Síntese de Vídeo para Evasão de Censura na Internet

Diogo Barradas, Nuno Santos, e Luís Rodrigues
{diogo.barradas, nuno.m.santos, ler}@tecnico.ulisboa.pt

INESC-ID / Instituto Superior Técnico, Universidade de Lisboa

Resumo Este projecto visa o estudo da possibilidade da utilização do canal vídeo presente em aplicações de vídeo-conferência, tais como o Skype, como transporte para um canal encoberto que pode ser utilizado para aceder a informação arbitrária na Internet. Para este fim, propomos e avaliamos diferentes alternativas para codificação de informação no canal de vídeo, tendo como objectivo a maximização da taxa de transferência e tendo em atenção a preservação das características de tráfego da transmissãode do vídeo original. O nosso protótipo oferece uma interface ao nível da camada de ligação, suportando qualquer protocolo transmitido sobre TCP/IP. Os resultados da nossa avaliação mostram que é possível atingir uma taxa de transferência de 0.4 KB/s sem um impacto significativo nas características de tráfego do vídeo original, permitindo a execução de aplicações comuns tais como FTP, Telnet ou Wget.

1 Introdução

Nos dias de hoje a maioria das comunicações efectuadas através da Internet podem ser controladas por governos e/ou por algumas empresas. Este facto permite que regimes repressivos controlem o acesso à Internet, impedindo os cidadãos de exercer os seus direitos civis, sendo-lhes retirada a possibilidade de aceder a informação, comunicar ou expressar opiniões de forma livre [2]. Mas nem os regimes mais opressivos podem bloquear todos os canais de comunicação electrónica com o mundo exterior. A experiência mostra que mesmo os países que restringem o acesso à informação mantêm operacionais alguns dos serviços mais usados pela sociedade, tais como o Skype.

Frequentemente, para impedir o acesso a certas fontes de informação, o censor obriga os ISPs a bloquearem o acesso directo a essas fontes. Uma estratégia típica para evadir este tipo de restrições consiste no acesso à informação através de um *proxy*. No entanto, esta estratégia só é exequível enquanto os endereços dos proxies não forem tornados públicos e as conexões aos mesmos não possam ser identificadas como suspeitas. Em particular, se não for feito um esforço no sentido de obfuscar o tráfego em direcção a um proxy, este pode exibir padrões que o tornam facilmente reconhecível [1]. Assim, para que a obfuscação seja bem sucedida, o tráfego resultante deve mimetizar protocolos existentes. No entanto, a imitação completa de um protocolo pode ser extraordinariamente complexa [8].

Uma abordagem mais recente consiste na facilitação do acesso a informação legítima através de um canal encoberto, ao *encapsular* os dados de forma furtiva através de protocolos autorizados pelo censor. Alguns exemplos que usam esta estratégia incluem o FreeWave [9], que codifica dados em sinais acústicos enviados através de conexões VoIP, e o Facet [10], que possibilita a transmissão de vídeos censurados de forma encoberta, através de uma vídeo-chamada.

Estes resultados são promissores mas exibem algumas limitações. O funcionamento correcto do FreeWave pode ser comprometido por ataques que geram perturbações controladas na rede. Para além disso, pode também ser detectado através da análise do tráfego de rede gerado. Por sua vez, o Facet emprega uma técnica denominada *video morphing*, resiliente a este tipo de ataques. No entanto, o Facet apenas pode ser utilizado para transferir vídeos, limitando a sua aplicação a outros tipos de comunicações críticas face à presença de um censor.

Neste projecto apresentamos um sistema que permite a codificação de dados arbitrários num canal vídeo, estendendo a técnica de *video morphing*. O desenho do sistema tem em vista o estabelecimento de uma camada de ligação entre os dois pontos da chamada, permitindo a transferência mútua de pacotes da camada de rede. Tal permite a utilização de vários protocolos do nível de aplicação, tais como Telnet ou HTTP. Propomos e avaliamos diversas alternativas para codificação de informação no canal de vídeo de uma vídeo chamada, de forma a maximizar a taxa de transferência sem alterar as características da ligação original. Os resultados da avaliação experimental atestam que o sistema permite a execução de aplicações comuns, exibindo um custo temporal adicional compreendido entre 10 a 20 vezes no estabelecimento de sessões interactivas, sem um impacto significativo nas características da vídeo-chamada original.

2 Trabalho Relacionado

Várias soluções para o problema da censura na Internet têm sido propostas ao longo dos últimos anos. Uma estratégia comum para a evasão da censura consiste em recorrer ao reencaminhamento de tráfego através de proxies, aliado a técnicas de esteganografia digital. No entanto, uma vez detectados, os proxies podem ser bloqueados de forma similar às fontes de informação originais.

