

Redes de Computadores

Atividade IV

Prof. Jó Ueyama

Nome: Pedro Fernando Christofolletti dos Santos

No. USP: 11218560

1) Por que se diz que o TCP provê um *reliable data transfer* (RDT)?

R: Já que o protocolo TCP é orientado a conexão ele garante que todos os pacotes cheguem ao destinatário, sendo assim, é um método de transferência de dados confiável.

2) O que é um *payload* em um pacote de dados?

R: Um pacote necessita de várias informações para que de fato possa ser transferido, dentre essas informações podemos ter o número de sequência, o número de reconhecimento, a porta de origem, a porta de destino, opções etc. O payload é carga útil de dados de um pacote, é o que de fato foi solicitado por uma determinada requisição, é o conteúdo que será utilizado pelo autor da requisição.

3) Estabeleça a diferença entre um pacote mensagem e o de segmento.

R: Uma mensagem é o dado completo que se quer enviar entre o remetente e o destinatário, quando estamos nos referenciando a um pacote na camada de transporte usamos o termo segmento, é uma parte da mensagem completa.

4) Qual a diferença entre um ACK e um NAK?

R: ACK é a abreviação de *acknowledgment* que significa “reconhecimento” e NAK é abreviação de *negative-acknowledgment* que significa “reconhecimento negativo”, ou seja, um pacote que possui um ACK com seu número de sequência é um pacote que foi recebido corretamente e foi reconhecido, e um pacote com NAK e seu número de sequência é um pacote que não foi recebido corretamente e portanto não foi reconhecido.

5) Descreva os seguintes termos técnicos existentes no protocolo da

camada de transporte:

- i) *Timeout*
- ii) **Número de sequência do segmento**
- iii) **Encapsulamento**
- iv) **ACK duplicado**

R: Timeout é o tempo de tolerância máxima de espera para um determinado pacote ser transmitido/retransmitido. **Número de sequência do segmento** é o número do primeiro byte do segmento utilizado para reconhecer a ordem dos pacotes.

Encapsulamento é o método utilizado para criar segmentos que contenham as informações de cabeçalho e dados de cada pacote. **ACK duplicado** indica ao emissor que um segmento foi recebido fora de ordem e informa qual o número de sequência esperado.

6) Considere uma conexão TCP entre o hospedeiro A e o hospedeiro B. Suponha que os segmentos TCP que trafegam do hospedeiro A para o hospedeiro B tenham número de porta da origem x e número de porta do destino y. Quais são os números de porta da origem e do destino para os segmentos que trafegam do hospedeiro B para o hospedeiro A?

R: Os números de porta são o contrário, origem é y e destino é x.

7) Descreva por que um desenvolvedor de aplicação pode escolher rodar uma aplicação sobre UDP em vez de sobre TCP.

R: Já que o protocolo UDP não é orientado a conexão ele não precisa estabelecer uma comunicação inicial antes de enviar os pacotes, ele os envia independentemente. Dessa forma, o tráfego de pacotes é muito mais rápido do que no protocolo TCP. Esse tipo de protocolo sem conexão é usado quando a perda de pacotes é tolerável, como em um serviço de streaming ou chamada de voz/vídeo pois não é preciso que todos os frames ou áudio chegue perfeitamente pois ainda sim será possível escutar e/ou ver a maior parte da informação, porém a rapidez é necessária para evitar delays e tempo de resposta demasiados.

8) Por que o tráfego de voz e de vídeo é frequentemente enviado por meio do UDP e não do TCP na Internet de hoje? (Dica: A resposta que procuramos não tem nenhuma relação com o mecanismo de controle de congestionamento no TCP.)

R: Porque é tolerável a perda de pacotes sem prejudicar o entendimento e/ou visualização do conteúdo transmitido. Além disso, a fim de evitar delays e/ou tempos de resposta altos o protocolo UDP é mais comumente utilizado por ser mais rápido que o TCP já que ele não necessita estabelecer uma comunicação com o destinatário antes do envio de pacotes.

9) É possível que uma aplicação desfrute de transferência confiável de dados mesmo quando roda sobre UDP? Caso a resposta seja afirmativa, como isso acontece?

R: Sim. É possível implementar a confiabilidade na própria aplicação, por intermédio de algum tipo de controle de entrega e/ou sequenciamento dos pacotes enviados, e posterior troca de informações de controle de entrega entre os processos pares que formam a aplicação.

10) Suponha que um processo no hospedeiro C possua um socket UDP com número de porta 6789 e que o hospedeiro A e o hospedeiro B, individualmente, enviem um segmento UDP ao hospedeiro C com número de porta de destino 6789. Os dois segmentos serão encaminhados para o mesmo socket no hospedeiro C? Se sim, como o processo no hospedeiro C saberá que os dois segmentos vieram de dois hospedeiros diferentes?

R: Sim, os dois segmentos serão direcionados para o mesmo socket e mesmo processo no sistema final C. O processo diferencia a origem dos segmentos pelo endereço IP de origem.