REDES DE COMPUTADORES

ANDREW S. TANENBAUM

SOLUÇÕES DOS PROBLEMAS

TRADUÇÃO DA QUARTA EDIÇÃO

TRADUÇÃO
VANDENBERG D. DE SOUZA
ANALISTA DE SISTEMAS E TRADUTOR

REVISÃO TÉCNICA
EDGAR JAMHOUR
PROFESSOR DE REDES DE COMPUTADORES
PUC-PR – PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

SOLUÇÕES DOS PROBLEMAS DO CAPÍTULO 1

1. O cão pode transportar 21 gigabytes, ou 168 gigabits. A velocidade de 18 km/h é igual a 0,005 km/s. O tempo para percorrer a distância x km é $x/0{,}005 = 200x$ segundos, o que significa uma taxa de dados de $168/200x$ Gbps ou $840/x$ Mbps. Para $x < 5{,}6$ km, o cão tem uma taxa mais alta que a linha de comunicação.

2. O modelo de LAN pode ser ampliado de forma incremental. Se a LAN é apenas um longo cabo, ela não pode ser desativada por uma falha isolada (se os servidores forem replicados). Provavelmente ela terá um custo mais baixo. Esse modelo oferece maior capacidade de computação e melhores interfaces interativas.

3. Um link de fibra transcontinental pode ter muitos gigabits/s de largura de banda, mas a latência também será alta devido à velocidade de propagação da luz por milhares de quilômetros. Em contraste, um modem de 56 kbps que chamar um computador no mesmo edifício terá baixa largura de banda e baixa latência.

4. É necessário um tempo de entrega uniforme para voz, e assim a quantidade de flutuação na rede é importante. Isso poderia ser expresso como o desvio padrão do tempo de entrega. A existência de um pequeno retardo mas com grande variabilidade na realidade é pior que um retardo um pouco mais longo com baixa variabilidade.

5. Não. A velocidade de propagação é 200.000 km/s ou 200 metros/μs. Em 10 μs, o sinal percorre 2 km. Desse modo, cada switch adiciona o equivalente a 2 km de cabo extra. Se o cliente e o servidor estiverem separados por 5000 km, o percurso de até mesmo 50 switches só adicionará 100 km ao caminho total, o que corresponde a apenas 2%. Portanto, o retardo de comutação não é um fator importante sob essas circunstâncias.

6. A solicitação tem de subir e descer, e a resposta também tem de subir e descer. O comprimento total do caminho percorrido é portanto 160.000 km. A velocidade da luz no ar e no vácuo é 300.000 km/s, e assim o retardo de propagação sozinho é 160.000/300.000 s ou cerca de 533 ms.

7. É óbvio que não existe apenas uma resposta correta nesse caso, mas os pontos a seguir parecem relevantes. O sistema atual tem muita inércia (cheques e saldos) incorporada a ele. Essa inércia pode servir para impedir que os sistemas legal, econômico e social sejam virados de cabeça para baixo toda vez que um partido diferente chegar ao poder. Além disso, muitas pessoas guardam opiniões fortes sobre questões sociais controvertidas, sem realmente conhecerem os fatos relevantes para o assunto. Permitir que opi-

niões mal debatidas sejam transformadas em lei pode ser algo indesejável. Os efeitos potenciais de campanhas de publicidade realizadas por grupos de interesses especiais de um tipo ou de outro também têm de ser considerados. Outra questão importante é a segurança. Muitas pessoas poderiam se preocupar com o fato de algum garoto de 14 anos invadir o sistema e falsificar os resultados.

8. Chame os roteadores de *A*, *B*, *C*, *D* e *E*. Existem dez linhas potenciais: *AB*, *AC*, *AD*, *AE*, *BC*, *BD*, *BE*, *CD*, *CE* e *DE*. Cada uma dessas linhas tem quatro possibilidades (três velocidades ou nenhuma linha). E assim, o número total de topologias é $4^{10} = 1.048.576$. A 100 ms cada, será necessário o tempo de 104.857,6 segundos, ou pouco mais de 29 horas para inspecionar todas elas.

9. O caminho médio de roteador para roteador é duas vezes o caminho médio de roteador para a raiz. Numere os níveis da árvore com a raiz tendo o número 1 e o nível mais profundo como n. O caminho desde a raiz até o nível n exige $n - 1$ hops (saltos), e 0,50 dos roteadores está nesse nível. O caminho desde a raiz até o nível n –1 tem 0,25 dos roteadores e um comprimento igual a n –2 hops. Conseqüentemente, o comprimento do caminho médio, l, é dado por:

$$l = 0{,}5 \times (n - 1) + 0{,}25 \times (n - 2) + 0{,}125 \times (n - 3) + \ldots$$

ou

$$l\sum_{i=1}^{\infty} n\,(0{,}5)^i - \sum_{i=1}^{\infty} n\,(0{,}5)^i$$

Essa expressão se reduz a $l = n - 2$. Portanto, o caminho médio de roteador a roteador é $2n - 4$.

10. Faça a distinção entre $n + 2$ eventos. Os eventos de 1 a n consistem na tentativa bem-sucedida do host correspondente de usar o canal, isto é, sem uma colisão. A probabilidade de cada um desses eventos é $p(1 - p)^{n-1}$. O evento $n + 1$ é um canal inativo, com probabilidade $(1 - p)^n$. O evento $n + 2$ é uma colisão. Tendo em vista que esses $n + 2$ eventos são exaustivos, a soma de suas probabilidades tem de ser a unidade. A probabilidade de uma colisão, que é igual à fração de slots desperdiçados, é então simplesmente: $1 - np(1 - p)^{n-1} - (1 - p)^n$.

11. Entre outras razões para a utilização de protocolos em camadas, seu emprego conduz à quebra do problema de projeto em fragmentos menores e mais manejáveis; além disso, a divisão em camadas significa que os protocolos podem ser alterados sem afetar protocolos de níveis mais altos ou mais baixos.

12. Não. No modelo de protocolos da ISO, a comunicação física só tem lugar na camada mais baixa, não em todas as camadas.

13. A comunicação orientada a conexões tem três fases. Na fase de estabelecimento, é feita uma solicitação para configurar uma conexão. Somente após essa fase ter sido concluída com sucesso, a fase de transferência de dados pode ser iniciada e os dados podem ser transportados. Em seguida, vem a fase de liberação. A comunicação sem conexões não tem essas fases. Ela simplesmente envia os dados.

14. Os fluxos de mensagens e bytes são diferentes. Em um fluxo de mensagens, a rede mantém o controle dos limites das mensagens. Em um fluxo de bytes, isso não acontece. Por exemplo, suponha que um processo grave 1.024 bytes para uma conexão, e que um pouco mais tarde grave outros 1.024 bytes. Em seguida, o receptor faz a leitura de 2.048 bytes. Com um fluxo de mensagens, o receptor obterá duas mensagens de 1.024 bytes cada. No caso de um fluxo de bytes, os limites de mensagens não são levados em consideração, e assim o receptor irá receber os 2.048 bytes como uma única unidade. O fato de terem existido originalmente duas mensagens distintas é perdido.

15. A negociação significa fazer ambos os lados concordarem sobre alguns parâmetros ou valores a serem usados durante a comunicação. O tamanho máximo do pacote é um exemplo, mas existem muitos outros.

16. O serviço mostrado é o serviço oferecido pela camada k à camada $k + 1$. Outro serviço que deve estar presente se encontra abaixo da camada k, ou seja, o serviço oferecido à camada k pela camada subjacente $k - 1$.

17. A probabilidade, Pk, de um quadro exigir exatamente k transmissões é a probabilidade das primeiras $k - 1$ tentativas falharem, p^{k-1}, vezes a probabilidade da k-ésima transmissão ser bem-sucedida, $(1 - p)$. O número médio de transmissões é então:

$$\sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} k(1-p)p^{k-1} = \frac{1}{1-p}$$

18. (a) Camada de enlace de dados. (b) Camada de rede.

19. Quadros encapsulam pacotes. Quando um pacote chega à camada de enlace de dados, todo o conjunto, cabeçalho, dados e tudo mais, é usado como campo de dados de um quadro. O pacote inteiro é inserido em um envelope (o quadro), por assim dizer (supondo-se que ele caiba no quadro).

20. Com n camadas e h bytes adicionados por camada, o número total de bytes de cabeçalho por mensagem é hn, e assim o espaço desperdiçado em cabeçalhos é hn. O tamanho total da mensagem é $M + nh$; portanto, a fração da largura de banda desperdiçada em cabeçalhos é $hn/(M + hn)$.

21. Ambos os modelos são baseados em protocolos colocados em camadas. Ambos têm camadas de rede, transporte e aplicação. Nos dois modelos, o serviço de transporte pode fornecer um fluxo de bytes fim a fim confiável. Por outro lado, eles diferem em diversos aspectos. O número de camadas é diferente, o TCP/IP não tem camadas de sessão ou de apresentação, o OSI não admite interligação de redes, e o OSI tem serviço orientado a conexões e sem conexões na camada de rede.

22. O TCP é orientado a conexões, enquanto o UDP é um serviço sem conexões.

23. Os dois nós do canto superior direito podem ser desconectados do restante por três bombas que derrubam os três nós aos quais eles estão conectados. O sistema pode resistir à perda de dois nós quaisquer.

24. A duplicação a cada 18 meses significa um ganho de quatro vezes em 3 anos. Em 9 anos, o ganho é então 4^3, ou 64, levando a 6,4 bilhões de hosts. Minha opinião pessoal é que esse número é muito conservador, pois provavelmente nessa época todo televisor do mundo e talvez bilhões de outros aparelhos eletrodomésticos estarão em LANs domésticas conectadas à Internet. O usuário médio no mundo desenvolvido talvez tenha então dezenas de hosts da Internet.

25. Se a rede tende a perder pacotes, é melhor confirmar cada um separadamente, de modo que os pacotes perdidos possam ser retransmitidos. Por outro lado, se a rede é altamente confiável, o envio de uma única confirmação no fim da transferência inteira poupa largura de banda no caso normal (mas exige que o arquivo inteiro seja retransmitido até mesmo se um único pacote se perder).

26. Células pequenas de tamanho fixo podem ser roteadas por switches com rapidez e completamente em hardware. Células de tamanho fixo e pequeno também tornam mais fácil a criação de hardware capaz de tratar muitas células em paralelo. Além disso, elas não bloqueiam as linhas de transmissão por um tempo muito longo, facilitando o oferecimento de garantias de qualidade de serviço.

27. A velocidade da luz no cabo coaxial é cerca de 200.000 km/s, que corresponde a 200 metros/μs. A 10 Mbps, é necessário 0,1 μs para transmitir um bit. Portanto, o bit dura 0,1 μs e, durante esse tempo, ele se propaga por 20 metros. Desse modo, um bit tem 20 metros de comprimento.

28. A imagem tem $1.024 \times 768 \times 3$ bytes ou 2.359.296 bytes. Isso corresponde a 18.874.368 bits. A 56.000 bits/s, ela demora cerca de 337,042 segundos. A 1.000.000 bits/s, ela leva cerca de 18,874 s. A 10.000.000 bits/s, ela demora aproximadamente 1,887 segundos. A 100.000.000 bits/s, ela demora cerca de 0,189 segundo.

29. Pense no problema do terminal oculto. Imagine uma rede sem fios de cinco estações, de *A* até *E*, tal que cada estação esteja no alcance apenas de seus vizinhos imediatos. Então, *A* pode se comunicar com *B* ao mesmo tempo que *D* está se comunicando com *E*. Redes sem fios têm paralelismo potencial e, nesse aspecto, são diferentes da Ethernet.

30. Uma desvantagem é a segurança. Todo entregador que por acaso esteja no edifício pode ouvir a rede. Outra desvantagem é a confiabilidade. As redes sem fios cometem muitos erros. Um terceiro problema potencial é o tempo de duração da bateria, pois a maioria dos dispositivos sem fios tende a ser móvel.

31. Uma vantagem é que, se todos usarem o padrão, cada um poderá se comunicar com todos os outros. Outra vantagem é que o uso disseminado de qualquer padrão proporcionará economias de escala, como ocorre com os chips VLSI. Uma desvantagem é o fato de os compromissos políticos necessários para se alcançar a padronização freqüentemente levarem a padrões pobres. Outra desvantagem é que, depois que um padrão é amplamente adotado, torna-se muito difícil alterá-lo, mesmo que sejam descobertas novas técnicas ou melhores métodos. Além disso, na época em que ele for aceito, talvez esteja obsoleto.

32. É claro que existem muitos exemplos. Alguns sistemas para os quais existe padronização internacional incluem os aparelhos reprodutores de CDs e seus discos, os reprodutores de fita do tipo walkman e as fitas cassetes de áudio, as câmeras e os filmes de 35 mm, e ainda os caixas eletrônicos e os cartões de bancos. As áreas em que tal padronização internacional é carente incluem aparelhos de videocassete e fitas de vídeo (NTSC VHS nos Estados Unidos, PAL VHS em partes da Europa, SECAM VHS em outros países), telefones portáteis, luminárias e lâmpadas (voltagens diferentes em países diferentes), tomadas elétricas e plugues de aparelhos eletrodomésticos (cada país tem padrões diferentes), fotocopiadoras e papel (8,5 × 11 polegadas nos Estados Unidos, A4 em todos os outros países), porcas e parafusos (medidas inglesas *versus* métricas) etc.

SOLUÇÕES DOS PROBLEMAS DO CAPÍTULO 2

1. $a_n = \dfrac{-1}{\pi n}, b_n = 0, c = 1.$

2. Um canal sem ruído pode transportar uma quantidade arbitrariamente grande de informações, não importando com que freqüência é feita a amostragem. Basta enviar uma grande quantidade de dados por amostra. No caso do canal de 4 kHz, crie 8.000 amostras/s. Se cada amostra tem 16 bits, o canal pode enviar 128 kbps. Se cada amostra tem 1.024 bits, o canal

pode enviar 8,2 Mbps. A expressão-chave aqui é "sem ruído". Com um canal normal de 4 kHz, o limite de Shannon não permitiria isso.

3. Usando o teorema de Nyquist, podemos fazer a amostragem 12 milhões de vezes/s. Sinais do nível quatro fornecem 2 bits por amostra, resultando em uma taxa de dados total de 24 Mbps.

4. Uma relação sinal/ruído igual a 20 dB significa $S/N = 100$. Tendo em vista que $\log_2 101$ é aproximadamente igual a 6,658, o limite de Shannon é cerca de 19.975 kbps. O limite de Nyquist é de 6 Kbps. Portanto, o gargalo é o limite de Nyquist, que resulta em uma capacidade máxima de canal de 6 kbps.

5. Para enviar um sinal T1, precisamos de $H\log_2(1 + S/N) = 1{,}544 \times 10^6$ com $H = 50.000$. Isso resulta em $S/N = 2^{30} - 1$, que corresponde a cerca de 93 dB.

6. Uma estrela passiva não tem nenhum componente eletrônico. A luz de uma fibra ilumina uma série de outras. Um repetidor ativo converte o sinal óptico em um sinal elétrico para processamento posterior.

7. Use $\Delta f = c\Delta\lambda/\lambda^2$ com $\Delta\lambda = 10^{-7}$ metros e $\lambda = 10^{-6}$ metros. Isso dá uma largura de banda $(\Delta f) = 30.000$ GHz.

8. A taxa de dados é $480 \times 640 \times 24 \times 60$ bps, que é igual a 442 Mbps. Por simplicidade, vamos supor 1 bps por Hz. Da equação (2-3), obtemos $\Delta\lambda = \lambda^2 \Delta f / c$. Temos $\Delta f = 4{,}42 \times 10^8$, e assim $\Delta\lambda = 2{,}5 \times 10^{-6}$ micra. O intervalo de comprimentos de onda utilizados é muito curto.

9. O teorema de Nyquist é uma propriedade matemática e não tem nenhuma relação com a tecnologia. Ele afirma que, se você tem uma função cujo espectro de Fourier não contém nenhum seno ou co-seno acima de f, então, por amostragem da função à freqüência de $2f$, você irá captar todas as informações que existem. Desse modo, o teorema de Nyquist é verdadeiro para todos os tipos de meios de transmissão.

10. No texto, foi declarado que as larguras de banda (isto é, os intervalos de freqüência) das três bandas eram aproximadamente iguais. A partir da fórmula $\Delta f = c\Delta\lambda/\lambda^2$ fica claro que, para se obter uma constante $\Delta\lambda$, quanto maior a freqüência maior tem de ser $\Delta\lambda$. O eixo x na figura é λ; assim, quanto maior a freqüência, maior o valor $\Delta\lambda$ necessário. De fato, $\Delta\lambda$ é quadrático em λ. O fato das bandas serem aproximadamente iguais é uma propriedade acidental do tipo de silício usado.

11. Comece com $\lambda f = c$. Sabemos que c é 3×10^8 m/s. Para $\lambda = 1$ cm, obtemos 30 GHz. Para $\lambda = 5$ m, obtemos 60 MHz. Desse modo, a banda coberta é de 60 MHz a 30 GHz.

