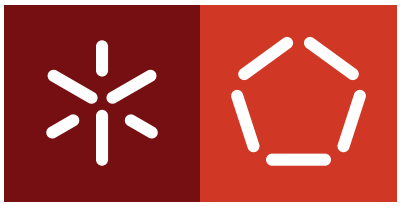
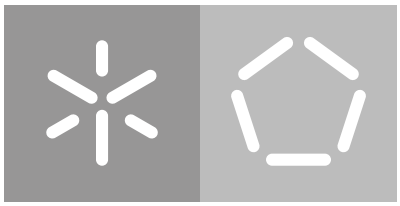


Universidade do Minho
Escola de Engenharia
Departamento de Informática

José Pedro Veiga da Silva

Caracterização de Tráfego Não Solicitado em Dispositivos Móveis

Novembro 2019



Universidade do Minho
Escola de Engenharia
Departamento de Informática

José Pedro Veiga da Silva

Caracterização de Tráfego Não Solicitado em Dispositivos Móveis

Dissertação de Mestrado
Mestrado Integrado em Engenharia Informática

Dissertação sob a orientação de
Professor Paulo Martins Carvalho
Professora Solange Rito Lima

Novembro 2019

Despacho RT - 31 /2019 - Anexo 3

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho

Atribuição
CC BY

https://creativecommons.org/licenses/by/4.0/

AGRADECIMENTOS

A realização desta dissertação de mestrado não se poderia tornar uma realidade sem importantes apoios de várias pessoas às quais estarei eternamente grato.

Em primeiro lugar, quero deixar um agradecimento especial ao meu orientador Professor Paulo Martins de Carvalho e à minha co-orientadora Professora Solange Rito Lima por toda a sua sabedoria e orientação essencial dada ao longo deste percurso.

De seguida, aos professores João Marco Silva, Kalil Araújo e Bruno Antunes por todas as ideias e sugestões que contribuíram para enriquecer a realização deste projeto.

A todos os amigos e colegas mais próximos deixo um agradecimento pelas discussões e ideias debatidas ao longo deste caminho.

Por último, mas não menos importante, à minha família, em particular aos meus pais e irmã, por estarem sempre ao meu lado em todas as etapas da minha vida, por não me deixarem desistir e por acreditarem sempre em mim e me incentivarem a continuar e a fazer mais e melhor.

Despacho RT - 31 /2019 - Anexo 4

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

RESUMO

Os *smartphones* cada vez mais desempenham um papel importante no quotidiano, seja para apenas comunicar ou realizar tarefas que exijam mais poder computacional. Este tornou-se um objeto indispensável na vida de muitas pessoas pelo que a sua utilização cada vez cresce mais. A este crescimento está associado um aumento de tráfego de rede devido às aplicações e serviços de Internet atualmente existentes.

Este aumento de tráfego, deve-se também ao desenvolvimento e crescimento das redes móveis do tipo 3G e 4G, que permitem o acesso à Internet quando se está fora de alcance de redes Wi-fi. Para tal acesso é necessário um plano de dados móveis que normalmente está limitado até um certo patamar. Esgotando este plano, o acesso à Internet a partir de redes móveis fica interdito ou é limitado a um certo débito muitas vezes bastante diminuto para as exigências das aplicações utilizadas.

Esta perturbação está associada ao consumo de largura de banda a fim de reproduzir conteúdos que venham da rede. Estes conteúdos muitas vezes estão associados ao funcionamento das aplicações, ou seja, tratam-se de conteúdo desejável, contudo, existem conteúdos aos quais não foram solicitados ao utilizador acabando por consumir o plano de dados.

Surge daqui o pretexto para a necessidade de um estudo e compreensão de todo o tráfego envolvido entre o dispositivo do utilizador e a rede que, vise mostrar ao utilizador final a identificar quantos dados consumiu e distinguir por tipo de tráfego envolvido. Em resposta a esta necessidade, uma metodologia sistemática será desenvolvida e proposta no decorrer deste trabalho considerando como objetos de estudo aplicações populares - YouTube, Facebook e Instagram - com potencial impacto no plano de dados do usufruidor. Assim sendo, a presente dissertação é uma contribuição para o campo de análise e caracterização de tráfego, abrindo caminhos no processo de identificação e medição de tráfego não solicitado pelo utilizador.

Palavras-chave: Tráfego de Rede, Dispositivos Móveis, Caracterização de Tráfego, Serviços de Internet, Aplicações Móveis, Análise de Dados.

ABSTRACT

Smartphones increasingly play an important role in everyday life, whether to communicate or perform tasks that require more computing power. This has become an indispensable object in the lives of many people so their use increases more and more. This growth is associated with an increase in network traffic due to the existing applications and Internet services.

This increase in traffic is also due to the development and growth of 3G and 4G mobile networks, which allow Internet access when out of Wi-Fi networks. Such access requires a mobile data plan that is normally limited to a certain level. Expiring this plan, access to the Internet from mobile networks is prohibited or limited to a certain rate which is often inadequate for the requirements of the applications used.

This disturbance is commonly associated with the consumption of bandwidth in order to reproduce contents downloaded from the network. These contents are often associated with the normal operation of the applications, i.e., expected contents, however, there are contents that were not requested by the user, for instance advertisements contributing to data plan exhaustion.

This justifies the need for studying and understanding the traffic involved between the user device and the network in order to assist the end user in identifying how much data has been consumed and distinguishing by the type of traffic involved. To answer this need, a systematic methodology is developed and proposed in this work considering as inputs popular user applications - YouTube, Facebook and Instagram - with potencial impact on the user data plan. Therefore, the present dissertation is a contribution in the field of traffic analysis and characterization, shedding light in the process of identifying and measuring traffic not requested by the user.

Keywords: Network Traffic, Mobile Devices, Traffic Characterization, Internet Services, Mobile Applications, Data Analysis.

CONTEÚDO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÓNIMOS

AP Access Point.
API Application Programming Interface.
ARP Address Resolution Protocol.

C2S Client-to-Server.
CDN Content Delivery Networks.

DDoS Distributed Denial of Service.
DNS Domain Name System.
DoS Denial of Service.
DPI Deep Packet Inspection.

FQDN Fully Qualified Domain Name.

HTTP Hypertext Transfer Protocol.
HTTPS Hyper Text Transfer Protocol Secure.

IANA Internet Assigned Numbers Authority.
IETF Internet Engineering Task Force.
IP Internet Protocol.
ISP Internet Service Provider.

LTE Long-Term Evolution.

MITM Man-in-The-Middle.

NetBIOS Network Basic Input/Output System.

OSI Open Systems Interconnection Model.

P2P Peer-to-Peer.

QoE Qualidade de Experiência.
QoS Qualidade de Serviço.
QUIC Quick UDP Internet Connections.

REST Representational State Transfer.
RFC Request For Comments.

S2C Server-to-Client.
SDK Software Development Kits.
SNI Server Name Identification.
SSDP Simple Service Discovery Protocol.
SSH Secure Shell.

TCP Transmission Control Protocol.
TLS Transport Layer Security.

UDP User Datagram Protocol.
UnPnP Universal Plug and Play.

VoIP Voice over Internet Protocol.
VPN Virtual Private Network.

WSDiscovery Web Services Dynamic Discovery.

1

INTRODUÇÃO

1.1 ENQUADRAMENTO

Os dispositivos de comunicação móveis têm evoluído de forma acentuada e contínua quer em termos de capacidade de computação e interação com o utilizador, quer em termos de recursos e formas de comunicação. Esta evolução reflete-se também ao nível dos sistemas operativos e aplicações desenvolvidas para as mais diversas áreas de atividade e interesse. Neste sentido o tráfego de dados móveis continuará a aumentar [7]. Com o número de utilizadores a crescer, as operadoras de redes móveis ou fornecedores de serviços de Internet (ISPs) oferecem planos de subscrição de acesso à Internet. Daqui surge uma necessidade de haver uma gestão por parte do utilizador no consumo do seu plano de dados móveis, pois muitas vezes está sujeito a encontrar **tráfego não solicitado** contribuindo para que este se esgote. Este esgotamento está relacionado com a necessidade de haver consumo largura de banda provocando por vezes uma latência na reprodução de conteúdo desejável. Este tráfego indesejado surge muitas vezes ligado a elementos de publicidade, mas pode ainda estar relacionado a atividades com finalidades maliciosas.

Atualmente muitos *smartphones*, independentemente do sistema operativo instalado, possuem aplicações já de origem que contabilizam o consumo de dados móveis. Essas opções são bastante úteis na medida em que indicam ao utilizador quanto consumiu no total e por aplicação. Contudo encontram-se limitadas em dizer ao utilizador o que de verdade aconteceu para que houvesse aquele consumo. Existem também aplicações de terceiros que contabilizam picos de consumo de dados móveis, mas só indicam quando este ocorreu contabilizando o que foi gasto. Pode-se constatar que aplicações multimédia envolvendo vídeo e voz serão mais custosas do que aplicações de "*instant messaging*". Nesse sentido, é conveniente observar se esses tipos de aplicações contêm tráfego indesejado, ou seja, tráfego que não esteja envolvido na sua finalidade e que envolvam um custo acrescentado ao seu normal funcionamento. Além do consumo no plano de dados, são observados outros inconvenientes nomeadamente a utilização de bateria a fim de reproduzir este tipo de tráfego e a ocupação de espaço no dispositivo levando a múltiplas transferências a fim de atualizar alguns conteúdos publicitários [30].

A caracterização de tráfego é bastante relevante neste contexto, pois permite realizar um estudo detalhado do tráfego envolvido e obter conhecimento sobre propriedades e comportamento de aplicações e serviços que fluem na rede. A partir deste conhecimento é possível traçar perfis de utilização e retirar características que possibilitam melhorar a Qualidade de Serviço (QoS) e Qualidade de Experiência (QoE) em produtos e serviços disponibilizados por aplicações instaladas nos dispositivos dos utilizadores.

1.2 Motivação e Objetivos

Observando um número crescente de pessoas com acesso à Internet através de dispositivos móveis, a caracterização de tráfego em redes móveis permite que existam diversas abordagens em como poder melhorar a interação e usabilidade da rede. Serve então de motivação, melhorar a qualidade de experiência do utilizador com o seu dispositivo móvel indicando e explicando a este, qual o tráfego envolvido durante a interação com as aplicações que utiliza e desfruta regularmente. Outra motivação é compreender como tráfego não solicitado (e.g. anúncios) chega aos utilizadores a partir das aplicações utilizadas e estudar a possibilidade de este querer evitar publicidade assim o possa fazer.

Neste contexto, o objetivo principal deste trabalho foca-se sobretudo na identificação e caracterização do tráfego não solicitado gerado pelo utilizador em dispositivos móveis a partir de aplicações. A partir deste, identificam-se como objetivos parciais:

- estudo detalhado dos vários tipos de tráfego envolvidos em aplicações móveis e demonstrar aspetos que estes possuam;
- estudo de mecanismos de obtenção e de análise de tráfego;
- caracterização de tráfego obtido dentro de um ambiente realista;
- concretizar um processo que permita indicar a um utilizador mais detalhes acerca dos dados móveis utilizados gastos nesta questão;
- analisar os resultados obtidos e retirar conclusões a partir do processo criado anteriormente.

1.3 ESTRUTURA DA DISSERTAÇÃO

No presente **Capítulo 1** é apresentado o contexto em que o trabalho se encontra. É neste capítulo que é exposta a motivação e delineação de objetivos a atingir para o desenvolvimento e concretização do projeto.

No **Capítulo 2** é apresentada uma abordagem ao estado da arte de caracterização de tráfego e tráfego não solicitado. Aqui serão expostos conceitos e fundamentos necessários ao desenvolvimento da solução tal como uma exposição de trabalhos relacionados com a temática ao longo da descrição.

O **Capítulo 3** apresenta o problema e os seus desafios a ultrapassar. No fundo, pretende explicar as dificuldades encontradas e indicar a estratégia adotada para a resolução do problema que levará à fase de desenvolvimento da solução.

Quanto ao **Capítulo 4**, este contém o desenvolvimento da solução do problema, explicando com detalhe decisões feitas, bem como o que foi feito e obtido.

De seguida, o **Capítulo 5** apresenta e reflete os resultados obtidos perante a solução realizada no capítulo anterior, de onde serão expostos vários pontos de vista para o trabalho.

Por fim, o **Capítulo 6** conclui esta dissertação fazendo uma avaliação tendo em conta os resultados obtidos e o trabalho desenvolvido. Daqui sairão possíveis projeções de futuros trabalhos que poderão partir deste tema.

2

ESTADO DE ARTE

Neste capítulo serão descritos conceitos relacionados com a temática de tráfego não solicitado, funcionamento de tráfego comercial e caracterização e análise de tráfego de redes. Estes conceitos servem de contextualização e aquisição de conhecimento e sensibilidade para a realização do trabalho.

2.1 TRÁFEGO NÃO SOLICITADO

2.1.1 *Definição*

A definição mais comum de tráfego indesejado é todo o tipo de tráfego não necessário para o bom funcionamento das aplicações ou serviços de rede. Como o nome indica, tráfego não solicitado, pode ainda ser tráfego não pedido pelo utilizador, cujo intuito final é consumir recursos computacionais da rede para benefício de terceiros que o colocaram [11]. Este tipo de tráfego é considerado por muitos um grande problema, pois a eficiência e utilidade da rede, bem como dos equipamentos finais dos utilizadores baixam.

Existem outras definições [23, 31] que se focam sobretudo no tráfego malicioso no seu sentido restrito, ou seja, está sempre associado a atividades criminosas com fim de realizar ataques de intrusão e sustentar uma economia paralela. Contudo uma evolução das tecnologias, permite criar formas de distinção em vários tipos de tráfego não solicitado. Além disso, existem situações onde o tráfego é demasiado e pode ser classificado como indesejado ainda que tenha intenções benignas tais como ataques de *Flash Crowds* numa linha de atendimento de urgências.

Neste tipo de tráfego em algumas redes, os operadores das mesmas expandem esta definição a tráfego VoIP não local (Skype), aplicações *peer-to-peer*(P2P) e ainda a serviços de *streaming* por forma a proteger os serviços locais do mesmo tipo (e.g. Netflix). Estas restrições, muitas vezes, são estabelecidas por governos por forma a integrar meios de controlo a certos serviços e páginas por forma a defender interesses de caráter económico e social. Além de governos, existem empresas às quais os direitos de autor são violados e daí

que aplicações de partilha de ficheiros por *peer-to-peer* sejam muitas vezes bloqueadas a fim de os preservar.

2.1.2 *Tipos de Tráfego Indesejado*

O tráfego indesejado, segundo o RFC 4948 [34], distingue-se sobretudo em três tipos distintos: Incómodo, Malicioso e Desconhecido.

Começando pelo tráfego Incómodo, aqui é possível encontrar as vulgares mensagens denominadas de *spam*. O *spam* surge muitas vezes associado a fins maliciosos contudo, a sua origem remonta a tráfego comercial pelo que, na sua maioria, tratam-se de anúncios publicitários não solicitados a produtos e serviços com os quais não se está relacionado. Estes anúncios tornam-se incomodativos quando se falam em redes empresariais pelo que existem medidas preventivas para tal. Focando num ambiente fora do empresarial, estes anúncios, por vezes, são evasivos e perturbam a experiência do utilizador com uma aplicação, surgindo muitas vezes sob forma de *pop-up*. Outras vezes, aparecem no meio da página num processo de consulta e ainda a meio da visualização de conteúdo multimédia, onde se pode observar vídeos publicitários. Estes anúncios têm bastante relevância neste tema pois, dependendo do seu tipo, isto é, se se trata de uma imagem ou de um vídeo, a sua apresentação requer um gasto de largura de banda e de outros recursos que pode vir a ser superior ao do normal funcionamento da página ou aplicação em questão.

O tráfego Malicioso trata-se de tráfego que normalmente causa mau funcionamento de certas aplicações e serviços, que podem comprometer dados de utilizadores explorando vulnerabilidades destas ou, até mesmo, a partir de vírus, *worms*, *spyware*, entre outros códigos maliciosos. Aqui também se encaixam os ataques por *Denial of Service (DoS)* e *Distributed Denial of Service (DDoS)* devido ao seu elevado potencial de bloquear serviços desperdiçando recursos e, consequentemente, aumentar o tráfego da rede. Normalmente este tipo de tráfego está ligado a atividades criminosas e por isso, do ponto de vista de segurança, possui um impacto inimaginável levando a que a sua recuperação seja custosa e frequentemente com pouco sucesso. Contudo, numa perspetiva do seu impacto num plano de dados móveis, esta torna-se difícil de avaliar pois apesar de às vezes a exposição a este tipo de tráfego seja extensa, a probabilidade de ocorrência de ataque depende de múltiplos fatores e não se consegue ter uma noção do que se trata ao certo, pois depende do ataque.

Por fim, existe o tráfego Desconhecido. Trata-se de tráfego que apesar de poder ser enquadrado nas categorias anteriores, não se sabe o seu propósito ou a sua origem. Tal classificação não pode ser levada a cabo, pois muitas vezes o tráfego é cifrado e por isso não se sabe se se trata de tráfego legítimo (ainda que possa incluir *spam*) ou de tráfego malicioso. Este tipo é importante pois pode conter dados sensíveis pelo que não é conveniente poder

ser inspecionado e classificado. Ao invés de dados sensíveis, torna-se indesejado quando se trata de um ataque "camuflado" e não existe forma de o poder combater.

Destes três tipos e fazendo uma análise abrangente, pode-se concluir que o tráfego com mais impacto potencial no consumo num plano de dados móveis e mais exposto ao público é o tráfego Incómodo gerado pelas aplicações instaladas no *smartphone*, pelos motivos acima descritos. Outra justificação que leva a esta escolha, é que normalmente o utilizador consegue ter maior perceção a tráfego incómodo do que por exemplo a tráfego malicioso. Pode-se comprovar que múltiplas páginas na Internet contêm publicidade relacionada com comércio de produtos e de aplicações. O mesmo pode ser dito das aplicações que se podem encontrar instaladas em qualquer *smartphone* e nas lojas de aplicações. Daqui se justifica o foco principal para tráfego incómodo, mais concretamente tráfego de caráter publicitário ou comercial.

2.2 TRÁFEGO COMERCIAL

2.2.1 *Origem*

O tráfego comercial é uma estratégia de *marketing* necessária para a promoção de ideias, produtos e serviços. Nos dispositivos móveis começou com a necessidade dos criadores de aplicações poderem obter proveito a partir de aplicações gratuitas, colocando publicidade para divulgação de produtos. A fim de obter rendimentos, recorre-se normalmente a uma rede de distribuição de anúncios por ser mais fácil de implementar e incorporar na aplicação, tal não impede os próprios de criarem a sua metodologia de entrega de anúncios.

Uma rede de distribuição de anúncios trata-se de uma ou várias empresas que fazem de intermediário entre publicitários e os *websites* ou aplicações que servirão de *host* para publicitar os anúncios criados. O objetivo principal de tais empresas é obter proveito através de empresas que necessitam de meios e estratégias de *marketing*, e que pretendam divulgar produtos que correspondam às preferências dos utilizadores com base em produtos comprados ou por páginas visitadas por estes. Por outro lado, quem contrata um serviço deste tipo, possui vários modelos de negócio por forma a obter receita:

- custo por clique - funciona mediante o número de cliques num anúncio dividido pelo número de vezes que este é mostrado;
- custo por milha - trata-se de um objetivo a atingir para que ganhe receita a partir daquela divulgação, neste caso até mil (daí o nome de milha);
- custo por ação - funciona mediante uma ação requerida ao utilizador podendo ser do género de subscrição e/ou preenchimento de inquéritos;
- custo por visualização - por cada visualização recebe uma percentagem, daí que esteja mais focado para campanhas de vídeo;

- custo por instalação - pelo número de vezes que o produto em divulgação foi instalado, o contratante recebe uma percentagem.

Para otimizar e melhorar a entrega, recorre-se normalmente a serviços de mediação para preencher o seu espaço com anúncios de forma estratégica e otimizada.

Contratar uma empresa deste género tem as suas vantagens e desvantagens. Das vantagens pode-se dizer que o dono de um *website* não necessita de criar e gerir servidores para a divulgação de anúncios. Além disso, como os anúncios começam a ficar mais direcionados e pessoais para os utilizadores, não necessita também de investir em *software* de *tracking* por forma a obter dados e estatísticas sobre os utilizadores. Contudo, a subscrição da este serviço no seu *website* obriga a uma perda de controlo sobre os anúncios publicados com o fim de obter o máximo de receita possível.

2.2.2 *Formatos de Apresentação*

Estas redes de distribuição, por norma, disponibilizam *Software Development Kits (SDK)* para que seja possível desenvolver aplicações que incluam anúncios. As redes de anúncios dão a opção aos programadores de integrar o formato de anúncio, como são entregues aos utilizadores e ainda a sua frequência de atualização. O formato é uma peça importante nos anúncios, pois é aquele que permite o envolvimento entre o utilizador e anunciante encontrando-se bastante formalizado. Destes, os mais comuns são os abaixo descritos.

- **Faixas** (*Banners*) - Este é o formato que mais facilmente é encontrado em múltiplas aplicações e *websites*. Consiste em mostrar faixas com imagens estáticas ou imagens animadas de produtos em qualquer local da interface, com maior incidência no fundo ou no topo desta. Muitas vezes ao ser clicado, este abre uma página que redireciona para a página do produto ou serviço em questão. A Figura 2.1 representa um anúncio neste formato de onde se observa o anúncio no fundo da interface.

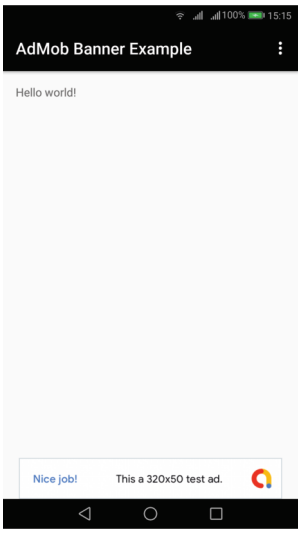


Figura 2.1.: Exemplo de anúncio em formato de Faixa.

- **Intersticial** - Leva o formato anterior mais longe pois, ao contrário de aparecer no topo ou no fim do ecrã este ocupa o ecrã na totalidade. É mais frequente encontrar este tipo de formato na transição de ecrãs dentro de uma aplicação. Estes possuem uma elevada taxa de cliques pois muitas vezes requerem que sejam clicados para que sejam fechados. Além disso, por serem apresentados em ecrã inteiro, acabam por ser mais cativantes e mais eficazes para divulgar o produto em questão. A sua eficácia diminui quando se torna demasiado incomodativo levando a utilizadores a evitar aquela aplicação por causa do formato usado. A Figura 2.2 indica a representação do formato Intersticial numa interface de uma aplicação.

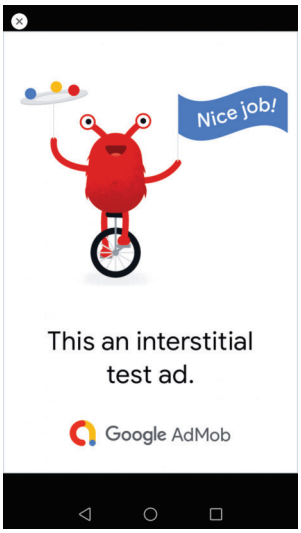


Figura 2.2.: Exemplo de anúncio em formato Intersticial.

- **Nativos** - Deriva do formato de faixas e intersticial, apresentando imagens ou vídeo, contudo apresenta conteúdo relacionado com a aplicação que está a utilizar, desde aplicações do mesmo tipo até aplicações do mesmo criador. Pretende ser menos intrusivo que outros formatos pois tenta encaixar na temática da aplicação. Este torna-se bastante competente pois a maioria dos utilizadores não o consegue identificar porque mistura-se com a página ou aplicação em questão com recurso aos mesmos formatos e diretrizes que esta utiliza. A Figura 2.3 realça o formato mais recorrente de um anúncio nativo que se mistura com o conteúdo da aplicação.

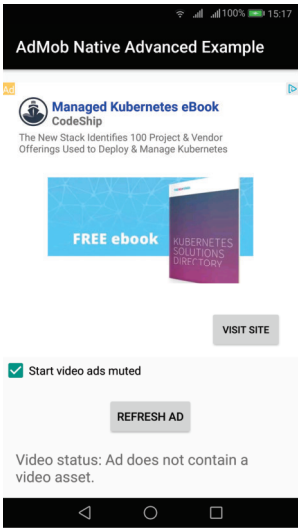


Figura 2.3.: Exemplo de anúncio em formato Nativo.

- **Vídeos** - Anúncios que se baseiam em vídeo- dão a conhecer produtos e serviços, sendo facilmente encontrados em aplicações de multimédia, jogos e de notícias. Estes vídeos normalmente não vão além de 60 segundos e pode haver dois tipos: *in-stream* e *out-stream*. No *in-stream*, o vídeo é apresentado antes, durante ou depois do conteúdo da aplicação em que está; um exemplo disso são os anúncios durante um vídeo na plataforma YouTube. Já no *out-stream*, o vídeo é apresentado em ambientes que não são de vídeo, por exemplo *feed* de aplicações de redes sociais, e que à medida que o utilizador faz *scroll*, um vídeo é reproduzido sem que tenha indicado a sua reprodução. Na Figura 2.4 é possível observar na imagem mais à esquerda uma opção "*Watch Video for 10 Additional Coins*" que ao ser carregado mostra o vídeo demonstrado na imagem à sua direita para dar a recompensa de 10 moedas no jogo. Este tipo de anúncios também é comum aparecer em aplicações obrigando utilizadores a visualizar publicidade em troca de algo na aplicação.

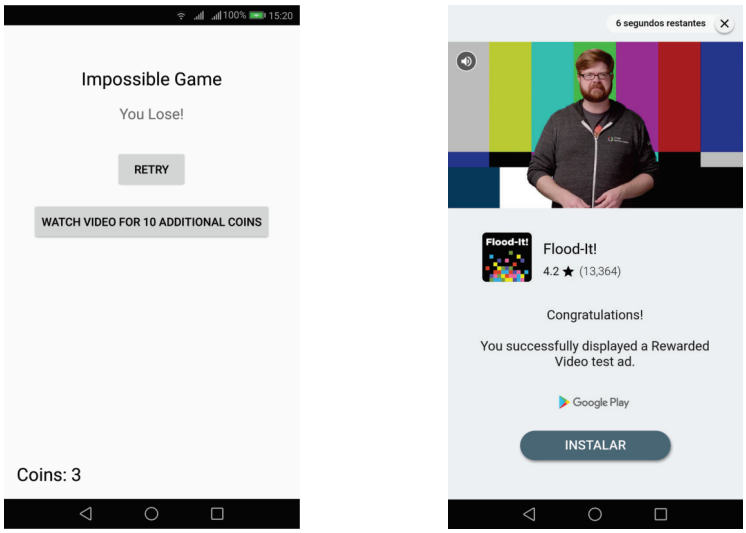


Figura 2.4.: Exemplo de aplicação com anúncio em formato de Vídeo que devolve recompensas.

- ***Offerwall*** - Trata-se de um espaço dedicado dentro das aplicações que pretende oferecer múltiplas ofertas, dentro ou fora da aplicação, em troca de alguma ação realizada pelo utilizador, como por exemplo descarregar uma aplicação ou visualizar um anúncio em formato de vídeo. Além de ofertas, dão a divulgar aplicações provenientes dos mesmos programadores da aplicação ou relacionadas com esta de outros criadores. Na Figura 2.5 pode ser visto que ao instalar aquelas aplicações, recebe-se moedas no jogo.

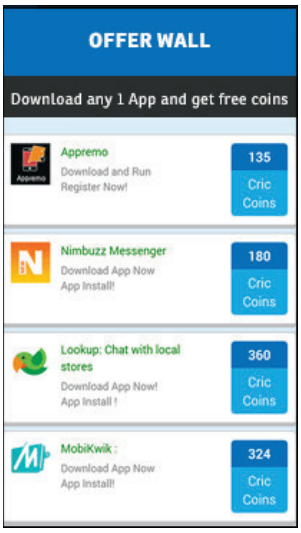


Figura 2.5.: Exemplo de anúncio em formato *Offerwall*.

2.2.3 *Ad Serving*

No final do desenvolvimento da aplicação, e depois de uma fase de testes, a aplicação é colocada no mercado. Uma vez disponibilizada no mercado, os utilizadores podem descarregá-la e utilizá-la. Este trata-se do primeiro passo para o *Ad Serving*, que se trata do conceito de entregar anúncios aos utilizadores finais de uma página *web* ou aplicação a partir de tecnologia desenhada com este propósito. No fundo são alguns passos essenciais que tornam a entrega de anúncios eficaz, podendo existir algumas diferenças. Esta diferenças refletem-se sobretudo sobre o facto se é uma empresa contratada ou se é o próprio a implementar a publicidade, ou ainda a plataforma na qual se visualiza, pois a partir de um computador, este possui muito mais espaço que um *smartphone* para colocar anúncios. Outro aspeto a ter conta é a latência numa plataforma móvel devido às redes móveis não podendo ser demasiado "pesado" para esta [33, 21].

Conforme o diagrama de sequência na Figura 2.6, abaixo apresentada, a aplicação trata de se ligar ao seu servidor próprio (primitiva 1) e este de seguida liga-se a servidores de entrega de anúncios, *Ad Servers*, (primitiva 2) por forma a transmitir conteúdos publicitários perante a interação com os utilizadores. Para a ligação, os protocolos usados pelos servidores para obter anúncios são baseados em pedidos HTTP através de `APIs REST`, utilizando sobretudo `HTTP GET` para outros servidores que possuam o conteúdo (ligação ao servidor do anunciante na primitiva 3). Estes servidores de anúncios possuem software de armazenamento de dados sobre conteúdo publicitário que procura a melhor forma de entregar anúncios aos *websites* e aplicações que os requisitem, recolhendo estatísticas sobre os mesmos (transações 4, 5 e 6). Uma vez feita a recolha de dados, a aplicação solicita diretamente ao servidor do anunciante anúncios atualizando-os numa frequência já estabelecida (representado por 7 e 8). Muitas vezes, os anúncios encontram-se integrados em redes distribuidoras de conteúdo (*Content Delivery Networks*) por forma a entregar conteúdo do mesmo local geográfico que o utilizador se encontre (transações 9 e 10). A utilização de CDNs adiciona eficácia na entrega de anúncios mais personalizados aos utilizadores, rapidez de resposta mediante o tipo de ligação (3G ou LTE) e operador. Além disso, coloca a possibilidade de entregar anúncios com maior resolução e mais elaborados.

