When network programmability meets security

The good. The bad. And the ugly.

Laurent Vanbever
nsg.ee.ethz.ch

FFSPIN 2022
Mon Aug 22 2022

"Van Gogh painting of
three cow boys
facing forward"

— OpenAI Dall.E 2

The bad

The ugly

The good

We ❤️ network programmability

Network programmability enables *unpreceedented*

visibility and controllability of network traffic

network programmability can...

network programmability can

- protect against

new types of network attacks

network programmability can

- protect against the good

new types of network attacks

network programmability can

- protect against
- be the target of

new types of network attacks

network programmability can

- protect against
- be the target of

the bad

new types of network attacks

network programmability can

- protect against
- be the target of
- enable

new types of network attacks

network programmability can

- protect against
- be the target of
- enable

the ugly

new types of network attacks

When network programmability meets network security

1. Protecting users
   The good
2. Being attacked
   The bad
3. Attacking others
   The ugly

When network programmability meets network security

1 Protecting users

The good

Being attacked

The bad

Attacking others

The ugly

Protecting users…

Protecting users

with programmable hardware

Protecting users
with programmable hardware

Proactively

Reactively

Protecting users
with programmable hardware

Proactively

Reactively

obfuscating network metrics
without loosing performance

Protecting users
with programmable hardware

Proactively

Reactively

neutralizing next-gen DDoS attacks
without impacting production traffic

Protecting users
with programmable hardware

Proactively

Reactively

obfuscating network metrics
without loosing performance

ditto:
Network Traffic Obfuscation at Line Rate

Roland Meier[1],
Vincent Lenders[2],
Laurent Vanbever[1]

NDSS 2022

(1) ETH zürich

(2) Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse

Traffic volume and timing allows

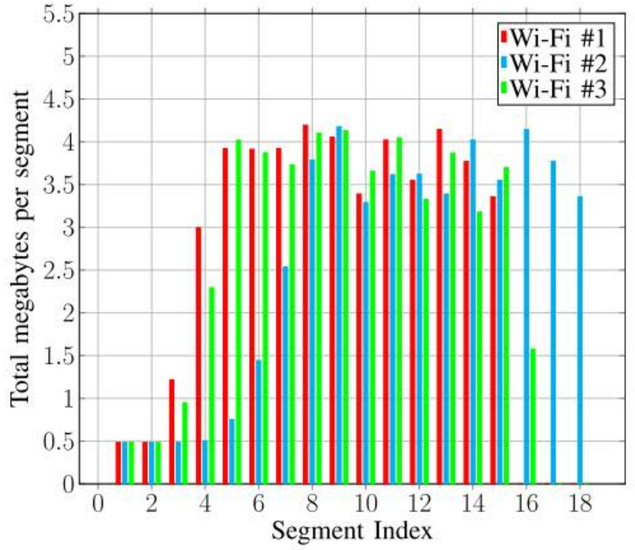
to determine which video somebody is watching

IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 12, NO. 12, DECEMBER 2017 3039

I Know What You Saw Last Minute—Encrypted HTTP Adaptive Video Streaming Title Classification

Ran Dubin, Amit Dvir, Ofir Pele, and Ofer Hadar, *Senior Member, IEEE*

***Abstract*—Desktops can be exploited to violate privacy. There are two main types of attack scenarios: active and passive. We consider the passive scenario where the adversary does not interact actively with the device, but is able to eavesdrop on the network traffic of the device from the network side. In the near future, most Internet traffic will be encrypted and thus passive attacks are challenging. Previous research has shown that information can be extracted from encrypted multimedia streams. This includes video title classification of non HTTP adaptive streams. This paper presents algorithms for encrypted HTTP adaptive video streaming title classification. We show that an external attacker can identify the video title from video HTTP adaptive streams sites, such as YouTube. To the best of our knowledge, this is the first work that shows this. We provide a large data set of 15 000 YouTube video streams of 2100 popular video titles that was collected under real-world network conditions. We present several machine learning algorithms for the task and run a thorough set of experiments,**



Traffic volume and timing allows to identify characteristics of the endpoint

Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application

Jonathan Muehlstein*, Yehonatan Zion*, Maor Bahumi†, Itay Kirshenboim*†
Ran Dubin‡, Amit Dvir*, Ofir Pele*†
* Center for Cyber Technologies, Department of Computer Science, Ariel University
† Center for Cyber Technologies, Department of Electrical and Electronics Engineering, Ariel University
‡ Department of Communication Systems Engineering, Ben-Gurion University of the Negev