12. A 1 GHz, as ondas têm o comprimento de 30 cm. Se uma onda percorrer 15 cm mais que a outra, elas chegarão fora de fase. O fato do link ter o comprimento de 50 km é irrelevante.

13. Se o feixe estiver desviado 1 mm no fim do percurso, ele perderá o detector. Isso significa um triângulo com base 100 m e altura 0,001 m. Portanto, o ângulo é aquele cuja tangente é 0,00001. Esse ângulo mede cerca de 0,00057 grau.

14. Com 66/6 ou 11 satélites por colar, a cada 90 minutos, 11 satélites passam por uma posição diretamente vertical. Isso significa que existe um trânsito a cada 491 segundos. Desse modo, haverá um handoff a cada 8 minutos e 11 segundos, aproximadamente.

15. O satélite se movimenta de uma posição diretamente vertical em direção ao horizonte meridional, com uma excursão máxima a partir da posição vertical igual a 2ϕ. Ele leva 24 horas para ir da posição diretamente vertical até a excursão máxima e voltar.

16. O número de códigos de área era $8 \times 2 \times 10$, que é igual a 160. O número de prefixos era $8 \times 8 \times 10$, ou 640. Desse modo, o número de centrais finais (end offices) se limitou a 102.400. Esse limite não é problema.

17. Com um número telefônico de 10 dígitos, poderia haver 10^{10} números, embora muitos códigos de área sejam inválidos, como 000. Porém, um limite muito mais restrito é dado pelo número de centrais finais. Existem 22.000 centrais finais, cada uma com um máximo de 10.000 linhas. Isso nos dá no máximo 220 milhões de telefones. Simplesmente não há lugar para conectar mais telefones. Isso nunca poderia ser conseguido na prática, porque algumas centrais finais não estão cheias. Uma central final em uma pequena cidade do Wyoming talvez não tenha 10.000 clientes perto dela, e portanto essas linhas são desperdiçadas.

18. Cada telefone faz 0,5 chamada/hora, de 6 minutos cada. Desse modo, um telefone ocupa um circuito por 3 minutos/hora. Vinte telefones podem compartilhar um circuito, embora a necessidade de manter a carga próxima a 100% ($\rho = 1$ em termos de enfileiramento) implique tempos de espera muito longos. Tendo em vista que 10% das chamadas são interurbanas, são necessários 200 telefones para ocupar em tempo integral um circuito interurbano. O tronco da estação tem $1.000.000/4.000 = 250$ circuitos multiplexados sobre ele. Com 200 telefones por circuito, uma estação pode admitir $200 \times 250 = 50.000$ telefones.

19. A seção transversal de cada fio de um par trançado mede $\pi/4$ mm^2. Uma extensão de 10 km desse material, com dois fios por par, tem um volume igual a $2\pi/4 \times 10^{-2}$ m^3. Esse volume é cerca de 15.708 cm^3. Com uma mas-

sa específica igual a 9,0, cada loop local tem massa igual a 141 kg. Portanto, a companhia telefônica possui $1{,}4 \times 10^9$ kg de cobre. A 3 dólares por quilograma, o cobre vale aproximadamente 4,2 bilhões de dólares.

20. Como uma única linha de estrada de ferro, ele é half-duplex. O óleo pode fluir em qualquer sentido, mas não em ambos os sentidos ao mesmo tempo.

21. Normalmente, os bits são enviados pela linha sem qualquer esquema de correção de erros na camada física. A presença de uma CPU em cada modem torna possível incluir um código de correção de erros na camada 1 para reduzir bastante a taxa de erros efetiva vista pela camada 2. O tratamento de erros pelos modems pode ser totalmente transparente para a camada 2. Muitos modems atuais incluem correção de erros.

22. Existem quatro valores válidos por baud, e assim a taxa de bits é duas vezes a taxa em bauds. A 1.200 bauds, a taxa de dados é 2.400 bps.

23. O deslocamento de fase é sempre 0, mas são usadas duas amplitudes; portanto, ele utiliza modulação por amplitude direta.

24. Se todos os pontos estiverem eqüidistantes da origem, todos eles terão a mesma amplitude, e assim a modulação de amplitude não está sendo usada. A modulação de freqüência nunca é utilizada em diagramas de constelação; portanto, a codificação é de chaveamento por deslocamento de fase puro.

25. Dois, um para upstream e um para downstream. O esquema de modulação propriamente dito utiliza apenas amplitude e fase. A freqüência não é modulada.

26. Há 256 canais ao todo, menos 6 para POTS e 2 para controle, restando 248 para dados. Se ¾ desses canais forem para downstream, isso dará 186 canais para downstream. A modulação ADSL é feita em 4.000 bauds; assim, com QAM-64 (6 bits/baud), teremos 24.000 bps em cada um dos 186 canais. A largura de banda total será então 4,464 Mbps downstream.

27. Uma página da Web de 5 KB tem 40.000 bits. O tempo de download sobre o canal de 36 Mbps é 1,1 ms. Se o retardo de enfileiramento também for de 1,1 ms, o tempo total será 2,2 ms. Sobre a ADSL não existe nenhum retardo de enfileiramento, e assim o tempo de download a 1 Mbps é 40 ms. A 56 kbps, ele é igual a 714 ms.

28. Existem dez sinais de 4.000 Hz. Precisamos de nove bandas de proteção para evitar qualquer interferência. A largura de banda mínima exigida é $4.000 \times 10 + 400 \times 9 = 43.600$ Hz.

29. Um tempo de amostragem de 125 μs corresponde a 8.000 amostras por segundo. De acordo com o teorema de Nyquist, essa é a freqüência de amostragem necessária para captar todas as informações em um canal de 4 kHz, como o de um canal telefônico. (Na realidade, a largura de banda nominal é um pouco menor, mas o corte não é nítido.)

30. Os usuários finais obtêm 7 × 24 = 168 dos 193 bits em um quadro. O overhead é portanto de 25/193 = 13%.

31. Em ambos os casos, são possíveis 8.000 amostras/s. Com a codificação dibit, são enviados dois bits por amostra. Com T1, são enviados 7 bits por período. As respectivas taxas de dados são 16 kbps e 56 kbps.

32. Dez quadros. A probabilidade de algum padrão aleatório ser 0101010101 (em um canal digital) é 1/1.024.

33. Um codificador aceita um sinal analógico arbitrário e gera um sinal digital a partir dele. Um demodulador aceita apenas uma onda senoidal modulada e gera um sinal digital.

34. (a) 64 kbps. (b) 32 kbps. (c) 8 kbps.

35. O sinal deve ir de 0 até A em um quarto de onda – isto é, em um tempo igual a $T/4$. Para controlar o sinal, devemos ter 8 etapas no quarto de onda, ou 32 amostras por onda completa. O tempo por amostra é $1/x$, e assim o período total deve ser longo o suficiente para conter 32 amostras – isto é, $T > 32/x$ ou $f_{\max} = x/32$.

36. Uma taxa de flutuação de 10^{-9} significa 1 segundo em 10^9 segundos, ou 1 nanossegundo por segundo. À velocidade OC-1, digamos 50 Mbps para simplificar, um bit perdura por 20 nanossegundos. Isso significa que demora apenas 20 segundos para a flutuação do clock alcançar um bit. Em conseqüência disso, os clocks devem ser continuamente sincronizados para impedir que eles fiquem afastados demais. Com certeza a cada 10 segundos, de preferência com freqüência muito maior.

37. Das 90 colunas, 86 estão disponíveis para dados do usuário em OC-1. Desse modo, a capacidade de usuário é 86 × 9 = 774 bytes/quadro. Com 8 bits/byte, 8.000 quadros/s e 3 portadoras OC-1 multiplexadas em conjunto, a capacidade total de usuário é 3 × 774 × 8 × 8.000, ou 148.608 Mbps.

38. O VT1.5 pode acomodar 8.000 quadros/s × 3 colunas × 9 linhas × 8 bits = 1,728 Mbps. Ele pode ser usado para acomodar DS-1. O VT2 pode acomodar 8.000 quadros/s × 4 colunas × 9 linhas × 8 bits = 2,304 Mbps. Ele pode ser usado para acomodar o serviço europeu CEPT-1. O VT6 pode acomodar 8.000 quadros/s × 12 colunas × 9 linhas × 8 bits = 6,912 Mbps. É possível utilizá-lo para acomodar o serviço DS-2.

39. A comutação de mensagens envia unidades de dados que podem ser arbitrariamente longas. A comutação de pacotes tem um tamanho máximo de pacote. Qualquer mensagem mais longa que esse tamanho máximo é dividida em vários pacotes.

40. Os quadros OC-12c têm $12 \times 90 = 1.080$ colunas de 9 linhas. Dessas, $12 \times 3 = 36$ colunas são ocupadas pelo overhead de linha e seção. Isso deixa uma SPE de 1.044 colunas. Uma coluna SPE é ocupada por overhead de caminho, restando 1.043 colunas para dados do usuário. Tendo em vista que cada coluna contém 9 bytes de 8 bits, um quadro de OC-12c contém 75.096 bits de dados do usuário. Com 8.000 quadros/s, a taxa de dados do usuário é 600,768 Mbps.

41. As três redes têm as seguintes propriedades:

Estrela: Melhor caso = 2, caso médio = 2, pior caso = 2

Anel: Melhor caso = 1, caso médio = $n/4$, pior caso = $n/2$

Interconexão total: Melhor caso = 1, caso médio = 1, pior caso = 1

42. Com a comutação de circuitos, em $t = s$, o circuito é configurado; em $t = s + x/b$, o último bit é enviado; em $t = s + x/b + kd$, a mensagem chega. Com a comutação de pacotes, o último bit é enviado em $t = x/b$. Para obter o destino final, o último pacote deve ser retransmitido $k - 1$ vezes pelos roteadores intermediários, cada retransmissão demorando p/b segundos; assim, o retardo total é $x/b + (k - 1)p/b + kd$. A comutação de pacotes é mais rápida se $s > (k - 1)p/b$.

43. O número total de pacotes necessários é x/p, e assim o tráfego total de dados + cabeçalho é $(p + h)x/p$ bits. A origem exige $(p + h)x/pb$ segundos para transmitir esses bits. As retransmissões do último pacote pelos roteadores intermediários demora um tempo total de $(k - 1)(p + h)/b$ segundos. Acrescentando o tempo para a origem enviar todos os bits, mais o tempo para os roteadores transportarem o último pacote até o destino, limpando assim o pipeline, obtemos um tempo total de $(p + h)x/pb + (p + h)(k - 1)/b$ segundos. Minimizando essa quantidade em relação a p, encontramos $p = \sqrt{hx / (k - 1)}$.

44. Cada célula tem seis vizinhas. Se a célula central utilizar o grupo de freqüências *A*, suas seis vizinhas poderão usar *B*, *C*, *B*, *C*, *B* e *C*, respectivamente. Em outras palavras, são necessárias apenas três células exclusivas. Conseqüentemente, cada célula pode ter 280 freqüências.

45. Primeiro, o desenvolvimento inicial simplesmente colocava células em regiões em que havia alta densidade de população humana ou de veículos. Uma vez posicionadas, o operador com freqüência não queria ter o traba-

lho de movê-las. Em segundo lugar, em geral as antenas são colocadas em edifícios ou montanhas. Dependendo da posição exata de tais estruturas, a área coberta por uma célula pode ser irregular devido a obstáculos próximos ao transmissor. Em terceiro, algumas comunidades ou donos de propriedades não permitem a montagem de uma torre no local em que está o centro de uma célula. Em tais casos, antenas direcionais são colocadas em uma posição que não corresponde ao centro da célula.

46. Se considerarmos que cada microcélula é um círculo com 100 m de diâmetro, então cada célula terá uma área de 2.500π. Se tomarmos a área de San Francisco, $1{,}2 \times 10^8$ m^2, e a dividirmos pela área de uma microcélula, obteremos 15.279 microcélulas. É claro que é impossível preencher o plano com círculos lado a lado (e San Francisco é decididamente tridimensional), mas com 20.000 microcélulas talvez pudéssemos fazer o trabalho.

47. As freqüências não podem ser reutilizadas em células adjacentes; assim, quando um usuário se desloca de uma célula para outra, uma nova freqüência deve ser alocada para a chamada. Se um usuário se mover para uma célula cujas freqüências estão todas em uso atualmente, a chamada do usuário terá de ser encerrada.

48. Ele não é causado diretamente pela necessidade de compatibilidade com versões anteriores. O canal de 30 kHz era de fato um requisito, mas os projetistas do D-AMPS não eram obrigados a colocar três usuários nele. Eles podiam ter colocado dois usuários em cada canal, aumentando a carga útil antes da correção de erros de $260 \times 50 = 13$ kbps para $260 \times 75 = 19{,}5$ kbps. Desse modo, a perda de qualidade foi um compromisso intencional para colocar mais usuários por célula e, portanto, perde o sentido com células maiores.

49. O D-AMPS utiliza 832 canais (em cada sentido) com três usuários compartilhando um único canal. Isso permite ao D-AMPS admitir até 2.496 usuários por célula simultaneamente. O GSM utiliza 124 canais com oito usuários compartilhando um único canal. Isso permite que ao GSM dar suporte a até 992 usuários ao mesmo tempo. Ambos os sistemas utilizam quase a mesma porção do espectro (25 MHz em cada sentido). O D-AMPS utiliza $30 \text{ kHz} \times 892 = 26{,}76$ MHz. O GSM usa $200 \text{ kHz} \times 124 = 24{,}80$ MHz. A diferença pode ser atribuída principalmente à melhor qualidade de voz oferecida pelo GSM (13 Kbps por usuário) em relação ao D-AMPS (8 Kbps por usuário).

50. O resultado é obtido pela negação de cada um dos valores *A*, *B* e *C*, e depois somando-se as três seqüências de chips. Como outra alternativa, as três seqüências podem ser somadas e depois negadas. O resultado é (+3 +1 +1 –1 –3 –1 –1 +1).

51. Por definição:

$$\mathbf{S} \bullet \mathbf{T} \equiv \frac{1}{m} \sum_{i=1}^{m} S_i T_i$$

Se T tende a 0 bit em vez de 1 bit, sua seqüência de chip é negada, com o i-ésimo elemento transformando-se em $-T_i$. Desse modo:

$$\mathbf{S} \bullet \mathbf{T} \equiv \frac{1}{m} \sum_{i=1}^{m} S_i (-T_i) = -\frac{1}{m} \sum_{i=1}^{m} S_i T_i = 0$$

52. Quando dois elementos coincidem, seu produto é + 1. Quando eles não coincidem, seu produto é –1. Para obter a soma 0, deve haver uma quantidade de coincidências igual ao número de não coincidências. Desse modo, duas seqüências de chips são ortogonais se exatamente metade dos elementos correspondentes coincide e exatamente metade não coincide.

53. Basta calcular os quatro produtos internos normalizados:

$$(-1\ +1\ -3\ +1\ -1\ -1\ +3\ +1) \bullet (-1\ -1\ -1\ +1\ +1\ -1\ +1\ +1)/8 = 1$$
$$(-1\ +1\ -3\ +1\ -1\ -1\ +3\ +1) \bullet (-1\ -1\ +1\ -1\ +1\ +1\ +1\ -1)/8 = -1$$
$$(-1\ +1\ -3\ +1\ -1\ -1\ +3\ +1) \bullet (-1\ +1\ -1\ +1\ +1\ +1\ -1\ -1)/8 = 0$$
$$(-1\ +1\ -3\ +1\ -1\ -1\ +3\ +1) \bullet (-1\ +1\ -1\ -1\ -1\ -1\ +1\ -1)/8 = 1$$

O resultado é que *A* e *D* enviaram bits 1, *B* enviou um bit 0 e *C* se manteve em silêncio.

54. Ignorando-se a compactação de voz, um telefone digital PCM precisa de 64 kbps. Se dividirmos 10 Gbps por 64 kbps, obtermos 156.250 casas por cabo. Os sistemas atuais têm centenas das casas por cabo.

55. Ambos. Cada um dos 100 canais recebe a atribuição de sua própria faixa de freqüência (FDM) e, em cada canal, os dois fluxos lógicos são entremeados pelo TDM. Esse exemplo é igual ao exemplo de rádio AM dado no texto, mas nenhum deles é um exemplo fantástico de TDM, porque a alternância é irregular.

56. Uma garantia de largura de banda downstream de 2 Mbps para cada casa implica no máximo 50 casas por cabo coaxial. Desse modo, a empresa de transmissão por cabo precisará dividir o cabo existente em 100 cabos coaxiais e conectar cada um deles diretamente a um nó de fibra.

57. A largura de banda upstream é 37 MHz. Usando QPSK com 2 bits/Hz, obtemos 74 Mbps upstream. Temos 200 MHz downstream. Usando-se QAM-64, isso equivale a 1.200 Mbps. Utilizando-se QAM-256, isso é igual a 1.600 Mbps.

