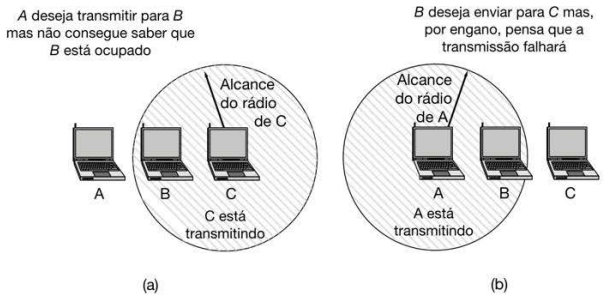


Figura 4.26 (a) O problema do terminal oculto. (b) O problema do terminal exposto.

meio para ver se existe um sinal válido. Com a detecção virtual, cada estação mantém um registro lógico de quando o canal está em uso rastreando o **vetor de alocação de rede**, ou **NAV** (**Network Allocation Vector**). Todo quadro transporta um campo NAV que diz quanto tempo levará para concluir a sequência da qual esse quadro faz parte. As estações que escutam esse quadro sabem que o canal estará ocupado pelo período indicado em NAV, independentemente se elas podem detectar um sinal físico. Por exemplo, o NAV de um quadro de dados inclui o tempo necessário para enviar uma confirmação. Todas as estações que escutarem o quadro de dados serão adiadas durante o período de confirmação, mesmo que não a escutem. Basicamente, o NAV funciona como um temporizador de contagem regressiva, em cujo período o transmissor assume que o canal está ocupado. No padrão 802.11, as unidades do NAV são microssegundos. Em locais mais densos, o NAV definido por um transmissor pode ser redefinido por outros transmissores no mesmo intervalo de transmissão, causando colisões e desempenho abaixo do ideal. Para mitigar esse efeito, o 802.11ax introduz dois NAVs: um modificado por quadros correspondentes aos quadros aos quais a estação está associada, e o outro modificado por quadros que são ouvidos pela estação, mas se formaram em redes sobrepostas.

Um mecanismo RTS/CTS opcional usa o NAV para impedir que os terminais transmitam quadros ao mesmo tempo que os terminais ocultos. Isso aparece na Figura 4.27. Nesse exemplo, *A* deseja enviar para *B*. *C* é uma estação dentro do alcance de *A* (e possivelmente dentro do alcance de *B*, mas isso não importa). *D* é uma estação dentro do alcance de *B*, mas não dentro do alcance de *A*.

O protocolo começa quando *A* decide enviar dados para *B*. *A* começa a transmitir um quadro RTS para *B*, pedindo permissão para lhe enviar (Request To Send) um quadro. Se *B* recebe esse pedido, responde com um quadro CTS, indicando que o canal está liberado para enviar (Clear To Send). Ao receber o CTS, *A* envia seu quadro e inicia um timer de ACK (confirmação). Ao recebimento correto do quadro de dados, a estação *B* responde com um quadro ACK, completando a troca. Se o timer de ACK de *A* expirar antes que o ACK retorne a ela, isso é tratado como uma colisão e o protocolo inteiro é realizado novamente, após um recuo.

Agora, vamos considerar essa troca do ponto de vista de *C* e *D*. Como *C* está dentro do alcance de *A*, ela pode receber o quadro RTS. Se receber, essa estação percebe que alguém transmitirá dados em breve. Pela informação fornecida no pedido de RTS, ela pode estimar o tempo que a sequência levará, incluindo o ACK final. Assim, para o

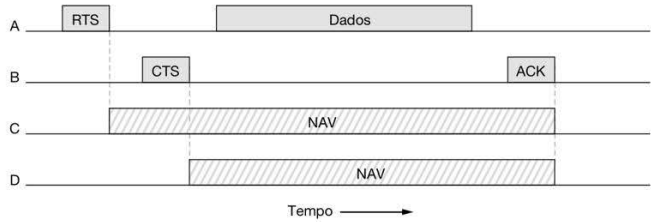


Figura 4.27 O uso da detecção de canal virtual com o CSMA/CA.

bem de todos, a estação C desiste de transmitir algo até que a troca seja concluída. Ela faz isso atualizando seu registro do NAV para indicar que o canal está ocupado, como mostra a Figura 4.27. *D* não escuta o RTS, mas escuta o CTS, de modo que também atualiza seu NAV. Observe que os sinais NAV não são transmitidos; eles são apenas lembretes internos para ficar em silêncio por determinado período.

Entretanto, embora RTS/CTS pareça ser bom na teoria, esse é um daqueles projetos que provaram ter pouco valor na prática. Existem vários motivos pelos quais ele raramente é usado. Ele não ajuda para quadros curtos (que são enviados no lugar do RTS) ou para o PA (que, por definição, todos podem ouvir). Para outras situações, ele só atrasa a operação. RTS/CTS no 802.11 é um pouco diferente daquele do protocolo MACA, que vimos na Seção 4.2, pois qualquer um que escuta o RTS ou o CTS permanece em silêncio por todo o período, para permitir que o ACK seja enviado sem colisão. Por causa disso, a técnica não ajuda com terminais expostos como o MACA fazia, somente com terminais ocultos. Com frequência, existem poucos terminais ocultos, e o CSMA/CA já os ajuda atrasando as estações que não têm êxito na transmissão, qualquer que seja a causa, para aumentar as chances de êxito das transmissões.

O CSMA/CA com detecções física e virtual é o núcleo do protocolo 802.11. Contudo, existem outros mecanismos que foram desenvolvidos para acompanhá-lo. Como cada um desses mecanismos foi controlado pelas necessidades da operação real, vamos examiná-los rapidamente.

A primeira necessidade que examinaremos é a confiabilidade. Ao contrário das redes fisicamente conectadas, as redes sem fio são ruidosas e pouco confiáveis, em grande parte em virtude da interferência com outros dispositivos, como fornos de micro-ondas, que também utilizam as bandas ISM não licenciadas. O uso de confirmações e retransmissões não ajuda muito se a probabilidade de transferir um quadro for pequena em primeiro lugar.

A estratégia principal usada para aumentar as transmissões com êxito é reduzir a taxa de transmissão. Taxas mais baixas usam modulações mais robustas, que mais provavelmente serão recebidas de modo correto para determinada relação sinal-ruído. Se muitos quadros se perderem, uma estação poderá reduzir a taxa. Se os quadros forem entregues com pouca perda, uma estação ocasionalmente poderá testar uma taxa mais alta para ver se ela deve ser usada.

Outra estratégia para melhorar as chances de o quadro atravessar a rede sem prejuízo é enviar quadros mais curtos. Se a probabilidade de ocorrer um erro em qualquer bit é p, então a probabilidade de um quadro de n bits ser recebido de forma inteiramente correta é $(1 - p)^n$. Por exemplo, para $p = 10^{-4}$, a probabilidade de receber um quadro Ethernet completo (12.144 bits) sem erros é menor que 30%. A maioria dos quadros será perdida. Mas, se os quadros tiverem apenas um terço desse tamanho (4.048 bits), dois terços deles serão recebidos corretamente. Agora, a maioria dos quadros passará e menos retransmissões serão necessárias.

Quadros mais curtos podem ser implementados reduzindo-se o tamanho máximo da mensagem que é aceita a partir da camada de rede. Como alternativa, o 802.11 permite que os quadros sejam divididos em partes menores, chamadas **fragmentos**, cada uma com seu próprio checksum. O tamanho do fragmento não é fixado pelo padrão, mas é um parâmetro que pode ser ajustado pelo PA. Os fragmentos são numerados individualmente e confirmados com o uso de um protocolo do tipo stop-and-wait (i.e., o transmissor não pode enviar o fragmento $k + 1$ enquanto não receber a confirmação do fragmento k). Depois que um canal é "apoderado", vários fragmentos podem ser enviados em rajada. Eles seguem um após o outro com uma confirmação (e, possivelmente, retransmissões) no intervalo, até que o quadro inteiro tenha sido transmitido com sucesso ou o tempo de transmissão atinja o máximo permitido. O mecanismo NAV mantém as outras estações inativas apenas até a próxima confirmação, mas outro mecanismo (veja a seguir) é usado para permitir que uma rajada de fragmentos seja enviada sem que outras estações enviem um quadro no meio.

A segunda necessidade que discutiremos é economizar energia. A duração da bateria é sempre um problema nos dispositivos móveis sem fio. O padrão 802.11 dedica atenção à questão do gerenciamento de energia, para que os clientes não a desperdicem quando não têm informações para enviar ou para receber.

O mecanismo básico para economizar energia é calcado em **quadros de baliza (beacon frames)**. As balizas são transmissões periódicas do PA (p. ex., a cada 100 ms). Os quadros de baliza anunciam a presença do PA aos clientes e transportam parâmetros do sistema, como o identificador do PA, a hora, o tempo até a próxima baliza e configurações de segurança.

Os clientes podem definir um bit de gerenciamento de energia nos quadros que eles enviam ao PA, para informar que estão entrando no **modo de economia de energia**. Nele, o cliente pode cochilar e o PA manterá em buffer o tráfego (barrado) voltado para ele. Para verificar o tráfego que chegou, o cliente acorda a cada baliza e verifica um mapa de tráfego enviado como parte do quadro de baliza. Esse mapa diz ao cliente se existe tráfego à espera no buffer. Se houver, o cliente envia uma pool message (consulta) ao PA, que em seguida envia o tráfego armazenado. O cliente pode, então, voltar a dormir até que a próxima baliza seja enviada.

Outro mecanismo de economia de energia, chamado **APSD (Automatic Power Save Delivery)**, também foi acrescentado ao 802.11 em 2005. Com esse novo mecanismo, o PA mantém quadros em buffer e os envia para um cliente logo depois de ele enviar quadros ao PA. O cliente pode, então, dormir até que tenha mais tráfego para enviar (e receber). Esse mecanismo funciona bem para aplicações como VoIP, que têm tráfego frequente nos dois sentidos. Por exemplo, um telefone sem fio VoIP poderia usá-lo para enviar e receber quadros a cada 20 ms, com muito mais

frequência do que o intervalo de baliza de 100 ms, enquanto chochila nos intervalos.

A terceira e última necessidade que examinaremos é a qualidade de serviço. Quando o tráfego VoIP no exemplo anterior competir com o tráfego peer-to-peer, o VoIP sofrerá. Ele será adiado em virtude da disputa com o tráfego peer-to-peer de alta largura de banda, embora a largura de banda VoIP seja baixa. Esses atrasos provavelmente degradarão as chamadas de voz. Para impedir que isso ocorra, gostaríamos de permitir que o tráfego VoIP siga antes do tráfego peer-to-peer, pois tem maior prioridade.

O IEEE 802.11 tem um mecanismo inteligente para fornecer esse tipo de qualidade de serviço que foi apresentado como um conjunto de extensões sob o nome 802.11e em 2005. Ele funciona estendendo o CSMA/CA com intervalos cuidadosamente definidos entre os quadros. Depois que um quadro é enviado, é necessária certa quantidade de tempo de inatividade antes que qualquer estação possa enviar um quadro para verificar se o canal não está mais sendo usado. O truque é definir diferentes intervalos para diferentes tipos de quadros.

Cinco intervalos são representados na Figura 4.28. O intervalo entre quadros de dados regulares é chamado de **espaçamento entre quadros DCF**, ou **DIFS (DCF InterFrame Spacing)**. Qualquer estação pode tentar adquirir o canal para enviar um novo quadro até que o meio tenha ficado ocioso por DIFS. As regras habituais de disputa se aplicam e, se ocorrer uma colisão, o algoritmo de backoff (recuo exponencial binário) pode ser necessário. O menor intervalo é o **espaçamento curto entre quadros**, ou **SIFS (Short InterFrame Spacing)**, usado para permitir que as partes de um único diálogo tenham a chance de transmitir primeiro. Isso inclui permitir que o receptor envie um ACK, outras sequências de quadro de controle, como RTS e CTS, ou permitir que o transmissor envie uma rajada de fragmentos. O envio do próximo fragmento após esperar apenas o SIFS é o que impede que outra estação entre com um quadro no meio da troca.

Os dois intervalos **AIFS (Arbitration InterFrame Space)** mostram exemplos de dois níveis de prioridade. O intervalo curto, $AIFS_1$, é menor que o DIFS, porém maior que o SIFS. Ele pode ser usado pelo PA para mover o tráfego de voz e outro tráfego de alta prioridade para o início da fila. O PA esperará por um intervalo mais curto antes de enviar o tráfego de voz e, assim, o fará antes do tráfego normal. O intervalo longo, $AIFS_4$, é maior que o DIFS. Ele é usado para o tráfego de segundo plano, que pode ser adiado para depois do tráfego regular. O PA esperará por um intervalo maior antes de enviá-lo, dando ao tráfego regular a oportunidade para transmitir primeiro. O mecanismo completo de qualidade de serviço define quatro níveis de prioridade, que têm diferentes parâmetros de recuo, bem como diferentes parâmetros ociosos.

O último intervalo, o **espaçamento estendido entre quadros**, ou **EIFS (Extended InterFrame Spacing)**, só é usado por uma estação que tenha acabado de receber um quadro defeituoso ou desconhecido, a fim de informar sobre o problema. A ideia é que, como o receptor talvez não tenha nenhum conhecimento do que está acontecendo, ele deve esperar um tempo significativo para evitar interferir em um diálogo em andamento entre duas estações.

Outra parte das extensões de qualidade de serviço é a noção de uma **TXOP**, ou **oportunidade de transmissão**. O mecanismo de CSMA/CA original permite que as estações enviem um quadro de cada vez. Esse projeto foi bom até o aumento das taxas de transferência. Com o 802.11a/g, uma estação poderia enviar a 6 Mbps e outra estação enviar a 54 Mbps. Cada uma delas passa a enviar um quadro, mas a estação de 6 Mbps leva nove vezes mais tempo (ignorando os overheads fixos) que a estação de 54 Mbps para enviar seu quadro. Essa disparidade tem o efeito colateral indesejado de atrasar um transmissor rápido que esteja competindo com um transmissor lento para aproximadamente a taxa do transmissor lento. Por exemplo, novamente ignorando overheads fixos, enviando sozinhos, os transmissores de 6 e 54 Mbps receberão em suas próprias taxas, mas, ao enviar juntos, ambos receberão 5,4 Mbps na média. Essa é uma penalidade cruel para o transmissor rápido. Esse problema é conhecido como **anomalia de taxa** (Heusse et al., 2003).

Com oportunidades de transmissão, cada estação recebe uma fatia igual de tempo no ar, não um número igual de

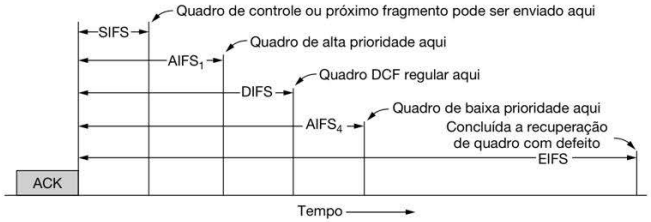


Figura 4.28 Espaçamento entre quadros no 802.11.

quadros. As estações que enviam a uma taxa mais alta que seu tempo receberão um throughput maior. Em nosso exemplo, ao enviar juntos, os transmissores de 6 Mbps e 54 Mbps agora receberão 3 Mbps e 27 Mbps, respectivamente.

4.4.4 802.11: estrutura do quadro

O padrão 802.11 define três classes de quadros em trânsito: dados, controle e gerenciamento. Cada um deles tem um cabeçalho com diversos campos usados na subcamada MAC. Além disso, existem alguns cabeçalhos usados pela camada física, mas eles lidam principalmente com as técnicas de modulação empregadas e, portanto, não os discutiremos aqui.