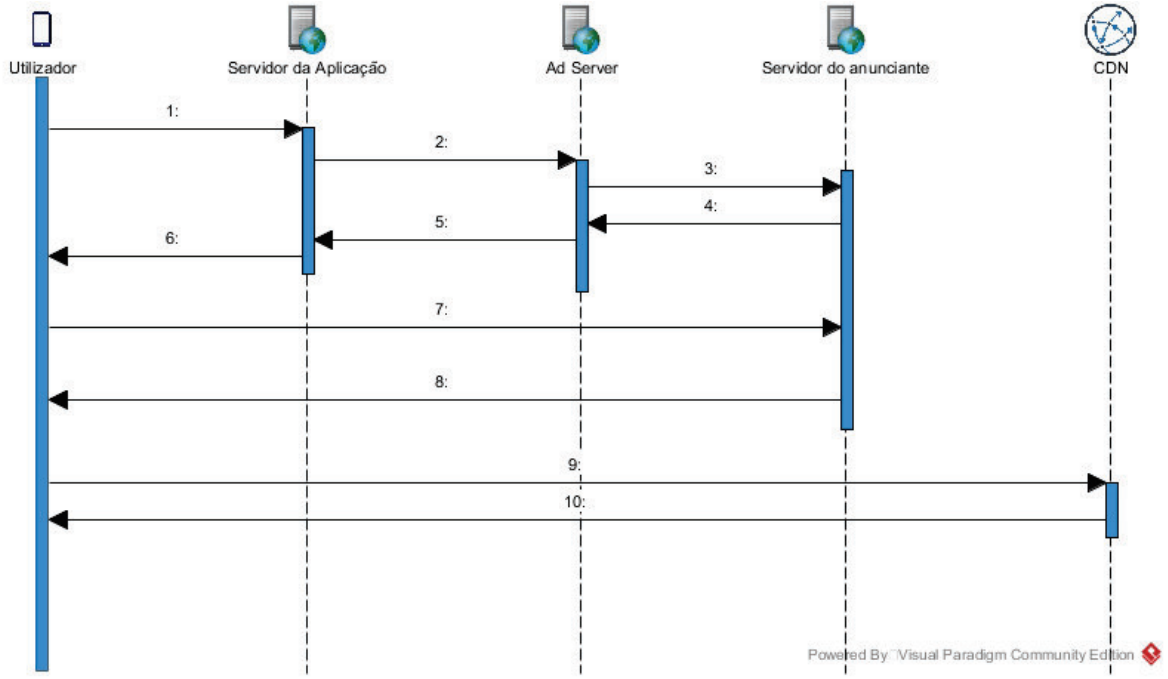


Figura 2.6.: Diagrama de sequência que demonstra o funcionamento de *Ad serving* em plataformas móveis.

2.2.4 *Principais Distribuidores*

O sistema operativo Android possui uma grande quota de mercado e pertence à Google, daí que é natural observar esta ser a principal distribuidora com a sua própria rede de distribuição designada de AdMob. Esta rede caracteriza-se por dar a oportunidade aos criadores de aplicações de obter receitas a partir das suas criações que sejam publicadas no mercado de aplicações da Google, o Google Play. Por outro lado, no que toca ao sistema operativo iOS também temos a Google como principal distribuidora.

Segundo [1], em ambos os sistemas operativos temos a rede pertencente ao Facebook, Facebook Ads, que pelo número de utilizadores das redes sociais pertencentes a este grupo (Facebook e Instagram), consegue ter bastante peso no mercado das redes distribuidoras de anúncios. Dentro das redes sociais, além do Facebook, temos os anúncios proprietários ou ainda anúncios que são divulgados dentro das mesmas utilizando protocolos proprietários da mesmas (Twitter).

Além de redes sociais, as redes distribuidoras podem também ser dedicadas a aplicações de jogos como é o caso da Unity, que ao fornecer o seu pacote de ferramentas de desenvolvimento de jogos para dispositivos móveis (*game engine*), permite colocar anúncios dando assim a possibilidade de retirar receita nos jogos.

2.3 CARACTERIZAÇÃO DE TRÁFEGO

A caracterização de tráfego dá a possibilidade de ter uma visão sobre uma rede e a sua usabilidade. Pretende oferecer formas para que se possa entender melhor as necessidades do tráfego a fim de melhorar a qualidade de serviço e experiência. Através da caracterização podem obter-se estatísticas sobre o tráfego da rede por forma a ter um melhor entendimento sobre a rede e o tipo de tráfego que nela circula. Para tal é necessária a sua captura e análise.

2.3.1 *Captura de Tráfego*

A captura de tráfego trata-se de uma técnica bastante utilizada no âmbito das redes de computadores. Consiste em intercetar pacotes de dados que se deslocam entre um ponto de origem e um ponto de destino numa rede. Quando um pacote é intercetado este é armazenado temporariamente para que depois seja analisado. Esta captura pode ocorrer de múltiplas formas e com múltiplos propósitos, tal como [4] refere.

Métodos de Captura

Numa rede, a captura pode ser feita sobretudo com recurso a três métodos distintos:

- *Switched Port Analyzer* (SPAN) - As portas SPAN, também designadas de portas de monitorização, são desenhadas para equipamentos de rede de nível empresarial cuja função é espelhar tráfego recebido nas outras portas desse mesmo equipamento. São vulgarmente utilizadas para captura de tráfego porque preservam ligações do tipo *full-duplex*. A principal desvantagem deste método é a necessidade de ocupar uma porta no *switch/router* no qual se está a fazer a captura. E existe outra desvantagem associada que é na sua configuração, nomeadamente pode levar a erros que são transportados pela rede e ainda a alteração do *timing* dos pacotes. Esta situação torna-se indesejável quando se quer recolher dados sobre o uso normal da rede ou detetar anomalias desta;

- *Terminal Access Points* (*Taps*) - Tratam-se de dispositivos de rede desenhados especificamente para aplicações de monitorização, sendo classificados como equipamentos de monitorização passiva. Ficam ligados em linha entre dois dispositivos de rede, entre um *router* e *firewall* ou ainda entre um *host* da rede e um *switch*, e preservam todo o tipo de ligações *full-duplex* atuais. Ao contrário do anterior método, efetuam a cópia exata do tráfego encaminhado pelo *switch*, armazenando-o externamente noutro equipamento com esse fim;

- Dispositivos em linha (*Inline devices*) - Como o nome *inline* indica significa que são dispositivos colocados "em linha" na rede tendo um comportamento bastante similar às *Taps*. A diferença entre estes e as *Taps* reside em possuírem uma maior complexidade e flexibilidade pois permitem manipulação e modelação de tráfego na rede. Este método é utilizado numa perspetiva dedicada à segurança pois permite investigar tolerância a falhas de segurança tornando-se bastante crítico numa rede empresarial. Além disso, o *deployment* destes na rede pode ser efetuado por um servidor ou outro tipo de *hardware* com esta funcionalidade instalada, em vez de ser um equipamento com esse propósito.

De notar que existe uma quarta opção que é a utilização de *Hubs*. Os *hubs* são equipamentos que repetem o tráfego para todas as interfaces de saída deste exceto aquela que o enviou. Isto significa que todos os equipamentos ligados a este conseguem observar todo o tráfego que circula. Esta técnica caiu em desuso pois apenas suportam *half-duplex* não sendo os mais apropriados para as redes atuais, enfrentando o problema de colisões. Outro problema relacionado a esta, a fiabilidade destes equipamentos fica reduzida quando ligado a uma rede de maiores dimensões cujos *uptimes* são bastante longos.

Abordagens de Captura

Os métodos anteriormente discutidos referem-se à recolha de tráfego de dados sem ter em consideração algumas estratégias de captura. De seguida são apresentadas algumas abordagens a ter em conta quando se faz uma captura de tráfego.

- **Centralizado** - Numa abordagem centralizada todo o tráfego recai apenas num dispositivo de captura. Como tal, torna-se vulnerável a ataques pois todo o tráfego recai num só local, contudo torna-se bastante simples de configurar.

- **Descentralizado** - Ao contrário da centralizada, esta abordagem assenta em múltiplos equipamentos que são distribuídos a realizar captura sobre a rede, sendo mais escalável que a anterior opção. Torna-se mais segura pois são múltiplos dispositivos a fazer captura sendo difícil de abranger a rede na totalidade. Como desvantagem exige um maior esforço de configuração e manutenção.

- **Aplicação de filtros** - Os filtros servem para obter uma visão confinada do tráfego recolhidos. Os filtros são bastante úteis quando se quer visualizar apenas, por exemplo, pacotes TCP que passaram por um *router* X.

- ***Full-packet capture*** - Em oposto à aplicação de filtros, uma captura pode ser simplesmente coligir todos os pacotes que passam na rede para posterior análise. Deste modo, pode-se filtrar no fim da captura e estudar as características pretendidas. De

notar que esta abordagem requere mais espaço em disco que a anterior, contudo é mais simples de implementar.

- **Participação Ativa** - uma participação ativa numa captura significa que, existe interferência por parte de quem está a realizar a captura. Este método consiste sobretudo em injetar pacotes na rede designadas por *probes*, normalmente por pedidos *Address Resolution Protocol (ARP)*, para que seja redirecionado para o local da captura ou ainda calcular atrasos e número de pacotes na rede. Tende a ser um método um pouco irrealista, pois força o tráfego a ser redirecionado ainda que possa ser configurado para ser o menos intrusivo possível e possa fornecer bastantes informações sobre as ligações.

- **Participação Passiva** - sendo passivo, significa que não participa no tráfego envolvido. Isto é bastante útil e vantajoso pois visualiza todo o tráfego que passa na rede. Contudo, existem algumas preocupações acerca do volume de dados armazenados que pode tornar difícil uma análise mais focada e ainda a cifragem dos dados que pode ser não tratável. Por vezes, são também usadas técnicas complementares de amostragem para reduzir o volume de tráfego capturado. Este método recorre a ferramentas muitas vezes denominadas de "*sniffers*" por "farejarem" a rede.

2.3.2 *Métodos de Caracterização de Tráfego*

Após a realização de uma captura sobre a rede a trabalhar, resta então classificar e caracterizar o tráfego. Com os métodos a seguir referenciados, será possível obter características do tipo de aplicação que o originou, se se trata de publicidade ou não, entre outros. O intuito final será identificar, descrever e representar como esse tipo de tráfego influencia num plano de dados móveis. De acordo com [20, 28] as principais técnicas de caracterização são descritas seguidamente, podendo existir variantes destes. Alguns trabalhos relacionados com esta temática são encontrados em [8, 27] e ao longo da descrição serão enunciados alguns projetos que envolvam a técnica descrita.

Análise à Porta (Port-Based Technique)

Este método consiste em verificar o cabeçalho de um pacote gerado por uma aplicação e fazer corresponder a porta de transporte em uso (maioritariamente TCP e UDP) com base nos registos da *Internet Assigned Numbers Authority (IANA)*. Segundo a IANA, quando se faz uma ligação por SSH a porta pelo qual o servidor se encontra à escuta é a 22 do TCP, por outro lado para aplicações *web* a porta utilizada é a 80 do TCP.

A sua implementação é bastante rápida e consome poucos recursos, pois basta captar o primeiro pacote de um fluxo, para fazer a identificação. Contudo, está dependente que todas as aplicações utilizem restritamente aquelas portas. Um exemplo onde esta técnica

falha - é em aplicações *Peer-to-Peer* (P2P), pois utiliza portas aleatórias e dinâmicas levando a que hajam muitos falsos positivos (aplicação não legítima a correr em portas conhecidas) e negativos (aplicação legítima a correr em portas não conhecidas) na sua classificação. Outra situação que a torna vulnerável é em situações em que o tráfego se esconde através de cifra sobre a camada IP sendo impossível identificar com protocolo de transporte e a porta usada.

Análise à Carga (Payload-Based Technique)

Esta análise é baseada na visualização do campo *payload* de um datagrama IP. Utiliza métodos conhecidos como *Deep Packet Inspection (DPI)* e daí ser conhecida vulgarmente por esta denominação. De acordo com [6], este método utiliza sobretudo uma análise de assinaturas para determinar e verificar diferentes aplicações. As assinaturas são umas *tags*, vulgarmente em formato de `string`, únicas e funcionam como um padrão que depois é associado a determinadas aplicações. De seguida, são comparadas numa base de dados e se existir, significa que pertence a um fluxo já conhecido. Daí poder ser classificado com aquela categoria, senão são referenciadas e armazenadas. Quer isto dizer que esta base de dados necessita de estar sempre atualizada com novas aplicações e novos desenvolvimentos de protocolos, pelo que se torna num inconveniente. Daqui decorre outra desvantagem. Quando não existe referência para a assinatura é necessário colocar na base de dados e para tal existe um peso computacional acrescido.

Os métodos DPI conseguem observar as Camadas 2 (Camada de Dados) e 3 (Camada de Rede) da pilha OSI e alguns conseguem ainda observar até à 7 (Camada Aplicacional) de tal modo que potenciam a existência de ataques por parte de terceiros. Para tal, existem medidas que tornam este método inútil utilizando métodos de cifra e técnicas de ofuscação nos pontos terminais.

Análise Estatística de Fluxos (Flow-Based Technique)

Esta metodologia baseia-se em explorar características estatísticas de fluxos de tráfego de rede. A partir de técnicas de medição de fluxos é possível obter, por exemplo, duração de pacotes e qual o seu tamanho. As aplicações possuem diferenças entre elas além da sua funcionalidade, pelo que estas características tornam-se distintas de aplicação para aplicação.

Para obter estas estatísticas, muitas vezes são conciliados conceitos de Inteligência Artificial, nomeadamente algoritmos de mineração de dados e algoritmos de *machine-learning* tornando-se num método que requer um conhecimento mais específico desta área que os anteriores. Além disso, existe uma componente de treino dos algoritmos que deve ser tomada com bastante importância e relevância para que a sua validação seja a mais precisa possível por forma a obter informação verdadeira. Daí que não existam ferramentas comer-

ciais que utilizem esta técnica, pois a sua precisão é mais baixa que DPI. Contudo, torna-se mais leve computacionalmente que DPI, na medida em que não necessita de verificar carga para obter uma assinatura, pois apenas necessita de fazer uma análise a nível de fluxo. Outra vantagem é, como o tráfego é classificado a nível de fluxo, esta técnica consegue lidar sem qualquer dificuldade quando se depara com tráfego cifrado.

Análise de Comportamentos (Behavior Analysis)

A análise de comportamentos observa o tráfego num patamar acima da rede, ou seja, olha para o tráfego nos sistemas terminais, sendo uma abordagem diferente dos anteriores métodos. A ideia principal centra-se na possibilidade de observar padrões de tráfego associados às aplicações que estão a correr num sistema terminal. O padrão aqui implica que todas as aplicações possuam um comportamento distinto de outras, como por exemplo, uma aplicação P2P pode ser identificada pela quantidade de *peers* que esta contacta a partir de uma única porta [17], enquanto uma aplicação servidor *web* será contactada por diferentes clientes a partir de múltiplas ligações paralelas [19].

Esta técnica possui as mesmas vantagens que a análise de fluxos, ou seja, é imune a tráfego cifrado e é leve no seu processamento, atingindo uma precisão semelhante a DPI com menos informação. Esta técnica torna-se um excelente método em redes mais restringidas e, por isso, é muito adotada em conjunto com estratégias de análise de fluxo. As sua desvantagens passam pela difícil escalabilidade pois necessita de ter garantias de que o fluxo corre nos dois sentidos (falhando em redes assimétricas) e que não hajam instabilidades na rede (balanceamento de carga, alteração de caminhos). Como o seu foco é em sistemas terminais ou *hosts* terminais não tem perceção da rede à sua volta.

2.3.3 *Reflexão sobre as técnicas*

Todos os métodos referidos possuem as suas vantagens e desvantagens. Devido à heterogeneidade existente de tráfego de rede, umas são mais indicadas para certas situações que outras. Complementar umas com outras não está fora de questão havendo já trabalhos que demonstraram resultados bastante favoráveis à temática de caracterização de tráfego.

A tabela 2.1 salienta os aspetos mais relevantes das técnicas estudadas apresentando vantagens e desvantagens do seu uso.

Estratégia	Vantagens	Desvantagens
Baseada em Portas	Fácil e rápida implementação Poucos recursos computacionais	Não deteta aplicações de porta dinâmica Falha com cifra aplicada sobre camada IP
Baseada em Carga	Elevada taxa de sucesso	Falha com tráfego cifrado Necessidade de BD com assinaturas atualizadas Levanta questões legais e de privacidade
Baseada em Estatísticas de Fluxo	Poucos recursos computacionais Analisa tráfego cifrado	Baixa taxa de sucesso
Baseada em Comportamento	Poucos recursos computacionais Analisa tráfego cifrado	Escalabilidade Não tolera falhas de rede

Tabela 2.1.: Vantagens e Desvantagens dos Métodos de Classificação

2.3.4 *Ferramentas de Classificação*

As ferramentas de classificação devem ter em conta as abordagens e estratégias mencionadas anteriormente. Estas ferramentas devem permitir a identificação da aplicação ou serviço que originou o tráfego em análise mediante fluxos ou características que estes possuam. O seu funcionamento pode ser em tempo real ou *offline*. Em tempo real, significa que à medida que captura ou monitoriza determina características do tráfego. Isto é muito útil em sistemas de deteção de intrusão onde uma ação rápida deve ser tomada. Já numa ferramenta de análise *offline*, esta permite que ao fim de uma captura se possa analisar o tráfego ao detalhe e retirar estatísticas sobre este.

Segunda a lista disponibilizada em [29] existem imensas ferramentas de análise e captura de tráfego com diferentes perspetivas e plataformas. Não sendo possível inumerá-las todas, vão ser destacadas algumas que utilizem estratégias de classificação acima descritas e algumas que irão ajudar a realizar este projeto.

As ferramentas baseadas em `SNMP`, `Netflow` e `RMON` são ferramentas pensadas para que utilizem propriedades dos *routers*, *switchs* e equipamentos terminais para retirar informações sobre a rede. São utilizadas sobretudo para monitorização da rede podendo vir a ser utilizadas para caracterização de tráfego.

Ferramentas como `Wireshark`, `Tcpdump` e `Tstat` são vulgarmente denominados de *sniffers* pelo facto de não interagirem com a rede na captura de tráfego e permitem uma análise

mais abrangente para caracterização da rede. Existe também o TIE (*Traffic Identification Engine*) que permite múltiplas estratégias de classificação e caracterização da rede.

Olhando para as estratégias, para uma análise de portas existe a `Nmap` que permite descoberta de *hosts*, análise de portas em *hosts* e redes específicas e ainda deteção de aplicações a correr. Ainda dentro de análise de portas existe também o `CoralReef`, uma ferramenta da CAIDA que nasceu em ambiente académico.

Para uma análise à carga, existem ferramentas como `L7-filter`, que recorrem a expressões regulares para fazer corresponder aos padrões de tráfego já conhecido [12]. Tal como referido anteriormente, esta técnica é muito usada para deteção de intrusão e para tal existem o `POCAD` e o `McPAD` que possuem esse foco.

Numa análise baseada em estatísticas, não existem ferramentas específicas, mas sim algoritmos de *machine-learning* que se encontram na biblioteca *open-source* WEKA e muitas outras. Existem alguns projetos que utilizam esta estratégia nomeadamente [24, 13].

Por fim, quanto à análise de comportamentos, existe o *BLINC* [18] que cria perfis de um *host* mediante uma captura em termos de destinos e portas pelo qual comunica, pode ainda identificar a aplicação que corre nesse mesmo *host*. Existem ainda trabalhos como [5] que utilizam esta estratégia para deteção de *malware*.

2.4 Sumário

Neste capítulo foi discutida matéria relativa ao tema da dissertação. Este capítulo enuncia o foco do projeto para tráfego comercial e é a partir deste que serão identificados todos os problemas e desafios do projeto.

Foi estudada a origem e funcionamento de como tráfego não solicitado chega ao utilizador final. De seguida, foram introduzidos conceitos acerca à caracterização de tráfego referentes aos métodos de captura e de caracterização de tráfego realçando pontos positivos e negativos das técnicas usadas. Daqui parte-se para a descoberta do problema e desafios associados, descritos no Capítulo 3.

3

PROBLEMA E DESAFIOS

3.1 METODOLOGIAS DE ANÁLISE

Neste capítulo são apresentados os desafios em relação ao problema de caracterizar tráfego não solicitado em dispositivos móveis num ambiente o mais fidedigno possível. Será ainda delineada uma arquitetura da metodologia adotada para a caracterização.

3.1.1 *Casos de Estudo*

Para a realização do projeto, é necessária uma noção de quais as aplicações e serviços que possuem bastante utilização e que lidem com quantidades favoráveis de dados a fim de verificar se existe tráfego não solicitado e o poder caracterizar. Para tal, foi necessário recorrer a várias estatísticas observadas no mercado de aplicações, Play Store, que refletem sobretudo o número de instalações e crescimento ao longo do tempo, com a finalidade de verificar se possuem boa utilização em ambiente Android.

Em [2], enunciado pela Figura 3.1, é possível observar uma tabela referente ao *ranking* de aplicações ordenado pelo número de instalações feitas até à data de consulta deste. Desta tabela, é possível verificar que a maioria das aplicações são disponibilizadas gratuitamente e daí que possuam uma maior adesão. Além de instalações, mostra também o crescimento que obteve em períodos de um mês e de dois meses (30 e 60 dias respetivamente) e indica a média de classificação dada pelos utilizadores.

rank	Title	Market icon	Total ratings	↓Installs	Average rating	Growth (30 days)	Growth (60 days)	Price
1.	YouTube		44,735,968	5000 M	4.37	1.8%	7.9%	Free
2.	Google Play services		22,768,720	5000 M	4.08	1.8%	5.5%	Free
3.	Google		11,317,663	5000 M	4.41	1.4%	5.5%	Free
4.	Maps - Navigate & Explore		10,212,830	5000 M	4.34	0.4%	1.6%	Free
5.	WhatsApp Messenger		89,067,734	1000 M	4.42	1.3%	4.0%	Free
6.	Facebook		87,171,800	1000 M	4.09	0.6%	2.0%	Free
7.	Instagram		81,887,896	1000 M	4.52	0.9%	3.4%	Free
8.	Messenger – Text and Video Chat for Free		66,373,258	1000 M	4.09	0.4%	1.8%	Free
9.	Clean Master - Antivirus, Applock & Cleaner		44,285,250	1000 M	4.66	0.1%	0.3%	Free
10.	Subway Surfers		30,159,797	1000 M	4.50	0.3%	1.4%	Free

Figura 3.1.: Top 10 de aplicações mais utilizadas.

Estes aspetos anteriormente referidos, indicam se existe utilização ou não, pois segundo [3], enunciado na Figura 3.2, existem aplicações que possuem números bastante altos de instalações mas que possuem fraca utilização por haver outras alternativas. Outra característica a retirar, é a tendência de ferramentas disponibilizadas pela Google que muitas vezes acabam por não serem usadas (por exemplo, o Google+ que foi recentemente descontinuado) apesar de, já virem embebidas no Android e daqui resultarem elevadas taxas de instalações. Outra observação daqui retirada é que muitas destas aplicações não possuem publicidade, até à data de consulta destas estatísticas. Exemplo disso são os Google Play Services, WhatsApp Messenger, Google Photos e Android Accessibility Suite que apesar de constituírem uma grande parte de utilizadores não serão considerados para o estudo. A retirar daqui também, é a utilização do *browser* da Google, o Google Chrome. Apesar de ser um condutor destes serviços não será considerado para o estudo porque os anúncios estão dependentes das páginas *web* que este exibe e não da aplicação *browser* em si. Este facto não retira a hipótese de poder ser utilizado para estudar algum serviço.

Market icon	Title	Category	Installs	Achieved on
	Google Play services	Tools	5000 million	2017-11-02
	YouTube	Video Players	5000 million	2018-12-16
	Maps - Navigate & Explore	Travel and Local	5000 million	2019-03-21
	Google	Tools	5000 million	2019-03-22
	Gmail	Communication	1000 million	2014-05-10
	Facebook	Social	1000 million	2014-08-28
	Google+ for G Suite	Social	1000 million	2014-12-27
	Google Text-to-Speech	Tools	1000 million	2015-03-11
	WhatsApp Messenger	Communication	1000 million	2015-03-12
	Google Play Books - Ebooks, Audiobooks, and Comics	Books and Reference	1000 million	2015-06-01

Figura 3.2.: Top 10 de aplicações mais instaladas.

Destes *rankings* é possível retirar também uma noção de quais os tipos de aplicações que devem ser tidos em conta para o estudo. Pode-se observar que, das diversas categorias de aplicações existentes, os tipos Ferramentas, Redes Sociais, Comunicação, Vídeo e Jogos são os mais utilizados em ambiente Android a nível mundial.

Daqui retiram-se que as escolhas mais acertadas serão Facebook e Instagram devido à sua forte adesão de utilizadores e de criação de dados. Estas plataformas são as formas mais utilizadas na categoria de redes sociais e comunicação e daí terem um peso bastante relevante para este estudo. Pois, em aplicações deste caráter, a criação de dados parte não só dos utilizadores destas, mas também de empresas que pretendam crescer e serem divulgadas. Deste interesse, verifica-se um aumento no que toca a tráfego comercial resultante de divulgação de produto que se tornam bons casos de estudo.

Por fim, outro serviço importante é o YouTube, a plataforma de vídeo mais utilizada à data. Este caracteriza-se por distribuir conteúdos que os utilizadores desejam consumir explorando os diversos gostos que estes possuam e assim fazer corresponder anúncios personalizados. Aqui os anúncios também partem de empresas que pretendem divulgar produtos e serviços, e que indicam à Google a sua intenção. De notar que, estas empresas possuem muitas vezes a versão *web*, apenas acessível por *browser*, e que mais à frente poder-se-á observar a importância que esta observação trará para o projeto.

3.1.2 *Metodologia e Desafios*

A partir de breves sessões de captura de tráfego utilizando as aplicações discutidas anteriormente, foram observadas algumas dificuldades que terão de ser ultrapassadas. Para tal, serão definidas algumas estratégias e ideias de como o método final irá recorrer.

Cifragem de tráfego

A principal dificuldade encontrada na caracterização de tráfego não solicitado prende-se com a cifragem do tráfego. Desta surgem outros pontos também importantes para apresentação de resultados, que serão expostos de seguida e que estão diretamente ligados a esta questão.

Desde a sua origem que a cifra serve fundamentalmente para proteger mensagens que possam ser intercetadas por terceiros. Neste caso é aplicada com a intenção de proteger os dados, quer de utilizadores, quer de servidores, aquando a comunicação entre ambos. Este componente é bastante importante pois cria uma camada de proteção que previne que informações não sejam expostas a terceiros e possivelmente roubadas. A estratégia de cifra mais comum de observar em ambiente Android é o uso de *Transport Layer Security (TLS)*, que se trata de um *standard* da *Internet Engineering Task Force (IETF)* que providencia uma comunicação fiável e segura entre dois sistemas terminais sobre uma rede não confiável[25]. O TLS é um protocolo de duas camadas principais *TLS Handshake* e *TLS Record*. O *TLS Handshake* é a camada encarregue de estabelecer contacto e criar autenticação entre cliente e servidor negociando chaves de segurança e/ou algoritmos de cifra antes de transmitir quaisquer dados. Já o *TLS Record* é a camada responsável por assegurar que a comunicação é segura por TCP. O TLS possui várias versões desde o seu desenvolvimento, algumas encontrando-se com previsão de descontinuação (1.0 e 1.1), onde atualmente conta com a versão 1.3.

Relativamente à cifra, outro aspeto relevante, são dispositivos cuja versão do Android é superior à versão *Nougat* (Android 7.0) onde as políticas de certificados autorizados e verificados mudaram e estratégias do tipo *Man-in-The-Middle (MITM)* (descrita na página seguinte) nem sempre resultam para resolver a cifra. Assim começa a surgir a expansão da utilização da técnica denominada de *Certificate Pinning* ou *Public Key Pinning* que se certifica de que o certificado trocado entre ambas as partes (cliente e servidor) é fidedigno e não se trata de um certificado forjado.

Captura de Tráfego

Do ponto anterior, consegue-se ainda identificar outro desafio, a estratégia de captura e análise de tráfego. Existem essencialmente duas estratégias de captura podendo ser no próprio dispositivo móvel (e.g. smartphone) ou tomada por um dispositivo com um *sniffer* tipo *Wireshark* instalado e ligado à mesma rede que o *smartphone*.

Para a primeira opção, existem aplicações como *Packet Capture*[14] e *tPacketCapture*[15] que coletam todo o tráfego envolvido estabelecendo uma ligação VPN para determinar que serviço realizou aquele pedido, tal como a Fig. 3.3 enuncia. Apesar de ser uma abordagem centralizada e isso tornar-se uma vantagem respetivamente à configuração, torna-se ineficaz quando exposta a aplicações que geram dados de maior volume (por exemplo:

YouTube e Facebook). O motivo que leva a esta falha é a dependência da ligação VPN com o servidor destas aplicações que não consegue lidar com tais pedidos. Além do volume, algumas aplicações implementam Cifragem de Dados, anteriormente explicado, que traz problemas na utilização desta estratégia. A esta abordagem existe o contratempo de a sua análise necessitar de ser exportada para outro dispositivo, ou se feita localmente seria necessária a criação de um serviço para tal. Outra desvantagem saída da maioria destas aplicações é, a necessidade de permissões de *root* a fim de desbloquear funcionalidades ou ainda ter permissões para realizar outro tipo de operações. Fazer *root* tem as suas vantagens pois permite ter um maior controlo como *superuser* e permite tirar partido de outras funcionalidades que possam ser bloqueadas de fábrica. Em contrapartida, ao ter novas funcionalidades desbloqueadas existem aplicações que se bloqueiam perante esta situação. Além disso, fazer *root* não é muito comum e exige um certo conhecimento para o utilizador normal que utiliza redes sociais e usa o seu *smartphone* para comunicação.

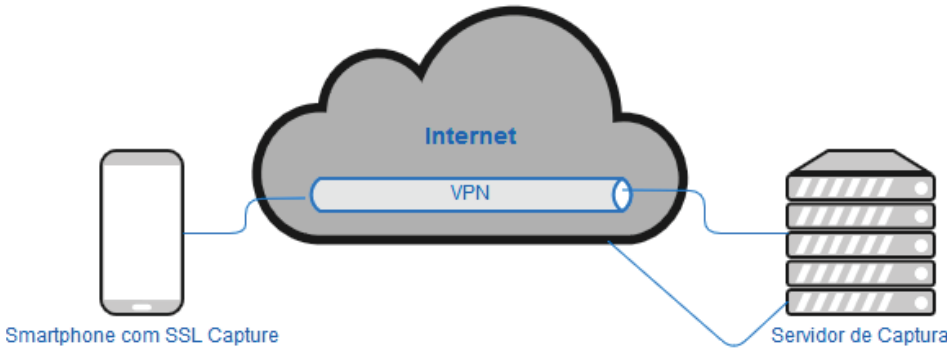


Figura 3.3.: Diagrama de captura por smartphone.

Assim sendo, a segunda solução parece ser a mais viável para capturar tráfego, pois não limita o volume de dados e não há necessidade de fazer *root*. Como representada pela Fig. 3.4, esta solução, passa por estabelecer uma rede local, a partir de um dispositivo que funciona como um *Access Point (AP)* desta rede. Este dispositivo possui instalado software de captura tipo Wireshark ou Tstat, encaixando-se num perfil de captura dentro dos *inline devices*. A partir daqui é possível obter todo o tráfego de um *smartphone* a partir de uma utilização normal deste. Esta técnica é muitas vezes denominada como *Man-in-The-Middle (MITM)* porque interceta todo o tráfego envolvido na comunicação e daí a referência de se encontrar no meio da ligação. De certa forma pode parecer similar à opção anterior na medida em que a captura é feita por um serviço externo e depois enviado para o smartphone, contudo agora a captura fica no lado do AP, o que facilita a posterior análise e caracterização.

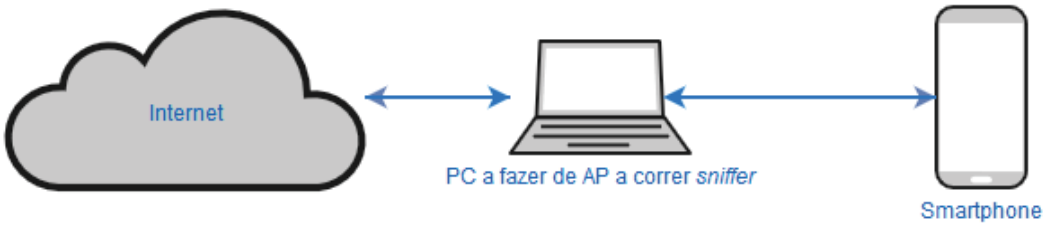


Figura 3.4.: Diagrama de captura por PC.