***Abstract*—Desktops and laptops can be maliciously exploited to violate privacy. There are two main types of attack scenarios: active and passive. In this paper, we consider the passive scenario where the adversary does not interact actively with the device, but he is able to eavesdrop on the network traffic of the device from the network side. Most of the internet traffic is encrypted and thus passive attacks are challenging. In this paper, we show that an external attacker can identify the operating system, browser and application of HTTP encrypted traffic (HTTPS). To the best of our knowledge, this is the first work that shows this. We provide a large data set of more than 20000 examples for this task. Additionally, we suggest new features for this task. We run a through a set of experiments, which shows that our classification accuracy is 96.06%.**

***Index Terms*—Encrypted Traffic, HTTPS, Operating System, Browser, Application**

I. Introduction

applications. Alshamarri et al. [36] compared AdaBoost, Support Vector Machines, Naïve Bayes, RIPPER and C4.5 in order to classify Skype traffic. Donato et al. [39] presented a method for application classification called the Traffic Identification Engine.

Niemczyk et al. [38] suggested to divide the session to time buckets (10 seconds). The features that were used for each bucket are packet size counts and the time differences between packets. They found the recognition rate of Skype was almost perfect. However, their method was not able to differentiate between browsers and between joint application and browser usage.

Feature extraction methods for traffic classification include session duration [36], number of packets in a session [32], [40], minimum, maximum and average values of inter-arrival

Traffic volume and timing allows to determine search queries

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies ; 2017 (4):251–270

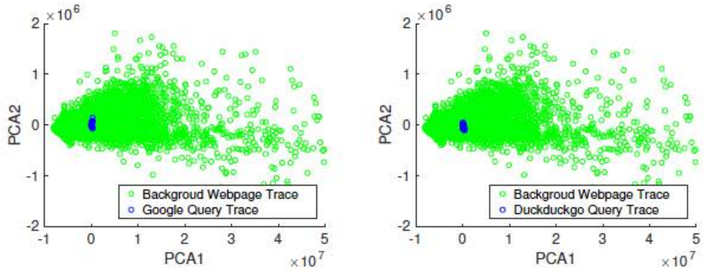
Se Eun Oh*, Shuai Li, and Nicholas Hopper

Fingerprinting Keywords in Search Queries over Tor

Abstract: Search engine queries contain a great deal of private and potentially compromising information about users. One technique to prevent search engines from identifying the source of a query, and Internet service providers (ISPs) from identifying the contents of queries is to query the search engine over an anonymous network such as Tor.

In this paper, we study the extent to which Website Fingerprinting can be extended to fingerprint individual queries or keywords to web applications, a task we call Keyword Fingerprinting (KF). We show that by augmenting traffic analysis using a two-stage approach with new task-specific feature sets, a passive network adversary can in many cases defeat the use of Tor to protect search engine queries.

We explore three popular search engines, Google, Bing, and Duckduckgo, and several machine learning techniques with various experimental scenarios. Our experi-



(a) Google (b) Duckduckgo

Fig. 1. Principal Components Analysis (PCA) Plot of Google and Duckduckgo query traces and background webpage traces based on CUMUL feature set

Bing and Google, and ISPs, are in a position to collect sensitive details about users. These search queries have also been among the targets of censorship [29] and surveillance infrastructures [17] built through the cooperation of state and private entities.

One mechanism to conceal the link between users

… and much (really much) more

Globecom 2014 - Communication and Information System Security Symposium

Website Fingerprinting using Traffic Analysis of Dynamic Webpages

Yan Shi and Subir Biswas
Electrical and Computer Engineering, Michigan State University, East Lansing, MI

Nothing But Net: Invading Android User Privacy Using Only Network Access Patterns

Mikhail Andreev[1], Avi Klausner[1], Trishita Tiwari[1], Ari Trachtenberg[1], and Arkady Yerukhimovich[2]

Speaker Recognition in Encrypted Voice Streams

Michael Backes[1,2], Goran Doychev[1], Markus Dürmuth[1], and Boris Köpf[2]

