EXERCÍCIOS DE FIXAÇÃO E PERGUNTAS

Questões de revisão do Capítulo 3

SEÇÕES 3.1–3.3

R1. Suponha que uma camada de rede forneça o seguinte serviço. A camada de rede no computador-fonte aceita um segmento de tamanho máximo de 1.200 bytes e um endereço de computador-alvo da camada de transporte. A camada de rede, então, garante encaminhar o segmento para a camada de transporte no computador-alvo. Suponha que muitos processos de aplicação de rede possam estar sendo executados no hospedeiro de destino.

a. Crie, da forma mais simples, o protocolo da camada de transporte possível que levará os dados da aplicação para o processo desejado no hospedeiro de destino. Suponha que o sistema operacional do hospedeiro de destino determinou um número de porta de 4 bytes para cada processo de aplicação em execução.

b. Modifique esse protocolo para que ele forneça um "endereço de retorno" ao processo-alvo.

c. Em seus protocolos, a camada de transporte "tem de fazer algo" no núcleo da rede de computadores?

R2. Considere um planeta onde todos possuam uma família com seis membros, cada família viva em sua própria casa, cada casa possua um endereço único e cada pessoa em certa casa possua um único nome. Suponha que esse planeta possua um serviço postal que entregue cartas da casa-fonte à casa-alvo. O serviço exige que (1) a carta esteja em um envelope e que (2) o endereço da casa-alvo (e nada mais) esteja escrito claramente no envelope. Imagine que cada família possua um membro representante que recebe e distribui cartas para as demais. As cartas não apresentam necessariamente qualquer indicação dos destinatários das cartas.

a. Utilizando a solução do Problema R1 como inspiração, descreva um protocolo que os representantes possam utilizar para entregar cartas de um membro remetente de uma família para um membro destinatário de outra família.

b. Em seu protocolo, o serviço postal precisa abrir o envelope e verificar a carta para fornecer o serviço?

R3. Considere uma conexão TCP entre o hospedeiro A e o hospedeiro B. Suponha que os segmentos TCP que trafegam do hospedeiro A para o hospedeiro B tenham número de porta da origem x e número de porta do destino y. Quais são os números de porta da origem e do destino para os segmentos que trafegam do hospedeiro B para o hospedeiro A?

R4. Descreva por que um desenvolvedor de aplicação pode escolher rodar uma aplicação sobre UDP em vez de sobre TCP.

R5. Por que o tráfego de voz e de vídeo é frequentemente enviado por meio do UDP e não do TCP na Internet de hoje? (*Dica:* A resposta que procuramos não tem nenhuma relação com o mecanismo de controle de congestionamento no TCP.)

R6. É possível que uma aplicação desfrute de transferência confiável de dados mesmo quando roda sobre UDP? Caso a resposta seja afirmativa, como isso acontece?

R7. Suponha que um processo no hospedeiro C possua um *socket* UDP com número de porta 6789 e que o hospedeiro A e o hospedeiro B, individualmente, enviem um segmento UDP ao hospedeiro C com número de porta de destino 6789. Os dois segmentos serão encaminhados para o mesmo *socket* no hospedeiro C? Se sim, como o processo no hospedeiro C saberá que os dois segmentos vieram de dois hospedeiros diferentes?

R8. Suponha que um servidor da Web seja executado no computador C na porta 80. Esse servidor utiliza conexões contínuas e, no momento, está recebendo solicitações de dois computadores diferentes, A e B. Todas as solicitações estão sendo enviadas por meio do mesmo *socket* no computador C? Se estão passando por diferentes *sockets*, dois deles possuem porta 80? Discuta e explique.

SEÇÃO 3.4

R9. Em nossos protocolos `rdt`, por que precisamos introduzir números de sequência?

R10. Em nossos protocolos `rdt`, por que precisamos introduzir temporizadores?

R11. Suponha que o atraso de viagem de ida e volta entre o emissor e o receptor seja constante e conhecido para o emissor. Ainda seria necessário um temporizador no protocolo `rdt 3.0`, supondo que os pacotes podem ser perdidos? Explique.

R12. Visite o applet Go-Back-N Java no site de apoio do livro.

a. A origem enviou cinco pacotes e depois interrompeu a animação antes que qualquer um dos cinco pacotes chegasse ao destino. Então, elimine o primeiro pacote e reinicie a animação. Descreva o que acontece.

b. Repita o experimento, mas agora deixe o primeiro pacote chegar ao destino e elimine o primeiro reconhecimento. Descreva novamente o que acontece.

c. Por fim, tente enviar seis pacotes. O que acontece?

R13. Repita a Questão 12, mas agora com o applet Java Selective Repeat. O que difere o Selective Repeat do Go-Back-N?

SEÇÃO 3.5

R14. Falso ou verdadeiro?

a. O hospedeiro A está enviando ao hospedeiro B um arquivo grande por uma conexão TCP. Suponha que o hospedeiro B não tenha dados para enviar para o hospedeiro A. O hospedeiro B não enviará reconhecimentos para A porque ele não pode dar carona aos reconhecimentos dos dados.

b. O tamanho de `rwnd` do TCP nunca muda enquanto dura a conexão.

c. Suponha que o hospedeiro A esteja enviando ao hospedeiro B um arquivo grande por uma conexão TCP. O número de bytes não reconhecidos que o hospedeiro A envia não pode exceder o tamanho do buffer de recepção.

d. Imagine que o hospedeiro A esteja enviando ao hospedeiro B um arquivo grande por uma conexão TCP. Se o número de sequência para um segmento dessa conexão for m, então o número de sequência para o segmento subsequente será necessariamente $m + 1$.

e. O segmento TCP tem um campo em seu cabeçalho para `rwnd`.

f. Suponha que o último `SampleRTT` de uma conexão TCP seja igual a 1 s. Então, o valor corrente de `TimeoutInterval` para a conexão será necessariamente ajustado para um valor ≥ 1 s.

g. Imagine que o hospedeiro A envie ao hospedeiro B, por uma conexão TCP, um segmento com o número de sequência 38 e 4 bytes de dados. Nesse mesmo segmento, o número de reconhecimento será necessariamente 42.