58. Ainda que o canal downstream funcione a 27 Mbps, a interface do usuário é quase sempre Ethernet de 10 Mbps. Não existe nenhum meio de enviar bits ao computador com velocidade maior que 10 Mbps sob essas circunstâncias. Se a conexão entre o PC e o modem a cabo for Ethernet rápida, o total de 27 Mbps poderá estar disponível. Em geral, as operadoras de cabo especificam Ethernet de 10 Mbps, porque não querem que um único usuário fique com toda a largura de banda.

SOLUÇÕES DOS PROBLEMAS DO CAPÍTULO 3

1. Tendo em vista que cada quadro tem uma chance de 0,8 de chegar, a chance da mensagem inteira chegar é $0{,}8^{10}$, que é cerca de 0,107. Chame esse valor de p. O número esperado de transmissões para uma mensagem inteira é então:

$$E = \sum_{i=1}^{\infty} ip(1-p)^{i-1} = p\sum_{i=1}^{\infty} i(1-p)^{i-1}$$

Para reduzir isso, use a conhecida fórmula da soma de uma série geométrica infinita:

$$S = \sum_{1=1}^{\infty} \alpha^{i} = \frac{1}{1-\alpha}$$

Diferencie ambos o lados em relação a α para obter:

$$S' = \sum_{i=1}^{\infty} i\alpha^{i-1} = \frac{1}{(1-\alpha)^2}$$

Agora use $\alpha = 1 - p$ para obter $E = 1/p$. Desse modo, isso ocupa em média 1/0,107, ou cerca de 9,3 transmissões.

2. A solução é:

(a) 00000100 01000111 11100011 11100000 01111110

(b) 01111110 01000111 11100011 11100000 11100000 11100000 01111110 01111110

(c) 01111110 01000111 110100011 111000000 011111010 01111110

3. Após a inserção, obtemos: A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D.

4. Se você sempre pudesse contar com uma série infinita de quadros, um byte de flag poderia ser suficiente. Porém, o que aconteceria se um quadro terminasse (com um byte de flag) e não houvesse nenhum novo quadro du-

rante 15 minutos? Como o receptor saberá que o próximo byte é na realidade o início de um novo quadro e não apenas ruído na linha? O protocolo é muito mais simples com bytes de flag iniciais e finais.

5. A saída é 011110111110011111010.

6. É possível. Suponha que o texto original contenha a seqüência de bits 01111110 como dados. Depois da inserção de bits, essa seqüência será representada por 011111010. Se o segundo 0 se perder devido a um erro de transmissão, a seqüência recebida será 01111110, que o receptor vê como um fim de quadro. Em seguida, ele observa imediatamente antes do fim do quadro a soma de verificação e a confirma. Se a soma de verificação é 16 bits, existe uma chance em 2^{16} de que ela esteja acidentalmente correta, levando à aceitação de um quadro incorreto. Quanto mais longa a soma de verificação, menor a probabilidade de um erro não ser detectado, mas a probabilidade nunca é zero.

7. Se o retardo de propagação é muito longo, como no caso de uma sonda para Marte ou Vênus, a correção antecipada de erros é indicada. Além disso, ela também é apropriada em uma instalação militar na qual o receptor não quer revelar sua posição transmitindo. Se a taxa de erros for baixa o suficiente para que um código de correção de erros seja bom o bastante, ele também poderá ser mais simples. Por fim, os sistemas de tempo real não podem tolerar a espera por retransmissões.

8. Uma mudança em qualquer caractere válido não pode gerar outro caractere válido, devido à natureza dos bits de paridade. Efetuar duas mudanças em bits pares ou duas mudanças em bits ímpares resultará em outro caractere válido, e assim a distância é 2.

9. Os bits de paridade são necessários nas posições 1, 2, 4, 8 e 16, de forma que as mensagens que não se estendem além do bit 31 (incluindo os bits de paridade) se adaptam. Desse modo, cinco bits de paridade são suficientes. O padrão de bits transmitido é 01101011001110101.

10. O valor codificado é 101001001111.

11. Se numerarmos os bits da esquerda para a direita começando no bit 1, o bit 2 desse exemplo (um bit de paridade) será incorreto. O valor de 12 bits transmitido (após a codificação de Hamming) foi 0xA4F. O valor de dados de 8 bits original foi 0xAF.

12. Um único erro tornará erradas ambas as verificações de paridade, horizontal e vertical. Dois erros também serão detectados com facilidade. Se eles estiverem em linhas diferentes, a paridade de linha os detectará. Se estiverem na mesma linha, a paridade de coluna irá captá-los. Três erros poderão

passar despercebidos, por exemplo, se algum bit for invertido juntamente com seus bits de paridade de linha e coluna. Nem mesmo o bit do canto irá captar isso.

13. Descreva um padrão de erro como uma matriz de n linhas por k colunas. Cada um dos bits corretos é 0 e cada um dos bits incorretos é 1. Com quatro erros por bloco, cada bloco terá exatamente quatro valores 1. Quantos desses blocos existem? Há nk maneiras de escolher onde colocar o primeiro bit 1, $nk - 1$ modos de escolher o segundo e assim por diante, até o número de blocos ser $nk(nk\text{-}1)(nk\text{-}2)(nk\text{-}3)$. Erros não detectados só ocorrem quando os quatro bits 1 estão nos vértices de um retângulo. Usando-se coordenadas cartesianas, todo bit 1 está em uma coordenada (x, y), onde $0 \le x < k$ e $0 \le y < n$. Suponha que o bit mais próximo à origem (o vértice inferior esquerdo) esteja em (p, q). O número de retângulos válidos é $(k - p - 1)(n - q - 1)$. Então, o número total de retângulos pode ser encontrado fazendo-se o somatório dessa fórmula para todos os valores p e q possíveis. A probabilidade de um erro não detectado é então o número de tais retângulos dividido pelo número de maneiras de distribuir os quatro bits:

$$\frac{\sum_{p=0}^{k-2}\sum_{q=0}^{n-2}(k-p-1)(n-1-1)}{nk(nk-1)(nk-2)(nk-3)}$$

14. O resto é $x^2 + x + 1$.

15. O quadro é 10011101. O gerador é 1001. A mensagem depois de acrescentar três zeros é 10011101000. O resto da divisão de 10011101000 por 1001 é 100. Assim, o string de bits real transmitido é 10011101100. O fluxo de bits recebido com um erro no terceiro bit a partir da esquerda é 10111101100. A divisão desse valor por 1001 produz o resto 100, que é diferente de zero. Desse modo, o receptor detecta o erro e pode solicitar uma retransmissão.

16. O CRC é calculado durante a transmissão e acrescentado ao fluxo de saída tão logo o último bit sai para o fio. Se o CRC estivesse no cabeçalho, seria necessário fazer uma passagem sobre o quadro para calcular o CRC antes da transmissão. Isso exigiria que cada byte fosse tratado duas vezes – uma vez para o cálculo da soma de verificação e uma para transmissão. O uso do final (trailer) reduz o trabalho à metade.

17. A eficiência será 50% quando o tempo para transmitir o quadro for igual ao retardo de propagação de ida e volta. A uma taxa de transmissão de 4 bits/ms, a transmissão de 160 bits demora 40 ms. Para tamanhos de quadros acima de 160 bits, o método de parar e esperar tem uma eficiência razoável.

18. Para operar de modo eficiente, o espaço da seqüência (na realidade, o tamanho da janela de envio) deve ser grande o bastante para permitir ao transmissor continuar transmitindo até receber a primeira confirmação. O tempo de propagação é 18 ms. À velocidade T1, que equivale a 1,536 Mbps (excluindo-se o bit 1 do cabeçalho), um quadro de 64 bytes demora 0,300 ms. Portanto, o primeiro quadro chega totalmente 18,3 ms depois de sua transmissão ter se iniciado. A confirmação demora outros 18 ms para voltar, mais um pequeno tempo (desprezível) para a confirmação chegar por completo. No total, esse tempo é de 36,3 ms. O transmissor deverá ter espaço de janela suficiente para continuar por 36,3 ms. Um quadro demora 0,3 ms, e assim serão necessários 121 quadros para preencher o canal. Será preciso usar sete números de seqüências de bits.

19. Pode acontecer. Suponha que o transmissor envie um quadro e que uma confirmação adulterada volte rapidamente. O loop principal será executado uma segunda vez e um quadro será enviado enquanto o timer ainda estiver executando.

20. Seja a janela do transmissor (S_l, S_u) e a do receptor (R_l, R_u). Seja W o tamanho da janela. As relações que devem ser válidas são:

$0 \le S_u - S_l + 1 \le W1$
$R_u - R_l + 1 = W$
$S_l \le R_l \le S_u + 1$

21. O protocolo seria incorreto. Suponha que estejam em uso números de seqüência de 3 bits. Considere o seguinte cenário:

A acaba de enviar o quadro 7.
B recebe o quadro e envia um ACK de piggyback.
A recebe o ACK e envia os quadros de 0 a 6, e todos eles são perdidos.
B chega ao tempo limite (timeout) e retransmite seu quadro atual, com o ACK 7.

Observe a situação em *A* quando chega o quadro com *r.ack* = 7. As variáveis-chave são *AckExpected* = 0, *r.ack* = 7 e *NextFrameToSend* = 7. O *between* modificado retornaria *true*, fazendo *A* imaginar que os quadros perdidos estavam sendo confirmados.

22. Sim. Ela poderia levar a um impasse. Suponha que um lote de quadros chegasse corretamente e fosse aceito. Então, o receptor avançaria sua janela. Agora, suponha que todas as confirmações se perdessem. O transmissor eventualmente chegaria ao tempo limite e enviaria o primeiro quadro de novo. O receptor enviaria um NAK. Suponha que ele se perdesse. Desse ponto em diante, o transmissor continuaria a entrar no tempo limite e a enviar um quadro que já havia sido aceito, mas o receptor simplesmente o ig-

noraria. A definição do timer auxiliar resulta, em vez disso, no envio de uma confirmação correta, o que resultaria na ressincronização.

23. Ele levaria ao impasse (deadlock), porque esse é o único lugar em que são processadas as confirmações que chegam. Sem esse código, o transmissor continuaria em timeout e nunca faria nenhum progresso.

24. Anularia o propósito de se ter NAKs, de forma que teríamos de recorrer a timeouts. Embora o desempenho se degradasse, a correção não seria afetada. Os NAKs não são essenciais.

25. Considere o seguinte cenário. *A* envia 0 a *B*. *B* o recebe e envia um ACK, mas o ACK é perdido. *A* chega ao tempo limite e repete 0, mas agora *B* espera 1, e assim envia um NAK. Se *A* simplesmente reenviasse r.ack+1, ele estaria enviando o quadro 1, que ainda não recebeu.

26. Não. O tamanho máximo da janela de recepção é 1. Suponha que fosse 2. Inicialmente, o transmissor envia quadros de 0 a 6. Todos os valores são recebidos e confirmados, mas a confirmação é perdida. O receptor agora está preparado para aceitar 7 e 0. Quando a retransmissão de 0 chegar ao receptor, ela será bufferizada e 6 será confirmado. Quando 7 chegar, 7 e 0 serão repassados ao host, levando a uma falha de protocolo.

27. Suponha que *A* enviasse a *B* um quadro e este chegasse corretamente, mas que não houvesse nenhum tráfego no sentido inverso. Depois de algum tempo, *A* chegaria ao tempo limite e retransmitiria. *B* notaria que o número de seqüência estava incorreto, pois o número de seqüência está abaixo de *FrameExpected*. Conseqüentemente, ele enviaria um NAK, que transportaria um número de confirmação. Cada quadro seria então enviado exatamente duas vezes.

28. Não. Essa implementação falha. Com *MaxSeq* = 4, obtemos *NrBufs* = 2. Os números de seqüência pares usam o buffer 0 e os números de seqüência ímpares usam o buffer 1. Esse mapeamento significa que os quadros 4 e 0 utilizam ambos o mesmo buffer. Suponha que os quadros 0 a 3 fossem recebidos e confirmados. Agora, a janela do receptor contém 4 e 0. Se 4 for perdido e 0 chegar, ele será inserido no buffer 0 e *arrived*[0] será definido como *true*. O loop no código de *FrameArrival* será executado uma vez, e uma mensagem fora de ordem será entregue ao host. Esse protocolo exige que *MaxSeq* seja ímpar para funcionar corretamente. Porém, outras implementações de protocolos de janelas deslizantes não têm todas essa propriedade.

29. Seja $t = 0$ o início da transmissão. Em $t = 1$ ms, o primeiro quadro é totalmente transmitido. Em $t = 271$ ms, o primeiro quadro chega por completo. Em $t = 272$ ms, o quadro que confirma o primeiro é completamente enviado. Em $t = 542$ ms, o quadro que conduz a confirmação chega por inteiro.

ro. Desse modo, o ciclo tem 542 ms. Ao todo, k quadros são enviados em 542 ms, o que dá uma eficiência de k/542. Conseqüentemente:

(a) $k = 1$, eficiência = 1/542 = 0,18%.
(b) $k = 7$, eficiência = 7/542 = 1,29%.
(c) $k = 4$, eficiência = 4/542 = 0,74%.

30. Com um canal de 50 kbps e oito números de seqüência de bits, o canal está sempre cheio. O número de retransmissões por quadro é cerca de 0,01. Cada quadro de boa qualidade desperdiça 40 bits de cabeçalho, mais 1% de 4.000 bits (retransmissão), mais um NAK de 40 bits uma vez a cada 100 quadros. O overhead total é 80,4 bits por 3.960 bits de dados, o que corresponde a uma fração 80,4/(3.960 + 80,4) = 1,99%.

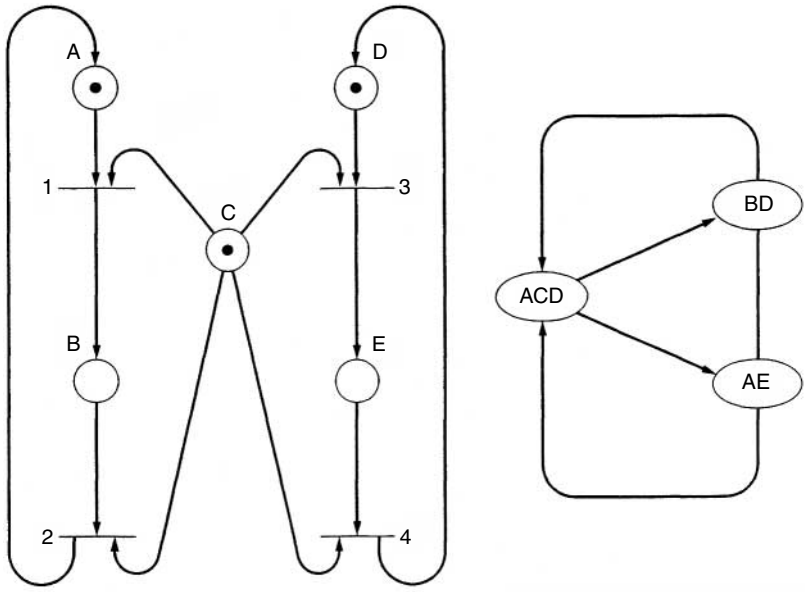
31. A transmissão começa em $t = 0$. Em t = 4.096/64.000 s = 64 ms, o último bit é enviado. Em $t = 334$ ms, o último bit chega ao satélite e o ACK muito breve é enviado. Em $t = 604$ ms, o ACK chega ao solo. Aqui, a taxa de dados é 4.096 bits em 604 ms, ou cerca de 6.781 bps. Com um tamanho de janela de 7 quadros, o tempo de transmissão é 448 ms para a janela completa, e nesse tempo o transmissor tem de parar. Em 604 ms, o primeiro ACK chega e o ciclo pode recomeçar. Nesse caso, temos $7 \times 4.096 = 28.672$ bits em 604 ms. A taxa de dados é 47.470,2 bps. A transmissão contínua só pode ocorrer se o transmissor ainda estiver enviando quando o primeiro ACK voltar, no tempo $t = 604$ ms. Em outras palavras, se o tamanho da janela for maior que 604 ms de transmissão, ele poderá funcionar a toda velocidade. Para um tamanho de janela igual a 10 ou maior, essa condição é satisfeita; assim, para qualquer janela de tamanho 10 ou maior (por exemplo, 15 ou 127), a taxa de dados é 64 kbps.

32. A velocidade de propagação no cabo é 200.000 km/s, ou 200 km/ms, e assim um cabo de 100 km será preenchido em 500 μs. Cada quadro T1 equivale a 193 bits enviados em 125 μs. Isso corresponde a quatro quadros, ou 772 bits no cabo.

33. Cada máquina tem duas variáveis importantes: *next_frame_to_send* e *frame_expected*, cada uma das quais pode assumir os valores 0 ou 1. Desse modo, cada máquina pode estar em um entre quatro estados possíveis. Uma mensagem no canal contém o número de seqüência do quadro que está sendo enviado e o número de seqüência do quadro que está sendo confirmado por ACK. Desse modo, existem quatro tipos de mensagens. O canal pode conter a mensagem 0 ou 1 em qualquer sentido. Portanto, o número de estados em que o canal pode estar é 1 com zero mensagens, 8 com uma mensagem e 16 com duas mensagens (uma mensagem em cada sentido). No total existem 1 + 8 + 16 = 25 estados possíveis do canal. Isso implica $4 \times 4 \times 25 = 400$ estados possíveis para o sistema completo.

