

# Engenharia de Serviços em Rede

## Trabalho Prático N°1

### Nível Aplicacional: Conceitos Introdutórios

Ana Luísa Carneiro, Ana Rita Peixoto, and Luís Pinto

University of Minho, Department of Informatics, 4710-057 Braga, Portugal  
e-mail: {pg46983,pg46988,pg47428}@alunos.uminho.pt

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

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

Na arquitetura cliente-servidor cada servidor está sempre disponível para receber pedidos vindos do cliente. Após processar cada pedido, o servidor direciona a resposta para o cliente respetivo. Nesta arquitetura os clientes não comunicam diretamente entre eles e os servidores tem um endereço de IP fixo que todos os clientes tem acesso. Desta forma, os clientes podem sempre conectar-se com o servidor se tiverem acesso ao seu IP.

Na arquitetura P2P ou *peer-to-peer* não existe um servidor sempre disponível, mas sim pares intermitentes de *hosts* que comunicam entre si, chamados *peers*. Os *peers* são controlados por utilizadores e podem se encontrar em casas, universidades, escritórios, entre outros. Nesta arquitetura os clientes pedem a outros *peers* o ficheiro ou porções do ficheiro pretendido sem que estes tenham de comunicar diretamente com o servidor.

**alínea b)** Enuncie vantagens e desvantagens de cada paradigma e casos de aplicação.

Uma das vantagens da arquitetura cliente-servidor é de este ser centralizado o que pode ser benéfico para o administrador da rede pois, assim, pode fazer a manutenção e administração dos servidores muito mais facilmente. Além disso, é uma arquitetura mais escalável, pois o aumento do número de servidores permite melhorar linearmente o tráfego na rede. A principal desvantagem desta arquitetura é a congestão de pedidos. Se um servidor receber muitos pedidos de uma vez é possível que este não consiga responder atempadamente criando um elevando nível de congestão. Outro dos problemas é a falta de robustez que existe em arquiteturas centralizadas, ou seja, caso um servidor falhe existe a possibilidade de falha em toda a rede.

Uma das vantagens da arquitetura P2P é a eficiência na transmissão de ficheiros, devido à facilidade de comunicação entre os *peers*. Nesta arquitetura quantos mais clientes fizerem parte da rede, mais fácil é obter ficheiros através dos *peers* e, por vezes, melhor é a performance da rede. Além disso, como é uma arquitetura que dá pouco uso aos servidores, não será necessário dispensar dinheiro na criação de grandes *data centers* e é mais robusta que o paradigma cliente-servidor. A principal desvantagem desta arquitetura é de este ser descentralizado. Como os ficheiros estão armazenados nos *peers* é mais difícil ter acesso à sua localização exata. Apesar do aumento do número de clientes promover a melhoria da rede, por vezes um número elevado de clientes pode afetar negativamente a performance da rede P2P. Além disso, existe falta de segurança na transmissão de ficheiros, pois se um dos *peers* tiver infetado com vírus e este afetar o ficheiro a ser transmitido é possível que os outros *peers* na rede fiquem infetados com o mesmo vírus.

Com estas vantagens e desvantagens de cada arquitetura conseguimos ver os diversos casos de aplicação para cada paradigma. Segundo as vantagens no P2P este é muito mais vantajoso para aplicações e plataformas que envolvam transferência de ficheiros em larga escala como o BitTorrent. As vantagens do cliente-servidor mostra que este paradigma é mais usado quando o cliente deseja algum serviço específico como o email ou aplicações WEB do que na transferência de ficheiros.

