Fundação CECIERJ - **Vice Presidência de Educação Superior a Distância**

Curso de Tecnologia em Sistemas de Computação

Disciplina: Redes de Computadores I

AP2 - 2° semestre de 2010.

Gabarito

---

1. **(1,5 pontos)** Tanto o TCP como o UDP utilizam o complementos de 1 para suas somas de verificação (*checksums*). Considere as seguintes três *words* de dezesseis bits: 0110011001100000, 0101010101010101 e 1000111100001100. Qual o complemento de 1 para as somas dessas *words*?

   Resposta:

   Somando as duas primeiras *words*:

```
  0110 0110 0110 0000
+ 0101 0101 0101 0101
---------------------
  1011 1011 1011 0101
```

   Somando o resultado da soma das duas primeiras *words* com a terceira *word*:

```
  1011 1011 1011 0101
+ 1000 1111 0000 1100
---------------------
  0100 1010 1100 0001
```

   (ocorre *overflow*, é preciso somar 1 ao resultado)

   Somando 1 (0000 0000 0000 0001) ao resultado da soma anterior:

```
  0100 1010 1100 0001
+ 0000 0000 0000 0001
---------------------
  0100 1010 1100 0010
```

   (resultado final da soma)

   O complemento de um é o valor que somado ao resultado final da soma, produz 1111 1111 1111 11111. Portanto:

   0100 1010 1100 0010 (resultado da soma)

   1011 0101 0011 1101 (é o complemento de 1 procurado)

2. **(1,5 pontos)** Qual a razão da necessidade dos "números de sequência" nos protocolos para transferência confiável de dados (*reliable data transfer protocol - rdt*)?

   Resposta: Os "números de sequência" são utilizados para que o receptor rdt possa detectar se um pacote que acaba de ser entregue contém dados novos ou se trata-se de um pacote retransmitido. Se o protocolo faz transferência confiável de dados com paralelismo (*pipelined)*, como é caso do protocolo Repetição Seletiva, os números de sequência servem também para que o receptor possa detectar pacotes perdidos ou recebidos fora de ordem. Nesse caso o receptor rdt deve preencher as lacunas no fluxo de *bytes* recebidos, antes da entrega dos dados para a aplicação.

3. **(1,5 pontos)** Qual a razão da necessidade de “temporizadores” nos protocolos para transferência confiável de dados (*reliable data transfer protocol - rdt*)?
Resposta: Os temporizadores servem para tratar as perdas de pacotes pelocanal de transmissão. Se uma confirmação para um pacote enviado, não é recebido, dentro do intervalo usado na temporização, então o pacote (ou seuACK ou NAK) é considerado perdido e o pacote é retransmitido. Obviamenteesta estratégia cria a possibilidade de pacotes em duplicata no receptor, mas este problema é resolvido através dos números de sequência.

4. Suponha que o hospedeiro A envia dois segmentos para o hospedeiro B em uma conexão TCP. O primeiro segmento tem número de sequência 31 e o segundo tem número de seqüência 126.

    a. **(0,5 pontos)** Quantos bytes de dados estão contidos no primeiro segmento? Explique.
    Resposta: 95. No TCP o próximo número de sequência do segmento é definido pelo número de sequência do segmento anterior, acrescido da quantidade de *bytes* de dados enviados no segmento anterior (observado o limite máximo do número de sequência). Neste caso, o número de sequência 126 é definido pelo número de sequência 31 somado ao número de *bytes* de dados transmitidos nesse primeiro segmento, que no caso é 95.

    b. **(1,0 pontos)** Suponha que o primeiro segmento foi perdido mas o segundo foi entregue a B. Na confirmação que B envia para A, qual o valor contido no campo ACK do segmento? Explique.
    Resposta: 31. Isto, caso o segmento anterior ao segmento com número de sequência 31 já tenha sido confirmado. O TCP sempre envia no campo de confirmação o próximo *byte* em sequência (isto é, em ordem), por ele esperado.

5. **(2,0 pontos)** Para que serve o controle de fluxo realizado na camada de transporte da Internet? Explique como esse serviço é implementado.
Resposta: O controle de fluxo serve para evitar que o transmissor da conexão TCP envie mais dados que a capacidade de recepção do receptor dessa conexão TCP. Para tal, o transmissor precisa conhecer o espaço disponível no *buffer* de recepção do lado receptor da comunicação. Este dado é fornecido através do campo “Receive Window”, que está presente no cabeçalho do TCP e que indica o espaço, em *bytes*, disponível no *buffer* de recepção. Como o TCP é *fullduplex,* o campo “Receive Window” é preenchido, no lado receptor da conexão, todas as vezes que um segmento é enviado para o lado transmissor. Com o mecanismo de controle de fluxo, um problema poderia ser ocasionado quando a janela de recepção disponível chegasse a zero. Nessa situação, o transmissor não enviaria dados para o receptor e caso o receptor não tivesse dados para enviar na conexão, o transmissor não teria como saber que o espaço na janela de recepção foi liberado. O TCP resolve este problema forçando o envio periódico de pacotes, por parte do transmissor, com apenas um *byte* de dados, quando a capacidade da janela de recepção chega a zero.

6. **(0,5 pontos cada item)** Considere o comportamento da janela de congestionamento do TCP – Reno mostrado no gráfico abaixo. Responda:

   a. Indique as regiões de operação em modo *slow start* (partida lenta)
   Resposta: 1 à 6 e de 18 à 22 (rodada de transmissão).

   b. Quais os intervalos de tempo em que a prevenção de congestionamento está em execução
   Resposta: 6 à 10, 11 à 17 e de 22 à 30 (rodada de transmissão).

   c. O que ocorre na 17ª rodada da transmissão?
   Resposta: Ocorreu um *timeout*, ou seja, o temporizador expirou e a janela de congestionamento é reduzida a seu tamanho mínimo (1 MSS) e o *threshold* é definido como a metade do tamanho da janela de congestionamento no momento do evento (12 MSS). A transmissão entra na fase de *slow start* (partida lenta).

   d. Qual o valor do limiar (*threshold*) na 20ª rodada da transmissão?
   Resposta: 12 segmentos.