34. A seqüência de disparo é 10, 6, 2, 8. Ela corresponde à aceitação de um quadro par, à perda da confirmação, ao timeout pelo transmissor e à regeneração da confirmação pelo receptor.

35. A rede de Petri e grafo do estado são:



O sistema modelado é de exclusão mútua. *B* e *E* são seções críticas que não podem estar ativas ao mesmo tempo, isto é, o estado *BE* não é permitido. A posição *C* representa um semáforo que pode ser ocupado por qualquer *A* ou *D*, mas não por ambos ao mesmo tempo.

36. O PPP foi claramente projetado para ser implementado em software e não em hardware, como o HDLC quase sempre é. Com uma implementação de software, funcionar inteiramente com bytes é muito mais simples que trabalhar com bits individuais. Além disso, o PPP foi criado para ser usado com modems, e os modems aceitam e transmitem dados em unidades múltiplas de 1 byte, e não de 1 bit.

37. No mínimo, cada quadro tem dois bytes de flag (sinalização), um byte de protocolo e dois bytes de total de verificação, dando um total de cinco bytes de overhead por quadro.

SOLUÇÕES DOS PROBLEMAS DO CAPÍTULO 4

1. A fórmula é a fórmula padrão para o enfileiramento de Markov, dada na Seção 4.1.1, ou seja, $T = 1(\mu C - \lambda)$. Nesse caso, $C = 10^8$ e $\mu = 10^{-4}$, e portanto $T = 1/(1.000 - lambda)$ s. Para as três taxas de chegada, obtemos (a) 0,1 ms, (b) 0,11 ms, (c) 1 ms. No caso (c), estamos operando um sistema de enfileiramento com $\rho = \lambda/\mu C = 0{,}9$, que corresponde ao retardo de 10×.

2. Com o ALOHA puro, a largura de banda utilizável é 0,184 × 56 kbps = 10,3 kbps. Cada estação requer 10 bps; assim, $N = 10.300/10 = 1.030$ estações.

3. Com o ALOHA puro, a transmissão pode começar instantaneamente. Com baixa carga, não é esperada nenhuma colisão, e assim a transmissão provavelmente será bem-sucedida. Com o slotted ALOHA, ela tem de esperar pelo próximo slot. Isso introduz um tempo de retardo igual à metade de um slot.

4. Cada terminal faz uma solicitação a cada 200 segundos, o que corresponde a uma carga total de 50 solicitações/s. Conseqüentemente, $G = 50/8.000 = 1/160$.

5. (a) Com $G = 2$, a lei de Poisson fornece uma probabilidade igual a e^{-2}.
(b) $(1 - e^{-G})^k e^{-G} = 0{,}135 \times 0{,}865^k$.
(c) O número esperado de transmissões é $e^G = 7{,}4$.

6. (a) Mais uma vez a partir da lei de Poisson, $P_0 = e^{-G}$, e assim $G = -\ln P_0 = -\ln 0{,}1 = 2{,}3$.
(b) Usando $S = Ge^{-G}$ com $G = 2{,}3$ e $e^{-G} = 0{,}1$, $S = 0{,}23$.
(c) Sempre que $G > 1$, o canal fica sobrecarregado; portanto, ele está sobrecarregado.

7. O número de transmissões é $E = e^G$. Os E eventos estão separados por $E - 1$ intervalos de quatro slots cada; assim, o retardo é $4(e^G - 1)$. O throughput é dado por $S = Ge^{-G}$. Desse modo, temos duas equações paramétricas, uma para retardo e uma para throughput, ambas em termos de G. Para cada valor de G, é possível encontrar o retardo e o throughput correspondentes, gerando um único ponto na curva.

8. (a) O pior caso é: Todas as estações querem enviar e *s* é a estação de número mais baixo. O tempo de espera é N períodos de disputa de bits + $(N - 1) \times d$ bit para transmissão de quadros. O total é $N + (N - 1)d$ tempos de bits.
(b) O pior caso é: Todas as estações têm quadros a transmitir e *s* tem o número de estação virtual mais baixo. Conseqüentemente, *s* terá sua vez de transmitir depois que as outras $N - 1$ estações tiverem transmitido um quadro cada uma, e depois de N períodos de disputa de tamanho $\log_2 N$ cada. O tempo de espera é portanto $(N - 1) \times d + N \times \log_2$ bits.

9. Quando a estação 4 envia, ela se torna 0, e 1, 2 e 3 são aumentados em 1. Quando a estação 3 envia, ela se torna 0, e 0, 1 e 2 são aumentados em 1. Finalmente, quando a estação 9 envia, ela se torna 0 e todas as outras estações são incrementadas em 1. O resultado é 9, 1, 2, 6, 4, 8, 5, 7, 0 e 3.

10. As estações 2, 3, 5, 7, 11 e 13 querem enviar. São necessários onze slots, sendo o conteúdo de cada slot:

slot 1: 2, 3, 5, 7, 11, 13
slot 2: 2, 3, 5, 7
slot 3: 2, 3
slot 4: 2
slot 5: 3
slot 6: 5, 7
slot 7: 5
slot 8: 7
slot 9: 11, 13
slot 10: 11
slot 11: 13

11. O número de slots necessários depende da distância que se deve percorrer de volta na árvore até encontrar um ancestral comum das duas estações. Se eles têm o mesmo pai (isto é, um nível de volta), o que acontece com probabilidade 2^{-n}, a demora é de $2n + 1$ slots para percorrer a árvore. Se as estações têm um avô comum, o que acontece com probabilidade $2^{-n} + 1$, o percurso na árvore demora $2n - 1$ slots etc. O pior caso é $2n + 1$ (pai comum), e o melhor caso é o de três slots (estações em metades diferentes da árvore). A média m é dada por:

$$m = \sum_{i=0}^{n-1} 2^{-(n-i)}(2n + 1 - 2i)$$

Essa expressão pode ser simplificada para:

$$m = (1 - 2^{-n})(2n + 1) - 2^{-(n-1)} \sum_{i=0}^{n-1} i2^i$$

12. Os rádios não podem receber e transmitir na mesma freqüência ao mesmo tempo, e assim o CSMA/CD não pode ser usado. Se esse problema pudesse ser resolvido (por exemplo, equipando-se cada estação com dois rádios), ainda haveria o problema de nem todas as estações estarem dentro do alcance de rádio de cada uma das outras. Somente se ambos os problemas puderem ser resolvidos, o CSMA/CD será um candidato.

13. Ambos utilizam uma combinação de FDM e TDM. Nos dois casos, estão disponíveis bandas de freqüências dedicadas (isto é, comprimentos de onda), e nos dois casos essas bandas são dotadas de slots para TDM.

14. Sim. Imagine que elas estejam em linha reta e que cada estação possa acessar apenas suas vizinhas mais próximas. Então *A* pode transmitir para *B* enquanto *E* está transmitindo para *F*.

15. (a) Numere os andares de 1 a 7. Na configuração de estrela, o roteador está no quarto andar. São necessários cabos para cada um dos $7 \times 15 - 1 = 104$ locais. O comprimento total desses cabos é:

$$4\sum_{i=1}^{7}\sum_{j=1}^{15}\sqrt{(i-4)^2 + (j-8)^2}$$

O comprimento total é aproximadamente 1.832 metros.

(b) Para 802.3, são necessários 7 cabos horizontais de 56 metros, mais um cabo vertical de 24 metros de comprimento, correspondendo ao total de 416 m.

16. A Ethernet utiliza a codificação Manchester, o que significa que ela tem dois períodos de sinal por bit enviado. A taxa de dados do padrão Ethernet é 10 Mbps, e assim a taxa de bauds é duas vezes esse valor, ou 20 megabauds.

17. O sinal é uma onda quadrada com dois valores, alto (H) e baixo (L). O padrão é LHLHLHHLHLHLLHHLLHHL.

18. Dessa vez, o padrão é HLHLHLLHHLLHLHHLHLLH.

19. O tempo de propagação de ida e volta do cabo é 10 μs. Uma transmissão completa tem seis fases:

O transmissor ocupa o cabo (10 μs).
Transmissão de dados (25,6 μs).
Retardo para o último bit chegar ao fim (5,0 μs).
O receptor ocupa o cabo (10 μs).
Confirmação enviada (3,2 μs).
Retardo para o último bit chegar ao fim (5,0 μs).

A soma desses valores é 58,8 μs. Nesse período, são enviados 224 bits de dados, o que corresponde à taxa de 3,8 Mbps.

20. Numere as tentativas de aquisição a partir de 1. A tentativa *i* é distribuída entre 2^{i-1} slots. Desse modo, a probabilidade de uma colisão na tentativa *i* é $2^{-(i-1)}$. A probabilidade de as primeiras $k - 1$ tentativas falharem, seguidas por um sucesso na rodada *k*, é:

$$p_k = (1 - 2^{(k-1)})\prod_{i=1}^{k-} 2^{-(i-1)}$$

Que pode ser simplificada para:

$$P_k = (1 - 2^{-(k-1)})\, 2^{-(k-1)(k-2)}$$

O número esperado de rodadas é então apenas $\sum kP_k$.

21. Para um cabo de 1 km, o tempo de propagação em um sentido é 5 μs, e assim $2\tau = 10$ μs. Para fazer CSMA/CD funcionar, tem de ser impossível transmitir um quadro inteiro nesse intervalo. A 1 Gbps, todos os quadros menores que 10.000 bits podem ser completamente transmitidos em um tempo abaixo de 10 μs, e portanto o quadro mínimo é de 10.000 bits ou 1.250 bytes.

22. O quadro Ethernet mínimo tem 64 bytes, incluindo ambos os endereços no cabeçalho de quadro Ethernet, o campo de tipo/comprimento e o total de verificação. Tendo em vista que os campos de cabeçalho ocupam 18 bytes e o pacote tem 60 bytes, o tamanho total do quadro é 78 bytes, que excede o mínimo de 64 bytes. Portanto, não é utilizada nenhuma inserção.

23. O comprimento máximo de cabo no Fast Ethernet é 1/10 do comprimento na Ethernet.

24. A carga útil é de 1.500 bytes mas, quando os campos de endereço de destino, endereço de origem, tipo/comprimento e total de verificação também são considerados, o total é na verdade 1.518.

25. A codificação tem apenas 80% de eficiência. Ela utiliza 10 bits de dados transmitidos para representar 8 bits de dados reais. Em um segundo, são transmitidos 1.250 megabits, o que significa 125 milhões de palavras de código. Cada palavra de código representa 8 bits de dados, e então a taxa de dados verdadeira é de fato 1.000 megabits/s.

26. O menor quadro Ethernet tem 512 bits; assim, a 1 Gbps, obtemos 1.953.125 ou quase 2 milhões de quadros/s. Porém, isso só funciona quando a rajada de quadros está operando. Sem a rajada de quadros, os quadros curtos são preenchidos por inserção até 4.096 bits e, nesse caso, o número máximo é 244.140. Para o maior quadro (12.144 bits), pode haver até 82.345 quadros/s.

27. A Ethernet de gigabit tem esse recurso, bem como o 802.16. Ele é útil para aumentar a eficiência de largura de banda (um único preâmbulo etc.), mas também quando existe um limite mais baixo sobre o tamanho dos quadros.

28. A estação *C* é a mais próxima de *A*, pois ouviu o RTS e respondeu a ele afirmando seu sinal NAV. *D* não respondeu, e portanto deve estar fora do alcance de rádio de *A*.

29. Um quadro contém 512 bits. A taxa de erros de bits é $p = 10^{-7}$. A probabilidade de todos os 512 bits sobreviverem corretamente é $(1 - p)^{512}$, que equivale a cerca de 0,9999488. A fração danificada é então cerca de 5×10^{-5}. O número de quadros/s é $11 \times 10^6/512$ ou aproximadamente 21.484. Multiplicando esses dois números, obtemos quase um quadro danificado por segundo.

30. Depende da distância em que se encontra o assinante. Se o assinante estiver perto, o QAM-64 será usado para 120 Mbps. Em distâncias médias, o QAM-16 é usado para 80 Mbps. No caso de estações distantes, o QPSK será usado para 40 Mbps.

31. O vídeo não-compactado tem uma taxa de bits constante. Cada quadro tem o mesmo número de pixels que o quadro anterior. Desse modo, é possível calcular com muita precisão a quantidade de largura de banda que será necessária e quando. Conseqüentemente, o serviço de taxa de bits constante é a melhor opção.

32. Uma razão é a necessidade de qualidade de serviço em tempo real. Se for descoberto um erro, não haverá tempo para uma retransmissão. O espetáculo tem de continuar. A correção de erros direta pode ser usada nesse caso. Outra razão é que, em linhas de qualidade muito baixa (por exemplo, canais sem fios), a taxa de erros pode ser tão alta que praticamente todos os quadros teriam de ser retransmitidos, e seria bem provável que a retransmissão também estivesse danificada. Para evitar isso, é usada a correção antecipada de erros, a fim de aumentar a fração de quadros que chegam corretamente.

33. É impossível um dispositivo ser mestre em duas piconets ao mesmo tempo. Há dois problemas. Primeiro, só estão disponíveis 3 bits de endereço no cabeçalho, enquanto até sete escravos poderiam estar em cada piconet. Desse modo, não haveria nenhum meio de endereçar de forma exclusiva cada escravo. Em segundo lugar, o código de acesso no começo do quadro é derivado da identidade do mestre. Essa é a maneira como os escravos sabem que mensagem pertence a cada piconet. Se duas piconets superpostas usassem o mesmo código de acesso, não haveria como saber qual quadro pertenceria a cada piconet. Na realidade, as duas piconets estariam fundidas em uma única piconet grande, e não em duas piconets separadas.

34. O Bluetooth utiliza FHSS, da mesma forma que o 802.11. A maior diferença é que o Bluetooth salta a uma taxa de 1.600 hops/s, bem mais rápido que o 802.11.

35. Um canal ACL é assíncrono, com os quadros chegando irregularmente à medida que os dados são produzidos. Um canal SCO é síncrono, com quadros chegando periodicamente a uma taxa bem definida.

36. Não. O tempo de parada no 802.11 não é padronizado, e assim ele tem de ser anunciado às novas estações que chegam. No Bluetooth, esse tempo é sempre 625 μs. Não há necessidade de anunciá-lo. Todos os dispositivos Bluetooth têm esse valor codificado no chip. O Bluetooth foi projetado para ser econômico, e a fixação da taxa de hops e do tempo de parada leva a um chip mais simples.

37. O primeiro quadro será encaminhado por cada ponte. Após essa transmissão, cada ponte terá uma entrada para o destino *a* com a porta apropriada em sua tabela de hash. Por exemplo, a tabela de hash de *D* terá agora uma entrada para quadros diretos destinados a *a* na LAN 2. A segunda mensagem será vista pelas pontes *B*, *D* e *A*. Essas pontes acrescentarão uma nova entrada em suas tabelas de hash para quadros destinados a *c*. Por exemplo, a tabela de hash da ponte *D* terá agora outra entrada para quadros diretos destinados a *c* na LAN 2. A terceira mensagem será vista pelas pontes *H*, *D*, *A* e *B*. Essas pontes acrescentarão uma nova entrada em suas tabelas de hash para quadros destinados a *d*. A quinta mensagem será vista pelas pontes *E*, *C*, *B*, *D* e *A*. As pontes *E* e *C* acrescentarão uma nova entrada em suas tabelas de hash para quadros destinados a *d*, enquanto as pontes *D*, *B* e *A* atualizarão as entradas de suas tabelas de hash para o destino *d*.

38. A pontes G, *I* e *J* não são usadas para encaminhar quaisquer quadros. A principal razão para termos loops em uma LAN estendida é o aumento da confiabilidade. Se qualquer ponte na árvore atual falhar, o algoritmo (dinâmico) de árvore de amplitude irá reconfigurar a árvore, formando uma nova árvore que poderá incluir uma ou mais dessas pontes que não faziam parte da árvore anterior.

39. A opção mais simples é não fazer nada de especial. Todo quadro de entrada é colocado no painel traseiro (backplane) e enviado à placa de destino, que poderia ser a placa de origem. Nesse caso, o tráfego entre placas passará pelo painel traseiro do switch. A outra opção é reconhecer esse caso e tratá-lo de modo especial, enviando o quadro diretamente sem passar pelo painel traseiro.

40. O pior caso é um fluxo infinito de quadros de 64 bytes (512 bits). Se o painel traseiro puder tratar 10^9 bps, o número de quadros que ele poderá manipular será $10^9/512$. Isso corresponde a 1.953.125 quadros/s.

41. A porta em *B1* para a LAN 3 precisaria ser rotulada novamente como GW.

42. Um switch de armazenar e encaminhar (store-and-forward) armazena cada quadro de entrada em sua totalidade, depois o examina e o encaminha. Um switch de corte (cut-through) começa a encaminhar os quadros de entrada antes que eles cheguem completamente. Assim que chega o endereço de destino, o encaminhamento pode começar.

