

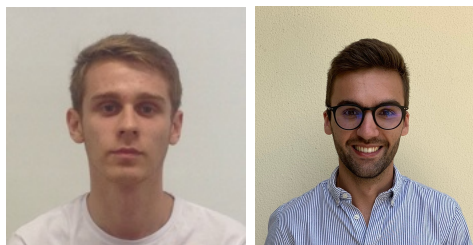


Universidade do Minho

TP1 - Nível Aplicacional: Conceitos Introdutórios

Engenharia de Serviços em Rede

Mestrado em Engenharia Informática



Grupo 83

João Amorim A74806

Tiago Matos PG45585

20 de Outubro de 2021

1 Questões e Respostas

1. As aplicações em rede assentam normalmente em paradigmas cliente-servidor ou peer-to-peer.

a. Explique em que se diferenciam ambos os modelos, salientando o papel das principais entidades envolvidas.

No paradigma do modelo de rede Cliente-Servidor, existe um componente, o Servidor, que responde aos pedidos dos vários Clientes, fornecendo-lhes dados ou serviços, consoante as suas especificações e de acordo com os pedidos que recebe.

Por outro lado, o modelo Peer-To-Peer é constituído por um conjunto de nodos, os "peers", onde não existe a distinção entre Cliente e Servidor, tendo cada um os mesmos privilégios, isto porque cada nodo pode efetuar ou receber pedidos a outros nodos, consoante a sua necessidade. A forma como estes nodos comunicam entre elas e a maneira como os pedidos dos diferentes nodos chegam ao destino pode variar de acordo com a implementação da rede Peer-To-Peer em si.

Adicionalmente, enquanto que no modelo Cliente-Servidor o importante é a partilha de informação, estando a informação contida no Servidor, no modelo Peer-To-Peer a preocupação é a conectividade entre os nodos, uma vez que a informação está contida em cada um dos nodos, sendo que cada novo nodo traz um aumento de capacidade à rede, assim como novas necessidades por serviços.

b. Enuncie vantagens e desvantagens de cada paradigma e casos de aplicação.

Cliente-Servidor:

No paradigma Cliente-Servidor encontramos várias vantagens. Em termos de segurança, os dados/aplicações encontram-se dentro dos servidores, em que estes possuem níveis de controlo de segurança muito mais elevados que a maior parte dos clientes que requisitam serviços, tendo, por exemplo, muitas das vezes diretórios privados em que apenas alguns Clientes con-

seguem aceder. Outro ponto é a escalabilidade, uma vez que podemos escalar a rede para fornecer vários serviços a serem usados por vários clientes, sendo que o update de software pode ser feito a partir de um único computador.

Em relação a desvantagens, este modelo é sem dúvida mais caro de ser implementado, uma vez que para redes de grande escala e de grandes empresas, é necessário um maior investimento, tanto em software como em hardware, para garantir que a rede responde às necessidades que a que vai ser exposta. Da mesma maneira que requer um maior investimento, esta também requer um maior nível de conhecimento por parte de quem gere todo o software e hardware. Quanto à eficácia, também são encontrados alguns problemas, uma vez que quando o servidor tem problemas ou vai abaixo, deixa de ser possível obter os serviços por ele fornecidos, assim como a ocorrência de atrasos na resposta do Servidor quando existe um grande fluxo de Clientes a fazer pedidos um mesmo Servidor ou aos mesmos recursos de um Servidor.

Em termos de casos de aplicação que usem este modelo, temos o File Transfer Protocol (FTP), Servidores Web, Web Browsers e o Domain Name System (DNS).

Peer-To-Peer:

No paradigma Peer-To-Peer encontramos acabamos por ver o oposto, em que a maior parte das vantagens de um modelo Cliente-Servidor, são as desvantagens deste, e o mesmo para as desvantagens. A facilidade e simplicidade de set up da rede, os baixos custo do mesmo e o facto de que se um nodo falhar, os outros conseguem continuar a sua comunicação e partilha de conteúdo, são de facto as maiores vantagens deste paradigma.

Por outro lado, o Peer-To-Peer não é conhecido pela sua segurança, uma vez que esta é feita pelos próprios nodos, ficando-se por umas passwords em ficheiros aos quais não queremos que outros tenham acesso, assim como a possibilidade de que os computadores da rede fiquem sobrecarregados uma vez que têm que não só suportar as necessidades do utilizador, mas também dos pedidos que recebe de outros nodos.