R15. Suponha que o hospedeiro A envie dois segmentos TCP um atrás do outro ao hospedeiro B sobre uma conexão TCP. O primeiro segmento tem número de sequência 90 e o segundo, número de sequência 110.

a. Quantos dados tem o primeiro segmento?

b. Suponha que o primeiro segmento seja perdido, mas o segundo chegue a B. No reconhecimento que B envia a A, qual será o número de reconhecimento?

R16. Considere o exemplo do Telnet discutido na Seção 3.5. Alguns segundos após o usuário digitar a letra "C", ele digitará a letra "R". Depois disso, quantos segmentos serão enviados e o que será colocado nos campos de número de sequência e de reconhecimento dos segmentos?

SEÇÃO 3.7

R17. Suponha que duas conexões TCP estejam presentes em algum enlace congestionado de velocidade R bits/s. As duas conexões têm um arquivo imenso para enviar (na mesma direção, pelo enlace congestionado).

As transmissões dos arquivos começam exatamente ao mesmo tempo. Qual é a velocidade de transmissão que o TCP gostaria de dar a cada uma das conexões?

R18. Verdadeiro ou falso: considere o controle de congestionamento no TCP. Quando um temporizador expira no remetente, o valor de `ssthresh` é ajustado para a metade de seu valor anterior.

R19. Na discussão sobre divisão do TCP, na nota em destaque da Seção 3.7, afirmamos que o tempo de resposta com a divisão do TCP é aproximadamente $4 \cdot RTT_{FE} + RTT_{BE}$ + tempo de processamento. Justifique essa afirmação.

PROBLEMAS

P1. Suponha que o cliente A inicie uma sessão Telnet com o servidor S. Quase ao mesmo tempo, o cliente B também inicia uma sessão Telnet com o servidor S. Forneça possíveis números de porta da fonte e do destino para:

a. Os segmentos enviados de A para S.

b. Os segmentos enviados de B para S.

c. Os segmentos enviados de S para A.

d. Os segmentos enviados de A para B.

e. Se A e B são hospedeiros diferentes, é possível que o número de porta da fonte nos segmentos de A para S seja o mesmo que nos de B para S?

f. E se forem o mesmo hospedeiro?

P2. Considere a Figura 3.5. Quais são os valores da porta de fonte e da porta de destino nos segmentos que fluem do servidor de volta aos processos clientes? Quais são os endereços IP nos datagramas de camada de rede que carregam os segmentos de camada de transporte?

P3. O UDP e o TCP usam complementos de 1 para suas somas de verificação. Suponha que você tenha as seguintes três palavras de 8 bits: 01010011, 01100110 e 01110100. Qual é o complemento de 1 para as somas dessas palavras? (Note que, embora o UDP e o TCP usem palavras de 16 bits no cálculo da soma de verificação, nesse problema solicitamos que você considere somas de 8 bits.) Mostre todo o trabalho. Por que o UDP toma o complemento de 1 da soma, isto é, por que não toma apenas a soma? Com o esquema de complemento de 1, como o destinatário detecta erros? É possível que um erro de 1 bit passe despercebido? E um erro de 2 bits?

P4 a. Suponha que você tenha os seguintes bytes: 01011100 e 01100101. Qual é o complemento de 1 da soma desses 2 bytes?

b. Suponha que você tenha os seguintes bytes: 11011010 e 01100101. Qual é o complemento de 1 da soma desses 2 bytes?

c. Para os bytes do item (a), dê um exemplo em que um bit é invertido em cada um dos 2 bytes e, mesmo assim, o complemento de um não muda.

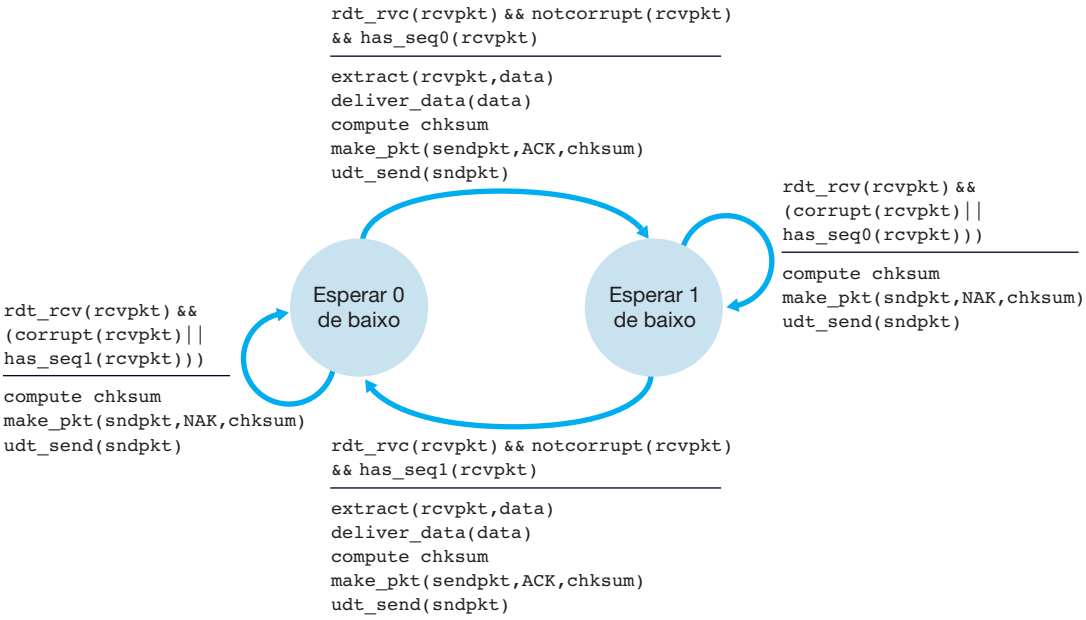
P5. Suponha que o receptor UDP calcule a soma de verificação da Internet para o segmento UDP recebido e encontre que essa soma coincide com o valor transportado no campo da soma de verificação. O receptor pode estar absolutamente certo de que não ocorreu nenhum erro de bit? Explique.