43. Os switches de armazenar e encaminhar armazenam quadros inteiros antes de transmiti-los. Depois que um quadro chega, o total de verificação pode ser verificado. Se o quadro estiver danificado, ele será imediatamente descartado. No caso do corte, quadros danificados não podem ser descartados pelo switch porque, no momento em que o erro for detectado, o quadro já terá ido. Tentar lidar com o problema é como trancar a porta da cocheira depois que o cavalo escapou.

44. Não. Os hubs simplesmente estabelecem conexões elétricas entre todas as linhas de entrada. Não existe nada para configurar. Nenhum roteamento é feito em um hub. Todo quadro que entra no hub sai dele por todas as outras linhas.

45. Funcionaria. Todos os quadros que entrassem no domínio do núcleo seriam quadros antigos; assim, caberia ao primeiro switch do núcleo a tarefa de identificá-los. Isso poderia ser feito com a utilização de endereços MAC ou endereços IP. De modo semelhante, no caminho de saída, esse switch teria de desmarcar os quadros de saída.

SOLUÇÕES DOS PROBLEMAS DO CAPÍTULO 5

1. Transferência de arquivos, login remoto e vídeo por demanda necessitam de um serviço orientado a conexões. Por outro lado, a verificação de cartões de crédito e outros terminais de pontos de venda, transferência eletrônica de fundos e muitas formas de acesso a bancos de dados remotos são inerentemente sem conexões, com uma consulta indo em um sentido e a resposta voltando no outro sentido.

2. Sim. Sinais de interrupção devem saltar à frente dos dados e serem entregues fora de seqüência. Um exemplo típico ocorre quando um usuário de terminal acessa a tecla de encerramento (eliminação). O pacote gerado a partir do sinal de encerramento deve ser enviado de imediato e deve saltar à frente de quaisquer dados enfileirados atualmente para o programa; isto é, dados já digitados mas ainda não lidos.

3. As redes de circuitos virtuais quase certamente têm necessidade desse recurso para rotear pacotes de configuração de conexão de uma origem arbitrária até um destino arbitrário.

4. A negociação poderia definir o tamanho da janela, o tamanho máximo de pacote, a taxa de dados e os valores de timers.

5. A existência de quatro hops significa que cinco roteadores estão envolvidos. A implementação do circuito virtual exige a alocação de $5 \times 8 = 40$ bytes de memória por 1.000 segundos. A implementação de datagrama requer a

transmissão de $12 \times 4 \times 200 = 9.600$ bytes de cabeçalho além das necessidades de implementação do circuito virtual. Portanto, a questão se reduz ao custo relativo de 40.000 bytes-segundo de memória *versus* 9.600 bytes-hops de capacidade do circuito. Se a memória é depreciada ao longo de $2 \times 52 \times 40 \times 3.600 = 1{,}5 \times 10^7$ segundos, um byte-segundo custa $6{,}7 \times 10^{-8}$ centavos, e 40.000 deles custam pouco mais de 2 milésimos de centavos. Se um byte-hop custa 10^{-6} centavos, 9.600 deles custam 9,6 milésimos de centavos. Os circuitos virtuais são mais econômicos para esse conjunto de parâmetros.

6. Sim. Uma grande rajada de ruído poderia adulterar terrivelmente um pacote. Com um total de verificação de k bits, existe uma probabilidade de 2^{-k} de que o erro não seja detectado. Se o campo de destino ou, de modo equivalente, o número do circuito virtual for alterado, o pacote será entregue ao destino errado e aceito como genuíno. Em outras palavras, uma rajada de ruído ocasional poderia transformar um pacote perfeitamente válido para um destino em um pacote perfeitamente válido para outro destino.

7. Ele seguirá todas estas rotas: *ABCD, ABCF, ABEF, ABEG, AGHD, AGHF, AGEB e AGEF*. O número de hops usados é 24.

8. Escolha uma rota que use o caminho mais curto. Agora, remova todos os arcos usados no caminho recém-encontrado e execute novamente o algoritmo do caminho mais curto. O segundo caminho será capaz de sobreviver à falha de qualquer linha no primeiro caminho e vice-versa. Contudo, é concebível que essa heurística possa falhar ainda que existam dois caminhos disjuntos. Para resolver o problema corretamente, deve ser usado um algoritmo de fluxo máximo.

9. A ida por *B* fornece (11, 6, 14, 18, 12, 8).
A ida por *D* fornece (19, 15, 9, 3, 9, 10).
A ida por *E* fornece (12, 11, 8, 14, 5, 9).

Tomando-se o mínimo para cada destino com exceção de *C*, tem-se (11, 6, 0, 3, 5, 8). As linhas de saída são (*B*, *B*, –, *D*, *E*, *B*).

10. A tabela de roteamento tem 400 bits. Duas vezes por segundo essa tabela é gravada em cada linha, e assim são necessários 800 bps em cada linha, em cada sentido.

11. Ele sempre é mantido. Se um pacote chegou em uma linha, ele deve ser confirmado. Se nenhum pacote chegou em uma linha, ele deve ser enviado para lá. Os casos 00 (não chegou e não será enviado) e 11 (chegou e será devolvido) são logicamente incorretos, e portanto não existem.

12. O mínimo ocorre em 15 agrupamentos, cada um com 16 regiões e cada região tendo 20 roteadores, ou em uma das formas equivalentes; por exem-

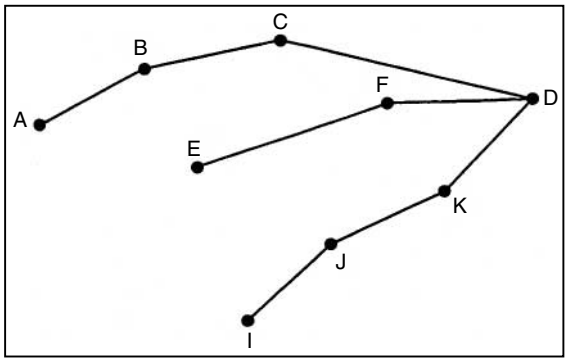
plo, 20 agrupamentos com 16 regiões de 15 roteadores. Em todos os casos, o tamanho da tabela é 15 + 16 + 20 = 51.

13. É concebível que ele possa entrar em modo promíscuo, lendo todos os quadros colocados na LAN, mas isso é muito ineficiente. Em vez disso, é normal o agente local levar o roteador a considerá-lo o host móvel, respondendo a solicitações ARP. Quando o roteador recebe um pacote IP destinado ao host móvel, ele transmite uma consulta ARP solicitando o endereço de nível MAC 802.3 da máquina com esse endereço IP. Quando o host móvel não está em atividade, o agente local responde ao ARP, e assim o roteador associa o endereço IP do usuário móvel ao endereço de nível MAC 802.3 do agente local.

14. (a) O algoritmo de encaminhamento pelo caminho inverso leva cinco rodadas para terminar. Os destinatários de pacotes nessas rodadas são *AC*, *DFIJ*, *DEGHIJKN*, *GHKN* e *LMO*, respectivamente. São gerados ao todo 21 pacotes.
(b) A árvore de escoamento necessita de quatro rodadas e 14 pacotes.

15. O nó *F* tem no momento dois descendentes, *A* e *D*. Agora, ele adquire um terceiro descendente *G* não circulado, porque o pacote que segue *IFG* não está na árvore de escoamento. O nó *G* adquire um segundo descendente, além de *D*, identificado por *F*. Esse descendente também não está circulado, pois não entra na árvore de escoamento.

16. Existem várias árvores de amplitude possíveis. Uma delas é:



17. Quando obtém o pacote, *H* o transmite. Porém, *I* sabe como chegar até *I*, e portanto não efetua a transmissão por difusão.

18. O nó *H* está a três hops de *B*, e assim leva três rodadas para encontrar a rota.

19. Ele pode fazê-lo de forma aproximada, mas não de forma exata. Suponha que existam 1.024 identificadores de nós. Se o nó 300 estiver procurando

pelo nó 800, talvez seja melhor percorrer o sentido horário, mas poderia acontecer de haver 20 nós reais entre 300 e 800 no sentido horário e apenas 16 nós reais entre eles no sentido anti-horário. O propósito da função de hash criptográfico SHA-1 é produzir uma distribuição muito suave, de forma que a densidade de nós seja aproximadamente a mesma ao longo de todo o círculo. Porém, sempre haverá flutuações estatísticas, e assim a escolha direta pode ser errada.

20. O nó na entrada 3 passa de 12 para 10.

21. O protocolo é terrível. Seja o tempo dividido em unidades de T segundos. No slot 1, o roteador de origem envia o primeiro pacote. No início do slot 2, o segundo roteador recebeu o pacote, mas ainda não pode confirmá-lo. No começo do slot 3, o terceiro roteador recebeu o pacote, mas também não pode confirmá-lo, e assim todos os roteadores atrás dele ainda estão suspensos. A primeira confirmação só pode ser enviada quando o host de destino receber o pacote do roteador de destino. Agora, a confirmação começa a se propagar de volta. São necessários dois períodos completos de trânsito da sub-rede, $2(n-1)T$ segundos, antes do roteador de origem poder enviar o segundo pacote. Desse modo, o throughput é de um pacote a cada $2(n-1)T$ segundos.

22. Cada pacote emitido pelo host de origem efetua 1, 2 ou 3 hops. A probabilidade de que o pacote efetue um hop é p. A probabilidade de ele efetuar dois hops é $p(1-p)$. A probabilidade de ele efetuar 3 hops é $(1-p)^2$. O comprimento do caminho médio que um pacote pode esperar percorrer é então a soma ponderada dessas três probabilidades, ou $p^2 - 3p + 3$. Note que, para $p = 0$ a média é 3 hops e, para $p = 1$, a média é de 1 hop. Com $0 < p < 1$, podem ser necessárias diversas transmissões. O número médio de transmissões pode ser encontrado observando-se que a probabilidade de uma transmissão bem-sucedida por toda a distância é $(1-p)^2$, que chamaremos α. O número esperado de transmissões é:

$$\alpha + 2\alpha(1-\alpha) + 3\alpha(1-\alpha)^2 + \ldots = \frac{1}{\alpha} = \frac{1}{(1-p)^2}$$

Finalmente, o total de hops usados é $(p^2 - 3p + 3)/(1-p)^2$.

23. Primeiro, o método de bit de advertência envia explicitamente uma notificação de congestionamento à origem pela definição de um bit, enquanto RED notifica implicitamente a origem, apenas descartando um de seus pacotes. Em segundo lugar, o método do bit de advertência só descarta um pacote quando não existe nenhum espaço restante no buffer, enquanto RED descarta pacotes antes que todo buffer esteja esgotado.

24. O roteador tem de fazer aproximadamente o mesmo trabalho para enfileirar um pacote, não importando o quanto ele seja grande. Existe pouca dúvida de que processar dez pacotes de 100 bytes cada um signifique muito mais trabalho que processar um pacote de 1.000 bytes.

25. Não é possível enviar nenhum pacote maior que 1.024 bytes, de qualquer modo.

26. Com um símbolo a cada 5 μs, podem ser enviadas 200.000 células/s. Cada célula contém 48 bytes de dados ou 384 bits. A taxa de dados líquida é então 76,8 Mbps.

27. A resposta ingênua é que, a 6 Mbps, ele leva 4/3 segundo para drenar um balde de 8 megabits. Porém, essa resposta está errada porque, durante esse intervalo, chegam mais símbolos. A resposta correta pode ser obtida usando-se a fórmula $S = C/(M - \rho)$. Por substituição, obtemos $S = 8/(6 - 1)$ ou 1,6 s.

28. Chame o comprimento do intervalo máximo de rajada Δt. No caso extremo, o balde está cheio no início do intervalo (1 Mbyte) e outros $10\Delta t$ Mbytes chegam durante o intervalo. A saída durante a rajada de transmissão contém $50\Delta t$ Mbytes. Igualando essas duas quantidades, temos $1 + 10\Delta t = 50\Delta t$. Resolvendo essa equação, concluímos que Δt é 25 ms.

29. As larguras de banda em MB/s são: *A*: 2, *B*: 0, *C*: 1, *E*: 3, *H*: 3, *J*: 3, *K*: 2 e *L*: 1.

30. Aqui, μ é 2 milhões e λ é 1,5 milhão; assim, $\rho = \lambda/\mu$ é igual a 0,75 e, a partir da teoria de enfileiramento, cada pacote experimenta um retardo quatro vezes maior do que teria em um sistema ocioso. O tempo em um sistema ocioso é 500 ns; aqui, ele é igual a 2 μs. Com 10 roteadores ao longo de um caminho, o tempo de enfileiramento somado ao tempo de serviço é 20 μs.

31. Não existe nenhuma garantia. Se muitos pacotes forem expedidos, seu canal talvez tenha desempenho pior que o do canal normal.

32. Ela é necessária em ambos. Mesmo em uma rede de circuitos virtuais concatenados, algumas redes ao longo do caminho poderiam aceitar pacotes de 1.024 bytes, enquanto outras só poderiam aceitar pacotes de 48 bytes. A fragmentação ainda é necessária.

33. Nenhum problema. Basta encapsular o pacote no campo de carga útil de um datagrama pertencente à sub-rede que está sendo percorrida e enviá-lo.

34. O datagrama IP inicial estará fragmentado em dois datagramas IP em I1. Não ocorrerá nenhuma outra fragmentação.

Enlace *A-R1*:

$$Comprimento = 940; ID = x; DF = 0; MF = 0; Deslocamento = 0$$

Enlace *R1-R2*:

(1)*Comprimento* = 500; *ID* = *x*; *DF* = 0; *MF* = 1; *Deslocamento* = 0
(2) *Comprimento* = 460; *ID* = *x*; *DF* = 0; *MF* = 0; *Deslocamento* = 60

Enlace *R2-B*:

(1)*Comprimento* = 500; *ID* = *x*; *DF* = 0; *MF* = 1; *Deslocamento* = 0
(2) *Comprimento* = 460; *ID* = *x*; *DF* = 0; *MF* = 0; *Deslocamento* = 60

35. Se a taxa de bits da linha é *b*, o número de pacotes/s que o roteador pode emitir é *b*/8192; assim, o número de segundos que ele leva para emitir um pacote é 8192/*b*. Para transmitir 65.536 pacotes, ele demora $2^{29}/b$ segundos. Igualando essa expressão à duração máxima de pacotes, obtemos $2^{29}/b = 10$. Então, *b* é cerca de 53.687.091 bps.

36. Tendo em vista que a informação é necessária para rotear cada fragmento, a opção deve aparecer em todo fragmento.

37. Com um prefixo de 2 bits, restariam 18 bits para indicar a rede. Conseqüentemente, o número de redes seria 2^{18}, ou 262.144. Porém, todos os valores 0 e todos os valores 1 são especiais, e assim apenas 262.142 estão disponíveis.

38. O endereço é 194.47.21.130.

39. A máscara tem 20 bits de comprimento, e assim a parte de rede tem 20 bits. Os 12 bits restantes são para o host, e portanto existem 4.096 endereços de hosts.

40. Para começar, todas as solicitações são arredondadas até uma potência de dois. O endereço inicial, o endereço final e a máscara são dados a seguir:

A: 198.16.0.0 – 198.16.15.255 escrito como 198.16.0.0/20
B: 198.16.16.0 – 198.23.15.255 escrito como 198.16.16.0/21
C: 198.16.32.0 – 198.47.15.255 escrito como 198.16.32.0/20
D: 198.16.64.0 – 198.95.15.255 escrito como 198.16.64.0/19

41. Eles podem ser agregados a 57.6.96/19.

42. É suficiente adicionar uma nova entrada de tabela: 29.18.0.0/22 para o novo bloco. Se um pacote de entrada corresponder a 29.18.0.0/17 e também a 29.18.0.0/22, o mais longo vencerá. Essa regra torna possível atribuir um grande bloco a uma única linha de saída, mas fazer uma exceção para um ou mais blocos pequenos dentro de seu intervalo.

43. Os pacotes são roteados como:

(a) Interface 1
(b) Interface 0
(c) Roteador 2

(d) Roteador 1
(e) Roteador 2

44. Depois que o NAT é instalado, é crucial que todos os pacotes pertencentes a uma única conexão entrem e saiam da empresa pelo mesmo roteador, pois é nele que o mapeamento é mantido. Se cada roteador tiver seu próprio endereço IP e todo o tráfego pertencente a uma dada conexão puder ser enviado ao mesmo roteador, o mapeamento poderá ser feito de modo correto, e o multihoming com o NAT poderá funcionar.

45. Você dirá que o ARP não fornece um serviço à camada da rede; ele faz parte da camada de rede e ajuda a fornecer um serviço à camada de transporte. A questão de endereçamento IP não ocorre na camada de enlace de dados. Os protocolos da camada de enlace de dados são semelhantes aos protocolos 1 a 6 do Capítulo 3, HDLC, PPP etc. Eles movem bits de uma extremidade de uma linha até a outra.

46. O RARP tem um servidor RARP que responde a solicitações. O ARP não tem esse recurso. Os próprios hosts respondem a consultas ARP.

