

IF678 – Infraestrutura de Comunicação (Turma EC)
2ª Avaliação Escrita (23/08/2022)
Prof. José Augusto Suruagy Monteiro

A pontuação total dá 14 pontos. Portanto, cada questão de 1 ponto vale cerca de 0,7 e a de 2 pontos 1,4.

1) (1 ponto) **Paradigma das aplicações:** Faça a correspondência entre as características abaixo das aplicações com o paradigma correspondente, de acordo com a seguinte notação: (1) Paradigma Peer-to-peer (p2p); (2) Paradigma cliente servidor; (3) Nenhum dos dois paradigmas.

- a) Um processo solicita serviço dos processos que ele contata e fornecerá serviço aos processos que o contatarem. [1]
- b) Comunicação direta entre máquinas de usuários. [1]
- c) Possui um servidor que está sempre conectado. [2]
- d) O HTTP usa esta estrutura de aplicação. [2]
- e) Possui um servidor com um endereço IP bem conhecido. [2]
- f) Não possui um servidor que está sempre conectado. [1]
- g) Máquinas dos usuários com endereços IP fixos. [3]

2) (2 pontos) **Requisitos das aplicações:** Explique o que são cada um dos requisitos diferenciados abaixo para diferentes tipos de aplicações apresentando (i) exemplos de aplicações e (ii) qual o serviço de transporte mais apropriado para cada uma delas.

(a) Tolerância a perdas

Algumas aplicações como as de multimídia toleram algumas perdas, outras como as de transferência de arquivos não admitem nenhuma perda. No primeiro caso pode-se usar o UDP, no segundo caso é recomendado o TCP.

(b) Tolerância a atrasos

Algumas aplicações como as de multimídia em tempo real desejam atrasos máximos limitados, outras como as de transferência de arquivos são tolerantes a atrasos, desde que estes sejam razoáveis. No primeiro caso deve-se usar o UDP por não introduzir atrasos adicionais, no segundo caso é recomendado o TCP.

(c) Vazão mínima

Algumas aplicações de tempo real para serem viáveis requerem uma vazão mínima outras são elásticas, usando a banda que estiver disponível. Embora nem o UDP nem o TCP garantam vazão mínima, como o TCP pode reduzir ainda mais a vazão por conta dos controles de erro, fluxo e congestionamento, para as aplicações que desejam vazão mínima é mais recomendável o uso do UDP. Para as elásticas pode-se usar o TCP.

(d) Segurança

Algumas aplicações desejam que os dados cheguem íntegros e até mesmo de forma sigilosa. Nem TCP nem UDP fornecem serviços de segurança, mas o TCP como TLS ou o UDP com o DTLS (não apresentado em aula) implementam segurança.

NOTAS: organize as suas respostas de acordo com estes 4 requisitos gerais (a-d). Lembrando que para cada requisito geral há aplicações com diferentes requisitos específicos: por exemplo, há aplicações tolerantes a perdas e outras que são intolerantes a perdas.

- 3) (1 ponto) **DNS:** Qual o papel de cada um dos quatro tipos de servidores DNS: local, raiz, TLD e com autoridade?

Local: não faz parte da hierarquia, serve como intermediário (proxy) para realizar as consultas para os demais servidores. Pode responder diretamente se a resposta se encontrar na sua cache.

Raiz: servidores no nível mais alto da hierarquia, contém os nomes e endereços dos servidores de domínio de topo como “.com”, “.br”, etc.

TLD: servidores responsáveis pelos domínios mais altos. Contêm os nomes e endereços IP dos servidores com autoridade do seu respectivo domínio.

Com autoridade: servidores responsáveis pelos mapeamentos de nomes a endereços IP dos servidores de uma determinada organização.

- 4) (2 pontos) **Tempo necessário para baixar uma página Web em diferentes versões do HTTP.** Suponha que queiramos baixar uma página Web composta por uma página base HTML e 10 objetos. Para cada um dos casos abaixo, indique quantos RTTs serão necessários para baixar todos os conteúdos desta página Web:

- a) HTTP 1.0 sem paralelismo [22]

A página html e os 10 objetos são baixados em série cada um levando 2 RTTs (um para abrir a conexão TCP e outro para recuperar o objeto)

- b) HTTP 1.0 com um grau de paralelismo 5 [6]

Após baixar a página html são abertas 5 conexões em paralelo para baixar os primeiros 5 objetos seguido de outras 5 conexões para baixar os demais objetos. Cada um destas três rodadas necessitam de 2 RTTs.

- c) HTTP 1.1 sem paralelismo [12]

Como o HTTP 1.1 é persistente, é aberta uma única conexão TCP seguido da recuperação da página base html e dos 10 objetos em série.

