

# Ferramenta para monitorização de dispositivos móveis

## Qualidade de Serviço na Internet TP3

Gabriela Martins<sup>1</sup>, Rafaela Silva<sup>1</sup>, and Ricardo Vaz<sup>1</sup>

Universidade do Minho [gci@reitoria.uminho.pt](mailto:gci@reitoria.uminho.pt)  
<https://www.uminho.pt/PT/{a81987,a79034,pg41097}@alunos.uminho.pt>

**Resumo** O presente documento expõe a abordagem do grupo perante a proposta de desenvolvimento de uma ferramenta de monitorização de tráfego. Neste caso, o estudo será entre a versão *app e browser* de uma mesma aplicação.

**Keywords:** QoS · Monitorização de dispositivos móveis

## 1 Definição do Problema

A utilização de *smartphones* parece, cada vez mais, afirmar-se em detrimento da utilização de um PC *Desktop* para o acesso a diversas plataformas digitais. Atualmente, os *smartphones* disponibilizam um meio mais cómodo para o acesso ao *email*, a redes sociais e outras plataformas não menos relevantes.

Este tipo de dispositivo móvel apesar de poder utilizar *Wi-Fi* como forma de acesso à Internet, nem sempre reúne as condições necessárias para tal, visto que a localização ou capacidade dos AP's podem condicionar a sua utilização e, assim, "obrigar" a um uso maioritário de dados móveis. Estes podem ser dispendiosos, pelo que se espera que a sua utilização seja otimizada.

Nesse sentido, o presente trabalho tem como objetivo principal a caracterização de aplicações em dispositivos móveis. Mais especificamente, numa comparação do tráfego gerado pelo *Youtube* na versão aplicacional vs. versão *browser*.

## 2 Tipo de medição

Será utilizada uma medição passiva, pois o principal objetivo é obter informações estatísticas sobre a rede, num contexto real. Assim, este tipo de medição é muito útil para perceber que tráfego está presente e de que forma é que esse flui na rede.

Além disso, a medição será por fluxo, porque neste contexto não há muito interesse em analisar o conteúdo dos pacotes.

Descartou-se a hipótese de medição ativa, uma vez que se espera que sejam feitas comparações reais, no intuito de se inferir que tipo de versão do *Youtube* é mais vantajosa para o utilizador. Sendo que a medição ativa inclui a injeção de tráfego, não faz sentido tendo em conta o objetivo.

### 3 Metodologia de medição

Para a resolução deste trabalho, o grupo optou por expandir uma ferramenta já existente, nomeadamente de captura e análise de tráfego gerado.

A captura será feita utilizando um *gateway* e capturando o tráfego gerado pelo dispositivo móvel. Desta forma, será utilizada uma arquitetura de captura constituída por um *Hotspot* e por um dispositivo de rede intermédio.

No que diz respeito ao *Hotspot*, este será um *Laptop*, de sistema operativo *Windows*. É necessário que este tenha acesso à Internet, à ferramenta *Wireshark* e que possua tecnologias *Hotspot*.

No que concerne ao dispositivo de rede intermédio, este será um *smartphone Android*. É essencial que este tenha acesso à Internet, assim como possua a aplicação *Youtube*.

Em suma, para que seja possível capturar o tráfego gerado no *smartphone*, é necessário que o PC seja o *Hotspot* móvel e que capture, através da ferramenta *Wireshark*, o tráfego gerado nos cenários pretendidos.

### 4 Métricas

Tendo em conta os objetivos estabelecidos, após alguma pesquisa, o grupo considerou que as métricas que se seguem são as que maior impacto têm na obtenção de uma resposta, no âmbito de diferenciar as duas versões. Além disso, as mesmas possuem uma forma de análise simples, existindo várias ferramentas disponíveis tal.

Ao contrário destas, descartou-se a medição de métricas como utilização de *CPU*, *RAM* e bateria, pela não obtenção de uma alternativa para as medições isoladas das mesmas.

