



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Redes de Computadores I**

**AD2 - 2º semestre de 2010.**

**Gabarito**

---

1. Responda verdadeiro ou falso, explicando sua escolha: **(0,5 ponto cada)**

1.1. O tamanho de RcvWindow (definida na Seção 3.5.5 do seu livro texto) do TCP nunca muda enquanto dura a conexão.

**Resposta:** Falso. O campo "RcvWindow" indica a quantidade de *bytes* disponível no "*buffer* de recepção" no lado receptor de uma conexão TCP. O conteúdo desse campo é usado para evitar que o transmissor envie mais dados do que o receptor é capaz de receber, evitando assim o descarte de dados, por falta de espaço de armazenamento no lado receptor de uma conexão TCP. Portanto, ele varia durante a conexão.

1.2. 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. O controle de fluxo do TCP através do campo "RcvWindow" indica para o lado transmissor a sua capacidade de recepção de dados através do campo "RcvWindow" (vide o item a).

1.3. 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 sequência para o segmento subsequente é  $m+1$ .

**Resposta:** Falso. No caso do TCP, o número de sequência do segmento subsequente seria:  $m + k$ , onde  $k$  é a quantidade de *bytes* de dados contidos no segmento anterior.

- 1.4. Imagine que o hospedeiro A envie ao hospedeiro B, por uma conexão TCP, um segmento contendo 4 *bytes* de dados e com número de sequência 38. Nesse mesmo segmento, o número contido no campo de confirmação é obrigatoriamente 42.

**Resposta:** Falso. Os valores iniciais para os números de sequência para cada um dos lados da conexão TCP são definidos durante o estabelecimento da conexão TCP (*three way handshake*), e cada uma das partes faz a escolha independentemente. Portanto o campo de confirmação, ao longo a transmissão na conexão TCP, reflete essa escolha inicial.

- 1.5. 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:** Verdadeiro. Esta é a estratégia adotada pelo mecanismo de controle do congestionamento do TCP. Além da redução do limiar à metade da janela de congestionamento anterior ao evento, a sua janela de congestionamento é reduzida para o tamanho de 1 MSS e a transmissão entra na fase de *Slow Start*. O caso de um temporizador expirar indica ao TCP que a rede se encontra congestionada, por isto uma medida mais cautelosa é adotada. Diferente do caso da recepção de três ACKs em duplicata, pois, neste caso, os pacotes de ACK ainda estão sendo recebidos no lado transmissor.

2. Explique para que serve o processo de demultiplexação realizado na camada de transporte. **(1,0 ponto)**

**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.

3. Suponha um processo que executa no hospedeiro C tem um *socket* com número de porta 6789. Suponha que dois hospedeiros A e B, enviem segmentos UDP para a porta de destino 6789 do hospedeiro C. Responda:

**(0,5 ponto cada)**

- 3.1. Ambos os segmentos serão direcionados para o mesmo *socket* no hospedeiro C?

**Resposta:** Sim, ambos os segmentos serão direcionados para o mesmo *socket* no hospedeiro C.

- 3.2. Se sua resposta é sim, como o processo que executa no hospedeiro C, sabe que esses dois segmentos têm origem em dois hospedeiros diferentes?

**Resposta:** Para cada segmento recebido, na interface do *socket*, o sistema operacional disponibilizará para o processo o endereço IP de forma a determinar as origens dos segmentos individuais.

4. Suponha que um servidor *Web* executa no hospedeiro C e escuta na porta 80. Suponha que esse servidor *Web* usa conexões persistentes e que está nesse momento, recebendo requisições de dois hospedeiros A e B diferentes. Explique:

- 4.1. Essas requisições são direcionadas para o mesmo *socket* no hospedeiro C? **(0,5 ponto)**

**Resposta:** Não. Para cada conexão persistente, o servidor de *Web* cria um *socket* de conexão separado. Cada *socket* de conexão é identificado pela 4-tupla <endereço IP de origem, número de porta de origem, endereço IP de destino, número de porta de destino>. Quando hospedeiro C recebe um datagrama IP, ele examina esses quatro campos nos cabeçalhos das camadas de rede e de transporte (datagrama/segmento) para determinar para qual *socket* deve passar a carga útil (*payload*) do segmento TCP.

- 4.2. Se as requisições são direcionadas para *sockets* diferentes, podem esses diferentes *sockets* ter o mesmo número de porta 80? **(1,0 ponto)**

**Resposta:** Sim. Como vimos na anterior, cada *socket* de conexão é identificado por quatro informações: <endereço IP origem, número da porta de origem, endereço IP destino e número da porta destino>. Quando o hospedeiro C recebe um datagrama IP, este examina estes quatro campos no datagrama/segmento para determinar para qual *socket* o conteúdo de informação do segmento TCP deve ser repassado. Assim, as requisições de A e B vão para *sockets* diferentes. O identificador para ambos *sockets* tem 80 como porta de destino, porém, os identificadores para estes *sockets* têm valores diferentes para o endereço IP de origem. Diferentemente do que ocorre no UDP, quando a camada de transporte repassa o conteúdo de informações de um segmento de TCP para o processo da aplicação, não especifica o IP de origem, pois isto é implicitamente especificado pelo identificador do *socket*.

5. Para que serve o campo “Janela de Recepção” (ou “*RcvWindow*”) no cabeçalho do segmento TCP? **(1,0 ponto)**

**Resposta:** O campo “*RcvWindow*” indica a quantidade de *bytes* disponível no “*buffer* de recepção” no lado receptor de uma conexão TCP. O conteúdo desse campo é usado para evitar que o transmissor envie mais dados do que o receptor é capaz de receber, evitando assim o descarte de dados, por falta de espaço de armazenamento no lado receptor de uma conexão TCP.

6. 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  $\beta$  é 0,25. Com os valores de “RTT estimado” e de “desvio RTT”, o valor do temporizador é calculado da seguinte forma:

$$\text{Temporizador} = \text{RTT\_estimado} + 4 * \text{desvio\_RTT}$$

7. Responda verdadeiro ou falso às perguntas abaixo e justifique suas respostas:

(0,5 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).