Análise Protocolar

A questão da cifra não está completamente resolvida com esta adoção de estratégia e por isso, surge a necessidade de estudar os protocolos de comunicação nos serviços em estudo. A maioria das aplicações utilizam protocolos HTTP para obter conteúdos publicitários. Contudo, após um breve contacto com as aplicações em estudo, observa-se que algumas apresentam outro tipo de protocolos além do HTTP, podendo dizer-se que não existe uniformização na entrega de anúncios, devido às diferentes redes distribuidoras e implementações destas.

Servem de exemplo as aplicações Facebook e YouTube. Para a aplicação Facebook é possível observar a obtenção de conteúdo, segundo a documentação deste[10], com recurso a HTTP/1.1 nas transferências de dados e HTTPS nos sistemas terminais. Sendo o HTTPS um protocolo aplicacional, num *sniffer* como o Wireshark este último é identificado como TLSv1.2 ou TLSv1.3 por TCP, na porta 443, daí que muitas vezes este seja denominado como HTTP sobre TLS. De seguida, a aplicação quando necessita de obter conteúdos fora do servidor ao qual está ligado, realiza pedidos DNS para descobrir onde estes se situam. Estes pedidos DNS são na maioria para os servidores ligados às CDNs que o Facebook possui, a título de exemplo a Fig.3.5, abaixo representada, mostra a obtenção de conteúdo apresentado na página ou *feed* inicial da aplicação. De destacar que o papel das CDNs é bastante relevante pois permite uma componente de personalização de conteúdos entregues ao utilizador final mediante a sua consulta de páginas, bem como uma personalização de anúncios mediante a localização.

192.168.137.1	192.168.137.65	DNS	110 Standard query response 0x3fe2 A z-m-scontent.fopo1-1.fna.fbcdn.net A 195.8.13.222
192.168.137.65	195.8.13.222	TCP	66 33783 → 443 [ACK] Seq=1272 Ack=21030 Win=125824 Len=0 TSval=6215260 TSecr=418480958
195.8.13.222	192.168.137.65	TCP	1414 443 → 33783 [ACK] Seq=37206 Ack=1272 Win=30720 Len=1348 TSval=4184809596 TSecr=6215
195.8.13.222	192.168.137.65	TLSv1.3	1414 Application Data [TCP segment of a reassembled PDU]
195.8.13.222	192.168.137.65	TLSv1.3	1414 Application Data [TCP segment of a reassembled PDU]
195.8.13.222	192.168.137.65	TCP	1414 443 → 33783 [ACK] Seq=41250 Ack=1272 Win=30720 Len=1348 TSval=4184809596 TSecr=6215
195.8.13.222	192.168.137.65	TCP	1414 443 → 33783 [ACK] Seq=42598 Ack=1272 Win=30720 Len=1348 TSval=4184809596 TSecr=6215

Figura 3.5.: Resposta de pedido DNS.

A presença de estes pedidos DNS seguidos de transmissão de dados indicia que a maioria dos conteúdos apresentados no ecrã do utilizador são assim entregues e consequentemente todos os anúncios publicitários. Após a observação de tráfego cifrado e após serem

clarificados alguns processos envolvidos neste, a sua quebra torna-se um caso difícil de contornar. Contudo, para o caso do Facebook, verificou-se que este possui o mesmo serviço para *browser*. A entrega de conteúdos em *browser* é idêntica à da entrega por uma aplicação, observando-se apenas diferenças no que toca a funcionamento dos serviços internos destas (conteúdo estático é mais vezes transferido em *browser* do que em aplicação devido à existência de *cache* e espaço em memória dedicado para esta, necessitando de atualizar apenas quando for estritamente necessário). Esta decisão foi tomada para dar a possibilidade de trazer dados relevantes ao anúncios e ainda permitir a utilização de outras ferramentas tais como, *Chrome Dev Tools*.

Considerando agora o YouTube, sendo uma plataforma de *video on demand* segue uma filosofia de ser dependente do estado da ligação e como tal deveria reger-se pelo TCP. Contudo, o TCP não é observado ao longo de múltiplas capturas de teste porque o TCP provoca atrasos na negociação de qualidade de serviço fim-a-fim. Então para dar resposta a esta situação, a Google começou a desenvolver o protocolo Quick UDP Internet Connections (QUIC). Como demonstra a Fig. 3.6, na ferramenta Wireshark pode-se observar a utilização deste mesmo protocolo sob a forma de GQUIC (*Google Quick UDP Internet Connections*), uma vez que a Google é o maior impulsionador de desenvolvimento e uso deste protocolo. O QUIC baseia-se no conceito de realizar uma entrega de conteúdos baseada no estado da ligação, ou seja, usar o conceito do protocolo TCP e aplicá-lo ao protocolo UDP a partir de múltiplas ligações (multiplexagem), melhorando assim o desempenho em aplicações que utilizem TCP na camada de transporte.

Deste ponto é possível retirar a ideia de que quando o vídeo é interrompido para visualizar um anúncio basta intercetar o fluxo relativo à reprodução deste. Mas, observando com mais detalhe conclui-se que não é possível verificar de imediato qual a *stream* relativa a este pois, mais uma vez, tem-se a cifra aplicada durante a sessão.

192.168.137.113	216.58.201.138	GQUIC	1392 Client Hello, PKN: 1, CID: 10115327076596101821
192.168.137.113	195.8.11.224	GQUIC	1392 Client Hello, PKN: 1, CID: 3760853582839078643
195.8.11.224	192.168.137.113	GQUIC	1392 Rejection, PKN: 1, CID: 3760853582839078643
195.8.11.224	192.168.137.113	GQUIC	1392 Payload (Encrypted), PKN: 2, CID: 3760853582839078643
192.168.137.113	195.8.11.224	GQUIC	70 Payload (Encrypted), PKN: 2, CID: 3760853582839078643
195.8.11.224	192.168.137.113	GQUIC	72 Payload (Encrypted), PKN: 3, CID: 3760853582839078643
192.168.137.113	195.8.11.224	TCP	74 51993 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1360 SACK_P
216.58.201.138	192.168.137.113	GQUIC	1392 Rejection, PKN: 1, CID: 10115327076596101821
216.58.201.138	192.168.137.113	GQUIC	1392 Payload (Encrypted), PKN: 2, CID: 10115327076596101821

Figura 3.6.: Protocolo QUIC.

Neste contexto, a utilização da ferramenta Tstat [26] revela-se bastante útil pois, perante o problema da cifra, consegue extrair a assinatura do servidor a partir da fase de *handshake* do TLS entre servidor e cliente, revelando o campo *server_name* envolvido. Esta ferramenta possibilita realizar análise durante uma captura em tempo real ou mediante uma captura já feita realizar uma análise posterior. Desta análise, são gerados *logs* acerca dos protocolos observados durante a captura e que podem ser ajustados mediante os parâmetros definidos

no ficheiro de configuração, `runtime.conf`. Obtida a assinatura, a análise passa agora por estabelecer uma caracterização de quais os vídeos vistos, tipos e tamanho de carga que possuam (vídeos de maior duração e resolução terão uma carga superior) e por fim a quantidade de vezes que os anúncios foram apresentados. Apesar da ferramenta Wireshark também possuir esta funcionalidade de extração de assinatura, a leitura posterior de fluxos e respetiva extração de informação torna-se um pouco mais dificultada que a do Tstat.

3.2 PROCESSAMENTO DOS DADOS

Após o levantamento de desafios, resta delinear uma arquitetura para processamento de dados. Todos os dados passarão por múltiplas fases desde a sua captura à obtenção de resultados finais. Passos como a obtenção de logs do Tstat, filtragem de anúncios e finalmente geração de estatísticas serão explicados no decorrer deste secção.

Os dados recolhidos partem de ficheiros `.pcap` ou `.pcapng` resultantes de sessões de captura na ferramenta Wireshark. Estes ficheiros estão associados às aplicações e serviços relevantes ao estudo e que serão sujeitos a várias etapas de tratamento.

Conforme dito anteriormente, o Tstat vai ser uma ferramenta complementar ao Wireshark utilizada para a obtenção de estatísticas e consequente caracterização do tráfego. Este devolve ficheiros `.txt` associados a um ficheiro de captura, `.pcap`, contendo informação acerca de um tipo de protocolo ou assunto. A título de exemplo, para tudo o que diz respeito a tráfego TCP, o Tstat devolve um ficheiro denominado `log_tcp_complete`, cujos parâmetros observados (IP cliente, IP Servidor, Porta utilizada, TTL, RTT, entre outros) encontram-se na primeira linha deste ficheiro e os fluxos e respetivas medições obtidas encontram-se nas restantes linhas. Estes ficheiros podem estar associados a tráfego TCP, UDP, HTTP, vídeo, *messaging* e ainda multimédia. Dentro destes `logs` gerados pelo Tstat os que mais se destacam são o `log_tcp_complete`, `log_udp_complete`, `log_tcp_nocomplete` e `log_http_complete`, por serem os protocolos mais utilizados para distribuição de conteúdo. De notar que, o ficheiro `log_tcp_nocomplete`, refere-se a tráfego TCP que terminou a ligação sem indicar as *flags* FIN e ACK ou que o processo de *TLS Handshake* tenha sido interrompido indevidamente.

De seguida, após a obtenção destes *logs*, é necessário verificar se possuem tráfego não solicitado. Para tal, foi necessário recorrer a listas que contém expressões regulares associadas a tráfego de anúncios, para poder identificá-los a partir de um URL ou nome de servidor retirado do *handshake* do TLS. Estas listas são vulgarmente disponibilizadas por aplicações e extensões de *browsers* que bloqueiem anúncios, como AdBlock Plus e uBlockOrigin. Daqui obtém-se um ficheiro que resulta de uma compilação de várias listas utilizadas por essas aplicações, denominado de *adlist.txt*. Este ficheiro conterá todas as expressões utilizadas para descobrir se um fluxo corresponde a obtenção de conteúdo não solicitado.

Para os serviços Facebook e Instagram, serão acrescentadas algumas etapas a este processo, nomeadamente, a utilização da ferramenta *Chrome Dev Tools*. Esta ferramenta será utilizada para facilitar a correspondência entre fluxo e conteúdo distribuído. Note-se que, dos fluxos retirados dos *logs* do Tstat apenas se retiram a quantidades de pacotes e de *bytes* transferidos, não havendo forma possível de distinguir tráfego não solicitado do solicitado. Esta ferramenta incorpora o *browser* da Google, o Google Chrome, que permite obter todos os pedidos realizados durante uma sessão de captura, funcionando de forma igual às ferramentas de programador em *browser* de PC mas, aplicado ao sistema operativo Android. Esta ferramenta dá como *output*, um ficheiro `har` (*HTTP Archive*) onde se retiram todos os pedidos HTTPS realizados. Contornando assim, a questão da cifra e permitir obter com precisão qual a percentagem do tráfego total se trata de tráfego não solicitado.

A última fase dá-se a partir de um programa desenvolvido em Kotlin. Este programa será discutido com mais detalhe no capítulo seguinte e é responsável pela classificação de tráfego e geração de respetivos resultados. Todos os resultados do Tstat são analisados e processados, resultando apenas os fluxos relativos a tráfego não solicitado. Para o caso do Facebook e Instagram, o ficheiro `.har` resultante da captura no Chrome Dev Tools será acrescentado à diretoria de *output* do Tstat para posterior processamento de dados. No ficheiro `.har` serão verificados todos os `urls` presentes e criado o ficheiro denominado `outputHAR.txt` que conterá todos os pedidos classificados como anúncios. Todo o processo de tratamento de dados discutido, encontra-se representado pelo diagrama de fluxo na Figura 3.7.

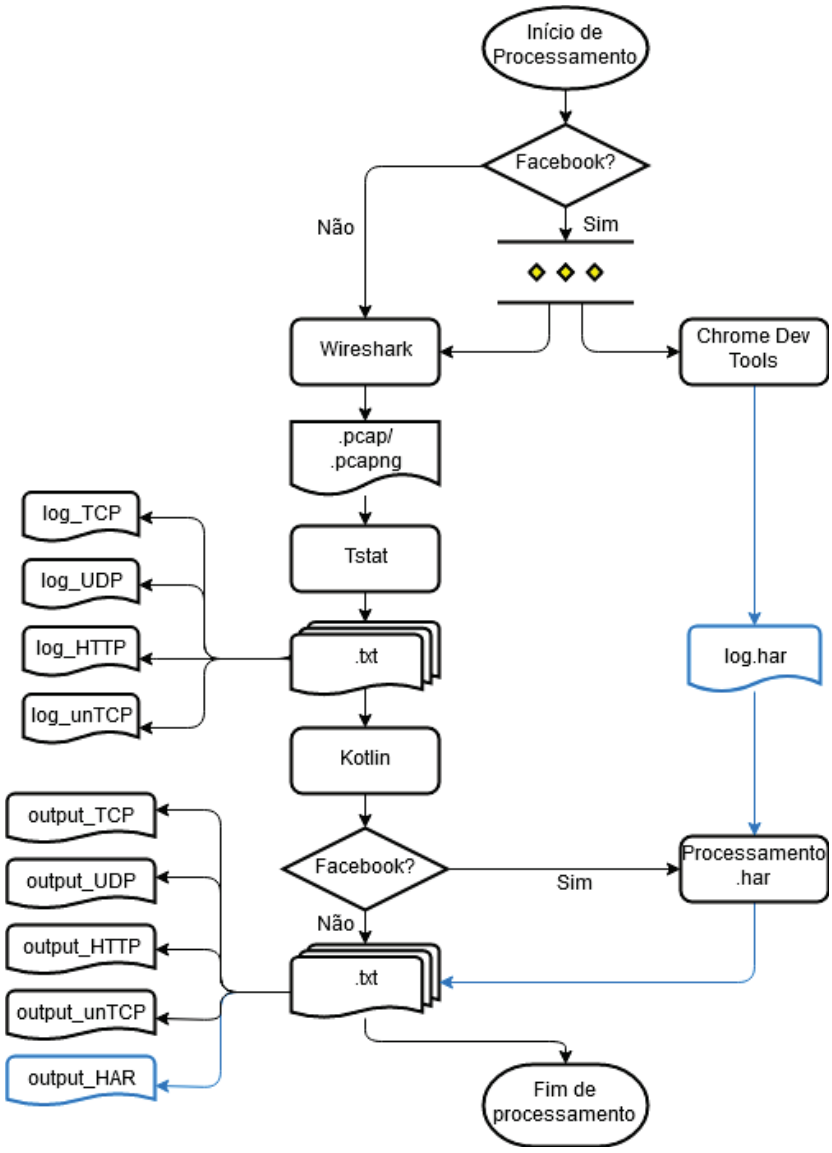


Figura 3.7.: Diagrama de processamento de dados.

3.3 PARÂMETROS DE CLASSIFICAÇÃO

De acordo com [30], foram estabelecidas métricas a fim de classificar o tráfego que foi observado. Daí que neste trabalho, surge a oportunidade de adotar e adaptar algumas métricas desse estudo a fim de obter resultados relevantes para o estudo de tráfego não solicitado e os poder expor de forma apelativa e intuitiva. Estes parâmetros têm em consideração do que será possível obter a partir dos resultados Tstat anteriormente divulgados.

3.3.1 *Volume*

O volume, representa o gasto de largura de banda num plano de dados móveis, sendo expresso em `Megabytes` (MB) ou `Gigabytes` (GB) dependendo da quantidade. Esta métrica, relaciona-se com a acumulação do tamanho de cada imagem ou vídeo que são apresentados na sua totalidade ao utilizador. A sua escolha justifica-se pelo motivo do impacto que tem para um utilizador, pois indica quanto de facto gastou no seu plano de dados móveis em tráfego que este não solicitou.

3.3.2 *Distribuidor*

O distribuidor do anúncio serve para ter uma noção da entidade que envia o tráfego e saber qual possui maior divulgação no mercado. Na sua maioria serão os servidores destinados a responder ao serviço, contudo será feito um balanço da origem deste anúncio, nomeadamente a companhia que representa e a sua relação ao serviço em estudo. De um outro ponto de vista, permite obter outros indicadores que possam ter relevância para o utilizador e criar estatísticas sobre que aplicações utilizam certos distribuidores, ajudando a entender como funcionam as redes distribuidoras perante o utilizador.

3.3.3 *Formato/Tipo de conteúdo*

Tal como enunciado na secção 2.2.2, existem várias escolhas de formato de anúncio. Ter uma noção de quais os que são mais utilizados é bastante relevante numa perspetiva de quem gere o seu plano de dados móveis, pois existem formatos que gastam mais largura de banda que outros. Aqui, serão retratados sobretudo os formatos de imagens estáticas, se, por exemplo, estão num determinado *aspect ratio* ou quantidade de pixeis, ou se no caso de vídeo determinar, por exemplo, o tipo de qualidade de imagem. No caso de vídeo, a resolução com que este é apresentado e duração são aspetos a serem considerados.

3.3.4 *Frequência*

Esta métrica está relacionada com a frequência com que os anúncios aparecem e são requisitados. Aqui serão exploradas formas de contabilizar anúncios que aparecem mediante uma ação na aplicação ou simplesmente contar as vezes com que são atualizados. Por consequência, acaba também por contribuir na contabilização de volume gasto no plano de dados móveis.

3.3.5 *Padrões observados*

Esta métrica diz respeito à regularidade com que são apresentados os anúncios, ou seja, se existem padrões de apresentação que permitam identificar se um anúncio aparece numa determinada posição sempre que a aplicação é aberta.

3.3.6 *Aplicações Gananciosas*

As aplicações gananciosas ou aplicações *greedy*, são aquelas que necessitam de um consumo elevado de largura de banda devido a se atualizarem com elevada frequência, descarregando múltiplas vezes anúncios. Pode-se dizer que, este indicador reflete a totalização de todas métricas referidas anteriormente e que no fundo indica a um utilizador quais as aplicações a evitar por forma a economizar dados móveis.

3.3.7 *Precisão de resultados*

A utilização de cifra impede dar garantia de certeza absoluta sobre a classificação dos serviços em estudo. Desta forma, este componente dá uma visão geral sobre o que passa no classificador de tráfego desenvolvido e se os resultados obtidos podem assegurar uma boa medição dos aspetos retirados. Considerando os valores previstos e verificando os resultados obtidos consegue-se avaliar a exatidão do processo de classificação.

3.4 SUMÁRIO

Neste capítulo foi discutida a abordagem ao problema observando as suas dificuldades e delineação de estratégias para as ultrapassar. Foram analisadas as aplicações e serviços com maior presença no mercado e maior utilização a fim de criar um conjunto restrito e representativo para investigar o impacto do tráfego não solicitado.

Neste capítulo foram também identificados parâmetros relevantes para apresentação de resultados, e que serão desenvolvidos e refinados na fase de implementação apresentada no capítulo seguinte.

4

DESENVOLVIMENTO

Neste capítulo é apresentada a solução para o problema e desafios analisados no capítulo anterior. Em concreto, será possível observar com mais detalhe as aplicações Facebook, Instagram e YouTube, selecionadas para o estudo e as suas exigências a fim de obter resultados relevantes à caracterização de tráfego não solicitado.

4.1 DECISÕES

Ainda que os padrões de anúncios sejam praticamente idênticos, tal como os exemplos anteriormente vistos, não existe uniformização de protocolos que asseguram a sua entrega ao utilizador final. Para tal, foram necessárias criar regras de classificação aplicadas a cada aplicação ou serviço em estudo. Nesta secção, serão demonstradas características de anúncios apresentados durante capturas de teste.

4.1.1 *YouTube*

A caracterização inicial da aplicação do YouTube passa por estudar o comportamento temporal, isto é, saber quando o anúncio foi apresentado. Esta ideia temporal não é tão fácil de concretizar ainda que, a nível conceptual, o pareça. Na realidade, existem diversos fluxos a ocorrer durante o tempo que supostamente foi apresentado apenas um vídeo anúncio, fazendo com que não seja possível realizar uma caracterização tão direta e facilitada. Para tal, o *Wireshark* será um complemento no processo de criar uma linha temporal de conteúdos que foram entregues.

A partir das estatísticas geradas em tempo real pelo *Wireshark*, mais especificamente as opções "*Statistics ->Conversations*", é possível observar o crescimento dos fluxos em relação ao número de *bytes* transmitidos entre servidor cliente. Deste indicador, consegue-se fazer corresponder o conteúdo em reprodução. Tendo uma correspondência fluxo-vídeo, é possível determinar qual destes pares correspondem aos anúncios em formato de vídeo.

Aplicando esta estratégia, foi possível observar a utilização de várias gamas de endereços IP para a entrega de diferentes conteúdos. Uma rápida comparação entre *browser* e aplicação, permitiu concluir que endereços IP que se encontrem na gama `195.8.0.0-195.8.255.255` correspondem na maioria a entrega de vídeo, na gama `172.217.0.0-172.217.255.255` serão de entrega de imagens e ainda na gama `216.58.0.0-216.58.255.255` serão de gestão de eventos (`javascript`, `json`) e apresentação de comentários abaixo dos vídeos visualizados. Claro está, que estas gamas apenas são referentes a Portugal onde, por exemplo, os IP de `195.8.0.0` a `195.8.255.255` pertencem à rede distribuidora de conteúdo do fornecedor de serviços de Internet MEO e, como tal, também não poderão ser excluídas eventuais exceções onde servidores externos a Portugal possam atuar na entrega de vídeo. A tabela 4.1 resume todas as gamas encontradas ao longo das sessões de teste.

Gama de Endereços IP	**Função**
`195.8.0.0` a `195.8.255.255`	Entrega de Vídeo
`172.217.0.0` a `172.217.255.255`	Entrega de Imagens
`216.58.0.0` a `216.58.255.255`	Gestão de Eventos

Tabela 4.1.: Gamas de Endereços IP encontradas.

Organizados os fluxos obtidos por gama e por inferência de conteúdo entregue, resta agora retirar a sua duração. É a partir da duração que se confirma se a *stream* corresponde a um vídeo em particular. Desta análise, é natural retirar que a duração do fluxo será diferente da duração do vídeo pois um está dependente da taxa de transmissão e o outro apenas será afetado caso esta seja baixa e leve a pausas indesejadas. Na eventualidade de se verificar correspondência é realizada a sua documentação retirando aspetos como título, `url`, resolução e sua duração. Já para a situação de se tratar de um vídeo de publicidade, este será acrescentado a todas as estatísticas resultantes da caracterização de tráfego não solicitado.

Por fim, para a classificação de imagens publicitárias, apresentadas na página de destaques ou inicial do YouTube, tal como a Figura 4.1 demonstra, foi decidido que serão considerados todos os fluxos quer do protocolo TCP, quer do UDP que contenham indícios de comunicação entre cliente e servidor de anúncios. A justificação de tal decisão deve-se ao facto de serem entregues com recurso a cifra onde não é possível determinar com certeza absoluta de que um fluxo tenha correspondência a uma e uma só imagem. Estes poderão conter múltiplas imagens que não sejam o anúncio observado.

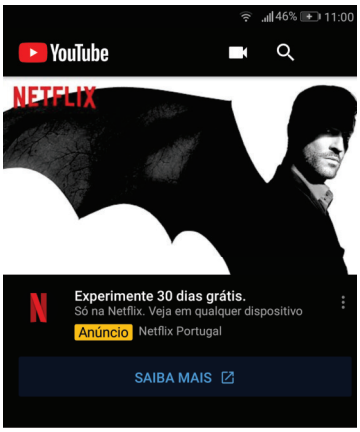


Figura 4.1.: Exemplo de Anúncio no YouTube.

4.1.2 *Facebook*

No capítulo 3, foram discutidos alguns pormenores que esta plataforma possui, desde a utilização de cifra até à adoção do seu serviço *web*. Considerando essa reflexão prévia, é investigado o funcionamento do serviço relativamente a tráfego não solicitado.

Numa primeira fase, foi necessário estudar a apresentação de anúncios no Facebook. Esta apresentação envolve múltiplos formatos desde imagens a vídeos que se enquadram nas publicações presentes no *feed* inicial. Visualmente estes anúncios encaixam-se num formato de anúncio nativo pelo facto de aparecerem com o mesmo formato de uma publicação normal. Dentro das imagens, existem diversos formatos que vão desde imagens quadradas ou retangulares até a imagens lado a lado, que será necessário caracterizar. No caso deste serviço, os anúncios são identificados por uma pequena descrição abaixo do nome da página que publicou aquele anúncio, conforme indica a Fig. 4.2.

Figura 4.2.: Exemplo de Anúncio no Facebook.

Nesta descrição observa-se escrito "Patrocinado"ou "*Sponsored*"(conforme a língua pré-definida no serviço). Isto é um indicativo de que a publicação se trata de um anúncio publicado pelo Facebook. Daqui, foi iniciado o procedimento de descoberta para esta particularidade. A partir do ficheiro `.har` obtido durante o processo de captura, foi possível concluir que se trata de um pormenor da página e acaba por estar embebido no método `javascript` que origina o evento de publicar o anúncio. De notar que, existem publicações com anúncios feitos a partir de páginas às quais existe relação de amizade ou de seguidor no Facebook, que não estão marcadas com essa descrição. Estas não serão consideradas para o estudo por trazerem uma componente subjetiva de interpretação, porque anúncios realizados por páginas com relação de amizade ou de seguidor podem ser considerados como conteúdo desejado (uma vez que possui relação com essa página) tanto como, conteúdo não desejado (por publicarem demasiados anúncios).

Foi iniciada uma nova fase de descoberta pelo conteúdo agora centrada no conteúdo que foi publicado. Mais uma vez recorrendo ao `.har`, foi possível retirar o conteúdo observado durante a captura. Deste estudo concluiu-se que a classificação mais imediata de anúncios assenta na resolução e proporção da tela (*aspect ratio*) das imagens apresentadas. Para o caso de vídeos, estes apresentam uma imagem em miniatura (*thumbnail*) de uma *frame* do vídeo, podendo também ser caracterizados comparando todas as *frames* à procura da miniatura. Na procura da imagem em todas as *frames* do vídeo foi decidido criar uma pequena *script* em Python que é chamada durante o processo de caracterização por parte do programa em Kotlin, que recebe como *input* todas a imagens classificadas como anúncio e todos os vídeos encontrados durante a captura. De seguida, a *script* divide os vídeos em múltiplas

frames e faz a comparação entre imagens com recurso à biblioteca OpenCV em Python, que possui a funcionalidade de poder comparar imagens e verificar se são iguais. No caso de sucesso, o tamanho do vídeo é acrescentado a todo o *payload* classificado como anúncio e será acrescentado aos formatos encontrados aquando a captura.

Ao longo desta fase, é de realçar uma particularidade encontrada aquando a análise do ficheiro *.har*. Os *urls* identificados utilizam com muita regularidade HTTPS, quer isto dizer que de vez em quando estas ligações expiram caso não sejam ativas com frequência. Isto serve para demonstrar que o processo de classificação não pode ser muito moroso para que não expirem as ligações e não se poderem retirar conclusões válidas para o estudo.

4.1.3 *Instagram*

O Instagram é um serviço pertencente ao Facebook, o que sugere que a estratégia de caracterização será a mesma. Contudo, observando o serviço *web*, constata-se que este não possui as mesmas funcionalidades que a sua própria aplicação.

Fazendo comparação entre serviço *web* e aplicação, os espaços onde existem anúncios na aplicação, mais especificamente publicações com a indicação de "Patrocinado"ou entre diferentes pessoas que publicaram "*Instagram Stories*", não estão presentes em *web*, concluindo que, de facto, não existe tráfego não solicitado no serviço *web* do Instagram.

Pela observação da versão *web*, apenas são apresentados publicações de páginas e pessoas às quais existe ligação (Seguidores). Ainda que estes possam conter publicidade por meio de publicação de fotografias ou vídeos, a sua caracterização está sempre dependente do fator subjetivo discutido anteriormente para o Facebook. Ao contrário do que acontece na aplicação, tal como a Figura 4.3 representa, não existe qualquer publicação que possua a indicação de "Patrocinado".

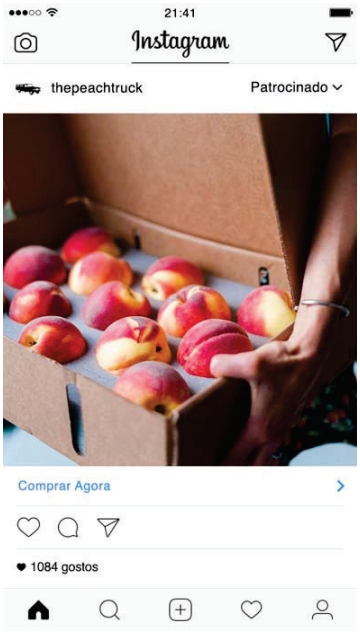


Figura 4.3.: Exemplo de Anúncio no Instagram.

Tal facto deve-se ao alvo de utilizadores, ou seja, o Instagram é um serviço cujo alvo de utilizadores prefere aplicação dedicada e não um serviço *web* que seja utilizado num *browser* e seja necessário fazer *login* sempre que deseja consultar este serviço. Além disso, as funcionalidades de publicação no Instagram ficam mais facilmente acessíveis quando é utilizada uma aplicação do que usando um *browser* onde podem surgir problemas de permissões de utilização de câmara ou de segurança.

Perante este comportamento identificado, a caracterização de tráfego não solicitado será mais dificultada e não conterá os pormenores que são possíveis de obter no Facebook. Assim a análise será realizada através de observar diferenças entre Facebook e Instagram no que toca a tráfego não solicitado.

Por fim, numa fase de testes, foi observado que a aplicação Instagram utiliza cifra na camada de transporte. Destes fluxos é possível deduzir quantos fluxos correspondem a entrega de conteúdos, obtendo as gamas dos endereços IP e nomes de servidores responsáveis por esta função. Destes indicadores, são conseguidos todos os resultados para comparar o processo de entrega de conteúdos com o Facebook. Como será ilustrado, apesar de serem muito similares na entrega de conteúdos, possuem claramente as suas diferenças.

4.2 IMPLEMENTAÇÃO

Toda as decisões discutidas na secção anterior foram implementadas no programa desenvolvido em Kotlin referido em 3.2. Ao longo desta secção serão explicados os processos de análise a cada *output* por parte do Tstat.

O processo de caracterização é desencadeado por um processo principal, *main*, que começa por verificar que ficheiros foram gerados pelo Tstat e fazê-los distribuir por tarefas (*threads*) responsáveis pela análise desses mesmos ficheiros.

Segundo o ficheiro de configuração `runtime.conf` do Tstat, existem várias métricas que se podem retirar a partir dos fluxos. Muitas destas métricas não são utilizadas por não trazerem relevância para o estudo, mas que são importantes para obter outras com maior grau de importância. Desta filtração, pode-se encontrar uma definição de fluxo para cada protocolo analisado.

4.2.1 *Análise TCP*

O Tstat realiza uma operação de distinção entre um fluxo TCP completo e um fluxo TCP incompleto. Apesar de não existir uma definição de TCP incompleto, a sua documentação identifica este protocolo como sendo TCP cujo processo de *Handshake* do TLS foi interrompido ou que o fluxo não tenha sido devidamente terminado com FIN/ACK. Daqui, surgem então duas aproximações possíveis de caracterização ao protocolo TCP, resultando em dois processos distintos de caracterização.

TCP

O processo responsável por analisar o ficheiro `log_tcp_complete.txt` começa pela leitura deste, colocando em memória o que foi lido.

Do ficheiro são retirados o endereço IP e porta do cliente e do servidor para que seja possível saber a quem o fluxo foi endereçado. Ao mesmo tempo é gerada uma *tag* de identificação (id) para facilitar a leitura posterior dos dados, que é incrementada segundo o número de linhas lidas. De seguida, são retirados os todos os *bytes* únicos transferidos entre cliente-servidor e servidor-cliente. Contabilizar os *bytes* únicos, ao contrário do total de *bytes* transferidos entre ambos, serve para dar uma aproximação ao tamanho real do conteúdo, tendo em consideração que as retransmissões efetuadas que não são somadas a estes valores. A acrescentar a estas métricas, *first* e *last* correspondem à data de entrega do primeiro e do último pacote do fluxo, e servem para identificar temporalmente quando ocorreu.