P6. Considere nosso motivo para corrigir o protocolo `rtd2.1`. Demonstre que o destinatário apresentado na Figura 3.57, quando em operação com o remetente mostrado na Figura 3.11, pode levar remetente e destinatário a entrar em estado de travamento, em que cada um espera por um evento que nunca vai ocorrer.

P7. No protocolo `rdt3.0`, os pacotes ACK que fluem do destinatário ao remetente não têm números de sequência (embora tenham um campo de ACK que contém o número de sequência do pacote que estão reconhecendo). Por que nossos pacotes ACK não requerem números de sequência?

P8. Elabore a FSM para o lado destinatário do protocolo `rdt3.0`.

FIGURA 3.57 UM RECEPTOR INCORRETO PARA O PROTOCOLO `rdt 2.1`.



P9. Elabore um diagrama de mensagens para a operação do protocolo `rdt3.0` quando pacotes de dados e de reconhecimento estão truncados. Seu diagrama deve ser semelhante ao usado na Figura 3.16.

P10. Considere um canal que pode perder pacotes, mas cujo atraso máximo é conhecido. Modifique o protocolo `rdt2.1` para incluir esgotamento de temporização do remetente e retransmissão. Informalmente, argumente por que seu protocolo pode se comunicar de modo correto por esse canal.

P11. Considere o `rdt2.2` destinatário da Figura 3.14 e a criação de um novo pacote na autotransição (isto é, a transição do estado de volta para si mesmo) nos estados "Esperar 0 de baixo" e "Esperar 1 de baixo": `sndpkt=make_pkt(ACK,0,checksum)` e `sndpkt=make_pkt(ACK,0, checksum)`. O protocolo funcionaria corretamente se essa ação fosse removida da autotransição no estado "Esperar 1 de baixo"? Justifique sua resposta. E se esse evento fosse removido da autotransição no estado "Esperar 0 de baixo"? [*Dica:* Neste último caso, considere o que aconteceria se o primeiro pacote do remetente ao destinatário fosse corrompido.]

P12. O lado remetente do `rdt3.0` simplesmente ignora (isto é, não realiza nenhuma ação) todos os pacotes recebidos que estão errados ou com o valor errado no campo `acknum` de um pacote de reconhecimento. Suponha que em tais circunstâncias o `rdt3.0` devesse apenas retransmitir o pacote de dados corrente. Nesse caso, o protocolo ainda funcionaria? (*Dica*: Considere o que aconteceria se houvesse apenas erros de bits; não há perdas de pacotes, mas podem ocorrer esgotamentos de temporização prematuros. Imagine quantas vezes o *enésimo* pacote é enviado, no limite em que *n* se aproximasse do infinito.)

P13. Considere o protocolo `rdt3.0`. Elabore um diagrama mostrando que, se a conexão de rede entre o remetente e o destinatário puder alterar a ordem de mensagens (isto é, se for possível reordenar duas mensagens que se propagam no meio entre o remetente e o destinatário), então o protocolo bit alternante não funcionará corretamente (lembre-se de identificar com clareza o sentido no qual o protocolo não funcionará de modo correto). Seu diagrama deve mostrar o remetente à esquerda e o destinatário à direita; o eixo do tempo deverá estar orientado de cima para baixo na página e mostrar a troca de mensagem de dados (D) e de reconhecimento (A). Não se esqueça de indicar o número de sequência associado com qualquer segmento de dados ou de reconhecimento.

P14. Considere um protocolo de transferência confiável de dados que use somente reconhecimentos negativos. Suponha que o remetente envie dados com pouca frequência. Um protocolo que utiliza somente NAKs seria preferível a um protocolo que utiliza ACKs? Por quê? Agora suponha que o remetente tenha uma grande

quantidade de dados para enviar e que a conexão fim a fim sofra poucas perdas. Nesse segundo caso, um protocolo que utilize somente NAKs seria preferível a um protocolo que utilize ACKs? Por quê?

P15. Considere o exemplo em que se atravessa os Estados Unidos mostrado na Figura 3.17. Que tamanho deveria ter a janela para que a utilização do canal fosse maior do que 98%? Suponha que o tamanho de um pacote seja 1.500 bytes, incluindo os campos do cabeçalho e os dados.

P16. Suponha que uma aplicação utilize `rdt3.0` como seu protocolo da camada de transporte. Como o protocolo pare e espere possui uma utilização do canal muito baixa (mostrada no exemplo de travessia dos Estados Unidos), os criadores dessa aplicação permitem que o receptor continue enviando de volta um número (mais do que dois) de ACK 0 alternado e ACK 1, mesmo que os dados correspondentes não cheguem ao receptor. O projeto dessa aplicação aumentaria a utilização do canal? Por quê? Há possíveis problemas com esse método? Explique.

