

Curso de Tecnologia em Sistemas de Computação
Disciplina: Redes de Computadores I
AD2 - 1º semestre de 2011 – Gabarito

1. Explique para que serve o processo de demultiplexação e como ele é realizado na camada de transporte da Internet . (2,0 pontos)

Resposta: Um processo (como parte de uma aplicação de rede) pode ter um ou mais *sockets*, portas pelas quais dados passam da rede para o processo e do processo para a rede. A tarefa de entregar dados contidos em um segmento da camada de transporte para a porta correta é denominada demultiplexação. Para tanto, na extremidade receptora, a camada de transporte examina os campos do cabeçalho do segmento para identificar a porta receptora e direciona o segmento para o *socket* correto.

2. Responda verdadeiro ou falso às perguntas abaixo e justifique suas respostas: (1,0 ponto cada)

a) Com o protocolo Repetição Seletiva (SR), é possível ao transmissor receber um ACK para um pacote que caia fora de sua janela atual.

Resposta: Verdadeiro. Suponha que o lado que envia as informações tem uma janela de tamanho 3 envie os segmentos 0, 1 e 2 em t_0 . Esses segmentos são entregues ao receptor, que avança sua janela para aguardar os segmentos 3, 4 e 5, enviando as confirmações correspondentes aos segmentos 0, 1 e 2. Porém as confirmações para esses segmentos são retidas em congestionamento. Em t_1 ($t_1 > t_0$) o transmissor tem o temporizador expirado para o segmento 0 que é então retransmitido. Quando o receptor recebe a duplicata do segmento 0, reenvia o ACK 0 (duplicata). Em t_3 o transmissor recebe o primeiro ACK 0 (o que estava retida no congestionamento), e avança a janela para 1, 2 e 3. Instantes depois (em t_4) o transmissor recebe o ACK 0 (duplicata) cujo segmento não se encontra mais na sua janela.

b) Com o protocolo Retorne-a-N (GBN), é possível ao transmissor receber um ACK para um pacote que caia fora de sua janela atual.

Resposta: Verdadeiro. Suponha que o lado que envia as informações tem uma janela de tamanho 3 envie os segmentos 0, 1 e 2 em t_0 . Esses segmentos são entregues ao receptor, um após o outro, que avança sua janela para aguardar o segmento 3, depois o 4 e depois o 5, enviando as confirmações correspondentes aos segmentos 0, 1 e 2. Porém as confirmações para esses segmentos são retidas em congestionamento.

Em t_1 ($t_1 > t_0$) o transmissor tem o temporizador expirado para o segmento 0 e então retransmite os segmentos 0, 1 e 2. Quando o receptor recebe a duplicata do segmento 0, reenvia o ACK 2 (duplicata). Em t_3 o transmissor recebe o primeiro ACK 2 (que estava retido no congestionamento), e avança a janela para 3, 4 e 5. Instantes depois (em t_4) o transmissor recebe o ACK 0 (duplicata) cujo segmento não se encontra mais na sua janela.

c) O protocolo de bit alternado (rdt3.0 no livro texto) é o mesmo que o protocolo Repetição Seletiva, com janelas de tamanho 1 no transmissor e no receptor.

Resposta: Verdadeiro. Note que para um tamanho de janela igual a 1, a Repetição Seletiva, o Retorne a N e o protocolo de bit alternado são funcionalmente equivalentes. O tamanho de janela de 1 impede a possibilidade de pacotes fora de ordem (dentro da janela). Um ACK cumulativo é apenas um ACK normal nesta situação, dado que este se refere apenas a um único pacote dentro da janela.

d) O protocolo de bit alternado (rdt3.0 no livro texto) é o mesmo que o protocolo Retorne-a-N, com janelas de tamanho 1 no transmissor e no receptor.

Resposta: Verdadeiro. Mesma resposta do item (c).

3. Sabemos que o TCP espera até receber três duplicatas de uma confirmação (ACK) para executar a fase de "retransmissão rápida". Na sua opinião qual razão dos projetistas do TCP não terem decidido iniciar a "retransmissão rápida" após a recepção da segunda duplicata. (1,5 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.