**Questão 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 ( <i>throughput</i> )	Atraso e/ou Jitter ( <i>time sensitive</i> )	Perda de dados ( <i>loss sensitive</i> )	Aplicações
Web browsing	elastic	yes, few secs	no loss	Google Chrome, Brave
Multimedia streaming	10kbps-5Mbps	yes, few secs	loss-tolerant	Netflix
IP Telephony (VoIP)	5kbps-1Mbps	yes, 10's msec	loss-tolerant	Skype
File transfer/sharing	elastic	no	no loss	weTransfer
Interactive Games	few kbps up	yes, 10's msec	loss-tolerant	League Of Legends
Video Conferencing	10kbps-5Mbps	yes, 10's msec	loss-tolerant	Zoom

Table 1

Tipos de Aplicações	Débito ( <i>throughput</i> )	Atraso e/ou Jitter ( <i>time sensitive</i> )	Perda de dados ( <i>loss sensitive</i> )	Aplicações
Web browsing	0.2-0.5 Mbps	<100 ms	aprox 0	Google Chrome
Multimedia streaming	3Mbps-25Mbps	< 100 ms	< 1%	Netflix
IP Telephony (VoIP)	100kbps	< 50 ms	< 1% packet loss	Skype (Calls)
File transfer/sharing	elastic	no	no loss	weTransfer
Interactive Games	14-64 kbps	< 70 ping	loss-tolerant	League Of Legends
Video Conferencing	600kbps-3.8Mbps	< 40 ms	< 2% packet loss	Zoom

Table 2

Nas tabelas acima é possível observar o débito, atraso e perda de dados verificada em diversos tipos de aplicações. Na tabela 1, a informação está expressa de forma quantitativa, sendo que o débito é referente à quantidade de informação transmitida pela rede, o atraso remete à possibilidade de *delay* na transmissão dos dados, e a coluna "perda de dados" é indicativa quanto à tolerância para que haja perdas na transmissão. A tabela 2 tem como objetivo atribuir dados concretos a cada uma das aplicações mencionadas como exemplo.

#### WEB BROWSING - Google Chrome:

Relativamente ao *browser Google Chrome*, podemos concluir que um débito de 0.2-0.5 Mbps deverá ser suficiente para permitir uma boa experiência de navegação, quando for utilizado para fins de pesquisa ou edição de documentos na plataforma google drive. Em termos de latência, para permitir uma boa experiência é expectável que não ultrapasse os 100 ms. No âmbito das perdas de dados, é necessário que esta seja pouco significativa, dado que grandes perdas podem impedir a visualização da página web.

Fonte Google Chrome:

<https://support.google.com/chrome/a/answer/3339263?hl=en#:~:text=5%20Mbps%20per%20user%20in,Mbps%20or%20greater%20per%20user.>

### MULTIMEDIA STREAMING - Netflix:

A partir da tabela podemos concluir que a *Netflix* tem um débito entre 3Mbps e 25Mbps dependendo da qualidade do vídeo pretendida, tal como podemos ver na tabela 3. Em termos de latência, para permitir uma boa experiência é expectável que não ultrapasse os 100 ms. No que toca à perda de dados quanto maior for a perda, menor será a largura de banda o que irá implicar uma redução na qualidade de vídeo transmitido. Contudo, é possível que o utilizador continue a ver o vídeo sem que haja perdas visíveis de frames.

Tipo de vídeo	<i>throughput</i> recomendado
SD 480p	3Mbps
HD 720p	5Mbps
4K 2160p (HQ)	25Mbps

Table 3

Não foi possível encontrar valores específicos para a perda de dados relativa à *netflix*. No entanto, foi encontrado um valor de referência para plataformas de *streaming* de vídeo correspondente a 1% de *loss*, ou seja, para o serviço funcionar corretamente não deverá haver perdas superiores a 1%.

Fonte Netflix:

<https://www.reviews.org/internet-service/internet-speed-for-netflix/>

[https://www.researchgate.net/publication/280154796\\_Network\\_level\\_characterization\\_of\\_adaptive\\_streaming\\_over\\_HTTP\\_applications](https://www.researchgate.net/publication/280154796_Network_level_characterization_of_adaptive_streaming_over_HTTP_applications)

<https://www.ringcentral.com/us/en/blog/what-is-jitter/>

### IP TELEPHONY (VoIP) - Skype:

No caso de aplicações VoIP, um exemplo que consideramos relevante foi a aplicação *Skype* pela sua possibilidade de realizar chamadas através da Internet. Pelas pesquisas efetuadas, podemos concluir que o débito relativo a esta aplicação depende da chamada em questão, isto é, nos casos em que se verifica apenas uma chamada de voz o débito recomendado é 100kbps, enquanto que numa chamada de vídeo é 500kbps. Em relação à latência, é possível observar que até 50 ms de atraso, é possível obter uma experiência satisfatória.