O Tstat associa o nome do servidor ao fluxo estabelecido com o cliente. Daqui resultam o `client_TLS_SNI`, ou seja, o *client* TLS *Server Name Identification* (SNI) obtido a partir das

mensagens de *Hello*, e ainda o `server_TLS_SCN`, *server* TLS *Subject Common Name* (SCN) que é o nome do sujeito indicado no certificado. É possível ainda acrescentar o FQDN, ou seja, *Fully Qualified Domain Name* que se trata do nome de domínio que específica a localização exata na árvore hierárquica do DNS. Este é obtido por DNSHunter na execução do Tstat, identificando assim a aplicação ou serviço a que o fluxo corresponde. Deste modo, um fluxo TCP é identificado tal como a expressão (1) indica.

$$f_{TCP} = (id, c_{ip}, c_{port}, c_{bytes\_uniq}, s_{ip}, s_{port}, s_{bytes\_uniq}, first, last, c_{tls\_SNI}, s_{tls\_SCN}, fqdn) \quad (1)$$

Tendo os fluxos em memória, resta agora a sua análise e caracterização. Para tal, é iniciado um processo que verifica o `client_TLS_SNI`, `server_TLS_SCN` e `FQDN` e tenta fazer corresponder a alguma expressão regular presente na lista que reúne todo o tipo de expressões indicadoras de tráfego publicitário. Este processo encontra-se descrito pelo Algoritmo 1.

Algoritmo 1: Heurística de Procura de Tráfego não Solicitado em TCP

```
for i ∈ Lista de Expressões Indicadoras de Anúncios do
    if i ⊂ (stream.c_tls_SNI ou stream.s_tls_SCN ou stream.fqdn) then
        True;
    else
        continua;
    end
end
```

Deste processo de análise surgirá um ficheiro `outputTCP.txt` que conterá todos os dados relativos ao tráfego envolvido com especial atenção ao tráfego não solicitado. Nele estão presentes a contagem de todos fluxos lidos, quantos destes são relativos a tráfego HTTP e HTTPS (verificação de presença das portas 80 e 443 no campo `s_port`), a quantidade de fluxos de anúncios dizendo quais são e qual a quantidade de *bytes* transferidos deste tipo de tráfego e ainda dois contadores de *bytes* trocados entre servidor-cliente e cliente-servidor.

Com o ficheiro *output* terminado e escrito em disco, a fase de análise a tráfego TCP completo dá-se por terminada, imprimindo em terminal que assim o fez.

TCP incompleto

O processo de análise de tráfego TCP incompleto começa da mesma forma que o de TCP completo, i.e., pela leitura do ficheiro `log_tcp_nocomplete.txt` e colocação em memória de tudo o que foi lido.

Um objeto em memória que contenha informação sobre tráfego TCP incompleto, conterá os seguintes campos: *tag* de identificação (id), endereço IP do cliente, porta do cliente e

quantidade de *bytes* únicos transferidos entre cliente-servidor. Quanto ao servidor, são retirados o seu endereço IP, porta e *bytes* únicos transmitidos entre servidor-cliente. A acrescentar a estas métricas e tal como no tráfego TCP, também são retiradas as datas do primeiro e último pacote do fluxo trocado entre ambos, sendo assim *first* e *last*. Assim, uma estrutura que contenha informação de tráfego TCP incompleto é representada como a expressão (2) demonstra.

$$f_{UnTCP} = (id, c_{ip}, c_{port}, c_{bytes\_uniq}, s_{ip}, s_{port}, s_{bytes\_uniq}, first, last) \tag{2}$$

Não tendo as métricas `client_TLS_SNI`, `server_TLS_SCN` e `FQDN`, a fase de comparação por expressões regulares não é tomada por não haver forma direta de classificar como tráfego não solicitado.

Contudo, a análise deste tipo de tráfego serve como forma de *debug*, na tentativa de classificar tráfego pelo endereço IP do servidor e tentar fazer corresponder que tipo de conteúdo foi entregue. Esta classificação baseia-se na aprendizagem realizada em testes a partir do tráfego TCP completo, onde se verificou a consistência de alguns endereços IP associados sempre ao mesmo FQDN. Desta análise, resulta a associação de endereços IP presentes em outras capturas, das quais é possível deduzir qual o servidor em atuação num determinado fluxo.

Ver-se-á mais à frente que, esta análise apenas tem peso para Facebook e Instagram. Esta particularidade foi encontrada em capturas de teste do Facebook, verificando-se que estas possuem bastante tráfego deste género. Considerando também o que foi capturado no `.har`, verifica-se uma relação entre os endereços IP e tamanhos de conteúdos, indicando que em *browser* os fluxos ficam em aberto. Isto revela que, em *web* as páginas estão sempre em constante atualização e requisição de conteúdos. Como estão sempre em aberto, estes fluxos não possuem previsão de final de vida e por isso acabam por ser interrompidos no momento de terminar a captura.

4.2.2 *Análise UDP*

Como todos os processos de análise anteriormente descritos, a análise a tráfego UDP começa pela leitura e colocação do ficheiro `log_udp_complete.txt` em memória.

De um fluxo UDP gerado pelo Tstat, são retirados os seguintes campos: endereço IP, porta, tempo absoluto do primeiro pacote, duração do fluxo e quantidade de *bytes* enviados. Estas métricas são retiradas e devidamente associadas tanto do lado do servidor como do cliente. O Tstat possui ainda um campo que identifica o tipo de conversa que servidor e cliente estão a ter, `c_type` e `s_type`. No caso de, por exemplo, estes contiverem valores iguais a 19 será uma conversa do tipo pedido DNS. Existe ainda o valor `0` que indica tratar-se de

tráfego desconhecido para o Tstat. Por fim é ainda acrescentado o FQDN obtido durante a captura. Desta forma um fluxo UDP é identificado como a expressão (3) demonstra.

$$f_{UDP} = (id, c_{ip}, c_{port}, c_{first\_abs}, c_{durat}, c_{bytes}, c_{type}, s_{ip}, s_{port}, s_{first\_abs}, s_{durat}, s_{bytes}, s_{type}, fqdn) \quad (3)$$

No processo de caracterização, é feita uma comparação do FQDN com as expressões presentes na lista que reúne todos os indícios de expressões que identifiquem como sendo tráfego não solicitado, como o Algoritmo 2, refere.

Algoritmo 2: Heurística de Procura de Tráfego não Solicitado em UDP

```
for i ∈ Lista de Expressões Indicadoras de Anúncios do
    if i ⊂ (stream.fqdn) then
        True;
    else
        continua;
    end
end
```

Sendo tráfego UDP, significa que existem diversos protocolos que fazem recurso deste protocolo como por exemplo o QUIC e DNS. Para tal, é necessário filtrar estes protocolos e realizar uma distinção entre todos os protocolos que utilizam UDP como camada de transporte. O Tstat ao classificar o tipo de conversa com a métrica `c_type` e `s_type` ajuda a facilitar este processo. Contudo, devido ao uso de cifra nem todo o tráfego é classificado devidamente resultando em ser classificado como tráfego desconhecido, ou seja, com o valor zero nesse campo. Em resposta, quando é apresentado valor zero, é dito tratar-se de conteúdo. A este conteúdo é feita uma verificação da porta que o servidor utiliza comparando com aquelas que se encontram registadas na IANA e verificar os serviços e protocolos a estas associadas. Os que são encontrados com maior frequência são TLS (porta 443), Network Basic Input/Output System (NetBIOS) (portas 137 e 138), Web Services Dynamic Discovery (WSDiscovery) (portas 139, 445, 1124, 3702), Multicast DNS (porta 5353) e ainda Simple Service Discovery Protocol (SSDP) que se baseia em Universal Plug and Play (UnPnP) (porta 1900). Para além destes protocolos mais comuns, existem ainda outros que devida à baixa utilização não chegam a ter uma classificação de protocolo com base em porta.

A acrescentar a este processo, é ainda feita uma análise em detalhe do tráfego face à aplicação ou serviço a caracterizar. Como visto anteriormente, a aplicação do YouTube utiliza gamas diversas de endereços IP que correspondem a determinados tipos de conteúdo. Para este caso em concreto, foi criada uma funcionalidade que verifica os endereços IP dos

servidores e separa os fluxos mediante a gama pertencente colocando-os numa estrutura que possui todos os fluxos relativos a esse propósito. A decisão de colocação em respetivas estruturas de memória é ilustrada pelo Algoritmo 3, onde se verifica que as gamas identificadas são: vídeo, imagens e eventos.

Algoritmo 3: Heurística de Distribuição em Gamas IP

```
1  switch IP do
2      case IP ∈ Gama de Vídeo do
3          Coloca na estrutura de Vídeo;
4      end
5      case IP ∈ Gama de Imagens do
6          Coloca na estrutura de Imagens;
7      end
8      case IP ∈ Gama de Eventos do
9          Coloca na estrutura de Eventos;
10     end
11 end
```

Daqui, é possível saber com maior detalhe se um fluxo retrata um vídeo, imagem ou evento provocado pela aplicação. Para os restantes serviços, este processo não é realizado por haver pouca informação relativamente ao protocolo de transporte UDP. Além disso, esses serviços possuem outro tipo de informação no `.har`, ficheiro que este não possui.

A distribuição em gamas de endereços IP não será a mais eficaz quanto a identificar os vídeos visualizados ao longo da sessões de captura. Surge então, a necessidade de haver uma filtração mais refinada destes fluxos anteriormente classificados como vídeo. O classificador retoma estes fluxos de vídeo e considera os campos duração e quantidade de *bytes* transferidos do servidor para o cliente e tenta encaixá-los dentro de um intervalo aceitável desta categoria. Este intervalo supõe que, qualquer vídeo tem de possuir tamanho e duração maior que zero e que necessitam de possuir tamanho maior que 15kB. Estes 15kB referem-se ao tamanho mínimo que um vídeo pode ter, ou seja, no caso de um vídeo muito curto, ou até mesmo um formato `.gif`, o tamanho deste será sempre maior que 15kB. A questão da duração mínima está dependente do débito de transmissão da rede pelo que a duração mínima de conteúdo em formato de vídeo será considerada igual a 10 segundos. Estes valores permitem excluir quaisquer comunicações servidor-cliente que apenas envolvam acordos de entrega de conteúdo e outros conteúdos que não vídeo. É de realçar que algumas conversas relativas a tráfego não solicitado serão excluídas nesta refinação mas encontram-se presentes na perspetiva geral de tráfego de rede. Todo este procedimento de apurar resultados obtidos encontra-se enunciada pelo Algoritmo 4.

Algoritmo 4: Heurística de Refinação de Procura de Vídeos Visualizados em UDP

```
1  for i ∈ Estrutura de Vídeo do
2      if i.bytes > 0 kB e i.duration > 0 s then
3          if i.bytes > 15 kB e i.duration > 10 s then
4              Coloca na Estrutura de Vídeo Refinados;
5          else
6              continua;
7          end
8      else
9          continua;
10     end
11 end
```

Por fim, do processo de análise UDP resulta o ficheiro `outputUDP.txt`. Nele estão contidas as seguintes informações: contagem total de fluxos lidos, contagem de pedidos DNS, contagem de fluxos que possuam conteúdo (tipo de conversa diferente de pedidos de DNS), contagem fluxos que façam recurso do protocolo QUIC e contagem de fluxos que foram classificados como tráfego desconhecido pelo Tstat distinguindo pelo protocolo em uso mediante a porta. Estão também presentes os fluxos que correspondem a eventuais conteúdos distinguidos pelas categorias de eventos, vídeo, imagens e desconhecidos que não possuam o seu devido filtro. No final deste ficheiro, são totalizados: o volume de tráfego em *kBytes* transferido entre servidor-cliente, cliente-servidor, quantos fluxos foram classificados como anúncios e o seu *payload* total. É ainda realizada uma contagem de todos os fluxos TLS sobre UDP e o seu *payload* total.

Para fluxos que foram classificados sendo tráfego não solicitado, são realçados detalhes como o tamanho e duração deste para que seja possível verificar se o *payload* é diferente de zero e duração maior que zero segundos.

4.2.3 *Análise HAR*

A análise HAR destaca-se por não ser um resultado do Tstat, mas um resultado das ferramentas de programador do Google Chrome. Um ficheiro `.har`, trata-se de um ficheiro em formato `.json` mas aplicado a *web browsers*, que possui toda a interação de uma página *web* na rede. Para a leitura do ficheiro har, o classificador conta com uma biblioteca externa, `harReader`, que efetua toda a leitura e coloca em memória. Depois, apenas é necessária a sua consulta e realizar a classificação.

Para cada entrada presente no `.har` são vistos os campos `url`, data, tamanho do pedido e endereço IP do servidor. Ao contrário dos outros processos, esta análise cria dois ficheiros, um denominado de `URLRequests.txt` e outro de `outputHAR.txt`.

No primeiro, estão impressos os campos retirados e respetivos resultados. Estes resultados são apenas de confirmação se o conteúdo presente foi transferido com sucesso. Para o caso de sucesso, classifica a ligação em uma de três categorias principais de conteúdo: gestão de página (`.js, .json, .zip`, ...), imagens e vídeo. Este primeiro ficheiro, surge como *debug* para verificar se houve problemas na obtenção de conteúdo. Constatando que a análise está mais orientada para as aplicações Facebook e Instagram, existe o problema de transferir conteúdos que não são relevantes para o estudo. Deste modo, foi criado um filtro que impede transferir fotografias de perfis de outras pessoas e páginas. Este filtro baseia-se em verificar a resolução da imagem a ser transferida e excluir qualquer imagem desta categoria. Esta exclusão apenas é possível tendo verificado que a resolução de imagens de perfil é fixa e que se possuem valores iguais a 100x100, 120x120 e 150x150. Este procedimento preserva a privacidade, sendo eficiente em questões de espaço e rapidez de obtenção dos conteúdos, visto que os pedidos podem expirar.

Durante esta primeira fase, existe um processo que caracteriza as imagens que são obtidas. Este processo trata de fazer *match* com as resoluções e *aspect ratio* de anúncios encontrados na documentação do Facebook e Instagram. Da documentação disponível e testes realizados, concluiu-se que as imagens possuem resoluções diferentes das publicações feitas por páginas às quais o utilizador possui relação (amizade e seguidor), o que significa que não se enquadram com resoluções de fotografias publicadas por outras pessoas. As resoluções mais vulgarmente encontradas foram 230x230, 370x370, 480x251, 800x926, 1024x1280, 1080x1350, 1137x640 e 1279x1919. Assim sendo, o classificador ao transferir um conteúdo compara a resolução desse com a lista de resoluções descobertas, tal como enuncia o Algoritmo 5.

Algoritmo 5: Heurística de Comparação de Resoluções

```
1 res = Imagem.width x Imagem.height;
2 if res ∈ [230x230, 370x370, 480x251, 800x926, 1024x1280, 1080x1350, 1137x640,
    1279x1919] then
3 |   True;
4 else
5 |   False;
6 end
```

É mediante estas imagens que vai ser possível identificar se um vídeo é um anúncio. Estes serviços utilizam uma *frame* do vídeo para que possa aparecer na eventualidade do utilizador restringir a reprodução deste. Esta *frame* é apresentada ao utilizador e no caso de

este clicar nesta imagem um vídeo será reproduzido. A classificação de vídeo é feita com recurso a uma *script* criada em Python que é invocada como co-rotina do classificador. Para cada vídeo encontrado e transferido, esta *script* tem como função dividir em várias *frames* e percorrer todas imagens classificadas como anúncio, verificando se existe alguma igual. No caso de serem iguais, será registado o vídeo que fez correspondência com uma imagem específica, sendo adicionado ao ficheiro `outputHAR.txt`. De notar que, este processo necessita de algum tempo para poder comparar imagem e *frame*. Ainda que seja acionado logo após a transferência de todos os conteúdos estar completa, demora sempre mais tempo do que a análise do restante tráfego e, por isso, é colocado a correr em plano de fundo até que complete.

Relativamente ao ficheiro `outputHAR.txt`, este conterá todos os resultados relevantes para o estudo. Aqui serão registados todos os endereços e nomes de servidores presentes no ficheiro `har`. Esta informação servirá futuramente para verificar quais os servidores mais participativos no tráfego de anúncios. De seguida, o ficheiro possui um apanhado de quantos pedidos foram lidos e destes quantos foram transferidos. Dos conteúdos transferidos, é ainda feita a contagem de quantos destes estão distribuídos pelas três categorias encontradas. A seguir, são registadas todas as ligações referentes a vídeo e feita a contagem de quantas vezes aparecem ao longo dos fluxos. Esta contagem é realizada porque ligações relativas a vídeo necessitam de ser repartidas por várias partes devido ao seu tamanho ser maior que outros conteúdos. Por fim, consideram-se todas as estatísticas relativas a tráfego não solicitado. Nestas, são apresentados a quantidade de anúncios apresentados e respetivo tamanho total em *kBytes*. São também indicadas percentagens de anúncios em relação a conteúdo total encontrado e conteúdo total transferido para futura apresentação de resultados. Os conteúdos publicitários identificados, são de seguida listados e é imprimido o respetivo *output* da *script* Python.

4.2.4 *Análise HTTP*

A análise de tráfego HTTP surge como complemento para o caso de algum dos serviços apresente tráfego deste tipo. Serve, sobretudo, para testar a funcionalidade de obtenção de conteúdo por HTTP e aprendizagem com outros serviços. O seu estudo é importante ainda que possa não ter impacto por não existir tráfego HTTP que não recorra a métodos criptográficos.

O ficheiro *output* `log_http_complete.txt` possui uma abordagem aos fluxos diferente dos outros protocolos. Tratando-se de tráfego HTTP o cliente faz pedidos (GET, POST) e o servidor responde a esses pedidos com uma *string* de resposta e respetivo código. Aqui o Tstat diferencia em fluxo cliente e servidor precisamente pela forma como o HTTP funciona. Um fluxo cliente-servidor possui campos diferentes a um fluxo servidor-cliente. Quer isto

dizer que é necessária uma distinção entre fluxos HTTP por cada uma das partes participantes. A forma encontrada foi criar duas estruturas distintas para suportar a leitura do ficheiro para memória.

A primeira foi denominada de **HTTPs**, HTTP por parte do servidor (diferente de HTTPS). Conta com um campo de identificação do fluxo, endereço IP e porta do cliente, tempo absoluto (*epoch*) do primeiro pacote, identificador de resposta (neste caso será sempre HTTP) e alguns campos presentes no cabeçalho de resposta do pedido HTTP tais como: código de resposta HTTP (2xx/3xx/4xx/5xx), comprimento e tipo de conteúdo, o nome do servidor, alcance de conteúdo (conteúdo parcial - código 206), localização (conteúdo encontrado - código 302) e o campo *set cookie* que mostra o *cookie* enviado. Um fluxo HTTPs fica caracterizado como a expressão (4) indica.

$$f_{HTTPs} = (id, c_{ip}, c_{port}, s_{ip}, s_{port}, time_{abs}, HTTP, response, \\ content_{len}, content_{type}, server, range, location, set\_cookie) \tag{4}$$

Quanto à segunda estrutura, foi denominada de **HTTPc**, HTTP por parte do cliente. Possui um campo de identificação do fluxo, endereço IP e porta do cliente, tempo absoluto (*epoch*) do primeiro pacote, método do pedido HTTP (`GET, POST, HEAD`), valor do campo `Host` de um pedido HTTP, FQDN retirado a partir de DNSHunter, *path* da ligação *url* e os valores presentes nos campos `Referer, User-Agent, Cookie` e `Do Not Track` no cabeçalho do pedido HTTP. Um fluxo HTTPc é caracterizado como a expressão (5) demonstra.

$$f_{HTTPc} = (id, c_{ip}, c_{port}, s_{ip}, s_{port}, time_{abs}, method, \\ hostname, fqdn, path, referer, user_{agent}, cookie, dnt) \tag{5}$$

Tendo ambas as estruturas já completas com todos os fluxos presentes no ficheiro, o próximo passo será a obtenção dos conteúdos transferidos. Por todos os fluxos cliente que contenham o método `GET` no campo *method*, é acionado um processo que tenta obter o conteúdo que foi transferido. Podendo ter vários tipos de conteúdos, verifica-se se o campo `path` contém indícios de conter tipos de ficheiro, i.e., no caso da expressão "`.jpg`"estar presente no `path`, o processo saberá que o conteúdo a transferir será uma imagem, podendo distinguir entre `.html`, `.json`, `.js`, `.css`, `.png` e `.db`. Todos os conteúdos possíveis são armazenados em disco para futura análise.

Numa fase final, é realizada uma comparação entre todos os elementos da lista que contém expressões acerca tráfego não solicitado, `adList.txt`, e os campos `content_type, server, set_cookie` de um fluxo HTTPs e os campos `hostname, fqdn, path, referer,`

`cookie` e `dnt` de um fluxo HTTPc. Esta procura de tráfego não solicitado encontra-se descrita pelo Algoritmo 6.

Algoritmo 6: Heurística de Procura de Tráfego não Solicitado em HTTP

```
1  switch stream do
2      case HTTPs do
3          for i ∈ Lista de Expressões Indicadoras de Anúncios do
4              if i ⊂ (stream.content_type ou stream.server ou stream.set_cookie) then
5                  True;
6              else
7                  continua;
8              end
9          end
10     end
11     case HTTPc do
12         for i ∈ Lista de Expressões Indicadoras de Anúncios do
13             if i ⊂ (stream.hostname ou stream.fqdn ou stream.path ou stream.referer ou
                stream.cookie ou stream.dnt) then
14                 True;
15             else
16                 continua;
17             end
18         end
19     end
20 end
```

Por fim, a análise a tráfego HTTP termina com a criação do ficheiro `outputHTTP.txt` que contém todos os fluxos que apresentam indícios de anúncios que resultam da comparação de expressões. O ficheiro contém também a contagem total de fluxos HTTP distinguida por HTTPs e HTTPc, a totalização de anúncios encontrados e o seu *payload* total.

4.3 RESULTADOS

Nesta secção são apresentados os resultados num estado mais elementar, ou seja, refletindo exatamente como os ficheiros *output* de cada protocolo são gerados. Os resultados aqui indicados, servem para provar o funcionamento do classificador nos bastidores e como os dados são gerados na sua forma mais crua. No capítulo 5, os resultados obtidos passam primeiro por esta fase e só depois será feito o seu devido tratamento.

4.3.1 *Análise TCP*

A análise ao protocolo TCP acontece de duas formas como enunciado anteriormente. Como tal, existem dois ficheiros relativos a ambos os paradigmas do protocolo TCP aqui mostrados.

TCP

Um ficheiro *output* que contenha estatísticas relativas ao protocolo TCP terá o formato abaixo representado.

```
    AD: StreamTCP(id=2, c_ip=192.168.137.239, c_port=60865, c_bytes_uniq=
1.1142578 KB, s_ip=216.58.201.162, s_port=443, s_bytes_uniq=3.9990234 KB,
first=09/08/2019 11:21:46.034, last=09/08/2019 11:25:46.583, c_tls_SNI=www.
googleadservices.com, s_tls_SCN=-, fqdn=www.googleadservices.com)
    AD: StreamTCP(id=4, c_ip=192.168.137.239, c_port=51175, c_bytes_uniq=
1.2441406 KB, s_ip=172.217.17.2, s_port=443, s_bytes_uniq=3.9111328 KB,
first=09/08/2019 11:23:26.514, last=09/08/2019 11:27:26.921, c_tls_SNI=
securepubads.g.doubleclick.net, s_tls_SCN=-, fqdn=securepubads.g.
doubleclick.net)

        Total Counted Streams: 33
        Total HTTP Streams: 7, On which 0 are ads
        Total HTTPS Streams: 20, On which 2 are ads
        Total Ads Found: 2
        Total Ads Payload: 7.9 kB
        Total KBytes downloaded C2S: 61.5 kB
        Total KBytes downloaded S2C: 234.8 kB
```

Inicialmente são apresentados os fluxos relativos a tráfego não solicitado e, neste caso, apresenta dois fluxos que foram classificados como sendo anúncios e, como tal, aparecem como prioridade no ficheiro. Existe uma marca que permite fazer a distinção de outros fluxos. Para tal, é colocada a *flag* "`AD:`"antes do fluxo.

De seguida, apresenta o total de fluxos lidos e quais destes correspondem a tráfego HTTP e HTTPS por, comparação de portas. Para o exemplo acima apontado, foram encontrados 7 fluxos do tipo HTTP, dos quais nenhum corresponde a tráfego não solicitado e 20 do tipo HTTPS em que foram identificados 2 como sendo anúncios. A contagem de anúncios é apresentada imediatamente a seguir. Para este caso, é possível visualizar que os 2 anúncios encontrados possuem uma carga total (*payload*) igual a 7.9 kB. Por fim, são apresentados o total de *bytes* trocados entre servidor-cliente (61.5 kB) e entre cliente-servidor (234.8 kB).

TCP incompleto

Passando a tráfego TCP incompleto, o ficheiro correspondente não terá mais detalhes que o ficheiro TCP completo. Uma vez que o processo responsável pela caracterização não compara expressões para deteção de tráfego não solicitado por não possuir campos para tal, serão imprimidos todos os fluxos em memória na sua devida estrutura.

```
    (Video)StreamUnTCP(id=0, c_ip=192.168.137.228, c_port=51090, c_bytes
_uniq=9.75293 KB, s_ip=195.8.13.210, s_port=443, s_bytes_uniq=4205.7734 KB,
first=25/06/2019 11:15:45.392, last=25/06/2019 11:16:18.508)
    StreamUnTCP(id=1, c_ip=192.168.137.228, c_port=33914, c_bytes_uniq=
2.7314453 KB, s_ip=172.217.168.163, s_port=443, s_bytes_uniq=5.8710938 KB,
first=25/06/2019 11:15:49.397, last=25/06/2019 11:15:49.631)
    (ImageScontent)StreamUnTCP(id=2, c_ip=192.168.137.228, c_port=45948,
c_bytes_uniq=1.6103516 KB, s_ip=195.8.13.209, s_port=443, s_bytes_uniq=
147.76855 KB, first=25/06/2019 11:16:10.404, last=25/06/2019 11:16:10.475)
```

Como foi dito na secção de implementação, o processo tenta fazer corresponder os endereços IP dos servidores a gamas de servidores conhecidas. No exemplo acima enunciado, é possível verificar que antes dos dados de alguns fluxos, estão presentes *strings* que indicam o tipo de fluxo identificado. Podem estar presentes os valores de `Video`, `ImageScontent`, `ImageZMScontent` e `External` que são nomes abreviados de servidores conhecidos durante captura de teste no caso do Facebook e Instagram.

```
Total Counted Streams: 5
Total Ads Found: 0
Total HTTP Streams: 0
Total HTTPS Streams: 5
Total Bytes downloaded C2S: 17.9 kB
Total Bytes downloaded S2C: 4595.8 kB
```

No final são disponibilizadas estatísticas generalizadas sobre quantos fluxos de TCP incompleto foram lidos, quantos anúncios foram encontrados e quantos dos fluxos lidos são HTTP e HTTPS. A quantidade de *bytes* trocados entre servidor-cliente e cliente-servidor são totalizados e mostrados como estatística. Os valores 5, 0, 0, 5, 17.9 kB e 4595.8 kB, enunciados no exemplo acima, correspondem às estatísticas descritas pela ordem respetiva.

4.3.2 *Análise UDP*

O ficheiro resultante da análise ao tráfego UDP varia consoante o serviço caracterizado. O exemplo abaixo enuncia o que foi registado do YouTube. Daqui observa-se a distinção

de fluxos por gamas de endereço IP discutida na secção de Implementação e como estão organizadas em formato de lista. É de notar que para os restantes serviços, este passo não é realizado, avançando para a fase de obtenção de estatísticas gerais.

```
    Streams to be considered as videos:
    StreamUDP(id=10, c_ip=192.168.137.239, c_port=57570, c_first_abs=09/08/
2019 11:21:48.805, c_durat=0.163378, c_bytes=3.4345703 KB, c_type=27, s_ip=
172.217.168.174, s_port=443, s_first_abs=09/08/2019 11:21:48.845, s_durat=
0.165940, s_bytes=6.461914 KB, s_type=27, fqdn=redirector.googlevideo.com)
(QUIC)
    ...
    ------------------------------------------------
    Streams to be considered as images(thumbnails, profile pics, ...):
    StreamUDP(id=6, c_ip=192.168.137.239, c_port=60757, c_first_abs=09/08/
2019 11:21:45.735, c_durat=5.496762, c_bytes=5.6845703 KB, c_type=0, s_ip=
172.217.16.246, s_port=443, s_first_abs=09/08/2019 11:21:45.776, s_durat=
5.474362, s_bytes=223.81836 KB, s_type=0, fqdn=i.ytimg.com) (Unknown)
    ...
    ------------------------------------------------
    Streams to be considered as event handling(json, js, ...):
    StreamUDP(id=60, c_ip=192.168.137.239, c_port=57818, c_first_abs=09/08/
2019 11:25:15.934, c_durat=14.780792, c_bytes=5.260742 KB, c_type=0, s_ip=
216.58.201.170,s_port=443, s_first_abs=09/08/2019 11:25:15.976, s_durat=
14.659973, s_bytes=19.766602 KB, s_type=0, fqdn=youtubei.googleapis.com)
(Unknown)
    ...
```

Para os fluxos anteriormente indicados, mais concretamente os de vídeo, a identificação obtida é apenas indicativa pois como explicado, mais à frente pode ser deduzido que não sejam desse tipo. Para tal, e apenas para o caso do YouTube, é feita uma seleção mais refinada dos fluxos do que a enunciada e descrita na fase de implementação. É também feito um cálculo da suposta duração do vídeo e ainda um cálculo de débito de transmissão em Mbps.

```
    Refined Video Selection
    StreamUDP(id=27, c_ip=192.168.137.239, c_port=42025, c_first_abs=09/08/
2019 11:21:49.848, c_durat=79.356144, c_bytes=49.166016 KB, c_type=27, s_ip
=195.8.11.206, s_port=443, s_first_abs=09/08/2019 11:21:49.855, s_durat=
79.344644, s_bytes=10186.776 KB, s_type=27, fqdn=r3---sn-2vgu0b5auxaxjvh-
v2ve.googlevideo.com) (QUIC)
```

```
Calculated Video Duration: 1.32 m
Calculated Download Rate: 1.05 Mbps
```

Para os restantes serviços são apenas indicados quais os fluxos que contenham anúncios, fazendo realçar aspetos como a duração de fluxo e o total de *bytes* transferidos.

```
Total Counted Streams: 481
Total DNS Requests: 110;
Total Content Streams: 371;
Total QUIC protocol Streams: 55;
Total Unknown Streams: 316;
(UnPnP: 229, NetBIOS: 4, TLS: 68, WSDiscovery: 14, Multicast DNS: 1)
Total TLS Streams: 123; -> On which, 4 are Ads
    Total Ads Found: 4 -> Total Ads Payload: 18.3 kB
Total C2S KBytes with QUIC protocol: 2488.7 kB
Total S2C KBytes with QUIC protocol: 993636.4 kB
Total C2S KBytes TLS Unknown: 875.2 kB
Total S2C KBytes TLS Unknown: 4749.7 kB
Total C2S Bytes Excluding DNS Requests: 3363.9 kB
Total S2C Bytes Excluding DNS Requests: 998386.0 kB
```

No final de cada ficheiro, e para todos os serviços em estudo, são mostrados todos os dados retirados no seu geral. Esta generalização permite a contagem de todos os fluxos lidos (481), pedidos DNS (110), pedidos considerados conteúdo (371), pedidos QUIC (55) e pedidos desconhecidos (316) identificados, logo a seguir, pelo seu tipo (Universal Plug and Play (UnPnP) - 229, Network Basic Input/Output System (NetBIOS) - 4, TLS - 68, Web Services Dynamic Discovery (WSDiscovery) - 14, MulticastDNS - 1). Depois são indicados quantos fluxos utilizam TLS (123) em UDP e quantos fluxos possuem indícios de serem anúncios (4). As estatísticas totais de anúncios encontrados são disponibilizadas logo de seguida (4 fluxos de tráfego não solicitado e 18.3 kB transferidos resultantes deste). Posteriormente, é realizada a totalização de *kbytes* transferidos de fluxos QUIC e fluxos desconhecidos entre cliente-servidor e servidor-cliente, tal como o exemplo acima demonstra.