P17. Considere duas entidades de rede, A e B, que estão conectadas por um canal bidirecional perfeito (isto é, qualquer mensagem enviada será recebida corretamente; o canal não corromperá, perderá nem reordenará pacotes). A e B devem entregar mensagens de dados entre si de forma alternada: primeiro, A deve entregar uma mensagem a B, depois B deve entregar uma mensagem a A, e assim por diante. Se uma entidade estiver em um estado onde não deve tentar entregar uma mensagem ao outro lado e houver um evento como a chamada `rdt_send(data)` de cima que tente transmitir dados para baixo, para o outro lado, essa chamada de cima pode apenas ser ignorada com uma chamada a `rdt_unable_to_send(data)`, que informa à camada de cima que atualmente não é possível enviar dados. [*Nota*: Essa suposição simplificada é para que você não tenha de se preocupar com o armazenamento de dados em buffer.]
Elabore uma especificação FSM para este protocolo (uma FSM para A e uma para B!). Observe que você não precisa se preocupar com um mecanismo de confiabilidade aqui; o ponto importante da questão é criar uma especificação FSM que reflita o comportamento sincronizado das duas entidades. Você deverá usar os seguintes eventos e ações que possuem o mesmo significado do protocolo `rdt1.0` da Figura 3.9: `rdt_send(data)`, `packet = make_pkt(data)`, `udt_send(packet)`, `rdt_rcv(packet)`, `extract (packet,data)`, `deliver_data(data)`. Cuide para que o protocolo reflita a alternância estrita de envio entre A e B. Além disso, não se esqueça de indicar os estados iniciais de A e B em suas especificações FSM.

P18. No protocolo genérico SR que estudamos na Seção 3.4.4, o remetente transmite uma mensagem assim que ela está disponível (se ela estiver na janela), sem esperar por um reconhecimento. Suponha, agora, que queiramos um protocolo SR que envie duas mensagens de cada vez. Isto é, o remetente enviará um par de mensagens, e o par de mensagens subsequente somente deverá ser enviado quando o remetente souber que ambas as mensagens do primeiro par foram recebidas corretamente.
Suponha que o canal possa perder mensagens, mas que não as corromperá nem as reordenará. Elabore um protocolo de controle de erro para a transferência confiável unidirecional de mensagens. Dê uma descrição FSM do remetente e do destinatário. Descreva o formato dos pacotes enviados entre o remetente e o destinatário e vice-versa. Se você usar quaisquer procedimentos de chamada que não sejam os da Seção 3.4 (por exemplo, `udt_send()`, `start_timer()`, `rdt_rcv()` etc.), esclareça as ações desses procedimentos. Dê um exemplo (um diagrama de mensagens para o remetente e para o destinatário) mostrando como seu protocolo se recupera de uma perda de pacote.

P19. Considere um cenário em que o hospedeiro A queira enviar pacotes para os hospedeiros B e C simultaneamente. O hospedeiro A está conectado a B e a C por um canal *broadcast* — um pacote enviado por A é levado pelo canal a B e a C. Suponha que o canal *broadcast* que conecta A, B e C possa, de modo independente, perder e corromper mensagens (e assim, por exemplo, uma mensagem enviada de A poderia ser recebida corretamente por B, mas não por C). Projete um protocolo de controle de erro do tipo pare e espere para a transferência confiável de um pacote de A para B e para C, tal que A não receba novos dados da camada superior até que saiba que B e C receberam corretamente o pacote em questão. Dê descrições FSM de A e C. (*Dica*: a FSM para B deve ser a mesma que para C.) Também dê uma descrição do(s) formato(s) de pacote usado(s).

P20. Considere um cenário em que os hospedeiros A e B queiram enviar mensagens ao hospedeiro C. Os hospedeiros A e C estão conectados por um canal que pode perder e corromper (e não reordenar) mensagens. B e C estão conectados por outro canal (independente do canal que conecta A e C) com as mesmas propriedades.

A camada de transporte no hospedeiro C deve alternar o envio de mensagens de A e B para a camada acima (ou seja, ela deve primeiro transmitir os dados de um pacote de A e depois os dados de um pacote de B, e assim por diante). Elabore um protocolo de controle de erro pare e espere para transferência confiável de pacotes de A e B para C, com envio alternado em C, como descrito acima. Dê descrições FSM de A e C. (*Dica:* a FSM para B deve ser basicamente a mesma de A.) Dê, também, uma descrição do(s) formato(s) de pacote utilizado(s).

P21. Suponha que haja duas entidades de rede A e B e que B tenha um suprimento de mensagens de dados que será enviado a A de acordo com as seguintes convenções: quando A recebe uma solicitação da camada superior para obter a mensagem de dados (D) seguinte de B, A deve enviar uma mensagem de requisição (R) para B no canal A-a-B; somente quando B receber uma mensagem R, ele poderá enviar uma mensagem de dados (D) de volta a A pelo canal B a A; A deve entregar uma cópia de cada mensagem D à camada superior; mensagens R podem ser perdidas (mas não corrompidas) no canal A-a-B; mensagens (D), uma vez enviadas, são sempre entregues corretamente; o atraso entre ambos os canais é desconhecido e variável.
Elabore um protocolo (dê uma descrição FSM) que incorpore os mecanismos apropriados para compensar a propensão à perda do canal A a B e implemente passagem de mensagem para a camada superior na entidade A, como discutido antes. Utilize apenas os mecanismos absolutamente necessários.

