

# Ferramenta para monitorização de dispositivos móveis

## Trabalho Prático 2

João Pedro Neves Félix<sup>1</sup>[PG47336], João Manuel Silva de Amorim<sup>2</sup>[PG47317], and  
João Miguel Santos Sá<sup>3</sup>[PG47320]

<sup>1</sup> Universidade do Minho, pg47336@alunos.uminho.pt

<sup>2</sup> Universidade do Minho, pg47317@alunos.uminho.pt

<sup>3</sup> Universidade do Minho, pg47320@alunos.uminho.pt

**Resumo** Para o trabalho prático 2 foi-nos proposto construir uma aplicação capaz de monitorizar redes móveis. Ao longo deste documento vamos falar um bocado da aplicação gerada e ainda o que decidimos analisar e que conclusões tirámos dessa análise.

**Keywords:** Jitter · Length · RTT · Bandwidth

## 1 Introdução

Ao longo deste semestre tivemos contacto com várias ferramentas de monitorização de rede e a principal conclusão que tiramos é que cada ferramenta é desenhada com um propósito em mente. Isto foi o pensamento que levámos durante todo o processo de realização deste trabalho prático.

Desenvolvemos esta aplicação com um objetivo muito claro em mente, analisar a utilização de rede de jogos online, mais precisamente jogos assíncronos pois notámos que todas as análises existentes tratam de jogos online em tempo real (*Call of Duty*, *League of Legends*, *Apex Legends*, *Valorant*, etc). O jogo que escolhemos analisar é o *Super Auto Pets*.

## 2 Super Auto Pets

Porque a aplicação foi desenvolvida com o objetivo de analisar a utilização de rede de um jogo em específico (*Super Auto Pets*). Achamos portanto importante por começar por explicar como funciona o jogo.

*Super Auto Pets* é um jogo online onde o jogador deve comprar e vender animais de forma a criar uma equipa forte o suficiente para ganhar 10 jogos. Existem dois modos de jogo: o modo **arena** onde jogamos contra equipas de outros jogadores de forma assíncrona (os adversários não precisam de estar online) e o modo **versus** onde criamos uma sala privada para jogar contra amigos. Existem mais diferenças entre ambos os modos mas estas são as mais relevantes, falaremos das outras à medida que for necessário.



**Figura 1.** Menu Principal

Durante uma partida (tanto versus ou arena) existem duas fases, a fase de compra e a fase de batalha. Ignorando alguns pormenores, na fase de compra o jogador deve comprar e/ou vender animais para melhorar a sua equipa. Na fase de batalha a equipa atual do jogador luta contra uma equipa de outro jogador que esteja no mesmo turno. Durante esta fase o jogador pode apenas observar o decorrer da batalha (não é possível fazer alterações à equipa nem quaisquer outras modificações), daí se dizer que o jogo é assíncrono.



**Figura 2.** Fase de Compra e Fase de Batalha

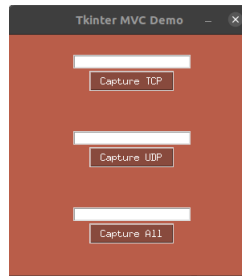
O jogo pode ser jogado em diferentes ambientes (IOs, Android, Steam) mas todos os testes foram realizados com o jogo a correr no browser [1] para normalizar e facilitar os testes.

### 3 A Ferramenta

Sabendo agora a aplicação que vamos analisar, temos que desenvolver uma ferramenta que se adequa ao tipo de análise que queremos fazer.

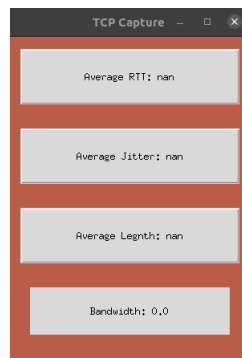
Em primeiro lugar procurámos uma ferramenta já existente de captura de pacotes que pudéssemos estender de forma a criar a nossa própria aplicação, desenvolvida para satisfazer as nossas necessidades. Decidimos usar o *tshark* com o pacote python *pyshark*.

Depois procurámos saber o tipo de protocolo de transporte que o jogo usa e chegámos à conclusão de que ele faz uso de pacotes TCP. Sabendo isto decidimos que a ferramenta teria uma opção de capturar apenas este tipo de pacotes. Adicionámos as opções de capturar apenas UDP e de capturar todos os pacotes pois não havia dificuldade acrescida em fazer tal coisa.



