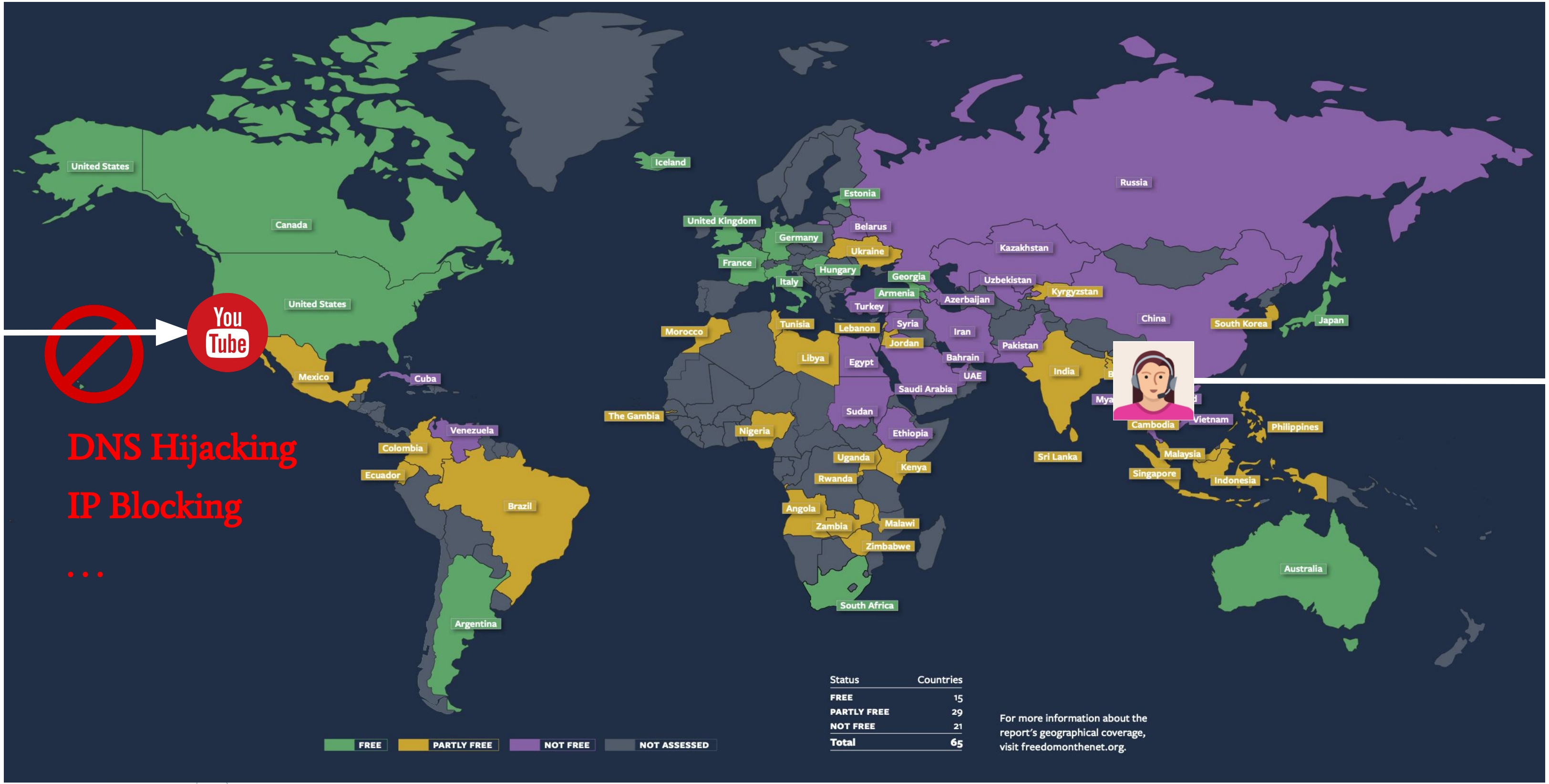
On the Unobservability of Multimedia-Based Covert Channels for Internet Censorship Circumvention

Diogo Barradas Nuno Santos Luís Rodrigues

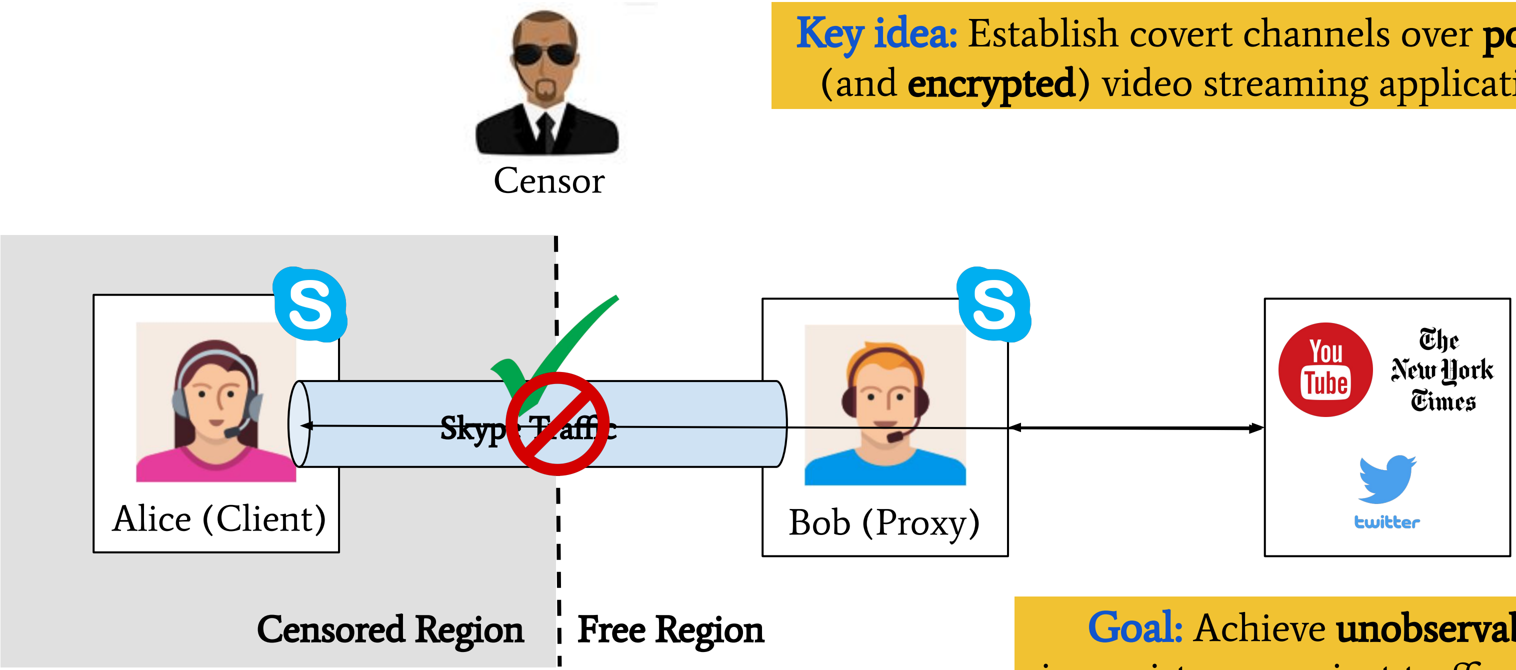
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa

Internet Censorship is Widespread



Bypassing Censorship with Video Streams

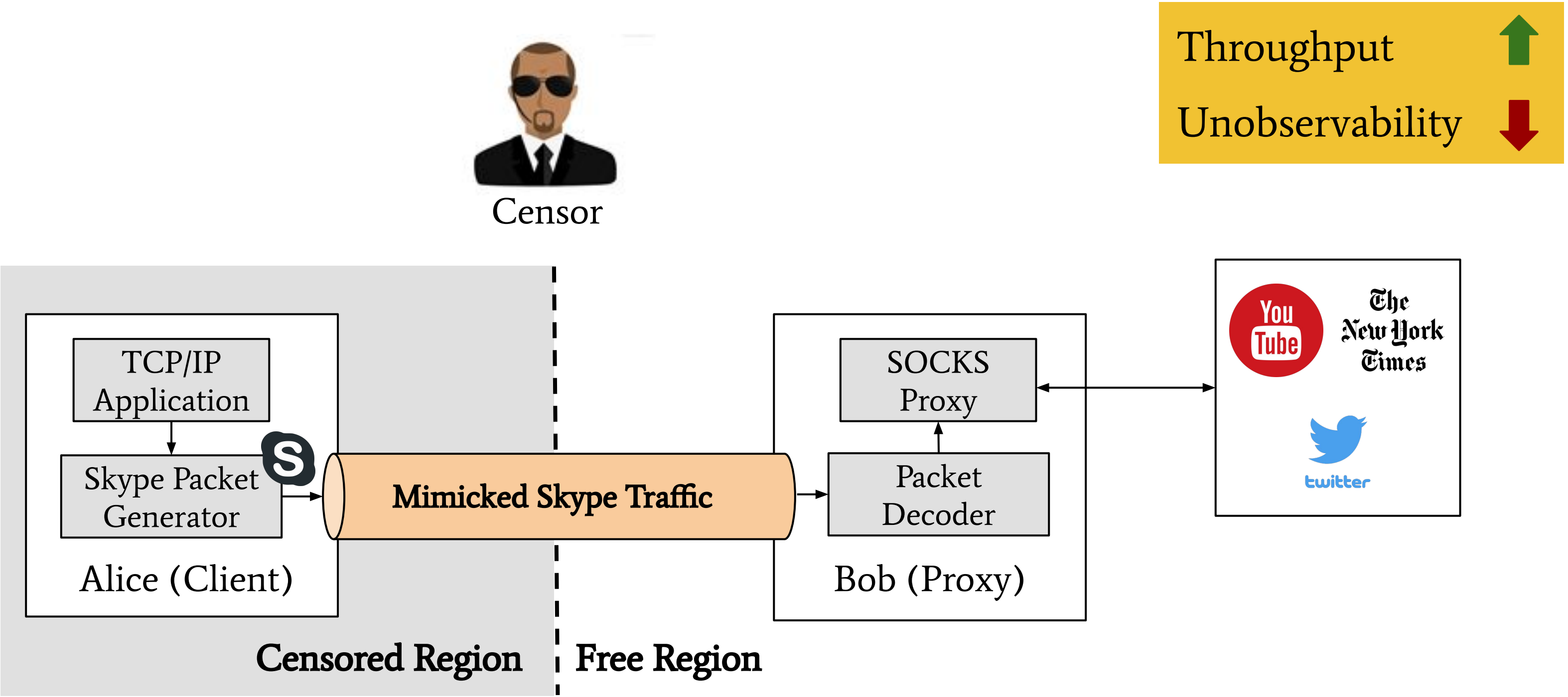
Key idea: Establish covert channels over **popular** (and **encrypted**) video streaming applications



Goal: Achieve **unobservability** i.e., resistance against traffic analysis

Mimicking Multimedia Protocols

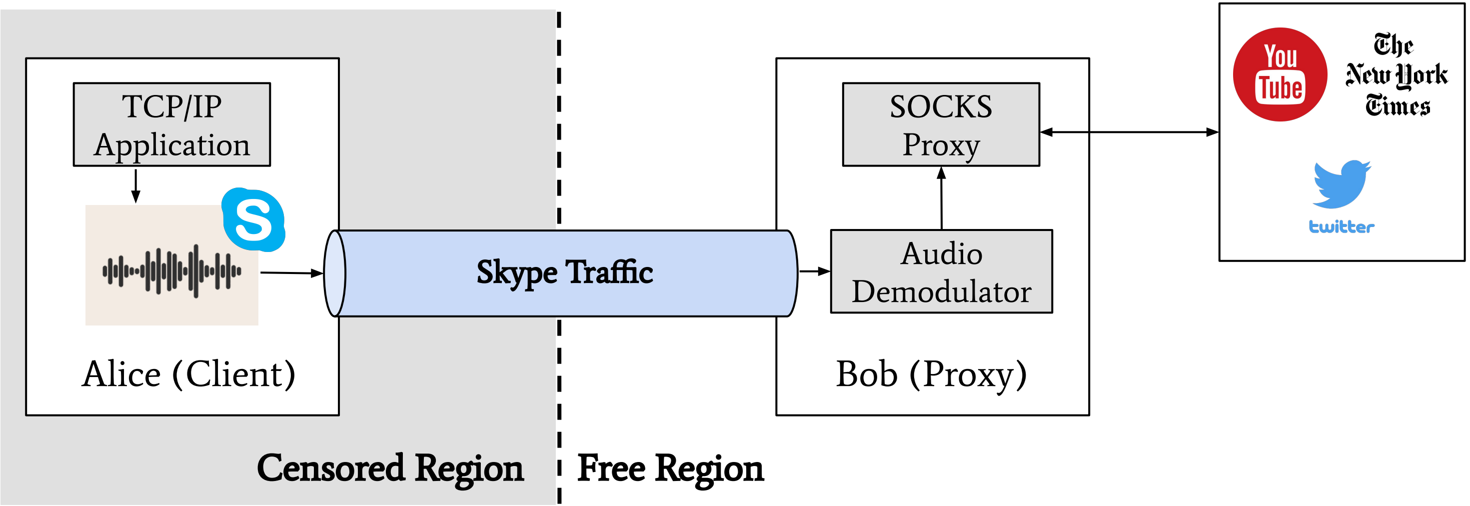
e.g. SkypeMorph [CCS '12]



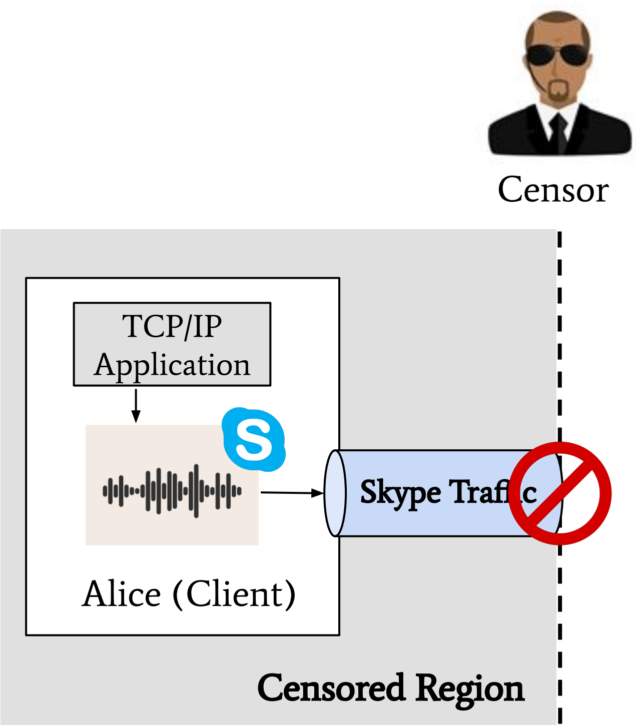
Tunneling Covert Data over Multimedia Protocols

e.g. FreeWave [NDSS '13]