Veremos como exemplo o formato do quadro de dados mostrado na Figura 4.29. Primeiro vem o campo *Controle de quadro*, que é composto de 11 subcampos. O primeiro desses subcampos denomina-se *Versão do protocolo*, definido como 00. Ele existe para permitir que versões futuras do 802.11 operem ao mesmo tempo na mesma célula. Depois, temos os campos *Tipo* (dados, controle ou gerenciamento) e *Subtipo* (p. ex., RTS ou CTS). Para um quadro de dados comum (sem qualidade de serviço), eles são definidos como 10 e 0000 em binário. Os bits *Para DS* e *De DS* indicam se o quadro está indo ou vindo da rede conectada aos PAs, o que é chamado de sistema de distribuição. O bit *Mais fragmentos* significa que mais fragmentos virão em seguida. O bit *Repetir* indica uma retransmissão de um quadro enviado anteriormente. O bit *Gerenciamento de energia* indica que o transmissor está entrando no modo de economia de energia. O bit *Mais dados* indica que o transmissor tem quadros adicionais para o receptor. O bit *Quadro protegido* especifica que o corpo do quadro foi criptografado por segurança. Discutiremos rapidamente sobre segurança na próxima seção. Por fim, o bit *Ordem* informa ao receptor que a camada superior espera que a sequência de quadros chegue estritamente em ordem.

O segundo campo do quadro de dados, *Duração*, informa por quanto tempo (em microssegundos) o quadro e sua confirmação ocuparão o canal. Esse campo está presente em todos os tipos de quadros, incluindo os de controle, e representa a forma como outras estações administram o mecanismo NAV.

Em seguida vêm os endereços. Os quadros de dados enviados de e para um PA contêm três endereços, todos em formato padrão IEEE 802. O primeiro é do receptor, e o segundo é do transmissor. É óbvio que eles são necessários, mas para que serve o terceiro endereço? Lembre-se de que o PA é simplesmente um ponto de repasse para os quadros enquanto trafegam entre um cliente e outro ponto na rede, talvez um cliente distante ou um portal para a Internet. O terceiro endereço indica esse ponto distante.

O campo *Sequência* numera os quadros, para que as duplicatas possam ser detectadas. Dos 16 bits disponíveis, 4 identificam o fragmento e 12 contêm um número que é avançado a cada nova transmissão. O campo *Dados* contém a carga útil de até 2.312 bytes. Os primeiros bytes dessa carga útil estão em um formato conhecido como **controle lógico do enlace**, ou **LLC (Logical Link Control)**. Essa camada é a tag (cola) que identifica o protocolo de nível mais alto (p. ex., IP) ao qual as cargas úteis devem ser passadas. Por último vem o *Checksum do quadro*, que é o mesmo CRC de 32 bits que vimos na Seção 3.2.2 e em outros lugares.

Os quadros de gerenciamento têm um formato semelhante ao dos quadros de dados, mais um formato para a parte de dados que varia com o subtipo (p. ex., parâmetros nos quadros de baliza). Os quadros de controle são curtos. Como todos os quadros, eles têm os campos de *Controle de quadro*, *Duração* e *Checksum do quadro*. Contudo, eles podem ter apenas um endereço e nenhuma parte de dados. A informação mais importante está no campo *Subtipo* (p. ex., ACK, RTS e CTS).

4.4.5 Serviços

O padrão 802.11 define que cada LAN sem fio compatível deve fornecer os serviços para clientes, para PAs e para a rede que os conecta. Esses serviços são agrupados em vários tipos.

Associação e entrega de dados

O serviço de **associação** é usado pelas estações móveis para conectá-las aos PAs. Em geral, ele é usado imediatamente

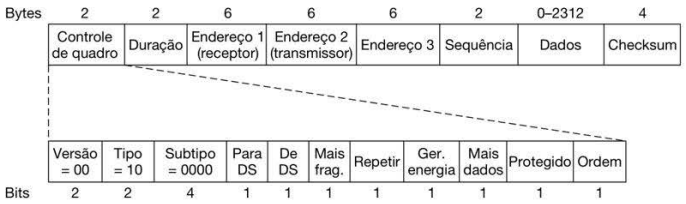


Figura 4.29 Formato do quadro de dados 802.11.

após uma estação se deslocar dentro do alcance de rádio do PA. Ao chegar, a estação descobre a identidade e os recursos do PA, seja pelos quadros de baliza, seja perguntando diretamente ao PA. Os recursos incluem as taxas de dados admitidas, os arranjos de segurança, os requisitos de economia de energia, o suporte para qualidade de serviço e outros. A mensagem de baliza do PA também inclui um **SSID (Service Set IDentifier)**, que a maioria das pessoas conhece como o nome da rede. A estação envia um pedido para se associar ao PA, o qual pode aceitá-lo ou rejeitá-lo. Embora as balizas sempre sejam um broadcast, o SSID pode ou não ser um broadcast. Se o SSID não for broadcast, a estação deverá conhecer (ou descobrir), de alguma forma, o nome para associar a esse PA.

A **reassociação** permite mudar seu PA preferido. Esse recurso é útil para estações móveis que se deslocam de um PA para outro na mesma LAN 802.11 estendida, como uma transferência (handoff) na rede celular. Se for usado corretamente, não haverá perda de dados em consequência da transferência. (Contudo, o 802.11, assim como o padrão Ethernet, é apenas um serviço que faz o melhor possível, sem garantias.) A estação móvel ou o PA também pode se **desassociar**, interrompendo assim o relacionamento. Uma estação deve usar esse serviço antes de se desligar ou sair da rede, mas o PA também pode usá-lo antes de se desativar para manutenção. O padrão 802.11w acrescentou os quadros de autenticação e desassociação.

Quando os quadros alcançam o PA, o **serviço de distribuição** determina como roteá-los. Se o destino for local para o PA, os quadros poderão ser enviados diretamente pelo ar. Caso contrário, eles terão de ser encaminhados pela rede fisicamente conectada. O **serviço de integração** trata de qualquer tradução necessária para um quadro ser enviado fora da LAN 802.11, ou para chegar de fora dela. O caso comum aqui é conectar a LAN sem fio à Internet.

A transmissão de dados é o objetivo de tudo isso, e assim o 802.11 oferece um **serviço de entrega de dados**, o qual permite que as estações transmitam e recebam dados usando os protocolos que descrevemos anteriormente no capítulo. Tendo em vista que o 802.11 foi modelado com base no padrão Ethernet e que a transmissão em uma rede Ethernet não oferece a garantia de ser 100% confiável, a transmissão sobre redes 802.11 também não oferece qualquer garantia de confiabilidade. As camadas mais altas devem lidar com a detecção e a correção de erros.

Segurança e privacidade

As estações também devem se **autenticar** antes que possam enviar quadros pelo PA, mas a autenticação é tratada de diferentes maneiras, dependendo da escolha do esquema de segurança. Se a rede 802.11 estiver "aberta", qualquer um tem permissão para usá-la. Caso contrário, são necessárias credenciais para autenticação.

Um método de autenticação comum, chamado **WPA2 (WiFi Protected Access 2)**, implementa a segurança conforme a definição no padrão 802.11i. (O WPA original é um esquema intermediário que implementa um subconjunto do 802.11i. Pularemos isso e iremos diretamente para o esquema completo.) Com o WPA2, o PA pode falar com um servidor de autenticação, que tem um banco de dados de nomes de usuários e senhas, para determinar se a estação tem permissão para acessar a rede. Como alternativa, pode-se configurar uma chave previamente compartilhada, que é um nome elegante para uma senha de rede. Vários quadros são trocados entre a estação e o PA por meio de desafio e resposta, permitindo que a estação prove que tem as credenciais corretas. Essa troca acontece após a associação.

Outro processo de autenticação comumente usado em redes corporativas é o **802.1X**, que implementa uma abordagem chamada **autenticação baseada em porta**. O 802.1X depende de autenticação centralizada (p. ex., autenticação de dispositivos em um servidor centralizado), o que cria as possibilidades de controle de acesso, contabilidade, cobrança e atribuição mais refinados. A estação que está autenticando às vezes é chamada de suplicante; esse dispositivo se autentica na rede por meio de um autenticador, que se comunica com o servidor de autenticação. O 802.1X conta com uma estrutura de autenticação chamada **EAP (Enhanced Authentication Protocol)**. A estrutura EAP define mais de 50 métodos diferentes para realizar a autenticação, mas os métodos comuns incluem **EAP-TLS**, que realiza autenticação com base em certificados; **EAP-TTLS** e **PEAP**, que permitem que o cliente se associe usando diversos métodos, incluindo autenticação baseada em senha; e **EAP-SIM**, em que um telefone móvel pode autenticar usando um SIM. O 802.1X tem muitas vantagens sobre o WPA simples, como a capacidade de executar controle de acesso minucioso com base no usuário, mas requer uma infraestrutura de certificado para administrar.

O esquema que foi usado antes do WPA se chamava **WEP (Wired Equivalent Privacy)**. Para esse esquema, a autenticação com uma chave previamente compartilhada acontece antes da associação. Agora, WEP é considerado inseguro e não é mais utilizado. A primeira demonstração prática de que o WEP foi quebrado apareceu quando Adam Stubblefield era um estagiário de verão na AT&T (Stubblefield et al., 2002). Ele foi capaz de codificar e testar um ataque em uma semana, grande parte desse tempo foi gasto para obter permissão da gerência para comprar as placas WiFi necessárias para as experiências. O software para descobrir senhas WEP agora está disponível gratuitamente.

Com o WEP falhando e o WPA desaprovado, a próxima tentativa foi o WPA2. Ele utiliza um serviço de privacidade que gerencia os detalhes da criptografia e da descriptografia. O algoritmo de criptografia para WPA2 é baseado no **padrão de criptografia avançado**, ou **AES (Advanced Encryption Standard)**, um padrão do governo dos Estados

Unidos aprovado em 2002. As chaves usadas para criptografia são determinadas durante o procedimento de autenticação. Infelizmente, o esquema do WPA2 foi quebrado em 2017 (Vanhoef e Piessens, 2017). Uma boa segurança é muito difícil, até mesmo com criptografia inquebrável, pois o gerenciamento da chave é o elo mais fraco.

Priorização e controle de energia

Para lidar com o tráfego com diferentes prioridades, existe um serviço de **tráfego QoS (o minúsculo) escalonado**. Ele usa os protocolos que descrevemos para dar tratamento preferencial ao tráfego de voz e vídeo em comparação com o melhor tráfego possível e o de segundo plano. Um serviço de acompanhamento também oferece sincronização de timer da camada mais alta. Isso permite que as estações coordenem suas ações, o que pode ser útil para o processamento de mídia.

Finalmente, existem dois serviços que ajudam as estações a gerenciar seu uso do espectro. O serviço de **controle de potência de transmissão** oferece às estações as informações que elas precisam para atender aos limites regulamentares sobre potência de transmissão, que variam de uma região para outra. O serviço de **seleção dinâmica de frequência** dá às estações a informação de que elas precisam para evitar transmitir em frequências na banda de 5 GHz que estão sendo usadas em um radar nas proximidades.

Com esses serviços, o 802.11 oferece um rico conjunto de funcionalidade para conectar à Internet clientes móveis vizinhos. Ele tem sido um grande sucesso, e o padrão repetidamente tem sido alterado para acrescentar mais funcionalidade. Para ter uma ideia de onde o padrão se encontra e para onde está se encaminhando, consulte Hiertz et al. (2010).

4.5 BLUETOOTH

Em 1994, a empresa sueca L. M. Ericsson ficou interessada em conectar seus telefones móveis a outros dispositivos (p. ex., laptops) sem cabos. Em 1998, junto com outras quatro empresas (IBM, Intel, Nokia e Toshiba), ela formou um SIG (Special Interest Group, ou seja, um consórcio) com o objetivo de desenvolver um padrão sem fio para interconectar dispositivos de computação e comunicação, além de acessórios, utilizando rádios sem fio de curto alcance, baixa potência e baixo custo. O projeto foi denominado **Bluetooth**, em homenagem a Harald Blaatand (Bluetooth) II (940-981), um rei viking que unificou (i.e., conquistou) a Dinamarca e a Noruega, também "sem cabos".

O Bluetooth 1.0 foi lançado em julho de 1999 e, desde então, o SIG nunca voltou atrás. Todo tipo de dispositivo eletrônico de consumo agora usa Bluetooth, desde telefones móveis e notebooks a headsets, impressoras, teclados, mouse, jogos, relógios, aparelhos de música, unidades de navegação e outros. Os protocolos Bluetooth permitem que esses dispositivos se encontrem e se conectem, um ato chamado **emparelhamento**, e transfiram dados com segurança.

Os protocolos também evoluíram durante a última década. Depois que os protocolos iniciais se estabilizaram, taxas de dados maiores foram acrescentadas ao Bluetooth 2.0 em 2004. Com a versão 3.0, em 2009, o Bluetooth pôde ser usado para o emparelhamento de dispositivo em combinação com o 802.11 para alcançar transferência de dados com alto throughput. A versão 4.0, em junho de 2010, especificou a operação em baixa potência. Isso será prático para pessoas que não querem trocar a bateria regularmente em todos esses dispositivos por toda a casa.

Explicaremos os principais aspectos do Bluetooth 4.0 a seguir, pois esta ainda é a versão mais utilizada. Depois, discutiremos o Bluetooth 5.0 e como ele difere da versão anterior (quase totalmente em pequenos detalhes).

4.5.1 Arquitetura do Bluetooth

Vamos começar nosso estudo do sistema Bluetooth com uma avaliação rápida do que ele contém e do que planeja fazer. A unidade básica de um sistema Bluetooth é uma **piconet**, que consiste em um nó mestre e até sete nós escravos ativos,* situados em uma distância de 10 m. Podem existir muitas piconets na mesma sala (grande) e elas podem até mesmo ser conectadas por um nó de ponte, que participa de várias piconets, como mostra a Figura 4.30. Uma coleção interconectada de piconets é chamada de **scatternet**.

Além dos sete nós escravos ativos em uma piconet, pode haver até 255 nós estacionários (inativos) na rede. Esses nós são dispositivos que o mestre comutou para um estado de baixa energia, a fim de reduzir o consumo em sua bateria. No estado estacionário, o dispositivo não pode fazer nada, exceto responder a um sinal de ativação ou de baliza do mestre. Também existem dois níveis intermediários de energia, *hold* e *sniff*, mas eles não serão estudados aqui.

A razão para a estrutura mestre/escravo é que os projetistas pretendiam facilitar a implementação de chips Bluetooth completos por menos de 5 dólares. Em consequência dessa decisão, os escravos são "não inteligentes", fazendo basicamente apenas o que o mestre determina. Em seu núcleo, uma piconet é um sistema TDM centralizado, no qual o mestre controla o clock e define qual dispositivo vai se comunicar em cada slot de tempo. Toda comunicação é feita entre o mestre e um escravo, não sendo possível a comunicação direta entre escravos.

*A terminologia oficial do Bluetooth é mestre/escravo. Pedimos desculpas se isso lhe ofender.

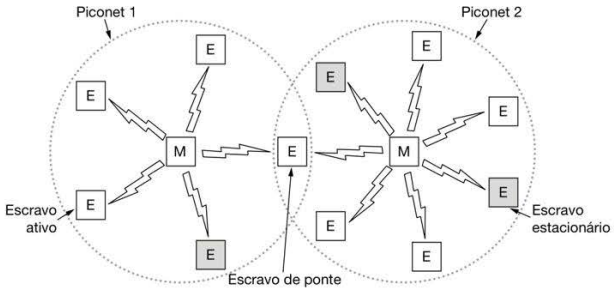


Figura 4.30 Duas piconets podem ser conectadas para formar uma scatternet.

4.5.2 Aplicações do Bluetooth

A maioria dos protocolos de rede só fornece canais entre entidades que se comunicam, deixando para os projetistas de aplicações a tarefa de descobrir a sua utilidade. Por exemplo, o 802.11 não especifica se os usuários devem usar seu notebook para ler e-mails, navegar na Web ou qualquer outra ação. Em contrapartida, a especificação Bluetooth SIG determina aplicações em particular para que tenham suporte e ofereçam diferentes pilhas de protocolos para cada um. No momento em que este livro foi escrito, havia mais de duas dúzias de aplicações específicas, chamadas **perfis**. Infelizmente, essa abordagem aumentou muito a complexidade. Omitiremos a complexidade aqui, mas veremos os perfis rapidamente, para entender de modo mais claro o que o Bluetooth SIG está tentando realizar.