Apresentam-se, de seguida, as métricas escolhidas para análise:

- Utilização de banda larga
- Débito médio
- Distribuição de protocolos em uso
- Origem
- Número de *Sources* Distintas
- RTT
- *Workload analysis*:
  - N° médio de pacotes
  - Tamanho médio dos pacotes (min e máx)
  - Quantidade de tráfego total

## 5 Implementação

A nossa implementação foi feita através da utilização do *Wireshark* num *Laptop*, o qual se encontra a fazer *hotspot* para o dispositivo móvel *Android*, que será analisado. É possível observar pelo esquema abaixo a implementação referida.



**Figura 1.** Implementação

Alguns aspetos a ter em conta referem-se ao ambiente da captura. No *smartphone*, os procedimentos para a captura na aplicação e *browser* são apresentados de seguida.

### Aplicação

1. Desativar as notificações de todas as aplicações;
2. Entrar na aplicação *Youtube*;
3. Ativar a funcionalidade de navegação anónima;
4. Limpar a *cache*;
5. Sair da aplicação;
6. Fechar todas as aplicações abertas no dispositivo móvel;
7. Iniciar a captura;
8. Abrir a aplicação e procurar o vídeo pela barra de pesquisa;
9. Clicar no vídeo e pausá-lo de imediato.
10. Definir a qualidade do vídeo para 360p e retomá-lo;
11. Esperar que o vídeo termine e parar a captura no *Wireshark*.

### *Browser*

1. Desativar as notificações de todas as aplicações;
2. Fechar todas as aplicações abertas no dispositivo móvel;
3. Abrir o *Browser*;
4. Fechar todos separadores abertos, limpar a *cache* pela versão avançada e abrir apenas um navegador anónimo;
5. Iniciar a captura;
6. Abrir o *site Youtube*;
7. Procurar o vídeo pela barra de pesquisa;
8. Clicar no vídeo e pausá-lo de imediato.
9. Definir a qualidade do vídeo para 360p e retomá-lo;
10. Esperar que o vídeo termine e parar a captura no *Wireshark*.

## 6 Limitações identificadas

As maiores limitações com que o grupo foi confrontado foram o facto de não se encontrar uma variedade considerável de ferramentas funcionais para análise das métricas seleccionadas.

Exemplo de tal insucesso foi a recorrência ao *site Packet Total* para a análise das capturas, contudo esta alternativa não é fiável, uma vez que o *site* encontra-se não funcional. Posteriormente, instalou-se as ferramentas *CoralReef* e *Autofocus*, mas não se obteve sucesso na execução das mesmas.

Além disso, como nem todos os elementos do grupo reúnem condições para efetuar as capturas, e as mesmas não podem ser feitas em paralelo, isto resultou num longo período de testes.

## 7 Capturas

Os vídeos escolhidos para análise foram seleccionados em função das tendências, estes encontram-se listados abaixo, juntamente com o *link* respetivo.

- Eminem - Godzilla ft. Juice WRLD - Duração: 4.26 min  
[https://www.youtube.com/watch?v=r\\_0JjYUe5jo](https://www.youtube.com/watch?v=r_0JjYUe5jo)
- Entrevista com Blaya — The Noite (02/09/19) - Duração: 10.29 min  
[https://www.youtube.com/watch?v=yRI\\_zYaR6zQ](https://www.youtube.com/watch?v=yRI_zYaR6zQ)

## 8 Testes

O grupo realizou alguns testes iniciais para experiência, de forma a estudar o ambiente para posterior realização das capturas finais.

Após este processo, os resultados obtidos foram analisados com recurso às ferramentas *Wireshark* e *Tstat*.

Analisaram-se todas as métricas já referidas anteriormente através do *Wireshark*, excepto a métrica RTT, tendo sido esta analisada através do *Tstat*, precisamente com recorrência aos *logs* TCP gerados por esta ferramenta. Para tal, concebeu-se um *script Python* capaz de extrair facilmente os campos relativos à origem, destino e RTT's dos mesmos.