P22. Considere o protocolo GBN com um tamanho de janela remetente de 4 e uma faixa de números de sequência de 1.024. Suponha que, no tempo *t*, o pacote seguinte na ordem, pelo qual o destinatário está esperando, tenha um número de sequência *k*. Admita que o meio não reordene as mensagens. Responda às seguintes perguntas:

a. Quais são os possíveis conjuntos de números de sequência dentro da janela do remetente no tempo *t*? Justifique sua resposta.

b. Quais são todos os possíveis valores do campo ACK em todas as mensagens que estão atualmente se propagando de volta ao remetente no tempo *t*? Justifique sua resposta.

P23. Considere os protocolos GBN e SR. Suponha que o espaço de números de sequência seja de tamanho *k*. Qual será o maior tamanho de janela permissível que evitará que ocorram problemas como os da Figura 3.27 para cada um desses protocolos?

P24. Responda verdadeiro ou falso às seguintes perguntas e justifique resumidamente sua resposta:

a. Com o protocolo SR, é possível o remetente receber um ACK para um pacote que caia fora de sua janela corrente.

b. Com o GBN, é possível o remetente receber um ACK para um pacote que caia fora de sua janela corrente.

c. O protocolo bit alternante é o mesmo que o protocolo SR com janela do remetente e do destinatário de tamanho 1.

d. O protocolo bit alternante é o mesmo que o protocolo GBN com janela do remetente e do destinatário de tamanho 1.

P25. Dissemos que um aplicação pode escolher o UDP para um protocolo de transporte, pois oferece um controle de aplicações melhor (do que o TCP) de quais dados são enviados em um segmento e quando isso ocorre.

a. Por que uma aplicação possui mais controle de quais dados são enviados em um segmento?

b. Por que uma aplicação possui mais controle de quando o segmento é enviado?

P26. Considere a transferência de um arquivo enorme de *L* bytes do hospedeiro A para o hospedeiro B. Suponha um MSS de 536 bytes.

a. Qual é o máximo valor de *L* tal que não sejam esgotados os números de sequência TCP? Lembre-se de que o campo de número de sequência TCP tem 4 bytes.

b. Para o *L* que obtiver em (a), descubra quanto tempo demora para transmitir o arquivo. Admita que um total de 66 bytes de cabeçalho de transporte, de rede e de enlace de dados seja adicionado a cada segmento antes que o pacote resultante seja enviado por um enlace de 155 Mbits/s. Ignore controle de fluxo e controle de congestionamento de modo que A possa enviar os segmentos um atrás do outro e continuamente.

P27. Os hospedeiros A e B estão se comunicando por meio de uma conexão TCP, e o hospedeiro B já recebeu de A todos os bytes até o byte 126. Suponha que A envie, então, dois segmentos para B sucessivamente. O primeiro e o segundo segmentos contêm 80 e 40 bytes de dados. No primeiro segmento, o número de sequência é 127, o número de porta de partida é 302, e o número de porta de destino é 80. O hospedeiro B envia um reconhecimento ao receber um segmento do hospedeiro A.

a. No segundo segmento enviado do hospedeiro A para B, quais são o número de sequência, da porta de origem e da porta de destino?

b. Se o primeiro segmento chegar antes do segundo, no reconhecimento do primeiro segmento que chegar, qual é o número do reconhecimento, da porta de origem e da porta de destino?

c. Se o segundo segmento chegar antes do primeiro, no reconhecimento do primeiro segmento que chegar, qual é o número do reconhecimento?

d. Suponha que dois segmentos enviados por A cheguem em ordem a B. O primeiro reconhecimento é perdido e o segundo chega após o primeiro intervalo do esgotamento de temporização. Elabore um diagrama de temporização, mostrando esses segmentos, e todos os outros, e os reconhecimentos enviados. (Suponha que não haja qualquer perda de pacote adicional.) Para cada segmento de seu desenho, apresente o número de sequência e o número de bytes de dados; para cada reconhecimento adicionado por você, informe o número do reconhecimento.

P28. Os hospedeiros A e B estão diretamente conectados com um enlace de 100 Mbits/s. Existe uma conexão TCP entre os dois hospedeiros, e A está enviando a B um arquivo enorme por meio dessa conexão. O hospedeiro A pode enviar seus dados da aplicação para o *socket* TCP a uma taxa que chega a 120 Mbits/s, mas o hospedeiro B pode ler o buffer de recebimento TCP a uma taxa de 50 Mbits/s. Descreva o efeito do controle de fluxo do TCP.

P29. Os *cookies* SYN foram discutidos na Seção 3.5.6.

a. Por que é necessário que o servidor use um número de sequência especial no SYNACK?

b. Suponha que um atacante saiba que um hospedeiro-alvo utilize SYN *cookies*. O atacante consegue criar conexões semiabertas ou completamente abertas apenas enviando um pacote ACK para o alvo? Por quê?

c. Suponha que um atacante receba uma grande quantidade de números de sequência enviados pelo servidor. O atacante consegue fazer que o servidor crie muitas conexões totalmente abertas enviando ACKs com esses números de sequência? Por quê?

P30. Considere a rede mostrada no Cenário 2 na Seção 3.6.1. Suponha que os hospedeiros emissores A e B possuam valores de esgotamento de temporização fixos.

a. Analise o fato de que aumentar o tamanho do buffer finito do roteador pode possivelmente reduzir a vazão (λ_{out}).

b. Agora suponha que os hospedeiros ajustem dinamicamente seus valores de esgotamento de temporização (como o que o TCP faz) baseado no atraso no buffer no roteador. Aumentar o tamanho do buffer ajudaria a aumentar a vazão? Por quê?

