Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Redes de Computadores I
AD2 - 1° semestre de 2013.
Gabarito

1. Afirmamos que uma aplicação pode escolher o UDP como protocolo da camada de transporte pois o UDP oferece um melhor controle para a aplicação (que o TCP) sobre quais dados são enviados e quando eles são enviados. Explique: **(1,0 ponto cada item)**
    1. Por que uma aplicação possui mais controle de quais dados são enviados em um segmento?
    RESPOSTA:
    Para o envio utilizando TCP, a aplicação escreverá seus dados em um *buffer* gerenciado por este protocolo. Sendo assim, não necessariamente o TCP enviará uma única mensagem em um segmento. O TCP pode encapsular em um único segmento mais ou menos de uma mensagem da aplicação. Diferentemente, o UDP encapsula e envia um segmento assim que a aplicação fornece dados para isto. Desta forma, fica claro que o UDP permitirá maior controle por parte da aplicação.

    2. Por que uma aplicação possui mais controle de quando o segmento é enviado?
    RESPOSTA:
    Com o TCP há controle de congestionamento e fluxo, inserindo assim atrasos significativos desde a gravação dos dados no *buffer* até que esses dados são enviados para a camada de rede. Já no UDP não há controle de congestionamento nem controle de fluxo, o que elimina os atrasos causados por esses controles.

2. É possível que uma aplicação que executa sobre UDP possa se beneficiar da transferência de dados confiável? Se sua resposta é afirmativa, explique como isso é possível. **(1,0 ponto)**
RESPOSTA:
Sim, desde que a transferência de dados confiável seja implementada na própria aplicação, caso contrário não seria possível, pois o UDP não implementa a transferência de dados confiável.

3. Responda verdadeiro ou falso, explicando sua escolha: **(0,5 pontos cada item)**
    1. O tamanho de **RcvWindow** (definida no seu livro texto) do TCP nunca muda enquanto dura a conexão.

RESPOSTA:
Falso. O tamanho da **RcvWindow** depende da quantidade de dados recebidos e ainda não entregues à aplicação, portanto a **RcvWindow** pode crescer ou diminuir durante a conexão.

2. Suponha que o hospedeiro A esteja enviando para o hospedeiro B um arquivo grande por meio de uma conexão TCP. O número de bytes não reconhecidos que o hospedeiro A envia não pode exceder o tamanho do *buffer* de recepção do hospedeiro B.
   RESPOSTA:
   Verdadeiro. Isto é garantido pelo mecanismo de controle de fluxo em relação à janela de recepção, que sempre menor ou igual ao *buffer* de recepção em B. Portanto o hospedeiro A nunca terá um número de bytes enviados e não reconhecidos que seja superior ao *buffer* de recepção do hospedeiro B.

3. Suponha que o hospedeiro A esteja enviando para o hospedeiro B um arquivo grande por meio de uma conexão TCP. Se o número de sequência para um segmento transmitido nessa conexão é **m**, então o número de seqüência para o segmento subsequente e **m+1.**
   RESPOSTA:
   Falso. O próximo número de segmento será **m** acrescido da quantidade de bytes de dados da aplicação enviados no segmento anterior.

4. Imagine que o hospedeiro A envie ao hospedeiro B, por uma conexão TCP, um segmento contendo 9 bytes de dados e com número de seqüência 42. Nesse mesmo segmento, o número contido no campo de confirmação é obrigatoriamente 51.
   RESPOSTA:
   Falso. O valor inicial para os números de seqüência e de confirmação, nos dois extremos da conexão são definidos durante o estabelecimento da conexão TCP (durante o *three way handshake*), e cada uma das partes é livre para escolher o número de seqüência inicial que desejar.

5. Considere o controle de congestionamento no TCP. Quando um temporizador expira no transmissor, o limiar (*threshold*) é ajustado para a metade do seu valor anterior.
   RESPOSTA:
   Falso. O limiar é ajustado à metade do tamanho da janela de congestionamento quando da ocorrência da expiração do temporizador.

4. Considere o protocolo rdt3.0 seja usado na comunicação entre dois sistemas finais A e B. Considere ainda que o tempo de ida e volta (*roung trip time* - RTT) entre A e B, é de 30 milissegundos. Suponha que A e B estejam conectados por um canal que tem taxa de transmissão igual a 1 gigabit por segundo. Considere finalmente que cada pacote enviado de A para B tem tamanho 1.500 bytes. Qual deve ser o tamanho da da janela para que a utilização do canal seja maior que 95%? **(1,0 ponto)**

RESPOSTA:
Tamanho do pacote = 1500 Bytes = 1500 * 8 bits
Taxa de transmissão = 1Gbps = 10^9 bits/s

Aplicando L/R = 1500*8/10^9 = 0,000012 = 12 microssegundos ou 0,012 milissegundos

Utilização do canal (U) >= 95%
Tamanho da janela (n) = ?

U = 0,95
RTT = 0,15 + 0,15 = 0,30 milissegundos (tempo de ida e volta)

Aplicando U = (L/R) / (L/R) + RTT

0,95 = (0,012 * n) / 30 + 0,012
n = (0,95 * 30,012) / 0,012
n = 2375,95

Portanto n é aproximadamente 2376 pacotes. Ou seja, sua janela é de aproximadamente 2376 pacotes.

