Lista 1 - Camada de Aplicação

Redes de Computadores

Instituto de Ciência e Tecnologia - ICT

Universidade Federal de São Paulo - UNIFESP

2o semestre de 2021

Observações:

- Os exercícios são de fixação.
- As respostas são dissertativas.
- Não copie respostas.
- Pondere completude e objetividade nas respostas.
- A resolução da lista é individual.
- Utilize o espaço nas caixas de texto para responder.

Nome:

Igor Ribeiro Ferreira de Matos

1) Explique as diferenças entre arquiteturas cliente-servidor e P2P.

A diferença nestas arquiteturas é, basicamente, quem está empenhando o papel “servidor”. Na Cliente-Servidor, temos uma arquitetura centralizada, com o cliente solicitando e recebendo serviços que se encontram em um servidor que - geralmente - está sempre ativo. Normalmente, tem um endereço ip público e permanente. Na P2P (peer to peer), como o nome indica, tem uma estrutura baseada em “pares” de “ponta a ponta” e é mais distribuída. O cliente é ao mesmo tempo cliente e servidor. Aqui temos um uso mínimo ou nulo de servidores, onde o próprio cliente passa a disponibilizar o serviço após usá-lo. (Como o exemplo de torrents, onde você começa a enviar os pacotes que você já tem baixado, quando termina o download).

2) Para ocorrer comunicação entre processos remotos, quais informações são importantes entre origem e destino?

Para o processo emissor, basta um identificador <IP, PORT>. Ou seja, basta saber o endereço IP do destinatário e alguma porta sua para que seja possível a comunicação.

3) Explique o que são e para que servem os *sockets*.

Os sockets são descritores especiais de arquivo. É através deles que há comunicação entre processos de hosts diferentes. Nele, uma ponta da camada de aplicação envia a mensagem que se encontra no seu buffer. A outra ponta recebe esta mensagem e a copia para o seu próprio buffer para a sua própria camada de aplicação. É semelhante à porta.

4) Explique em quais tipos de aplicações recomenda-se transporte TCP ou UDP.

O transporte TCP é recomendado em aplicações que exigem um transporte confiável entre o processo emissor e receptor, mas que a latência não seja importante. Exemplo: transferência de arquivos, envio de emails, etc; onde o que importa é o dado chegar íntegro do outro lado.
Já o transporte UDP é recomendado em aplicações que exigem uma baixa latência, independente da confiabilidade entre o emissor e o receptor. Exemplo: transmissões multimídia ao vivo, jogos interativos, etc; onde o que importa é o dado chegar rápido do outro lado, independente da integridade do dado.

5) Explique como ocorrem as requisições e respostas HTTP.

As requisições são feitas no modelo de transporte TCP. Inicia-se com o cliente fazendo a conexão TCP com o servidor (com a criação do socket), na porta 80. Então o servidor deve aceitar esta conexão. Após o estabelecimento da conexão, mensagens HTTP (do protocolo da camada de aplicação) são trocadas entre o navegador e o servidor web (respectivamente, cliente http e servidor http). Assim, a conexão TCP é fechada.

6) Explique as diferenças entre conexão HTTP persistentes e não-persistentes.

A conexão não persistente requer o restabelecimento da conexão para cada objeto requisitado. É necessário fazer o “handshake” para cada um dos objetos, sendo que cada objeto tem um tempo total de resposta de 2 RTT, sem contar o tempo da transmissão em si. Apesar dele sobrecarregar o sistema operacional, ele é mais vantajoso do ponto de vista do usuário, pois navegadores costumam abrir múltiplas conexões para fazer a busca de objetos em forma paralela (carregando-os bem mais rápido).
A conexão persistente requer apenas um “handshake” para a conexão. A conexão continua aberta após o envio das respostas, fazendo o processo de buscar mais objetos mais dinâmico.

7) Explique o que são, para que servem e como funcionam os *cookies*.

O COOKIE é um identificador de sessão, tecnicamente. Ele identifica sessões do usuário e salva determinados dados, já que o HTTP não salva. Ao fazer o primeiro acesso HTTP em um determinado site, é criado então um id exclusivo para a sessão (o cookie) e uma entrada para um banco de dados de apoio. Neste banco de dados será armazenado informações sobre requisições específicas. Os próximos acessos (ou sessões) poderão contar que algumas determinadas informações estarão salvas.

8) Explique como funcionam caches web.

Todas as requisições feitas por um cliente passa por um servidor proxy, para que este repasse para o servidor de origem. O servidor proxy, no entanto, na primeira requisição vai armazenar um objeto e sua data de última modificação. Quando uma mesma requisição for feita outras vezes, o servidor proxy vai passar a requisição para o servidor de origem e comparar a data da última modificação. Se a data for a mesma, o servidor proxy passa ao usuário o objeto que já está armazenado em cache. Se não, o cache é atualizado e então repassado ao cliente.

9) Explique como funciona o protocolo FTP.

O FTP utiliza o TCP na camada de transporte. Quando o cliente faz a conexão, é aberto dois canais de comunicação.
O primeiro canal, com a porta 21, é onde o cliente executa comandos para upload e download.
O segundo, a porta 20, é onde acontecerá a transmissão de dados.
O cliente passa os comandos e recebe as respostas pela porta 21, enquanto recebe/envia os arquivos efetivamente na porta 20.

10) Explique o funcionamento e a interação dos protocolos envolvidos no envio e recuperação de mensagens de email.