Silhouette – Identifying YouTube Video Flows from Encrypted Traffic

Feng Li
Verizon Labs
Waltham, MA, 02145

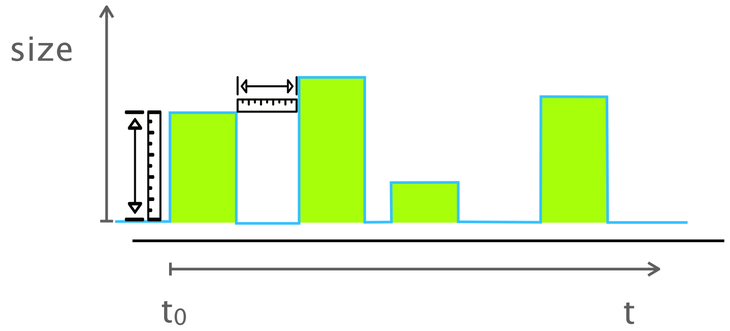
Jae Won Chung
Verizon Labs
Waltham, MA, 02145

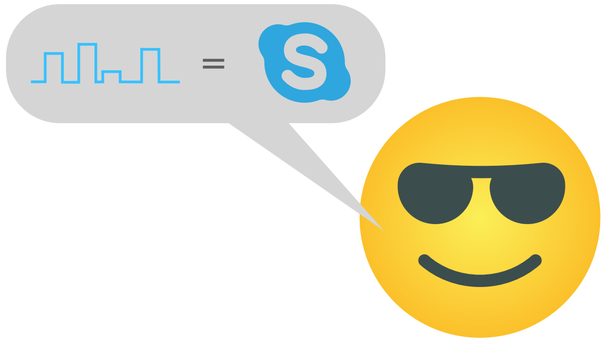
Mark Claypool
Worcester Polytechnic Institute
Worcester, MA, 01609

Inferring Users' Online Activities Through Traffic Analysis

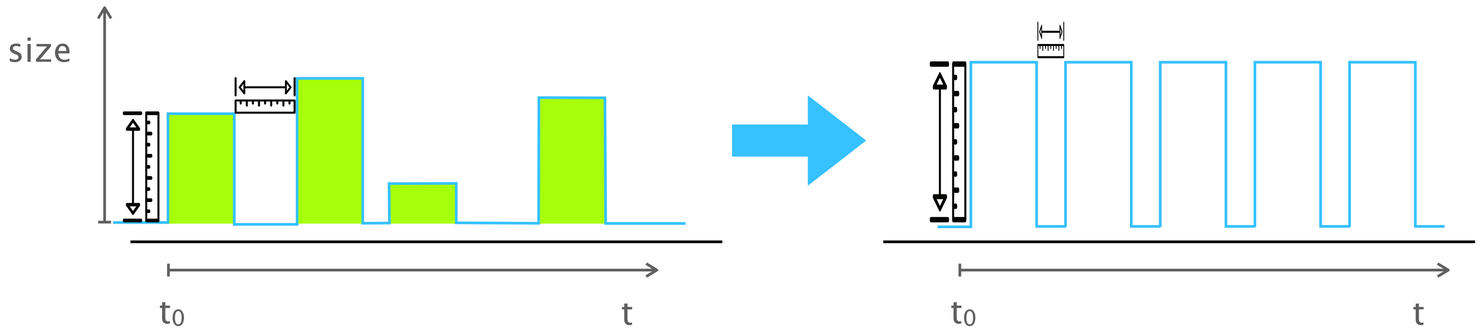
Fan Zhang[1,3], Wenbo He[1], Xue Liu[2] and Patrick G. Bridges[4]
Department of Electrical Engineering, University of Nebraska-Lincoln, NE, USA[1]
School of Computer Science, McGill University, Montreal, Quebec, Canada[2]
Department of Electronics and Information, Huazhong University of Sci. & Tech., Wuhan, China[3]

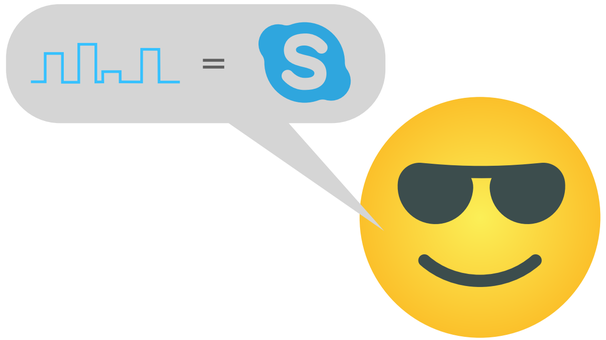
Packet sizes and timing allow traffic-analysis attacks, even if the traffic is encrypted