Seis dos perfis são para diferentes usos de áudio e vídeo. Por exemplo, os perfis de intercomunicação permitem que dois telefones se conectem como walkie-talkies. Os perfis de headset e hands-free oferecem comunicação por voz entre um headset e sua estação-base, pois poderiam ser usados para telefonia hands-free enquanto se dirige um carro. Outros perfis são para streaming de áudio e vídeo com qualidade estéreo, digamos, de um aparelho de música portátil para fones de ouvido, ou de uma câmera digital para uma TV.

O perfil de dispositivo de interface humana é para conectar teclado e mouse aos computadores. Outros perfis permitem que um telefone móvel ou outro computador receba imagens de uma câmera ou envie imagens para uma impressora. Talvez seja mais interessante um perfil para usar um telefone móvel como um controle remoto para uma TV (habilitada para Bluetooth).

Outros perfis ainda permitem o uso de rede. O perfil de rede pessoal permite que dispositivos Bluetooth formem uma rede ad hoc ou acessem outra rede remotamente, como uma LAN 802.11, por meio de um PA. O perfil de rede discada foi realmente a motivação original para o projeto inteiro. Ele permite que um notebook se conecte a um telefone móvel contendo um modem embutido, sem usar fios, apenas sinais de rádio.

Os perfis para troca de informações da camada mais alta também foram definidos. O perfil de sincronização serve para carregar dados para um telefone móvel quando ele sai de casa e coleta dados dele ao retornar.

Pularemos o restante dos perfis, exceto para mencionar que alguns servem como blocos de montagem sobre os quais os perfis citados são baseados. O perfil de acesso genérico, no qual todos os outros perfis são baseados, oferece um modo de estabelecer e manter enlaces seguros (canais) entre o mestre e os escravos. Os outros perfis genéricos definem os fundamentos da troca de objeto e transporte de áudio e vídeo. Os perfis utilitários são muito usados para funções como emular uma linha serial, o que é especialmente útil para muitas aplicações legadas.

Seria realmente necessário explicar todas essas aplicações em detalhes e fornecer diferentes pilhas de protocolos para cada uma? É provável que não, mas surgiram diversos grupos de trabalho que elaboraram partes distintas do padrão, e cada um se concentrou em seu problema específico, gerando seu próprio perfil. Imagine tudo isso como uma aplicação da lei de Conway. (Na edição de abril de 1968 da revista *Datamation*, Melvin Conway observou que, se designar *n* pessoas para escrever um compilador, você obterá um compilador de *n* passagens ou, de modo mais geral, a estrutura de software reflete a estrutura do grupo que o produziu.) Provavelmente teria sido possível concluir o trabalho com duas pilhas de protocolos em vez de 25, uma para transferência de arquivos e uma para comunicação em tempo real.

4.5.3 A pilha de protocolos do Bluetooth

O padrão Bluetooth tem muitos protocolos agrupados livremente em camadas, como mostra a Figura 4.31. A primeira observação a fazer é que a estrutura não segue o modelo

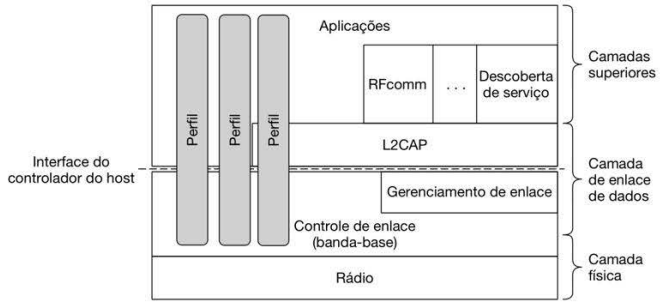


Figura 4.31 A arquitetura de protocolos do Bluetooth.

OSI, o modelo TCP/IP, o modelo 802 ou qualquer outro modelo conhecido.

A camada inferior é a camada física de rádio, que corresponde muito bem à camada física nos modelos OSI e 802. Ela lida com a transmissão e a modulação de rádio. Muitas das preocupações aqui estão relacionadas ao objetivo de tornar o sistema mais econômico, para que possa vir a ser um item do mercado de massa.

A camada de controle de enlace (ou banda-base) é de certa forma análoga à subcamada MAC, mas também inclui elementos da camada física. Ela lida com a maneira como o mestre controla os slots de tempo e como esses slots estão agrupados em quadros.

Em seguida, temos dois protocolos que usam o protocolo de controle de enlace. O gerenciador de enlaces cuida do estabelecimento de canais lógicos entre dispositivos, incluindo gerenciamento de energia, emparelhamento e criptografia, e qualidade de serviço. Ele se encontra abaixo da linha da interface do controlador do host. Essa interface é uma conveniência para a implementação: normalmente, os protocolos abaixo da linha serão implementados em um chip Bluetooth, e os acima dela serão implementados no dispositivo Bluetooth que hospeda o chip.

O protocolo de enlace acima da linha é o **L2CAP** (**Logical Link Control Adaptation Protocol**). Ele enquadra mensagens de tamanho variável e oferece confiabilidade, se necessário. Muitos protocolos utilizam L2CAP, como os dois protocolos utilitários mostrados. O protocolo de descoberta de serviço é usado para localizar serviços dentro da rede e o protocolo RFcomm (comunicação por radiofrequência) simula a porta serial padrão encontrada nos PCs para a conexão de teclado, mouse e modem, entre outros dispositivos.

A camada superior é onde as aplicações estão localizadas. Os perfis são representados por caixas verticais, pois cada uma delas define uma fatia da pilha de protocolos para determinada finalidade. Perfis específicos, como o de headset, normalmente contêm apenas os protocolos necessários para essa aplicação e nenhum outro. Por exemplo, os perfis podem incluir o L2CAP se tiverem pacotes para enviar, mas o pulam se tiverem apenas um fluxo contínuo de amostras de áudio.

Nas próximas seções, examinaremos a camada de rádio Bluetooth e diversos protocolos de enlace, pois eles correspondem aproximadamente à camada física e à subcamada MAC nas outras pilhas de protocolos que estudamos.

4.5.4 A camada de rádio do Bluetooth

A camada de rádio move os bits do mestre para o escravo, ou vice-versa. Ela é um sistema de baixa potência com um alcance de 10 m, operando na banda ISM de 2,4 GHz, como o 802.11. A banda está dividida em 79 canais de 1 MHz cada um. Para coexistir com as outras redes usando a banda ISM, é utilizado o espectro de espalhamento por salto de frequência. Pode haver até 1.600 saltos/s pelos slots com um tempo de parada de 625 μs. Todos os nós em uma piconet mudam de frequência simultaneamente, seguindo a temporização de slot e a sequência de salto pseudoaleatória ditada pelo mestre.

Infelizmente, as primeiras versões de Bluetooth e 802.11 interferiram o suficiente para arruinar as transmissões um do outro. Algumas empresas responderam banindo o Bluetooth completamente, mas, por fim, uma solução técnica foi elaborada. A solução é que o Bluetooth adapte sua sequência de saltos para excluir canais em que existam outros sinais de ondas de rádio, o que reduz a interferência prejudicial. Esse processo é chamado de **salto de frequência adaptativo**.

Três formas de modulação são usadas para enviar bits em um canal. O esquema básico é usar o chaveamento por mudança de frequência para enviar um símbolo de 1 bit a cada microssegundo, dando uma taxa de dados bruta de 1 Mbps. As taxas melhoradas foram introduzidas com a versão 2.0 do Bluetooth. Essas taxas utilizam o chaveamento por deslocamento de fase para enviar 2 ou 3 bits por símbolo, para taxas de dados brutas de 2 ou 3 Mbps. As taxas melhoradas são usadas apenas na parte de dados dos quadros.

4.5.5 As camadas de enlace do Bluetooth

A camada de controle de enlace (ou banda-base) é a estrutura mais próxima de uma subcamada MAC que o Bluetooth tem. Ela transforma o fluxo bruto de bits em quadros e define alguns formatos importantes. Em sua forma mais simples, o mestre em cada piconet define uma série de slots de tempo de 625 μs, com as transmissões do mestre começando nos slots pares e as transmissões dos escravos começando nos slots ímpares. Esse esquema é a tradicional multiplexação por divisão de tempo (TDM), em que o mestre fica com metade dos slots e os escravos compartilham a outra metade. Os quadros podem ter 1, 3 ou 5 slots de duração. Cada quadro tem um overhead de 126 bits para um código de acesso e cabeçalho, mais um tempo de acomodação de 250-260 μs por salto, para permitir que os circuitos de rádio se estabilizem. Por questão de confidencialidade, a carga útil do quadro pode ser criptografada, com uma chave escolhida quando o mestre e o escravo se conectam. Os saltos só acontecem entre os quadros, e não durante um quadro. O resultado é que um quadro de 5 slots é muito mais eficiente do que um quadro de 1 slot, pois o overhead é constante, porém mais dados são enviados.

O protocolo gerenciador de enlace estabelece canais lógicos, chamados **enlaces**, para transportar quadros entre um dispositivo mestre e um escravo que descobriram um ao outro. Um procedimento de emparelhamento é seguido para garantir que os dois dispositivos tenham permissão para se comunicar antes que o enlace seja usado. O antigo método de emparelhamento determinava que os dois dispositivos fossem configurados com o mesmo número de identificação pessoal, ou PIN (Personal Identification Number), de quatro dígitos. O PIN correspondente é o modo como cada dispositivo sabe que está se conectando ao dispositivo remoto correto. Contudo, usuários e dispositivos sem criatividade usam PINs padrão, como "0000" e "1234", significando que, na prática, esse método fornece pouquíssima segurança.

O novo método de **emparelhamento simples seguro** permite que os usuários confirmem se os dois dispositivos estão exibindo a mesma passkey, ou que a observem em um dispositivo e a insiram no segundo dispositivo. Esse método é mais seguro, pois os usuários não precisam escolher ou definir um PIN. Eles simplesmente confirmam uma passkey mais longa, gerada pelo dispositivo. Naturalmente, ela não pode ser usada em alguns dispositivos com entrada/saída limitada, como um headset portátil.

Quando o emparelhamento está concluído, o protocolo do gerenciador de enlace estabelece os enlaces. Existem dois tipos principais de enlaces para transportar a carga útil (dados do usuário). O primeiro é o enlace **síncrono orientado a conexões**, ou **SCO** (**Synchronous Connection Oriented**). Ele é usado para dados em tempo real, como conexões de telefone. Esse tipo de enlace aloca um slot fixo em cada sentido. Um escravo pode ter até três enlaces SCO com seu mestre. Cada enlace SCO pode transmitir um canal de áudio PCM de 64.000 bps. Em virtude da natureza crítica no tempo dos enlaces SCO, os quadros enviados por eles nunca são retransmitidos. Em vez disso, para aumentar a confiabilidade, pode-se usar a correção de erros antecipada.

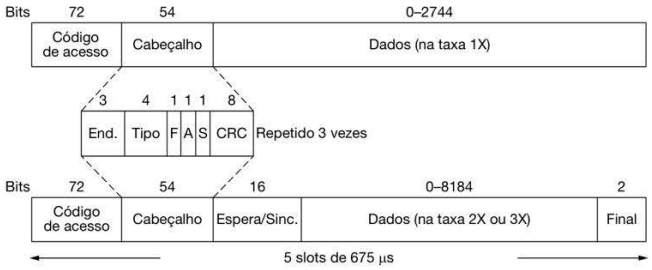
O outro tipo é o enlace **assíncrono não orientado a conexões**, ou **ACL** (**Asynchronous ConnectionLess**). Esse tipo de enlace é usado para dados de comutação de pacotes, disponíveis em intervalos irregulares. O tráfego ACL é entregue com base no melhor serviço possível, sem garantias. Os quadros podem se perder e precisar ser retransmitidos. Um escravo só pode ter um enlace ACL com seu mestre.

Os dados enviados por enlaces ACL vêm da camada L2CAP, que tem quatro funções principais. Primeiro, ela aceita pacotes de até 64 KB das camadas superiores e os divide em quadros para transmissão. Na extremidade distante, os quadros são montados novamente em pacotes. Em segundo lugar, ela lida com a multiplexação e a demultiplexação de várias origens de pacotes. Quando um pacote é montado novamente, a L2CAP determina a qual protocolo da camada superior ele será entregue, por exemplo, RFcomm ou descoberta de serviço. Terceiro, lida com controle de erros e retransmissão. Ela detecta os erros e retransmite os pacotes que não foram confirmados. Por fim, impõe requisitos de qualidade de serviço entre enlaces múltiplos.

4.5.6 A estrutura de quadro do Bluetooth

Há vários formatos de quadros definidos no Bluetooth, mas o mais importante é apresentado de duas formas na Figura 4.32. Ele começa com um código de acesso que normalmente identifica o mestre, para que os escravos situados dentro do alcance de rádio de dois mestres possam conhecer o destino de cada tráfego. Em seguida, há um cabeçalho de 54 bits contendo campos típicos da subcamada MAC. Se o quadro for enviado na taxa de transferência básica, o campo de dados vem em seguida. Ele tem até 2.744 bits para uma transmissão de cinco slots. Para um único slot de tempo, o formato é o mesmo, exceto pelo fato de o campo de dados ter 240 bits.

Se o quadro for enviado na taxa melhorada, a parte de dados pode ter até duas ou três vezes a quantidade de bits, pois cada símbolo transporta 2 ou 3 bits em vez de 1. Esses dados são precedidos por um campo de espera e um padrão de sincronismo que é usado para mudar para a taxa de dados mais rápida. Ou seja, o código de acesso e o cabeçalho



(a) Acima, o quadro de dados na taxa básica (b) Abaixo, o quadro de dados na taxa melhorada

Figura 4.32 Um quadro de dados típico do Bluetooth nas taxas de dados (a) básica e (b) melhorada.

são transportados na taxa básica e somente a parte dos dados é transportada na taxa mais rápida. Os quadros na taxa melhorada terminam com um final curto.

Vamos examinar rapidamente o cabeçalho. O campo *Endereço* identifica qual dos oito dispositivos ativos é o destino do quadro. O campo *Tipo* identifica o tipo de quadro (ACL, SCO, polling ou nulo), o tipo de correção de erros usado no campo de dados, e de quantos slots é a duração do quadro. O bit *Fluxo* é definido por um escravo quando seu buffer está cheio e não pode receber mais dados. Esse bit habilita uma forma primitiva de controle de fluxo. O bit *Confirmação* é usado para transportar uma mensagem ACK "de carona" em um quadro. O bit *Sequência* é usado para numerar os quadros, a fim de detectar retransmissões. O protocolo é stop-and-wait e, assim, 1 bit é suficiente. Em seguida, temos o cabeçalho de 8 bits *Checksum*. O cabeçalho de 18 bits inteiro é repetido três vezes para formar o cabeçalho de 54 bits mostrado na Figura 4.32. No lado receptor, um circuito simples examina as três cópias de cada bit. Se todas forem iguais, o bit será aceito. Caso contrário, vence a opinião da maioria. Desse modo, 54 bits de capacidade de transmissão são usados para enviar 10 bits de cabeçalho. A razão para isso é que, para transmitir dados de maneira confiável em um ambiente ruidoso usando dispositivos de baixo custo e de baixa potência (2,5 mW), com pouca capacidade de computação, é necessária uma grande redundância.

São usados vários formatos para o campo de dados de quadros ACL e SCO. Entretanto, os quadros SCO na taxa básica são mais simples: o campo de dados tem sempre 240 bits. São definidas três variantes, permitindo 80, 160 ou 240 bits de carga útil real, sendo os bits restantes usados para a correção de erros. Na versão mais confiável (carga útil de 80 bits), o conteúdo é simplesmente repetido três vezes, da mesma forma que o cabeçalho.