Uma outra classe de sistemas que auxiliam na evasão de censura encontra-se focada em obfuscar tráfego de forma a que o protocolo subjacente não possa ser identificado. Esta técnica é denominada *traffic morphing* [15]. Por exemplo, o SkypeMorph [11] imita as propriedades estatísticas de uma vídeo-chamada através do Skype. No entanto, a imitação de todos os comportamentos de um protocolo, incluindo a resposta a excepções, revela-se difícil, o que torna a imitação de protocolos vulnerável a vários ataques activos. O Marionette [4] tenta colmatar esta limitação ao possibilitar o controlo de vários aspectos da mimetização de um protocolo. No entanto, a imitação de protocolos proprietários pode requerer esforços de engenharia reversa de forma sistemática.

Alguns sistemas de evasão de censura tomam partido do encapsulamento *estagiado* de informação, onde um servidor oblívio reencaminha a comunicação en-

tre o cliente e servidor do sistema. O CloudTransport [3] utiliza serviços públicos de armazenamento na nuvem para reencaminhamento das mensagens encobertas. O *meek* [6] toma partido da técnica de *domain fronting* para encapsular tráfego através de ligações HTTPS dirigidas a destinos autorizados, enquanto que estabelece uma ligação encoberta a um destino bloqueado pelo censor. O sistema Castle [7] oferece outra alternativa, encapsulando a informação nos comandos e estado mantido por jogos de estratégia em tempo real. No entanto, vários sistemas desta categoria são vulneráveis a diferentes ataques, tais como ataques de negação de serviço ou análise de tráfego.

Outra abordagem ao encapsulamento de informação tira partido dos protocolos de *streaming* multimédia. O FreeWave [9] usa ligações VoIP para encapsular tráfego da Internet, permitindo a navegação através de conteúdos web bloqueados. No entanto, este sistema é vulnerável a ataques passivos e activos por parte do censor, tendo este a possibilidade de detectar a ligação através de análise de tráfego ou impedir a negociação de parâmetros necessários à modulação áudio.

O Facet [10] toma partido de ligações de vídeo-conferência, como o Skype, sobrepondo vídeos censurados sobre vídeo-chamadas regulares. Esta técnica, denominada *video morphing*, assegura que os pacotes de rede gerados pelo sistema de vídeo-conferência não reflectem directamente as características do vídeo censurado, aproximando-se daquelas que dizem respeito a vídeo-chamadas regulares. A abordagem oferece uma resistência inerente a ataques activos, sendo que as perturbações na rede causadas pelo censor terão exactamente os mesmos efeitos quer na transmissão encoberta, quer numa transmissão regular. No entanto, devido ao seu desenho, este sistema apenas permite a transmissão de vídeo, limitando a sua aplicabilidade a outros tipos de comunicação.

3 Modelo de Ameaça

O adversário tem como objectivo a detecção e bloqueio de fluxos de comunicação Skype que transportam mensagens encobertas. Assumimos um adversário omnisciente, capaz de observar, armazenar, analisar e interferir com todos os fluxos de rede entre dois pontos da comunicação. O adversário tem também a capacidade de executar *deep packet inspection*. No entanto, assumimos que o adversário possui uma capacidade computacional limitada: se o conteúdo dos pacotes se encontrar cifrado, o adversário não terá a possibilidade de quebrar as primitivas criptográficas usadas para cifrar o mesmo. Assumimos que o adversário não tem controlo sobre o software instalado nos computadores dos utilizadores, pelo que os pontos que executam os clientes Skype são considerados como sendo de confiança. Consideramos que o adversário não tem interesse em interferir com a utilização normal do Skype, sendo penalizado se quebrar ligações de forma arbitrária. O adversário apenas quebrará uma ligação caso possua evidências fortes de que a chamada é utilizada como veículo para um canal de informação encoberto. Por último, admitimos que o fornecedor do serviço de vídeo-conferência não entrará em conluio com o adversário e que não irá permitir inspecção do conteúdo vídeo projectado nos pontos terminais.

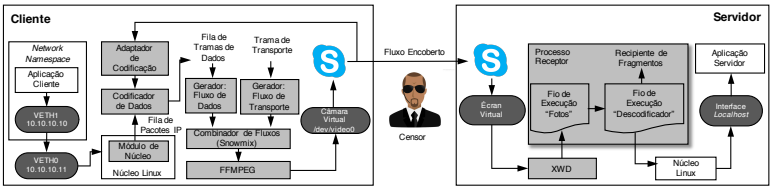


Figura 1: Componentes do protótipo (caixas sombreadas).

