

1. **(1,0 pontos)** Tanto o TCP como o UDP utilizam o complementos de 1 para suas somas de verificação (*checksums*). Considere os seguintes três *words* de 16 bits: 0110 0110 0110 0000, 0101 0101 0101 0101 e 1000 1111 0000 1100. Qual o complemento de 1 para as somas desses *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 1111. Portanto:

0100 1010 1100 0010 (resultado da soma)  
+ 1011 0101 0011 1101 (é complemento de 1 procurado)  
-----

1111 1111 1111 1111

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

**Resposta:**

Os “números de seqüência” 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 por exemplo 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)** Uma confirmação TCP perdida não necessariamente força uma retransmissão. Por que?

**Resposta:**

Por causa do uso da confirmação cumulativa. Ou seja, o recebimento da confirmação  $n$  pelo transmissor TCP, indica que todos  $n - 1$  *bytes* anteriores do fluxo de *bytes*, foram recebidos no receptor TCP. Essa confirmação  $n$  pode estar confirmando *bytes* contidos em um ou mais segmentos, cujas confirmações foram perdidas (ou não enviadas).

4. **(1,5 pontos)** 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.

**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 ( $RecWindow=0$ ). O hospedeiro A para de enviar dados até que receba um valor para a janela de recepção superior a zero ( $RecWindow>0$ ). 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.

5. **(1,5 pontos cada item)** Os hospedeiros A e B comunicam-se através de uma conexão TCP, e B já recebeu de A todos os *bytes* até o *byte* 248. Suponha que nessa situação, A envia dois segmentos para B. O primeiro e o segundo segmento contêm respectivamente 40 e 60 *bytes* de dados. No primeiro segmento o número de sequência é 249, o número da porta de origem é 503, e o número da porta de destino é 80. O hospedeiro B envia uma confirmação sempre que ele recebe um segmento de A.

a. No segundo segmento enviado de A para B, quais são os valores do número de sequência, do número da porta de origem e do o número da porta de destino? Justifique sua resposta.

**Resposta:**

No segundo segmento do hospedeiro A para o B, o número de sequência é 289, a porta de origem é 503 e o número da porta de destino é 80. Isto se justifica porque o primeiro segmento tem 40 *bytes* e a sequência é acrescida deste valor, ou seja, passa de 249 para 289. As portas de origem e destino dependem da direção de envio dos pacotes,

de A para B a origem é 503 e o destino é 80, na direção contrária (B para A) a porta de origem é 80 e destino 503, este mesmo raciocínio justifica a definição das portas origem e destino neste item bem como nos itens b e c.

**b.** Se o primeiro segmento chega antes do segundo segmento no destino, na confirmação enviada para o segmento, quais são os valores contidos nos campos de confirmação, de número da porta de origem e de o número da porta de destino? Justifique sua resposta.

**Resposta:**

Se o primeiro segmento chega antes do segundo, na confirmação da recepção do primeiro o valor é 289, a porta de origem é 80 e o número da porta de destino é 503. A justificativa é que o TCP sempre confirma o último pacote recebido corretamente e na ordem esperada.

**c.** Se o segundo segmento chega antes do primeiro segmento no destino, na confirmação enviada para o segmento, quais são os valores contidos nos campos de confirmação, de número da porta de origem e de o número da porta de destino? Justifique sua resposta.

**Resposta:**

Se o segundo segmento chega antes do primeiro, na confirmação da recepção o valor é 249, a porta de origem é 80 e o número da porta de destino é 503. A justificativa é que o TCP sempre confirma o último pacote recebido corretamente e na ordem esperada.