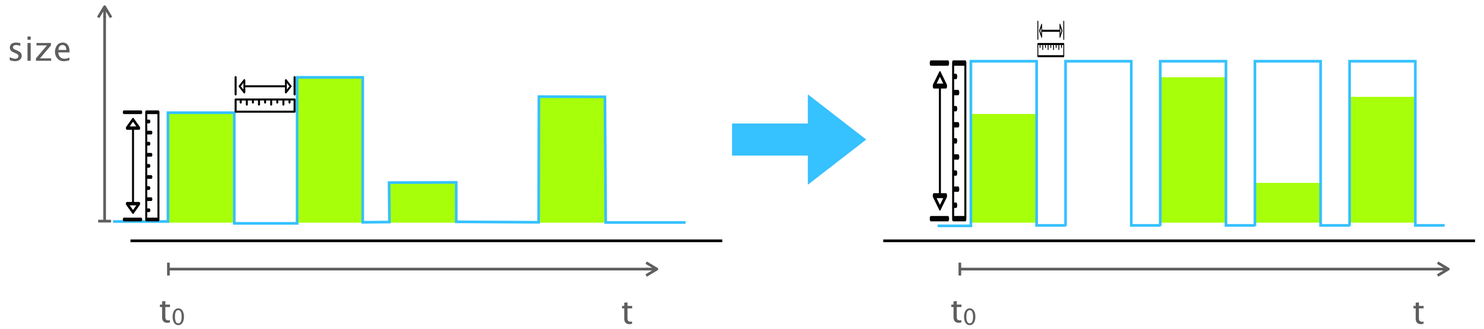


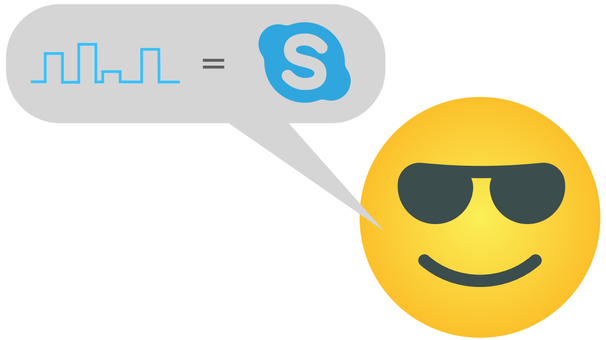
ditto renders the observed traffic *independent* from the real traffic