Podemos calcular a capacidade com esse quadro da seguinte forma: tendo em vista que o escravo só pode usar os slots ímpares, ele recebe 800 slots/s, da mesma maneira que o mestre. Com uma carga útil de 80 bits, a capacidade de canal do escravo é de 64.000 bps, assim como a capacidade de canal do mestre. Essa capacidade é exatamente o bastante para um único canal de voz PCM full-duplex (e esse é o motivo de ter sido escolhida uma taxa de saltos de 1.600 saltos/s). Ou seja, apesar de uma largura de banda bruta de 1 Mbps, um único canal de voz satura completamente a piconet. A eficiência de 10% é o resultado de gastar 41% da capacidade no tempo de preparação, 20% com cabeçalhos e 26% na codificação de repetição. Essa deficiência destaca o valor de taxas melhoradas e quadros com mais de um único slot.

4.5.7 Bluetooth 5

Em junho de 2016, o Bluetooth Special Interest Group introduziu o Bluetooth 5. Em janeiro de 2019, apareceu o Bluetooth 5.1. Estas foram atualizações relativamente pequenas ao padrão Bluetooth 4. Apesar disso, existem algumas diferenças entre o Bluetooth 4 e ambos os padrões Bluetooth 5. Aqui está uma lista das principais diferenças no Bluetooth 5.0:

1. Suporte para dispositivos da IoT (Internet das Coisas).
2. A velocidade foi aumentada de 1 Mbps para 2 Mbps.
3. O tamanho da mensagem subiu de 31 bytes para 255 bytes.
4. O alcance interno passou de 10 m para 40 m.
5. Os requisitos de potência foram ligeiramente reduzidos.
6. O intervalo das balizas aumentou ligeiramente.
7. A segurança é ligeiramente melhor.

No todo, não houve muita mudança, mas, dada a necessidade de compatibilidade, não era de se esperar que houvesse. O padrão Bluetooth 5.1 teve algumas pequenas atualizações nas áreas de rastreamento de dispositivo, caching e alguns outros itens secundários.

4.6 DOCSIS

A rede de TV a cabo foi projetada originalmente para levar programas de televisão aos lares. Agora, ela também é muito utilizada como uma alternativa ao sistema telefônico, para levar Internet aos lares. A seguir, descrevemos a "camada MAC" no padrão DOCSIS, implementada pela maioria dos provedores de serviço a cabo.

4.6.1 Visão geral

De certa forma, a especificação DOCIS também possui uma subcamada MAC, embora ela seja um pouco menos distinta da camada de enlace do que em outros protocolos, conforme estudamos em capítulos anteriores. No entanto, o protocolo tem vários aspectos que se enquadram nos objetivos padrão da subcamada MAC, incluindo alocação de canal (que ocorre por meio de um processo de solicitação-concessão), configuração de qualidade de serviço e um modelo de encaminhamento exclusivo. Esta seção trata de todas essas questões. Mais recentemente, o DOCSIS 3.1 full-duplex (agora chamado DOCSIS 4.0) introduziu novas tecnologias para escalonamento e cancelamento de interferência.

O DOCSIS tem um formato de quadro MAC padrão, que abarca um conjunto de campos, incluindo o comprimento do quadro MAC, uma soma de verificação e um campo de cabeçalho estendido, que oferece suporte a uma variedade de funções, incluindo a segurança da camada de enlace. Alguns cabeçalhos oferecem suporte a funções específicas, incluindo temporização de downstream, ajuste de energia de upstream, solicitações de largura de banda e concatenação de quadros. Um tipo específico de quadro é chamado de quadro de solicitação, que é a forma como o modem a cabo solicita a largura de banda, conforme descrito mais adiante nesta seção.

4.6.2 Alcance

Um modem a cabo transmite o que é chamado de solicitação de **variação**, que permite ao CMTS (headend) determinar o atraso da rede até o modem a cabo, bem como realizar os ajustes de potência necessários. A variação é efetivamente o ajuste periódico de vários parâmetros de transmissão, especificamente temporização, frequência e potência. O CMTS sonda o modem a cabo, o que o aciona para enviar uma solicitação de variação. Com base nessa mensagem, o CMTS fornece ao modem uma resposta para ajudá-lo a ajustar a temporização e a potência de transmissão do sinal. Por padrão, o intervalo ocorre aproximadamente uma vez a cada 30 segundos, mas pode ser configurado para ocorrer com mais frequência; intervalos de variação típicos podem ser de cerca de 10 a 20 segundos.

4.6.3 Alocação de largura de banda do canal

Um CMTS DOCSIS reserva largura de banda para cada modem a cabo por meio de um processo de solicitação-concessão. Cada fluxo de tráfego upstream ou downstream normalmente recebe um fluxo de serviço, e cada fluxo de serviço tem sua largura de banda alocada pelo CMTS.

Fluxos de serviço

Em geral, a alocação de canais no DOCSIS envolve a alocação de canais entre um **CMTS** e um ou mais modems a cabo, que estão localizados nas casas dos assinantes. O CMTS deve atender a todos os canais upstream e downstream e descarta qualquer quadro com um endereço MAC de origem que não seja um dos modems a cabo atribuídos no grupo. De grande importância para a camada MAC do DOCSIS é a noção de um **fluxo de serviço**, que fornece uma maneira de gerenciar a qualidade da gestão do serviço tanto upstream quanto downstream. Cada modem a cabo tem um ID de fluxo de serviço associado, que é negociado durante o registro do modem a cabo; cada modem a cabo pode ter vários fluxos de serviço associados. Diferentes fluxos de serviço podem ter diferentes limitações associadas a diferentes tipos de tráfego. Por exemplo, cada fluxo de serviço pode ter um tamanho máximo de pacote, ou, então, um fluxo de serviço pode ser dedicado a um determinado tipo de aplicativo, como um aplicativo de taxa de bits constante. Todos os modems a cabo devem suportar pelo menos um fluxo de serviço upstream e um downstream, chamado de fluxo de serviço primário.

O processo de solicitação-concessão e o DOCSIS de baixa latência

Quando um modem a cabo tem dados para enviar, ele faz uma solicitação curta que informa ao CMTS quantos dados ele deve enviar e aguarda uma mensagem de alocação de largura de banda subsequente, que descreve as oportunidades de transmissão upstream que um transmissor pode ter para transmitir dados.

A transmissão upstream é dividida em intervalos discretos por um mecanismo de alocação de largura de banda upstream chamado **minislot**. Um minislot é simplesmente uma unidade de granularidade de tempo para transmissão upstream, normalmente em incrementos de 6,25 μs. Dependendo da versão do DOCSIS, um minislot pode precisar ser um múltiplo de potência de dois desse incremento; nas versões mais modernas, essa restrição não se aplica. Ajustando os minislots que são concedidos a um fluxo de serviço específico, o CMTS pode implementar com eficácia a qualidade de serviço e a priorização para diferentes fluxos de tráfego.

De modo geral, a qualidade do serviço permitiu ao CMTS alocar mais largura de banda para diferentes modems a cabo (permitindo assim que um assinante que é provisionado para um nível de serviço superior alcance um nível de serviço superior). Mais recentemente, no entanto, as revisões do DOCSIS também permitiram um serviço diferenciado para aplicações sensíveis à latência. Especificamente, uma nova revisão do protocolo DOCSIS permite baixa latência, por meio de uma nova especificação chamada **LLD** (**Low-Latency DOCSIS**). O LLD reconhece que, para muitas aplicações interativas, como jogos e videoconferência, a baixa latência é tão importante quanto o alto throughput. Em alguns casos, em redes DOCSIS existentes, a latência para alguns fluxos pode ser bastante alta, devido ao tempo de aquisição da mídia compartilhada e ao tempo de enfileiramento.

O LLD trata desses problemas reduzindo o atraso de ida e volta associado ao processo de concessão de solicitação e usando duas filas – uma para tráfego de aplicações sensíveis à latência e outra para tráfego que não é sensível à latência. O menor atraso de solicitação-concessão reduz a quantidade de tempo que o CMTS usa para realizar cálculos de agendamento, dos 2–4 milissegundos anteriores para 1 milissegundo. O LLD também usa mecanismos para escalonar concessões proativamente a um fluxo de serviço, a fim de eliminar totalmente o atraso associado ao processo de solicitação-concessão. Também permite que as aplicações determinem se têm pacotes que não podem ser enfileirados, por meio da marcação de um campo de serviço diferenciado no quadro DOCSIS. Para obter mais informações sobre LLD, consulte White (2019).

4.7 COMUTAÇÃO NA CAMADA DE ENLACE DE DADOS

Muitas empresas têm diversas LANs e desejam conectá-las. Não seria conveniente se pudéssemos apenas juntar as LANs para criar uma LAN maior? De fato, podemos fazer isso quando as conexões são feitas com dispositivos chamados **bridges**. Os switches Ethernet que descrevemos na Seção 4.3.4 são um nome moderno para as bridges; eles oferecem funcionalidade que vai além da Ethernet clássica e de hubs Ethernet, facilitando a junção de várias LANs em uma rede maior e mais rápida. Usaremos os termos "bridge" e "switch" para indicar a mesma coisa.

As bridges operam na camada de enlace de dados, de modo que examinam os endereços nessa camada para encaminhar quadros. Tendo em vista que elas não têm de examinar o campo de carga útil dos quadros que encaminham, as bridges podem tratar dos pacotes IP ou de quaisquer outros tipos de pacotes, como os pacotes AppleTalk. Em contrapartida, os *roteadores* examinam os endereços em pacotes e efetuam o roteamento com base neles, de modo que só trabalham com os protocolos para os quais foram projetados para lidar.

Nesta seção, examinaremos como as bridges funcionam e como são usadas para juntar várias LANs físicas em uma única LAN lógica. Também veremos como realizar o inverso e tratar uma LAN física como múltiplas LANs lógicas, chamadas LANs virtuais. As duas tecnologias oferecem flexibilidade útil para o gerenciamento de redes. Para ver um estudo abrangente sobre bridges, switches e tópicos relacionados, consulte Perlman (2000) e Yu (2011).

4.7.1 Uso de bridges

Antes de iniciarmos o estudo da tecnologia de bridges, vale a pena examinar algumas situações comuns em que elas são usadas. Mencionaremos três razões pelas quais uma única organização pode ter várias LANs.

Primeiro, muitas universidades e departamentos de empresas têm suas próprias LANs, principalmente para conectar seus computadores pessoais, servidores e dispositivos como impressoras. Como os objetivos dos diversos departamentos são diferentes, muitos deles escolhem LANs distintas, sem se importar com o que outros setores estão fazendo. Mais cedo ou mais tarde, surge a necessidade de interação, por isso as bridges são necessárias. Nesse exemplo, a existência de diversas LANs deve-se à autonomia de seus proprietários.

Segundo, a organização pode estar geograficamente dispersa em vários edifícios separados por distâncias consideráveis. Talvez seja mais econômico ter LANs separadas em cada edifício e conectá-las com bridges e enlaces de fibra óptica por longa distância que estender todos os cabos até um único switch central. Mesmo que estender os cabos fosse fácil de fazer, existem limites para seu tamanho (p. ex., 200 m para a gigabit Ethernet com par trançado). A rede não funcionaria para cabos maiores em virtude de atenuação excessiva do sinal ou pelo atraso de ida e volta. A única solução é partir a LAN e instalar bridges para juntar as partes, aumentando a distância física total que pode ser coberta.

Terceiro, talvez seja necessário dividir aquilo que logicamente é uma única LAN em LANs separadas (conectadas por bridges), a fim de acomodar a carga. Por exemplo, em muitas universidades grandes, há milhares de estações de trabalho disponíveis para as necessidades de computação dos funcionários e dos alunos. As empresas também podem ter milhares de funcionários. A escala desse sistema impede que se coloquem todas as estações de trabalho em uma única LAN – existem mais computadores do que portas em qualquer hub Ethernet, e mais estações do que é permitido em uma única Ethernet clássica.

Mesmo que fosse possível conectar todas as estações de trabalho com fios, colocar mais estações em um hub

Ethernet ou na Ethernet clássica não aumentaria a capacidade. Todas as estações compartilham a mesma quantidade fixa de largura de banda. Quanto mais estações houver, menor a largura de banda média por estação.

Entretanto, duas LANs separadas têm o dobro da capacidade de uma única LAN. As bridges permitem que as LANs sejam reunidas enquanto mantêm essa capacidade. O importante é não enviar tráfego para portas onde ele não seja necessário, para que cada LAN possa trabalhar em velocidade plena. Esse comportamento também aumenta a confiabilidade, pois, em uma única LAN, um nó com defeito que continua enviando um fluxo contínuo de lixo pode travar a LAN inteira. Decidindo o que encaminhar e o que não encaminhar, as bridges atuam como portas de incêndio em um prédio, impedindo que um único nó descontrolado trave o sistema inteiro.

Para tornar esses benefícios facilmente disponíveis, o ideal é que as bridges sejam completamente transparentes. Deverá ser possível sair e comprar bridges, conectar os cabos da LAN nas bridges e tudo funcionar perfeitamente, instantaneamente. Não deve ser preciso fazer mudanças de hardware ou de software, nem configuração de endereço de switches, nem baixar tabelas de roteamento ou de parâmetros, nada mesmo. Basta conectar os cabos e sair. Além disso, a operação das LANs existentes não deverá ser afetada de forma alguma pelas bridges. Em relação às estações, não deverá haver diferença observável por estarem ou não fazendo parte de uma LAN com bridge. Deverá ser tão fácil mover estações pela LAN com bridge quanto em uma LAN isolada.

É surpreendente como realmente é possível criar bridges transparentes. Dois algoritmos são utilizados: um de aprendizado reverso, para evitar que o tráfego seja enviado para onde não é necessário, e um spanning tree, para interromper loops que possam ser formados quando os switches são conectados de forma incorreta. Agora, vejamos esses algoritmos, um por vez, para aprender como essa mágica é realizada.

4.7.2 Learning bridges

A topologia de duas LANs unidas por bridge aparece na Figura 4.33 para dois casos. No lado esquerdo, duas LANs multidrop, como as Ethernets clássicas, são unidas por uma estação especial – a bridge – que fica nas duas LANs. No lado direito, as LANs com cabos ponto a ponto, incluindo um hub, são reunidas. As bridges são os dispositivos aos quais as estações e o hub são conectados. Se a tecnologia de LAN é Ethernet, as bridges são mais bem conhecidas como switches Ethernet.

As bridges foram desenvolvidas quando as Ethernets clássicas estavam sendo usadas, de modo que normalmente aparecem em topologias com cabos multidrop, como na Figura 4.33(a). Todavia, todas as topologias encontradas hoje são compostas de cabos e switches ponto a ponto. As bridges funcionam da mesma maneira nas duas configurações. Todas as estações conectadas à mesma porta em uma bridge pertencem ao mesmo domínio de colisão, e este é diferente do domínio de colisão para outras portas. Se houver mais de uma estação, como em uma Ethernet clássica, um hub ou um enlace half-duplex, o protocolo CSMA/CD é usado para transmitir quadros.

Contudo, há uma diferença no modo como são montadas as LANs conectadas com bridges. Para unir LANs multidrop, uma bridge é acrescentada como uma nova estação em cada uma das LANs multidrop, como na Figura 4.33(a). Para unir LANs ponto a ponto, os hubs são conectados a uma bridge ou, de preferência, substituídos por uma bridge, para aumentar o desempenho. Na Figura 4.33(b), as bridges substituíram todos menos um hub.

Diferentes tipos de cabos também podem ser conectados a uma bridge. Por exemplo, o cabo que conecta a bridge *B1* à *B2* na Figura 4.33(b) poderia ser um enlace de fibra óptica de longa distância, enquanto o cabo que conecta as bridges às estações poderia ser um cabo de par trançado em curta distância. Esse arranjo é útil para unir LANs em prédios diferentes.