P31. Suponha que os cinco valores de `SampleRTT` medidos (ver Seção 3.5.3) sejam 106 ms, 120 ms, 140 ms, 90 ms e 115 ms. Calcule o `EstimatedRTT` depois que forem obtidos cada um desses valores de `SampleRTT`, usando um valor de $\alpha = 0{,}125$ e supondo que o valor de `EstimatedRTT` seja 100 ms imediatamente antes que a primeira dessas cinco amostras seja obtida. Calcule também o `DevRTT` após a obtenção de cada amostra, considerando um valor de $\beta = 0{,}25$ e que o valor de `DevRTT` seja 5 ms imediatamente antes que a primeira dessas cinco amostras seja obtida. Por fim, calcule o `TimeoutInterval` do TCP após a obtenção de cada uma dessas amostras.

P32. Considere o procedimento do TCP para estimar o RTT. Suponha que $\alpha = 0{,}1$. Considere que $\texttt{SampleRTT}_1$ seja a amostra de RTT mais recente, $\texttt{SampleRTT}_2$ seja a próxima amostra mais recente, e assim por diante.

a. Para uma dada conexão TCP, suponha que quatro reconhecimentos foram devolvidos com as amostras RTT correspondentes $\texttt{SampleRTT}_4$, $\texttt{SampleRTT}_3$, $\texttt{SampleRTT}_2$ e $\texttt{SampleRTT}_1$. Expresse `EstimatedRTT` em termos das quatro amostras RTT.

b. Generalize sua fórmula para *n* amostras de RTTs.

c. Para a fórmula em (b), considere *n* tendendo ao infinito. Comente por que esse procedimento de média é denominado média móvel exponencial.

P33. Na Seção 3.5.3 discutimos estimativa de RTT para o TCP. Em sua opinião, por que o TCP evita medir o `SampleRTT` para segmentos retransmitidos?

P34. Qual é a relação entre a variável `SendBase` na Seção 3.5.4 e a variável `LastByteRcvd` na Seção 3.5.5?

P35. Qual é a relação entre a variável `LastByteRcvd` na Seção 3.5.5 e a variável y na Seção 3.5.4?

P36. Na Seção 3.5.4 vimos que o TCP espera até receber três ACKs duplicados antes de realizar uma retransmissão rápida. Em sua opinião, por que os projetistas do TCP preferiram não realizar uma retransmissão rápida após ser recebido o primeiro ACK duplicado para um segmento?

P37. Compare o GBN, SR e o TCP (sem ACK retardado). Admita que os valores do esgotamento de temporização para os três protocolos sejam longos o suficiente de tal modo que cinco segmentos de dados consecutivos e seus ACKs correspondentes possam ser recebidos (se não perdidos no canal) por um hospedeiro receptor (hospedeiro B) e por um hospedeiro emissor (hospedeiro A), respectivamente. Suponha que A envie cinco segmentos de dados para B, e que o segundo segmento (enviado de A) esteja perdido. No fim, todos os cinco segmentos de dados foram corretamente recebidos pelo hospedeiro B.

a. Quantos segmentos A enviou no total e quantos ACKs o hospedeiro B enviou no total? Quais são seus números de sequência? Responda essa questão para todos os três protocolos.

b. Se os valores do esgotamento de temporização para os três protocolos forem muito maiores do que 5 RTT, então qual protocolo envia com sucesso todos os cinco segmentos de dados em um menor intervalo de tempo?

P38. Em nossa descrição sobre o TCP na Figura 3.53, o valor do limiar, `ssthresh`, é definido como `ssthresh = cwnd/2` em diversos lugares e o valor `ssthresh` é referido como sendo definido para metade do tamanho da janela quando ocorreu um evento de perda. A taxa à qual o emissor está enviando quando ocorreu o evento de perda deve ser mais ou menos igual a segmentos `cwnd` por RTT? Explique sua resposta. Se for negativa, você pode sugerir uma maneira diferente pela qual `ssthresh` deva ser definido?

P39. Considere a Figura 3.46(b). Se λ'_{in} aumentar para mais do que $R/2$, λ_{out} poderá aumentar para mais do que $R/3$? Explique. Agora considere a Figura 3.46(c). Se λ'_{in} aumentar para mais do que $R/2$, λ_{out} poderá aumentar para mais de $R/4$ admitindo-se que um pacote será transmitido duas vezes, em média, do roteador para o destinatário? Explique.

P40. Considere a Figura 3.58. Admitindo-se que TCP Reno é o protocolo que experimenta o comportamento mostrado no gráfico, responda às seguintes perguntas. Em todos os casos você deverá apresentar uma justificativa resumida para sua resposta.

a. Quais os intervalos de tempo em que a partida lenta do TCP está em execução?

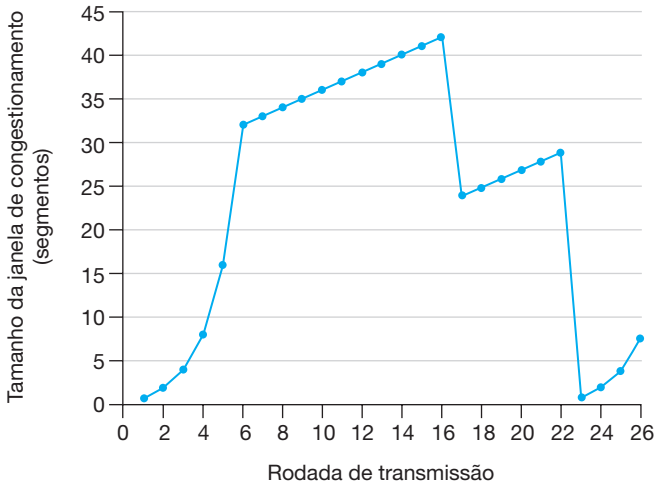
b. Quais os intervalos de tempo em que a prevenção de congestionamento do TCP está em execução?

c. Após a 16ª rodada de transmissão, a perda de segmento será detectada por três ACKs duplicados ou por um esgotamento de temporização?

d. Após a 22ª rodada de transmissão, a perda de segmento será detectada por três ACKs duplicados ou por um esgotamento de temporização?

e. Qual é o valor inicial de `ssthresh` na primeira rodada de transmissão?