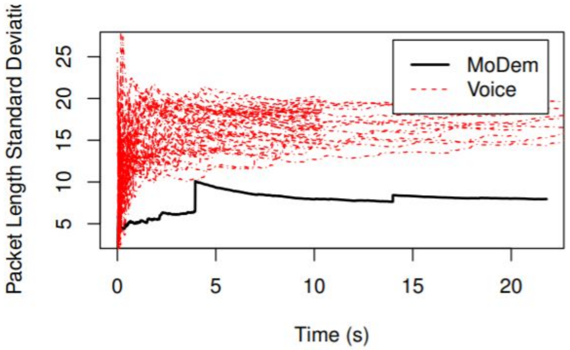
Throughput ⬇
Unobservability ⬆



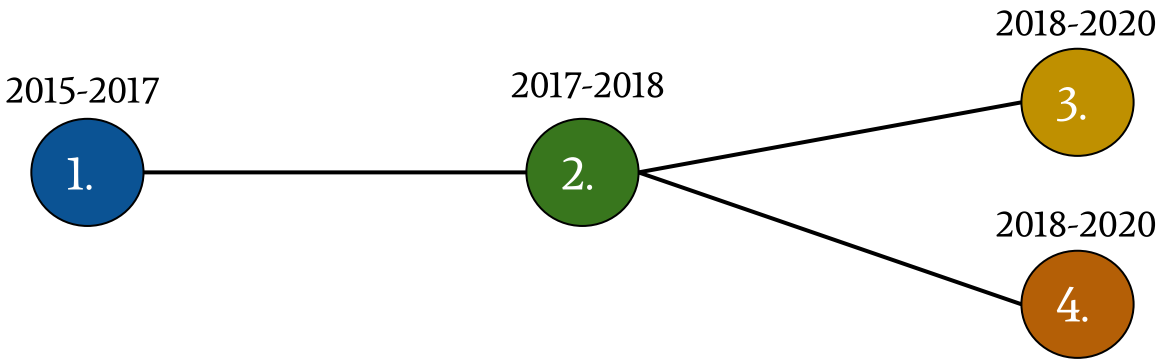
Multimedia Protocol Tunneling is Not a Silver Bullet



FreeWave is easily detected by checking **packet length standard deviation**



Our Research Path over the Past Five Years



1. Improvement of multimedia tunneling approaches
2. Evaluation of the unobservability of multimedia covert channels
3. Deployment of traffic analysis tools within the network
4. Development of a new encoded media tunneling tool

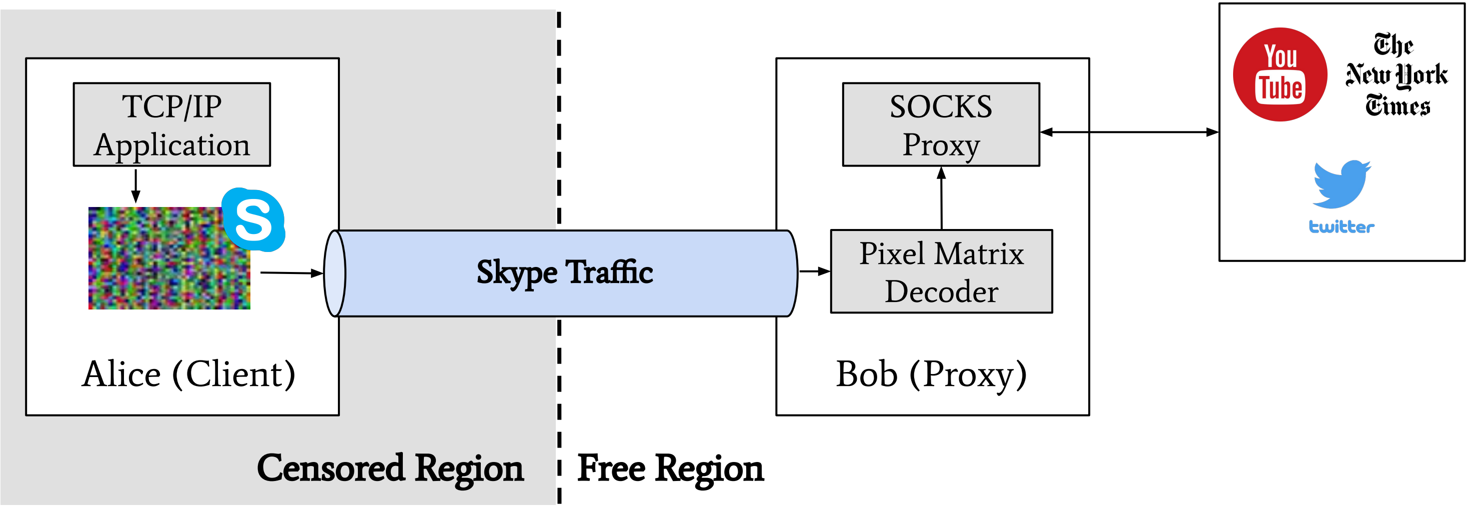
Can We Build a Better Multimedia Protocol Tunneling Tool?

- **Strive to maintain unobservability**
  - Adapt modulation to resist traffic analysis
- **Leverage a higher-bandwidth medium**
  - Use video-conferencing applications' video layer

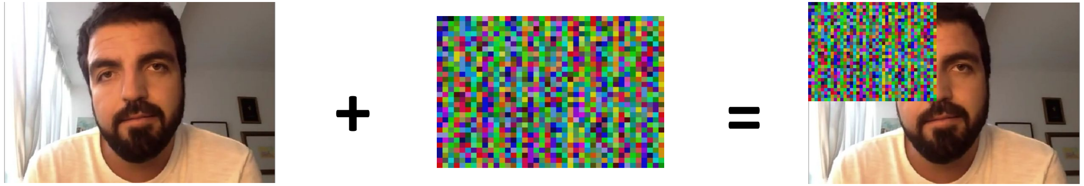
DeltaShaper: An Improved Tunneling Approach

How to maintain unobservability?

How to resist lossy compression?