4 Desenho do Sistema

O objectivo do sistema é permitir encapsular um canal (bi-direccional) numa vídeo-chamada, de forma a que esta não possa ser identificada como suspeita pelo adversário. O funcionamento do sistema encontra-se ilustrado na Figura 1. O emissor recebe os dados da rede e codifica-os num fluxo de vídeo que é lido pelo Skype. O Skype transmite este vídeo para a instância remota do Skype e o fluxo recebido é capturado periodicamente. Um descodificador extrai os dados encobertos do fluxo de vídeo e entrega-os à aplicação. Para tornar o sistema tão geral quanto possível, a arquitectura expõe um protocolo de camada de ligação às camadas superiores, possibilitando a aceitação, codificação, descodificação e entrega remota de um pacote IP.

4.1 Codificação e Descodificação de Dados

Um *fluxo* de vídeo consiste numa sequência de *tramas*, constituídas por um conjunto de pixeis, por sua vez definidos por componentes RGB. De acordo com a especificação do formato XWD, utilizado para armazenar imagens capturadas do écran, os componentes RGB de um pixel ocupam 24 dígitos.

Em teoria, poderiam ser codificados até um máximo de 7,372,800 dígitos, assumindo que lidamos com tramas de tamanho 640x480 (resolução VGA). Na prática, existem várias razões que impedem tal esquema de codificação. Em primeiro lugar, a compressão vídeo efectuada pelo Skype pode alterar as cores de cada pixel e omitir diferenças entre pixeis adjacentes. Em segundo lugar, a assinatura do tráfego gerado por uma chamada Skype que transporta dados encobertos deverá ser indistinguível de uma chamada Skype "normal", de forma a preservar a *não-observabilidade*. Para lidar com estes desafios, propomos um esquema de codificação baseado em três ideias principais:

Combinar vídeo sintético numa chamada Skype "normal": As tramas de vídeo transmitidas (tramas *e*ncobertas) são criadas através da combinação de dois componentes representados na Figura 2: (a) tramas *de transporte* e (b) tramas *de dados*. As tramas de transporte são obtidas a partir de chamadas Skype pré-gravadas. As tramas de dados consistem em tramas de vídeo sintetizadas que codificam os dados da aplicação a ser transmitidos para o receptor. As duas tramas são combinadas em (c) tramas encobertas e entregues ao Skype. Cada

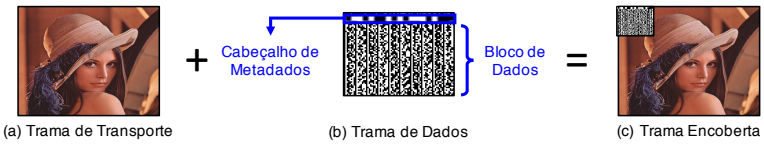


Figura 2: Combinação de *tramas*.

trama encoberta é auto-contida, possuindo um campo de metadados com os parâmetros a utilizar na descodificação do conteúdo útil dos dados.

Suportar uma codificação ajustável: Cada trama de dados codifica N dígitos da mensagem encoberta num *bloco de dados*, que consiste por sua vez numa imagem constituída por uma grelha de *células*. Cada célula consiste numa área formada por pixeis da mesma cor, utilizada para codificar b_c dígitos do bloco de dados. O número total de dígitos (N) que poderá ser codificado por trama é então dado por: $N = b_c \times n_c$, onde n_c é o número de células por trama. Como resultado, a taxa de transferência T é dada por $N \times r_p$, onde r_p é o número de tramas de dados enviadas por unidade de tempo. A codificação é então definida pelos seguintes parâmetros: tamanho da trama de dados em pixeis (s_p), tamanho das células em pixeis (s_c), codificação de dígitos em cores (b_c), e o número de tramas transferidas por unidade de tempo(r_p), dado pelo tuplo $S : \langle s_p, s_c, b_c, r_p \rangle$, por exemplo $\langle 160 \times 120, 4 \times 4, 1, 3 \rangle$. Os parâmetros de codificação utilizados pelo emissor são colocados numa banda fixa no topo da trama de dados.

A redução do número de cores para representar dígitos torna o sistema mais resiliente às alterações de cor por pixel, ao passo que o aumento do tamanho das células permite tolerar a perda de informação entre pixeis adjacentes, resultante da compressão vídeo executada pelo Skype. É ainda possível controlar a quantidade de dados combinados no vídeo de transporte, o que irá determinar quão próximo o vídeo encoberto será relativamente a uma chamada Skype "normal".