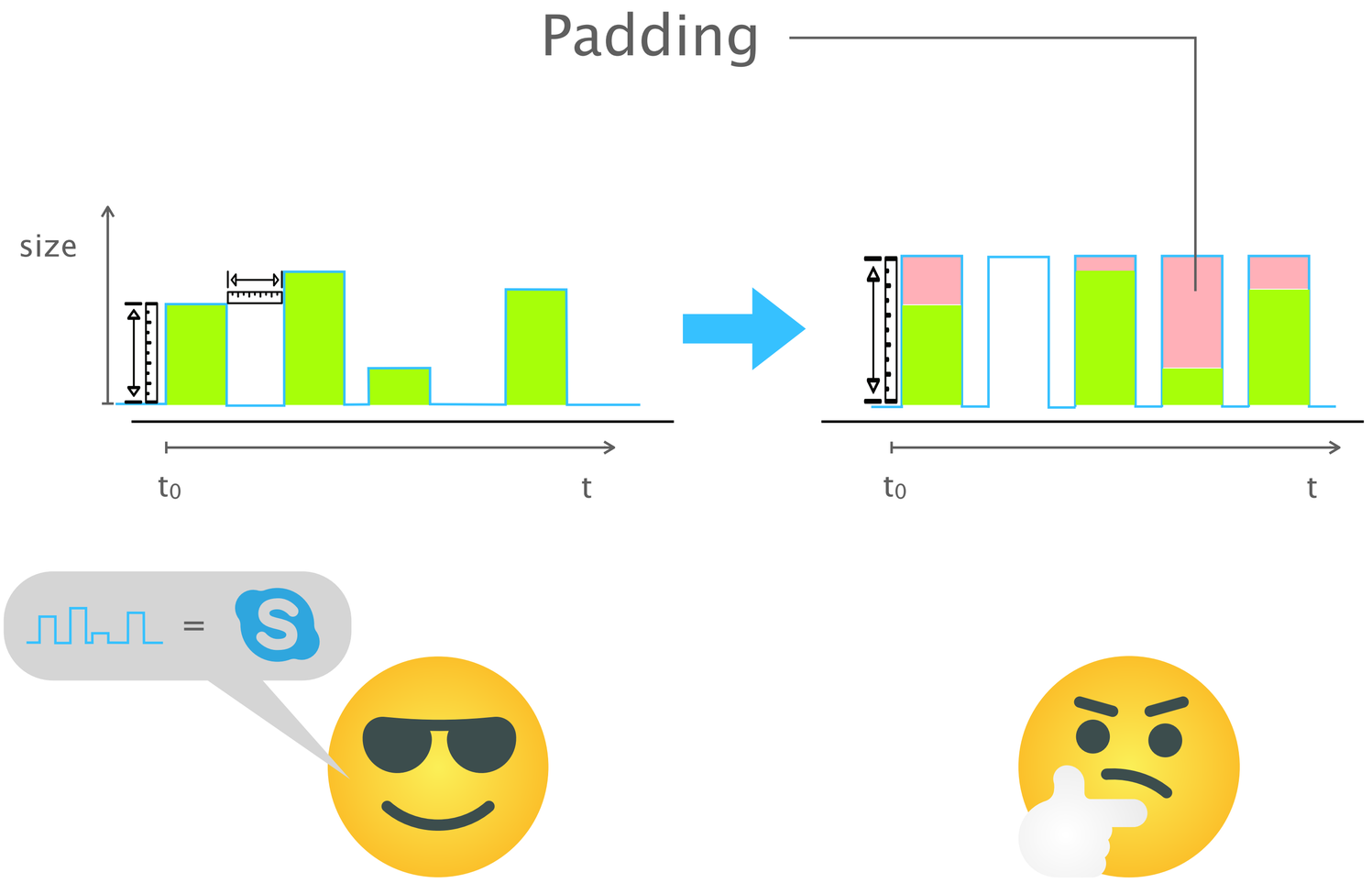


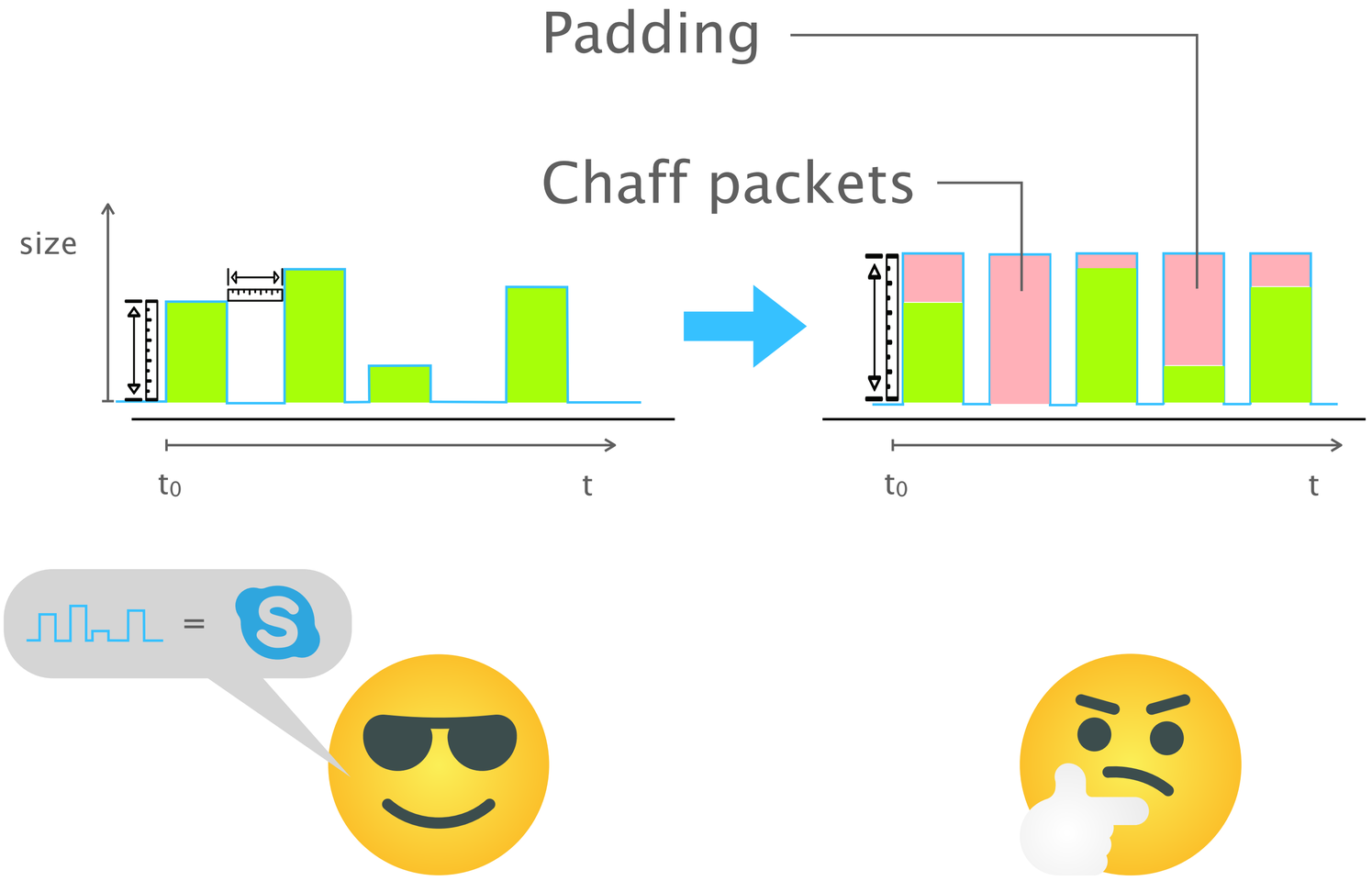


ditto renders the observed traffic *independent* from the real traffic

