

1. **(1,5 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 o complemento de 1 procurado)

2. **(1,5 pontos)** Suponha que o receptor UDP calcule a soma de verificação da Internet para o segmento recebido e constate que essa soma coincide com o valor transportado no campo soma de verificação (campo *checksum*). Nesse caso o receptor pode estar absolutamente certo de que não ocorreu nenhum erro de bit? Explique.

Resposta:

Não, o receptor não pode estar certo de que alterações em bits não ocorreram. Isto se dá por causa do maneira que o *checksum* para o segmento é determinado. Se por exemplo, bits correspondentes (aqueles que são somados um com o outro) de duas *words* de 16 bits no segmento são 0 e 1, e seus valores forem alterados respectivamente, para 1 e 0, a soma obtida será a mesma em ambos os casos. Portanto, o complemento de 1 calculado no receptor também será o mesmo. Isso significa que embora bits tenham sido modificados durante a transmissão do segmento, o *checksum* no receptor mostrará que o segmento não teve bits alterados durante a transmissão.

3. **(1,0 ponto cada item)** Responda verdadeiro ou falso, explicando sua escolha:
1. 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.
 2. Suponha que o hospedeiro A esteja enviando para o hospedeiro B um arquivo grande por meio de uma conexão TCP. Se o número de seqüência para um segmento transmitido nessa conexão é m , então o número de seqüência para o segmento subsequente é $m+1$.
Resposta:
Falso. O próximo número de segmento será m acrescido da quantidade de bytes de dados da aplicação enviados no segmento anterior.
 3. Imagine que o hospedeiro A envie ao hospedeiro B, por uma conexão TCP, um segmento contendo 4 bytes de dados e com número de seqüência 38. Nesse mesmo segmento, o número contido no campo de confirmação é obrigatoriamente 42.
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.
 4. 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. **(1,5 pontos cada item)** Considere a transferência de um grande arquivo de L bytes do hospedeiro A para o hospedeiro B. Suponha o Tamanho Máximo do Segmento (*Maximum Segment Size - MSS*) é de 536 bytes.
1. Qual o valor máximo de L tal que não sejam exauridos os números de seqüência TCP? Lembre-se de que o campo de número de seqüência TCP tem quatro bytes.
Resposta:
Existem $2^{32} = 4.294.967.296$ números de seqüência possíveis. No TCP o número de seqüência não é incrementado de 1 para cada segmento enviado. Na verdade no TCP o valor do número de seqüência é incrementado pelo 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 pode ser representado por $2^{32} \approx 4,19$ Gbytes.

2. Para o L que obtiver em (4.1), descubra quando tempo demora para transmitir o arquivo. Admita que um total de 66 bytes de cabeçalho das camadas de transporte, de rede e de enlace sejam adicionados a cada segmento antes que o pacote resultante seja enviado por um enlace de 155Mbps. Ignore o controle de fluxo e o controle de congestionamento de modo que A possa enviar os segmentos um atrás do outro e continuamente.

Resposta:

O número de segmentos é $\left\lceil \frac{2^{32}}{536} \right\rceil = 8.012.999$. Devem ser acrescentados

66 bytes de cabeçalho a cada segmento, resultando em um total de 528.857.934 bytes. Assim, o número total de bytes transmitidos é $2^{32} + 528.857.934 = 4,824 \times 10^9$ bytes. Para transmitir o arquivo serão gastos 249 segundos em um enlace 155Mbps.