5. Considere o protocolo Retorne a N (Go-Back-N) com tamanho de janela do transmissor igual a 4 e uma faixa de números de seqüência de 512. Suponha que no tempo **t** o pacote que o receptor está esperando, tenha o número de seqüência **k**. Suponha também que o meio de transmissão não reordene as mensagens. Responda: **(1,0 ponto cada item)**
    1. Quais são os possíveis de números de seqüência dentro da janela do transmissor no tempo **t** ? Justifique sua resposta.
    RESPOSTA:
    No problema temos o tamanho de janela igual a 4. A faixa de números de seqüência possível é de **k - 4** até **k + 3**, respeitados os limites nos quais à faixa de números de seqüência começa a ser reutilizada.
    Podemos justificar esta faixa de números de seqüência através de seus casos extremos, apresentados a seguir:
        - Janela do transmissor **(k-4**, **k-3, k-2, k-1)**: Neste caso o transmissor já enviou os quatro pacotes e o receptor já transmitiu seus respectivos ACKs e estes estão à caminho do transmissor. Portanto o receptor aguarda o pacote com o número de seqüência **k**.
        - Janela do transmissor (**k**, **k+1**, **k+2**, **k+3**): Neste caso o transmissor já recebeu os ACKs até o número de seqüência k-1, recebeu da aplicação os próximos quatro pacotes, já os enviou para o receptor e o receptor ainda não recebeu o pacote com número de seqüência **k**.

    2. Quais os possíveis valores do campo ACK na mensagem que está correntemente se propagando de volta para o transmissor no tempo **t**? Justifique sua resposta.

RESPOSTA:
k-4, k-3, **k-2 , k-1**
Se o receptor está aguardando pelo pacote com o número de seqüência **k**, obrigatoriamente o ACK com o número de seqüência **k-1** já foi enviando. Dado que a janela do transmissor comporta até quatro pacotes podemos concluir que no máximo quatro pacotes de ACK estão se propagando na direção do transmissor. Portanto, os números de seqüência destes três pacotes seriam k-4, k-3, k-2, k-1.

6. Considere o protocolo Repetição Seletiva para transferência de dados confiável. Considere ainda que a faixa de números de seqüência é: **0, 1, 2, ..., k-1.** Qual o maior tamanho de janela possível para que o protocolo não falhe? Explique sua resposta. **(0,5 pontos)**

   RESPOSTA:
   Tamanho máximo da janela para que o protocolo não falhe é igual a k / 2. Como na Questão 7, Suponha por exemplo, k=8 e portanto os números de seqüências possíveis são 0, 1, ..., 7. Admita os tamanho das janelas do transmissor e do receptor igual a 7 (podemos tentar esse tamanho de janela já, que esse valor nos garante que o protocolo Retorne a N não falha). Nesse caso, na janela do receptor são aguardados os segmentos 0, 1. 2, ...,6. Imagine então, que o remetente envia os segmentos 0, 1, 2, ..., 6 e todos eles são recebidos no destino e confirmados, passando o receptor a aguardar sete novos segmentos, os segmentos 7, 0, 1, ..., 5. Porém por uma infeliz coincidência, todas as confirmações enviadas são perdidas! Após a temporização o remetente re- envia o segmento 0, que chega ao destino e o receptor recebe essa cópia como sendo um novo dado. Então o protocolo falha (o protocolo com tamanho de janela igual a 7, não faz transferência de dados confiável)! Para o protocolo funcionar corretamente, o tamanho da janela pode ser no máximo 4, isto é, k / 2. Na verdade, para que o protocolo não falhe, o conjunto dos números de seqüência de uma janela e a da janela seguinte, devem ter intercessão vazia. **Sugestão: repita o raciocínio considerando a janela do transmissor de tamanho 6 e 5 e certifique-se que nos dois casos o protocolo continua falhando. Finalmente repita o raciocínio considerando a janela do transmissor de tamanho 4 e verifique que desse modo, o protocolo não falha!**

7. Considere a transferência do hospedeiro A para o hospedeiro B, de um arquivo (muito grande) com L bytes. Assuma que o Tamanho Máximo do Segmento (*Maximum Segment Size* - MSS) é de 1460 bytes. Qual o máximo valor de L de modo a que os faixa de números de seqüência não seja esgotada? EXPLIQUE sua resposta. **(0,5 pontos)**

   RESPOSTA:
   Os números de seqüência no TCP não são incrementados de um a cada segmento e sim do número de bytes dos dados enviados. Assim o tamanho do MSS é irrelevante – o tamanho máximo do arquivo que pode ser enviado de A para B é simplesmente o número de bytes que podem ser representados por 2^{32}. Isto é aproximadamente 4 Gbytes.

8. Sabemos que o TCP espera até receber três confirmações antes de realizar a “retransmissão rápida”. Na sua opinião por que os projetistas do TCP preferiram não realizar a “retransmissão rápida” após receber a primeira confirmação duplicada para um segmento?. (0,5 pontos)

   RESPOSTA:

   Vamos supor que os pacotes n, n+1 e n+2 foram enviados e o pacote n foi recebido e confirmado. Se os pacotes n+1 e n+2 foram reordenados no caminho (ou seja, foram recebidos na ordem n+2 e n+1), então o receptor do pacote n+2 gera uma confirmação duplicada para o pacote n. No caso de uma política que aguarda apenas a segunda confirmação em duplicata para retransmitir dados, o transmissor dispararia uma retransmissão. Na situação onde o transmissor espera pela terceira confirmação em duplicata, o receptor tem que receber dois pacotes corretamente após o pacote n, antes de receber o pacote n+1. Os projetistas provavelmente usaram a estratégia de esperar pela terceira confirmação em duplicata por acharem que esta teria o melhor compromisso entre disparar a retransmissão rápida quando necessária e evitar a retransmissão prematura de pacotes no caso da reordenação de pacotes ao longo do caminho entre os dois hospedeiros.