Agora, vamos considerar o que acontece dentro das bridges. Cada uma delas opera em modo promíscuo, ou

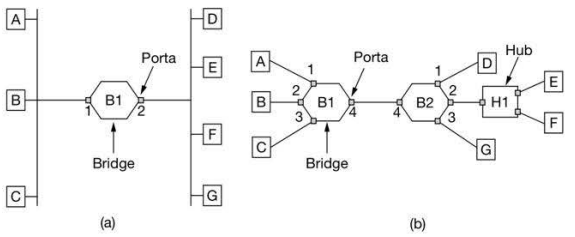


Figura 4.33 (a) Bridge conectando duas LANs multidrop. (b) Bridges (e um hub) conectando sete estações ponto a ponto.

seja, aceita cada quadro transmitido pelas estações conectadas a cada uma das portas. A bridge precisa decidir se encaminhará ou descartará cada quadro e, no primeiro caso, a que porta o enviará. Essa decisão é tomada usando o endereço de destino. Como exemplo, considere a topologia da Figura 4.33(a). Se a estação *A* enviar um quadro à estação *B*, a bridge *B1* receberá o quadro na porta 1. Esse quadro pode ser descartado imediatamente, sem mais cerimônias, pois já está na porta correta. Contudo, na topologia da Figura 4.33(b), suponha que *A* envie um quadro para *D*. A bridge *B1* receberá o quadro na porta 1 e o enviará pela porta 4. A bridge *B2* receberá, então, o quadro em sua porta 4 e o enviará pela sua porta 1.

Um modo simples de implementar esse esquema é ter uma grande tabela (hash) dentro da bridge. Ela pode listar cada destino possível e a que porta de saída ele pertence. Por exemplo, na Figura 4.33(b), a tabela em *B1* listaria *D* como pertencente à porta 4, pois tudo o que *B1* precisa saber é em que porta colocar os quadros para alcançar *D*. Ou seja, na verdade, haverá outros encaminhamentos, caso o quadro que alcança *B2* não seja de interesse para *B1*.

Quando as bridges são conectadas pela primeira vez, todas as tabelas de hash estão vazias. Nenhuma das bridges sabe onde estão os destinatários e, por isso, elas usam o algoritmo de inundação: cada quadro de entrada para um destino desconhecido é enviado para todas as portas às quais a bridge está conectada, com exceção da porta de onde o quadro chegou. Com o passar do tempo, as bridges aprendem onde estão os destinatários. A partir do momento em que um destinatário se torna conhecido, os quadros destinados a ele são colocados somente na porta apropriada e não são mais inundados para as demais.

O algoritmo usado pelas bridges é o de **aprendizado reverso**. Como já dissemos, as bridges operam em modo promíscuo, portanto, elas veem todo quadro enviado em qualquer uma de suas portas. Examinando o endereço de origem, elas podem descobrir que máquina está acessível em quais portas. Por exemplo, se a bridge *B1* da Figura 4.33(b) vir um quadro na porta 3 vindo de *C*, ela saberá que *C* pode ser alcançada através da porta 3, assim, ela cria uma entrada em sua tabela hash. Qualquer quadro subsequente endereçado a *C* que chegue na *B1* ou em qualquer outra porta será encaminhado para a porta 3.

A topologia pode ser alterada à medida que máquinas e bridges são ativadas, desativadas e deslocadas. Para tratar de topologias dinâmicas, sempre que é criada uma entrada da tabela hash o tempo de chegada do quadro é indicado na entrada. Sempre que chega um quadro cuja origem já está na tabela, sua entrada é atualizada com a hora atual. Desse modo, o tempo associado a cada entrada informa a última vez que foi visto um quadro proveniente dessa máquina.

Periodicamente, um processo na bridge varre a tabela hash e elimina todas as entradas que tenham mais de alguns minutos. Dessa forma, se um computador for desconectado de sua LAN, levado para outro lugar no prédio e reconectado nesse outro local, dentro de poucos minutos ele estará de volta à operação normal, sem nenhuma intervenção manual. Esse algoritmo também significa que, se uma máquina estiver inativa por alguns minutos, qualquer tráfego enviado a ela terá de ser difundido por inundação, até que ela mesma envie um quadro em seguida.

O procedimento de roteamento para um quadro de entrada depende da porta em que ele chega (a porta de origem) e do endereço ao qual ele se destina (o endereço de destino). O procedimento é o seguinte:

1. Se a porta para o endereço de destino e a porta de origem forem uma só, o quadro será descartado.
2. Se a porta para o endereço de destino e a porta de origem forem diferentes, o quadro será encaminhado para a porta de destino.
3. Se a porta de destino for desconhecida, o quadro será inundado e enviado para todas as portas, com exceção da porta de origem.

Você pode estar questionando se o primeiro caso pode ocorrer com enlaces ponto a ponto. A resposta é que ele pode ocorrer se forem usados hubs para conectar um grupo de computadores a uma bridge. Um exemplo aparece na Figura 4.33(b), em que as estações *E* e *F* estão conectadas ao hub *H1*, que por sua vez está conectado à bridge *B2*. Se *E* envia um quadro para *F*, o hub o repassará para *B2*, bem como para *F*. É isso que os hubs fazem – conectam todas as portas de modo que um quadro que chega a uma porta é simplesmente enviado por todas as outras portas. O quadro chegará a *B2* na porta 2, que já é a porta de saída certa para alcançar o destino. A bridge *B2* só precisa descartar o quadro.

À medida que cada quadro chegar, esse algoritmo será aplicado, de modo que ele normalmente é implementado com chips VLSI de uso especial. Os chips pesquisam e atualizam a entrada na tabela, em alguns microssegundos. Como as bridges só examinam os endereços MAC para decidir como encaminhar os quadros, é possível começar a encaminhar assim que o campo do cabeçalho de destino tiver chegado, antes que o restante do quadro tenha chegado (é claro, desde que a linha de saída esteja disponível). Esse projeto reduz a latência da passagem pela bridge, bem como o número de quadros que a bridge terá de manter em buffer. Ele é conhecido como **comutação cut-through** ou **roteamento wormhole**, e normalmente é tratado no hardware.

Podemos ver a operação de uma bridge em termos de pilhas de protocolo para entender o que significa ser um dispositivo da camada de enlace. Considere um quadro enviado da estação *A* para a estação *D* na configuração da Figura 4.33(a), em que as LANs são Ethernet. O quadro passará por uma bridge. A visão de processamento da pilha de protocolos aparece na Figura 4.34.

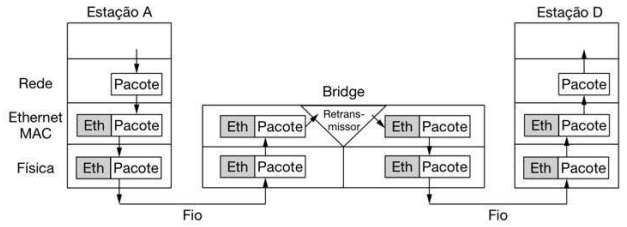


Figura 4.34 Processamento de protocolos em uma bridge.

O pacote vem de uma camada mais alta e desce até a camada Ethernet MAC. Ele adquire um cabeçalho Ethernet (e também um final, não mostrado na figura). Essa unidade é passada para a camada física, sai pelo cabo e é apanhada pela bridge.

Na bridge, o quadro é passado da camada física para a camada Ethernet MAC. Essa camada estende o processamento em comparação com a camada Ethernet MAC em uma estação. Ela passa o quadro para um retransmissor, ainda dentro da camada MAC. O serviço de retransmissão da bridge usa apenas o cabeçalho Ethernet MAC para determinar como lidar com o quadro. Nesse caso, ela passa o quadro para a camada Ethernet MAC da porta usada para atingir a estação *D*, e o quadro continua seu caminho.

No caso geral, os retransmissores em determinada camada podem reescrever os cabeçalhos dessa camada. As LANs virtuais oferecerão um exemplo em breve. A bridge nunca deve examinar o interior do quadro e descobrir que ele está transportando um pacote IP; isso é irrelevante para o processamento interno da bridge e violaria o uso do modelo em camadas do protocolo. Observe também que uma bridge com *k* portas terá *k* ocorrências de camadas MAC e física. O valor de *k* é 2 para nosso exemplo simples.

4.7.3 Spanning Tree Bridges

Para aumentar a confiabilidade, os enlaces redundantes podem ser usados entre as bridges. No exemplo da Figura 4.35, existem dois enlaces em paralelo entre um par de bridges. Esse projeto garante que, se um enlace for interrompido, a rede não será dividida em dois conjuntos de computadores que não podem conversar entre si.

Entretanto, essa redundância também introduz alguns problemas adicionais, porque cria loops na topologia. Podemos ver um exemplo simples desses problemas observando como um quadro enviado por *A* para um destino previamente não observado é tratado na Figura 4.35. Cada bridge segue as regras normais para tratamento de destinos desconhecidos, que é inundar o quadro. Vamos chamar o quadro de *A* que alcança a bridge *B1* de quadro F_0. A bridge envia cópias desse quadro por todas as suas outras portas. Só consideraremos as portas da bridge que conectam *B1* a *B2* (embora o quadro também seja enviado para as outras portas). Como existem dois enlaces de *B1* para *B2*, duas cópias do quadro alcançarão *B2*. Elas são mostradas na Figura 4.35 como F_1 e F_2.

Pouco tempo depois, a bridge *B2* recebe esses quadros. Contudo, ela não sabe (e não pode saber) que eles são cópias do mesmo quadro, em vez de dois quadros diferentes enviados um após o outro. Assim, a bridge *B2* apanha F_1 e envia cópias dele para todas as outras portas, e também apanha F_2 e envia cópias dele por todas as outras portas. Isso produz quadros F_3 e F_4, enviados ao longo dos dois enlaces para *B1*. A bridge *B1*, então, vê dois novos quadros com destinos desconhecidos e os copia novamente. Esse ciclo prossegue indefinidamente.

A solução para essa dificuldade é estabelecer a comunicação entre as bridges e sobrepor a topologia real com uma spanning tree que alcance cada bridge. Na realidade, algumas conexões potenciais entre as bridges são ignoradas para que se construa uma topologia fictícia livre de loops, que é um subconjunto da topologia real.

Por exemplo, na Figura 4.36, vemos cinco bridges interconectadas que também têm estações conectadas a elas. Essa estação se conecta a apenas uma bridge. Existem algumas conexões redundantes entre as bridges para que os quadros sejam encaminhados em loops se todos os enlaces forem usados. Essa topologia pode ser considerada um grafo em que as bridges são os nós e os enlaces ponto a ponto

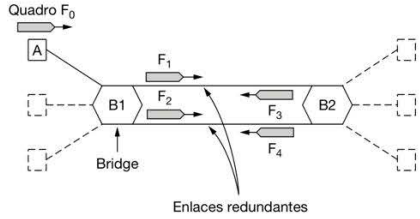


Figura 4.35 Bridges com dois enlaces paralelos.

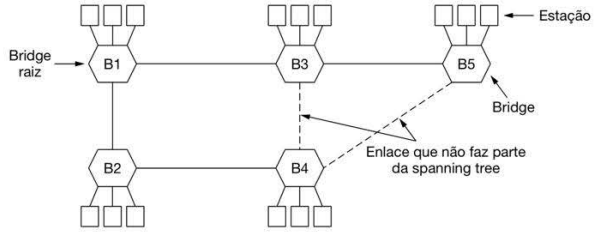


Figura 4.36 Uma spanning tree conectando cinco bridges. As linhas pontilhadas não fazem parte da spanning tree.

são as arestas. O grafo pode ser reduzido a uma spanning tree que, por definição, não contém ciclos, eliminando os enlaces mostrados como linhas tracejadas na Figura 4.36. Com a utilização da spanning tree, existe exatamente um caminho de cada estação para qualquer outra estação. Uma vez que as bridges entram em acordo em relação à spanning tree, tudo o que é encaminhado entre as estações segue a spanning tree. Como existe um único caminho de cada origem até cada destino, é impossível haver loops.

Para construir a spanning tree, as bridges executam um algoritmo distribuído. Cada bridge transmite periodicamente por broadcast uma mensagem de configuração por todas as suas portas aos seus vizinhos e processa as mensagens que recebe de outras bridges, conforme descreveremos a seguir. Essas mensagens não são encaminhadas, pois seu propósito é construir a árvore, que pode, então, ser usada para o encaminhamento.

Primeiro as bridges precisam escolher, entre elas, a que será usada como raiz. Para fazer essa escolha, cada uma delas inclui um identificador com base no endereço MAC na mensagem de configuração, bem como o identificador da bridge que se acredita ser a raiz. Os endereços MAC são instalados pelo fabricante com a garantia de ser exclusivos em todo o mundo, o que torna esses identificadores convenientes e exclusivos. As bridges escolhem aquela com o menor identificador para ser a raiz. Depois que mensagens suficientes tiverem sido trocadas para espalhar a notícia, todas as bridges chegarão a um acordo sobre qual delas é a raiz. Na Figura 4.36, a bridge *B1* tem o menor identificador e se torna a raiz.

Em seguida, é construída uma árvore de caminhos mais curtos a partir da raiz até cada bridge. Na Figura 4.36, as bridges *B2* e *B3* podem ser alcançadas diretamente a partir da *B1*, em um salto que é o caminho mais curto. A *B4* pode ser alcançada em dois saltos, por meio de *B2* ou *B3*. Para desempatar, é escolhido o caminho por meio da bridge com o menor identificador, de modo que *B4* é alcançada por meio de *B2*. A *B5* pode ser alcançada em dois saltos por meio de *B3*.

Para encontrar esses caminhos mais curtos, as bridges incluem a distância a partir da raiz em suas mensagens de configuração. Cada uma delas se lembra do caminho mais curto que encontra até a raiz. As bridges, então, desativam as portas que não fazem parte do caminho mais curto.

Embora a árvore se espalhe por todas as bridges, nem todos os enlaces (ou mesmo as bridges) estão necessariamente presentes na árvore. Isso acontece porque o desligamento das portas poda alguns enlaces da rede e impede loops. Mesmo depois que a spanning tree é estabelecida, o algoritmo continua a funcionar durante a operação normal, com a finalidade de detectar automaticamente mudanças na topologia e atualizar a árvore.

O algoritmo usado para a construção da spanning tree foi inventado por Radia Perlman. Seu trabalho foi resolver o problema de juntar LANs sem loops. Ela teve uma semana para fazer isso, mas teve a ideia do algoritmo spanning tree em um dia. Felizmente, ela teve tempo suficiente para escrevê-lo em forma de poema (Perlman, 1985):

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span.
So packets can reach every LAN.
First the Root must be selected
By ID it is elected.
Least cost paths from Root are traced
In the tree these paths are placed.
A mesh is made by folks like me
*Then bridges find a spanning tree.**

O algoritmo spanning tree foi, então, padronizado como IEEE 802.1D e usado por muitos anos. Em 2001, ele foi revisado para encontrar mais rapidamente uma nova spanning tree após uma mudança de topologia. Para ver um estudo mais detalhado sobre as bridges, consulte Perlman (2000).

*N. de T. (Creio que nunca verei/Um grafo melhor que uma árvore./Uma árvore cuja propriedade fundamental/É a conectividade sem loops./Uma árvore que precisa se espalhar/Para que os pacotes possam alcançar cada LAN./Primeiro a raiz deve ser selecionada/Por ID ela é eleita./Caminhos de menor custo a partir da raiz são traçados./Na árvore esses caminhos são colocados./Uma malha é feita por pessoas como eu/Depois as bridges acham uma spanning tree.)

4.7.4 Repetidores, hubs, bridges, switches, roteadores e gateways

Até agora neste livro, examinamos diversas maneiras de transferir quadros e pacotes de um computador para outro. Mencionamos repetidores, hubs, bridges, switches, roteadores e gateways. Todos esses dispositivos são de uso comum, mas diferem entre si em detalhes sutis e não muito sutis. Por existir uma grande quantidade desses dispositivos, provavelmente vale a pena examiná-los em conjunto para ver quais são as semelhanças e as diferenças entre eles.