Adaptação às condições da rede: Uma codificação ajustável permite a calibração dos parâmetros do sistema, na eventualidade das condições da rede sofrerem alterações, tornando a ligação observável devido ao uso de um dado conjunto de parâmetros. O cliente pode iniciar o processo de calibração periodicamente para determinar novos parâmetros. Cada período composto por uma fase de calibração e uma fase de transmissão de dados é denominado *época*.

4.2 Preservando a Assinatura de Tráfego Skype "Normal"

Designamos fluxos Skype "normais" como *fluxos regulares*. Um fluxo Skype é regular se resulta de uma chamada vídeo legítima entre dois utilizadores, sem a transmissão de qualquer mensagem encoberta. Consideramos que um fluxo é *irregular* se a diferença para um fluxo regular excede um limiar Δ, onde Δ é obtido através de uma *função de similaridade* σ. Considerando s_R um fluxo regular, f uma *função característica* do fluxo (ex., distribuição do comprimento

dos pacotes), e s_C um fluxo arbitrário (que poderá conter um canal encoberto), dizemos que s_C é indistinguível de s_R se:

$$\sigma(f(s_C[P]), f(s_R)) \leq \Delta$$

Assim, devem ser escolhidos os parâmetros de codificação P para s_C tal que o fluxo encoberto resultante obedeça à condição acima. Para alcançar este objectivo, seguimos quatro passos:

Encontrar uma função característica precisa (f): Através de avaliação experimental, consideramos que a *distribuição de frequência do comprimento dos pacotes* (f_l) caracteriza com precisão um dado padrão de fluxos Skype. Uma função alternativa baseada na distribuição de bi-gramas do comprimento dos pacotes permitiu a diferenciação de vídeos provenientes do YouTube de vídeos de chamadas regulares, quando transmitidos através do Skype [10]. Neste contexto, esta função produz resultados similares aos de f_l.

Encontrar uma função de similaridade (σ): Dado que f_l oferece como resultado uma distribuição de frequência, foram procuradas métricas que permitem o cálculo da similaridade entre duas distribuições de probabilidade. Trabalhos anteriores adoptaram o teste Kolmogorov–Smirnov [11]. No entanto, verificou-se a obtenção de melhores resultados com a Earth Mover's Distance (EMD) [12]. Intuitivamente, $\text{EMD}(f_l(s_R), f_l(s_C))$ representa a quantidade de trabalho necessário para transformar $f_l(s_C)$ em $f_l(s_R)$.

Calcular o limiar de similaridade máximo (Δ): O limiar de similaridade Δ permite fixar um limite na diferença máxima espectável entre chamadas regulares Skype. Este valor pode ser determinado através da criação de um conjunto de treino com N vídeos de chamadas legítimas, transmitindo cada vídeo M vezes e armazenando a distribuição do comprimento dos pacotes do fluxo de teste resultante s_{ij}, onde $0 \leq i < N$ e $0 \leq j < M$. De seguida, é calculada a distribuição de similaridade entre cada fluxo de teste e o fluxo de referência (um fluxo regular fixado), sendo o valor máximo obtido Δ. Este valor é determinado por:

$$\Delta = max(\text{EMD}(f_l(s_{ij}), f_l(s_R)))$$

Obter um selector de codificação válido (P): O passo final consiste na determinação dos conjuntos válidos de instâncias de parâmetros (P) para o esquema de codificação. Chamamos *selector de codificação* a uma instância específica de P. Para ser válido, um selector terá que produzir fluxos não-observáveis. Se vários selectores forem válidos, será seleccionado aquele que permite obter a maior taxa de transferência. Os selectores que satisfazem tal condição podem ser encontrados ao explorar o espaço de P, gerando um fluxo de treino $s_C[P]$ e verificar que s_C é indistinguível de s_R. Mais precisamente:

$$\text{EMD}(f_l(s_C[P]), f_l(s)) = \delta,\ P \text{ é } \textit{válido} \text{ se } \delta < \Delta$$

5 Concretização

Foi desenvolvido um protótipo do sistema para Linux, como representado na Figura 1. O protótipo é constituído por vários componentes que implementam a "pipeline" do cliente e do servidor. Alguns componentes foram codificados de raiz em C++ e Python; os demais são baseados em ferramentas existentes, endereçadas nos parágrafos seguintes. Durante o desenvolvimento do protótipo foi necessário superar vários desafios técnicos, de forma a concretizar: a interface com a rede, o processamento de vídeo, e a interface com o Skype.