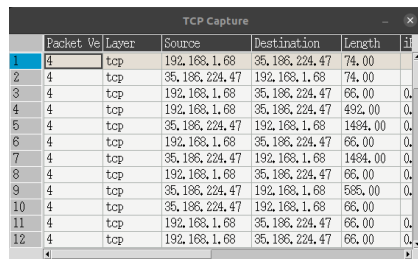
**Figura 3.** Menu Principal

De seguida pensámos nas métricas que queríamos medir e chegámos à conclusão de que o RTT, Jitter, Packet e Length e Bandwidth eram aquelas que mais nos interessavam.



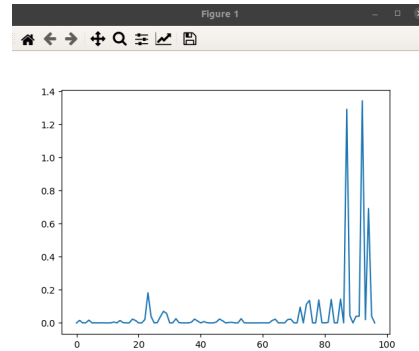
**Figura 4.** Menu das Métricas

Sabendo tudo isto, começámos finalmente a desenvolver a nossa aplicação. Decidimos que íamos fazer as capturas em tempo real (em vez de analisar ficheiros resultantes de capturas passadas), e que o utilizador ia poder escolher quanto tempo deveria a aplicação. Criámos uma interface gráfica muito simples para facilitar a sua utilização com o *tkinter*, tabelamos as capturas com os pacotes *pandas* [2] e *pandastable* [3], e mostramos os gráficos das capturas com o *matplotlib* [4]. Esta ferramenta segue o padrão MVC.



|    | Packet No | Layer | Source        | Destination   | Length  | Time |
|----|-----------|-------|---------------|---------------|---------|------|
| 1  | 0         | tcp   | 192.168.1.68  | 35.186.224.47 | 74,00   |      |
| 2  | 4         | tcp   | 35.186.224.47 | 192.168.1.68  | 74,00   |      |
| 3  | 4         | tcp   | 192.168.1.68  | 35.186.224.47 | 66,00   | 0.0  |
| 4  | 4         | tcp   | 192.168.1.68  | 35.186.224.47 | 492,00  | 0.0  |
| 5  | 4         | tcp   | 35.186.224.47 | 192.168.1.68  | 1494,00 | 0.0  |
| 6  | 4         | tcp   | 192.168.1.68  | 35.186.224.47 | 66,00   | 0.0  |
| 7  | 4         | tcp   | 35.186.224.47 | 192.168.1.68  | 1494,00 | 0.0  |
| 8  | 4         | tcp   | 192.168.1.68  | 35.186.224.47 | 66,00   | 0.0  |
| 9  | 4         | tcp   | 35.186.224.47 | 192.168.1.68  | 565,00  | 0.0  |
| 10 | 4         | tcp   | 35.186.224.47 | 192.168.1.68  | 66,00   | 0.0  |
| 11 | 4         | tcp   | 192.168.1.68  | 35.186.224.47 | 66,00   | 0.0  |
| 12 | 4         | tcp   | 192.168.1.68  | 35.186.224.47 | 66,00   | 0.0  |

**Figura 5.** Tabela de Captura



**Figura 6.** Gráfico do Jitter

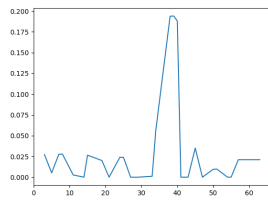
## 4 Análise

Vamos agora analisar o jogo *Super Auto Pets* com a ferramenta que desenvolvemos. Nós pretendemos analisar as duas fases diferentes tendo em conta as métricas que estamos a usar e ainda analisar também a utilização de rede quando o jogo é carregado pela primeira vez.

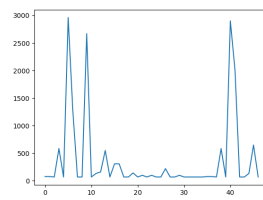
Ao inicializar o jogo no browser é nos mostrada uma barra com um valor de percentagem que vai aumentando à medida que o jogo vai carregando. Repetimos a medição várias vezes e em todas as vezes notámos que a barra para uns segundos nos 90%. A nosso entender isto acontece pois é nesse momento que os pacotes mais pesados são enviados como podemos ver pelo gráfico do RTT e pelo tamanho dos pacotes captados. Também o jitter se mostrou mais elevado nesta altura.

Passando agora para a fase de compra, existem 4 ações que um jogador pode realizar: comprar um animal/comida, vender um animal, dar re-roll nos animais disponíveis na loja e congelar um animal na loja. Realizámos vários testes sempre com dinheiro máximo e equipa vazia. Os resultados foram de certa forma diferentes do que aquilo que esperávamos.