A chave para entendê-los é observar que eles operam em camadas diferentes, como ilustra a Figura 4.37(a). A camada é importante, porque diferentes dispositivos utilizam fragmentos de informações diferentes para decidir como realizar a comutação. Em um cenário típico, o usuário gera alguns dados a serem enviados para uma máquina remota. Esses dados são repassados à camada de transporte, que então acrescenta um cabeçalho (p. ex., um cabeçalho TCP) e repassa o pacote resultante à camada de rede situada abaixo dela. Essa camada adiciona seu próprio cabeçalho para formar um pacote da camada de rede (p. ex., um pacote IP). Na Figura 4.37(b), vemos o pacote IP sombreado na cor cinza. Em seguida, o pacote vai para a camada de enlace de dados, que adiciona seu próprio cabeçalho e seu checksum (CRC) e entrega o quadro resultante à camada física para transmissão, digamos, por uma LAN.

Agora, vamos examinar os dispositivos de comutação e ver como eles se relacionam aos pacotes e quadros. Na parte inferior, na camada física, encontramos os repetidores. Estes são dispositivos analógicos que trabalham com sinais nos cabos aos quais estão conectados. Um sinal que aparece em um deles é limpo, amplificado e colocado em outro cabo. Os repetidores não reconhecem quadros, pacotes ou cabeçalhos. Eles entendem os símbolos codificados em bits como volts. Por exemplo, a Ethernet clássica foi projetada para permitir quatro repetidores, que amplificam o sinal a fim de estender o comprimento máximo do cabo de 500 para 2.500 m.

Em seguida, temos os hubs. Um hub tem várias interfaces de entrada que ele conecta eletricamente. Os quadros que chegam a quaisquer dessas interfaces são enviados a todas as outras. Se dois quadros chegarem ao mesmo tempo, eles colidirão, exatamente como ocorre em um cabo coaxial. Todas as linhas que chegam a um hub devem operar na mesma velocidade. Os hubs diferem dos repetidores pelo fato de (normalmente) não amplificarem os sinais de entrada e serem projetados para conter várias linhas de entrada, mas as diferenças são pequenas. Assim como os repetidores, os hubs são dispositivos da camada física que não examinam os endereços da camada de enlace nem os utilizam de maneira alguma.

Agora, vamos subir até a camada de enlace de dados, em que encontramos bridges e switches. Acabamos de estudar as bridges com certa profundidade. Uma bridge conecta duas ou mais LANs. Como um hub, uma bridge moderna tem várias portas, normalmente o suficiente para 4 a 48 linhas de entrada de um certo tipo. Diferentemente de um hub, cada porta é isolada para ser seu próprio domínio de colisão; se a porta tem uma linha ponto a ponto full-duplex, o algoritmo CSMA/CD não é necessário. Quando um quadro chega, a bridge extrai o endereço de destino do cabeçalho de quadro e examina uma tabela, a fim de verificar para onde deve enviá-lo. No caso de uma rede Ethernet, esse endereço é o endereço de destino de 48 bits mostrado na Figura 4.14. A bridge só envia o quadro à porta onde ele é necessário, e pode encaminhar vários quadros ao mesmo tempo.

As bridges oferecem desempenho muito melhor que os hubs, e o isolamento entre suas portas também significa que as linhas de entrada podem trabalhar com diferentes velocidades, possivelmente ainda com diferentes tipos de rede. Um exemplo comum é uma bridge com portas que se conectam à Ethernet de 10, 100 e 1.000 Mbps. O uso de buffer dentro da bridge é necessário para aceitar um quadro em uma porta e transmiti-lo por uma porta diferente.

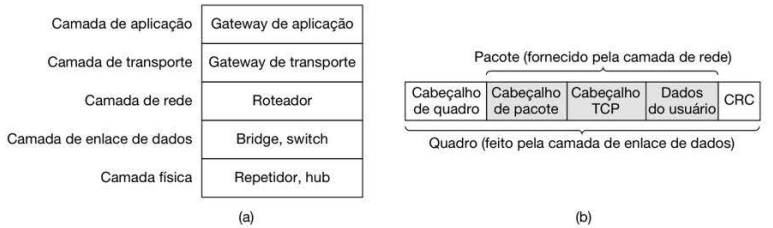


Figura 4.37 (a) Dispositivos presentes em cada camada. (b) Quadros, pacotes e cabeçalhos.

Se os quadros entrarem mais rapidamente do que podem ser retransmitidos, a bridge poderá ficar sem espaço em buffer e ter de começar a descartar quadros. Por exemplo, se uma gigabit Ethernet estiver empurrando bits para uma Ethernet de 10 Mbps na velocidade máxima, a bridge terá de mantê-los em buffer, na esperança de não ficar sem memória. Esse problema ainda existe mesmo que todas as portas trabalhem na mesma velocidade, pois mais de uma porta pode estar enviando quadros a determinada porta de destino.

As bridges visavam originalmente à união de diferentes tipos de LANs, por exemplo, uma LAN Ethernet e uma Token Ring. Contudo, isso nunca funcionou bem, em razão das diferenças entre elas. Diferentes formatos de quadro exigem cópia e reformatação, o que requer tempo de CPU, um novo cálculo de checksum e introduz a possibilidade de erros não detectados, em decorrência de bits incorretos na memória da bridge. O uso de diferentes tamanhos máximos de quadro também é um problema sério sem uma boa solução. Basicamente, quadros muito grandes para ser encaminhados devem ser descartados. Muita coisa para se evidenciar.

Duas áreas em que as LANs podem diferir são segurança e qualidade de serviço. Algumas têm criptografia da camada de enlace (p. ex., 802.11) e outras não (p. ex., Ethernet). Algumas têm recursos de qualidade de serviço, como prioridades (p. ex., 802.11) e outras não (p. ex., Ethernet). Consequentemente, quando um quadro precisa trafegar entre essas LANs, pode não ser possível fornecer a segurança ou a qualidade de serviço esperadas pelo emissor. Por todos esses motivos, as bridges modernas normalmente funcionam para um tipo de rede, e os roteadores, que veremos em breve, são usados em seu lugar para unir redes de diferentes tipos.

Os switches são bridges modernas com outro nome. As diferenças são mais por questões de marketing do que técnicas, mas existem alguns pontos que precisam ser conhecidos. As bridges foram desenvolvidas quando a Ethernet clássica estava em uso, de modo que tendem a unir relativamente poucas LANs e, portanto, ter relativamente poucas portas. O termo "switch" é mais popular hoje em dia. Além disso, todas as instalações modernas usam enlaces ponto a ponto, como cabos de par trançado, de modo que computadores individuais se conectam diretamente a um switch e, portanto, este costuma ter muitas portas. Finalmente, "switch" também é usado como um termo geral. Com uma bridge, a funcionalidade é clara. Em contrapartida, um switch pode se referir a um switch Ethernet ou a um tipo de dispositivo completamente diferente, que toma decisões de encaminhamento, como um switch usado em telefonia.

Até o momento, vimos repetidores e hubs, que são bastante semelhantes, bem como bridges e switches, que são ainda mais parecidos. Agora vamos passar para os roteadores, os quais são diferentes de todos os dispositivos anteriores. Quando um pacote entra em um roteador, o cabeçalho de quadro e o final são retirados, e o pacote localizado no campo de carga útil do quadro (sombreado na Figura 4.37) é repassado ao software de roteamento. Esse software utiliza o cabeçalho de pacote para escolher uma linha de saída. No caso de um pacote IP, o cabeçalho do pacote conterá um endereço de 32 bits (IPv4) ou de 128 bits (IPv6), mas não um endereço IEEE 802 de 48 bits. O software de roteamento não vê os endereços de quadro e nem mesmo sabe se o pacote veio de uma LAN ou de uma linha ponto a ponto. Estudaremos os roteadores e o roteamento no Capítulo 5.

Subindo até outra camada, encontramos gateways de transporte. Esses dispositivos conectam dois computadores que utilizam diferentes protocolos de transporte orientados a conexões. Por exemplo, suponha que um computador que utiliza o protocolo TCP/IP orientado a conexões precise se comunicar com um computador que utiliza um protocolo de transporte orientado a conexões diferente, chamado SCTP. O gateway de transporte pode copiar os pacotes de uma conexão para a outra, reformatando-os caso seja necessário.

Finalmente, os gateways de aplicação reconhecem o formato e o conteúdo dos dados e convertem mensagens de um formato para outro. Por exemplo, um gateway de correio eletrônico poderia converter mensagens da Internet em mensagens SMS para telefones móveis. Assim como "switch", "gateway" é um termo bastante genérico. Ele se refere a um processo de encaminhamento que atua em uma camada superior.

4.7.5 LANs virtuais

Quando foram criadas as primeiras redes locais, grossos cabos amarelos se estendiam pelos conduítes de muitos prédios comerciais. Todo computador era conectado a esses cabos por onde eles passavam. Ninguém parava para pensar sobre que computador pertencia a qual LAN. Todas as pessoas que trabalhavam em escritórios adjacentes tinham seus equipamentos conectados à mesma LAN, quer elas trabalhassem juntas, quer não. A geografia era mais importante que os organogramas corporativos.

Com o advento do par trançado e dos hubs na década de 1990, tudo isso mudou. A fiação dos prédios foi trocada (a um custo considerável) para eliminar todos os grossos cabos amarelos e instalar pares trançados, que iam de cada escritório até os armários centrais de fiação instalados no fim de cada corredor ou em uma sala de máquinas central, como ilustra a Figura 4.38. Se o encarregado da fiação fosse um visionário, eram instalados pares trançados da Categoria 5; se fosse um avarento, a fiação telefônica existente (da Categoria 3) era usada (até ser substituída alguns anos mais tarde, quando surgiu a Fast Ethernet).

Hoje, os cabos mudaram e os hubs se tornaram switches, mas o padrão de fiação ainda é o mesmo. Esse padrão

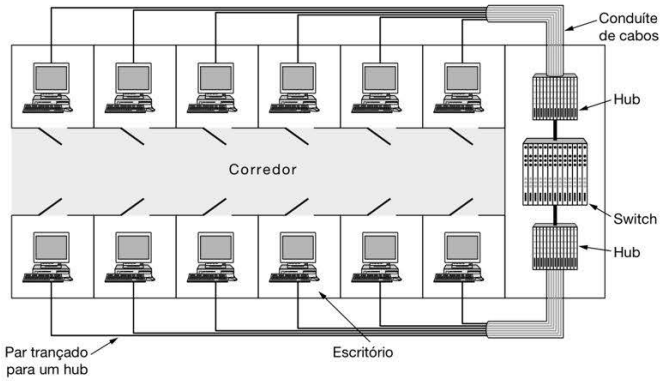


Figura 4.38 Um prédio com fiação centralizada, utilizando hubs e um switch.

possibilita configurar LANs logicamente, em vez de fisicamente. Por exemplo, se uma empresa deseja *k* LANs, ela pode comprar *k* switches. Escolhendo cuidadosamente quais conectores ligar a quais switches, os ocupantes de uma LAN podem ser escolhidos de um modo que faça sentido para a organização, sem considerar muito a geografia.

É importante saber quem está conectado a cada LAN? Afinal, em quase todas as organizações, todas as LANs estão interconectadas. A resposta é sim, isso com frequência é importante. Os administradores de redes gostam de agrupar os usuários em LANs de modo a refletir a estrutura organizacional, em lugar do layout físico do prédio, por várias razões. Uma delas é a segurança. Uma LAN poderia hospedar servidores Web e outros computadores voltados para uso público. Outra LAN poderia hospedar computadores que contivessem os registros do departamento de recursos humanos, que não devem ser passados para fora do departamento. Nessa situação, faz sentido colocar todos os computadores em uma única LAN e não permitir que nenhum servidor seja acessado de fora dela. A gerência tende a franzir a testa quando escuta que esse arranjo é impossível.

Uma segunda questão é a carga. Algumas LANs são utilizadas mais intensamente que outras, e pode ser interessante separá-las. Por exemplo, se o pessoal da área de pesquisa estiver realizando várias experiências e alguma delas sair do controle e saturar a LAN, é bem possível que o pessoal da gerência não fique muito entusiasmado por ter de doar uma parte de sua capacidade de computação reservada para uma videoconferência para ajudar os colegas do outro departamento. Novamente, isso pode causar, na gerência, a impressão da necessidade de instalar uma rede mais rápida.

Uma terceira questão é o tráfego de broadcast. As bridges enviam tráfego de broadcast quando o local do destino é desconhecido, e os protocolos da camada superior também o utilizam. Por exemplo, quando um usuário quer enviar um pacote a um endereço IP representado por *x*, como saber qual endereço MAC colocar no quadro? Estudaremos essa questão no Capítulo 5, mas, em resumo, o usuário transmitirá um quadro contendo a seguinte pergunta: "A quem pertence o endereço IP *x*?". Em seguida, o usuário aguardará uma resposta. À medida que o número de comunicações em uma LAN aumenta, o mesmo acontece com a quantidade de broadcasts circulando. Cada broadcast consome mais capacidade da LAN do que um quadro normal, pois ele é entregue a cada computador na LAN. Evitando que as LANs sejam maiores do que precisam ser, reduzimos o impacto do tráfego de broadcast.

Um problema relacionado ao broadcast é que, de vez em quando, uma interface de rede sofrerá uma pane e começará a gerar um fluxo infinito de quadros de broadcast. Se a rede realmente estiver sem sorte, alguns desses quadros gerarão respostas que levarão a ainda mais tráfego. O resultado dessa **tempestade de broadcast** é que (1) a capacidade da LAN inteira será ocupada por esses quadros e (2) todas as máquinas em todas as LANs interconectadas serão danificadas, processando e descartando todos os quadros que estiverem sendo transmitidos.

A princípio, pode parecer que seria possível limitar o escopo das tempestades de broadcast separando as LANs com bridges ou switches; porém, se o objetivo é conseguir transparência (i.e., poder mover uma máquina de uma LAN diferente usando a bridge sem que alguém note a mudança), então as bridges têm de encaminhar quadros de broadcast.

Depois de verificarmos por que seria interessante para as empresas ter várias LANs com escopo restrito, vamos voltar ao problema de desacoplar a topologia lógica

da topologia física. A criação de uma topologia física para refletir a estrutura organizacional pode acrescentar trabalho e custo, mesmo com fiação centralizada e switches. Por exemplo, se duas pessoas no mesmo departamento trabalham em prédios diferentes, pode ser mais fácil conectá-las a diferentes switches que fazem parte de LANs diferentes. Mesmo que esse não seja o caso, um usuário poderia ser deslocado dentro da empresa de um departamento para outro sem mudar de escritório, ou poderia mudar de escritório sem mudar de departamento. Isso pode fazer o usuário estar na LAN errada até que o administrador mude o conector do usuário de um switch para outro. Além disso, o número de computadores que pertencem a diferentes departamentos pode não corresponder bem ao número de portas nos switches; alguns departamentos podem ser muito pequenos e outros tão grandes que exigem vários switches. Isso resulta em portas do switch desperdiçadas, que não são usadas.

Em muitas empresas, as mudanças organizacionais ocorrem o tempo todo; isso significa que os administradores de sistemas passam muito tempo retirando plugues e inserindo-os de novo em algum outro lugar. Além disso, em alguns casos, a mudança não pode ser feita de modo algum, porque o par trançado da máquina do usuário está longe demais do hub correto (p. ex., em outro prédio), ou então as portas do switch disponíveis estão na LAN errada.

Em resposta à solicitação de usuários que desejam maior flexibilidade, os fornecedores de redes começaram a buscar um meio de recompor a fiação dos prédios inteiramente via software. O conceito resultante é chamado LAN virtual, ou **VLAN (Virtual LAN)** e foi até mesmo padronizado pelo comitê IEEE 802. Atualmente, ele está sendo implementado em muitas organizações. Vamos examiná-lo em seguida.