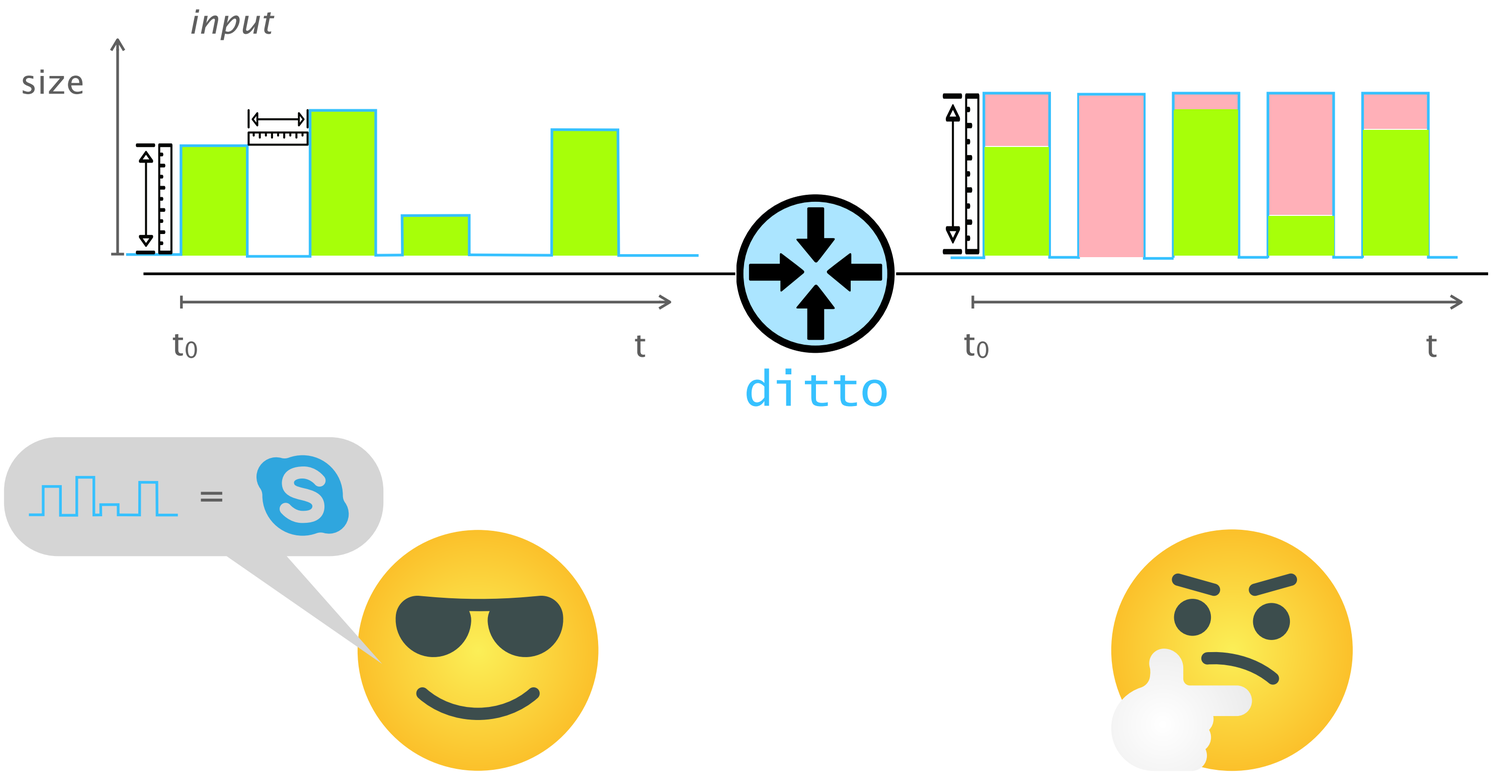




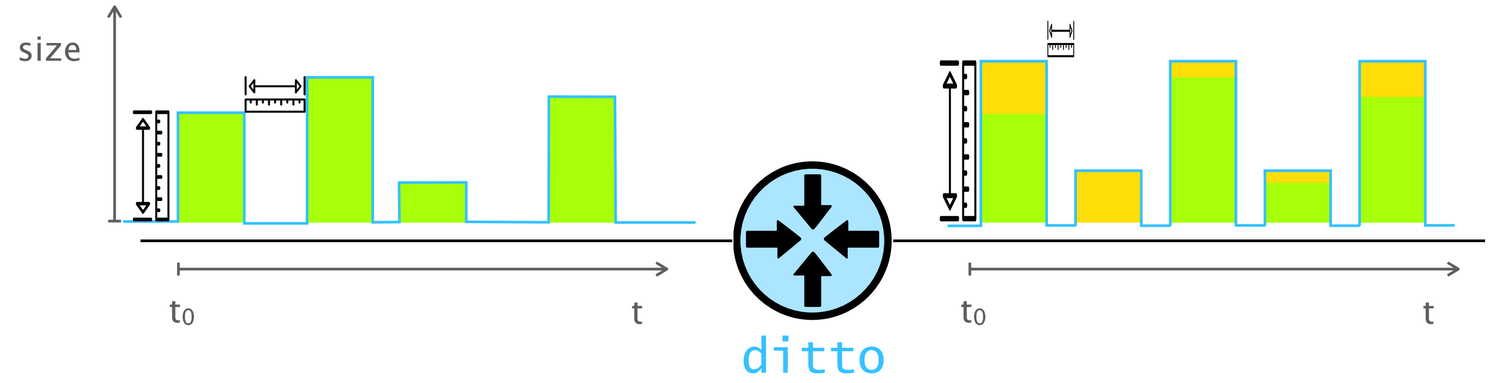

Padding
size
t0
t
t0
t
=

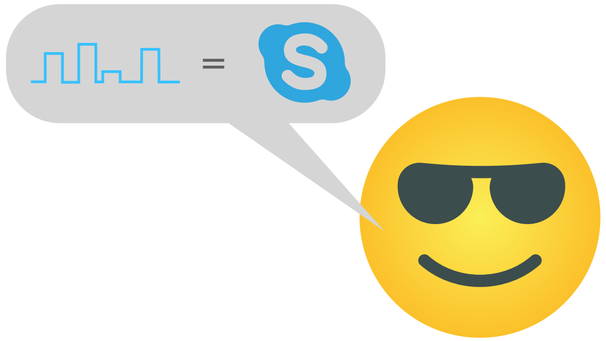

Padding
Chaff packets
size
t0