Tipo de chamada	<i>throughput</i> mínimo	<i>throughput</i> recomendado
Chamada de voz	30kbps	100kbps
Chamada de vídeo	128kbps	300kbps
Chamada de vídeo (HQ)	400kbps	500kbps
Chamada de vídeo (HD)	1.2Mbps	1.5Mbps

Table 4

Tal como é possível observar na tabela acima, para diferentes tipos de chamadas existem diferentes débitos. Passando de uma chamada de voz para uma chamada de vídeo, o débito aumenta cerca de 4 vezes. No âmbito das video-chamadas os requisitos também variam, dependendo da qualidade da mesma, sendo que uma transmissão *high definition* necessita de 3 vezes mais *throughput* do que uma transmissão *high quality*.

Por último, no que toca à perda de dados, as informações obtidas revelam que para manter a qualidade do serviço as perdas não devem ultrapassar 1%.

Fonte Skype:

<https://support.skype.com/en/faq/FA1417/how-much-bandwidth-does-skype-need>

<https://tomtalks.blog/2018/05/what-are-thresholds-for-good-and-poor-network-packet-loss-jitter-and-round-trip-time-for-unified-communications/>

#### **FILE TRANSFER - weTransfer:**

Nos serviços de transferência de ficheiros o débito pode ser considerado elástico, isto é, pode variar ao longo da transferência do ficheiro em questão. Estas aplicações não são *time sensitive* dado que o tempo não é impactante no serviço prestado, ou seja, apesar do *download* poder demorar mais ou menos tempo, dependendo das condições da rede, isso não irá afetar a integridade do ficheiro resultante. Além disso, estas aplicações não são tolerantes a perdas, dado que pode comprometer a consistência dos dados.

#### **INTERACTIVE GAMES - League Of Legends:**

No jogo de *League Of Legends* existe um débito de 14 a 64 kbps, tal como está na tabela 2. No que toca à latência, caso o ping esteja entre 10 a 70 ms então consegue-se ter uma boa experiência no jogo. Se o ping for menor que 10 então tem-se uma experiência de jogo ideal.

Em relação à perda de pacotes, é esperado que se aproxime o máximo possível de 0, de modo a melhorar a experiência do utilizador. Caso exista muita perda de pacotes, haverá necessidade de os retransmitir o que irá provocar *delay* na aplicação.

Fonte *League of Legends*:

<https://oce.learnwithleague.com/knowledgebase/what-is-lols-typical-data-and-bandwidth-usage/>

<https://win.gg/news/this-is-how-you-can-lower-your-ping-in-league-of-legends/>

#### **VIDEO CONFERENCING - Zoom:**

A partir da tabela 2 podemos concluir que o *Zoom* tem um débito entre 600kbps a 3.8Mbps dependendo da qualidade do vídeo pretendida, tal como podemos ver na tabela 5. Em termos de latência, para permitir uma boa experiência é expectável que não ultrapasse os 40 ms. No que toca à perda de dados, o *Zoom* admite que um packet loss de menor que 2% garante uma boa qualidade de serviço.

Tipo de vídeo	<i>throughput</i> recomendado
High-quality video	600kpbs
720p HD video	1.2 Mbps
1080p HD video	3.8Mbps/3.0Mbps

Table 5

Fonte Zoom:

<https://support.zoom.us/hc/en-us/articles/201362023>

<https://support.zoom.us/hc/en-us/articles/202920719-Meeting-and-phone-statistics>

**Questão 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.,  $X = ID_{Grupo}/10$ .

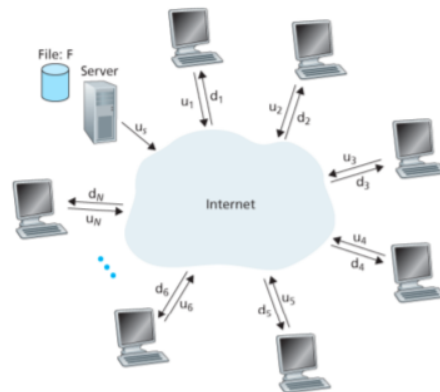


Figura 1 - Distribuição do ficheiro F [Kurose, and Ross, 2016 ].