Interface com a Rede: O protótipo toma partido dos *network namespaces* do Linux, bem como do mecanismo de filtragem de pacotes *netfilter*, para construir a camada de ligação no lado do cliente. Os pacotes IP enviados por uma dada aplicação (executada dentro de um *network namespace* designado) são capturados por um módulo do núcleo, sendo entregues a uma aplicação no espaço de utilizador que os codifica e envia através do Skype. No lado do servidor, os pacotes IP são descodificados e reencaminhados à interface "localhost", sendo entregues de forma transparente à aplicação que actua como servidor.

Processamento de Vídeo: Para realizar a síntese de vídeo no lado do cliente de forma eficiente, o codificador de dados analisa cada pacote IP, gera a trama de dados correspondente, e reencaminha-a para um processo ao nível do utilizador, que transfere as tramas de dados para o Snowmix [13]. O Snowmix permite a combinação de fluxos de vídeo produzidos em directo, sendo utilizado de forma a combinar as tramas de dados com as tramas de transporte. O vídeo resultante é então enviado via Skype ao receptor.

Interface com o Skype: Para o cliente interagir com o Skype, o vídeo resultante do Snowmix é codificado através de uma ferramenta de processamento vídeo, o FFMPEG [5], e entregue a um dispositivo de câmara virtual. O Skype é então configurado para ler as tramas entregues ao dispositivo. No receptor, as imagens mostradas pelo cliente Skype são capturadas por um fio de execução que corre periodicamente o utilitário XWD para obter uma captura da janela da chamada Skype, projectada num écran virtual.

Formato das Mensagens Foi criado um protocolo para suportar a projecção de mensagens entre a camada de alto nível IP e a camada de tramas baixo nível suportada através dos blocos de dados. Por falta de espaço, omitimos aqui uma descrição do formato das mensagens usadas pelo protocolo.

Correcção de Erros Devido à compressão vídeo efectuada pelo Skype, os blocos de dados recuperados podem incluir erros ao nível da sequência de dígitos. Por este motivo, os blocos de dados foram definidos de forma a suportar códigos de correcção de erros configuráveis. O esquema Reed-Solomon [14] foi adoptado no protótipo, após uma avaliação empírica. Foi utilizado o código denotado por $(n, k) = (255, 223)$, onde n corresponde a 255 octetos de um símbolo de dados, dos quais $k = 223$ octetos consistem em dados da aplicação e os restantes 32 octetos codificam dígitos de paridade. Este código pode corrigir até 16 octetos por cada bloco de símbolos.

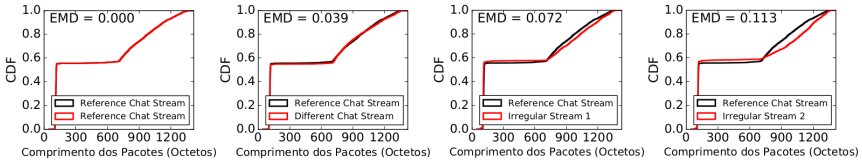


Figura 3: Diferenças entre CDFs.

6 Avaliação

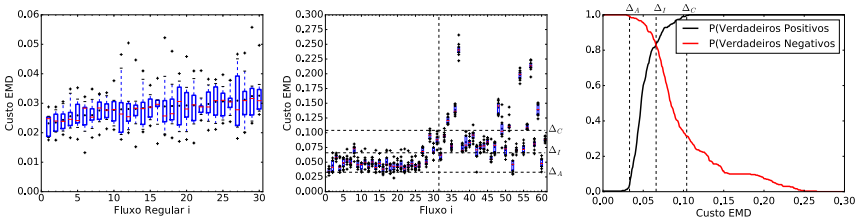
6.1 Configuração das Experiências

A avaliação do sistema foi executada numa máquina quad-core Intel Xeon CPU E3-1220 v3 3.10GHz com 32GB de RAM. Foram configuradas duas máquinas virtuais (VMs) 32bit Ubuntu 14.04.3 LTS com 2GB RAM e 4 CPUs virtuais. Cada VM executa uma instância do Skype, actuando respectivamente como emissor e receptor da vídeo-chamada. Foram usados 30 vídeos representativos de vídeo-chamadas comuns como conjunto de treino para fluxos regulares. O conjunto de treino para fluxos irregulares é composto por 30 vídeos do YouTube, onde são comuns as mudanças de cena e artefactos introduzidos por ferramentas de edição de vídeo. As amostras obtidas têm como base capturas de tráfego com a duração de 30 segundos, em condições de rede não limitadas artificialmente.