47. No caso geral, o problema não é trivial. Os fragmentos podem chegar fora de ordem e alguns podem estar faltando. Em uma retransmissão, o datagrama pode estar fragmentado em blocos de diferentes tamanhos. Além disso, o tamanho total não é conhecido até chegar o último fragmento. Talvez o único modo de realizar a remontagem seja inserir no buffer todos os fragmentos, até chegar o último e o tamanho ser conhecido. Depois, crie um buffer com o tamanho correto e insira os fragmentos no buffer, mantendo um mapa de bits com 1 bit por 8 bytes para controlar os bytes que estão presentes no buffer. Quando todos os bits no mapa de bits forem iguais a 1, o datagrama estará completo.

48. No que se refere ao receptor, isso faz parte de um novo datagrama, pois nenhuma outra parte dele é conhecida. Portanto, ele será enfileirado até aparecer o restante. Se as outras partes não forem recebidas, essa também chegará ao tempo limite.

49. Um erro no cabeçalho é muito mais sério que um erro nos dados. Por exemplo, um endereço incorreto poderia resultar na entrega de um pacote ao host errado. Muitos hosts não verificam se um pacote que lhes foi entregue é de fato destinado a eles, supondo que a rede nunca lhes dará pacotes destinados a outro host. Algumas vezes, os dados não são verificados porque isso é dispendioso, e as camadas superiores com freqüência fazem essa verificação de qualquer modo, tornando-a redundante aqui.

50. Sim. O fato de a LAN Minneapolis ser sem fio não faz os pacotes que chegam para ela em Boston saltarem repentinamente para Minneapolis. O

agente local (home-agent) em Boston deve canalizá-los para o agente externo (foreign-agent) na LAN sem fios em Minneapolis. A melhor maneira de imaginar essa situação é considerar que o usuário se conectou à LAN de Minneapolis, da mesma forma que todos os outros usuários de Minneapolis. O fato da conexão usar rádio em vez de cabos é irrelevante.

51. Com 16 bytes, existem 2^{128} ou $3{,}4 \times 10^{38}$ endereços. Se os alocarmos à velocidade de 10^{18} por segundo, eles irão durar por 10^{13} anos. Esse número é 1.000 vezes a idade do universo. É claro que o espaço de endereços não é plano, de modo que eles não são alocados linearmente, mas esse cálculo mostra que até mesmo um esquema de alocação que tenha uma eficiência de 1/1.000 (0,1%), nunca se esgotará.

52. O campo *Protocol* informa ao host de destino a que tratador de protocolos deve ser dado o pacote IP. Roteadores intermediários não precisam dessa informação, e assim ela não é necessária no cabeçalho principal. Na realidade, ela está lá, mas disfarçada. O campo *Next header* do último cabeçalho (de extensão) é usado para esse propósito.

53. Conceitualmente, não há mudanças. Tecnicamente, os endereços IP solicitados agora são maiores, e assim são necessários campos maiores.

SOLUÇÕES DOS PROBLEMAS DO CAPÍTULO 6

1. A chamada LISTEN poderia indicar a vontade de estabelecer novas conexões, mas não de bloquear. Quando fosse feita uma tentativa de conexão, o chamador poderia receber um sinal. Então ele executaria, digamos, OK ou REJECT para aceitar ou rejeitar a conexão. Em nosso esquema original, essa flexibilidade está ausente.

2. A linha tracejada de *PASSIVE ESTABLISHMENT PENDING* até *ESTABLISHED* não depende mais da chegada de uma confirmação. A transição pode acontecer imediatamente. Em essência, o estado *PASSIVE ESTABLISHMENT PENDING* desaparece, pois ele nunca é visível em qualquer nível.

3. Se o cliente envia um pacote a *SERVER_PORT* e o servidor não está escutando essa porta, o pacote não será entregue ao servidor.

4. (a) O clock leva 32.768 pulsos ou 3.276,8 segundos para completar o ciclo. À taxa de geração zero, o transmissor entraria na zona proibida em $3.276{,}8 - 60 = 3.216{,}8$ segundos.
(b) A 240 números de seqüência/min, o número de seqüência real é $4t$, onde t é medido em segundos. A margem esquerda da região proibida é $10(t - 3.216{,}8)$. Igualando essas duas fórmulas, descobrimos que a interseção entre elas ocorre em $t = 5.361{,}3$ segundos.

5. Observe o segundo pacote duplicado na Figura 6.11(b). Quando esse pacote chegasse, seria um desastre se as confirmações para *y* ainda estivessem presentes.

6. Os impasses são possíveis. Por exemplo, um pacote chega inesperadamente a *A*, e *A* o confirma. A confirmação se perde, mas agora *A* está aberto, enquanto *B* nada sabe sobre o que aconteceu. Em seguida, o mesmo acontece a *B*, e então ambos ficam abertos, mas esperando números de seqüência diferentes. Os tempos limite têm de ser introduzidos para evitar os impasses.

7. Não. O problema é essencialmente o mesmo com mais de dois exércitos.

8. Se o tempo AW ou WA é pequeno, os eventos AC(W) e WC(A) são eventos improváveis. O transmissor deve retransmitir no estado *S1*; a ordem do receptor não importa.

9. Sim. Ambos os lados poderiam executar RECEIVE simultaneamente.

10. Sim, $n_2 + n_3 + n_6 + n_7 = 1$. Os estados *listening*, *waiting*, *sending* e *receiving* implicam todos que o usuário está bloqueado e portanto não pode estar também em outro estado.

11. Uma mensagem de comprimento zero é recebida pelo outro lado. Ela poderia ser usada para assinalar o fim do arquivo.

12. Nenhuma das primitivas pode ser executada, porque o usuário está bloqueado. Desse modo, somente são possíveis eventos de chegada de pacotes, e não todos eles. *CallReq*, *ClearReq*, *DataPkt* e *Credit* são os únicos válidos.

13. A janela deslizante é mais simples, tendo apenas um conjunto de parâmetros (as arestas da janela) para administrar. Além disso, o problema de uma janela ser aumentada e depois diminuída, com as TPDUs chegando na ordem errada, não ocorre. Porém, o esquema de crédito é mais flexível, permitindo um gerenciamento dinâmico da bufferização, separada das confirmações.

14. Não. Os pacotes IP contêm endereços IP que especificam uma máquina de destino. Uma vez que tal pacote chegasse, como o tratador de rede saberia a que processo entregá-lo? Os pacotes UDP contêm uma porta de destino. Essa informação é essencial para que eles possam ser entregues ao processo correto.

15. É possível que um cliente possa obter o arquivo errado. Suponha que o cliente *A* envie uma solicitação para o arquivo *f*1 e depois sofra uma pane. Então, outro cliente *B* usa o mesmo protocolo para solicitar outro arquivo *f*2. Suponha que o cliente *B*, funcionando na mesma máquina que *A* (com

o mesmo endereço IP), vincule seu soquete UDP à mesma porta que *A* esteve usando antes. Além disso, suponha que a solicitação de *B* se perca. Quando a resposta do servidor chegar (para a solicitação de *A*), o cliente *B* a receberá e presumirá que ela é uma resposta à sua própria solicitação.

16. O envio de 1.000 bits sobre uma linha de 1 Gbps demora um 1 μs. A velocidade da luz na fibra óptica é 200 km/ms; assim, demora 0,5 ms para a solicitação chegar e outro 0,5 ms para a resposta voltar. Ao todo, 1.000 bits são transmitidos em 1 ms. Isso é equivalente a 1 megabit/s, ou 0,1% de eficiência.

17. A 1 Gbps, o tempo de resposta é determinado pela velocidade da luz. O melhor que pode ser alcançado é 1 ms. A 1 Mbps, a linha demora cerca de 1 ms para bombear os 1.024 bits, 0,5 ms para o último chegar ao servidor e 0,5 ms para a resposta voltar, no melhor caso. O melhor tempo de RPC possível é então 2 ms. A conclusão é que melhorar a velocidade da linha por um fator de 1.000 só rende um fator de dois em desempenho. A menos que a linha de gigabits seja incrivelmente econômica, é provável que não compense ter essa aplicação.

18. Aqui estão três razões. Primeiro, as IDs de processos são específicas do SO. O uso de IDs de processos tornaria esses protocolos dependentes do SO. Em segundo lugar, um único processo pode estabelecer vários canais de comunicações. Uma única ID de processo (por processo) não pode ser usada como identificador de destino para fazer distinção entre esses canais. Em terceiro lugar, fazer os processos escutarem em portas conhecidas é fácil, mas são impossíveis IDs de processos conhecidas.

19. O segmento padrão tem 536 bytes. O TCP acrescenta 20 bytes, da mesma forma que o IP, o que resulta no padrão de 576 bytes ao todo.

20. Embora cada datagrama chegue intacto, é possível que os datagramas cheguem na ordem errada; assim, o TCP tem de estar preparado para montar novamente as peças de uma mensagem de maneira apropriada.

21. Cada amostra ocupa 4 bytes. Isso dá um total de 256 amostras por pacote. Existem 44.100 amostras/s; assim, com 256 amostras/pacote, ele levará 44.100/256 ou 172 pacotes para transmitir o equivalente a um segundo de música.

22. Sem dúvida. O chamador teria de fornecer todas as informações necessárias, mas não há razão para que o RTP não possa estar no núcleo, da mesma maneira que o UDP.

23. Não. Uma conexão é identificada apenas por seus soquetes. Desse modo, $(1, p) – (2, q)$ é a única conexão possível entre essas duas portas.

24. O bit ACK é usado para informar se o campo de 32 bits é usado. Porém, se ele não existisse, o campo de 32 bits sempre teria de ser usado, se necessário confirmando um byte que já tivesse sido confirmado. Em resumo, ele não é absolutamente essencial para o tráfego de dados normal. No entanto, ele desempenha um papel crucial durante o estabelecimento de conexões, onde é usado na segunda e na terceira mensagem do handshake de três vias.

25. O segmento de TCP inteiro deve caber no campo de carga útil de 65.515 bytes de um pacote IP. Tendo em vista que o cabeçalho de TCP tem no mínimo 20 bytes, restam somente 65.495 bytes para dados de TCP.

26. Uma alternativa é começar com um LISTEN. Se for recebido um *SYN*, o protocolo entrará no estado *SYN RECD*. O outro modo ocorre quando um processo tenta fazer uma abertura ativa e enviar um *SYN*. Se o outro lado também estivesse se abrindo e fosse recebido um *SYN*, ele também entraria no estado *SYN RECD*.

27. Embora o usuário esteja digitando a uma velocidade uniforme, os caracteres serão ecoados em rajadas. O usuário pode pressionar várias teclas sem que nada apareça na tela e depois, repentinamente, o conteúdo da tela alcançar a digitação. É possível que as pessoas considerem esse efeito incômodo.

28. As primeiras rajadas contêm respectivamente 2K, 4K, 8K e 16K bytes. A rajada seguinte é de 24 KB e ocorre depois de 40 ms.

29. A próxima transmissão será um segmento de tamanho máximo 1. Depois, teremos 2, 4 e 8. Assim, após quatro sucessos, ele terá 8 KB.

30. As estimativas sucessivas são 29,6, 29,84, 29,256.

31. Uma janela pode ser enviada a cada 20 ms. Isso dá 50 janelas/s, correspondendo a uma taxa de dados máxima de cerca de 3,3 milhões de bytes/s. A eficiência da linha é então 26,4 Mbps/1.000 Mbps, ou 2,6%.

32. A meta é enviar 2^{32} bytes em 120 segundos ou a carga útil de 35.791.394 bytes/s. Isso equivale a 23.860 quadros de 1.500 bytes por segundo. O overhead do TCP é de 20 bytes. O overhead do IP é de 20 bytes. O overhead da Ethernet é de 26 bytes. Isso significa que, para 1.500 bytes de carga útil, devem ser enviados 1.566 bytes. Se fôssemos enviar 23.860 quadros de 1.566 bytes a cada segundo, precisaríamos de uma linha de 299 Mbps. Com qualquer velocidade maior que essa, correremos o risco de encontrar dois segmentos TCP diferentes com o mesmo número de seqüência ao mesmo tempo.

33. Um transmissor pode não enviar mais de 255 TPDUs, isto é, $255 \times 128 \times 8$ bits em 30 segundos. Portanto, a taxa de dados não é maior que 8,704 kbps.

34. Calcule a média: (270.000 × 0 + 730.000 × 1 ms)/1.000.000. A demora é de 730 μs.

35. Ela leva 4 × 10 = 40 instruções para copiar 8 bytes. Quarenta instruções demoram 40 ns. Desse modo, cada byte exige 5 ns de tempo da CPU para cópia. Assim, o sistema é capaz de manipular 200 MB/s ou 1.600 Mbps. Ele pode tratar uma linha de 1 Gbps, se não houver nenhum outro gargalo.

36. O tamanho do espaço de seqüências é 2^{64} bytes, o que equivale a cerca de 2×10^{19} bytes. Um transmissor de 75 Tbps consome um espaço de seqüências à taxa de $9{,}375 \times 10^{12}$ números de seqüências por segundo. Ele demora 2 milhões de segundos para recomeçar. Tendo em vista que há 86.400 segundos em um dia, ele irá demorar mais de três semanas para recomeçar, mesmo a 75 Tbps. Um tempo máximo de duração de pacote menor que três semanas evitará o problema. Em resumo, a mudança para 64 bits provavelmente funcionará por um bom tempo.

37. O RPC sobre o UDP leva apenas dois pacotes em vez de três. Entretanto, o RPC terá problemas se a resposta não couber em um único pacote.

38. Sim. O pacote 6 confirma tanto a solicitação quanto o FIN. Se cada um fosse confirmado separadamente, teríamos 10 pacotes na seqüência. Como outra alternativa, o pacote 9 – que confirma a resposta – e o FIN também poderiam ser divididos em dois pacotes separados. Desse modo, o fato de haver nove pacotes se deve simplesmente à boa sorte.

39. Com um pacote 11,72 vezes menor, você obtém 11,72 vezes mais pacotes por segundo, e assim cada pacote só recebe 6.250/11,72, ou 533 instruções.

40. A velocidade da luz na fibra e no cobre é aproximadamente 200 km/ms. Para uma linha de 20 km, o retardo é de 100 μs em um sentido e 200 μs no percurso de ida e volta. Um pacote de 1 KB tem 8.192 bits. Se o tempo para enviar 8.192 bits e receber a confirmação é de 200 μs, os retardos de transmissão e propagação são iguais. Se B é o tempo de bit, temos $8.192B = 2 \times 10^{-4}$ s. A taxa de dados, $1/B$, é então cerca de 40 Mbps.

41. As respostas são: (1) 18,75 KB, (2) 125 KB, (3) 562,5 KB, (4) 1,937 MB. Uma janela de tamanho 16 bits significa que um transmissor pode enviar no máximo 64 KB antes de ser obrigado a esperar por uma confirmação. Isso significa que um transmissor não pode transmitir continuamente usando o TCP e manter o canal cheio, se a tecnologia de rede utilizada for Ethernet, T3 ou STS-3.

42. O retardo de ida e volta é cerca de 540 ms, e assim com um canal de 50 Mbps, o retardo de produto da largura de banda é 27 megabits, ou

3.375.000 bytes. Com pacotes de 1.500 bytes, ele leva 2.250 pacotes para preencher o canal; portanto, a janela deve ter pelo menos a capacidade de 2.250 pacotes.

SOLUÇÕES DOS PROBLEMAS DO CAPÍTULO 7

1. Eles são o nome DNS, o endereço IP e o endereço Ethernet.
2. Seu endereço IP começa com 130; portanto, ele está em uma rede da classe B. Veja no Capítulo 5 o mapeamento de endereços IP.
3. Não é um nome absoluto, mas relativo a *.cs.vu.nl*. Na realidade, é apenas uma notação abreviada para *rowboat.cs.vu.nl*.
4. Significa: meus lábios estão selados. É usado em resposta a uma solicitação para guardar um segredo.
5. O DNS é idempotente. As operações podem ser repetidas sem danos. Quando um processo faz uma solicitação DNS, ele inicia um timer. Se o timer expirar, ele simplesmente repetirá a solicitação. Não haverá nenhum dano.
6. O problema não acontece. Os nomes DNS *têm de ser* mais curtos que 256 bytes. O padrão exige isso. Desse modo, todos os nomes DNS cabem em um único pacote de comprimento mínimo.
7. Sim. De fato, na Figura 7.3, vemos um exemplo de um endereço IP duplicado. Lembre-se de que um endereço IP consiste em um número de rede e um número de host. Se uma máquina tem duas placas Ethernet, ela pode estar em duas redes separadas e, nesse caso, precisa de dois endereços IP.
8. É possível. *www.large-bank.com* e *www.Iarge-bank.ny.us* poderiam ter o mesmo endereço IP. Desse modo, uma entrada sob *com* e sob um dos domínios de países é sem dúvida possível (e comum).
9. É óbvio que existem muitas abordagens. Uma delas é transformar o servidor de nível superior em uma server farm (fazenda de servidores). Outra é ter 26 servidores separados, um para nomes que começam com *a*, um para *b* e assim por diante. Durante algum período de tempo (digamos, três anos) após a introdução dos novos servidores, o antigo poderia continuar a operar, a fim de dar às pessoas a oportunidade de adaptar seu software.
10. Ele pertence ao envelope, porque o sistema de entrega precisa conhecer seu valor para tratar o correio eletrônico que não pode ser entregue.
11. Isso é muito mais complicado do que se poderia imaginar. Para começar, cerca de metade do mundo escreve os primeiros nomes antes, seguidos

pelo nome da família, enquanto a outra metade (por exemplo, China e Japão) faz o contrário. Um sistema de nomenclatura teria de distinguir um número arbitrário de nomes dados, mais um nome de família, embora esse último possa ter várias partes, como em John von Neumann. Então, há pessoas que têm uma inicial intermediária, mas nenhum nome intermediário. Diversos títulos, como Sr., Srta., Sra., Dr., Prof. ou Lorde podem servir de prefixos para o nome. As pessoas se incluem em gerações, e assim Jr., Sr., III, IV e assim por diante têm de ser acrescentados. Algumas pessoas usam seus títulos acadêmicos em seus nomes, e portanto precisamos de B.A., B.Sc., M.A., M.Sc., Ph.D. e outros títulos. Finalmente, há pessoas que incluem certos prêmios e honrarias em seus nomes. Um Fellow da Royal Society na Inglaterra poderia acrescentar FRS, por exemplo. Agora, devemos entender nomes como:

Profª. Drª. Abigail Barbara Cynthia Doris E. de Vries III, Ph.D., FRS.