t
t0
t
=

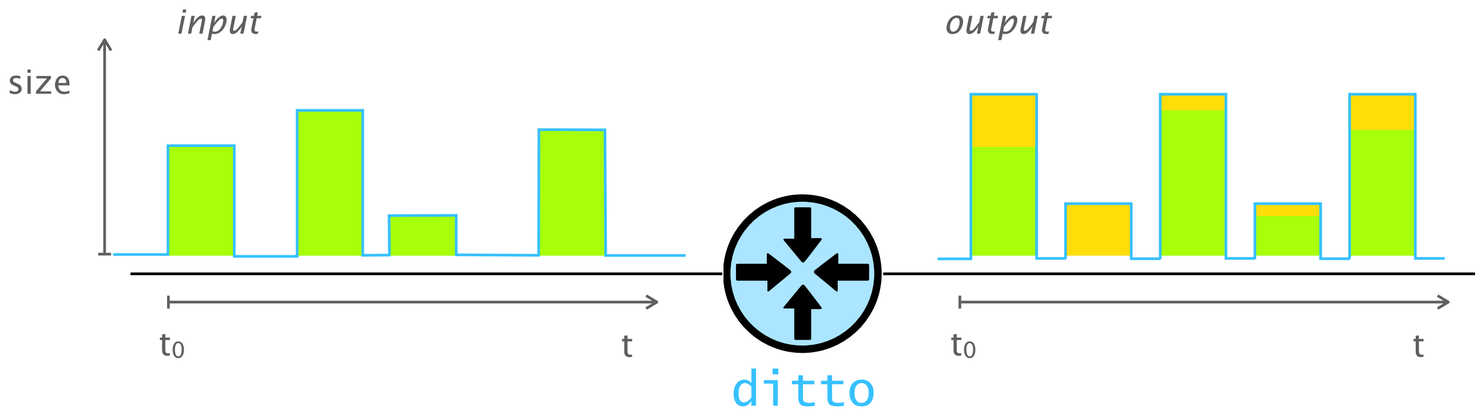
To ensure performance,
ditto runs directly in the network data plane



To ensure performance,
ditto shapes traffic according