- d) HTTP 1.1 com paralelismo [3]

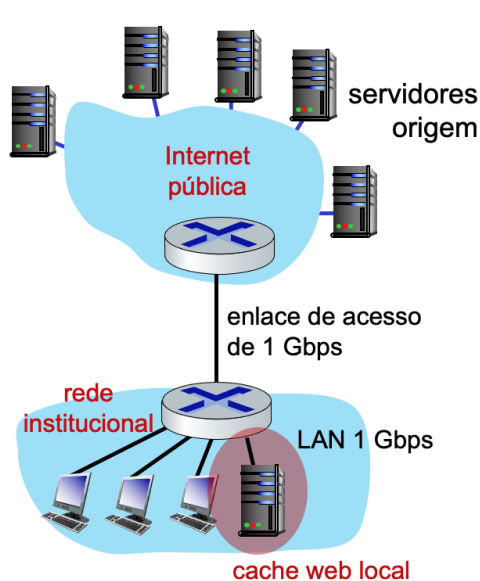
Mais uma vez basta uma conexão TCP, sendo com paralelismo significa que após baixar a página html pode solicitar todos os 10 objetos de uma vez. A limitação do item b de apenas 5 conexões em paralelo era para limitar o número de conexões para um mesmo cliente o que não é o caso aqui.

- e) HTTP/2 [2]

No HTTP/2 quando o usuário pede uma URL o servidor se antecipa e já responde com todos os objetos referenciados na página html.

DICA: Convém apresentar a “memória de cálculo” para que eu possa verificar o porquê de alguma resposta diferente da esperada!

- 5) (1 ponto) **Atrasos para baixar 100 objetos (HTTP 1.1 com cache web local).** Considere um cliente e um servidor HTTP 1.1. O atraso RTT entre o cliente e o servidor é de 2 segundos. Suponha que o tempo que o servidor necessita para transmitir um objeto através do seu enlace de saída seja de 3 segundos.



Há também um cache web local, como mostrado na figura ao lado, com atraso de propagação e tempo de transmissão de objetos desprezíveis (zero). O cliente faz 100 solicitações, uma após a outra, esperando pela resposta a uma solicitação antes de enviar a próxima. Todas as solicitações vão primeiro para a cache (que também tem um atraso RTT de 2 seg. para o servidor, mas zero RTT para o cliente).

Quanto tempo levaria desde que o cliente transmite a primeira solicitação até que ele receba o último objeto solicitado, assumindo que o cliente não use a linha de cabeçalho IF-MODIFIED-SINCE, e assumindo que 50% dos objetos selecionados sejam encontrados na cache local?

Das 100 solicitações, metade serão atendidas pela cache web local com atrasos de propagação e de transmissão desprezíveis. As demais solicitações serão atendidas pelo servidor remoto. Para isso, será feita uma conexão TCP (1 RTT) e cada uma das 50 solicitações levam 1RTT + o tempo de transmissão do arquivo, para um total de $2+3=5\text{seg.}$ Então: Atraso total = $2 + 50 \times 5 = 252 \text{ seg.}$

- 6) (1 ponto) **HTTP/2 versus HTTP/1.1.** Quais das seguintes alternativas são mudanças entre o HTTP 1.1 e o HTTP/2? Selecione uma ou mais respostas.
- a) ☒ O HTTP/2 permite que objetos numa conexão persistente sejam enviados numa ordem de prioridade especificada pelo cliente.
 - b) ☒ O HTTP/2 permite que um grande objeto seja quebrado em pedaços menores, e que a transmissão destes pedaços seja intercalada com a transmissão de outros objetos menores, evitando que um grande objeto force muitos pequenos objetos a esperar a sua vez de transmissão.
 - c) ☐ O HTTP/2 possui diversos novos métodos e códigos de status HTTP.
 - d) ☐ O HTTP/2 provê uma segurança melhorada através do uso da segurança da camada de transporte (TLS – *Transport Layer Security*).
- 7) (1 ponto) **Comparando e diferenciando o HTTP do SMTP.** Quais das seguintes características são aplicáveis tanto ao HTTP como ao SMTP? Nota: marque uma ou mais das características abaixo.
- a) ☐ Usa CRLF.CRLF para indicar o fim da mensagem.
 - b) ☒ Interação de comando e respostas e códigos de status em ASCII.
 - c) ☐ Opera em geral como um protocolo de envio pelo cliente.
 - d) ☐ Usa uma linha em branco (CRLF) para indicar o fim de um cabeçalho de solicitação.
 - e) ☐ Opera em geral como um protocolo de recuperação pelo cliente.
 - f) ☒ É capaz de usar uma conexão TCP persistente para transferir múltiplos objetos.
- 8) (2 pontos) **Streaming e CDNs.** Quais as estratégias adotadas para a transmissão de vídeos em *streaming* em termos de:
- a) (1 ponto) Estruturação dos dados, transmissão e reprodução, e
Os vídeos são divididos em pedaços codificados com diferentes resoluções (exigindo maiores ou maiores taxas). As URLs dos diferentes pedaços com