6.2 Caracterização de Fluxos do Skype

Foi estudada em primeiro lugar a possibilidade das chamadas Skype exibirem padrões mensuráveis que permitam a diferenciação de chamadas regulares de irregulares. Os dados da Figura 3 indicam que tais padrões existem. Encontram-se representadas as distribuições cumulativas de probabilidade (CDF) do comprimento dos pacotes para quatro fluxos de teste: (a) o fluxo de uma vídeo-chamada tomada como fluxo de referência; (b) o fluxo de uma vídeo-chamada regular realizada por outro utilizador; e dois fluxos irregulares, correspondendo a (c) um jogo de futebol e (d) um concerto. É possível verificar que a distância EMD face ao fluxo de referência aumenta progressivamente.

De forma a verificar se estes padrões de tráfego podem ser utilizados de forma confiável para a caracterização de fluxos regulares, analisamos a existência de diferenças significativas nas distribuições do comprimentos dos pacotes ao transmitir a mesma vídeo-chamada via Skype, múltiplas vezes. Repetimos a transmissão de cada vídeo-chamada regular do nosso conjunto de treino, dez vezes, calculando o custo EMD para cada um dos fluxos resultantes, tomando como referência a distribuição média de todas as dez ligações. É possível observar na Figura 4a os indicadores estatísticos de maior relevância para os valores EMD resultantes para cada vídeo. Por um lado, a distribuição dos pacotes do mesmo vídeo tende a ser bastante similar, sendo que a maior diferença, observada entre os quartis 25º e 75º, é apenas de 0.02. Para além disso, o valor EMD médio



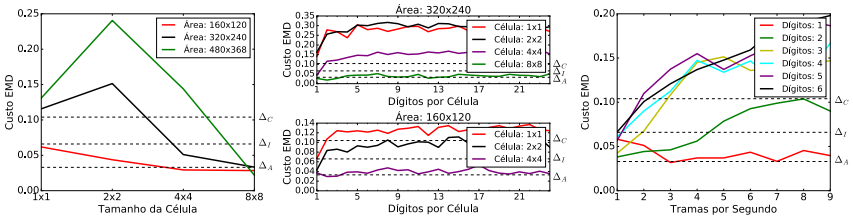
(a) (b) (c)

Figura 4: Custo EMD como classificador de fluxos Skype.

tende a ser bastante similar entre os vários vídeos, variando entre 0.025 e 0.031. Concluímos que, para as mesmas condições de rede, os fluxos regulares mostram um grau de similaridade elevado.

De seguida, avaliamos a capacidade do censor diferenciar fluxos regulares de irregulares. Para esse fim, tomamos um fluxo regular como fluxo de referência, em relação ao qual será calculado o custo EMD dos fluxos restantes. A Figura 4b ilustra os resultados obtidos, representando à esquerda a similaridade para fluxos regulares, e à direita a similaridade para fluxos irregulares. Os fluxos regulares apresentam de forma constante um custo EMD baixo (abaixo de 0.1), ao passo que os fluxos irregulares produzem um padrão disperso, variando o custo EMD de 0.025 a 0.25, ou seja, uma ordem de magnitude.

A questão prende-se então com a possibilidade de definir um limiar de EMD (Δ) que possa ser utilizado para realizar a classificação dos fluxos, tal que um fluxo s é considerado regular caso $\text{EMD}(s_R, s) < \Delta$ ou irregular caso contrário. Para avaliar a eficácia deste classificador, ilustramos na Figura 4c a probabilidade de *verdadeiros positivos* e *verdadeiros negativos* ao variar o valor de Δ (no eixo dos x). É possível estabelecer várias políticas consoante a escolha de Δ. Se o censor desejar bloquear todos os fluxos irregulares (ou seja, uma *política de classificação agressiva*), Δ deve ser fixado em Δ_A. A desvantagem desta política é que levaria ao bloqueio de cerca de 95% dos fluxos regulares, causando um ataque de negação de serviço massivo a utilizadores legítimos do Skype. Por outro lado, caso o censor deseje evitar o bloqueio de qualquer vídeo-chamada Skype legítima (ou seja, uma *política de classificação conservadora*), Δ deve ser fixado em Δ_C. A desvantagem desta política é a perda de especificidade, sendo que aproximadamente 70% dos fluxos irregulares seriam classificadas como regulares (falsos negativos). Uma política intermédia tomaria como Δ o ponto onde a probabilidade de *verdadeiros positivos* iguala a probabilidade de *verdadeiros negativos*. Para o nosso caso de estudo, este ponto corresponde ao limiar de EMD 0.066 (Δ_I), resultando numa precisão de 83% na classificação de um fluxo, permitindo a identificação de fluxos regulares com elevada probabilidade.