4. Suponha que o *round trip time* (RTT) entre o transmissor e o receptor TCP é constante e conhecido pelo transmissor. Assumindo que perdas de segmentos podem ocorrer, explique: Será necessário manter a temporização no protocolo rdt 3.0 apresentado no nosso livro texto? (1,5 pontos)

Resposta: Sim, a temporização ainda será necessária no protocolo rdt 3.0. Se o RTT é conhecido a única vantagem é que o lado transmissor da comunicação saberá, com certeza, que o pacote ou o ACK (ou NACK) do pacote foi perdido, quando compararmos com o cenário real, no qual o pacote pode estar a caminho do lado transmissor, após a expiração do temporizador. No entanto, para detectar a perda, para cada pacote, um temporizador com duração constante será necessário no lado transmissor da comunicação.

5. Como é escolhido o valor do temporizador de uma conexão TCP? (1,0 ponto)

Resposta: Denominamos "RTT amostra" (*Round Trip Time* amostra), o tempo transcorrido desde a entrega do segmento para o IP até o momento em que é recebida a sua confirmação. A maioria das implementações de TCP faz uma única medição "RTT amostra" por vez (isto é, para apenas um dos segmentos transmitidos e ainda não confirmados). As medidas obtidas para os "RTT amostra" de diferentes segmentos, variam devido a congestionamento nos roteadores e a variação de carga nos hospedeiros. Por causa dessa variação qualquer valor medido para "RTT amostra" pode ser atípico. Portanto, para estimar um valor para o RTT típico, é preciso determinar algum tipo de média sobre os valores "RTT amostra" obtidos. Para tal, o TCP mantém uma média, denominada "RTT estimado" dos valores "RTT amostra". Essa média é atualizada sempre que um novo valor de "RTT amostra" é obtido, da seguinte forma:

$$\text{RTT_estimado} = (1 - a) * \text{RTT_estimado} + a * \text{RTT_amostra}$$

Este cálculo fornece a média exponencial ponderada, na qual a influência de cada "RTT_amostra" diminui exponencialmente com o tempo. Um valor típico usado para a é 0,125. Para determinar o valor da temporização de uma conexão TCP, ao "RTT_estimado" é adicionada uma margem de segurança. Esta margem de segurança é calculada através do desvio da amostra (denominado "desvio RTT") em relação ao "RTT estimado". Assim "desvio RTT" é determinado da seguinte forma:

$$\text{desvio_RTT} = (1 - \beta) * \text{desvio_RTT} + \beta * | \text{RTT_amostra} - \text{RTT_estimado} |$$

O valor recomendado para β é 0,25. Com os valores de "RTT estimado" e de "desvio RTT", o valor do temporizador é calculado da seguinte forma:

$$\text{Temporizador} = \text{RTT_estimado} + 4 * \text{desvio_RTT}$$

6. Os hospedeiros A e B estão conectados diretamente por um enlace de 200 Mbps. Existe uma conexão TCP entre esses dois hospedeiros, e o hospedeiro A está enviando um grande arquivo para o hospedeiro B através dessa conexão. O hospedeiro A pode enviar dados da aplicação a taxa de 100Mbps porém no hospedeiro B o *buffer* de recepção TCP (*receive buffer*) é lido pela aplicação a taxa de 50Mbps. Descreva o efeito do controle de fluxo do TCP nesse cenário. (1,0 ponto)

Resposta: O hospedeiro A envia dados para o *buffer* do receptor mais rapidamente do que o receptor do hospedeiro B pode retirar dados do *buffer*. O *buffer* de recepção em B rapidamente fica cheio devido a maior velocidade na entrada de dados que na retirada dos mesmos. Quando o *buffer* estiver cheio, o hospedeiro B sinaliza para A que pare de enviar dados atribuindo à janela de recepção o valor zero ($\text{RecWindow}=0$). O hospedeiro A para de enviar dados até que receba um valor para a janela de recepção superior a zero ($\text{RecWindow}>0$). Desta forma, o hospedeiro A pára e inicia a transmissão de dados como uma função dos valores recebidos para a janela de recepção do hospedeiro B. Na média, ao longo do tempo, a taxa de transmissão na qual o hospedeiro A envia dados para o hospedeiro B, durante a conexão não será superior a 50Mbps, evitando que o lado transmissor suplante a capacidade de recepção do outro lado da conexão.