FIGURA 3.58 TAMANHO DA JANELA TCP EM RELAÇÃO AO TEMPO



f. Qual é o valor inicial de `ssthresh` na 18ª rodada de transmissão?

g. Qual é o valor de `ssthresh` na 24ª rodada de transmissão?

h. Durante qual rodada de transmissão é enviado o 70º segmento?

i. Admitindo-se que uma perda de pacote será detectada após a 26ª rodada pelo recebimento de três ACKs duplicados, quais serão os valores do tamanho da janela de congestionamento e de `ssthresh`?

j. Suponha que o TCP Tahoe seja usado (em vez do TCP Reno) e que ACKs duplicados triplos sejam recebidos na 16ª rodada. Quais são o `ssthresh` e o tamanho da janela de congestionamento na 19ª rodada?

k. Suponha novamente que o TCP Tahoe seja usado, e que exista um evento de esgotamento de temporização na 22ª sessão. Quantos pacotes foram enviados da 17ª sessão até a 22ª, inclusive?

P41. Consulte a Figura 3.56, que ilustra a convergência do algoritmo AIMD do TCP. Suponha que, em vez de uma diminuição multiplicativa, o TCP reduza o tamanho da janela de uma quantidade constante. O AIMD resultante convergiria a um algoritmo de igual compartilhamento? Justifique sua resposta usando um diagrama semelhante ao da Figura 3.56.

P42. Na Seção 3.5.4 discutimos a duplicação do intervalo de temporização após um evento de esgotamento de temporização. Esse mecanismo é uma forma de controle de congestionamento. Por que o TCP precisa de um mecanismo de controle de congestionamento que utiliza janelas (como estudado na Seção 3.7) além desse mecanismo de duplicação do intervalo de esgotamento de temporização?

P43. O hospedeiro A está enviando um arquivo enorme ao hospedeiro B por uma conexão TCP. Nessa conexão nunca há perda de pacotes e os temporizadores nunca se esgotam. Seja R bits/s a taxa de transmissão do enlace que liga o hospedeiro A à Internet. Suponha que o processo em A consiga enviar dados para seu *socket* TCP a uma taxa de S bits/s, em que $S = 10 \cdot R$. Suponha ainda que o buffer de recepção do TCP seja grande o suficiente para conter o arquivo inteiro e que o buffer de envio possa conter apenas 1% do arquivo. O que impediria o hospedeiro A de passar dados continuamente para seu *socket* TCP à taxa de S bits/s: o controle de fluxo do TCP; o controle de congestionamento do TCP; ou alguma outra coisa? Elabore sua resposta.

P44. Considere enviar um arquivo grande de um computador a outro por meio de uma conexão TCP em que não haja perda.

a. Suponha que o TCP utilize AIMD para seu controle de congestionamento sem partida lenta. Admitindo que `cwnd` aumenta 1 MSS sempre que um lote de ACK é recebido e os tempos da viagem de ida e volta

constantes, quanto tempo leva para `cwnd` aumentar de 6 MSS para 12 MSS? (admitindo nenhum evento de perda)?

b. Qual é a vazão média (em termos de MSS e RTT) para essa conexão sendo o tempo = 6 RTT?

P45. Relembre a descrição macroscópica da vazão do TCP. No período de tempo transcorrido para a taxa da conexão variar de $W/(2 \cdot RTT)$ a W/RTT, apenas um pacote é perdido (bem ao final do período).

a. Mostre que a taxa de perda (fração de pacotes perdidos) é igual a

$$L = \text{taxa de perda} = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

b. Use o resultado anterior para mostrar que, se uma conexão tiver taxa de perda L, sua largura de banda média é dada aproximadamente por:

$$\approx \frac{1{,}22 \cdot MSS}{RTT\sqrt{L}}$$

P46. Considere que somente uma única conexão TCP (Reno) utiliza um enlace de 10 Mbits/s que não armazena nenhum dado. Suponha que esse enlace seja o único congestionado entre os hospedeiros emissor e receptor. Admita que o emissor TCP tenha um arquivo enorme para enviar ao receptor e o buffer de recebimento do receptor é muito maior do que a janela de congestionamento. Também fazemos as seguintes suposições: o tamanho de cada segmento TCP é 1.500 bytes; o atraso de propagação bidirecional dessa conexão é 150 ms; e essa conexão TCP está sempre na fase de prevenção de congestionamento, ou seja, ignore a partida lenta.

a. Qual é o tamanho máximo da janela (em segmentos) que a conexão TCP pode atingir?

b. Qual é o tamanho médio da janela (em segmentos) e a vazão média (em bits/s) dessa conexão TCP?

c. Quanto tempo essa conexão TCP leva para alcançar sua janela máxima novamente após se recuperar da perda de um pacote?

P47. Considere o cenário descrito na questão anterior. Suponha que o enlace de 10 Mbits/s possa armazenar um número finito de segmentos. Demonstre que para o enlace sempre enviar dados, teríamos que escolher um tamanho de buffer que é, pelo menos, o produto da velocidade do enlace C e o atraso de propagação bidirecional entre o emissor e o receptor.

P48. Repita a Questão 46, mas substituindo o enlace de 10 Mbits/s por um de 10 Gbits/s. Observe que em sua resposta ao item (c), verá que o tamanho da janela de congestionamento leva muito tempo para atingir seu tamanho máximo após se recuperar de uma perda de pacote. Elabore uma solução para resolver o problema.