DeltaShaper's Data Modulation Approach

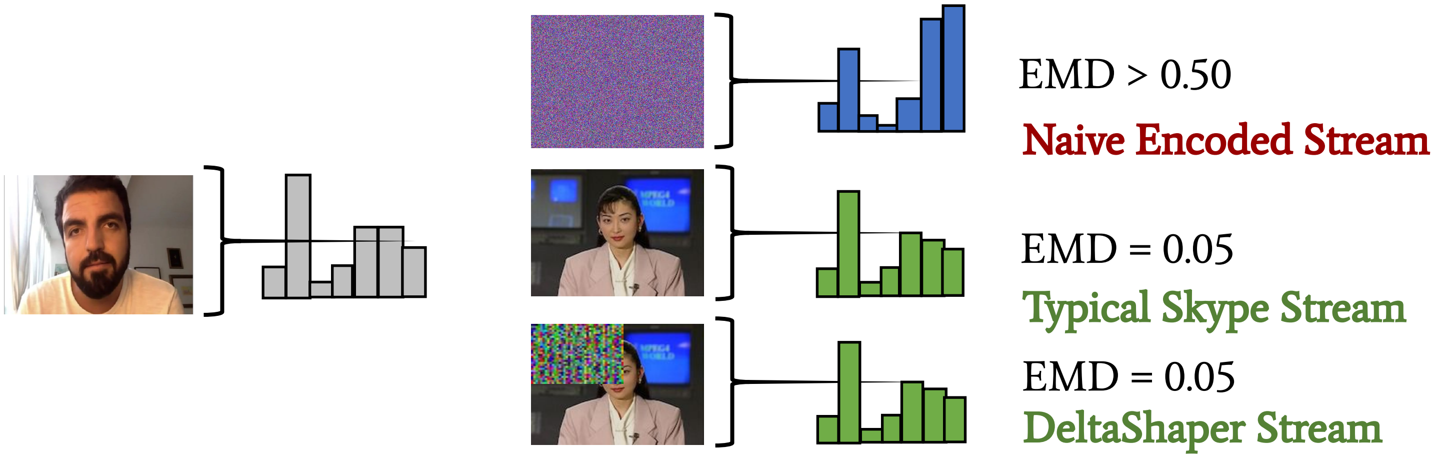


(a) Carrier Frame (b) Payload Frame (c) Covert Frame

Parameter	Description
ap	payload frame area (pixel×pixel)
ac	cell size (pixel×pixel)
bc	color encoding (bits)
rp	payload frame rate (frames/s)

Unobservability Assessment

- **Quantify differences between signatures with similarity metrics**
  - Packet lenght / inter-packet timing distributions
  - e.g., Earth Movers' Distance (EMD)

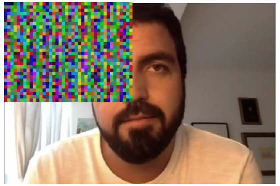


Performance of DeltaShaper

- **Performance**
  - Raw throughput: **7.2 Kbps**
  - Supports low-throughput, high-latency applications

- Achieved Configuration:

Parameter	Description	Configuration
a_p	payload frame area (pixel×pixel)	320 x 240
a_c	cell size (pixel×pixel)	8 x 8
b_c	color encoding (bits)	6
r_p	payload frame rate (frames/s)	1



Summary

- **DeltaShaper: A new censorship-resistant system**
    - Supports high-latency / low-throughput TCP applications
- **Maximizes throughput while preserving unobservability**
    - Greedy exploration of encoding configurations

Diogo Barradas, Nuno Santos, Luís Rodrigues
DeltaShaper: Enabling Unobservable Censorship-resistant TCP Tunneling over Videoconferencing Streams
In *Proc. of Privacy Enhancing Technologies (PETS)*, **2017**

Are We Doing a Good Job at Assessing Unobservability?

- **Evaluation with *ad hoc* similarity-based classifiers that:**
  - Depend on small (and similar) sets of traffic features
  - Have not been compared in the literature

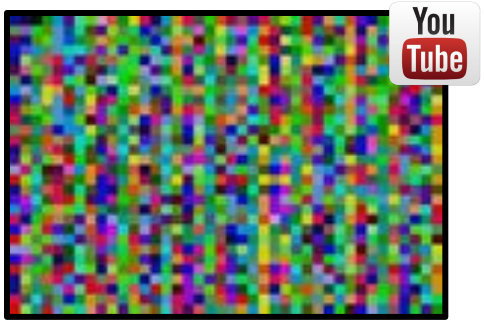
- **Poor evaluation leads to optimistic unobservability claims**
  - Ignores a wealth of research in machine learning techniques
  - Users of censorship-resistant tools may be endangered

Detection of Multimedia Protocol Tunneling

- The first extensive experimental study of the unobservability of covert channels produced by state-of-the-art MPT tools

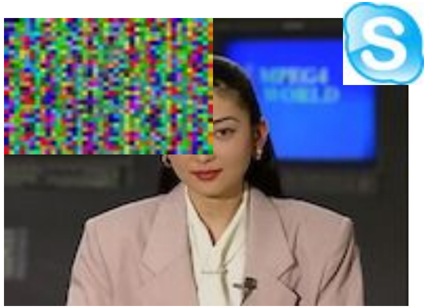
Facet (WPES'14)

Unidirectional (A/V)
Video Transmission



CovertCast (PETS'16)

Unidirectional (V)
Censored Websites Transmission



DeltaShaper (PETS'17)

Bidirectional (V)
Arbitrary Data Transmission

How was Unobservability Evaluation Performed?

- **Previous systems were evaluated with different similarity-based classifiers**
  - **Facet** : Pearson's Chi-squared Test ($\chi 2$)
  - **CovertCast** : Kullback-Leibler Divergence (KL)
  - **DeltaShaper** : Earth Mover's Distance (EMD)

- **Feature sets are similar (quantized frequency distributions)**
  - **Facet** : Packet size bi-grams
  - **CovertCast** : Packet size, inter-arrival delay
  - **DeltaShaper** : Packet size, inter-arrival delay

How Effective were Existing Detection Techniques?

Protocol Tunneling System	χ^2 Classifier (acc%)	KL Classifier (acc%)	EMD Classifier (acc%)
Facet ($s = 50\%$)	**74.3**	57.5	57.5

χ^2 is the most accurate classifier

KL and EMD are comparable
Recent classifiers offer worse accuracy

Can Other ML Techniques Better Detect Covert Channels?

- **Assess the effectiveness of multiple decision tree-based classifiers**
  - Decision Trees
  - Random Forests
  - **eXtreme Gradient Boosting (XGBoost)**
- **Models are easily interpretable**
- **Provide the ability to assess feature importance**

kaggle

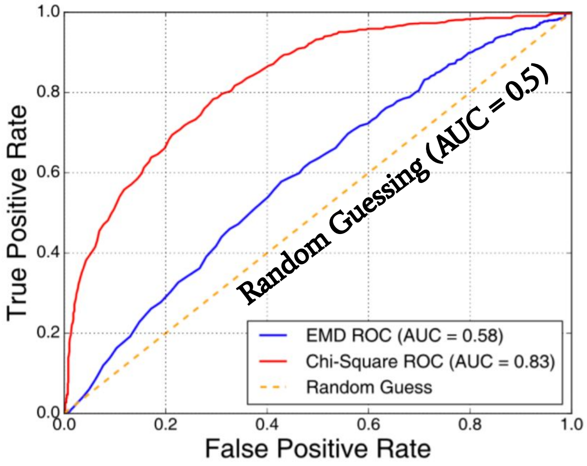
KDD2015

Which Features Could an Adversary Use?