4.3.3 *Análise HAR*

A análise a tráfego HTTP por meio do ficheiro `.har` resulta num ficheiro que contém dados relativos à captura, apenas totalizando em categorias a quantidade de pedidos HTTP encontrados. É inicializado por uma contagem de quantos pedidos HTTP foram lidos, quantos foram transferidos, quantos correspondem a conteúdo, quantos são conteúdos estáticos e quantos correspondem a imagens de perfil.

A seguir, são contabilizados quantos vídeos apareceram durante a captura. È feita uma contabilização de vídeos apresentados apresentado e quantos fluxos correspondem a vídeo. Devido ao tamanho que os vídeos apresentam, necessitam de ser fragmentados em múltiplas ligações e pedidos e daí surge uma contagem de fluxos de vídeo.

```
Total URL Requests: 252
Total URL Requests Payload: 43244 kB
Total Downloaded Requests: 66
Total Downloaded Requests Payload: 37781 kB
Total Content Requests: 79
Total Content Requests Payload: 37790 kB
Total Static Content Requests: 103
Total Static Content Payload: 5454 kB
Total Profile Pics Found: 25
Total Videos Found: 14
Total Video Streams Found: 70
```

Quanto ao tráfego não solicitado, as estatísticas respetivas são apresentadas no fim do ficheiro. Aqui, são totalizadas todas as imagens classificadas como anúncio e respetivo tamanho em *kbytes*. A seguir, são calculadas percentagens em relação ao tráfego obtido. Para o exemplo abaixo enunciado, os anúncios encontrados foram 7, correspondendo a 2.78% (7/252 = 0.0278) de pedidos encontrados no `.har` com um total de *kBytes* transferidos cerca de 0.27% (116/43244 = 0.00268). Posteriormente, é feita a relação entre conteúdo realmente transferido pelo classificador resultando em 10.61% (7/66 = 0.10606) de fluxos de anúncios face a fluxos de conteúdo transferido. De seguida, é calculada a porção de volume total gasto de fluxos de anúncios em volume total de todos os fluxos transferidos com sucesso pelo classificador em *kbytes*, correspondendo aos 0.31% (116/37781 = 0.00307) indicados abaixo. A justificação destas duas últimas percentagens deve-se ao facto do conteúdo ser tratado como tudo o que aparece no *feed* inicial da página Facebook ou Instagram. Sendo as imagens ou vídeos publicadas por outras pessoas consideradas conteúdo desejável, estes cálculos tornam-se relevantes para indicar que percentagem das publicações vistas é publicidade, mostrando o impacto dos anúncios.

```
Total Ads: 7 -> Total Ads Payload: 116 KB
Percentage of Ads found in General:
(quantity / size): 2.78% / 0.27%
Percentage of Ads found in Downloaded Content:
(quantity / size): 10.61% / 0.31%
```

Conforme a descrição da análise HAR, as imagens transferidas correspondentes a anúncios são registadas sob o formato de lista. Esta lista contém o nome de cada imagem que foi gerado aquando a sua transferência.

```
Images corresponding to adverts:
[image_19.png, image_24.jpg, image_45.png, image_55.png, image_56.png,
image_57.png, image_63.jpg]
```

Por fim, o processo para análise de vídeos publicitários em Python escreve para o ficheiro `outputHAR.txt` quais as imagens que estão presentes em *frames* de um dos vídeos considerados. No caso de haver imagens presentes, tal como o exemplo abaixo demonstra, o tamanho do vídeo terá de ser acrescentado manualmente às estatísticas relativas a anúncios.

```
Ad images corresponding to video adverts:
[I: image_63.jpg, V: video_66.mp4, S: 1808.3 kB]
```

4.3.4 *Análise HTTP*

O ficheiro resultante da análise HTTP começa por indicar quais os fluxos do lado do servidor que contêm tráfego não solicitado e, de seguida, os do lado do cliente. De seguida, exemplificam-se dois fluxos considerados anúncio, sendo um do tipo **HTTPs** e outro **HTTPc**.

```
StreamHTTPs(id=19, c_ip=192.168.137.56, c_port=37936, s_ip=216.58.201
.166, s_port=80, time_abs=17/05/2019 15:47:51.244, HTTP=HTTP, response=302,
content_len=0, content_type=text/html; charset=ISO-8859-1, server=cafe,
range=-, location=https://adservice.google.com/ddm/fls/z/src=4904904;cat=
932ddwl8; type=invmedia;dc_muid=*;ord=1716477479, set_cookie=-)
```

```
StreamHTTPc(id=51, c_ip=192.168.137.56, c_port=34917, s_ip=13.249.11.93,
s_port=80, time_abs=17/05/2019 15:48:50.017, method=GET, hostname=dl.cm
.ksmobile.com, fqdn=dl.cm.ksmobile.com, path=/static/res/e9/19/notification
cleaner_header_image_social.png, referer=-, user_agent=Dalvik/2.1.0
(Linux; U; Android 7.0; NEM-L51 Build/HONORNEM-L51), cookie=-, dnt=-)
```

No fim, são apresentadas as estatísticas gerais acerca da análise ao tráfego. São contabilizadas linhas lidas e linhas avaliadas. Depois, são contados pedidos do lado do cliente (27) e servidor (28). Por fim, são totalizados os anúncios encontrados (3),indicando se se referem a cliente ou a servidor e mostrando a sua carga total.

```
Total Lines Counted: 55
Total Lines Evaluated: 55
Total Client Side Requests: 27
Total Server Side Requests: 28
Total Ads Found: 3
Total Ads Found in C2S(quantity): 2
Total Ads Found in C2S(size): 8187 Bytes
Total Ads Found in S2C(quantity): 1
Total Ads Found in S2C(size): 0.0 Bytes
```

4.4 SUMÁRIO

Ao longo deste capítulo, foi efetuada uma reflexão sobre as decisões necessárias para a fase de implementação. Desta fase de implementação, foi possível visualizar como os tipos de tecnologias utilizadas se complementam entre si no projeto, tecnologias essas que incluem o Wireshark, Tstat, Kotlin, Python e Chrome Dev Tools. Foram discutidos todos os processos envolvidos na análise de todos os tipos de tráfego presentes nas capturas para posteriormente serem gerados resultados relevantes ao estudo.

Este capítulo dá por terminada a parte mais prática do trabalho. Inicia-se uma fase de apreciação e reflexão detalhada dos resultados obtidos.

5

RESULTADOS

Será ao longo deste capítulo que serão apresentados e consolidados todos os resultados obtidos ao longo do projeto, utilizando o método de classificação de tráfego desenvolvido no Capítulo 4. A apresentação será tomada por serviço a caracterizar e para cada serviço será feita uma generalização do tráfego capturado, dando depois ênfase a tráfego não solicitado.

5.1 CONFIGURAÇÃO EXPERIMENTAL

A caracterização de tráfego tem como foco os dispositivos móveis e, como tal, foi necessário recorrer a um *smartphone* Huawei Honor 5C com sistema operativo Android 7.0 (Nougat). De seguida, foi necessário criar a rede que sustenta tanto o *smartphone* como um PC equipado com um analisador de tráfego (*sniffer*). Para tal, foi utilizada uma funcionalidade do sistema operativo Windows 10, denominada Windows Mobile Hotspot, que permite que um PC se torne num *Access Point*, desde que possua uma antena WiFi.

Para cada serviço foram realizadas múltiplas capturas de teste pelo que, não podendo ser todas apresentadas, serão apresentadas apenas as últimas três de cada tipo, resultado do conhecimento adquirido ao longo da fase de testes. Estas três sessões possuem variedades distintas de publicidade presente nas plataformas em estudo. Refere-se ainda que, os parâmetros de caracterização resultam da média combinada destas capturas.

Mediante o serviço, são especificados um conjunto de características que vão desde a sua duração à quantidade de anúncios visualizados. Para o YouTube, são descritas algumas circunstâncias das sessões explicando o ambiente simulado. Por sua vez, no Facebook e no Instagram as capturas pretendem demonstrar a diversidade de formatos de publicidade existentes apenas na página inicial respetiva (sem contar com outras páginas de publicações que estas plataformas possuam). Em anexo, são disponibilizados detalhes mais específicos como total de *bytes* transferidos e pacotes, horário de captura e vídeos visualizados, se for o caso.

De seguida, descrevem-se resumidamente os vários *datasets* capturados para os serviços YouTube, Facebook e Instagram considerados no estudo.

- **YouTube 1** - Esta captura representa a situação de um vídeo relativamente curto em relação ao tempo normal despendido nesta plataforma, que normalmente se situa nos 40 minutos [16]. Desta sessão resultou 1 vídeo e 2 anúncios, sendo estes também formato de vídeo. O vídeo em questão possuía uma duração de 7min e 20s e foi visto na totalidade.
- **YouTube 2** - Esta captura corresponde a uma situação onde são visualizados múltiplos vídeos. Sendo a mais longa de todas, foram visualizados 3 vídeos e ainda uma pré-visualização de um vídeo nas "Sugestões"iniciado pela própria aplicação YouTube. O primeiro vídeo possuía uma duração de 21min e 48s, tendo sido avançado com o intuito de forçar publicidade. O segundo vídeo contava com uma duração de 11min e 12s e foi visto na totalidade. Por fim, o último vídeo possuía uma duração de 12min e 50s e também foi visto na integra. Desta captura, resultaram 2 anúncios em formato de vídeo.
- **YouTube 3** - Esta última captura contou apenas com um vídeo visualizado, descrevendo uma situação de visualização de um vídeo mais longo. Este vídeo tinha uma duração de 27min e 11s e foi visto para mostrar os espaços que o YouTube cria de publicidade durante a visualização. Daqui, resultaram 4 conteúdos publicitários todos eles em formato de vídeo.

- **Facebook 1** - A primeira captura referente ao Facebook resultou da visualização de 40 publicações das quais 7 correspondiam a publicações do tipo "Patrocinado". Dos 7 anúncios, um era em formato de vídeo e os restantes eram imagens singulares.
- **Facebook 2** - Quanto à segunda captura, foram visualizadas 45 publicações no total. Destas 8 eram publicidade. Estes 8 anúncios caracterizam-se por um ter sido em formato de vídeo e os restantes 7 imagens singulares.
- **Facebook 3** - Esta última captura possui uma duração maior que as anteriores, na qual, foram visualizadas 50 publicações. Daqui, 13 eram relativas a anúncios. Ao longo desta sessão não ocorreu publicidade em formato de vídeo, contudo 3 das publicações de anúncio vistas eram imagens agregadas em formato Coleção, aumentando o volume de conteúdo transferido em tráfego não solicitado.

- **Instagram 1** - Para o Instagram, a primeira captura é diferente das seguintes por ser dedicada ao serviço *web* correspondente, com o intuito de analisar as particularidades e diferenças relativamente à aplicação. Contou com 36 publicações visualizadas e nenhumas delas correspondentes a anúncios.
- **Instagram 2** - Para esta captura foi utilizada a aplicação para testar as funcionalidades de descoberta de conteúdo através do endereço IP e assinatura. Foram visualizadas 38 publicações das quais 6 eram anúncios e 6 "*Instastories*"em que 3 eram publicidade em formato de vídeo. As 6 publicações publicitárias possuíam a indicação de "Patrocinado"sendo que 2 eram vídeo e as restantes imagem.

- **Instagram 3** - Tal como a anterior, esta captura também se baseia na aplicação. É mais extensa que as anteriores e conta com 47 publicações vistas e 54 "*Instastories*". Das publicações, identificou-se que 10 eram anúncios, sendo que 3 eram em formato de vídeo e os restantes em imagem. Por fim, das "*Instastories*"retiraram-se 3 anúncios visualizados também representados em vídeo.

Por fim, a Tabela 5.1 faz uma síntese dos *datasets* em estudo. De notar que uma duração inferior a 10 minutos será considerada como curta, entre 10 e 30 minutos média e acima de 30 minutos longa.

Sessão	Duração	Tipo de Conteúdo	Quantidade de Conteúdo	Quantidade de Anúncios	Formatos de Anúncio
YouTube 1	Curta	Vídeo	1	2	Vídeo
YouTube 2	Longa	Vídeo	4	2	Vídeo
YouTube 3	Média	Vídeo	1	4	Vídeo
Facebook 1	Curta	Imagem e Vídeo	40	8	Imagem e Vídeo
Facebook 2	Curta	Imagem e Vídeo	45	8	Imagem e Vídeo
Facebook 3	Curta	Imagem e Vídeo	50	13	Imagem e Vídeo
Instagram 1	Curta	Imagem e Vídeo	36	0	-
Instagram 2	Curta	Imagem e Vídeo	44	9	Imagem e Vídeo
Instagram 3	Curta	Imagem e Vídeo	101	13	Imagem e Vídeo

Tabela 5.1.: Resumo de Capturas.

5.2 RESULTADOS

Ao longo desta secção serão apresentados e explicados todos os resultados obtidos durante a fase de captura de tráfego. Desta discussão resulta, toda a informação obtida e consolidada para a secção de reflexão de resultados.

5.2.1 *YouTube*

TCP

Começando pela análise ao protocolo TCP, o YouTube, numa visão geral, não possui fluxos fora do normal funcionamento, ou seja, a constante utilização de cifra demonstra a boa implementação para o propósito da cifra. Claro está que, destes fluxos HTTPS é possível

observar a intervenção de servidores de anúncios. Para cada uma das três capturas realizadas, foram determinadas estatísticas gerais acerca do tráfego no seu geral, realçando as percentagens obtidas de tráfego não solicitado tanto em frequência como em quantidade de *bytes* transferidos.

A Tabela 5.2, abaixo ilustrada, demonstra as percentagens de fluxos obtidos relativos a tráfego não solicitado face ao tráfego total. De relembrar que, a definição de tráfego não solicitado decorre da comparação dos FQDN dos servidores com a lista de expressões indicadoras de anúncio. Observando a tabela, verifica-se que as percentagens baixam de sessão para sessão, demonstrando que a duração da captura desempenha aqui um papel importante. Tal facto, deve-se ao aumento da quantidade de fluxos TCP no seu total e a quantidade de fluxos não solicitados não aumentar significativamente em relação a este valor. Outro pormenor retirado da tabela terá sido, na segunda sessão, a inexistência de relação direta entre duração de visualização e quantidade de vídeos visualizados pois, esta é a que possui maior diversidade de vídeos e maior duração. Resultando daqui a existência de outras variáveis que influenciam o comportamento do surgimento de publicidade.

Por último, na última sessão, apenas um vídeo foi visto no qual o YouTube, nestas situações de vídeos mais longos, tende a criar espaços de publicidade durante o vídeo, interrompendo a sua visualização. Apesar de terem sido observados anúncios no decorrer da transmissão, os fluxos correspondentes não foram observados em protocolo TCP, tal como a Tabela 5.2 reforça.

Sessão	Total de Fluxos	Total de Fluxos Não Solicitados	Tráfego Não Solicitado (%)
YouTube 1	10	2	20%
YouTube 2	33	2	6.1%
YouTube 3	32	2	6.3%

Tabela 5.2.: Fluxos TCP Obtidos nas Sessões YouTube.

Por sua vez, a Tabela 5.3, ilustra o volume total transferido em tráfego não solicitado face ao volume total transferido. Aqui, destaca-se o facto do volume transferido em tráfego não solicitado não variar mais do que 1kB entre sessões. Isto significa que os fluxos de tráfego não solicitado possuem o mesmo conteúdo no corpo da mensagem, indo ao encontro de se tratarem de acordos acerca de entrega de anúncios.

Sessão	Total Transferido (kB)	Total Transferido em Tráfego Não Solicitado (kB)	Tráfego Não Solicitado (%)
YouTube 1	40.6kB	7.8kB	19.2%
YouTube 2	234.8kB	7.9kB	3.4%
YouTube 3	187kB	6.9kB	3.7%

Tabela 5.3.: Volume em kBytes transferidos nas Sessões YouTube.

Não sendo possível decifrar os fluxos TCP, pela extração das assinaturas do FQDN dos servidores, é possível demonstrar de onde provieram os fluxos e, consequentemente, obter características do distribuidor. Essa representação está assente nos grafos nas Figuras 5.1, 5.2 e 5.3, onde a quantidade de pedidos é denotada pelo tamanho do nodo e respetivos nomes dos servidores (em legenda).

Do grafo representado na Fig. 5.1, retira-se que o servidor com maior afluência é `s08`. O seu nome `redirector.googlevideo.com` é referente ao redirecionamento de localizações de conteúdo, quer seja ele de vídeo ou de imagem. Os restantes servidores são relativos a entrega de conteúdo ou de gestão do serviço, à exceção de `s02` e `s03`. Estes dois, `s02` e `s03`, são respetivos à entrega de anúncios. Ambos mantiveram apenas um fluxo, cada um com 3.9 kB de tamanho. É possível observar que, pelo nome, estes entregam conteúdo não solicitado por possuírem a expressão "`ad`", algo que é consistente com as restantes capturas.

Uma característica verificada foi o sufixo `1e100.net` estar presente no nome do servidor em `s04`. Todos os servidores envolvidos em tráfego YouTube pertencem a este domínio. Mesmo com assinaturas de fluxo diferentes, o seu nome de domínio na árvore hierárquica de DNS corresponde a uma iteração deste sufixo. Este corresponde ao nome do domínio da Google referente à notação científica de um *googol*, que se trata de uma unidade sem valor matemático mais próximo do infinito, dando a ideia de quão grande é algo (10^{100}).

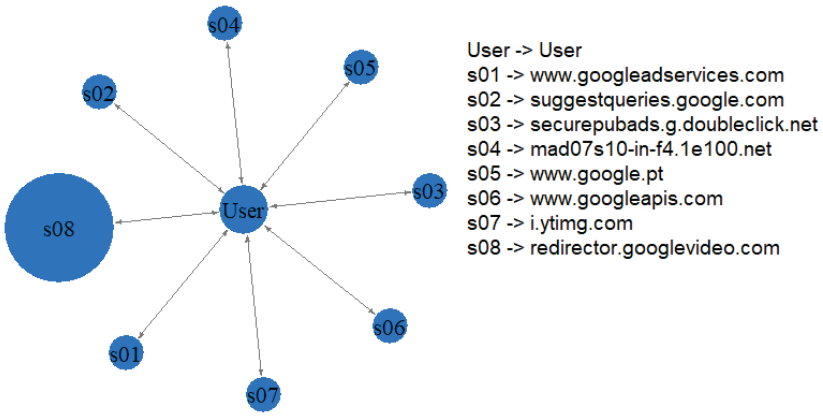


Figura 5.1.: Servidores TCP envolvidos durante a sessão YouTube 1.

Passando à segunda captura, o servidor com mais realce da Figura 5.2, é `s21` que obteve a maioria de pedidos realizados. Estes pedidos referem-se ao tráfego HTTP encontrado no ficheiro *output* de TCP. A sua intervenção de 21% no tráfego TCP encontrado, deve-se na totalidade à aplicação de meteorologia a correr de fundo que, não sendo possível interrompê-la, apareceu ao longo da sessão.

Seguidamente, os servidores com maior afluência são: `s11`, visto também na primeira captura, `s02`, responsável por criar e enviar dados analíticos relativos aos vídeos em visualização e, finalmente `s04`, responsável por criar sugestões de visualização de outros vídeos. Os res-

tantes servidores intervenientes apesar do seu papel, não possuem relevância ao estudo de tráfego não solicitado. Por fim, os dois fluxos de tráfego não solicitado tiveram origem nos servidores `s03` e `s05`.

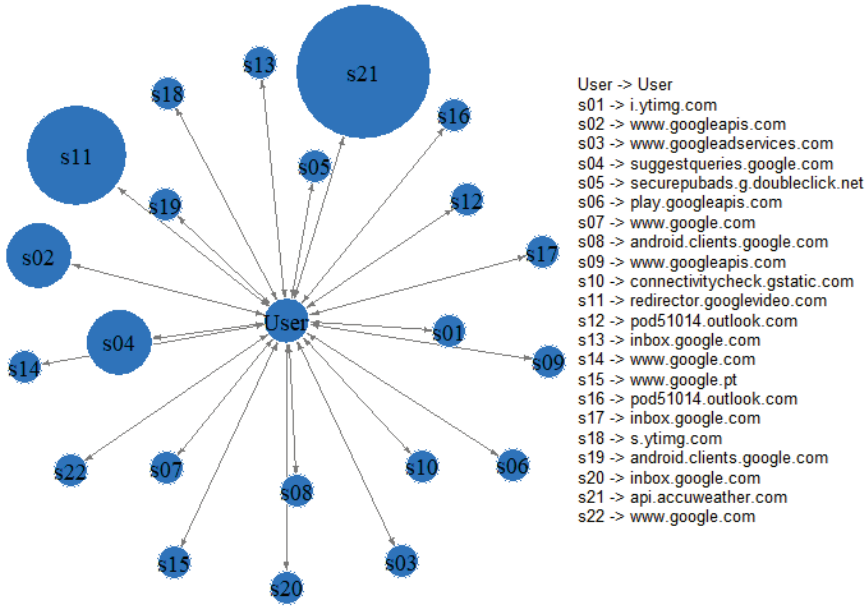


Figura 5.2.: Servidores TCP envolvidos durante a sessão YouTube 2.

Por fim, na terceira sessão os servidores intervenientes acabaram por ser semelhantes aos da segunda sessão, tal como a Figura 5.3 indica. Em acréscimo a tais servidores, foram identificados três cujos nomes estão ligados a entrega de conteúdo relativo a vídeo. Esses servidores são `s02`, `s03` e `s04` cujo sufixo `googlevideo.com` é indicativo à entrega de vídeo. No prefixo, está presente a localização exata do conteúdo em questão. Com a ajuda do *Wireshark* foi possível observar que estes possuem mais relevância para o protocolo UDP do que com TCP, ou seja, observando o campo do tamanho dos fluxos verifica-se que estes encontram-se no intervalo [3,5] kB. Estes valores são baixos mediante os conteúdos que foram visualizados, no entanto, em TCP não existem certezas acerca do tipo de conteúdo transmitido por este tipo de servidores [22]. Mais à frente, na análise UDP, vai-se poder observar estes mesmos servidores a responder a pedidos relacionados com entrega de vídeo mas em portas diferentes.

Recorrendo ao grafo representado na Fig. 5.3, os servidores `s14` e `s16` foram os mais intervenientes durante a captura. Tendo visto na primeira captura a função de `s16`, o servidor `s14` trata-se de um servidor da Google responsável por albergar conteúdos no Google quer sejam imagens ou vídeos e os entregar ao utilizados final. De notar que, `s19` é o nome do servidor de *mail* Outlook que atualiza o correio eletrónico. Este irá aparecer em múltiplas sessões tanto YouTube como nos restantes serviços.

Para esta última sessão, todos os fluxos não solicitados provieram dos servidores `s10` e `s12` e possuíam um tamanho igual a 3.3kB e 3.6kB, respetivamente.

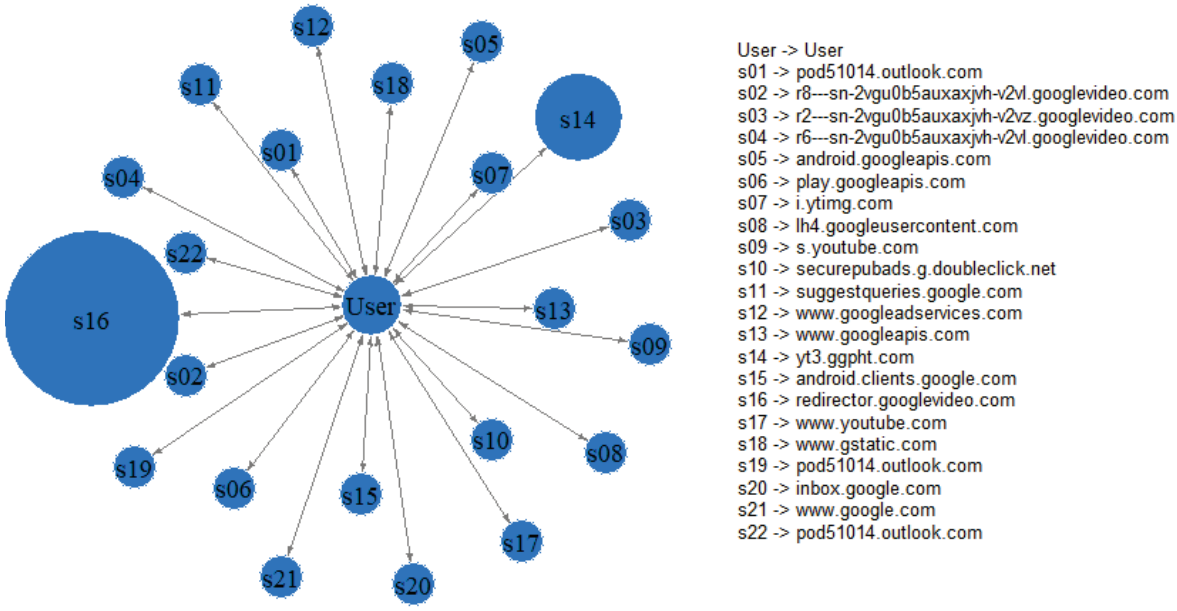


Figura 5.3.: Servidores TCP envolvidos durante a sessão YouTube 3.

Na questão de caracterizar tráfego não solicitado, o YouTube mantém fluxos TCP com servidores de entrega de anúncios cujo intuito será o acordo entre ambos para entregar publicidade via UDP. Constatando que não apareceram imagens de caráter publicitário, não foi possível indicar se ocorre transferência de tais conteúdos via TCP.

UDP

Passando agora ao protocolo UDP, sendo o YouTube uma plataforma de entrega de vídeo que faz recurso ao protocolo QUIC, será normal observar-se uma maior quantidade de informação retirada durante as capturas. Primeiro será visto o tráfego no seu geral, mostrando características não muito detalhadas, para que depois seja feita uma análise mais específica ao tráfego não solicitado.

Começando por uma vista geral do tráfego UDP, na Tabela 5.4 observa-se a quantidade de fluxos classificados como conteúdo face à quantidade de fluxos total. De notar que, nesta primeira tabela todos os fluxos não DNS são considerados como transporte de conteúdo.

Sessão	Total de Fluxos	Total de Fluxos DNS	Total de Fluxos Considerados Conteúdo
YouTube 1	164	61	103
YouTube 2	481	110	371
YouTube 3	439	102	337

Tabela 5.4.: Fluxos UDP Obtidos nas Sessões YouTube.

Conforme dito na subsecção 4.2.2, os fluxos de tráfego UDP passam por uma análise de tipo, de portas e, por fim, uma análise de gama de endereços IP. Começando pela análise de tipo, a Tabela 5.5 reflete quanto do tráfego anteriormente classificado como conteúdo, é classificado como desconhecido ou como tráfego QUIC pelo Tstat.

Sessão	Fluxos do tipo QUIC	Fluxos do tipo QUIC (%)	Fluxos do tipo Desconhecido	Fluxos do tipo Desconhecido (%)	Outros
YouTube 1	14	13.6%	89	86.4%	0
YouTube 2	55	14.8%	316	85.2%	0
YouTube 3	43	12.8%	294	87.2%	0

Tabela 5.5.: Tipos de Fluxos UDP Obtidos nas Sessões YouTube.

Na primeira captura, do tráfego classificado como desconhecido, o gráfico na Fig. 5.4 indica que, da análise às portas, 66% dos fluxos foram relativos a Universal Plug and Play, 1% a Web Services Dynamic Discovery, 1% a MulticastDNS, 1% de NetBIOS, 1% a outros protocolos sem filtro adequado e finalmente 26% de fluxos TLS. São estes fluxos TLS que despertam maior interesse pois nem todos os vídeos são demarcados pelo Tstat como usando QUIC e que resultam em serem classificados como desconhecido.

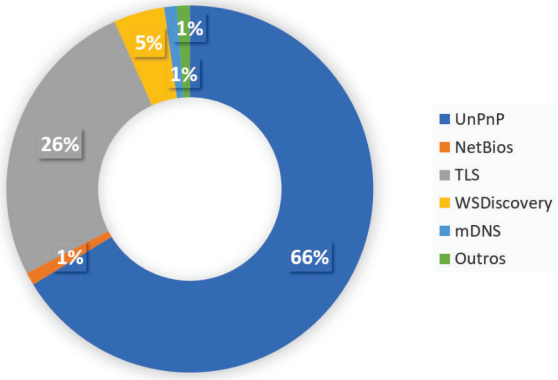


Figura 5.4.: Fluxos UDP Obtidos na sessão YouTube1.

Assimilando o tráfego TLS e QUIC encontrado no *output* UDP, parte-se à análise das gamas de endereços IP. Mediante o que foi dito na Secção 4.1.1, o tráfego é distinguido em três gamas principais: Gestão de Eventos, Imagens e Vídeo. Para esta primeira sessão,

resultaram 13 fluxos classificados como vídeo, 12 como imagem, 8 como eventos e 5 sem filtro. Resultando assim, em 37 fluxos efetivos de conteúdo transferido. O gráfico na Figura 5.5, identifica a porção de fluxos de conteúdo encontrados correspondentes à sua gama.

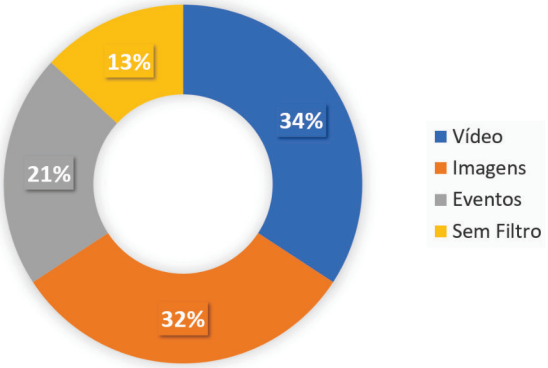


Figura 5.5.: Outros Protocolos Encontrados na sessão YouTube 1.

Os 13 fluxos de vídeos passaram de seguida pela fase de refinação e daí resultaram 4 vídeos. A partir da relação vídeo-fluxo realizada ao longo da captura em *Wireshark*, verifica-se que 3 destes 4 vídeos são de facto os vídeos visualizados durante a sessão, revelando uma precisão de 75% na procura de vídeos no tráfego UDP resultante. Resta dizer que, o fluxo sobrante corresponde a sugestão de outros vídeos iniciado pela aplicação, ou seja, este fluxo manteve-se vivo ao longo da transmissão e visualização do vídeo com a diferença de que a quantidade de *bytes* transferidos era muito menor quando comparado com um vídeo.

Sabendo-se que foram encontrados dois vídeos publicitários nesta primeira sessão, a estes correspondem dois fluxos relativos a tráfego não solicitado identificado a partir da verificação de assinaturas. Estes contribuem com uma carga total igual a 9.4kB e correspondem a conversas entre os servidores cuja assinatura é igual a `googleads.g.doubleclick.net`. Somando tudo, resultam 2 MB gastos em tráfego não solicitado o que corresponde a uma percentagem de 0.7% em relação a todo o tráfego considerado conteúdo. Significando que, o seu peso face a tráfego solicitado é reduzido.

