

1. **(2,0 pontos)** Considere o protocolo Retorne a N para transferência confiável de dados. Suponha que o espaço de números de sequência usado é de tamanho k (isto é, vai de 0 até $k-1$). Responda: qual o maior tamanho de janela (do transmissor) que pode ser usado, para garantir que o protocolo não falhe? Explique sua resposta.

Resposta:

No protocolo Retorne a N o tamanho da janela do receptor tem por definição tamanho 1 (isto é, o receptor só recebe o segmento em sequência). Considerando o espaço dos números de sequência é de k diferentes números (isto é, os números vão de 0 até $k-1$), o tamanho máximo da janela do transmissor tem que ser no máximo igual a $k-1$. Suponha por exemplo, $k=8$ e portanto os números de seqüências possíveis são $0, 1, \dots, 7$. Isto nos levaria a imaginar que o tamanho da janela do transmissor pode ser 8 (isto é, o transmissor pode enviar 8 segmentos, numerados de 0 até 7 , sem ter recebido qualquer confirmação). Imagine então, que o transmissor tem janela de tamanho 8 e que ele envia os 8 segmentos, e recebe a confirmação para cada um deles individualmente ou cumulativamente para todos eles. Tendo recebido essas confirmações, o transmissor avança sua janela, podendo enviar oito novos segmentos reutilizando os números de seqüência de 0 até 7 . Imagine então que esses oito novos segmentos são perdidos e que o receptor envia uma duplicata de confirmação para o último segmento recebido corretamente em seqüência, isto é, o receptor confirma mais uma vez o segmento 7 da primeira leva de segmentos. O transmissor não tem como distinguir que trata-se de uma duplicata da confirmação e considera-a como sendo a confirmação que está aguardando para a nova leva de 8 segmentos enviados e não confirmados, descartando então os oito segmentos que não foram entregues ao receptor. Então o protocolo falha (o protocolo com tamanho de janela igual a 8, não faz transferência de dados confiável)! Desse modo, o tamanho da janela do transmissor pode ser no máximo 7, isto é, $k-1$.

2. **(2,0 pontos)** Descreva por que um programador de uma aplicação distribuída decide optar por usar UDP em vez de TCP na aplicação que está desenvolvendo.

Resposta:

A escolha do UDP como o protocolo da camada de transporte é baseada em:

- **A aplicação não necessita da transferência de dados confiável:** Algumas aplicações suportam que certo percentual de dados sejam perdidos e portanto o uso do UDP é indicado.
- **Taxa de envio não regulada:** O desenvolvedor da aplicação não quer sua aplicação fique sujeita ao controle de congestionamento realizado pelo TCP que pode, em tempo de congestionamento, reduzir a taxa de transmissão da aplicação. No TCP existem os mecanismos de controle de congestionamento e de fluxo que regulam a taxa de envio de segmentos, sem a interferência da aplicação transmissora. Este controle na taxa de envio de segmentos pode ser crítica para aplicações que requisitam uma taxa de transmissão mínima e que são tolerantes a perda de pacotes até um certo nível.
- **Não há estabelecimento de conexão:** No TCP existe o *three way handshake* antes que a transferência de dados seja iniciada.

- **Não há estado de conexão:** O TCP mantém o estado de conexão nos sistemas finais de origem e destino. Esse estado inclui *buffers* de envio e recepção, parâmetros de controle de congestionamento e parâmetros numéricos de sequência e de reconhecimento, desta forma esta manutenção do estado tem custos de processamento e espaço alocado.
- **Pequeno overhead no cabeçalho do pacote:** O segmento TCP tem 20 *bytes* de cabeçalho, enquanto o do UDP tem somente 8 *bytes*.

3. **(2,0 pontos)** Na Internet, os objetivos do controle de fluxo e do controle de congestionamento são os mesmos? Explique sua resposta.

Resposta:

Não. O TCP provê um serviço de controle de fluxo para as aplicações para eliminar a possibilidade de o remetente saturar o *buffer* do receptor. Isto é, trata-se de um serviço de compatibilização de velocidades que compatibiliza a taxa à qual o remetente está enviando com a que aplicação receptora está lendo. Já o controle de congestionamento tem como alvo a infraestrutura de comunicação da Internet que interliga os dois hospedeiros, e tem como objetivo evitar um colapso de comunicação dentro da rede IP. Controle de fluxo e controle de congestionamento são executados por razões diferentes.

4. **(1,0 ponto cada item)** Suponha que o hospedeiro A envia dois segmentos para o hospedeiro B em uma conexão TCP. O primeiro segmento tem número de sequência 150 e o segundo tem número de sequência 205.

a) Quantos *bytes* de dados estão contidos no primeiro segmento? Explique.

Resposta:

55. 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 205 é definido pelo número de sequência 150 somado ao número de *bytes* de dados transmitidos nesse primeiro segmento, que no caso é 55.

b) 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:

150. Isto, caso o segmento anterior ao segmento com número de sequência 150 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)** 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.

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 duplicata 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.