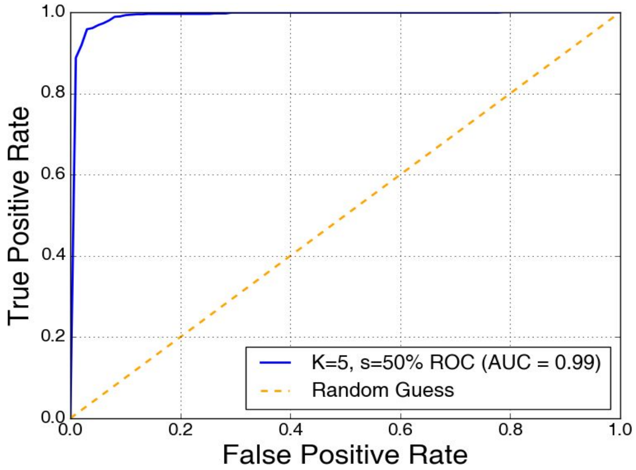
- **Feature set 1: summary statistics (ST)**
  - Total of 166 features, including simple statistics (e.g., max, min, percentiles), high order statistics (e.g., skew), and bursts

- **Feature set 2: quantized packet lengths (PL)**
  - Quantized PL frequency distribution for the flow carrying covert data
  - Each K size bin acts as an individual feature (K = 5 bytes)

Detection of Facet



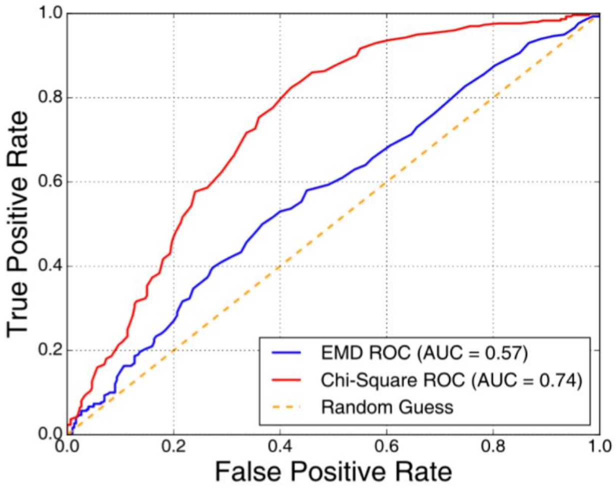
χ^2: 90% TPR = **45% FPR**

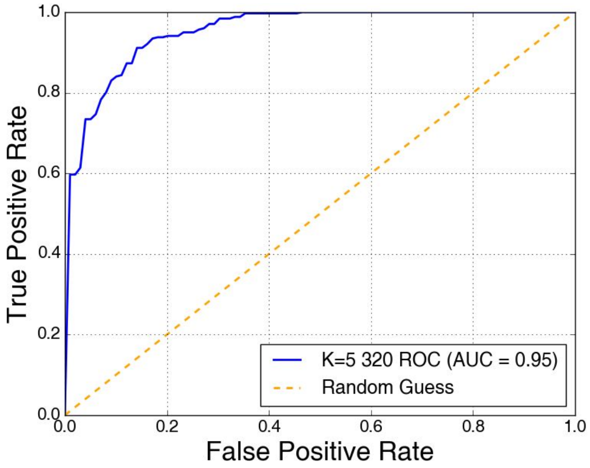


XGBoost-PL: 90% TPR = **2% FPR**

XGBoost-PL reduces the FPR when flagging the same amount of covert channels

Detection of DeltaShaper





χ^2: 90% TPR = **51% FPR**

XGBoost-PL: 90% TPR = **14% FPR**

DeltaShaper detection results follow a similar trend to those of Facet detection

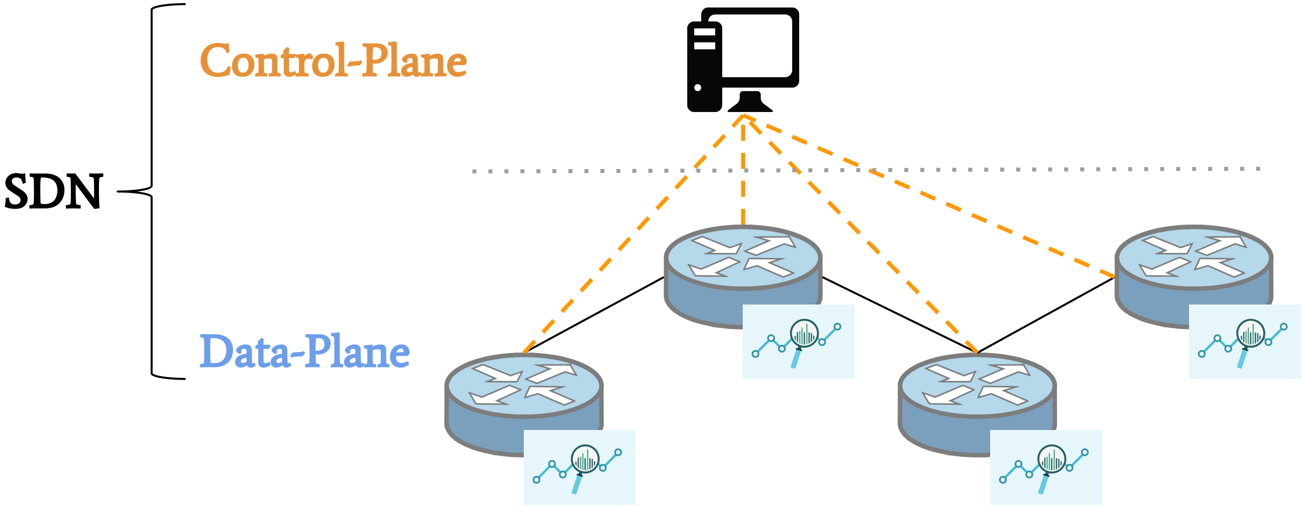
Summary

- **Compared similarity-based classifiers on the detection of MPT tools**
  - In general, unable to accurately detect covert channels
- **Explored multiple ML techniques for the detection of covert channels**
  - Decision tree-based classifiers can effectively detect existing MPT tools
- **Previous unobservability claims were flawed**

Diogo Barradas, Nuno Santos, Luís Rodrigues
Effective Detection of Multimedia Protocol Tunneling using Machine Learning
In *Proc. of USENIX Security Symposium*, **2018**

Can a Censor Leverage Programmable Switches to Gather and Classify Packet Distributions Efficiently?

In-network traffic analysis

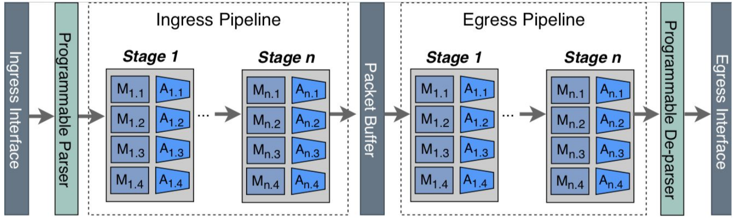


Line speed

No additional infrastructure

Less management costs

Collecting Packet Distributions in the Switches



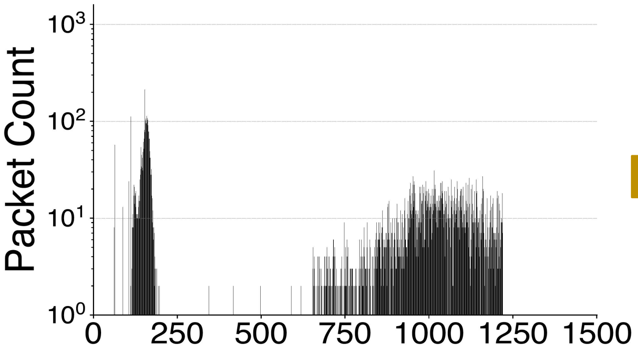
- **Stateful memory is severely limited**
  - O(100)MBs SRAM
  - No memory for storing many flows
- **Packets must be processed at line speed (actions < 1ns)**
  - No multiplications or floating point operations
  - Existing packet distribution compression techniques **do not work**
- **We need a packet distribution representation that:**
  - Provides **high accuracy** and requires **small amount of memory**
  - Can be **implemented efficiently** in programmable switches

How Can We Compress Packet Distributions?