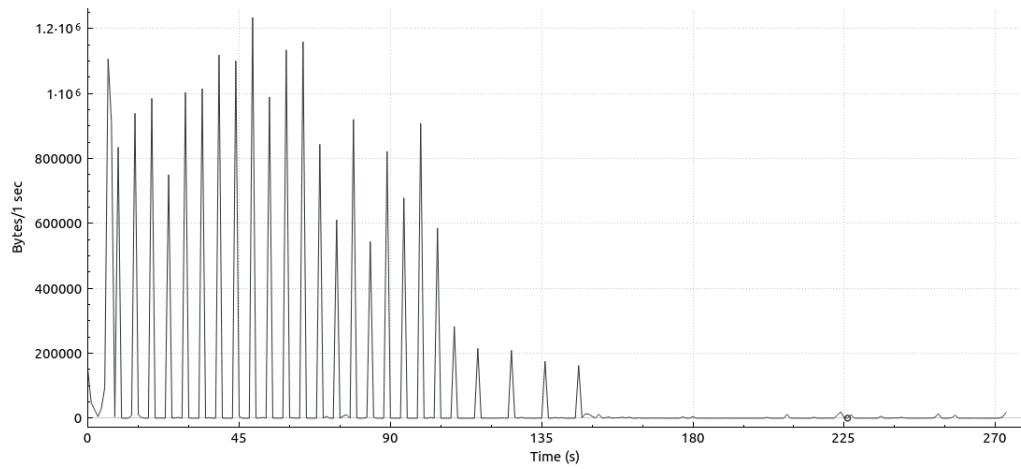
De referir que se tentou utilizar outras ferramentas tais como *PacketTotal*, *CoralReef* e *Autofocus*, contudo não se obteve sucesso na sua utilização.

### 8.1 Eminem - Godzilla (App vs Browser)

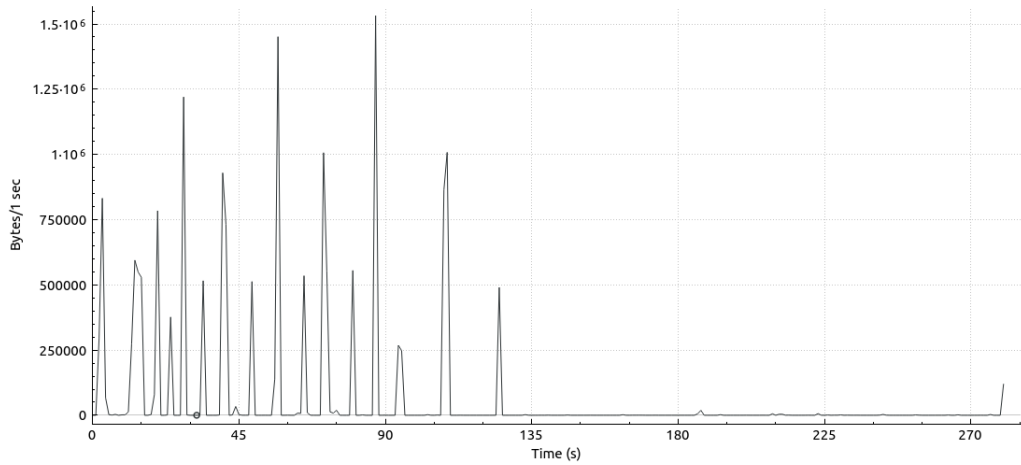
Inicialmente, geraram-se os gráficos referentes ao número de pacotes em função do tempo, em ambas as versões *app* e *browser*.

Através destes, pode-se constatar que as versões diferem na quantidade de pacotes enviados ao longo do tempo. Na versão *browser*, estes são enviados entre

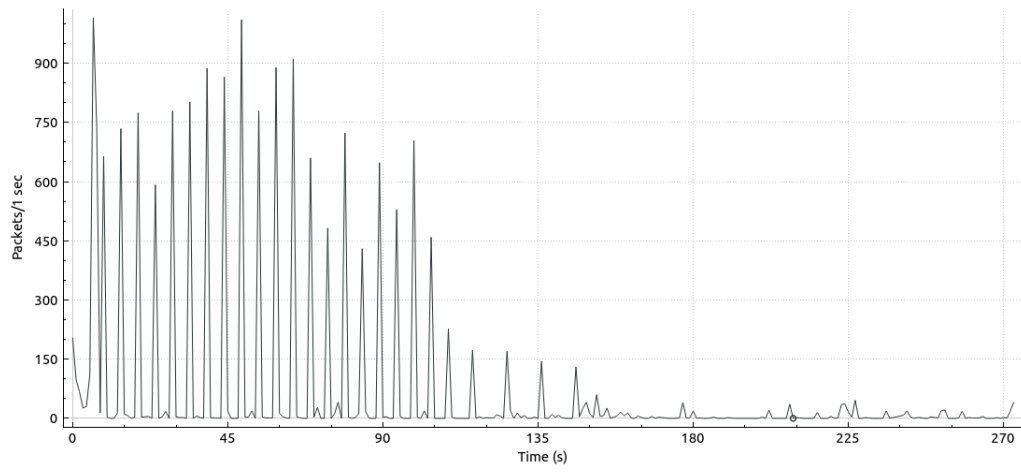
um intervalo de tempo de aproximadamente dez segundos, o que não se sucede na versão *app*, pelo que este apresenta um intervalo de tempo mais reduzido.