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: de: a)  $1\text{Mbps}$ ; b)  $5\text{Mbps}$  e c)  $10\text{Mbps}$ , 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:  $1\text{kbits}$  de dados a transmitir são  $1024\text{ bits}$  e um débito de  $1\text{kbps}$  são  $1000\text{ bits}$  por segundo.

**Dados do Problema :**

Tamanho ficheiro,  $X = \text{IDGrupo} / 10 = 20 / 10 = 2$  Gbits

Número de nodos,  $N = 10, 100$  ou  $1000$ .

Débito de upload server,  $U_s = 1$  Gbps

Débito de download dos nodos,  $D_i = 100$  Mbps

Débito de upload dos nodos,  $U_i = 1, 5$  ou  $10$  Mbps

#### **Cliente-Servidor** Resultados :

$$D_{C-S} \geq \max\left\{\frac{N \times X}{U_s}, \frac{X}{D_{\min}}\right\}$$

Para  $N = 10$ :

$$D_{C-S} \geq \max\left\{\frac{10 \times 2 \times 2^{30}}{1 \times 10^9}, \frac{2 \times 2^{30}}{100 \times 10^6}\right\} = \max\{21.475, 21.475\}$$

Para  $N = 100$ :

$$D_{C-S} \geq \max\left\{\frac{100 \times 0.7 \times 2^{30}}{1 \times 10^9}, 21.475\right\} = \max\{214.75, 21.475\}$$

Para  $N = 1000$ :

$$D_{C-S} \geq \max\left\{\frac{1000 \times 0.7 \times 2^{30}}{1 \times 10^9}, 21.475\right\} = \max\{2147.5, 21.475\}$$

Nota: Tal como era esperado, podemos ver na Tabela 6 que o crescimento é linear.

<b>N</b>	<b>10</b>	<b>100</b>	<b>1000</b>
<i>Dc-s</i>	21,475	214,75	2147,5

Table 6: CS - Tempo Mínimo Distribuição em segundos

#### **Peer-to-Peer** Resultados :

$$D_{P2P} \geq \max\left\{\frac{X}{U_s}, \frac{X}{D_{\min}}, \frac{N \times X}{U_s + N \times U_i}\right\}$$

$N = 10$  e  $U_i = 1$  Mbps :

$$D_{P2P} \geq \max\left\{\frac{2 \times 2^{30}}{1 \times 10^9}, \frac{2 \times 2^{30}}{100 \times 10^6}, \frac{10 \times 2 \times 2^{30}}{1 \times 10^9 + 10 \times 1 \times 10^6}\right\} = \max\{2.1475, 21.475, 21.26\}$$

$N = 100$  e  $U_i = 1$  Mbps :

$$D_{P2P} \geq \max\{2.1475, 21.475, \frac{100 \times 2 \times 2^{30}}{1 \times 10^9 + 100 \times 1 \times 10^6}\} = \max\{2.1475, 21.475, 195.26\}$$

$N = 1000$  e  $U_i = 1$  Mbps :

$$D_{P2P} \geq \max\{2.1475, 21.475, \frac{1000 \times 2 \times 2^{30}}{1 \times 10^9 + 1000 \times 1 \times 10^6}\} = \max\{2.1475, 21.475, 1073.74\}$$

$N = 10$  e  $U_i = 5$  Mbps :

$$D_{P2P} \geq \max\{2.1475, 21.475, \frac{10 \times 2 \times 2^{30}}{1 \times 10^9 + 10 \times 5 \times 10^6}\} = \max\{2.1475, 21.475, 20.45\}$$

$N = 100$  e  $U_i = 5$  Mbps :

$$D_{P2P} \geq \max\{2.1475, 21.475, \frac{100 \times 2 \times 2^{30}}{1 \times 10^9 + 100 \times 5 \times 10^6}\} = \max\{2.1475, 21.475, 143.17\}$$