Todo o processo usa como base o protocolo SMTP para a transferência de email, junto de algum outro (como POP3 ou IMAP) para a recuperação.
Primeiro, para um usuário enviar um email, é estabelecida uma conexão (persistente) SMTP. Esta conexão é TCP para transferir a mensagem de modo confiável e roda na porta 25, por padrão. Então, se é feita uma confirmação de conexão (handshaking), para daí sim começar a transferência de mensagens. Esta mensagem deve estar em ASCII de 7 bits. Estas mensagens são interações de requests e replys, onde todos os comando estão no formato ASCII de 7 bits e as respostas são recebidas em códigos e frases de estado. Através destes comandos, é possível enviar o email e armazená-lo no servidor remoto.
Para que a recuperação ocorra, há o protocolo POP3, que é basicamente dividido em duas etapas: a fase de autorização e a fase da transação. Na fase de autorização, deve ser efetuado um login com usuário e senha para confirmar o acesso. Na fase de transação, o usuário pode acessar os emails armazenados enviados para ele no servidor remoto e, caso tenha algum, ele pode lê-los ou excluí-los.
Além do POP3, há outros protocolos como o IMAP, que tem como principais diferenças: mensagens são mantidas em servidores. Usuário pode organizar mensagem por pastas e estado do usuário é mantido entre as sessões.

11) Explique o que é e para que serve o protocolo DNS.

DNS (Domain Name System) é um protocolo de suporte para os usuários finais. É uma base de dados distribuída implementada na hierarquia de muitos servidores de nomes, com o intuito de resolver nomes digitados pelo usuário. Fazer a relação de um endereço de ip para um “endereço web” (apelidos de host), ou vice-versa.

12) Explique como é organizada a distribuição de nomes na hierarquia DNS.

Ela é dividida entre três níveis: Root, Top-Level e Autorizado.
O nível “Root”, ou "raiz" são servidores que vão responder requisições e resolver quem são os servidores DNS responsáveis pelo domínio top-level. Há apenas 13 sistemas destes espalhados pelo mundo. Eles **não** resolvem nomes de domínios autorizados.
O top-level são os servidores em nível alto, os que resolvem domínios“com”, “org”, “net”, “edu”, etc. Além de resolver os domínios de países de alto nível, como “br”, “uk”, “fr”, “ca”, “jp”, etc.
O nível Autorizado são os servidores DNS primário ou secundário local na rede de organização. Eles são mantidos pela organização ou por um provedor de serviços para resolver nomes dos servidores da organização.
Além deles, tem um quarto nível que não entra exatamente na hierarquia, mas que é responsável pelo primeiro contato do cliente: o servidor DNS local. Cada ISP pode ter um, e ele basicamente atua como proxy encaminhando a consulta do usuário para a hierarquia.

13) Explique quais são as diferenças entre consultas DNS iterativas e recursivas.

Na consulta iterativa, primeiro o cliente faz a consulta no servidor local em busca do mapeamento. Caso não tenha, ele consulta o servidor raiz para achar qual seria o servidor Top-Level correspondente. Então, consulta o servidor Top-Level para obter o ip (processo tal que pode se expandir para ir acessando servidores Autorizados e seus sub-domínios). Durante todas as etapas, o cliente recebe uma resposta e ele mesmo é redirecionado.
Na consulta recursiva, as resoluções vão ficando de forma pendente, uma vez que os próprios servidores vão fazendo as consultas subsequentes, retornando apenas o ip para o usuário no final. Isto iria sobrecarregar os servidores gerando carga excessiva.

14) Explique como funcionam as aplicações de bittorrent, DHT e Skype.

Uma aplicação bittorrent gira em torno de um servidor Rastreador (tracker). Quando o usuário baixa um arquivo de metadados da extensão .torrent, ele se registra neste tracker e baixa uma lista de possíveis pares. Essa lista contém o endereço de IP e as portas destes pares. Então, o rastreador começa a direcionar o cliente para que ele forme conexões TCP simultânea com os pares para baixar blocos (chunks) que compõem o arquivo. Sempre que um chunk é finalizado, o cliente pode passar a exercer o papel de servidor, e distribuir este chunk baixado para um outro cliente. A escolha de qual chunk baixar e para qual par enviar chunk são, respectivamente, representadas pelos algoritmos *rarest first* (mais raro primeiro, o chunk com menos pares disponíveis tem prioridade) e *tit-for-tat (*olho por olho, distribui os chunks para os pares mais recíprocos).

Uma aplicação Distributed Hash Table (DHT) é um banco de dados p2p distribuído. Ela funciona como um banco de dados de tuplas no formato (chave, valor). Cada par possui seu identificador e podem consultar/inserir tupla através das chaves. O identificador é um número inteiro entre 0 e (2^n)-1, onde n é o tamanho da chave. Assim, cada par pode ser identificado por um identificador, e caso alguém saia abruptamente ou há a inclusão de novos pares, é possível realizar o procedimento de uma remoção/inserção de nós de uma lista encadeada circular, considerando que cada nó saberia seus próximos dois sucessores.

Já a aplicação Skype, é um modelo híbrido entre cliente-servidor e p2p. Nela, há o servidor de login Skype e a distribuição de nós. Os usuários (pares) conectam entre si para começar a comunicação. Este processo da conexão envolve um sistema hierárquico onde há super-nós (SNs) que ficam responsáveis por obter o mapeamento NAT de outros nós próximos e age como intermediário da aplicação, para redirecionar a comunicação até o seu destino. Por conta disto, os SNs quebram a conexão fim-a-fim, aumentando a latência.