No que diz respeito a casos de aplicação do modelo Peer-To-Peer, temos casos como o BitTorrent, Gnutella, Kazaa, Azureus, WinMX, entre muitos outros.

2. A Tabela 1 identifica tipos de aplicações amplamente usadas na Internet. Essas aplicações ou serviços apresentam diferente sensibilidade ao comportamento e desempenho da rede em si. Para cada tipo de aplicação (ou serviço), identifique qualitativamente os seus requisitos em termos de débito (throughput) necessário, atraso e suas variações (time sensitive) e perda de dados (loss sensitive). Dê exemplo concreto de aplicações da sua preferência que encaixem em cada tipo. Complemente a resposta quantificando os parâmetros em análise (referencie as suas fontes de informação).

Tipos de Aplicações	Débito (throughput)	Atraso e/ou Jitter (time sensitive)	Perda de Dados (loss sensitive)	Aplicações
Web browsing	Elástico	Não	Sem perda	Google Chrome, Mozilla Firefox, Opera
Multimedia streaming	Áudio: 5kbps-1Mbps Video: 10kbps-5Mbps	Sim, 10's ms	Tolerante a perdas	Netflix, Youtube, Kankan
IP Telephony (VoIP)	5kbps-1Mbps	Sim, 10's ms	Tolerante a perdas	Skype, Whatsapp, Zoiper
File transfer/sharing	Elástico	Não	Sem perda	BitTorrent, We-transfer, Filezilla
Interactive Games	Entre 10kbps e 45kbps	Sim, 10's ms	Tolerante a perdas	Google Stadia, Steam
Video Conferencing	Áudio: 5kbps-1Mbps Video: 10kbps-5Mbps	Sim, 10's ms	Tolerante a perdas	Zoom, Skype, Teams

Table 1: Tipos de aplicações e os seus comportamentos/desempenho na rede.

- **Web Browsing**

- Débito: Para as necessidades básicas diárias normais de um utilizador, seja consultar uma determinada página ou enviar um email, 5Mbps ou menos de largura de banda serão suficientes. De acordo com o *httparchive*, o tamanho médio de uma página web é de 3MB, que será carregada em poucos segundos.

- Atraso e/ou Jitter: Aqui, os atrasos não são significativos, uma vez que o importante é garantir a integridade das páginas/dados carregados. Apesar disto, é sempre relevante diminuir qualquer tipo de atraso existente.
- Perda de Dados: Tal como observamos na tabela, normalmente não existe perda de dados, uma vez que a quantidade de dados é relativamente pequena.

- **Multimedia Streaming**

- Débito: Para streams de video em 720p or 1080p, o ideal será a existência de uma largura de banda de 5Mbps enquanto que para streaming em 4K já chegamos a uma necessidade de 25Mbps, tendo em conta a existência de apenas um utilizador na rede a consumir conteúdos do género.
- Atraso e/ou Jitter: O jitter existente deverá ser menor que 30ms. Algo superior a este valor poderá resultar por exemplo em problemas com a qualidade do áudio.
- Perda de Dados: Tolerante a uma certa perda de dados, não devendo ultrapassar o 1% de packet loss.

- **IP Telephony (VoIP)**

- Débito: O débito para VoIP deverá estar entre os 5kbps e 1Mbps, dependendo do número de chamadas em simultâneo, com valores entre os 3Mbps e 10Mbps de upload e download.
- Atraso e/ou Jitter: Aqui, o One-Way Delay não deverá ser maior que 150ms, sendo o alvo os 30ms, enquanto que o Round-Trip Delay não deverá exceder os 300ms.
- Perda de Dados: O packet loss ideal corresponde a valores entre 0% e 1%, no entanto, é possível obter qualidades aceitáveis com packet losses entre valores à volta dos 5% e os 10%.

- **File transfer/sharing**

No caso de aplicações de File Transfer/Sharing, os requisitos são bastante semelhantes.

- **Interactive Games**

- Débito: Ao contrário da *multimedia streaming*, os jogos têm um throughput baixo, entre 10kbps e 45kbps, dependendo de jogo para jogo.
- Atraso e/ou Jitter: Na tabela, está presente um jitter de 10ms, no entanto, qualquer valor abaixo dos 30 ms, será mais que suficiente para uma boa jogabilidade.