diferentes resoluções são divulgadas usando um arquivo de manifesto. Os clientes avaliam a vazão disponível e solicitam o pedaço com a melhor qualidade adequada à vazão disponível. O cliente ainda bufferiza os pacotes recebidos para a reprodução para compensar a variação do atraso e minimizar ou evitar interrupções durante a reprodução.

- b) (1 ponto) Redes de distribuição de conteúdo (CDN)

Servidores distribuídos com os conteúdos replicados para atender melhor os clientes em suas respectivas “áreas de cobertura”. Utilizam estratégias de “ir fundo” ou “levar para casa”.

- 9) (1 ponto) **Recurso tipo NS do DNS.** Qual a informação contida no registro de recurso de tipo “NS” na base de dados DNS?

Nos registros NS o nome contém o nome do domínio e o valor corresponde ao nome do servidor com autoridade do domínio correspondente.

- 10) (2 pontos) **Ações dos sockets cliente ou servidor do UDP ou TCP.** Faça a correspondência entre a ação geral do lado cliente ou servidor apresentada com a ação específica relacionada com o socket que a implementa.

Ação geral do lado cliente ou servidor:

- | | |
|--|--------|
| a) Cliente UDP: cria um socket | [1] |
| b) Servidor UDP: cria um socket | [1] |
| c) Cliente TCP: cria um socket | [2] |
| d) Servidor TCP: cria um socket | [2] |
| e) Cliente TCP: cria conexão ao servidor | [3] |
| f) Cliente UDP: Ao transmitir para um servidor, é assim que é identificado um servidor específico. | [10] |
| g) Servidor UDP: Ao transmitir para um cliente, é assim que é identificado um cliente específico. | [12] |
| h) Cliente TCP: Ao transmitir para um servidor, é assim que é identificado um servidor específico. | [4] |
| i) Servidor TCP: Ao transmitir para um cliente, é assim que é identificado um cliente específico. | [6] |
| j) Cliente UDP: Envia ao servidor, usando este socket. | [7] |
| k) Servidor UDP: Envia ao cliente, usando este socket. | [7] |
| l) Cliente TCP: Envia ao servidor, usando este socket. | [8] |
| m) Servidor TCP: Envia ao cliente, usando este socket. | [9] |
| n) Cliente UDP: Ao enviar para o servidor, é assim que o cliente sabe o endereço IP e número de porta do servidor. | [11] |
| o) Servidor UDP: Ao enviar para o cliente, é assim que o servidor sabe o endereço IP e número de porta do cliente. | [12] |

Ações específicas relacionadas com o socket que a implementa:

- (1) Usa a chamada `socket(AF_INET, SOCK_DGRAM)`
- (2) Usa a chamada `socket(AF_INET, SOCK_STREAM)`
- (3) Usa um `connect()` especificando o nome do servidor e o número da porta do servidor.
- (4) O cliente usa `connect()` para vincular explicitamente o seu socket a um servidor específico, portanto, o endereço IP e o número de porta do servidor não precisa ser explicitamente incluído numa operação de envio.
- (5) O servidor usa `connect()` para vincular explicitamente o seu socket a um cliente específico, portanto, o endereço IP e o número de porta do cliente não precisa ser explicitamente incluído numa operação de envio.

- (6) Como resultado de um `accept()`, é criado um novo socket, que vincula o cliente ao servidor através deste socket sem a necessidade de especificar explicitamente o endereço IP e porta destino a cada transmissão.
- (7) Envia usando o socket criado usando `socket(AF_INET, SOCK_DGRAM)`
- (8) Envia usando o socket criado usando `socket(AF_INET, SOCK_STREAM)`
- (9) Envia usando um socket que não foi explicitamente criado através de uma chamada a `socket()`
- (10) Ao transmitir, o cliente inclui explicitamente o endereço IP e o número da porta destino.
- (11) O cliente determina o endereço IP do servidor usando o DNS e o número de porta do servidor é uma porta bem conhecida associada ao tipo de serviço prestado.
- (12) O servidor determina o endereço IP e número de porta do cliente a partir de um datagrama anterior enviado por este cliente.