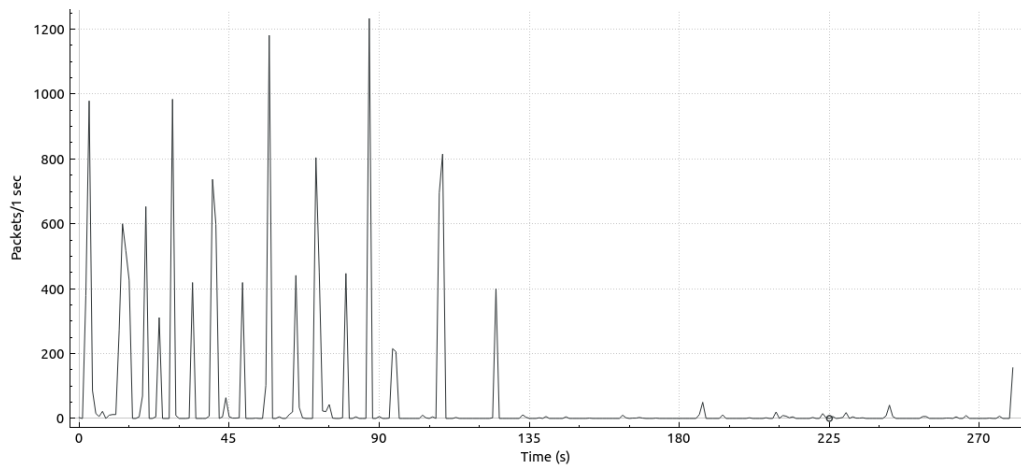
**Figura 2.** *Bandwidth* na aplicação



**Figura 3.** *Bandwidth* no browser



**Figura 4.** Packets na aplicação



**Figura 5.** Packets no browser

De seguida, apresenta-se a tabela correspondente à recolha de informação da captura relativa ao vídeo *Eminem - Godzilla ft. Juice WRLD*.

	<i>App</i>	<i>Browser</i>
<b>Total de conexões pelo Protocolo de Transporte</b>	133	95
Número de conexões UDP	96	57
Número de conexões TCP	37	38
<b>Total de conexões por Serviço</b>	65	67
Número de conexões HTTP	0	0
Número de conexões SSL	32	36
Número de conexões DNS	33	31
<b>Total de conexões por <i>Source IP</i></b>	165	126
Número de conexões pelo IP 192.168.137.231	121	93
Número de conexões pelo IP 192.168.137.1	40	33
Número de conexões pelo IP fe80::ac0a:993:63:3129	4	0
<b>Total de conexões por <i>Destination Port</i></b>	123	93
Número de conexões pela porta 80	1	0
Número de conexões pela porta 5353	3	1
Número de conexões pela porta 1900	30	2
Número de conexões pela porta 53	33	31
Número de conexões pela porta 443	56	59
<b>Total de pacotes</b>	18625	15293
Tamanho médio de pacotes ( <i>bytes</i> )	1175,22	1129,16
Tamanho do pacote mais pequeno <i>bytes</i>	54	42
Tamanho do pacote maior <i>bytes</i>	1514	1514
Total de <i>bytes</i>	21888472.5	17268243.88
Débito <i>bytes/s</i>	79955,40	61508,34

Nas tabelas que se seguem apresentam-se os RTT's (*Round Trip Time*) entre o cliente e os três servidores com quem estabeleceu maior número de conexões, na aplicação e no *browser* respetivamente.