12. É realizável e relativamente simples. Ao chegar o correio eletrônico de entrada, o daemon de SMTP que o aceita tem de procurar o nome de login na mensagem *RCPT TO*. É claro que existe um arquivo ou um banco de dados em que esses nomes estão localizados. Esse arquivo poderia ser estendido para admitir nomes alternativos (aliases) da forma "Ellen.Johnson" que conduzissem à caixa de correio da pessoa. Então, o correio eletrônico sempre pode ser enviado pela utilização do nome real da pessoa.

13. A codificação de base 64 dividirá a mensagem em até 1.024 unidades de 3 bytes cada. Cada uma dessas unidades será codificada como 4 bytes, dando um total de 4.096 bytes. Se esses forem então divididos em linhas de 80 bytes, serão necessárias 52 linhas, acrescentando-se 52 CRs e 52 LFs. O comprimento total será então 4.200 bytes.

14. Se uma seqüência iniciada com um sinal de igualdade e seguida por dois dígitos hexadecimais aparecer no texto – por exemplo, =FF – essa seqüência será interpretada erroneamente como uma seqüência de escape. A solução é codificar o próprio sinal de igualdade, de forma que todos os sinais de igualdade sempre iniciem seqüências de escape.

15. Alguns exemplos e auxiliares possíveis são application/msexcel (Excel), application/ppt (PowerPoint), audio/midi (som MIDI), image/tiff (qualquer visualizador de imagens gráficas), video/x-dv (reprodutor QuickTime).

16. Sim, use o subtipo *message/external-body* e simplesmente envie o URL do arquivo em vez do arquivo real.

17. A mensagem enviada imediatamente antes da desconexão irá gerar uma resposta pronta. Sua chegada também gerará uma resposta pronta. Supon-

do que cada máquina registre os endereços de correio eletrônico aos quais já respondeu, não será enviada mais nenhuma resposta pronta.

18. A primeira definição é qualquer seqüência de um ou mais espaços e/ou tabulações. A segunda é qualquer seqüência de um ou mais espaços e/ou tabulações e/ou retrocessos, sujeita à condição de que o resultado final da aplicação de todos os retrocessos ainda deixe pelo menos um espaço ou uma tabulação restante.

19. As respostas reais têm de ser dadas pelo agente de transferência de mensagens. Quando uma conexão SMTP é estabelecida, o agente de transferência de mensagens tem de verificar se há um daemon de férias configurado para responder ao correio eletrônico recebido e, nesse caso, enviar uma resposta. O agente de transferência do usuário não pode fazer isso, porque nem mesmo será invocado enquanto o usuário não retornar das férias.

20. Não. Na realidade, o programa POP3 não toca na caixa de correio remota. Ele envia comandos ao daemon POP3 no servidor de correio. Desde que esse daemon reconheça o formato da caixa de correio, ele poderá funcionar. Desse modo, um servidor de correio poderia passar de um formato para outro durante a noite sem informar a seus clientes, desde que ele alterasse simultaneamente seu daemon POP3 para que este reconhecesse o novo formato.

21. O armazenamento das mensagens de correio eletrônico dos usuários ocupa espaço em disco, que custa dinheiro. Esse fator é um argumento em favor da utilização do POP3. Por outro lado, o ISP poderia cobrar pelo espaço de armazenamento de disco acima de alguns megabytes, transformando assim o correio eletrônico em uma fonte de renda. Esse último argumento leva o IMAP a incentivar os usuários a conservarem o correio eletrônico no servidor (e pagar pelo espaço em disco).

22. Ele não utiliza nenhum dos dois. Porém, é bastante semelhante em espírito ao IMAP, porque os dois permitem a um cliente remoto examinar e administrar uma caixa de correio remota. Em contraste, o POP3 simplesmente envia a caixa de correio ao cliente para processamento no local.

23. O navegador tem de ser capaz de saber se a página é de texto, áudio, vídeo ou algo diferente. Os cabeçalhos MIME fornecem essa informação.

24. Se um navegador recebe uma página com um tipo MIME que não pode tratar, ele chama um visualizador externo para exibir a página. Ele encontra o nome do visualizador em uma tabela de configuração ou o recebe do usuário.

25. Sim, é possível. O auxiliar que será iniciado depende das tabelas de configuração internas do navegador, e o Netscape e o IE podem estar configura-

dos de modo diferente. Além disso, o IE leva mais a sério a extensão do arquivo que o tipo MIME, e a extensão do arquivo pode indicar um auxiliar diferente do tipo MIME.

26. Se um módulo fizer duas solicitações, uma delas será um acerto de cache e uma será um erro de cache, em média. O tempo total de CPU consumido é 1 ms, e o tempo total de espera é 9 ms. Isso nos dá 10% de utilização da CPU; assim, com 10 módulos, a CPU será mantida ocupada.

27. A maneira oficial da RFC 1738 fazer isso é *http://nome-dns:porta/arquivo*.

28. Os nomes DNS não podem terminar com um dígito, e assim não existe nenhuma ambigüidade.

29. O URL deve ser: *ftp://www.cs.stanford.edu/ftp/pub/freebies/newprog.c* .

30. Faça como *toms-cassino*: simplesmente coloque uma ID de cliente no cookie e armazene as preferências em um banco de dados no servidor indexado pela ID de cliente. Desse modo, o tamanho do registro será ilimitado.

31. Tecnicamente, funcionará, mas é uma idéia terrível. Tudo que o cliente tem a fazer é modificar o cookie para conseguir acesso à conta bancária de outra pessoa. Fazer o cookie fornecer a identidade do cliente é seguro, mas o cliente deve ser obrigado a digitar uma senha para provar sua identidade.

32. Se o usuário desativou a exibição automática de imagens ou se as imagens não podem ser exibidas por alguma outra razão, então o texto dado em *ALT* será exibido em lugar da imagem. Além disso, se o mouse se movimentar sobre a imagem, é possível que o texto seja exibido.

33. Um hiperlink consiste em e . Entre essas tags está o texto clicável. Também é possível inserir uma imagem nesse local. Por exemplo:

<img src="http://www.abcd.com/im/im2">.

34. Ela seria: ACM .

35. Aqui está uma maneira de fazê-lo.

```
<html>
<head> <title> INTERBURGER </title> </head>
<body>
<h1> interburger's order form </h1>
<form action=http://interburger.com/cgi-bin/burgerorder"
method=POST> <p> Name <input name="customer" size=46> </p>
<p> Street Address <input name="address" size=40> </p>
<p> City <input name="city" size=20> </p>
Burger size Gigantic <<input name="size" type=radio
value="gigantic">
```

```
Immense <input name="size" type=radio value="immense">
Cheese <input name="cheese" type=checkbox>
<p> <input type=submit value="submit order"> </p>
</form>
</body> </html>
```

36. A página que exibe o formulário é semelhante a esta:

```
<html>
<head> <title> Adder </title> </head>
<body>
<form action="action.php" method="post">
<p> Please enter first number: <input type="text" name="first"> </p>
<p> Please enter second number: <input type="text" name="second"> </p>
<input type="submit">
</form>
</body>
</html>
```

O script de PHP que faz o processamento é semelhante a:

```
<html>
<head> <title> Addition </title> </head>
<body>
The sum is <?PHP echo $first + $second; ?>
</body>
</html>
```

37. (a) Existem apenas 14 calendários anuais, dependendo do dia da semana em que cai o dia 1º de janeiro e do fato de o ano ser bissexto ou não. Desse modo, um programa JavaScript poderia conter com facilidade todos os 14 calendários e um pequeno banco de dados do ano que corresponde a cada calendário. Um script PHP também poderia ser utilizado, mas seria mais lento.

(b) Isso exige um grande banco de dados. Ele tem de ser criado no servidor, usando-se PHP.
(c) Ambos funcionam, mas JavaScript é mais rápido.

38. É óbvio que existem muitas soluções possíveis. Aqui está uma.

```
<html>
<head> <title> JavaScript test </title> </head>
<script language="javascript" type="text/javascript">
function response(test_form) {
   var n =2;
   var has_factors = 0;
   var number = eval(test_form.number.value);
   var limit = Math.sqrt(number);
   while (n++ << limit) if (number % n == 0) has_factors = 1;
```

```
    document.open();
    document.writeln("<html> <body>");
    if (has_factors > 0) document.writeln(number, " is not a prime");
    if (has_factors == 0) document.writeln(number, " is a prime");
    document.writeln("</body> </html>");
    document.close();

}
</script>
</head>

<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute primality"
onclick="response(this.form)">
</form>
</body>
</html>
```

É claro que isso pode ser melhorado de várias maneiras, mas esses aperfeiçoamentos exigem um pouco mais de conhecimento de JavaScript.

39. Os comandos enviados são:

```
GET /welcome.html HTTP/1.1
Host: www.info-source.com
```

Observe a linha em branco no final. Ela é obrigatória.

40. É mais provável que a páginas HTML mudem com maior freqüência do que arquivos JPEG. Um grande número de sites altera seus arquivos HTML o tempo todo, mas não muda muito as imagens. Porém, a efetividade se relaciona não apenas à taxa de acertos, mas também à compensação. Não existe muita diferença entre receber uma mensagem 304 e receber 500 linhas de HTML. O retardo é essencialmente o mesmo em ambos os casos, porque os arquivos HTML são muito pequenos. Os arquivos de imagens são grandes, e então é sempre vantajoso não ter de enviar um arquivo desse tipo .

41. Não. No caso dos esportes, sabe-se com vários dias de antecedência que haverá um grande movimento no Web site e podem ser construídas réplicas em vários lugares. Inesperada é a essência do sucesso instantâneo. Houve um grande movimento no Web site da Flórida, mas não nos sites de Iowa ou de Minnesota. Ninguém poderia prever com antecedência tal sucesso.

42. Sem dúvida. O ISP vai a vários provedores de conteúdo e consegue permissão para replicar o conteúdo no site do ISP. O provedor de conteúdo pode

até pagar por isso. A desvantagem é o fato de ser muito trabalhoso para o ISP entrar em contato com muitos provedores de conteúdo. É mais fácil deixar um CDN fazer isso.

43. É uma idéia ruim se o conteúdo muda rapidamente. Por exemplo, páginas repletas de resultados esportivos ou de cotações de ações atualizados até o último segundo não são boas candidatas. Páginas que são geradas dinamicamente não são apropriadas.

44. Cada kanji (palavra) em japonês recebe a atribuição de um número. Existem mais de 20.000 deles em Unicode. Para um sistema totalmente em inglês, seria possível atribuir às 65.000 palavras mais comuns um código de 16 bits e simplesmente transmitir o código. O terminal adicionaria automaticamente um espaço entre palavras. Palavras não existentes na lista seriam grafadas em ASCII. Usando-se esse esquema, a maioria das palavras ocuparia 2 bytes, bem menos que se fossem transmitidas caractere por caractere. Outros esquemas poderiam envolver a utilização de códigos de 8 bits para representar as palavras mais comuns e códigos mais longos para palavras menos freqüentes (codificação de Huffman primitiva).

45. O áudio necessita de 1,4 Mbps, que equivale a 175 KB/s. Em um dispositivo de 650 MB, há espaço para 3.714 segundos de áudio, o que significa pouco mais de uma hora. Os CDs nunca têm mais de uma hora de duração, e portanto não há necessidade de compactação, e ela não é usada.

46. Os valores verdadeiros são $\text{sen}(2\pi i/32)$, para i de 1 a 3. Numericamente, esses senos são 0,195, 0,383 e 0,556. Eles são representados como 0,250, 0,500 e 0,500, respectivamente. Desse modo, os erros percentuais são 28%, 31% e 10%, respectivamente.

47. Na teoria ele poderia ser usado, mas a telefonia da Internet ocorre em tempo real. No caso de música, não há objeção em se gastar cinco minutos para se codificar uma canção de três minutos. Para voz em tempo real, isso não funcionaria. A compactação psicoacústica poderia funcionar no caso da telefonia, mas apenas se existisse um chip que pudesse realizar a compactação durante a execução com um retardo de aproximadamente 1 ms.

48. Ele demora 50 ms para enviar um comando de pausa ao servidor e, durante esse tempo, chegarão 6.250 bytes; assim, a marca d'água baixa deve ficar acima de 6.250, provavelmente em 50.000, por segurança. De modo semelhante, a marca d'água alta deve ficar pelo menos 6.250 bytes distante do máximo mas, digamos, 50.000 seria um valor mais seguro.

49. Ele introduz um retardo extra. No esquema direto, após terem decorrido 5 ms, o primeiro pacote pode ser enviado. Nesse esquema, o sistema tem de esperar até 10 ms para poder enviar as amostras correspondentes aos primeiros 5 ms.

50. Depende. Se o chamador não estiver atrás de um firewall e o chamado estiver em um telefone comum, não haverá nenhum problema. Se o chamador estiver atrás de um firewall e o firewall não for exigente quanto ao que sai do site, ele também funcionará. Se o chamado estiver atrás de um firewall que não permita a saída de pacotes UDP, ele não funcionará.

51. O número de bits/s é exatamente $800 \times 600 \times 40 \times 8$, ou 153,6 Mbps.

52. Sim. Um erro em um 1 quadro I causará erros na reconstrução de quadros P e de quadros B subseqüentes. De fato, o erro pode continuar a se propagar até o próximo quadro I.

53. Com 100.000 clientes, cada um adquirindo dois filmes por mês, o servidor transmite 200.000 filmes por mês, ou cerca de 6.600 por dia. Se metade deles é transmitida no horário nobre, o servidor deve manipular aproximadamente 3.300 filmes ao mesmo tempo. Se o servidor tiver de transmitir 3.300 filmes a 4 Mbps cada, a largura de banda necessária será de 13,2 Gbps. Usando-se conexões OC-12, com uma capacidade SPE de 594 Mbps cada, serão necessárias no mínimo 23 conexões. Uma máquina que serve 3.300 filmes simultaneamente sobre 23 conexões OC-12 não é de modo algum uma máquina pequena.

54. A fração de todas as referências aos primeiros r filmes é dada por:

$$C/1 + C/2 + C/3 + C/4 + \bullet\bullet\bullet + C/r$$

Desse modo, a razão entre os 1000 primeiros filmes e os 10.000 primeiros é:

$$\frac{1/1 + 1/2 + 1/3 + 1/4 + \ldots + 1/1.000}{1/1 + 1/2 + 1/3 + 1/4 + \ldots + 1/10.000}$$

porque os valores de C se cancelam. Avaliando essa equação numericamente, obtemos 7,486/9,788. Portanto, cerca de 0,764 de todas as solicitações serão de filmes em disco magnético. Vale a pena notar que a lei de Zipf implica uma proporção substancial da distribuição no final, comparada, digamos, ao decaimento exponencial.

SOLUÇÕES DOS PROBLEMAS DO CAPÍTULO 8

1. the time has come the walrus said to talk of many things
of ships and shoes and sealing wax of cabbages and kings
and why the sea is boiling hot and whether pigs have wings
but wait a bit the oysters cried before we have our chat
for some of us are out of breath and all of us are fat
no hurry said the carpenter they thanked him much for that

De *Through the Looking Glass* (Tweedledum e Tweedledee).

2. O texto simples (em inglês) é: "a digital computer is a machine that can solve problems for people by carrying out instructions given to it".

De *Organização Estruturada de Computadores*, de A. S. Tanenbaum.