- Perda de Dados: Apesar de serem tolerantes a perdas de dados, estas causam uma perda de qualidade na jogabilidade, devendo o packet loss ser sempre inferior a 1% e a latência da rede nunca superior a 150ms.

- **Video Conferencing**

Neste caso, e tal como nas aplicações de Streaming de Multimédia, o débito deverá andar entre os 5kbps e 1Mbps para Áudio e 10kbps a 5Mbps para Vídeo e mais uma vez a perda de dados não deverá ser superior a 1%, para evitar falhas na qualidade de áudio/vídeo. Uma perda maior poderá ser aceitável mas já estará certamente longe do ideal. Em relação a atrasos, é mais do que vimos até agora, 30ms será o *ceiling* que deverá existir para manter uma boa qualidade de serviço.

3. Considere a topologia da Figura 1 onde será distribuído um ficheiro de tamanho X Gbits entre N nodos (hosts), Assuma que os débitos de download e upload do nodo i . são respetivamente d_i e u_i . Assuma ainda que: (i) os hosts estão dedicados à distribuição do ficheiro, i.e. não realizam outras tarefas; e (ii) o núcleo da rede (core) não apresenta qualquer estrangulamento (bottleneck) em termos de largura de banda, i.e., qualquer eventual limitação existe nas redes de acesso dos vários n_i . O valor de X deve ser indexado ao identificador de cada grupo de trabalho, i.e., $X = ID_{Grupo}/10$.

Sabendo que o servidor tem um débito de upload $u_s = 1\text{Gbps}$, e que $d_i = 100\text{Mbps}$, calcule, justificando, o tempo mínimo de distribuição de F pelos N nodos quando $N=10$, $N=100$ e $N=1000$, e para débitos de upload u_i de: a) 1Mbps; b) 5Mbps e c) 10Mbps, usando os modelos de distribuição: (i) cliente-servidor e (ii) peer-to-peer. Apresente os resultados numa tabela comparativa, bem como o processo de cálculo. Que conclusões pode tirar? Note que: 1kbits de dados a transmitir são 1024 bits e um débito de 1kbps são 1000 bits por segundo.

De acordo com os dados fornecidos, vamos começar por agrupar os mesmos numa tabela:

F	8,3	Gb = 8,91E+09	bits
Us	1	Gbps = 1,00E+09	bps
Dmin	100	Mbps = 1,00E+08	bps
Ui	1	Mbps = 1,00E+06	bps
Ui	5	Mbps = 5,00E+06	bps
Ui	10	Mbps = 1,00E+07	bps

F/Us - Tempo de envio de uma cópia.

NF/Us - Tempo de envio de N cópias.

F/Dmin - Tempo mínimo de download de um cliente.

NF/(Us+ $\sum U_i$) - Valor máximo de upload

Para uma arquitetura de Servidor-Cliente, com o aumento do número de clientes ligados no servidor (N), verificamos um crescimento linear do tempo de distribuição. Verificamos também que neste cenário, o débito de upload dos clientes (U_i) não tem qualquer impacto no tempo de distribuição do ficheiro, ao contrário do Peer-to-Peer, pois apenas o servidor faz o upload do ficheiro e este sim é um fator importante a ter em conta (U_s).

N	NF/Us	F/Dmin
10	89.12	89.12
100	891.21	89.12
1000	8912.06	89.12

Table 2: Arquitetura Server-Client - Cálculo do tempo de distribuição.