### ***RTT* na Aplicação**

Cliente	Servidor	RTT médio cliente	RTT médio servidor
192.168.137.231	149.154.167.91	60.058700	80.640219
192.168.137.231	172.217.16.238	25.363923	15.557944
192.168.137.231	216.58.211.226	26.255562	11.684979

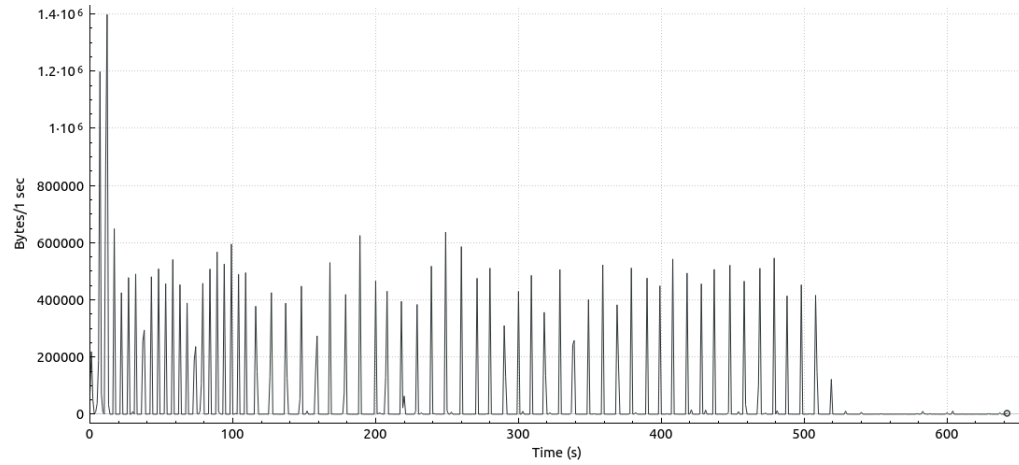
### ***RTT* no Browser**

Cliente	Servidor	RTT médio cliente	RTT médio servidor
192.168.137.231	216.58.201.174	30.385885	10.086368
192.168.137.231	216.58.211.237	29.074013	10.318445
192.168.137.231	193.126.242.78	12.198593	14.041046