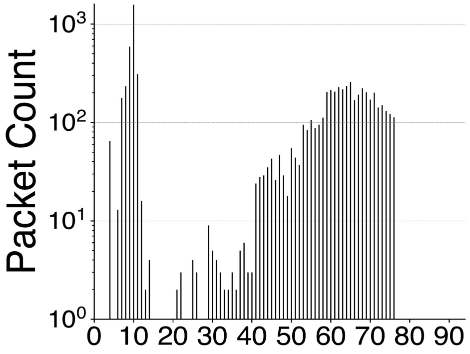
In-network traffic analysis

- **Produce flow markers with two simple operators:**
  - **Quantization** - discretize the packet distribution into bins
  - **Truncation** - select the most relevant bins for classification

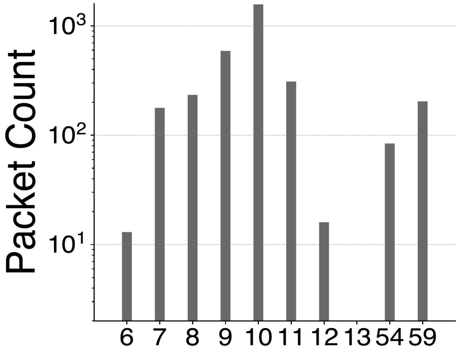
Up to 150x size reduction



Raw packet size distribution



Quantized distribution
QL = 16

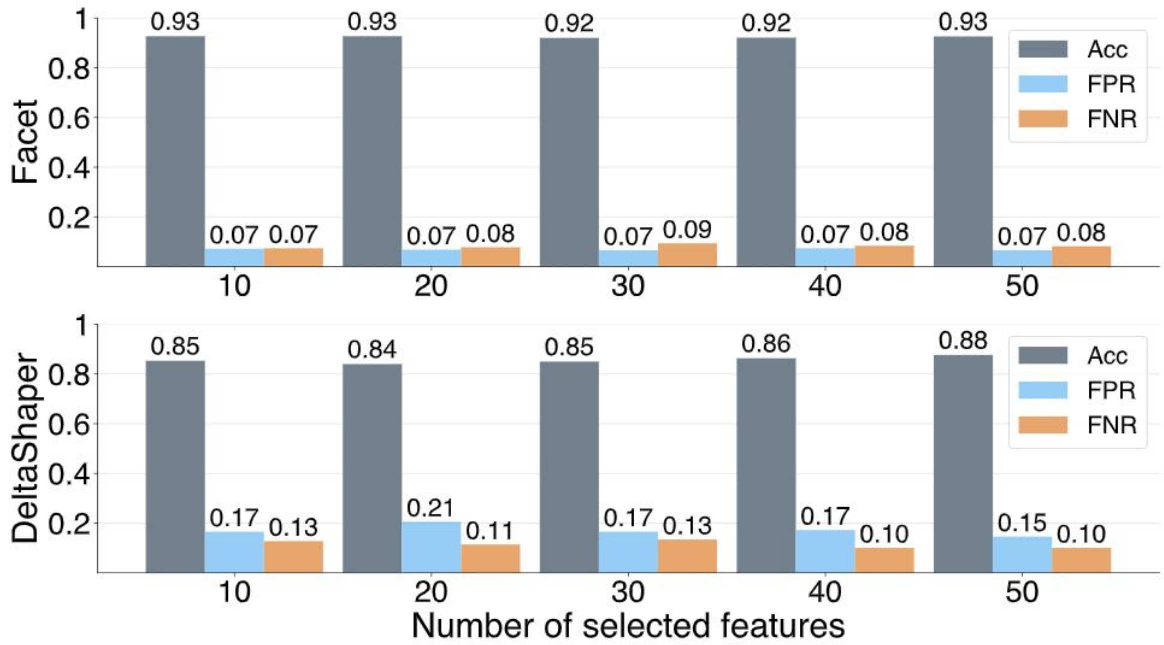


Truncated distribution
Top-10 bins

Truncation (w/ Quantization QL=16)

Applied to Multimedia Covert Channel Detection

In-network traffic analysis



Full information = 3000B
Facet: 96% acc.
DeltaShaper: 87% acc

Quant + Trunc = 20B
Facet: 93% acc.
DeltaShaper: 85% acc

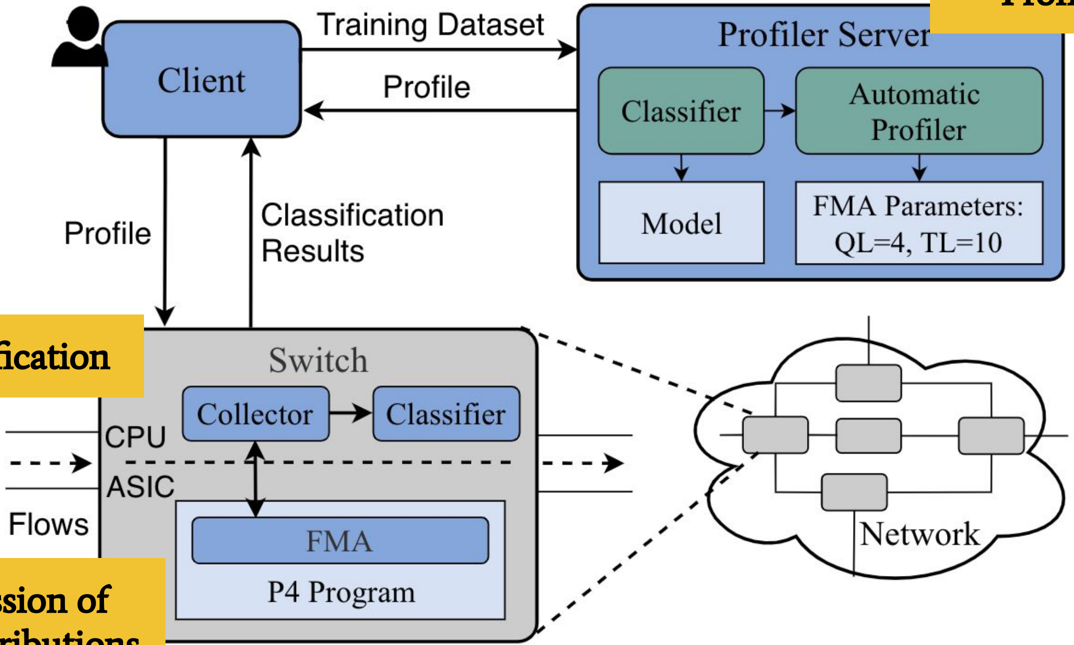
Only up to - 3% accuracy
150x less memory

Automatic Profiling

In-network traffic analysis

- **Automate the configuration choice**
  - Large configuration space = **Quantization x Truncation**
- **Leverage Bayesian Optimization**
- **Three different criteria for selecting a configuration**
  - Smaller marker for target accuracy
  - Best accuracy given a size constraint
  - Fully automatic (compromise between marker size and accuracy)