(a) (b) (c)

Figura 5: Impacto da escolha de parâmetros na observabilidade dos fluxos.

6.3 Não-Observabilidade dos Canais Encobertos

Para produzir fluxos Skype encobertos, o sistema deve ser configurado de forma a que o custo EMD dos mesmos permaneça abaixo do Δ escolhido para o classificador. Devem então ser estudadas quais as configurações de parâmetros que permitem a produção de fluxos não-observáveis.

Iniciamos a nossa análise pelos efeitos combinados do tamanho da área utilizada para transportar dados encobertos e do tamanho da célula, fixando o número de dígitos em 1 dígito/célula e a taxa de transmissão em 1 trama por segundo. A Figura 5a revela o custo EMD para várias configurações, variando o tamanho da célula e o tamanho da área de dados. As áreas foram escolhidas de forma a cobrir cerca de 1/16, 1/4 e 1/2 do tamanho total da *trama*. O gráfico encontra-se anotado com os valores Δ para as três políticas discutidas na secção anterior: agressiva (Δ_A), conservadora (Δ_C), e intermédia (Δ_I). É possível observar que, para uma política intermédia, existem cinco configurações que produzem fluxos não-observáveis: áreas de tamanho 160x120 ou 320x240 e tamanhos de célula 4x4 ou 8x8; área de tamanho 480x368 e tamanho de célula 8x8. Verifica-se que, à medida que o tamanho da célula aumenta, o custo EMD tende a diminuir, uma vez que áreas maiores da trama serão preenchidas com a mesma cor, aumentado a eficiência do algoritmo de compressão vídeo.

Para as configurações válidas encontradas, estudamos como varia a não-observabilidade em função do número de dígitos codificados por célula. A configuração com área 480x368 deu constantemente origem a fluxos identificados como irregulares pelo classificador, ao codificar mais que 1 dígito por célula. A Figura 5b ilustra os resultados para as primeiras quatro configurações, ao cobrir o domínio do número de dígitos possíveis, entre 1 a 24 dígitos. De forma geral, a não-observabilidade tende a degradar-se à medida que o número de dígitos aumenta. No entanto, existem duas configurações que se colocam de forma consistente abaixo do valor Δ para a política de bloqueio intermédia (Δ_I), nomeadamente (160x120, 4x4) e (320x240,8x8). Estes resultados evidenciam que ambas as configurações são candidatas a gerar fluxos não-observáveis.

Por último, é estudado o efeito da taxa de transmissão de tramas na não-observabilidade. A Figura 5c ilustra a variação do custo EMD à medida que a taxa de transmissão aumenta. O tamanho da área de dados encontra-se fixado

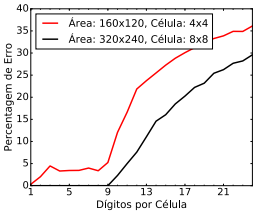


Figura 6: Taxa de erros.

Caso de Uso	Com Sistema	Sem Sistema	Custo
A. Wget	1m 9s 830ms	7ms	9,975.7×
B. FTP	2m 45s	8s 528ms	19×
C. SMTP	2m 42s	37s 913ms	4.3×
D. SSH	1m 51s 493ms	6s 485ms	17.2×
E. Telnet	1m 17s 471ms	7s 670ms	10.1×
F. Netcat Chat	1s 147ms	11ms	133×
G. SSH Tunnel	3m 46s 55ms	21s 940ms	10.3×

Tabela 1: Tempo de execução dos casos de uso.

em 320x240 e o tamanho da célula em 8x8. Os resultados demonstram que o aumento da taxa de transmissão resulta rapidamente num custo EMD acima do valor Δ definido. Encontra-se uma excepção notável no esquema de codificação de apenas 1 dígito por célula, que permanece abaixo de Δ para todas as taxas testadas. Esquemas de codificação com números de dígitos mais elevados apenas toleram a taxa mínima testada (1 trama por segundo).

É ainda possível verificar que algoritmo de compressão vídeo introduz alterações nos dígitos menos significativos que compõem a cor de cada pixel, introduzindo erros na descodificação. Este efeito pode ser observado na Figura 6, que ilustra o aumento da taxa de erro em função do número de dígitos codificados por célula. Decidimos ser mais conservadores com respeito à codificação de dígitos por célula (menos de 9 dígitos/célula) uma vez que rajadas de erros esporádicas no processo de descodificação podem resultar na perda de uma trama de dados, afectando significativamente a taxa de transferência. A taxa de transferência máxima que foi possível atingir com este esquema foi de 0.32 e 0.39 KB/s, com e sem códigos de correcção de erro, respectivamente. Tendo em conta as restrições em termos de não-observabilidade e erros de descodificação, foi identificado um selector de codificação candidato, consistindo no tuplo (320x240, 8x8, 6, 1).