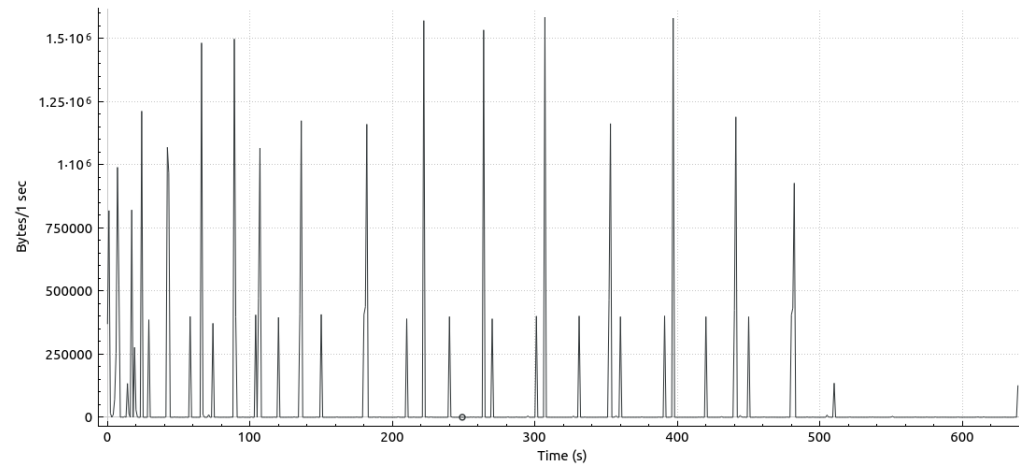


## 8.2 Entrevista com Blaya — The Noite (02/09/19))

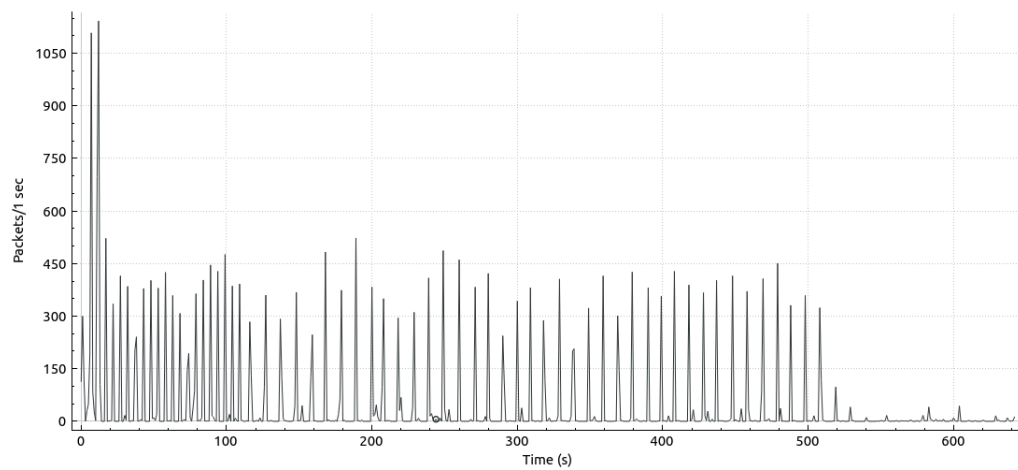
Tal como se verificou nas capturas anteriores, na versão *browser* são enviados mais pacotes em intervalos de tempo de aproximadamente dez segundos. Já na aplicação, o número de pacotes enviado é menor, mas mais uniforme e frequente.



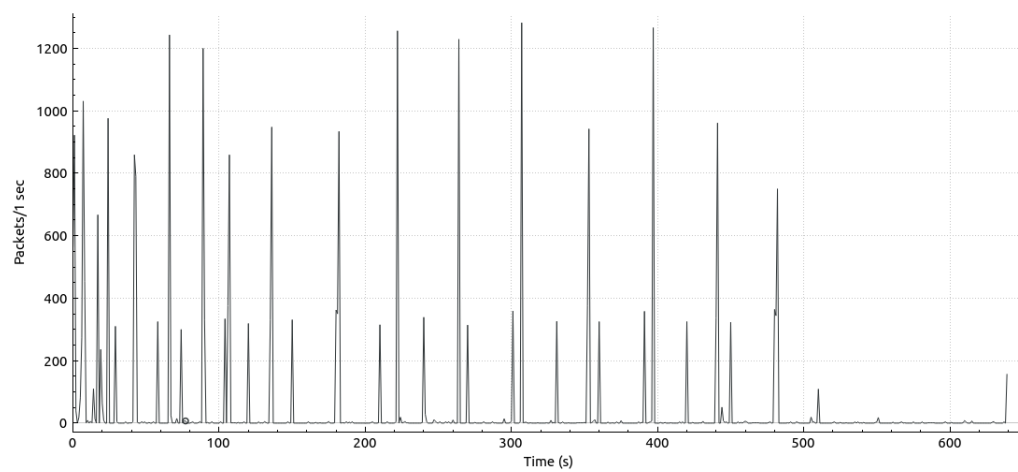
**Figura 6.** *Bandwidth* na aplicação



**Figura 7.** *Bandwidth* no *browser*



**Figura 8.** *Packets* na aplicação



**Figura 9.** *Packets* no browser

De seguida, apresenta-se a tabela correspondente à recolha de informação da captura relativa ao vídeo *Entrevista com Blaya — The Noite (02/09/19)*.

	<i>App</i>	<i>Browser</i>
<b>Total de conexões pelo Protocolo de Transporte</b>	192	131
Número de conexões UDP	153	86
Número de conexões TCP	39	45
<b>Total de conexões por Serviço</b>	69	90
Número de conexões HTTP	0	2
Número de conexões SSL	32	40
Número de conexões DNS	37	48
<b>Total de conexões por <i>Source IP</i></b>	228	179
Número de conexões pelo IP 192.168.137.175	173	121
Número de conexões pelo IP 192.168.137.1	49	56
Número de conexões pelo IP fe80::ac0a:993:63:3129	6	2
<b>Total de conexões por <i>Destination Port</i></b>	178	124
Número de conexões pela porta 80	1	2
Número de conexões pela porta 5353	3	3
Número de conexões pela porta 1900	70	5
Número de conexões pela porta 53	37	48
Número de conexões pela porta 443	67	66
<b>Total de pacotes</b>	29530	29029
Tamanho médio de pacotes <i>bytes</i>	1163,82	1160,60
Tamanho do pacote mais pequeno <i>bytes</i>	42	42
Tamanho do pacote maior <i>bytes</i>	1484	1514
Total de <i>bytes</i>	34367604,6	33691057,4
Débito ( <i>bytes/s</i> )	53507,65	52663,08

