

Lista de Exercícios 5 - Redes de Computadores
Júlio Melo Campos - 22250349

Seção 2.6

R24 - As CDNs geralmente adotam uma de duas filosofias de posicionamento de servidores. Nomeie e descreva brevemente essas filosofias.

R: As CDNs seguem duas principais filosofias de posicionamento de servidores:

1. **Push to the Edge (Distribuição Extrema):** Servidores são distribuídos globalmente, próximos aos usuários, reduzindo a latência e acelerando a entrega de conteúdo estático.
2. **Deep into ISP (Implantação no ISP):** Servidores são colocados dentro das redes dos provedores de internet, otimizando o tráfego local e melhorando a entrega de conteúdo dinâmico ou sensível à latência.

Ambas visam melhorar a experiência do usuário com base em escala e tipo de conteúdo.

R25 - Além das considerações relacionadas à rede, como atraso, perda e desempenho de largura de banda, há outros fatores importantes na definição de uma estratégia de seleção de servidores de CDN. Quais são eles?

R: Outros fatores incluídos são: **localização geográfica dos usuários** (proximidade entre servidor e usuário para reduzir a latência), **capacidade do servidor** (limite de processamento e armazenamento para evitar sobrecarga), **custo de operação** (despesas associadas à transferência de dados e uso regional), **disponibilidade** (capacidade de manter o serviço continuamente ativo) e **confiabilidade** (garantia de que o serviço funcione sem interrupções). Tais fatores garantem que a CDN ofereça uma experiência confiável e econômica.

Seção 3.1 a 3.3

R3 - Considere uma conexão TCP entre o Host A e o Host B. Suponha que os segmentos TCP que viajam do Host A para o Host B tenham o número da porta de origem x e o número da porta de destino y. Quais são os números das portas de origem e destino para os segmentos que viajam do Host B para o Host A?

R: Para os segmentos TCP que viajam do Host B para o Host A, o número da porta de origem e o número da porta de destino serão invertidos em relação aos segmentos que viajam do Host A para o Host B. Ou seja:

- A porta de origem será a porta y (porta de destino dos segmentos A para B);
- Enquanto a porta de destino será a porta x (porta de origem dos segmentos A para B).

R4 - Descreva por que um desenvolvedor de aplicação pode optar por executar uma aplicação sobre UDP em vez de TCP.

R: Um desenvolvedor pode optar UDP pelo TCP, pois há **baixa latência**, já que o UDP é um protocolo sem conexão e não realiza controle de fluxo e verificação de erros como TCP; **eficiência**, pois o UDP não realiza processo de estabelecimento e encerramento de conexão, sem necessidade de enviar confirmações por causa de recebimentos de pacotes, assim reduzindo o overhead, economizando largura de banda e processamento; além do **controle de fluxo de dados**, pois o UDP permite maior controle de envio e gerenciamento de pacotes para o desenvolvedor pelo fato de não existir retransmissão automática em caso de erros; e por fim, **escalabilidade**, onde O UDP se torna mais escalável, pois não exige conexão individual como o TCP e sim, simultânea a todos.

R5 - Por que o tráfego de voz e vídeo é frequentemente enviado por TCP em vez de UDP na Internet atual?

R: É preferível na internet atual o TCP ao invés do UDP para o tráfego de voz e vídeo, mesmo que o UDP seja tradicionalmente usado para voz e vídeo devido à sua baixa latência. Alguns fatores podem ser citados como compatibilidade com firewalls e NAT, controle de congestionamento, confiabilidade na entrega de pacotes, buffering e adaptação em streaming, além da infraestrutura e conveniência.

Esses fatores pesam pelo fato de certas redes, especialmente corporativas, bloquearem ou limitarem o UDP por razões de segurança, enquanto permitem o TCP, que é mais amplamente aceito e tratado como tráfego "confiável".

R6 - É possível que uma aplicação tenha transferência confiável de dados mesmo quando executada sobre UDP? Se sim, como?

R: Sim, é possível que uma aplicação tenha transferência confiável de dados mesmo quando executada sobre UDP. Para isso, o desenvolvedor pode implementar mecanismos de controle de confiabilidade na camada da aplicação, simulando algumas das funcionalidades que o TCP oferece, como confirmação de recebimento (ACK), numeração de pacotes, retransmissão de pacotes perdidos, controle de fluxo e congestionamento, checksums e detecção de erros.

Estes artifícios podem transformar o UDP em uma alternativa confiável, principalmente onde se deseja controle mais granular e baixa latência.

R7 - Suponha que um processo no Host C tenha um socket UDP com o número da porta 6789. Suponha que tanto o Host A quanto o Host B enviem um segmento UDP para o Host C com o número da porta de destino 6789. Ambos os segmentos serão direcionados para o mesmo socket no Host C? Em caso afirmativo, como o processo no Host C saberá que esses dois segmentos se originaram de dois hosts diferentes?

R: Sim, ambos segmentos serão direcionados para o Host C com a porta de destino 6789, pois o UDP usa apenas o número da porta de destino para identificar o socket. Para diferenciar as origens de hosts diferentes, o processo em C necessita verificar e inspecionar as informações do cabeçalho UDP e IP sendo tais informações fornecidas pela camada de transporte. O endereço IP de origem contém o endereço de cada remetente (A e B), permitindo assim a identificação do host específico, além do número da porta de origem que ajuda a encontrar a origem do segmento caso haja múltiplas portas no mesmo host.

R8 - Suponha que um servidor Web esteja sendo executado no Host C na porta 80. Suponha que este servidor Web use conexões persistentes e esteja atualmente recebendo solicitações de dois hosts diferentes, A e B. Todas as solicitações estão sendo enviadas pelo mesmo socket no Host C? Se estiverem sendo passadas por diferentes sockets, ambos os sockets têm a porta 80?

R: Não, as solicitações de cada host (A e B) não estão sendo processadas pelo mesmo socket no Host C. No caso de conexões TCP persistentes, o servidor Web no Host C cria um socket separado para cada conexão com um cliente, mesmo que ambos usem a porta 80 como porta de destino. Isso significa que os sockets que são criados para conexões de A e B tem a porta 80 como porta de destino no servidor, mas são diferenciados pelo par IP de origem dos clientes. Assim, o servidor consegue distinguir e manter conexões usando a mesma porta 80 de destino.