Passando às conversas observadas durante a sessão, enunciado na Figura 5.6, `s01` corresponde ao servidor onde estão alojados parte das funcionalidades do YouTube e é o que possuiu maior afluência. Os servidores correspondentes a entrega de vídeo são `s10`, `s11`, `s12`, `s17` e `s19` por causa do sufixo `googlevideo.com`. Quando é dito que são de entrega de vídeo, não está 100% correto pois durante esta sessão foi possível observar que `s10`, `s11` e `s19` correspondem às listas de vídeos a sugerir ou a vídeos por visualizar. Os restantes (`s12` e `s17`) são os que de facto participaram ativamente na entrega de vídeo propriamente dita. Destaca-se que, `s12` foi responsável pela entrega de vídeo e publicidade em fluxos distintos para portas distintas. Do grafo retira-se ainda que, os servidores de vídeo acabam

por ser chamados apenas uma vez por prestarem apenas um serviço como, por exemplo, o de distribuir vídeo, onde o fluxo acaba por ser único se não houver retransmissões ou interrupções que o façam retomar de seguida. Por este motivo, verifica-se que outros servidores possuem maior afluência em pedidos por estarem em constante atualização.

Os restantes servidores representados possuem ligações para assegurar o serviço e retirar estatísticas acerca da sessão. Concluiu-se ainda que, a partir do endereço IP, `s08` e `s16` foram classificados como sendo relacionados a eventos e que, ao possuírem assinaturas semelhantes, estão ligados a serviços de mediação que retiram informações acerca de níveis de QoS e QoE.

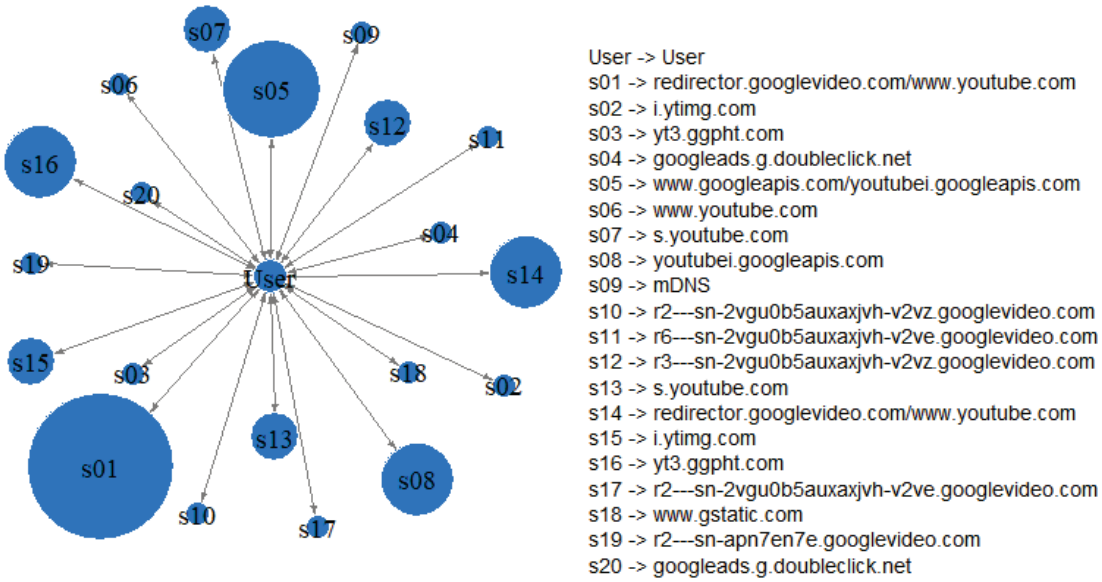


Figura 5.6.: Servidores UDP envolvidos durante YouTube 1.

Para a segunda sessão, a análise às portas revelou que UnPnP foi o protocolo dominante ao longo da sessão, tendo registado 22% de fluxos TLS visto serem os de maior importância nesta situação. A Fig. 5.7 indica a distribuição dos fluxos UDP classificados como tráfego desconhecido pelo Tstat por protocolo encontrado.

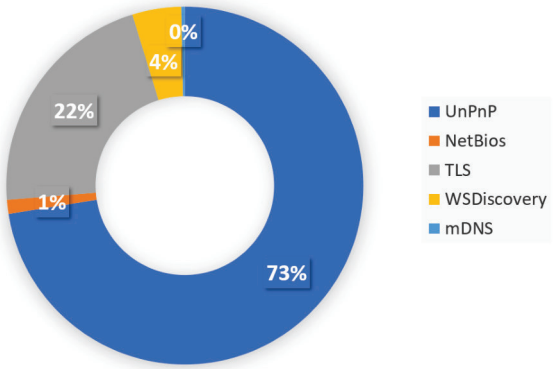


Figura 5.7.: Outros Protocolos Encontrados durante YouTube 2.

Mais uma vez assimilando todo o tráfego QUIC e TLS, a distribuição por gamas de endereços IP demonstra que foram encontrados 47 fluxos relacionados a vídeo, 37 de imagens, 26 de eventos e 13 sem filtro adequado. O gráfico na Figura 5.8 indica qual a sua proporção relativamente à junção de fluxos QUIC e TLS. Uma particularidade observada foi terem sido visualizados 6 vídeos no total, contando com a pré-visualização e anúncios, e o classificador ter apresentado um número excessivo igual a 47. Este valor justifica-se pelo facto do filtro ter em conta apenas o protocolo identificado ainda que estes possam não ser de vídeo. É aqui que se começa a observar algumas falhas do processo de filtragem e daí se justificar haver uma análise mais refinada destes fluxos. Dessa análise, resultaram apenas 5 fluxos, valor bastante mais baixo que o anterior. Destes 5 fluxos resultantes, apenas 3 possuem a relação direta com vídeo visualizado e os outros 2 serem de controlo de sugestão de vídeos.

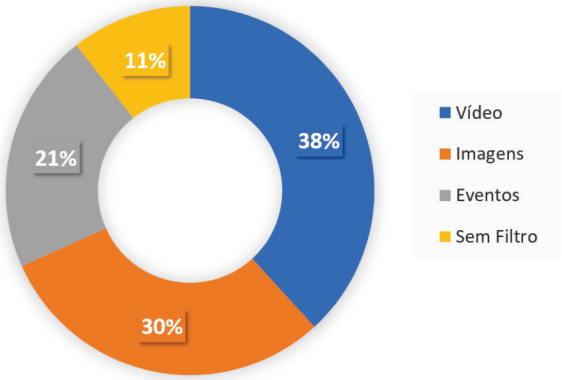


Figura 5.8.: Fluxos Distinguidos pela Gama IP na sessão YouTube 2.

Esta sessão foi escolhida por possuir algumas particularidades que não foram observadas anteriormente. A primeira, foi a observação de servidores de entrega de vídeo fora das gamas identificadas anteriormente. O primeiro vídeo foi entregue por um servidor cujo endereço IP era igual a `173.194.187.71` e tinha poucas visualizações na data em que

foi feita a captura. Embora pertencente à Google, o classificador não possuía forma de identificar este pormenor pelo que, não foi devidamente caracterizado. Isto comprova que, o YouTube para vídeos que não tenham muita popularidade, não transfere os conteúdos para as CDNs da localização do utilizador, demonstrando que a popularidade do vídeo é uma variável que faz depender não só a publicidade como a caracterização desta plataforma. Esta questão levanta também a variável da localização do utilizador, que influencia na publicidade que vai aparecer ao longo da sessão do vídeo.

A segunda particularidade observada foi um dos anúncios não ter sido encontrado, ou seja, não foi possível concretizar a associação fluxo-vídeo para o segundo vídeo publicitário. Aqui concluiu-se que a classificação de tráfego não solicitado não foi tomada na sua totalidade visto não haver padrões definidos para entrega de tais conteúdos.

Assim, em todo o tráfego UDP, foram encontrados 4 fluxos associados a entrega de anúncios que juntando ao único fluxo de vídeo encontrado e correspondente a publicidade, perfaz 0.002% da quantidade de tráfego total tranferida. Esta sessão serve para comprovar que a classificação de tráfego não é exata e que está sujeita a falhas.

Olhando para as conversas servidor cliente na segunda sessão, enunciado pela Figura 5.9, é possível verificar que os servidores com maior afluência são `s03`, `s04`, `s22` e `s23` realçando que, pelo seu FQDN são relativos a outras funcionalidades que não entrega de vídeo. Mais especificamente `s04` foi classificado por pertencer à gama de entrega de imagens, `s22` na gama de eventos e por fim `s03` e `s23` pertencentes à gama de suposta entrega de vídeo por possuírem registo de protocolo QUIC, indo ao encontro das falhas anteriormente enunciadas.

Efetuando uma análise mais profunda e verificando os resultados da refinação, os responsáveis pela entrega de vídeo foram `s07`, `s09`, `s25`, `s32` e `s47` respetivamente, de onde se retira que `s25` é o servidor cujo IP é externo a Portugal e os restantes pertencerem à CDN do provedor de serviços de Internet MEO.

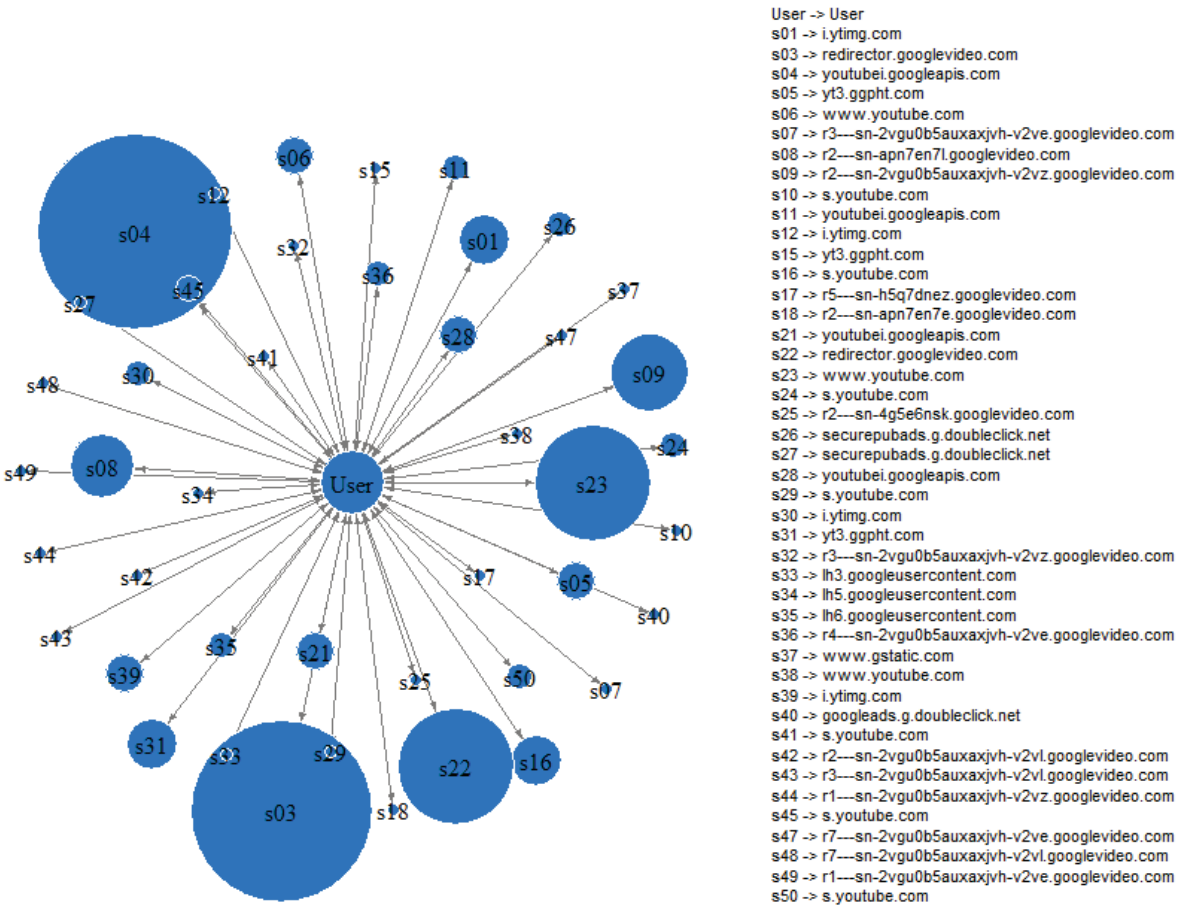


Figura 5.9.: Servidores UDP envolvidos durante YouTube 2.

Passando à última sessão, quanto ao tráfego desconhecido a maior parte corresponde, mais uma vez, a tráfego UnPnP, com 71%, e 21% a tráfego TLS, como refere o gráfico da Fig. 5.10. Relativamente ao protocolo UnPnP, é natural observar-se bastante presença (visto também em outras capturas) deste em tráfego desconhecido, porque trata-se do protocolo que múltiplos dispositivos fazem uso para descoberta entre os dispositivos na rede.

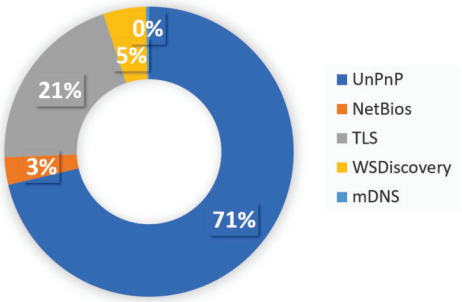


Figura 5.10.: Outros Protocolos Encontrados na sessão YouTube 3.

Analisando o tráfego TLS e QUIC, através da distribuição de gamas de endereços IP, foi possível verificar que ocorreram 23 fluxos relativos à gama de vídeo, 52 da gama de imagem, 23 de gestão de eventos e por fim 6 fluxos sem filtro adequado. Estes fluxos representam a proporção enunciada no gráfico da Figura 5.11. Tal como levantado anteriormente, aqui não foi exceção, o classificador observa apenas o protocolo em uso e, deste modo, faz a correspondência direta para vídeo, mesmo que não o seja.

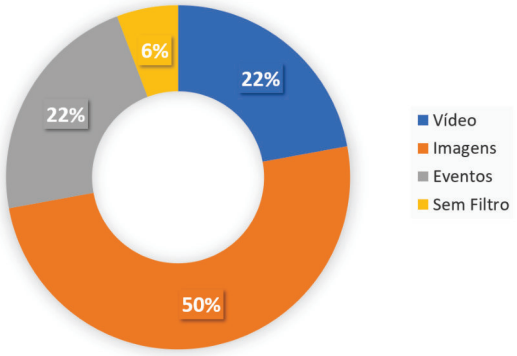


Figura 5.11.: Fluxos Distinguidos pela Gama IP na sessão YouTube 3.

Após refinação, resultaram 7 fluxos relativos a vídeo. Tendo sido visualizados 5 vídeos (publicidade e conteúdo a visualizar) a correspondência vídeo-fluxo revelou que o classificador conseguiu coligir todos os fluxos relativos à sessão. Aqui, estão incluídos fluxos correspondentes à fragmentação do vídeo visualizado para dar lugar ao espaço de publicidade. Nestes 7 fluxos, falta o anúncio inicial, que foi adicionado manualmente para posterior análise de tráfego não solicitado. Do ficheiro `outputUDP.txt` desta sessão, além dos vídeos de anúncio, foram encontrados 10 fluxos indiciados como fluxos de tráfego não solicitado. Estes 10 fluxos prefazem um volume de 36kB num total de 19.2MB, correspondendo a 5% do tráfego de conteúdo em UDP.

Por último, na terceira captura, pelo grafo na Fig. 5.12, observa-se que o servidor de vídeo `s17` começa a ter mais pedidos do que nas anteriormente sessões. O servidor `s17` foi responsável pela entrega do único vídeo desejável e foi interrompido para apresentar publicidade. Daqui resultou em ser 6 vezes solicitado, o que faz com que esteja incluído no grupo de servidores com maior número de fluxos. Quanto aos que entregaram os anúncios, `s09`, `s32` e `s39` são os responsáveis e que tendo, algum peso no que toca a número de fluxos gerados, não correspondem à maioria de servidores. Outro aspeto a realçar ao longo da apresentação de resultados UDP para o YouTube é que servidores que não são de entrega de vídeo e que estão ligados a funcionalidades que necessitem constante atualização, acabam por ser os que geram mais quantidade de fluxos, tal como `s06`, `s22` e `s23` demonstram.

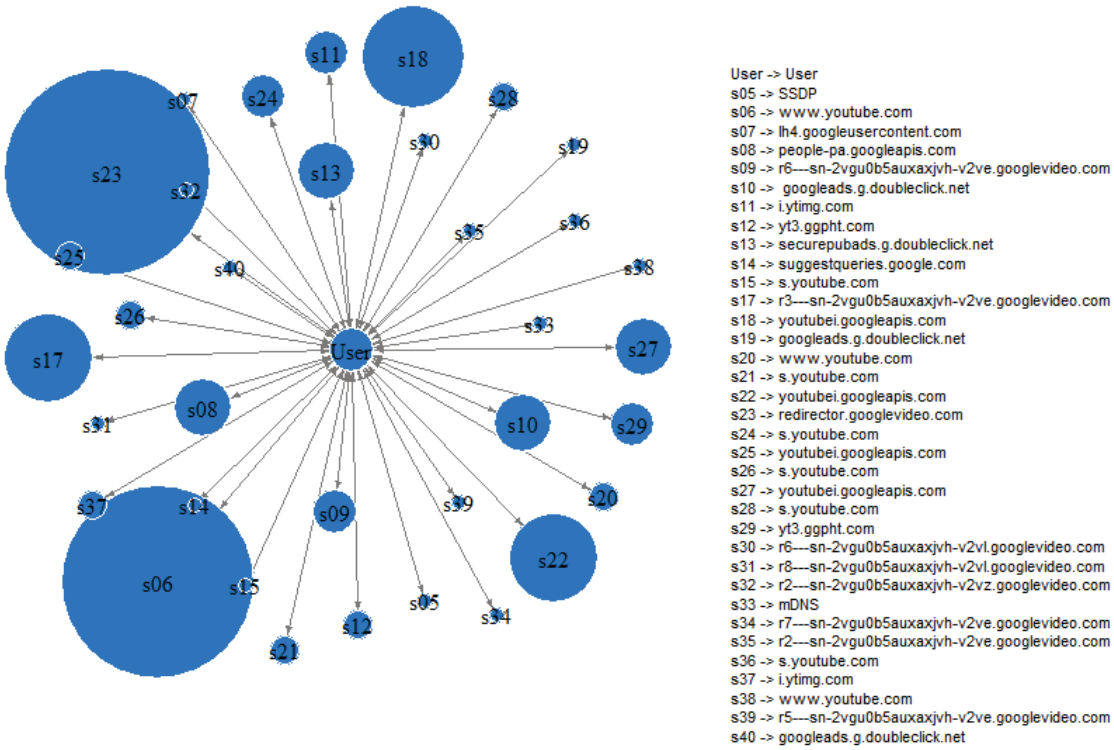


Figura 5.12.: Servidores UDP envolvidos durante YouTube 3.

Da análise UDP, pode-se retirar que o YouTube é uma aplicação pensada sobretudo para entrega de conteúdo via este protocolo. Foram observadas maiores cargas e mais conversas mantidas ao longo desta plataforma por este mesmo motivo. Muitas destas servem para manter QoS e QoE para que seja possível entregar sempre o conteúdo da melhor forma de acordo com as circunstâncias da rede.

Quanto ao tráfego não solicitado, foi observado que vídeos publicitários são entregues por fluxos não identificados como tráfego não solicitado, sendo semelhantes aquando comparados com fluxos de entrega de conteúdo desejável. A sua caracterização tornar-se-ia dificultada se não existisse uma relação vídeo-fluxo inicialmente realizada, por não haver padrão definido para conteúdos publicitários.

TCP incompleto e HTTP

Relativamente ao tráfego HTTP que foi encontrado nas capturas 2 e 3, este não possui qualquer informação relevante para o estudo por serem muito poucos fluxos comparativamente ao total de tráfego TCP e UDP encontrado para esta aplicação em concreto. Além disso, foi demonstrado que o tráfego HTTP foi provocado por outros serviços que não a aplicação YouTube.

Por fim, o tráfego TCP incompleto acaba por ser também desprezado por ter sido nulo ou quase nulo ao longo das sessões. Outro motivo, para o que foi gerado, foi o facto de

não revelar informação relevante e não demonstrar aspetos que ajudem a caracterização da aplicação em si.

5.2.2 *Facebook*

TCP

O tráfego TCP obtido ao longo das três sessões demonstra que a utilização de cifra, mais especificamente o uso de HTTPS, não traz resultados concretos à caracterização de tráfego não solicitado.

A Tabela 5.6 enuncia a quantidade de fluxos TCP do tipo não solicitado face aos fluxos TCP encontrados no decorrer das três sessões. Daqui retira-se que, a comunicação entre servidor de anúncio e cliente inclui fluxos que negociam quais os anúncios que poderá eventualmente apresentar. Em mais detalhe, os fluxos aqui observados foram, na sua maioria, estabelecidos com o servidor cujo FQDN é `www.googleadservices.com`. A partir da caracterização da aplicação YouTube, significa que o servidor pertence à Google e nada tem a ver com a caracterização do serviço Facebook. Contudo, não é excluído da análise pois trata-se de tráfego não solicitado e que apesar de não estar diretamente ligado ao serviço, está ligado ao funcionamento da aplicação em *browser* ou a outra aplicação a correr em segundo plano que naquele instante decidiu intervir. Da tabela destaca-se que, na segunda captura não foi possível obter quaisquer indícios de tráfego não solicitado pelo que resultou em 100% do tráfego TCP a ser classificado como desejável, devido à cifra.

Sessão	Total de Fluxos TCP	Fluxos de Tráfego Não Solicitado	Tráfego Não Solicitado (%)
Facebook 1	7	1	14%
Facebook 2	11	0	0%
Facebook 3	20	3	15%

Tabela 5.6.: Fluxos TCP Obtidos nas Sessões Facebook.

A Tabela 5.7, abaixo ilustrada, demonstra as percentagens obtidas de tráfego não solicitado em relação aos fluxos gerados. Observa-se que, o volume de tráfego não solicitado é similar ao encontrado na aplicação YouTube pois, como indicado acima, os servidores envolvidos são os mesmos a intervir nestas sessões.

Um aspeto relevante foi o aumento de tráfego total transferido face à primeira sessão. Na segunda sessão a maioria do tráfego transferido (680kB) tem origem em apenas um fluxo para um servidor de entrega de vídeo cuja assinatura é `video.fopo1.1.fna.fbcdn.net` e endereço IP é `195.8.13.210`. Para a terceira sessão, este aumento também se verifica, mas a maioria do tráfego tem origem (1094kB) num servidor de entrega de conteúdo generalizado com assinatura igual a `scontent.fopo1-1.fna.fbcdn.net`. Isto deve-se a conteúdo que foi

transferido na sua totalidade por fluxos TCP completo onde não ocorreram interrupções no decorrer destas conversas.

Sessão	Total Transferido (kB)	Total Transferido em Tráfego Não Solicitado (kB)	Tráfego Não Solicitado (%)
Facebook 1	34.9	3.5	10%
Facebook 2	722.5	0	0%
Facebook 3	1396.4	12.5	0.9%

Tabela 5.7.: Volume em kBytes transferidos nas Sessões Facebook.

Conforme o grafo na Figura 5.13, na primeira sessão todos os servidores possuem o mesmo número de fluxos, destacando que a maioria é referente a serviços da Google com especial atenção a `s03` que se trata de um servidor destinado a entrega de tráfego não solicitado. O único servidor afeto ao Facebook é `s05`, cuja assinatura é `edge-chat.facebook.com` e a sua função é fazer chegar e entregar mensagens enviadas através do Facebook Messenger, serviço que está diretamente ligado a este serviço.

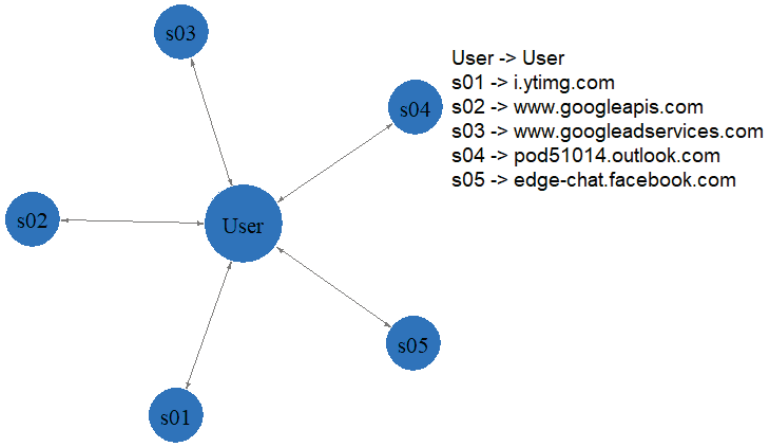


Figura 5.13.: Servidores TCP envolvidos durante Facebook 1.

Para a segunda captura, no grafo da Figura 5.14, observa-se que a maioria dos fluxos são provenientes do servidor de vídeo que obteve maioria de tráfego transferido (`s01`). Os restantes servidores TCP representados no grafo são relativos a outras funcionalidades enunciadas na primeira sessão.

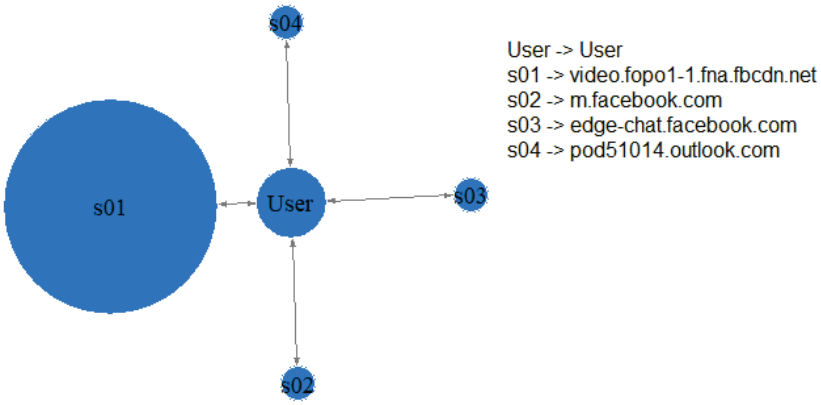


Figura 5.14.: Servidores TCP envolvidos durante Facebook 2.

Quanto à terceira captura, as conversas servidor de anúncios e cliente foram sobretudo para servidores Google com os nomes `www.googleadservices.com`, `ad.doubleclick.net` e `googleads.g.doubleclick.net`, todos sem relação com o Facebook. Recorrendo ao grafo na Figura 5.15 é possível constatar mais uma vez que o servidor com maior número de fluxos TCP é o de vídeo, indo de encontro com a questão da fragmentação do vídeo em partes para que seja possível chegar ao utilizador. De destacar também, os dois servidores `s01` e `s02` por também possuírem mais pedidos que os restantes, estando relacionados com o funcionamento do sistema operativo Android.

Por fim, foi nesta captura onde melhor se observa os nomes dos servidores de distribuição de conteúdo do Facebook. Os servidores `s03`, `s05`, `s06` e `s12` são do Facebook. A indicação que os distingue são as expressões `fbcdn` e `facebook` presentes na assinatura, das quais, se retira que `fbcdn` corresponde à `Facebook CDN`.

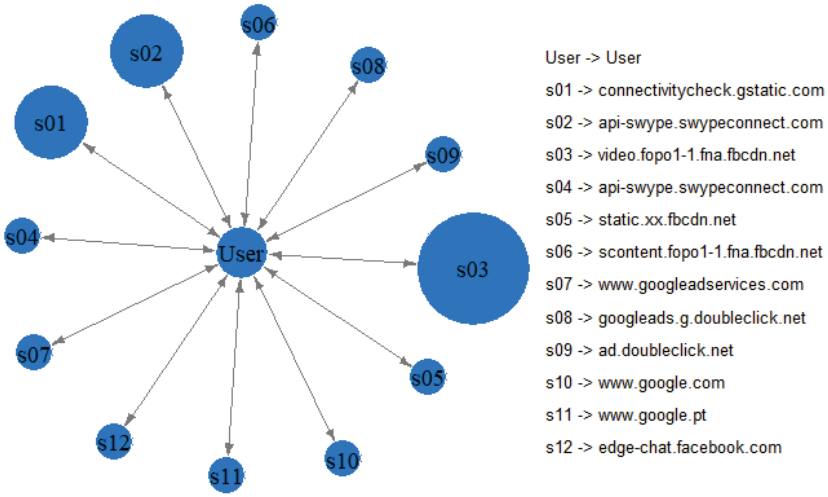


Figura 5.15.: Servidores TCP envolvidos durante Facebook 3.

Em suma, a análise TCP revela dificuldades em retirar conclusões acerca de tráfego não solicitado. A utilização de HTTPS impede os resultados de irem mais longe que o esperado. A análise não está inteiramente concluída porque ainda existe a componente de tráfego TCP incompleto onde se retiram informações que complementam estas aqui obtidas.

TCP incompleto

Passando agora a tráfego TCP Incompleto, a associação de tráfego não solicitado mediante o tráfego observado não existe. No entanto, não havendo tráfego não solicitado, esta análise a tráfego TCP incompleto é realizada para tentativa de dedução de conteúdo entregue por TCP. Como enunciado na Secção 4.2.1 existe uma associação realizada entre nomes e endereços IP conhecidos por meio da análise a TCP completo.

A Tabela 5.8 demonstra os resultados obtidos para cada uma das categorias possíveis de conhecer. Reforça também que houve um aumento de volume transferido face ao tráfego TCP completo. Este aumento justifica-se pelas transferências dos conteúdos registados no ficheiro `.har`, terem sido, na sua maioria, entregues por via TCP. Isto remete a que, ao contrário do que acontece em aplicação, a ligação fica em aberto sendo terminada abruptamente sem que depois se possa deduzir qual o nome do servidor. Além disso, o facto de a ligação estar em aberto permite a chegada de mais conteúdos ao utilizador à medida que este desce na página.

Na tabela é refletida, também, o volume de tráfego calculado referente a entrega de vídeo, imagem e desconhecidos.

Sessão	Total Transferido (kB)	Total Transferido em Vídeo (kB)	Total Transferido em Imagem (kB)	Desconhecidos (kB)
Facebook 1	9448.1	7509.8	1923.5	14.8
Facebook 2	6551.2	0	5329.3	1221.9
Facebook 3	258.2	0	10.6	247.6

Tabela 5.8.: Volume em kBytes transferidos nas Sessões Facebook.

Para as três sessões, os servidores responsáveis por entregar vídeo e imagem possuem FQDN igual a `video.fopo1-1.fna.fbcdn.net` e `scontent.fopo1-1.fna.fbcdn.net` respetivamente. Em acréscimo, na segunda e terceira capturas não foram observadas transferências por parte do servidor de entrega de vídeo, `video.fopo1-1.fna.fbcdn.net`. Pelo que, os vídeos visualizados terão sido transferidos na totalidade e registados em TCP completo pelo Tstat ou entregues pelo mesmo servidor de imagens. Este segundo servidor de entrega de conteúdo denominado `scontent.fopo1-1.fna.fbcdn.net`, tanto pode entregar imagens como vídeo, mas na maioria entrega imagens, e consequentemente foi definido que o classificador iria indicar como sendo entrega de imagens.

Em conclusão, à análise TCP incompleto, todo o tráfego observado recai sobre o que foi observado na página inicial do Facebook e que deverá ter associação com a análise ao ficheiro `.har`. Quanto ao tráfego TCP classificado na categoria "Outros", foi constatado que a maioria correspondia a servidores do Facebook responsáveis por entrega de conteúdo de manutenção da página, conteúdo estático e ainda outros serviços relacionadas com a rede de distribuição do Facebook.