6.4 Casos de Uso

O nosso sistema é capaz de suster a execução de aplicações que toleram alta latência/baixa taxa de transferência. Foram testados 6 casos de uso: transferência de uma página web com 4KB hospedada no receptor (Caso A), transferência de um ficheiro de 4KB disponibilizado por um servidor FTP residente no receptor (Caso B), enviar um pequeno email (duas frases escritas de forma interactiva) através de um servidor SMTP residente no receptor (Caso C), estabelecer uma sessão SSH com o receptor (Caso D), estabelecer uma sessão Telnet com o receptor (Caso E), enviar uma mensagem para o receptor através de um servidor netcat, simulando um "webchat" (Caso F), estabelecer uma sessão SSH com um servidor remoto, após estabelecer uma ligação SSH com o receptor (Caso G). A Tabela 1 oferece um sumário do tempo de execução para cada caso de uso, quando usado com e sem o nosso sistema, isto é, ao utilizar canais de comunicação abertos entre o cliente e servidor. Para além da latência experimentada pelos utilizadores, todos os casos de uso se encontram completamente funcionais.

7 Conclusão

Neste projecto apresentamos um sistema de evasão de censura na Internet que toma partido do canal vídeo de aplicações de vídeo-conferência populares para encapsular dados encobertos. O sistema oferece uma interface de camada de ligação, suportando qualquer aplicação que tolere alta latência / baixa taxa de transferência e que execute sobre TCP/IP, oferecendo aos utilizadores uma vasta gama de possibilidades para transferir informação de forma não-observável. Foi realizada uma avaliação experimental extensiva de forma a definir que combinações de codificação podem defender o sistema contra análise de tráfego.

Agradecimentos Este trabalho foi parcialmente suportado pela Fundação para a Ciência e Tecnologia (FCT) e pelo PIDDAC através do projecto com a referência UID/CEC/50021/2013.

Referências

1. A Child's Garden Of Pluggable Transports: `https://trac.torproject.org/projects/tor/wiki/doc/AChildsGardenOfPluggableTransports`
2. Aryan, S., Aryan, H., Halderman, J.A.: Internet censorship in Iran : A first look. In: Proc. of FOCI (2013)
3. Brubaker, C., Houmansadr, A., Shmatikov, V.: CloudTransport: Using Cloud Storage for Censorship-Resistant Networking. In: Proc. of PETS (2014)
4. Dyer, K.P., Coull, S.E., Shrimpton, T.: Marionette: A programmable network-traffic obfuscation system. In: Proc. of USENIX Security Symposium (2015)
5. FFmpeg: `https://sourceforge.net/projects/ffmpeg/`
6. Fifield, D., Lan, C., Hynes, R., Wegmann, P., Paxson, V.: Blocking-resistant communication through domain fronting. In: Proc. of PETS (2015)
7. Hahn, B., Nithyanand, R., Gill, P., Johnson, R.: Games without frontiers: Investigating video games as a covert channel. In: arXiv:1503.05904 [cs.CR] (2015)
8. Houmansadr, A., Brubaker, C., Shmatikov, V.: The parrot is dead: Observing unobservable network communications. In: Proc. of IEEE S&P (2013)
9. Houmansadr, A., Riedl, T.J., Borisov, N., Singer, A.C.: I want my voice to be heard: Ip over voice-over-ip for unobservable censorship circumvention. In: Proc. of NDSS (2013)
10. Li, S., Schliep, M., Hopper, N.: Facet: Streaming over videoconferencing for censorship circumvention. In: Proc. of WPES (2014)
11. Moghaddam, H., Li, B., Derakhshani, M., Goldberg, I.: Skypemorph: Protocol obfuscation for Tor bridges. In: Proc. of CCS (2012)
12. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth Mover's Distance As a Metric for Image Retrieval. Int. J. Comput. Vision 40(2), 99–121 (Nov 2000)
13. Snowmix: `https://sourceforge.net/projects/snowmix/`
14. Wicker, S.B.: Reed-Solomon Codes and Their Applications. IEEE Press (1994)
15. Wright, C.V., Coull, S.E., Monrose, F.: Traffic morphing: An efficient defense against statistical traffic analysis. In: Proc. of NDSS (2009)