FlowLens



Profiling

Classification

Compression of packet distributions

Summary

- **FlowLens: ML-based traffic analysis system for programmable switches**
- **Compress packet distributions into flow markers**
  - Reduction of memory footprint (1-2 orders of magnitude)
  - Comparable accuracy to full information
- **Automatic profiling to choose optimal configurations**

Diogo Barradas, Nuno Santos, Luís Rodrigues, Salvatore Signorello, Fernando Ramos, André Madeira
FlowLens: Enabling Efficient Flow Classification for ML-based Network Security Applications
In *Proc. of Network and Distributed Systems Symposium (NDSS),* **2021** *(to appear)*

Revisiting the Design of Multimedia Covert Channels

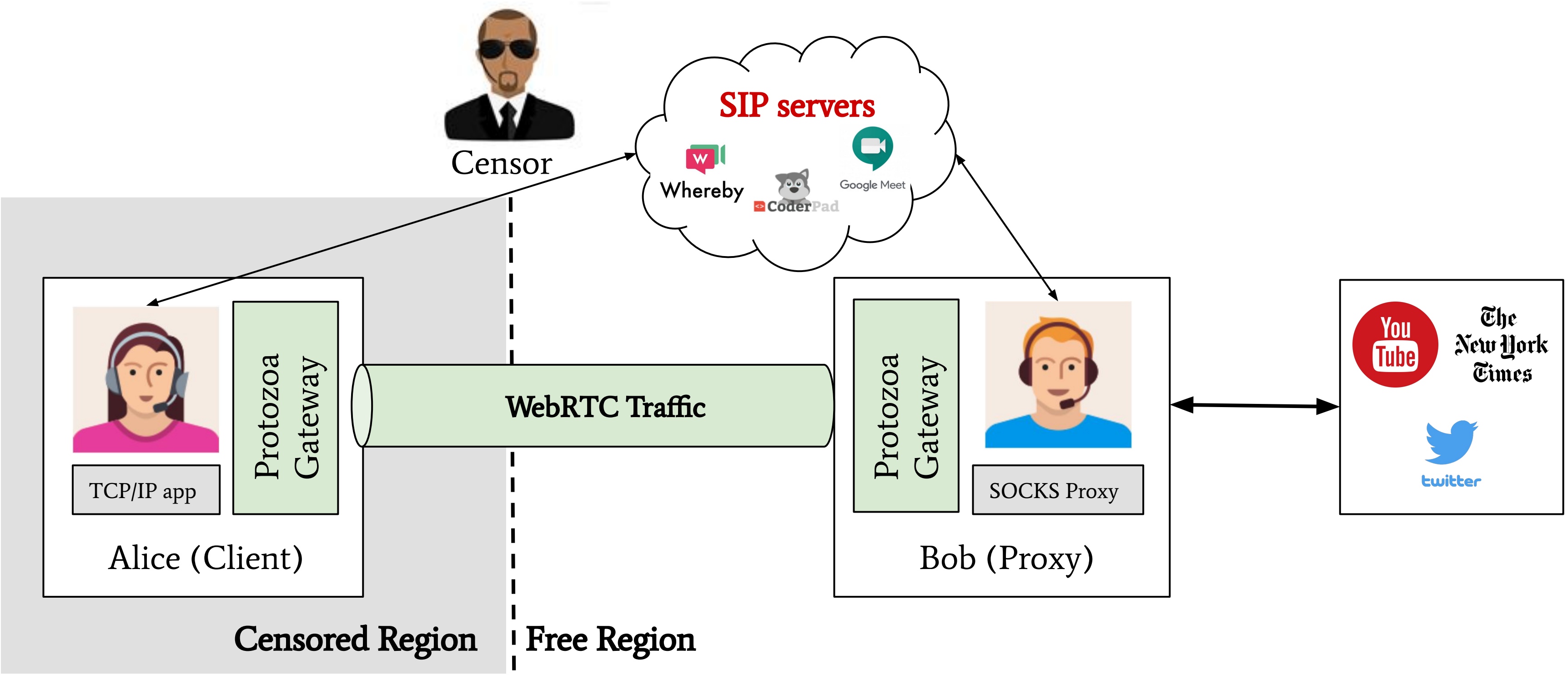
Encoded Media Tunneling

- **Can we generate covert streams that resist traffic analysis?**
- **Can we increase throughput w.r.t. existing tunneling approaches?**
- **Tunneling works without access to implementation**
  - But what if we could access the **innards of the multimedia pipeline**?
  - Are there any **widely used applications** that match this profile?

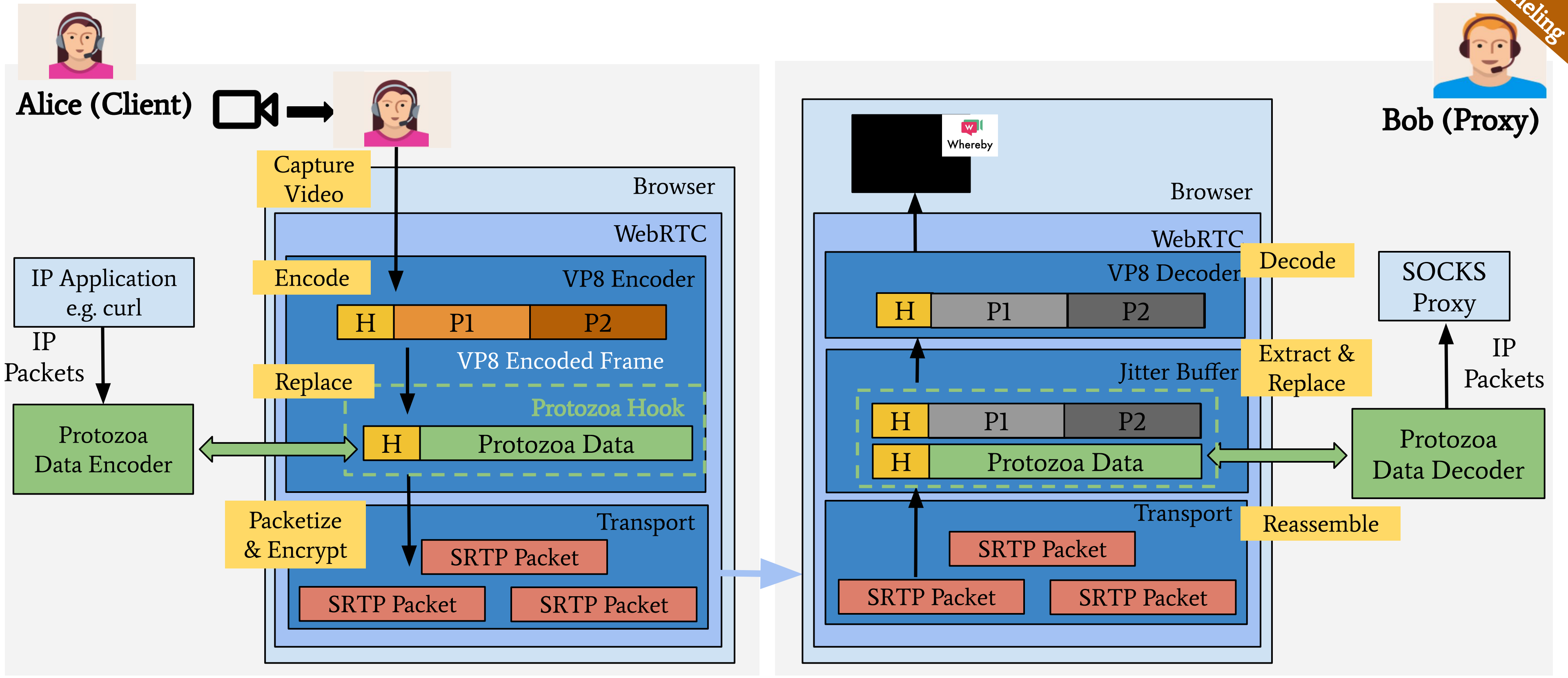
WebRTC

- **Framework that provides real-time communication capabilities**
    - Exposes a set of JavaScript APIs on **all major browsers**
    - Used by an **increasing number of trending applications**
    - **Open-source**