As VLANs se baseiam em switches especialmente projetados para reconhecê-las. Para configurar uma rede baseada em VLANs, o administrador da rede decide quantas delas haverá, quais computadores estarão em qual VLAN e qual será o nome de cada uma. Geralmente, elas são identificadas (informalmente) por cores, pois assim é possível imprimir diagramas de cores que mostram o layout físico das máquinas, com os membros da LAN vermelha em vermelho, os membros da LAN verde em verde, e assim por diante. Desse modo, os layouts físico e lógico são visíveis em um único diagrama.

Como exemplo, considere as LANs conectadas por bridges da Figura 4.39, em que nove das máquinas pertencem à VLAN G (*gray* – cinza) e cinco pertencem à VLAN W (*white* – branca). As máquinas da VLAN cinza estão espalhadas por dois switches, incluindo as duas máquinas que se conectam a um switch por meio de um hub.

Para fazer as VLANs funcionarem corretamente, é necessário definir tabelas de configuração nas bridges. Essas tabelas informam quais são as VLANs acessíveis através de cada uma das portas. Quando um quadro chega, digamos, da VLAN cinza, ele tem de ser encaminhado para todas as portas marcadas com um G. Isso é válido para o tráfego comum (i.e., de unicast) para o qual as pontes não descobriram o local do destino, bem como para os tráfegos de multicast e de broadcast. Observe que uma porta pode ser rotulada com várias cores de VLAN.

Como um exemplo, suponha que uma das estações cinza conectadas à bridge *B1* na Figura 4.39 envie um quadro para um destino que não tenha sido observado anteriormente. A bridge *B1* receberá o quadro e verá que ele veio de uma máquina na VLAN cinza e, por isso, o quadro inundará todas as portas rotuladas com G (exceto a porta de chegada). O quadro será enviado às cinco outras estações cinza conectadas a *B1*, além do enlace de *B1* até a bridge *B2*. Na *B2*, o quadro será igualmente encaminhado por todas as portas rotuladas com G. Isso envia o quadro a uma estação adiante e ao hub (que transmitirá o quadro a todas as suas estações). O hub tem os dois rótulos, pois se conecta a máquinas das duas VLANs. O quadro não é enviado pelas outras portas sem G no rótulo, pois a bridge sabe que não existem máquinas na VLAN cinza que possam ser alcançadas por meio dessas portas.

Em nosso exemplo, o quadro é enviado apenas da bridge *B1* para a bridge *B2*, pois existem máquinas na VLAN cinza que estão conectadas a *B2*. Examinando a VLAN branca, podemos ver que a porta da *B2* que se conecta à *B1 não* está rotulada com W. Isso significa que um quadro na VLAN branca não será encaminhado da *B2* para

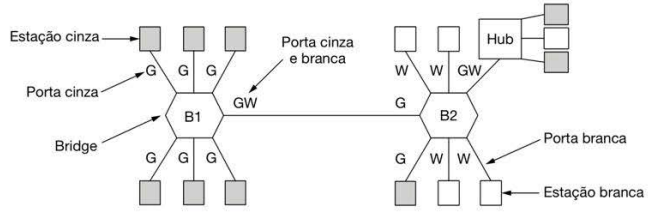


Figura 4.39 Duas VLANs, cinza e branca, em uma LAN com bridge.

a *B1*. Esse comportamento é correto, pois nenhuma estação na VLAN branca está conectada à *B1*.

O padrão IEEE 802.1Q

Para implementar esse esquema, as bridges precisam saber a qual VLAN pertence um quadro que chega. Sem essa informação, por exemplo, quando a bridge *B2* receber um quadro da bridge *B1*, na Figura 4.39, ela não sabe se encaminha o quadro na VLAN cinza ou branca. Se estivéssemos criando um novo tipo de LAN, seria muito fácil apenas acrescentar um campo VLAN no cabeçalho. Mas o que fazer no caso do padrão Ethernet, que é a LAN dominante e que não tem um campo sobressalente que possa ser usado como identificador da VLAN?

O comitê 802 do IEEE enfrentou esse problema em 1995. Depois de muita discussão, ele fez o inconcebível e mudou o cabeçalho do padrão Ethernet. O novo formato foi publicado no padrão **802.1Q** do IEEE, emitido em 1998. O novo formato contém uma tag de VLAN; vamos examiná-lo rapidamente. Não surpreende que a mudança de algo tão bem estabelecido quanto o cabeçalho Ethernet não seja inteiramente trivial. Algumas questões que surgem são:

1. Precisaremos jogar fora várias centenas de milhões de placas Ethernet existentes?
2. Se não, quem gerará os novos campos?
3. O que acontecerá com os quadros que já têm o tamanho máximo?

É claro que o comitê 802 estava (ainda que de forma muito dolorosa) consciente desses problemas e teve de apresentar soluções, o que realmente fez.

A chave para a solução é perceber que os campos VLAN só são realmente usados pelas bridges e switches, e *não* pelas máquinas dos usuários. Desse modo, na Figura 4.39, não é realmente essencial que eles estejam presentes nas linhas que saem para as estações finais, desde que estejam na linha entre as bridges. Portanto, para usar VLANs, as bridges têm de reconhecer a VLAN. Esse fato torna o projeto viável.

Quanto a descartar todas as placas Ethernet existentes, a resposta é não. Lembre-se de que o comitê 802.3 não poderia nem mesmo fazer as pessoas transformarem o campo *Tipo* em um campo *Tamanho*. Você pode imaginar a reação ao anúncio de que todas as placas Ethernet existentes teriam de ser jogadas fora. Contudo, as novas placas Ethernet são compatíveis com o 802.1Q e podem preencher corretamente os campos VLAN.

Como pode haver computadores (e switches) que não reconhecem a VLAN, a primeira bridge que a reconhece e toca em um quadro acrescenta os campos de VLAN e a última no caminho os remove. Um exemplo de topologia mista aparece na Figura 4.40. Nela, os computadores que reconhecem a VLAN geram quadros marcados (ou seja, 802.1Q) diretamente, e a comutação posterior utiliza essas tags. Os símbolos sombreados são máquinas que reconhecem VLANs e os símbolos vazios não as reconhecem.

Com o 802.1Q, os quadros são coloridos dependendo da porta na qual são recebidos. Para que esse método funcione, todas as máquinas em uma porta precisam pertencer à mesma VLAN, o que reduz a flexibilidade. Por exemplo, na Figura 4.40, essa propriedade é mantida para todas as portas nas quais um computador individual se conecta a uma bridge, mas não para a porta na qual o hub se conecta a *B2*.

Além disso, a bridge pode usar o protocolo da camada mais alta para selecionar a cor. Desse modo, os quadros que chegam a uma porta podem ser colocados em VLANs diferentes, dependendo se elas transportam pacotes IP ou quadros PPP.

Outros métodos são viáveis, mas não são admitidos pelo 802.1Q. Como exemplo, o endereço MAC pode ser usado para selecionar a cor da VLAN. Isso poderia ser útil para quadros chegando de uma LAN 802.11 próxima, em que notebooks enviam quadros por portas diferentes enquanto se movem. Um endereço MAC seria, então, mapeado a uma VLAN fixa, independentemente da porta em que ele entrou na LAN.

Quanto ao problema de quadros maiores que 1.518 bytes, o 802.1Q simplesmente aumentou o limite para 1.522 bytes.

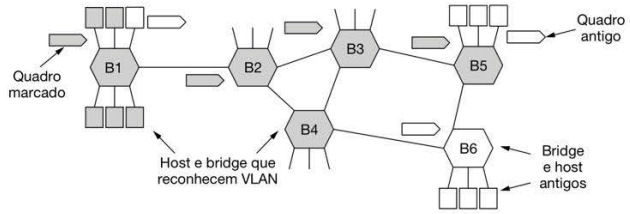


Figura 4.40 LAN com bridge, parcialmente consciente da VLAN. Os sombreados são máquinas que reconhecem VLANs e os vazios não as reconhecem.

Felizmente, apenas computadores e switches que reconhecem a VLAN precisam dar suporte a eles.

Agora, vamos examinar o formato de quadro 802.1Q. Ele está representado na Figura 4.41. A única mudança é o acréscimo de um par de campos de 2 bytes. O primeiro é o campo *ID de protocolo de VLAN*, que sempre tem o valor 0x8100. Como esse número é maior que 1.500, todas as placas Ethernet o interpretam como um tipo, e não como um tamanho. O que uma placa antiga faz com um quadro desse tipo é discutível, pois tais quadros não deveriam ser enviados a placas antigas.

O segundo campo de 2 bytes contém três subcampos. O principal é o *Identificador de VLAN*, que ocupa os 12 bits de baixa ordem. É isso que interessa – a cor da VLAN à qual o quadro pertence. O campo de 3 bits *Prioridade* não tem nenhuma relação com VLANs, mas, como a mudança no cabeçalho Ethernet é um evento que acontece uma vez a cada década, demora três anos e envolve uma centena de pessoas, por que não incluir alguns outros benefícios? Esse campo torna possível distinguir o tráfego em tempo real permanente do tráfego em tempo real provisório e do tráfego não relacionado ao tempo, a fim de fornecer melhor qualidade de serviço nas redes Ethernet. Ele é necessário para voz sobre a Ethernet (embora o IP tivesse um campo semelhante durante um quarto de século sem que ninguém jamais o tenha usado).

O último campo, o indicador de formato canônico, ou *CFI* (*Canonical Format Indicator*), deveria ter sido chamado indicador de ego corporativo, ou *CEI* (*Corporate Ego Indicator*). Originalmente, ele foi criado para indicar endereços MAC little-endian *versus* endereços MAC big-endian, mas esse uso se perdeu em outras controvérsias. Sua presença agora indica que a carga útil contém um quadro 802.5 congelado que está esperando encontrar outra LAN 802.5 no destino, enquanto é transportado por uma rede Ethernet nesse meio-tempo. É claro que toda essa organização não tem nenhuma relação com as VLANs. No entanto, a política do comitê de padrões não é diferente da política comum: se você votar a favor do meu bit, eu votarei a favor do seu – uma negociação mais sofisticada que uma barganha comum.

Como já dissemos, quando um quadro marcado chega a um switch que reconhece VLANs, o switch utiliza a ID da VLAN como um índice em uma tabela para descobrir por meio de que portas deve enviar o quadro. Contudo, de onde vem a tabela? Se ela for construída manualmente, voltaremos à estaca zero: a configuração manual de bridges. A beleza da ponte transparente é o fato de ela ser plug-and-play e não exigir qualquer configuração manual. Seria uma vergonha terrível perder essa propriedade. Felizmente, as bridges que reconhecem VLANs também podem se autoconfigurar com base na observação das tags que passam por elas. Se um quadro marcado como VLAN 4 chegar à porta 3, então aparentemente alguma máquina na porta 3 está na VLAN 4. O padrão 802.1Q explica como construir as tabelas dinamicamente, em grande parte por meio de referências a partes apropriadas do padrão 802.1D.

Antes de encerrarmos o assunto de roteamento de VLANs, vale a pena fazer uma última observação. Muitas pessoas no universo da Internet e das redes Ethernet defendem, de forma enfática, a interligação de redes não orientadas a conexões e se opõem violentamente a qualquer sinal de conexões na camada de enlace de dados ou na camada de rede. Ainda assim, as VLANs introduzem algo surpreendentemente semelhante a uma conexão. Para usar as VLANs de forma apropriada, cada quadro transporta um novo identificador especial que é usado como um índice para uma tabela interna do switch, a fim de procurar o local para onde o quadro deve ser enviado. É isso mesmo que acontece nas redes orientadas a conexões. Nas redes não orientadas a conexões, deve-se utilizar o endereço de destino para roteamento e não alguma espécie de identificador de conexão. Veremos mais detalhes sobre essa rejeição às conexões no Capítulo 5.

4.8 RESUMO

Algumas redes administram todo o fluxo de comunicações por meio de um único canal. Nessas redes, a grande questão é a alocação desse canal entre as estações que desejam utilizá-lo. FDM e TDM são esquemas de alocação simples

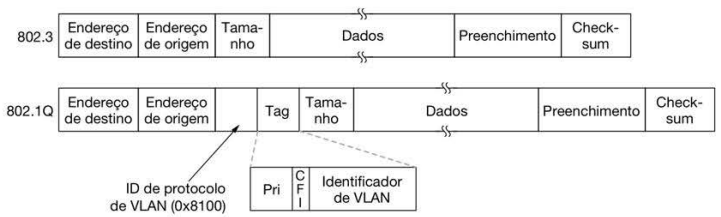


Figura 4.41 Os formatos de quadros Ethernet 802.3 (antigo) e 802.1Q.

e eficientes quando o número de estações é pequeno e fixo, e o tráfego é contínuo. Ambos são amplamente utilizados nessas circunstâncias, como para dividir a largura de banda nos enlaces usados como troncos telefônicos. No entanto, quando o número de estações é grande e variável, ou quando o tráfego ocorre em rajadas – o caso comum nas redes de computadores – FDM e TDM não são boas opções.

Foram criados diversos algoritmos dinâmicos de alocação de canal. O protocolo ALOHA, com e sem segmentação (o ALOHA original ou o slotted ALOHA), é usado em muitas derivações nos sistemas reais, por exemplo, nas redes DOCSIS. Como uma melhoria quando o estado do canal pode ser detectado, as estações podem evitar iniciar uma transmissão enquanto outra estação está transmitindo. Essa técnica, a detecção de portadora, levou a uma série de protocolos CSMA para LANs e MANs. Essa é a base para a Ethernet clássica e para as redes 802.11.

Uma classe de protocolos que elimina por completo a disputa, ou pelo menos a reduz consideravelmente, é bastante conhecida há anos. O protocolo bit-map, topologias em anéis e a contagem regressiva binária eliminam totalmente a disputa. O protocolo tree-walk reduz a disputa dividindo de forma dinâmica as estações em dois grupos com tamanhos diferentes, e permitindo a disputa apenas dentro de um grupo; o ideal é que esse grupo seja escolhido de tal forma que apenas uma estação que esteja pronta para transmitir tenha permissão para fazê-lo. As versões modernas dos protocolos MAC, incluindo DOCSIS e Bluetooth, tomam medidas explícitas para evitar a disputa, atribuindo intervalos de transmissão aos transmissores.

As LANs sem fio têm os problemas adicionais de dificuldade em detectar as transmissões que colidem, e as regiões de cobertura das estações podem ser diferentes. Na LAN sem fio dominante, IEEE 802.11, as estações usam CSMA/CA para aliviar o problema de deixar pequenas lacunas para impedir colisões. As estações também podem usar o protocolo RTS/CTS para combater terminais ocultos que surgem em decorrência do segundo problema, embora o overhead do RTS/CTS seja muito alto na prática, devido ao problema do terminal exposto, que quase nunca é usado, especialmente em ambientes densos.

Em contrapartida, muitos clientes agora utilizam mecanismos para realizar a seleção de canal a fim de evitar disputa. O IEEE 802.11 normalmente é usado para conectar notebooks e outros dispositivos a PAs wireless, mas também pode ser usado entre dispositivos. Qualquer uma das várias camadas físicas pode ser usada, incluindo o FDM multicanal com e sem várias antenas, e o espectro de dispersão. As versões modernas do 802.11 incluem recursos de segurança na camada de enlace, incluindo o suporte para autenticação, bem como a codificação avançada para dar suporte à transmissão MIMO.

A Ethernet é a forma dominante de LAN com fio. A Ethernet clássica usava CSMA/CD para alocação de canal em um cabo amarelo do tamanho de uma mangueira de jardim, esticado de uma máquina para outra. A arquitetura mudou quando as velocidades passaram de 10 Mbps para 10 Gbps, e continuam subindo. Agora, enlaces ponto a ponto, como o par trançado, são conectados a hubs e switches. Com os switches modernos e enlaces full-duplex, não existe disputa nos enlaces e o switch pode encaminhar os quadros em paralelo entre diferentes portas.