3. Aqui está:

1011111 0000100 1110000 1011011 1001000 1100010 0001011 0010111 1001101 1110000 1101110

4. A 100 Gbps, um bit leva 10^{-11} segundos para ser transmitido. Com a velocidade da luz igual a 2×10^8 m/s, no tempo de 1 bit, o pulso de luz alcança um comprimento de 2 mm ou 2.000 micra. Tendo em vista que um fóton tem mais ou menos 1 mícron de comprimento, o pulso tem o comprimento de 2.000 fótons. Desse modo, não estamos nem perto de um único fóton por bit, até mesmo a 100 Gbps. Somente a 200 Tbps conseguimos alcançar 1 bit por fóton.

5. Durante metade do tempo Trudy fará palpites corretos. Todos esses bits serão regenerados de forma adequada. Na outra metade do tempo, ela errará em seus palpites e enviará bits aleatórios a Bob. Metade desses bits estará errada. Desse modo, 25% dos bits que ela transmitir pela fibra estarão errados. As cifras de uso único (one-time pads) de Bob estarão portanto 75% corretas e 25% erradas.

6. Se o intruso tivesse capacidade de computação infinita, as duas situações seriam idênticas mas, considerando-se que isso não acontece, a segunda alternativa é melhor. Ela obriga o intruso a executar uma computação para verificar se cada chave experimentada é correta. Se essa computação for dispendiosa, ela irá diminuir a velocidade do intruso.

7. Sim. Uma seqüência contígua de caixas P pode ser substituída por uma única caixa P. Ocorre algo semelhante no caso das caixas S.

8. Para cada chave de 56 bits possível descriptografe o primeiro bloco de texto cifrado. Se o texto simples resultante for válido, experimente o próximo bloco etc. Se o texto simples for inválido, experimente a próxima chave.

9. A equação $2^n = 10^{15}$ nos informa n, o número de períodos de duplicação necessários. Resolvendo-a, obtemos $n = 15 \log_2 10$ ou $n = 50$ períodos de duplicação, que correspondem a 75 anos. A simples construção dessa máquina é uma possibilidade distante, e talvez a Lei de Moore não venha a durar por mais 75 anos.

10. A equação que precisamos resolver é $2^{256} - 10^n$. Usando logaritmos comuns, obtemos $n = 256 \log_2$, e então $n = 77$. O número de chaves é portanto 10^{77}. O número de estrelas em nossa galáxia é aproximadamente

10^{12}, e o número de galáxias é cerca de 10^8; portanto, há mais ou menos 10^{20} estrelas no universo. A massa do sol, uma estrela típica, é 2×10^{33} gramas. O sol é constituído principalmente de hidrogênio, e o número de átomos em 1 grama de hidrogênio é cerca de 6×10^{23} (o número de Avogadro). Desse modo, o número de átomos no sol é aproximadamente 1,2 $\times 10^{57}$. Com 10^{20} estrelas, o número de átomos em todas as estrelas do universo é cerca de 10^{77}. Portanto, o número de chaves AES de 256 bits é igual ao número de átomos do universo inteiro (ignorando-se a matéria escura). Conclusão: a quebra do AES-256 por força bruta não é provável em qualquer momento próximo.

11. O DES mistura os bits quase inteiramente; assim, o único erro de bit no bloco C_i irá adulterar por completo o bloco P_i. Além disso, um bit estará errado no bloco P_{i+1}. Porém, todos os blocos de texto simples subseqüentes estarão corretos. Portanto, um único erro de bit só afeta dois blocos de texto simples.

12. Infelizmente, todo bloco de texto simples que começa em P_i+1 estará errado agora, pois todas as entradas para as caixas EXCLUSIVE OR estarão erradas. Portanto, um erro de enquadramento é muito mais sério que um bit invertido.

13. O encadeamento de blocos de cifras produz 8 bytes de saída por criptografia. O modo de feedback de cifra produz 1 byte de saída por criptografia. Desse modo, o encadeamento de blocos de cifras é oito vezes mais eficiente (isto é, com o mesmo número de ciclos, você pode criptografar oito vezes mais texto simples).

14. (a) Para esses parâmetros, $z = 60$, e assim devemos escolher d primo em relação a 60. São valores possíveis: 7, 11, 13, 17 e 19.
(b) Se e satisfaz à equação $7e = 1 \bmod 360$, então $7e$ deve ser 361, 721, 1.081, 1.441 etc. Dividindo-se cada um desses valores por 7 para ver qual deles é divisível por 7, descobrimos que $721/7 = 103$; conseqüentemente $e = 103$.
(c) Com esses parâmetros, $e = 3$. Para criptografar P, usamos a função $C = P^3 \bmod 55$. Para $P = 1$ a 10, $C = 1, 8, 27, 9, 15, 51, 13, 17, 14$ e 10, respectivamente.

15. Maria deve considerar a mudança de suas chaves. Isso deve ocorrer porque é relativamente fácil para Frances decifrar a chave privada de Maria, como a seguir. Frances sabe que a chave pública de Maria é (e 1, n 1). Frances nota que n 2 = n 1. Agora, Frances pode adivinhar a chave privada de Maria (d 1, n 1) simplesmente enumerando diferentes soluções da equação d 1 $\times$ e 1 = 1 *modn* 1.

16. Não. A segurança se baseia no fato de haver um algoritmo de criptografia forte e uma chave longa. O *IV* não é realmente essencial. O importante é a chave.

17. Os R_As da última mensagem ainda podem estar na RAM. Se isso se perder, Trudy poderá tentar reproduzir a mensagem mais recente para Bob, na esperança de que ele não veja que é uma duplicata. Uma solução é Bob gravar em disco o R_A de cada mensagem recebida *antes* de fazer o trabalho. Nesse caso, o ataque de reprodução não funcionará. Contudo, agora existe o perigo de, se uma solicitação for gravada em disco e logo em seguida houver uma pane, a solicitação nunca ser executada.

18. Se Trudy substituir ambas as partes, quando Bob aplicar a chave pública de Alice à assinatura, ele obterá algo que não é o sumário de mensagem do texto simples. Trudy poderá inserir uma mensagem falsa e será capaz de efetuar o hash, mas não conseguirá assiná-la com a chave privada de Alice.

19. Quando um cliente, digamos Sam, indica que deseja adquirir algum material pornográfico, jogos de azar ou qualquer outro item, a Máfia compra um diamante usando o número do cartão de crédito de Sam em uma joalheria. Quando a joalheria envia um contrato para ser assinado (talvez incluindo o número do cartão de crédito e uma caixa postal da Máfia como endereço), a Máfia encaminha o hash da mensagem da joalheria a Sam, juntamente com um contrato assinado por Sam como cliente de material pornográfico ou de jogos de azar. Se Sam simplesmente assinar sem prestar atenção ao fato de que o contrato e a assinatura não correspondem, a Máfia encaminhará a assinatura à joalheria, que então enviará o diamante aos mafiosos. Se mais tarde Sam afirmar que não comprou nenhum diamante, a joalheria será capaz de produzir um contrato assinado mostrando que ele o fez.

20. Com 20 alunos, existem $(20 \times 19)/2 = 190$ pares de alunos. A probabilidade de que os alunos de qualquer par façam aniversário no mesmo dia é 1/365, e a probabilidade de que eles tenham dias de aniversário diferentes é 364/365. A probabilidade de que todos os 190 pares façam aniversário em dias diferentes é portanto $(364/365)^{190}$. Esse número é próximo de 0,594. Se a probabilidade de que todos os pares tenham dias de aniversário diferentes é 0,594, então a probabilidade de que um ou mais pares tenham o mesmo dia de aniversário é cerca de 0,406.

21. A secretária pode escolher algum número (por exemplo, 32) de espaços na carta e potencialmente substituir cada um por espaço, retrocesso, espaço. Quando vistas no terminal, todas as variações parecerão semelhantes, mas todas terão diferentes resumos de mensagens (message digests), e assim o ataque de aniversário ainda funcionará. Como alternativa, adicione espaços ao final das linhas e troque os espaços por tabulações.

22. É realizável. Alice codifica um nonce com a chave compartilhada e o envia a Bob, que envia de volta uma mensagem criptografada com a chave compartilhada contendo o nonce, seu próprio nonce e a chave pública. Trudy não

pode forjar essa mensagem e, se enviar lixo ao acaso, quando ele for descriptografado, não conterá o nonce de Alice. Para completar o protocolo, Alice envia de volta o nonce de Bob codificado com a chave pública de Bob.

23. A etapa 1 é verificar o certificado X.509 usando a chave pública do CA raiz. Se for genuína, ela terá agora a chave pública de Bob, embora deva verificar o CRL, se houver um. Porém, para saber se é Bob quem está na outra extremidade da conexão, ela precisa saber se Bob tem a chave privada correspondente. Alice escolhe um nonce e o envia a Bob com sua chave pública. Se Bob conseguir enviá-lo de volta em texto simples, ela ficará convencida de que é mesmo Bob.

24. Primeiro, Alice estabelece um canal de comunicação com *X* e pede a *X* um certificado para verificar sua chave pública. Suponha que *X* forneça um certificado assinado por outro CA *Y*. Se Alice não conhecer *Y*, ela repetirá a etapa anterior com *Y*. Alice continuará a fazer isso até receber um certificado confirmando a chave pública de um CA *Z* assinada por *A* e reconhecer a chave pública de *A*. Observe que isso pode continuar até ser alcançada uma raiz, ou seja, *A* é a raiz. Depois disso, Alice verifica as chaves públicas em ordem inversa, a partir do certificado que *Z* forneceu. Em cada etapa durante a verificação, ela também confere para ter certeza de que o certificado fornecido não foi revogado. Finalmente, depois de verificar a chave pública de Bob, Alice assegura que está de fato se comunicando com Bob, utilizando o mesmo método do problema anterior.

25. Não. AH em modo de transporte inclui o cabeçalho IP no total de verificação. A caixa NAT muda o endereço de origem, arruinando o total de verificação. Todos os pacotes serão percebidos como pacotes contendo erros.

26. HMACs são muito mais rápidos em termos computacionais.

27. O tráfego de entrada poderia ser inspecionado para verificar a presença de vírus. O tráfego de partida poderia ser inspecionado para verificar se estão vazando informações confidenciais da empresa. A verificação de vírus deve funcionar, se for utilizado um bom programa antivírus. A verificação do tráfego de saída, que pode estar codificado, é praticamente inútil diante de uma tentativa séria de vazar informações.

28. Se não quiser revelar a ninguém com quem está se comunicando (incluindo seu próprio administrador de sistema), Jim precisará utilizar mecanismos de segurança adicionais. Lembre-se de que a VPN só oferece segurança para comunicações pela Internet (fora da organização). Ela não fornece qualquer segurança para comunicação dentro da organização. Se Jim só quiser manter sua comunicação protegida contra pessoas de fora da empresa, uma VPN será suficiente.

29. Sim. Suponha que Trudy efetue o XOR de uma palavra aleatória com o início da carga útil, e então efetue o XOR da mesma palavra com o total de verificação. O total de verificação ainda será correto. Desse modo, Trudy será capaz de adulterar mensagens e essas adulterações não serão detectadas, porque ela pode manipular o total de verificação usando criptografia.

30. Na mensagem 2, coloque *R*B dentro da mensagem criptografada, e não fora dela. Desse modo, Trudy não será capaz de descobrir R_B e o ataque por reflexão não funcionará.

31. Bob sabe que $g^x \bmod n = 191$. Ele calcula $191^{15} \bmod 719 = 40$. Alice sabe que $g^y \bmod n = 543$. Ela calcula $543^{16} \bmod n = 40$. A chave é 40. O modo mais simples de efetuar os cálculos anteriores é usar o programa *bc* do UNIX.

32. Não existe nada que Bob saiba que Trudy não saiba. Qualquer resposta que Bob pode dar, Trudy também pode dar. Sob essas circunstâncias, é impossível para Alice saber se está se comunicando com Bob ou com Trudy.

33. O KDC precisa ter algum meio de saber quem enviou a mensagem, e conseqüentemente qual chave de descriptografia deve aplicar a ela.

34. Não. Trudy só precisa capturar duas mensagens de ou para o mesmo usuário. Ela poderá então descriptografar ambas com a mesma chave. Se o campo de número aleatório nas duas mensagens for o mesmo, isso significa que ela tem a chave correta. Tudo que esse esquema faz é duplicar sua carga de trabalho.

35. Os dois números aleatórios são usados para diferentes propósitos. *R*A é usado para convencer Alice de que ela está se comunicando com o KDC. R_{A2} é usado para convencer Alice de que ela estará se comunicando com Bob mais tarde. Ambos são necessários.

36. Se o AS cair, novos usuários legítimos não serão capazes de se autenticar, ou seja, obter um ingresso TGS. Assim, eles não poderão acessar qualquer servidor na organização. Os usuários que já têm um ingresso TGS (obtido de AS antes da queda) poderão continuar a acessar os servidores até expirar a vigência de seu ingresso TGS. Se o TGS ficar inativo, somente os usuários que já têm um ingresso do servidor (obtido do TGS antes da queda) referente a um servidor S terão a possibilidade de acessar S até expirar a vigência do seu ingresso de servidor. Em ambos os casos, não ocorrerá nenhuma violação de segurança.

37. Não é essencial enviar *R*B criptografado. Trudy não tem como saber disso, e ele não será usado de novo; portanto, não é realmente secreto. Por outro lado, fazer isso desse modo permite experimentar K_S, a fim de se ter abso-

luta certeza de que está tudo correto antes de enviar os dados. Além disso, por que dar a Trudy informações gratuitas sobre o gerador de números aleatórios de Bob? Em geral, quanto menos for enviado em texto simples melhor; e como nesse caso o custo é muito baixo, Alice também poderia criptografar R_B.

38. O banco envia um desafio (um longo número aleatório) ao computador do comerciante, que então fornece esse número ao cartão. A CPU no cartão transforma o número de uma forma complexa que depende do código PIN digitado diretamente no cartão. O resultado dessa transformação é dado ao computador do comerciante para transmitir ao banco. Se o comerciante ligar para o banco novamente para executar outra transação, o banco enviará um novo desafio, e assim o conhecimento total do antigo será inútil. Mesmo que conheça o algoritmo usado pelos cartões inteligentes, o comerciante não conhece o código PIN do cliente, pois ele foi digitado diretamente no cartão. O visor no cartão é necessário para evitar que o comerciante mostre: "o preço da compra é 49,95", mas informe ao banco que o valor é 499,95.

39. A compactação economiza largura de banda; porém, muito mais importante, ela também elimina as informações de freqüência contidas no texto simples (por exemplo, que "e" é a letra mais comum no texto em inglês). Na realidade, ela converte o texto simples em texto sem sentido, aumentando o trabalho do criptoanalista para quebrar o código da mensagem.

40. Não. Suponha que o endereço fosse uma lista de debate. Cada pessoa teria sua própria chave pública. Criptografar a chave IDEA com apenas uma chave pública não funcionaria. Ela teria de ser criptografada com várias chaves públicas.

41. Na etapa 3, o ISP solicita o endereço *www.trudy-a-intrusa.com* e ele nunca é fornecido. Seria melhor fornecer o endereço IP para ficar menos evidente. O resultado deve ser marcado como impossível de armazenar na cache, de forma que o truque possa ser usado mais tarde, se necessário.

42. O código DNS é público, e assim o algoritmo usado para a geração de ID é público. Se for um gerador de números aleatórios, o uso de IDs ao acaso dificilmente ajudará. Utilizando o mesmo ataque de spoofing mostrado no texto, Trudy pode aprender a ID atual (aleatória). Tendo em vista que os geradores de números aleatórios são completamente determinísticos, se Trudy conhecer uma única ID, poderá calcular facilmente a próxima. Se o número aleatório gerado pelo algoritmo for submetido a uma operação XOR com a hora, isso o tornará menos previsível, a não ser pelo fato de que Trudy também conhece a hora. É muito melhor efetuar a operação XOR entre o número aleatório e a hora, e também entre ele e o número de

pesquisas que o servidor executou no minuto anterior (algo que Trudy não conhece) e depois obter o hash SHA-1 desse resultado. A dificuldade aqui é que o SHA-1 demora um período de tempo não desprezível, e o DNS tem de ser rápido.

43. Os nonces protegem contra ataques de repetição. Tendo em vista que cada partido contribui para a chave, se um intruso tentar reproduzir mensagens antigas, a nova chave não coincidirá com a antiga.

44. Fácil. A música é simplesmente um arquivo. Não importa o que está no arquivo. Existe espaço para 294.912 bytes nos bits de baixa ordem. O MP3 exige aproximadamente 1 MB por minuto, e assim cerca de 18 segundos de música poderiam ser armazenados.

45. Alice poderia efetuar o hash de cada mensagem e assiná-la com sua chave privada. Em seguida, poderia acrescentar o hash assinado e sua chave pública à mensagem. As pessoas poderiam verificar a assinatura e comparar a chave pública à que Alice usou da última vez. Se Trudy tentasse passar por Alice e acrescentasse a chave pública de Alice, não seria capaz de obter o hash correto. Se usasse sua própria chave pública, as pessoas veriam que a chave não era a mesma empregada anteriormente.