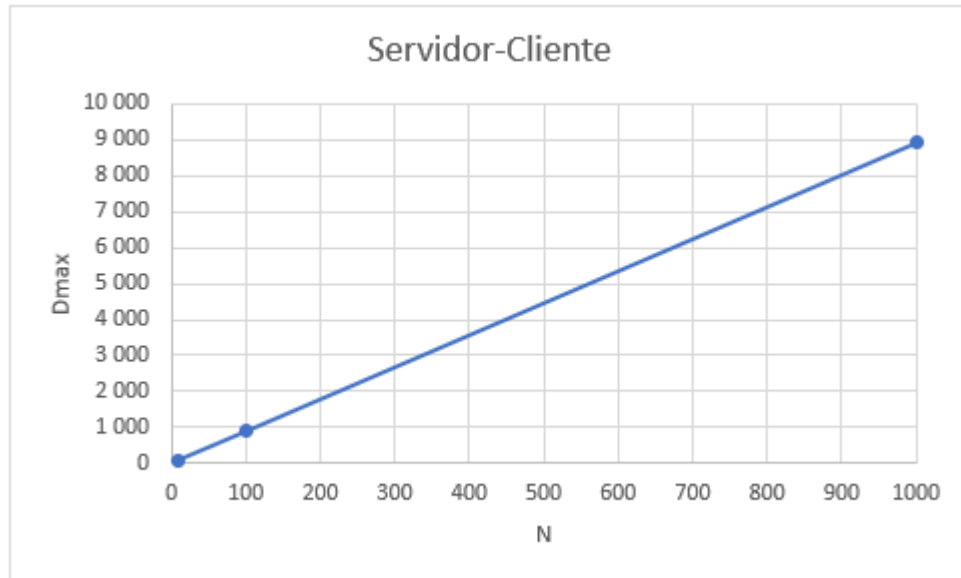


Figure 1: Arquitetura Server-Cliente - Evolução do tempo de distribuição.

Comparando o cenário Servidor-Cliente e P2P, verificamos que para um número reduzido de hosts (N) o tempo de distribuição máximo é praticamente o mesmo, com um número reduzido de hosts a arquitetura Peer-to-Peer não tem vantagens, em termos de tempo, comparado com Servidor-Client (como podemos confirmar quando o número de hosts é 10).

À medida que aumentamos o número de hosts verificamos que o tempo de distribuição na arquitetura P2P é claramente inferior à outra arquitetura. Se o número de hosts se mantiver, mas os débitos de upload dos hosts aumentar, verificamos também que o tempo de distribuição diminui, isto porque qualquer host pode efetuar também o upload de chunks para outros peers.

U_i	N	F/ U_s	F/Dmin	NF/($U_s + \sum U_i$)
1.00E+06	10	8.91	89.12	88.24
	100	8.91	89.12	810.19
	1000	8.91	89.12	4456.03
5.00E+06	10	8.91	89.12	84.88
	100	8.91	89.12	594.14
	1000	8.91	89.12	1485.34
1.00E+07	10	8.91	89.12	81.02
	100	8.91	89.12	445.60
	1000	8.91	89.12	810.19

Table 3: Arquitetura Peer-To-Peer - Cálculo do tempo de distribuição.

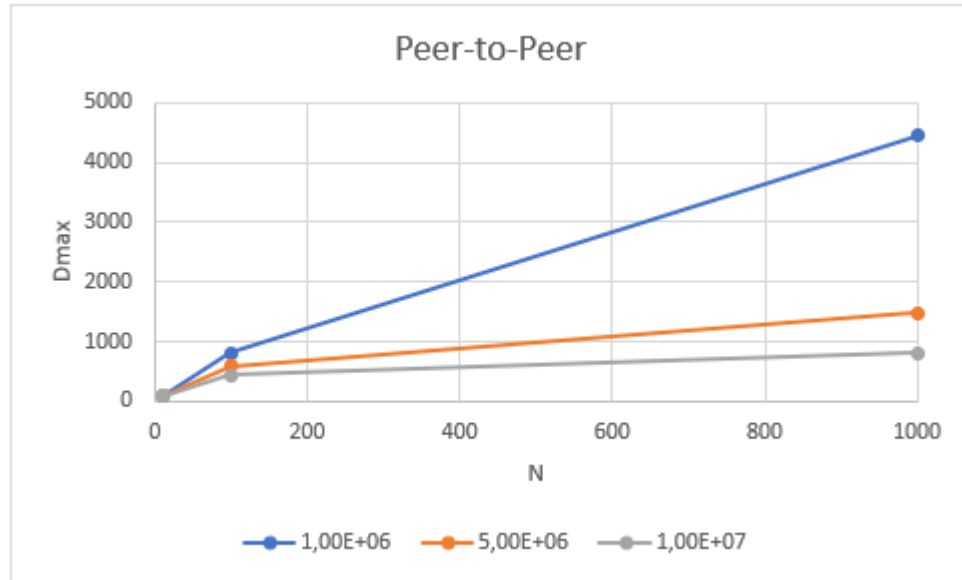


Figure 2: Arquitetura Peer-To-Peer - Evolução do tempo de distribuição.