UDP

Para a análise UDP, começa-se por ver qual a porção de tráfego DNS face ao tráfego classificado como sendo referente a conteúdo. Assim a Tabela 5.9 demonstra a quantidade de fluxos referentes a DNS e quantidade de fluxos tomados como conteúdo. Esta tabela reforça que o Facebook utiliza o protocolo UDP para resolução de nomes e marginalmente para transferir conteúdo. Observando os valores obtidos de conteúdo transferido, o seu volume é baixo quando comparado com fluxos TCP no Facebook.

Sessão	Total de Fluxos	Total de Fluxos DNS	Total de Fluxos Considerados Conteúdo	Total de kBytes Transferidos em Conteúdo
Facebook 1	24	16	8	13.9kB
Facebook 2	37	31	6	4.9kB
Facebook 3	49	38	11	13.8kB

Tabela 5.9.: Fluxos UDP nas Sessões Facebook.

Da tabela de fluxos UDP encontrados, passa-se à análise de protocolos encontrados em UDP. A Tabela 5.10, abaixo ilustrada reflete a quantidade de fluxos encontrados para cada um dos protocolos enunciados na subsecção 4.2.2. Aqui, os fluxos com maior interesse são os de TLS porque são os podem transportar tráfego não solicitado.

Sessão	UnPnP	NetBIOS	TLS	WSDiscovery	mDNS
Facebook 1	1	2	3	2	0
Facebook 2	3	2	1	0	0
Facebook 3	3	2	4	2	0

Tabela 5.10.: Tipos de Fluxos UDP nas Sessões Facebook.

Da primeira sessão retira-se o facto de ter sido observado apenas um fluxo que continha indícios de tráfego não solicitado. Este fluxo equivale a uma largura de banda gasta igual a 4.3 kB perfazendo 31.1% face ao total de 13.9 kB. Além disso, este fluxo foi proveniente do servidor cujo FQDN é `googleads.g.doubleclick.net` que, mais uma vez, nada tem a ver com o serviço em questão, mas que é contabilizado.

Por sua vez, na segunda captura, não foi observada qualquer ocorrência de fluxos de tráfego não solicitado, podendo-se dizer que 100% do tráfego UDP tratava-se de tráfego solicitado.

Por fim, na terceira captura foi também possível obter tráfego não solicitado em UDP. Um fluxo com carga igual a 4.7 kB foi detetado nos 4 fluxos referentes ao TLS. A percentagem de tráfego não solicitado face ao total transferido (em *kBytes*) foi de 33.9%, excluindo pedidos DNS.

Esta análise ao tráfego UDP pretende demonstrar o quão o Facebook foi desenvolvido para utilizar o protocolo TCP. O UDP neste serviço tem como funcionalidade principal descobrir localizações de servidores através do DNS. Tal observação, foi também confirmada na versão aplicação pois, de uma breve análise, verificou-se que os fluxos sobre UDP, na sua totalidade apenas são utilizados para DNS e outros protocolos de descoberta de dispositivos na rede (UnPnP, NetBIOS e WSDiscovery) aqui analisados.

HAR

Chegando à análise dos ficheiros `.har`, começa-se por ver que conteúdos são encontrados ao longo da captura. Considerando nas três sessões do Facebook, a Tabela 5.11 reflete a quantidade de ligações encontradas para cada uma das categorias encontradas: conteúdo estático, conteúdo na página inicial (*feed*), fotografias de perfil e outros conteúdos. Aqui o conteúdo classificado como "Outros"não possui qualquer tipo de filtro adequado por não se saber em que gama se encontra.

Numa visão geral, o conteúdo estático é o que obtém a maioria de ligações ao longo das sessões. Nas fotografias de perfis, note-se que não foram transferidas por motivos de privacidade. Para a análise HAR, os conteúdos mais relevantes são os conteúdos apresentados na página inicial do Facebook.

Sessão	Total de ligações	Conteúdo Estático	Conteúdo Página Inicial	Fotografias de Perfil	Outros Conteúdos
Facebook 1	380	136	94	39	111
Facebook 2	357	207	107	43	0
Facebook 3	402	206	134	53	9

Tabela 5.11.: Conteúdo Encontrado nas Sessões Facebook.

Avançando para a classificação de tráfego não solicitado, como explicado na subsecção 4.2.3, a análise assimila os conteúdos da página inicial e verifica quais destes são referentes a anúncios. Para isso a Tabela 5.12 possui como dados a quantidade de ligações de conteúdos da página inicial e seu volume em MB. Inclui ainda, a quantidade de ligações relativas a tráfego não solicitado e respetivo volume em MB.

Sessão	Conteúdo Página Inicial	Conteúdo Transferido (MB)	Ligações Não Solicitadas	Ligações Não Solicitadas (MB)
Facebook 1	94	13.7 MB	8	4.3 MB
Facebook 2	107	5.7 MB	24	2.0 MB
Facebook 3	134	7.3 MB	19	1.7 MB

Tabela 5.12.: Total de Ligações Não Solicitadas nas Sessões Facebook.

Para a primeira sessão sabe-se que foram encontradas 7 imagens relativas a anúncios e que uma destas se tratava de um *thumbnail* de vídeo que, por sua vez, fora classificado também como anúncio. Ficando com um total de 8 conteúdos publicitários. Juntando o que apresentado na Tabela 5.11 com o que foi apresentado na Tabela 5.12, retira-se que 15% da largura de banda disponível foi gasta em tráfego não solicitado, correspondendo a 2% das ligações retiradas do ficheiro *.har*.

No entanto, consegue-se ir mais longe com esta questão. Para saber se uma aplicação é gananciosa ou não (parâmetro divulgado na Secção 3.3) é necessário saber se uma aplicação apresenta demasiados anúncios. O classificador, ao possuir todo o conteúdo transferido e o distinguir em várias categorias consegue saber qual a percentagem da página inicial que corresponde a anúncios. Então, desta primeira captura, verificou-se que 9% do que foi visto eram anúncios e que a sua carga total ocupava 31% do total transferido para esta página em específico.

Seguidamente, da segunda captura, retiram-se 23 ligações referentes a imagens publicitárias. O número 23 pode ser excessivo quando visto no total de publicações (8 publicações com a indicação "Patrocinado"), contudo, estas ligações correspondem de facto a 23 imagens de anúncios que resultam de publicações com múltiplas imagens numa só publicação. Normalmente são publicadas em conjuntos de 4 a 6 imagens e, como neste caso apareceram 3 publicações publicitárias deste tipo, surgiram 17 imagens deste género de publicações, ao qual o Facebook os denomina de Carrossel e Coleção[10]. Juntando a estas imagens um vídeo de caráter publicitário também encontrado, resultam 24 conteúdos não solicitados. Da perspetiva geral de tráfego observado no `.har`, 7% das ligações eram relativas a anúncios, prefazendo um total de 15% do tráfego transferido.

Mais uma vez focando apenas no tráfego transferido para a página inicial, a quantidade de pedidos de anúncios aumenta para 22%, ou seja, 24 em 107 pedidos de conteúdo são anúncios. Desta feita, 2.0 MB em 5.7 MB, corresponde a um total de 36% gasto em largura de banda exclusivamente para esta página. Aqui é notório que o Facebook torna-se invasivo ao preencher alguns espaços com publicidade, revelando alguns comportamentos que se tornam padrão.

Por fim, na última sessão, quanto ao comportamento do tráfego não solicitado, foi verificado que 5% eram anúncios, correspondendo a 18 imagens e um vídeo. A partir da carga total transferida de todas as ligações, este conteúdo corresponde a 11%, o que mostra

que não é muito significativo em termos de impacto num plano de dados móveis quando comparado com as outras sessões.

Contudo, olhando apenas para o tráfego considerado conteúdo, as ligações obtidas correspondem a 14%, onde este aumento se justifica devido à restrição do tráfego (geral para página inicial). Assim sendo, as ligações de anúncios correspondem a 23%, o que por si só indica que na página inicial 23 publicações em 100 são anúncios.

Em suma, a análise HAR consegue entrar em mais detalhe que os restantes protocolos vistos para o Facebook. Tal nível de detalhe, deve-se à posse de todo o conteúdo visualizado no decorrer das sessões em estudo. Esta análise revela alguns comportamentos do Facebook em relação ao tráfego não solicitado que serão interpretados na discussão de resultados.

5.2.3 *Instagram*

TCP

Ao contrário do que foi feito para as anteriores plataformas, o Instagram funciona de maneira um pouco diferente no que diz respeito a entrega de tráfego não solicitado. Como dito anteriormente, esta primeira captura foi retirada da componente *web* deste e as restantes servirão de reflexão sobre a aplicação correspondente.

Na primeira sessão resultaram 5.6 MB de tráfego TCP transferido, nos quais a totalidade de fluxos (19) eram referentes a HTTPS. Não sendo nenhum identificado como anúncio, avança-se para as conversas servidor cliente observadas em TCP. O grafo na Figura 5.16 mostra que os servidores `s02`, `s03` e `s05` foram os que mais vezes foram solicitados. Destes, apenas `s02` é responsável por entregar conteúdo para a página do serviço. Focando apenas no servidor `s03`, este respondeu com dois nomes onde se retiram as expressões `chat` e `graph` indicando que `s03` é responsável por gerir tanto o serviço de mensagens Instagram como o serviço dos grupos e sugestões de páginas. De destacar também, a intervenção de servidores pertencentes à plataforma Facebook que, ao contrário do que aconteceu no YouTube onde serviços externos não tinham qualquer destaque. Aqui `s05` e `s06` respetivamente são responsáveis por serviços Facebook aos quais o Instagram acede para obter e mostrar o conteúdo. Desta forma, verifica-se que ambos os serviços, apesar de distintos, estão relacionados.

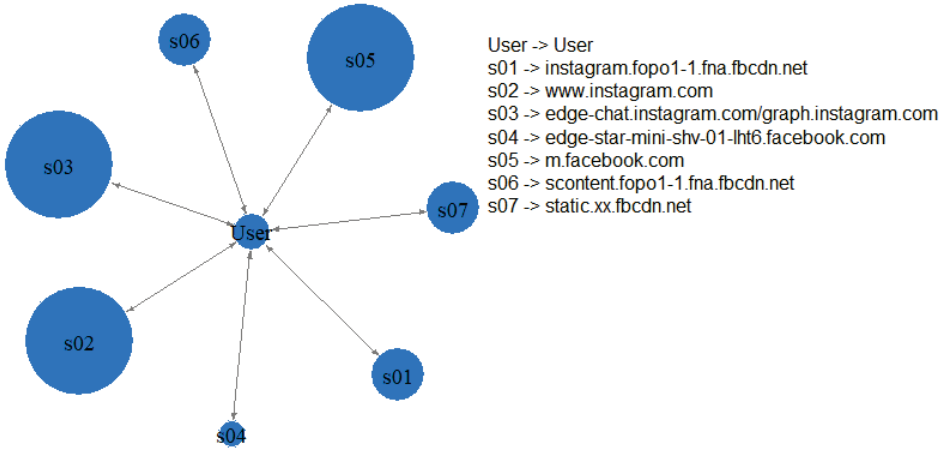


Figura 5.16.: Servidores TCP envolvidos durante Instagram 1.

Para a segunda sessão, foi utilizada a aplicação Instagram da qual resultaram 21 fluxos TCP e nenhum deles foi classificado como não solicitado. Todos eles utilizam as portas 443 e 993 pelo que, recorrem a HTTPS. Observando o grafo na Figura 5.17, o servidor com mais pedidos realizados foi o de entrega de conteúdo na página inicial da aplicação (`s02`). Tratando-se de múltiplos fluxos, significa que, os conteúdos são entregues em conjuntos. Para esses conjuntos não existe método que consiga isolar imagens, vídeos e anúncios pelo que, é aqui que se revela mais difícil a caracterização deste serviço. Um aspeto a ser referido é, a quantidade de *bytes* ter aumentado para 41.5 MB o que demonstra que a reprodução automática de vídeos faz aumentar significativamente o tráfego. Comparando com a anterior sessão, para que um vídeo fosse reproduzido ter-se-ia de dar a indicação de reprodução. Por este motivo, o servidor `s02` consegue ter fluxos com cargas altas quando comparadas com os restantes.

Para esta segunda sessão, consegue-se, mais uma vez, observar a intervenção de servidores do Facebook, `s03`, `s06` e `s08`. Quanto ao primeiro, é responsável por gestão de grupos e pessoas às quais possuem Facebook e Instagram. O segundo é responsável por entrega de conteúdos externos ao Facebook, ou seja, quando uma notícia é publicada no Facebook ou Instagram a imagem que aparece na publicação é muitas vezes a imagem que aparece no artigo na página da notícia. Desta forma, a plataforma importa o conteúdo dessa imagem e coloca num servidor denominado `external.fopo1-1.fna.fbcdn.net` referenciando que o conteúdo não foi publicado pela plataforma, tratando-se de um recurso externo a esta. Por fim, `s08` caracteriza-se por ser um servidor de gestão de mensagens do qual se destaca a expressão `mqtt` no nome de `s08`. A expressão refere-se a um protocolo desenvolvido para *messaging* (MQTT - *Message Queuing Telemetry Transport*).

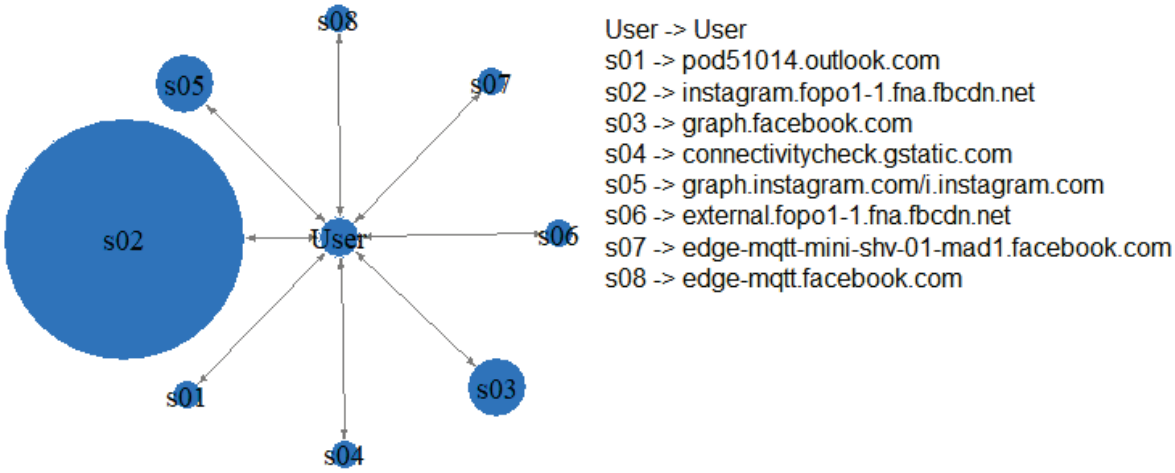


Figura 5.17.: Servidores TCP envolvidos durante Instagram 2.

Por último, a terceira sessão serve como complemento ao que foi observado na segunda captura, acrescentando mais nomes aos quais os servidores podem estar associados. Observando o grafo na Figura 5.18, `s02` foi o servidor mais vezes chamado e a sua responsabilidade é fazer chegar conteúdo Instagram ao utilizador, quer seja desejado ou não. Outro servidor com a mesma responsabilidade é `s03`, que foi correspondido a dois nomes, `external.fopo1-1.fna.fbcdn.net` anteriormente visto e ainda `scontent.fopo1-1.fna.fbcdn.net`. Este segundo nome é importante aparecer nesta aplicação, pois trata-se do servidor que faz entregas de conteúdos no Facebook, no qual recaem a maioria dos pedidos. Isto reforça a ideia de aplicações diferentes pertencentes à mesma entidade, estarem relacionadas de alguma forma.

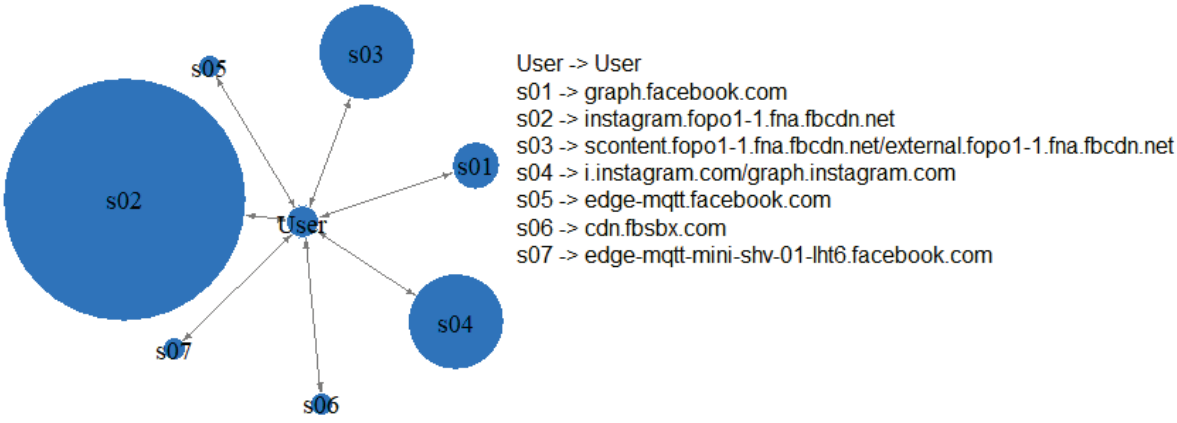


Figura 5.18.: Servidores TCP envolvidos durante Instagram 3.

A análise do protocolo TCP no Instagram revela poucos detalhes referentes ao tema da caracterização de tráfego não solicitado. O Instagram, ao incluir os anúncios nos servidores

de entrega de conteúdo desejável, torna qualquer método de caracterização de anúncios um pouco inútil devido à implementação da cifra na camada de transporte.

TCP incompleto

Avançando para tráfego TCP incompleto, quanto à primeira sessão do Instagram, é possível deduzir 29kB transferidos. O grafo ilustrado na Figura 5.19 indica as conversas que existiram ao longo da sessão. Não sendo possível demonstrar o valor da assinatura, os nomes indicados são referentes ao valor resultado do comando `nslookup` ao endereço IP destes. Na legenda, verificam-se outros nomes que não os anteriormente observados porque o `nslookup` mostra o nome no estado mais cru, dando o nome verdadeiro do servidor e não da assinatura obtida em TCP.

Pela aprendizagem resultante da análise tomada em TCP, `s01` e `s02` correspondem aos endereços IP de `graph.instagram.com` pelo que iria ao encontro a essa funcionalidade em específico. Por sua vez, `s03` corresponde a `facebook.com` e `s04` a `static.xx.fbcdn.net`, ambos pertencentes à plataforma Facebook.

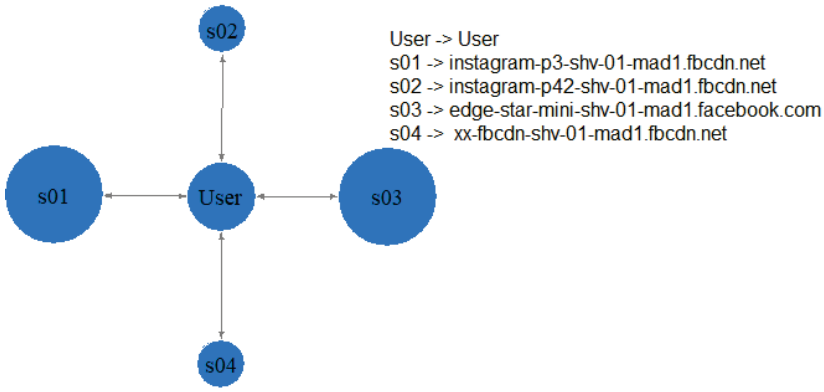


Figura 5.19.: Servidores TCP interrompidos durante Instagram 1.

Para as outras duas sessões foram observados apenas 2 e 1 fluxos referentes a TCP incompleto. Por serem de baixa carga, não trazem relevância para o estudo pois não acrescentam informação à anteriormente retirada.

UDP

Conforme observado no Facebook, o UDP é utilizado, sobretudo, para resolução de nomes. Para o caso do Instagram, em todas as capturas realizadas resultou na maioria do tráfego UDP ser relativo a DNS. Isto perfez uma média total igual a 80% no tráfego observado para as três captura. Podendo adiantar que, não ocorreram fluxos que continham conteúdo, ou seja, os restantes tratam-se dos protocolos refletidos em outras aplicações.

Para os restantes protocolos observados, não existe conteúdo a ser transferido usando UDP. Não existem particularidades observadas neste protocolo e retira-se a mesma conclusão que para o Facebook, já que estes serviços têm por base o TCP na sua camada de transporte com recurso ao HTTPS.

HAR

A diferença principal entre Instagram e Facebook, pelo menos na versão *web*, é a não existência de quaisquer campos de anúncios. Quer isto dizer que não conta com tráfego não solicitado de cariz publicitário em toda a experiência de interação com esta rede social.

O gráfico apresentado na Fig. 5.20 enuncia os tipos principais de conteúdos encontrados ao longo da sessão. Sendo 40% correspondente a conteúdos estáticos, 42% conteúdo desejável, 18% a fotos de perfil e ainda 0% na categoria "Outros". Neste conteúdo desejável, todo correspondia a imagens estáticas e não publicitárias. A justificação para apenas ter aparecido tráfego do tipo imagem e não vídeo foi por não se ter forçado a reprodução deste.

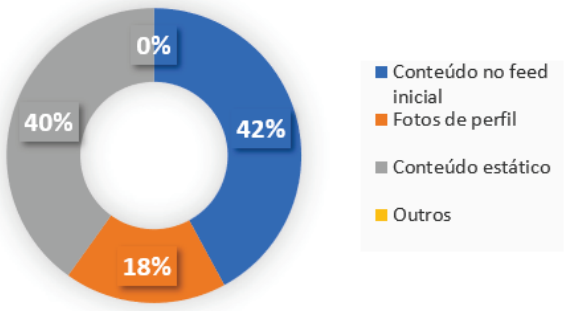


Figura 5.20.: Conteúdo Encontrado na Sessão Instagram 1.

Relativamente às outras sessões, não existe análise a este ficheiro em particular por terem sido refletidas na interação aplicação-rede. Não havendo nada a acrescentar a esta sessão pela falta de presença de publicidade e não havendo mais *outputs* de *HTTP Archive*, dá-se por terminada a fase de apresentação de resultados referentes ao Instagram.

5.3 DISCUSSÃO

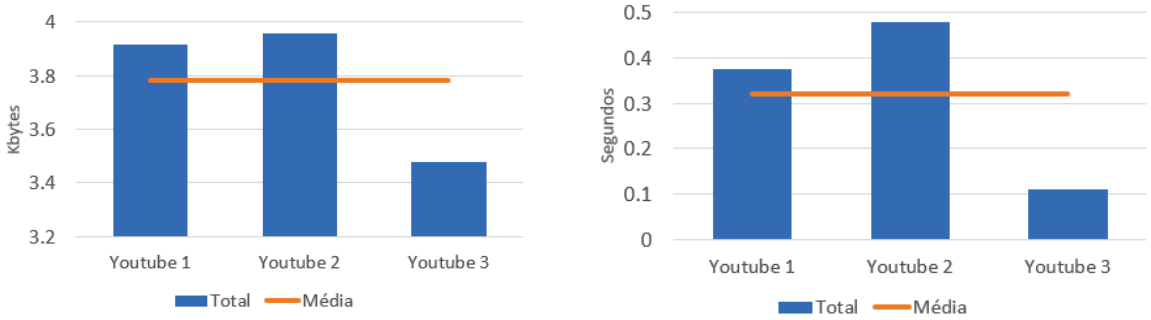
Nesta secção é feita a discussão dos resultados apresentados na secção anterior de forma a determinar os parâmetros estabelecidos em 3.3 e dar o estudo por terminado.

5.3.1 *Resultados YouTube*

Conforme os grafos de conversas servidor cliente ilustrados anteriormente, no YouTube os servidores de anúncios acabam por ser poucas vezes solicitados em TCP e UDP, não sendo contactados a maioria das vezes. Ainda que com nomes e propósitos diferentes, foi possível observar que muitas vezes existem servidores em que acabam fluxos estabelecido nada têm a ver com tráfego não solicitado.

Para o protocolo TCP, observando estes fluxos não solicitados mais detalhadamente, os seus tamanhos variam entre 3 e 5 *kBytes*. Isto significa que, podendo ser conteúdos com alta compressão (imagens), na realidade correspondem a simples interação entre servidor de anúncios e cliente, que permite chegar a acordos sobre qual o conteúdo a entregar a partir de dados enviados pelo cliente. Tal conclusão, foi possível por se tratarem de fluxos de curta duração, não mais que 1 segundo, e que possuem poucos *bytes* a ser entregues tanto ao servidor como ao cliente.

O gráfico na Fig. 5.21a reforça que o tamanho médio total dos fluxos é igual a 3.8 kB e o gráfico na Fig. 5.21b indica a sua duração média que é de 0.32 segundos. Estes valores caso correspondessem a imagens publicitárias necessitariam de ser de baixa resolução ou estarem altamente comprimidas para depois serem descomprimidas e mostradas. Uma imagem com tais tamanhos teria de ter uma resolução até 100x100 (caso fosse quadrada), o que corresponde aos tamanhos de ícones e imagens de perfis. Além disso, sendo tamanhos pequenos, é natural observar durações de fluxo reduzidas, na ordem das décimas de segundo.

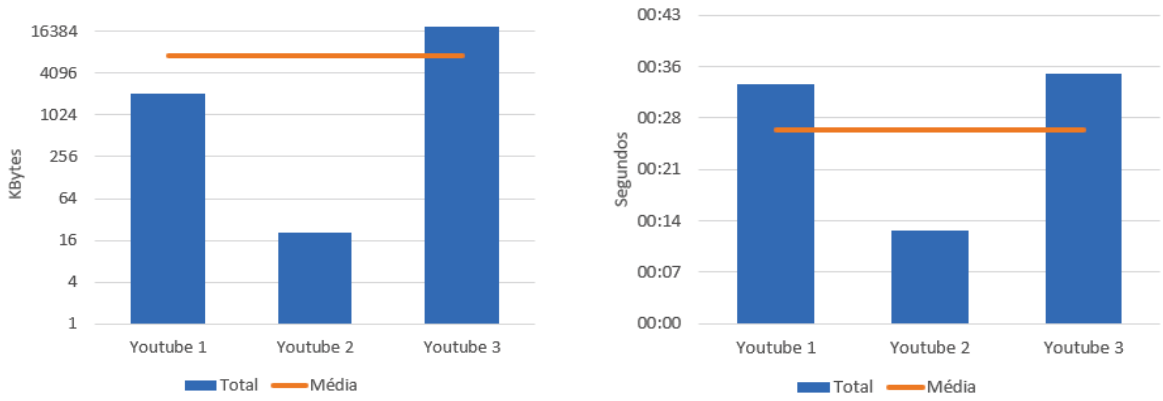


(a) Média de Tamanho de Fluxos

(b) Média de Duração dos Fluxos

Figura 5.21.: Tamanho e Duração Média ao longo das Sessões YouTube em TCP.

Por sua vez, em UDP, pelo gráfico na Fig. 5.22a, verifica-se que a média de *MBytes* transferidos anda na ordem dos 7.2 MB, indicando que os conteúdos publicitários apesar de curtos, num plano de dados móveis seriam gastos para algo não desejável durante a experiência de visualização. Em comparação com o total de largura de banda gasto ao longo das sessões, este valor é bastante reduzido. Isto deve-se à qualidade de vídeo entregue. As resoluções observadas para este tipo de conteúdo são iguais a 480x360 (360p), 858x480 (480p) e apenas uma vez 1280x720 (720p), querendo dizer que, para este tipo de conteúdo, a qualidade de vídeo não é a prioridade mas sim a mensagem. Resoluções baixas conduzem a tamanhos baixos e consequentemente faz com que a latência de entrega deste conteúdo seja baixa, permitindo uma entrega mais rápida ao utilizador sem haver tempo de espera (*buffering*). Outra observação foi que a duração média deste tipo de vídeo foi de 27 segundos, ver Figura 5.22b, demonstrando que é acrescentado cerca de meio minuto ao tempo de visualização. Além disso, este valor pode-se repetir múltiplas vezes, o que acrescenta um *overhead* ao tempo disponível para visualizar os vídeos desejados.



(a) Média de Tamanho de Fluxos

(b) Média de Duração de Vídeos

Figura 5.22.: Tamanho e Duração Média dos Vídeos ao longo das Sessões YouTube em UDP.

A média obtida está de acordo com a documentação do YouTube [32], onde vídeos os publicitários vistos se encaixavam no perfil de *Skippable video ads* e *Non-Skippable video ads*. Para o primeiro, não existe limite de tempo pois passados 5 segundos podem ser avançados. Para o segundo, um vídeo não deve ter mais do que 20 segundos e não pode ser avançado. Contando que para o primeiro género de vídeo as durações mais comuns são de 30 a 45 segundos, a média de 27 segundos está quase dentro dos valores esperados.

Ao longo destas sessões verificou-se que, vídeos de anúncios estão diretamente ligados à localização do utilizador, ou seja, para o país de Portugal a linguagem e legendas estavam em português, e que são entregues por servidores de vídeo normais sem quaisquer indicações de que se tratam de publicidade. Quanto aos que são explicitamente indicativos de tráfego não solicitado, os nomes dos servidores intervenientes são os mesmos durante to-

das capturas da aplicação YouTube, mais especificamente, `googleads.g.doubleclick.net` e `securepubads.g.doubleclick.net`. Estes nomes estão a associados à questão de decisão de conteúdo publicitário a ser mostrado e aparecem com endereços IP diferentes associados, assegurando, assim, a entrega de anúncios ao utilizador. É aqui que a componente de personalização é desencadeada. Dentro das subscrições do utilizador, vídeos visualizados e até mesmo páginas visitadas fora do serviço possuem diversos parâmetros guardados em *cookies* destinados a este propósito.

Outro detalhe observado na segunda sessão foi a variável da popularidade do vídeo. Esta indica que, se um vídeo está dentro das tendências atuais terá um maior alcance no público alvo, pelo que obterá melhores resultados no que toca à remuneração dos anúncios mostrados aos utilizadores. Ora se um vídeo foi publicado com a monetização como objetivo, a publicidade atuará conforme o alcance que este terá. Nesta segunda sessão, o primeiro vídeo tinha sido publicado, à data, com 1 dia de antecedência pelo que as suas visualizações ainda estavam a crescer. A acrescentar a este ponto, este primeiro vídeo, não possuía tantos espetadores como os vídeos a seguir visualizados, pelo que provocou um impacto na restante sessão, uma vez que a publicidade começou a intervir no fim deste e no início do terceiro vídeo, e daí ser a que menos tráfego deste género possui.

De uma forma geral, na perspectiva de tráfego de rede, foi possível classificar todos os vídeos que apareceram ao longo das sessões, ainda que com intervenção manual. Desta combinação surge necessidade de calcular a precisão dos resultados obtidos. Para tal, são contados todos os vídeos vistos, incluindo todos os anúncios, e contados todos os vídeos resultantes da análise refinada efetuada na fase de classificação. Por fim, é deduzido a sua acuidade. Para esta plataforma foram obtidos 75%, 83% e 71% para cada uma das sessões, resultando numa média de 76% de sucesso em deteção de vídeo em fluxos UDP registados.

Focando agora no tráfego não solicitado, foram contados todos os anúncios vistos, vídeo e imagem, e todos os anúncios classificados. De seguida as percentagens 100%, 33% e 80% respetivamente foram obtidas, considerando o valor de visto e o valor obtido. No final perfaz uma média final de 71%. Este valor demonstra que existiram alguns fluxos que não se conseguiu interpretar o fluxo correspondente.