E agora, mais uma vez, as tabelas correspondentes aos RTT's (*Round Trip Time*) entre o cliente e os três servidores com quem estabeleceu maior número de conexões.

***RTT na Aplicação***

Cliente	Servidor	RTT médio cliente	RTT médio servidor
192.168.137.175	216.58.201.163	24.008600	21.283984
192.168.137.175	216.58.209.78	25.152784	11.156444
192.168.137.175	216.58.201.162	34.411080	27.762318

***RTT no Browser***

Cliente	Servidor	RTT médio cliente	RTT médio servidor
192.168.137.175	216.58.211.238	27.120588	7.498191
192.168.137.175	212.113.172.15	16.469338	14.602831
192.168.137.175	172.217.16.237	27.056328	16.744456

## 9 Análise Conclusiva

Perante os testes apresentados, procedeu-se agora a uma pequena reflexão geral dos mesmos.

Primeiramente, evidenciou-se que o total de pacotes na versão aplicacional é superior à da versão *browser*, em ambos os casos. Além disso, o mesmo se verifica com o tamanho médio dos pacotes, superior na aplicação. Isto traduz-se num valor total de *bytes* mais elevado também na plataforma aplicacional, comparativamente à *browser*. A junção destes fatores com o tempo de captura, resulta num débito superior na aplicação face ao *browser*.

Relativamente à *bandwidth*, observando os gráficos, verifica-se uma grande diferença entre as duas versões, que se reflete em qualquer uma das capturas. Pode-se visivelmente verificar uma maior uniformidade no uso de banda larga na versão aplicacional, pelo que só existem picos altos no início, mantendo-se constante até ao fim. Já na versão *browser* dá-se uma grande discrepância de valores ao longo do tempo, existindo períodos de tempo em que a largura de banda é nula e outros em que assume valores muito elevados.

Verifica-se também que, relativamente ao RTT (*Round Trip Time*), o *browser* apresenta valores menores do que a aplicação. Apesar da diferença não ser muito significativa, isto pode refletir-se numa utilização mais rápida, quando se acede ao *Youtube* via *browser*.

Após se analisar tudo o que foi descrito acima, a utilização do *Youtube* aparenta ser mais vantajosa com o *browser* do que com a aplicação. No entanto, os resultados obtidos são meramente representativos das capturas e métricas alvo de estudo. Para que fosse possível obter conclusões mais fiáveis, deveria ser feito um estudo mais aprofundado.

## Referências

1. Areal, N. G. (2019). Caracterização de Tráfego e Desempenho em Dispositivos Móveis
2. Wireshark. URL  
<https://www.wireshark.org>.
3. Tstat  
<http://tstat.polito.it/>
4. Chowdhury, S. A., Makaroff, D. (2012). Characterizing videos and users in youtube: A survey. Proceedings - 2012 7th International Conference on Broadband, Wireless Computing, Communication and Applications, BWCCA 2012, 244–251.  
<https://doi.org/10.1109/BWCCA.2012.47>.
5. Papadamou, K., Papasavva, A., Zannettou, S., Blackburn, J., Kourtellis, N., Leontiadis, I., Stringhini, G., Sirivianos, M. (2019). Disturbed YouTube for Kids: Characterizing and Detecting Inappropriate Videos Targeting Young Children.  
<http://arxiv.org/abs/1901.07046>.
6. Gill, P., Arlitt, M., Li, Z., Mahanti, A. (2008). Characterizing user sessions on YouTube. Multimedia Computing and Networking 2008, 6818, 681806–681806–681808.  
<https://doi.org/10.1117/12.775130>.