P49. Suponha que T (medido por RTT) seja o intervalo de tempo que uma conexão TCP leva para aumentar seu tamanho de janela de congestionamento de $W/2$ para W, sendo W o tamanho máximo da janela de congestionamento. Demonstre que T é uma função da vazão média do TCP.

P50. Considere um algoritmo AIMD do TCP simplificado, sendo o tamanho da janela de congestionamento medido em número de segmentos e não em bytes. No aumento aditivo, o tamanho da janela de congestionamento aumenta por um segmento em cada RTT. Na diminuição multiplicativa, o tamanho da janela de congestionamento diminui para metade (se o resultado não for um número inteiro, arredondar para o número inteiro mais próximo). Suponha que duas conexões TCP, C_1 e C_2 compartilhem um único enlace congestionado com 30 segmentos por segundo de velocidade. Admita que C_1 e C_2 estejam na fase de prevenção de congestionamento. O RTT da conexão C_1 é de 100 ms e o da conexão C_2 é de 200 ms. Suponha que quando a taxa de dados no enlace excede a velocidade do enlace, todas as conexões TCP sofrem perda de segmento de dados.

a. Se C_1 e C_2 no tempo t_0 possui uma janela de congestionamento de 10 segmentos, quais são seus tamanhos de janela de congestionamento após 1.000 ms?

b. No final das contas, essas duas conexões obterão a mesma porção da largura de banda do enlace congestionado? Explique.

P51. Considere a rede descrita na questão anterior. Agora suponha que as duas conexões TCP, C_1 e C_2, possuam o mesmo RTT de 100 ms e que, no tempo t_0, o tamanho da janela de congestionamento de C_1 seja 15 segmentos, e que o de C_2 seja 10 segmentos.

a. Quais são os tamanhos de suas janelas de congestionamento após 2.200 ms?

b. No final das contas, essas duas conexões obterão a mesma porção da largura de banda do enlace congestionado?

c. Dizemos que duas conexões são sincronizadas se ambas atingirem o tamanho máximo e mínimo de janela ao mesmo tempo. No final das contas, essas duas conexões serão sincronizadas? Se sim, quais os tamanhos máximos de janela?

d. Essa sincronização ajudará a melhorar a utilização do enlace compartilhado? Por quê? Elabore alguma ideia para romper essa sincronização.

P52. Considere uma modificação ao algoritmo de controle de congestionamento do TCP. Em vez do aumento aditivo, podemos utilizar o aumento multiplicativo. Um emissor TCP aumenta seu tamanho de janela por uma constante positiva pequena a ($0 < a < 1$) ao receber um ACK válido. Encontre a relação funcional entre a taxa de perda L e a janela máxima de congestionamento W. Para esse TCP modificado, demonstre, independentemente da vazão média TCP, que uma conexão TCP sempre gasta a mesma quantidade de tempo para aumentar seu tamanho da janela de congestionamento de $W/2$ para W.

P53. Quando discutimos TCPs futuros na Seção 3.7, observamos que, para conseguir uma vazão de 10 Gbits/s, o TCP apenas poderia tolerar uma probabilidade de perda de segmentos de $2 \cdot 10^{-10}$ (ou, equivalentemente, uma perda para cada 5 milhões de segmentos). Mostre a derivação dos valores para $2 \cdot 10^{-10}$ (ou 1: 5.000.000) a partir dos valores de RTT e do MSS dados na Seção 3.7. Se o TCP precisasse suportar uma conexão de 100 Gbits/s, qual seria a perda tolerável?

P54. Quando discutimos controle de congestionamento em TCP na Seção 3.7, admitimos implicitamente que o remetente TCP sempre tinha dados para enviar. Agora considere o caso em que o remetente TCP envia uma grande quantidade de dados e então fica ocioso em t_1 (já que não há mais dados a enviar). O TCP permanecerá ocioso por um período de tempo relativamente longo e então irá querer enviar mais dados em t_2. Quais são as vantagens e desvantagens de o TCP utilizar os valores `cwnd` e `ssthresh` de t_1 quando começar a enviar dados em t_2? Que alternativa você recomendaria? Por quê?

P55. Neste problema, verificamos se o UDP ou o TCP apresentam um grau de autenticação do ponto de chegada.

a. Considere um servidor que receba uma solicitação dentro de um pacote UDP e responda a essa solicitação dentro de um pacote UDP (por exemplo, como feito por um servidor DNS). Se um cliente com endereço IP X o engana com o endereço Y, para onde o servidor enviará sua resposta?

b. Suponha que um servidor receba um SYN de endereço IP de origem Y, e depois de responder com um SYNACK, recebe um ACK com o endereço IP de origem Y com o número de reconhecimento correto. Admitindo que o servidor escolha um número de sequência aleatório e que não haja um "*man-in-the-middle*", o servidor pode ter certeza de que o cliente realmente está em Y (e não em outro endereço X que está se passando por Y)?

P56. Neste problema, consideramos o atraso apresentado pela fase de partida lenta do TCP. Considere um cliente e um servidor da Web diretamente conectados por um enlace de taxa R. Suponha que o cliente queira restaurar um objeto cujo tamanho seja exatamente igual a 15 S, sendo S o tamanho máximo do segmento (MSS). Considere RTT o tempo de viagem de ida e volta entre o cliente e o servidor (admitindo que seja constante). Ignorando os cabeçalhos do protocolo, determine o tempo para restaurar o objeto (incluindo o estabelecimento da conexão TCP) quando

a. $4\,S/R > S/R + RTT > 2S/R$

b. $S/R + RTT > 4\,S/R$

c. $S/R > RTT$.