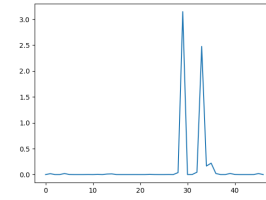
Para começar, a operação de congelar os animais não parece necessitar de conexão para ser realizada, isto porque em todos os teste que fizemos não houve



**Figura 7.** Tabela de Cap-  
tura



**Figura 8.** Gráfico do Ta-  
manho dos Pacotes



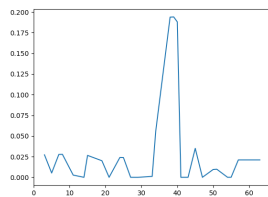
**Figura 9.** Gráfico do Ji-  
ter

troca de pacotes no momento dessa ação. Não conseguimos no entanto ter a certeza no seu impacto no que a toca a re-rolls.

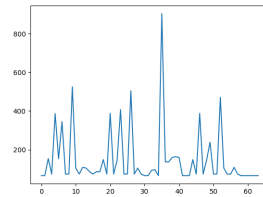
Em termos de tamanho de pacotes a ação que envolve troca de pacotes de maiores tamanhos foi constantemente re-roll, o que era de esperar visto que é a que tem maior impacto gráfico.

Já a falar de jitter e RTT não temos resultados conclusivos. De uma maneira geral o re-roll foi aquele que obteve os valores mais elevados mas isto não se verificou de forma tão acentuada como com o tamanho dos pacotes.

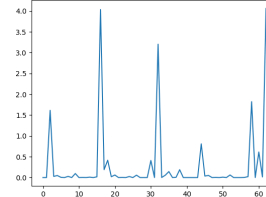
As 3 ações que registaram trocas de pacotes usara quantidades semelhantes destes quando efetuadas. Nos três gráficos mostrados abaixo o eixo dos yy mostra as métricas a medir e o eixo dos xx representa o número do pacote.



**Figura 10.** Tabela de  
Captura



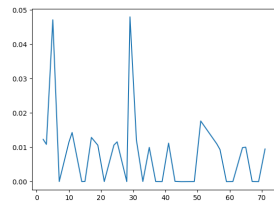
**Figura 11.** Gráfico do  
Tamanho dos Pacotes



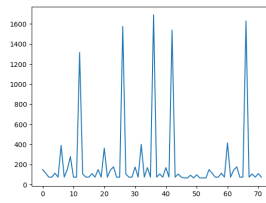
**Figura 12.** Gráfico do Ji-  
ter

Finalmente, falando da fase de batalha, não é possível distinguir muitas operações para além da procura por um adversário, a batalha em si e o menu de resultado (vitória, derrota ou empate). Estas 3 fases apresentam valores relativamente semelhantes mas notamos que estes valores tendem a atingir o seu pico na fase de batalha.

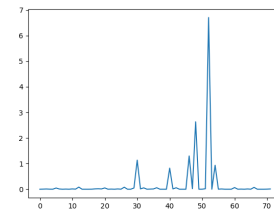
Fizemos vários testes com equipas de tamanhos diferentes para verificar se o tamanho tinha impacto na rede mas não notámos mudanças significativas.



**Figura 13.** Tabela de Captura



**Figura 14.** Gráfico do Tamanho dos Pacotes



**Figura 15.** Gráfico do Jitter

## 5 Conclusão

As ferramentas de medição são feitas com um certo propósito em mente, e a nossa não é diferente. Esta ferramenta foi desenvolvida para analisar o jogo *Super Auto Pets* na rede e como este funciona sobre TCP, a maior parte das funcionalidades focam-se nesta camada. O tamanho dos pacotes neste caso é uma medida imperativa pois como estamos a analisar um jogo, a parte gráfica influencia estes valores.

Fizemos várias análises do jogo tendo em conta a fase e o tamanho da equipa no momento da medição e somos capazes de fazer algumas conclusões. Em primeiro lugar é a iniciar o jogo que verificamos os pacotes de maior tamanho (isto porque é quando estamos a fazer o seu download). Já entre as fases de compra e batalha, a fase de batalha tende a ter um peso maior na rede, isto porque é quando acontecem mais coisas. Dentro da própria fase de batalha o tamanho da equipa não parece ter muito impacto nos valores captados.

Um ponto importante a mencionar é que os valores capturados durante a fase de compra não envolviam animais com habilidades que se ativam durante a própria fase de compra, por exemplo o sagui quando é vendido dá re-roll na loja.

## Referências

1. [teamwood.itch.io/super-auto-pets](https://teamwood.itch.io/super-auto-pets)
2. [pandas.pydata.org](https://pandas.pydata.org)
3. [pandastable.readthedocs.io](https://pandastable.readthedocs.io)
4. [matplotlib.org](https://matplotlib.org)