Protozoa: A New Censorship Circumvention Tool

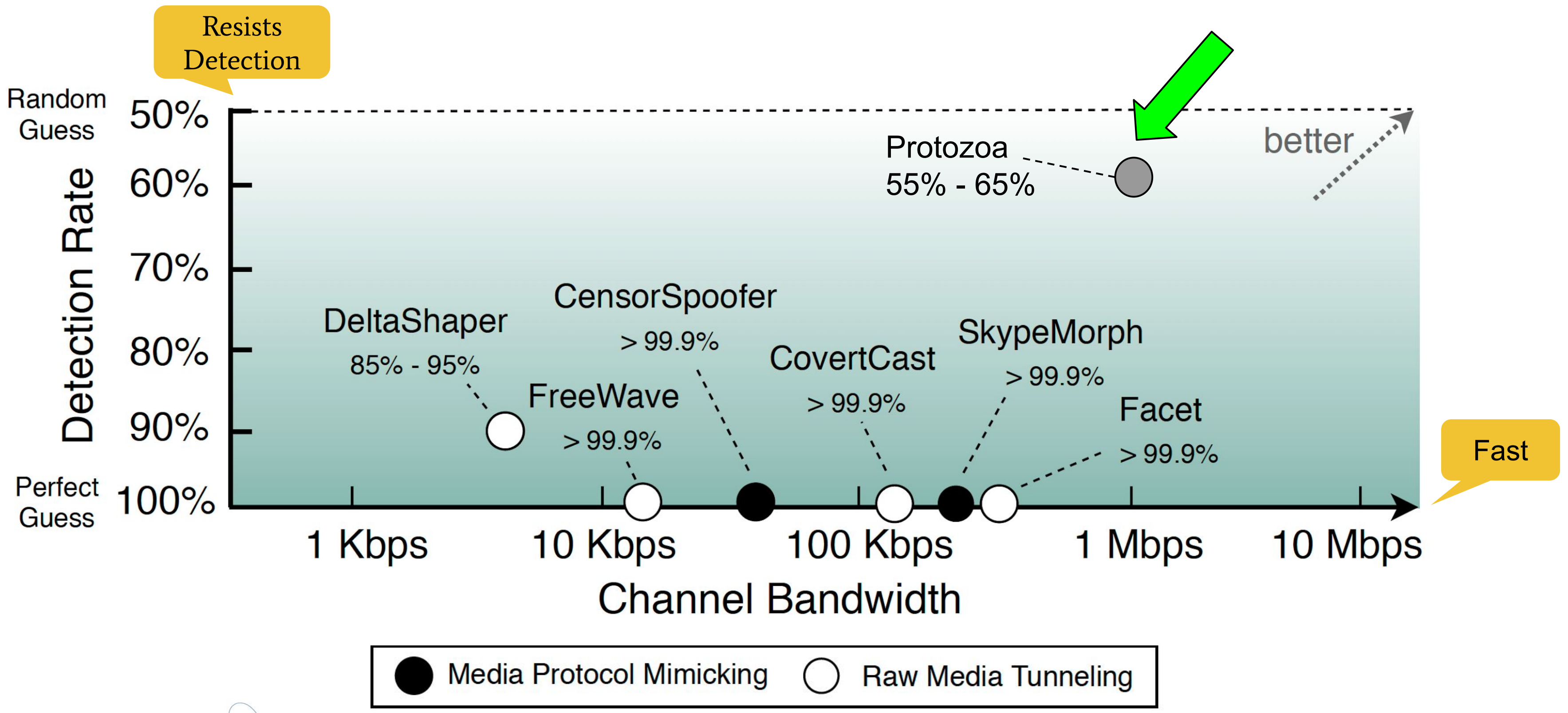


How Does Protozoa Encode Covert Data?



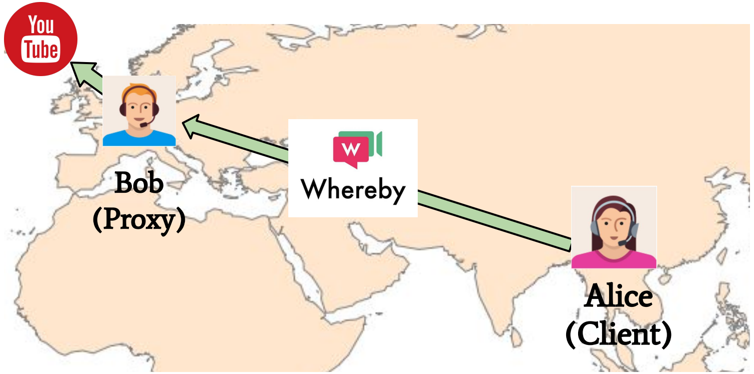
Protozoa is Fast and Resistant against Traffic Analysis

Encoded Media Tunneling



Validation in the Real-World

WebRTC Application	Reachability		
	China	Russia	India
appr.tc	-	✓	✓
aws.amazon.com/chime	✓	✓	✓
codassium.com	✓	✓	✓
coderpad.io	✓	✓	✓
discordapp.com	-	✓	✓
gotomeeting.com	✓	✓	✓
hangouts.google.com	-	✓	✓
messenger.com	-	✓	✓
slack.com	✓	✓	✓
whereby.com	✓	✓	✓



Multiple WebRTC apps are available in countries known to experience Internet censorship

Protozoa makes it possible to access blocked content / services (e.g. YouTube)

Summary

- First to leverage **WebRTC video streams** to create covert channels
- Introduces a new encoding mechanism: **encoded media tunneling**
    - Instruments the media pipeline in the WebRTC stack to replace encoded video
- Works over a range of existing **unmodified WebRTC apps** (e.g., Whereby)
    - Deployed against real censors (China, Russia, India)

Diogo Barradas, Nuno Santos, Luís Rodrigues, Vítor Nunes
Poking a Hole in the Wall: Efficient Censorship-Resistant Internet Communications by Parasitizing on WebRTC
In *Proc. of ACM Conference on Computer and Communications Security (CCS),* **2020**

Conclusions and Future Directions

- **MPT's unobservability is only as strong as the classifier used to assess it**
  - Can we apply information theoretical frameworks to assess unobservability?
- **So far, unobservability has been tested in the lab with synthesized traffic**
  - Is it possible to gather more realistic data (e.g. campus network)?
- **Censors' traffic analysis capabilities are getting more sophisticated**
  - Able to inspect large volumes of traffic at Tbps speeds
  - Understanding the innards of media pipelines is an important step towards unobservable multimedia covert channels

`https://web.ist.utl.pt/diogo.barradas`

Thank You!