Assim, da aplicação YouTube pode-se concluir que os formatos mais vistos são vídeo (antes, durante e no final de um vídeo em visualização) e ainda imagens estáticas na página de sugestões inicial. O volume de tráfego não solicitado corresponde a uma média total de 7.2MB, ou seja, cerca de 2% do tráfego em análise. Conforme salientado anteriormente, estes valores estão dependentes da quantidade de vídeos, duração dos mesmos e popularidade associada. O distribuidor destes anúncios trata-se da própria Google. A distribuição efetua-se a partir de servidores locais pertencentes a operadores Internet também locais, tendo origem em canais YouTube de empresas que pretendem divulgar produtos ou serviços. Não foram encontrados padrões porque, como foi visto, um vídeo

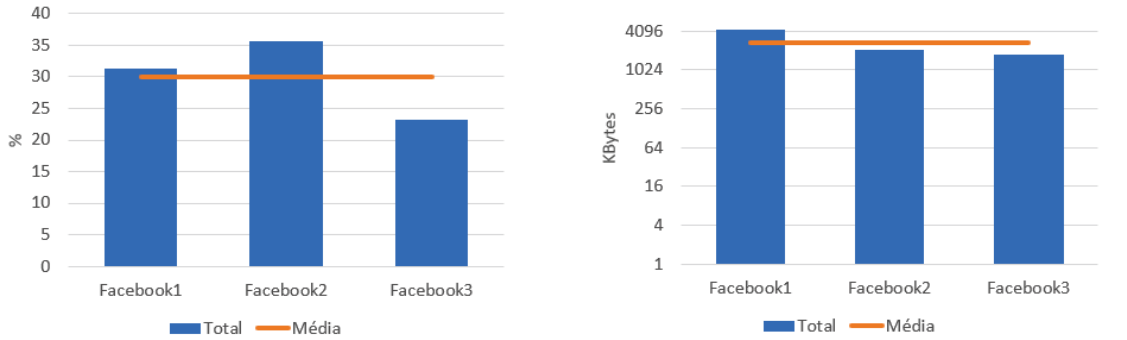
de curta duração possuía quase a mesma quantidade de publicidade apresentada que um vídeo mais longo. Além disso, também foi observado que múltiplos vídeos seguidos podem trazer ou não mais tempo de publicidade.

Para concluir, considerando todas as métricas retiradas, pode-se dizer que a aplicação YouTube trata-se de uma aplicação gananciosa mesmo que não apresente muito conteúdo publicitário. Isto significa que, os serviços sobre TCP estão sempre ativos para chegar a acordos de entrega de publicidade ainda que esta não seja entregue. Este comportamento embora diminuto é algo que foi sempre observado nas 3 sessões apresentadas. Complementando a análise com a entrega por UDP, o YouTube acrescenta um *overhead* desnecessário de tráfego não solicitado, tendo impacto no plano de dados.

5.3.2 *Resultados Facebook*

Dos protocolos de transporte TCP e UDP, não se retiram dados concretos em relação ao tráfego não solicitado. É necessária uma observação na camada protocolar acima para que a classificação deste tráfego possa ser realizável. Contudo, observou-se que o protocolo escolhido para transporte é TCP por ser orientado à ligação e assim facilitar a implementação de HTTPS.

Considerando o resultado obtido nas três capturas dos resultados `.har`, os conteúdos não solicitados perfazem uma percentagem média de 29% ao longo da página inicial do Facebook. Em termos médios, estes conteúdos totalizam cerca de 2.7MB e correspondem na maioria a imagens e, pelo menos uma vez, a vídeo, para sessões cuja duração ronda os 3 minutos. Claro está que, ao ser repetido este processo de consulta, começa-se a evidenciar um impacto maior e esta percentagem aumenta. O gráfico ilustrado na Fig. 5.23a reflete as percentagens obtidas de anúncios, enquanto a Figura 5.23b ilustra o volume de dados correspondente.



(a) Percentagens Obtidos de Anúncios

(b) Quantidade de kBytes Transferidos em Anúncios

Figura 5.23.: Quantidade de Fluxos e Tamanho ao longo das Sessões do Facebook.

Das capturas pode-se retirar que o conteúdo não desejado é entregue da mesma forma que conteúdo desejado. Não existem padrões definidos na camada de transporte para que se possa constatar que um fluxo corresponde a entrega de um anúncio. Contudo, por observação da página foi possível observar alguns padrões visuais em como se estruturam as publicações. O primeiro padrão encontrado é dado o seguinte: 1 publicação por parte de pessoas ou página seguida, 1 anúncio, 4 publicações semelhantes à inicial, 1 anúncio, repetindo-se até uma percentagem de página percorrida. Daqui também se destaca outro padrão observado: 1 publicação por parte de pessoas ou página seguida, 1 anúncio, 3 publicações semelhantes à inicial, 1 anúncio, repetindo-se igualmente tal como o primeiro. Destes padrões, o primeiro foi mais vezes encontrado por ser identificado também na componente aplicação deste serviço. Estes padrões apenas se quebram quando é feito um *refresh* à página dando a origem a uma ordem aleatória onde não é possível identificar padrão.

Tal como aconteceu para o YouTube, a localização desempenha um papel muito importante. Ao conseguir obter a localização, consegue-se personalizar o conteúdo para um público abrangente de uma zona em específico. Contudo este parâmetro, por si, não é suficiente. Por isso partir de *cookies* retirados de páginas recentemente consultadas, o Facebook consegue saber qual o conteúdo que o utilizador está mais interessado em receber. Esta personalização justifica-se por se tratar de uma rede social onde interesses múltiplos são divulgados, tornando o alvo mais pessoal que o YouTube. Os *cookies* retirados incluem: localização, plataforma pela qual é visitado, páginas recentemente visualizadas, interesses da pessoa em questão (políticos, sociais, económicos) e ainda estatísticas que vão desde medição de desempenho de certas campanhas publicitárias até medição de tempo gasto no Facebook por utilizador. Estes interesses são estudados intensivamente pelas redes sociais, chegando ao ponto de se poder tornar invasivo em termos de privacidade.

Em relação à precisão do resultados obtidos, o classificador consegue indicar quantos conteúdos correspondem a anúncios, o que requere a existência de conhecimento adquirido ao longo da fase de testes para que o possa fazer. Desta forma, considerando todos os conteúdos obtidos, verificou-se que nem todos os anúncios são devidamente caracterizados. Assim sendo, para a primeira captura foi obtida uma precisão igual a 100%, para a segunda 92% e, finalmente, 72% para a terceira. De notar que os valores são referentes apenas a anúncios presentes em publicações com a indicação de "Patrocinado", significando que páginas que não possuam tal indicação não foram contadas. Quer isto dizer que o classificador é eficaz em 88% dos casos apresentados.

Em conclusão, o volume médio total retirado foi de 2.7 MB o que equiparando com o que aconteceu na página inicial, corresponde a 29% do que foi visualizado. Esta correspondência apenas é possível por terem sido transferidos todos os conteúdos nas páginas das três sessões. Tendo em conta que 29 imagens em 100 são publicidade, este serviço torna-se ganancioso em obter o máximo de rentabilização por visualização e por clique, aumentando

tráfego deste tipo. Os formatos mais recorrentes são as imagens por melhor atraírem os utilizadores e conseguir melhor sucesso na divulgação [9]. A ocorrência de vídeo está quase sempre presente, contudo, apareceu apenas uma vez por sessão. A distribuição é toda ela da responsabilidade do próprio Facebook e ao contrário do YouTube, os servidores que respondem à maioria dos pedidos localizam-se fora do território português, mas próximos deste. Destaca-se o facto de terem existido algumas exceções em vídeos e imagens que chegaram por servidores locais para um melhor desempenho de correspondência, e cujas assinaturas correspondiam aos nomes `scontent.fopo1-1.fna.fbcdn.net` e `video.fopo1-1.-fna.fbcdn.net`.

5.3.3 *Resultados Instagram*

Chegando à última componente de estudo, e a que menos sucesso teve no processo de caracterização, o Instagram possui muitas semelhanças com o Facebook desde padrões visuais de publicidade até aos servidores de entrega. Claro está que as assinaturas de servidores são diferentes das do Facebook, mas os nomes não diferem muito entre ambos. Algumas dessas parecenças são os servidores `graph.instagram.com` e `graph.facebook.com` cuja função é a gestão de ligações envolvidas, nomeadamente grupos e reações a certas publicações bem como serviços de mediação destes. Outro servidor também similar em nome e função é `edge-chat.instagram.com` e `edge-mqtt.facebook.com` cuja função é atualizar e gerir a funcionalidade de *messaging*, onde no Facebook se trata de um serviço separado (Facebook Messenger). Este aspeto demonstra uma certa consistência no critério de nomeação usado para os servidores de ambos os serviços.

Ao longo das sessões foram estudadas todas as formas de interação com o serviço. Começando com a questão das "*Instastories*", trata-se de um conceito iniciado pelo Snapchat e que foi adaptado ao Instagram, obtendo bastante sucesso tal que o Facebook implementa também uma funcionalidade de Histórias. O Facebook apenas consegue ter tantos utilizadores nesta funcionalidade como o Instagram devido à aplicação associada Facebook Messenger. No entanto, esta aplicação não é tão utilizada porque o Facebook possui outras funcionalidades que a podem tornar um pormenor da página. Focando no Instagram, a funcionalidade "*Instastories*"trata-se de um dos alvos para criar publicidade em formato de vídeo curto e que está muito presente em transações entres histórias. Das três sessões, este comportamento apenas foi visto nas duas últimas por haver mais facilidade na reprodução.

Outro tipo de interação, mais tradicional, é trazer publicidade em formato nativo contida nas publicações de páginas e pessoas às quais se segue. Aqui os tipos de anúncios não variam muito do Facebook, sendo imagens tanto singulares como em coleção e vídeos também eles singulares ou em coleção. Daqui conclui-se que não foi encontrado qualquer padrão na apresentação de publicidade.

Por um processo de inferência e tendo em conta que o tráfego não solicitado foi transferido por TCP, pode-se calcular o total médio de largura de banda gasta. Começando por visualização, verifica-se que 0% do tráfego corresponde a publicidade devido à questão de não haver espaços dedicados a este em plataforma *web*. Para a segunda sessão, tendo em conta que foram visualizadas 38 publicações, incluindo 6 anúncios, 6 "*Instatories*", das quais 3 publicidade, estima-se um consumo de largura de banda de 20%. Por último, para a terceira sessão, 10 publicações do total de publicações observadas na página inicial eram anúncios, e que 3 das 54 "Instatories"eram também publicidades, resultando uma estimativa de 13% para tráfego não solicitado. Considerando apenas as 2 últimas sessões, porque a primeira foi inconclusiva em relação a anúncios, obtém-se uma percentagem de 16% do conteúdo visualizado no Instagram foi publicidade. Equiparando estes resultados ao total de *bytes* transferido, equivale a cerca de 8.7 MB para a segunda captura e 8.5 MB para a terceira. Tendo em conta que a primeira sessão a percentagem foi nula, é retirada da média final de largura de banda. Assim no final, perfaz uma média de 8.6 MB gastos em publicidade no Instagram.

A precisão de resultados obtidos para este serviço é indefinida devido à inexistência de publicidade na versão *web* e pela incapacidade de ultrapassar a questão da cifra. Apesar disto, foram estudados pontos alvo para que uma boa caracterização pudesse ter lugar. Em acréscimo, foi possível concluir que o Instagram é dependente de serviços Facebook, realçando que qualquer entrega de conteúdo não solicitado ocorre da mesma forma para ambos, tornando-se num serviço ganancioso. Ao possuírem os mesmos padrões de formatos, e até mesmo, de colocação de publicidade, os serviços de mediação estão sempre ativos para, assim que possível anúncios sejam entregues tal como acontece com o serviço anteriormente caracterizado.

5.3.4 *Parâmetros de Classificação*

De acordo com o objetivo de identificação e caracterização de tráfego não solicitado em dispositivos móveis, as métricas delineadas na Secção 3.3 encontram-se descritas de forma breve e explícita na Tabela 5.13 com a avaliação realizada ao longo do estudo.

Parâmetro	YouTube	Facebook	Instagram
Volume	7.2MB ou 2% do tráfego total	2.7MB ou 29% do tráfego total	8.7MB ou 16% do tráfego total
Distribuidor	O YouTube na maioria dos casos entrega por CDNs locais ao utilizador todo o conteúdo publicitário, sendo que estas CDNs pertencem ao ISP. O conteúdo parte de entidades interessadas em publicitar.	O Facebook realiza entrega por servidores em CDNs pertencentes ao ISP local ao utilizador. Todo o conteúdo publicitário parte de entidades interessadas em publicitar.	O Instagram entrega conteúdo com origem em CDNs locais ao utilizador, contando com ação de servidores pertencentes ao Facebook. Todos os anúncios no Instagram partem de entidades interessadas em publicitar.
Formatos	Vídeo e Imagem	Vídeo e Imagem (Coleção, Carrossel, Singulares)	Vídeo e Imagem (Coleção, Carrossel, Singulares)
Frequência	Depende de fatores como a popularidade, duração e quantidade de vídeos vistos ao longo da sessão. Outro fator é a forma como a entidade publicadora do vídeo pretende rentabilizar o seu vídeo por forma a obter maior remuneração.	Varia consoante a interação com este serviço. O seu surgimento depende de outros fatores como páginas consultadas, páginas seguidas e popularidade de publicações.	Depende também da interação com esta aplicação. Fatores semelhantes ao Facebook possuem impacto na frequência de anúncios nesta plataforma.
Padrões Observados	Os padrões mais comuns resultam na colocação de anúncios antes, durante e depois de um vídeo. Outro padrão, foi o surgimento de imagens ou vídeo após abertura da aplicação deste.	O padrão mais recorrente ao longo das sessões foi: 1 publicação de página seguida, 1 publicação de anúncio, 4 publicações de páginas seguidas, 1 publicação de anúncio, mantendo-se assim até uma parte da página percorrida. Todos os anúncios identificados com o indicativo "Patrocinado".	O padrão mais vezes encontrado ao longo das capturas foi a ocorrência de anúncios entre transações de "*Instastories*". Todos os anúncios identificados com o indicativo "Patrocinado".
Aplicação Gananciosa	Sim	Sim	Sim
Precisão de Resultados	76%	88%	Não atribuído

Tabela 5.13.: Resumo da avaliação de tráfego não solicitado (anúncios) para os serviços em estudo

5.4 SUMÁRIO

Ao longo deste capítulo foram analisadas várias estatísticas retiradas dos ficheiros `.txt` resultantes da caracterização efetuada pelo classificador desenvolvido. Para cada serviço em estudo, foram mostrados os principais resultados obtidos de cada uma das estratégias adotadas e cada protocolo identificado.

Da consolidação da análise efetuada aos três serviços, foram comparados os parâmetros de classificação que traduzem os aspetos mais relevantes para a caracterização de tráfego não solicitado.

6

CONCLUSÃO E TRABALHO FUTURO

O estudo realizado foca-se na caracterização de tráfego não solicitado para perceber o impacto que aplicações de uso globalizado tais como, YouTube, Facebook e Instagram podem causar num plano de dados móveis. Neste capítulo final é efetuado um sumário das principais conclusões retiradas no decorrer do estudo. Por fim, efetua-se uma reflexão sobre trabalho futuro.

6.1 CONCLUSÕES

A caracterização de tráfego possui um papel muito importante no desenvolvimento e gestão das redes, mostrando ser uma técnica recorrente a vários níveis protocolares que vai progredindo ao longo da evolução das tecnologias.

Para a concretização do estudo, inicialmente foram estabelecidos os objetivos a atingir, bem como dada ênfase à aquisição de conhecimento de suporte. Então, numa primeira etapa, foi feito um levantamento de conceitos relevantes para a temática de caracterização de tráfego, estabelecendo o foco do estudo para tráfego comercial. Dos conceitos, foram vistos aspetos tais como o funcionamento e formatos mais divulgados de anúncios observados em aplicações e serviços recorrentes. Daqui, decorre a explicação de *Ad Serving* e que se trata do conceito mais refletido ao longo do trabalho. Seguidamente, partiu-se para a descoberta de técnicas associadas à caracterização de tráfego no seu âmbito geral. Com o conhecimento adquirido, foi delineada uma estratégia para selecionar aplicações no mercado com maior afluência de utilizadores e dados gerados. Como aplicações ou serviços representativos considerou-se o YouTube, Facebook e Instagram. Foi também efetuado um levantamento de problemas encontrados face aos objetivos traçado. Daqui realça-se a questão da cifra como maior adversidade na caracterização do tipo de tráfego em estudo. Para cada serviço em questão, foram estudadas formas de contornar esta problemática analisando a metodologia de entrega de conteúdo e interação na rede entre cliente e servidores.

Prosseguindo para a etapa seguinte, foi definida uma metodologia de caracterização de tráfego não solicitado focado em publicidade, objeto principal do estudo. Dos problemas

levantados na fase anterior, foram implementadas todas as ideias para a concretização do trabalho, devidamente enquadradas nos objetivos definidos inicialmente. Daqui resultaram múltiplos processos de análise para cada um dos protocolos de transporte associados a cada serviço.

De acordo com os objetivos, foram estabelecidos parâmetros de classificação e caracterização deste tipo de tráfego não solicitado do tipo anúncio. Destes parâmetros, conclui-se que sendo serviços diferentes possuem formas de apresentar publicidade diferentes, refletindo uma não uniformização no que diz respeito a entrega e apresentação de conteúdo publicitário. Verificou-se que, para o YouTube, a base de anúncios é feita sobretudo com recurso a vídeo cujo aparecimento está dependente de múltiplas variáveis e que as imagens nem sempre estão presentes. Para o Facebook, viu-se que a base de anúncios é feita recorrendo maioritariamente a imagens por ser mais apelativo e fácil leitura a fim de não retirar tempo ao utilizador, embora o vídeo possa aparecer mas com menos regularidade. Finalmente, para o Instagram observa-se um misto de regras entre Facebook e algumas características do YouTube, nomeadamente ao recurso a vídeo entre histórias publicadas. Fazendo uma apreciação global aos resultados obtidos, verificou-se que a nível protocolar na camada de transporte, TCP e UDP, não foi possível ter uma ideia exata do conteúdo que era mostrado no ecrã do utilizador e daí ter-se estendido à camada aplicacional para os casos do Facebook e do Instagram.

Em conclusão, todos os objetivos foram alcançados no decorrer do projeto, realçando que este estudo não só ajuda a perceber o processo de entrega de conteúdo não solicitado, como também permite obter mais dados acerca de comportamentos e estatísticas de aplicações e serviços no que toca a entrega de conteúdo no seu geral.

6.2 Resumo das principais contribuições

Como principal contribuição, destaca-se a proposta e implementação de uma metodologia para caracterização de tráfego de não solicitado do tipo anúncio publicitário em dispositivos móveis. Para tal, foi delineado um método de análise que permite reunir métricas associadas a parâmetros de classificação. Estes parâmetros são: Volume, Distribuidor, Formatos, Frequência, Padrões Observados, Aplicação Gananciosa e Precisão de Resultados. De certa forma, o parâmetro Aplicação Gananciosa reúne tudo o foi escrito para os restantes parâmetros em avaliação, tendo-se verificado que todas as aplicações foram conotadas como sendo aplicações gananciosas. À sua medida, cada uma contribui para o esgotamento do plano de dados de forma mais rápida. Para o YouTube, o aparecimento de anúncios está sujeito a fatores como popularidade e duração de um determinado vídeo, concluindo-se que o esgotamento do plano de dados em anúncios será significativo quando estes fatores estiverem em crescimento. Para o Facebook e Instagram, o cenário altera-se, pois os

anúncios apesar de terem uma componente também de popularidade, o esgotamento do plano de dados não é influenciado por este fator mas, sim da interação com a rede social em si. Isto significa que quantas mais vezes se interage com redes sociais, maior será a exposição a tráfego não solicitado em forma de anúncio.

É de refletir que a metodologia de classificação proposta recorre ao processo de comparação de assinaturas dos servidores intervenientes no transporte de dados das aplicações com expressões regulares indicadoras de ocorrências de tráfego não solicitado. Este processo, quando complementado com uma análise comportamental do serviço, demonstrou ter satisfeito os objetivos estabelecidos. Consequentemente, a metodologia delineada é uma mais-valia para a temática de caracterização de tráfego.

Uma contribuição adicional do trabalho foi a caracterização do processo de *Ad Serving*. Apesar do processo de cifragem, é na camada de transporte que melhor se observa o funcionamento de *Ad Serving* tornando-se mais eficaz a revelar como chegam os anúncios até ao utilizador. Sendo exceção o YouTube, onde foi realizada associação entre fluxos UDP e vídeos observados, as restantes aplicações, Facebook e Instagram, necessitaram de uma análise na camada aplicacional de modo a obter um grão mais fino de resultados. Daqui retira-se que, complementando a camada aplicacional e a camada de transporte, é possível retirar múltiplas conclusões no que diz respeito ao tipo, distribuição e localização de servidores de tráfego não solicitado do tipo anúncio.

6.3 TRABALHO FUTURO

A metodologia desenvolvida poderia ter conduzido a um maior detalhe de resultados com o uso de ferramentas de análise de tráfego adicionais. Sendo um trabalho limitado em tempo de realização, este nível de complexidade não foi concretizado.

Após a conclusão do estudo referente a tráfego não solicitado, como trabalho futuro a análise pode ser alargada a outros serviços e aplicações existentes no mercado, no sentido de obter uma maior diversidade de informação acerca do tipo de tráfego em estudo.

Além de uma abertura a outros serviços, sugere-se a adaptação a novas metodologias capazes de contornar o maior problema observado no decorrer do estudo, a cifra. Apesar de ser um segredo de negócio das aplicações para melhorar a segurança perante ataques realizados por terceiros, podem existir comportamentos que conduzam a informações relevantes para este estudo em concreto.

Finalmente, abre-se a sugestão de uma possível expansão abrangente a todo o restante tráfego não solicitado, dando lugar a um estudo que inclui a segurança de serviços e aplicações mais utilizados globalmente.

BIBLIOGRAFIA

[1] A. Dogtiev. Top Mobile Ad Networks 2018 - Business of Apps, 2018. URL http://www.businessofapps.com/guide/top-mobile-ad-networks/.

[2] Android Rank. Android application ranking - All applications, 2018. URL https://www.androidrank.org/app/ranking?hl=en.

[3] Android Rank. androidrank.org - Android market history data and rankings — 2011 - 2018, 2018. URL https://www.androidrank.org/.

[4] R. Bejtlich. Enterprise Network Instrumentation. In *Extrusion Detection: Security Monitoring for Internal Intrusions*, chapter 4, pages 105–136. Addison-Wesley Professional, 1st edition, 2005. ISBN ISBN-10: 0-321-34996-2 ISBN-13: 978-0-321-34996-5.

[5] E. Bocchi, L. Grimaudo, M. Mellia, E. Baralis, S. Saha, S. Miskovic, G. Modelo-Howard, and S. J. Lee. MAGMA network behavior classifier for malware traffic. *Computer Networks*, 109:142–156, nov 2016. ISSN 13891286. doi: 10.1016/j.comnet.2016.03.021. URL https://www.sciencedirect.com/science/article/pii/S1389128616300949.

[6] Cisco. Traffic Classification. In *WAN and Application Optimization Solution Guide*, chapter 5th, pages 1–12. Cisco, 2008. URL https://www.cisco.com/c/en/us/td/docs/nsite/enterprise/wan/wan{_}optimization/wan{_}opt{_}sg/chap05.pdf.

[7] Cisco. Cisco Visual Networking Index: Forecast and Methodology. Technical report, Cisco, 2017. URL https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf.

[8] M. Conti, Q. Q. Li, A. Maragno, and R. Spolaor. The dark side(-channel) of mobile devices: A survey on network traffic analysis. *IEEE Communications Surveys & Tutorials*, page 1–1, 2018. doi: 10.1109/comst.2018.2843533.

[9] Facebook. Creative combinations that work. URL https://www.facebook.com/business/news/insights/creative-combinations-that-work.

[10] Facebook. Using the Graph API, 2019. URL https://developers.facebook.com/docs/graph-api/using-graph-api/.

[11] E. Feitosa, E. Souto, and D. Sadok. Tráfego internet não desejado: Conceitos, caracterização e soluções, 09 2008.

[12] M. Finsterbusch, C. Richter, E. Rocha, J. A. Müller, and K. Hänßgen. A survey of payload-based traffic classification approaches. *IEEE Communications Surveys and Tutorials*, 16(2):1135–1156, 2014. ISSN 1553877X. doi: 10.1109/SURV.2013.100613.00161. URL http://ieeexplore.ieee.org/document/6644335/.

[13] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, dec 2009. URL http://arxiv.org/abs/0912.5410.

[14] Google Play. Packet capture, . URL https://play.google.com/store/apps/details?id=app.greyshirts.sslcapture&hl=en_US.

[15] Google Play. tpacketcapture, . URL https://play.google.com/store/apps/details?id=jp.co.taosoftware.android.packetcapture&hl=en_US.

[16] A. Hernandez, M. Ephraim, and C. Vega. Biographon, 2019. URL https://biographon.com/youtube-stats/.

[17] T. Karagiannis, A. Broido, M. Faloutsos, and k. claffy. Transport Layer Identification of P2P Traffic. In *Internet Measurement Conference (IMC)*, pages 121–134, Oct 2004.

[18] T. Karagiannis, M. Faloutsos, and K. Papagiannaki. BLINC: Multilevel Traffic Classification in the Dark, 2005. URL http://cs.unc.edu/{~}fabian/course{_}papers/BLINC.pdf.

[19] T. Karagiannis, K. Papagiannaki, N. Taft, and M. Faloutsos. Profiling the end host. In S. Uhlig, K. Papagiannaki, and Olivier Bonaventure, editors, *Passive and Active Network Measurement*, pages 186–196, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-71617-4.

[20] N. A. Khater and R. E. Overill. Network traffic classification techniques and challenges. *2015 Tenth International Conference on Digital Information Management (ICDIM)*, 2015. doi: 10.1109/icdim.2015.7381869.

[21] B. Kneen. How ad serving works – mobile vs. web environments - ad ops insider, 2013. URL http://www.adopsinsider.com/ad-serving/how-ad-serving-works-mobile-vs-web-environments/.

[22] Feng Li, Jae Chung, and Mark Claypool. Silhouette: Identifying youtube video flows from encrypted traffic. pages 19–24, 06 2018. doi: 10.1145/3210445.3210448.

[23] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of internet background radiation. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, pages 27–40, New York, NY, USA, 2004. ACM. ISBN 1-58113-821-0. doi: 10.1145/1028788.1028794. URL http://doi.acm.org/10.1145/1028788.1028794.

[24] A. Peris, M. Chinea-Rios, and F. Casacuberta. Neural Networks Classifier for Data Selection in Statistical Machine Translation. *The Prague Bulletin of Mathematical Linguistics*, dec 2016. URL http://arxiv.org/abs/1612.05555.

[25] A. Razaghpanah, A. A. Niaki, N. Vallina-Rodriguez, S. Sundaresan, J. Amann, and P. Gill. Studying TLS Usage in Android Apps. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies - CoNEXT '17*, pages 350–362, New York, New York, USA, 2017. ACM Press. ISBN 9781450354226. doi: 10.1145/3143361.3143400. URL http://dl.acm.org/citation.cfm?doid=3143361.3143400.

[26] D. Rossi. Tcp statistic and analysis tool. URL http://tstat.polito.it/.

[27] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger. *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience*. Springer, 2013. ISBN 978-3-642-36783-0. doi: 10.1007/978-3-642-36784-7. URL http://link.springer.com/chapter/10.1007{%}2F978-3-642-36784-7{_}10.

[28] M. Shafiq, Xiangzhan Y., A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia. Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms. *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 2451–2455, 2016. doi: 10.1109/CompComm.2016.7925139. URL http://ieeexplore.ieee.org/document/7925139/.

[29] SLAC. Network Monitoring Tools *, 2018. URL https://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html{#}contents.

[30] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft. Breaking for commercials: Characterizing mobile advertising. *Proceedings of the 2012 ACM conference on Internet measurement conference - IMC 12*, 2012. doi: 10.1145/2398776.2398812.

[31] K. Xu, Z. Zhang, and S. Bhattacharyya. Reducing unwanted traffic in a backbone network. In *SRUTI*, 2005.

[32] Youtube. Youtube advertising formats - youtube help. URL https://support.google.com/youtube/answer/2467968?hl=en.

[33] M. Zawadzinski. What is an Ad Server and How Does It Work? - Clearcode Blog, 2018. URL https://clearcode.cc/blog/what-is-an-ad-server/.

[34] L. Zhang, E. B. Davies, and L. Andersson. Report from the IAB workshop on Unwanted Traffic March 9-10, 2006. RFC 4948, August 2007. URL https://rfc-editor.org/rfc/rfc4948.txt.

A

DETALHES CAPTURAS

	YouTube1.pcap	**YouTube2.pcap**	**YouTube3.pcap**
Duração	9min 13s	37min 26s	34min 8s
Horário de Captura	20 de setembro de 2019	9 de agosto de 2019	27 de agosto de 2019
Anúncios Vistos	2	2	4
Vídeos Visualizados	1	3	1
Duração Individual dos Vídeos Visualizados	7min 20s (visto na sua totalidade cuja prolongação de tempo foi provocada pela publicidade que apareceu)	Pré-visualização de vídeo sugerido durante 30s; Vídeo 1 - 21min 48s (avançando algumas partes do vídeo); Vídeo 2 - 11min 12s (visto até ao fim); Vídeo 3 - 12min 50s (visto até ao fim	27min 11s (vídeo visto na totalidade prolongado para o tempo de captura referido devido à publicidade e anotações)
Total de MBytes transferidos	292.42 MB	1011.67 MB	400.01 MB
Total de Pacotes transferidos	249357	824023	342857
Vídeos Em Questão	Vídeo 1- https://tinyurl.com/y3sm77ek; Anúncio 1- https://tinyurl.com/y53gnp3q; Anúncio 2- https://tinyurl.com/y4oo9ogx	Preview- https://tinyurl.com/y2583wpq; Vídeo 1- https://tinyurl.com/y68hr5sa; Anúncio 1- https://tinyurl.com/yy7aaq5a; Vídeo 2- https://tinyurl.com/y4m4bm5y; Anúncio 2- Não foi possível encontrar url; Vídeo 3- https://tinyurl.com/y4vst9h3;	Vídeo 1-https://tinyurl.com/yxspxn9g; Anúncio 1 e 2- https://tinyurl.com/y4u3fu4f(encurtado para 45s); Anúncio 3- https://tinyurl.com/yy7aaq5a; Anúncio 4- https://tinyurl.com/y2sro3kt

Tabela A.1.: Detalhes relativos às capturas do YouTube

	Facebook1.pcap	**Facebook2.pcap**	**Facebook3.pcap**
Duração	2min 20s	3min 32s	4min 18s
Horário de Captura	20 de Agosto de 2019	21 de Agosto de 2019	27 de Agosto de 2019
Publicações do Tipo ″Patrocinado″	7	8	13
Publicações Visualizadas	40	45	50
Total de MBytes transferidos	13.37 MB	8.62 MB	11.01 MB
Total de Pacotes	14754	10286	13455

Tabela A.2.: Detalhes relativos às capturas do Facebook

	Instagram1.pcap	**Instagram2.pcap**	**Instagram3.pcap**
Duração	3min 2s		
Horário de Captura	13 de setembro de 2019	13 de setembro de 2019	17 de setembro de 2019
Publicações do Tipo ″Patrocinado″	0	6	10
Publicações Visualizadas	39	38	47
Instastories do Tipo ″Patrocinado″	0	3	3
Instastories Visualizadas	9	6	54
Total de MBytes transferidos	6.58 MB	45.02 MB	70.5 MB
Total de Pacotes	7860	46200	72193

Tabela A.3.: Detalhes relativos às capturas do Instagram