$N = 1000$  e  $U_i = 5$  Mbps :

$$D_{P2P} \geq \max\{2.1475, 21.475, \frac{1000 \times 2 \times 2^{30}}{1 \times 10^9 + 1000 \times 5 \times 10^6}\} = \max\{2.1475, 21.475, 357.91\}$$

$N = 10$  e  $U_i = 10$  Mbps :

$$D_{P2P} \geq \max\{2.1475, 21.475, \frac{10 \times 2 \times 2^{30}}{1 \times 10^9 + 10 \times 10 \times 10^6}\} = \max\{2.1475, 21.475, 19.52\}$$

$N = 100$  e  $U_i = 10$  Mbps :

$$D_{P2P} \geq \max\{2.1475, 21.475, \frac{100 \times 2 \times 2^{30}}{1 \times 10^9 + 100 \times 10 \times 10^6}\} = \max\{2.1475, 21.475, 107.37\}$$

$N = 1000$  e  $U_i = 10$  Mbps :

$$D_{P2P} \geq \max\{2.1475, 21.475, \frac{1000 \times 2 \times 2^{30}}{1 \times 10^9 + 1000 \times 10 \times 10^6}\} = \max\{2.1475, 21.475, 195.23\}$$

Nota: Tal como era esperado, podemos ver na tabela 6 que o crescimento é linear.

<b>N U<sub>i</sub></b>	<b>10</b>	<b>100</b>	<b>1000</b>
<i>1 Mbps</i>	21,47	195,23	1073.74
<i>5 Mbps</i>	21,47	143,17	357,91
<i>10 Mbps</i>	21,47	107,37	195,23

Table 7: P2P - Tempo Mínimo Distribuição em segundos

### Conclusões :

Podemos concluir que no modelo de distribuição cliente-servidor (CS) o crescimento do tempo de distribuição segue uma projeção linear conforme o aumento do número de *hosts*,  $N$ , ao passo que no modelo de distribuição *peer-to-peer* (P2P) o crescimento assemelha-se mais a um crescimento logarítmico.

Deste modo, verifica-se que o modelo P2P tem maior escalabilidade comparativamente ao modelo CS.

Observamos ainda que tanto para o modelo de distribuição P2P como CS o impacto do aumento do  $N$  resulta num aumento de tempo de distribuição. E que, para o modelo P2P o aumento do débito de *upload* dos nodos tende a diminuir o tempo de distribuição.

Numa análise crítica podemos tentar inferir qual será o fator com o maior impacto, isto é, se prevalece o aumento do número de nodos,  $N$ , ou o aumento do débito dos nodos,  $U_i$ . Para isto iremos nos basear na tabela 7, quando o  $n$  é igual a 10 o bottleneck a nível do servidor tem maior preponderância uma vez que a rede ainda não expandiu muito, neste caso verificou-se que o aumento do débito dos nodos não irá provocar nenhuma diferença em relação ao tempo de distribuição. Contudo, no modelo P2P, para  $n = 100$  ou  $n = 1000$ , o efeito do bottleneck do servidor não é tão significativo e a vantagem da troca de ficheiros entre peers torna-se cada vez mais notória, no último caso ( $N = 1000$ ) conseguimos observar que o aumento do débito dos nodos tem um efeito bastante significativo.

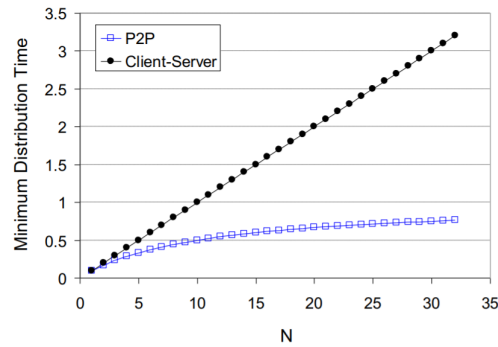


Fig. 1: Exemplo ilustrativo P2P vs Client-Server