Com prédios repletos de LANs, é preciso que haja uma maneira de interconectar todas elas. As bridges plug-and-play são usadas para essa finalidade, sendo construídas com um algoritmo de aprendizado e um algoritmo spanning tree. Como essa funcionalidade está embutida nos switches modernos, os termos "bridge" e "switch" são usados para indicar a mesma coisa. Para ajudar no gerenciamento de LANs com bridges, as VLANs permitem que a topologia física seja dividida em diferentes topologias lógicas. O padrão de VLAN, o IEEE 802.1Q, introduz um novo formato de quadros Ethernet.

PROBLEMAS

1. Para resolver este problema, use uma fórmula deste capítulo, mas primeiro a enuncie. Os quadros chegam aleatoriamente a um canal de 100 Mbps para transmissão. Se estiver ocupado quando um quadro chegar, o canal aguardará sua vez em uma fila. O comprimento do quadro está distribuído exponencialmente com uma média de 10.000 bits/quadro. Para cada uma das taxas de chegada de quadros a seguir, determine o atraso médio experimentado pelo quadro, incluindo tanto o tempo de enfileiramento quanto o tempo de transmissão.
   (a) 90 quadros/s.
   (b) 900 quadros/s.
   (c) 9.000 quadros/s.
2. Um grupo de N estações compartilha um canal ALOHA original de 56 kbps. Cada estação transmite em média um quadro de 1.000 bits a cada 100 s, mesmo que o anterior ainda não tenha sido enviado (as estações podem, por exemplo, armazenar os quadros enviados em um buffer). Qual é o valor máximo de N?
3. Dez mil estações de reserva de uma companhia aérea estão competindo pelo uso de um único canal slotted ALOHA. Em média, a estação faz 18 solicitações/hora. Um slot ocupa 125 μs. Qual é a carga total aproximada do canal?
4. As medições de um canal slotted ALOHA com um número infinito de usuários mostram que 10% dos slots estão ociosos.
   (a) Qual é a carga do canal, G?
   (b) Qual é o throughput?
   (c) O canal está subcarregado ou sobrecarregado?
5. A Figura 4.4 ilustra que o throughput máximo varia do ALOHA original (mais baixo) até o CSMA 0,01-persistente (mais alto). Para alcançar o throughput máximo, um

protocolo precisa fazer algumas escolhas, por exemplo, dar suporte ao hardware extra ou aumentar o tempo de espera. Para os protocolos representados nessa figura, explique qual escolha cada protocolo faz para alcançar o throughput mais alto.

6. Qual é o tamanho de um slot de disputa em CSMA/CD para (a) um cabo twin de 2 km (a velocidade de propagação do sinal é 82% da velocidade de propagação do sinal no vácuo)? E (b) um cabo de fibra óptica multimodo de 40 km (a velocidade de propagação é 65% da velocidade de propagação do sinal no vácuo)?
7. Quanto tempo uma estação s terá de esperar, na pior das hipóteses, antes de poder começar a transmitir seu quadro sobre uma LAN que use o protocolo bit-map básico?
8. No protocolo de contagem regressiva binária, explique como uma estação com número mais baixo pode ser impedida de enviar um pacote.
9. Veja a Figura 4.10. Suponha que as estações saibam que existem quatro estações prontas: *B*, *D*, *G* e *H*. Como o protocolo adaptativo tree-walk atravessa a árvore para permitir que todas as quatro estações enviem seu quadro? Quantas colisões adicionais acontecem se a busca começar da raiz?
10. Um grupo de amigos se reúne para jogar videogames interativos com alto uso de CPU e rede. Os amigos jogam juntos usando uma rede sem fio de alta largura de banda. O sinal sem fio não pode se propagar através das paredes, mas os amigos estão todos na mesma sala. Em tal configuração, seria melhor usar o CSMA não persistente ou o protocolo token ring? Por favor, explique sua resposta.
11. Uma coleção de 2^n estações usa o protocolo adaptativo tree-walk para arbitrar o acesso a um cabo compartilhado. Em determinado instante, duas delas se aprontam. Qual é o número mínimo, máximo e médio de slots para percorrer a árvore se $2^n >> 1$?
12. As LANs sem fio que estudamos usavam protocolos como CSMA/CA e RTS/CTS no lugar de CSMA/CD. Sob quais condições, se houver alguma, seria possível usar CSMA/CD em seu lugar?
13. Seis estações, de *A* até *F*, se comunicam usando o protocolo MACA. Seria possível duas transmissões ocorrerem simultaneamente? Explique sua resposta.
14. Um prédio comercial de sete andares tem 15 escritórios adjacentes por andar. Cada escritório contém uma tomada (um soquete) para um terminal na parede frontal. Dessa forma, as tomadas formam uma grade retangular em um plano vertical, com uma distância de 4 m entre as tomadas, tanto na direção horizontal quanto na vertical. Supondo que seja viável passar um cabo linear entre qualquer par de tomadas, seja na horizontal, na vertical ou na diagonal, quantos metros de cabo seriam necessários para conectar todas as tomadas usando:
    (a) Uma configuração em estrela com um único roteador no centro?
    (b) Uma LAN 802.3 clássica?
15. Qual é a taxa baud da rede Ethernet clássica de 10 Mbps?
16. Estruture a codificação Manchester em uma Ethernet clássica para o fluxo de bits 0001110101.
17. Uma LAN CSMA/CD de 10 Mbps (não 802.3), com 1 km de extensão, tem uma velocidade de propagação de 200 m/μs. Não são permitidos repetidores nesse sistema. Os quadros de dados têm 256 bits, incluindo 32 bits de cabeçalho, checksum e outras formas de overhead. O primeiro slot de bits depois de uma transmissão bem-sucedida é reservado para o receptor capturar o canal com o objetivo de enviar um quadro de confirmação de 32 bits. Qual será a taxa de dados efetiva, excluindo o overhead, se partirmos do princípio de que não há colisões?
18. Considere a montagem de uma rede CSMA/CD operando a 1 Gbps por um cabo de 1 km sem repetidores. A velocidade do sinal no cabo é de 200.000 km/s. Qual é o tamanho mínimo do quadro?
19. Um pacote IP a ser transmitido por uma rede Ethernet tem 60 bytes de comprimento, incluindo todos os seus cabeçalhos. Se o LLC não estiver em uso, será necessário utilizar preenchimento no quadro Ethernet? Em caso afirmativo, de quantos bytes?
20. Os quadros Ethernet devem ter pelo menos 64 bytes para garantir que o transmissor ainda esteja ativo na eventualidade de ocorrer uma colisão na extremidade remota do cabo. O tamanho mínimo do quadro nas redes Fast Ethernet também é de 64 bytes, mas é capaz de transportar o mesmo número de bits com uma velocidade dez vezes maior. Como é possível manter o mesmo tamanho mínimo de quadro?
21. A especificação 1000Base-SX afirma que o clock deverá operar a 1250 MHz, embora a gigabit Ethernet só deva oferecer uma taxa de dados máxima de 1 Gbps. Essa velocidade mais alta serve para oferecer uma margem de segurança extra? Se não, o que acontece nesse caso?
22. Quantos quadros por segundo a gigabit Ethernet pode manipular? Pense cuidadosamente e leve em conta todos os casos relevantes. *Dica*: o fato de ela ser uma gigabit Ethernet é importante.
23. Identifique duas redes que permitam que os quadros sejam reunidos em sequência. Por que é importante haver essa característica?
24. Na Figura 4.27 são mostradas quatro estações, *A*, *B*, *C* e *D*. Qual das duas últimas estações você acha que está mais próxima de *A*, e por quê?
25. Dê um exemplo para mostrar que o RTC/CTS no protocolo 802.11 é um pouco diferente daquele do protocolo MACA.
26. Veja a Figura 4.33(b). Imagine que todas as estações, bridges e hubs mostrados na figura sejam estações sem fio, e que os enlaces indicam que duas estações estão dentro do alcance uma da outra. Se *B2* estiver transmitindo para *D* quando *B*1 quiser transmitir para *A* e *H*1 quiser transmitir para *F*, quais pares de estações são terminais ocultos ou expostos?
27. Uma LAN sem fio com um PA tem dez estações clientes. Quatro delas têm taxas de dados de 6 Mbps, quatro têm taxas de dados de 18 Mbps e as duas últimas têm taxas de dados de 54 Mbps. Qual é a taxa de dados experimentada por cada estação quando todas as dez estações estão transmitindo dados juntas e
    (a) TXOP não é usada?
    (b) TXOP é usada?
28. Suponha que uma LAN 802.11b de 11 Mbps esteja transmitindo quadros de 64 bytes em sequência por um canal de rádio com uma taxa de erros de bits igual a 10^{-7}. Quantos quadros por segundo serão danificados em média?
29. Dois dispositivos conectados à mesma rede 802.11 estão baixando um arquivo grande da Internet. Explique como um dispositivo poderia obter uma taxa de dados mais alta que o outro (ab)usando um mecanismo do 802.11 voltado para oferecer qualidade de serviço.
30. A Figura 4.28 mostra diferentes tempos de espera na 802.11 para quadros com diferentes prioridades. Essa técnica evita que o tráfego de alta prioridade, como os quadros transportando dados em tempo real, fiquem presos pelo tráfego comum. Cite uma desvantagem dessa técnica.
31. Apresente duas razões pelas quais as redes poderiam usar um código de correção de erros em vez de detecção de erros e retransmissão.
32. Por que soluções como PCF (Point Coordination Function) são mais adequadas para as versões do 802.11 que operam em frequências mais altas?
33. Uma desvantagem dos perfis do Bluetooth é que eles aumentam bastante a complexidade do protocolo. Qual é a vantagem desses perfis do ponto de vista das aplicações?
34. Imagine uma rede onde as estações se comunicam usando raios laser, semelhante à montagem que aparece na Figura 2.11. Explique as semelhanças (e as diferenças) entre essa montagem e as redes Ethernet e 802.11, e também como isso afetaria o projeto dessa camada de enlace de dados e os protocolos MAC.
35. Na Figura 4.30, observamos que um dispositivo Bluetooth pode estar em duas piconets ao mesmo tempo. Existe alguma razão pela qual um dispositivo não possa ser o mestre em ambas as piconets ao mesmo tempo?
36. Qual é o tamanho máximo do campo de dados para um quadro Bluetooth de 3 slots na taxa básica? Explique sua reposta.
37. O Bluetooth admite dois tipos de enlaces entre um mestre e um escravo. Quais são eles e para que é usado cada um?
38. Mencionamos que a eficiência de um quadro de 1 slot com codificação de repetição é de cerca de 13% na taxa de dados básica. Qual será a eficiência se, em vez disso, for usado um quadro de 5 slots com codificação de repetição na taxa de dados básica?
39. Os quadros de baliza na variante de espectro de dispersão de salto de frequência do 802.11 contêm o tempo de parada. Você acha que os quadros de baliza análogos no Bluetooth também contêm o tempo de parada? Explique sua resposta.
40. Um switch projetado para uso com Fast Ethernet tem uma placa integrada que pode mover 10 Gbps. Na pior das hipóteses, quantos quadros/s ela pode tratar?
41. Considere a LAN estendida conectada usando as bridges *B1* e *B2* na Figura 4.33(b). Suponha que as tabelas hash nas duas bridges estejam vazias. Qual será a tabela hash de *B*2 após a seguinte sequência de transmissões de dados:
    (a) *B* transmite um quadro para *E*.
    (b) *F* transmite um quadro para *A*.
    (c) *A* transmite um quadro para *B*.
    (d) *G* transmite um quadro para *E*.
    (e) *D* transmite um quadro para *C*.
    (f) *C* transmite um quadro para *A*.

    Suponha que cada quadro seja transmitido após o quadro anterior ter sido recebido.
42. Considere a LAN estendida conectada usando as bridges *B1* e *B2* na Figura 4.33(b). Suponha que as tabelas hash nas duas bridges estejam vazias. Qual destas transmissões de dados leva a um broadcast:
    (a) *A* transmite um quadro para *C*.
    (b) *B* transmite um quadro para *E*.
    (c) *C* transmite um quadro para *B*.
    (d) *G* transmite um quadro para *C*.
    (e) *E* transmite um quadro para *F*.
    (f) *D* transmite um quadro para *C*.

    Suponha que cada quadro seja transmitido após o quadro anterior ter sido recebido.
43. Considere a LAN estendida conectada usando as bridges *B1* e *B2* na Figura 4.33(b). Suponha que as tabelas hash nas duas bridges estejam vazias. Liste todas as portas em que um pacote será encaminhado para a seguinte sequência de transmissões de dados:
    (a) *A* transmite um quadro para *C*.
    (b) *E* transmite um quadro para *F*.
    (c) *F* transmite um quadro para *E*.
    (d) *G* transmite um quadro para *E*.
    (e) *D* transmite um quadro para *A*.
    (f) *B* transmite um quadro para *F*.
44. Veja a Figura 4.36. Imagine que uma bridge adicional, *B*0, esteja conectada às bridges *B*4 e *B*5. Desenhe a nova spanning tree para essa topologia.
45. Considere a rede da Figura 4.39. Se uma máquina conectada à bridge *B*1 de repente se tornasse branca, seriam necessárias mudanças nos rótulos? Neste caso, quais?
46. Considere uma LAN Ethernet com sete bridges. A bridge 0 está conectada a 1 e 2. As bridges 3, 4, 5 e 6 estão conectadas a ambas 1 e 2. Suponha que a maioria dos quadros seja endereçada a estações conectadas à bridge 2. Primeiro esboce a spanning tree construída pelo protocolo Ethernet e, em seguida, esboce uma spanning tree alternativa que reduza a latência média do quadro.
47. Considere duas redes Ethernet. Na rede (a), as estações estão conectadas a um hub por meio de cabos full-duplex. Na rede (b), as estações estão conectadas a um switch usando cabos half-duplex. Para cada uma dessas redes, por que o CSMA/CD (não) é necessário?
48. Os switches store-and-forward levam vantagem sobre os switches cut-through no que se refere a quadros danificados. Explique qual é essa vantagem.
49. Mencionamos, na Seção 4.8.3, que algumas bridges podem nem sequer estar presentes na spanning tree. Mostre um cenário no qual uma bridge pode não estar presente na spanning tree.

50. Para fazer as VLANs funcionarem, são necessárias tabelas de configuração nas bridges. E se as VLANs da Figura 4.39 usarem hubs em vez de switches? Os hubs também necessitam de tabelas de configuração? Por quê?

51. Na Figura 4.40, o switch no domínio final da tecnologia antiga do lado direito é um switch que reconhece VLANs. Seria possível usar ali um switch da tecnologia antiga? Nesse caso, como isso funcionaria? Se não, por que não?

52. Capture traços de mensagens enviados pelo seu próprio computador usando o modo promíscuo várias vezes por alguns minutos. Crie um simulador para um único canal de comunicação e implemente os protocolos CSMA/CD. Avalie a eficiência desses protocolos usando seus próprios traços para representar diferentes estações disputando pelo canal. Discuta as representações desses traços como cargas de trabalho da camada de enlace.

53. Escreva um programa que simule o comportamento do protocolo CSMA/CD sobre Ethernet quando existirem N estações prontas para transmitir enquanto um quadro está sendo transmitido. Seu programa deve informar os momentos em que cada estação inicia com êxito a transmissão de seu quadro. Suponha que ocorra um pulso de clock em cada período de slot (51,2 μs) e que uma sequência de detecção de colisão e transmissão de interferência demore um período de slot. Todos os quadros têm o comprimento máximo permitido.

54. Baixe o programa wireshark em *www.wireshark.org*. Esse é um programa de código aberto gratuito para monitorar redes e relatar o que está acontecendo lá. Aprenda sobre ele assistindo a um dos muitos tutoriais no YouTube. Existem muitas páginas da Web discutindo experimentos que você pode fazer com ele. Essa é uma boa maneira de ter uma ideia prática daquilo que acontece em uma rede.