to an efficient pattern



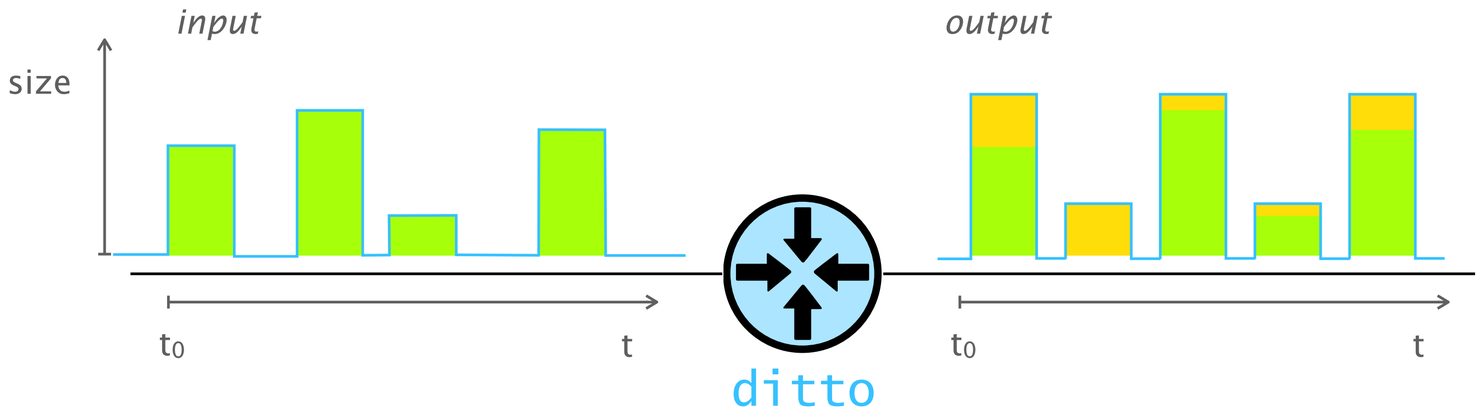




Computing efficient traffic patterns

Traffic shaping in the data plane

Experimental results

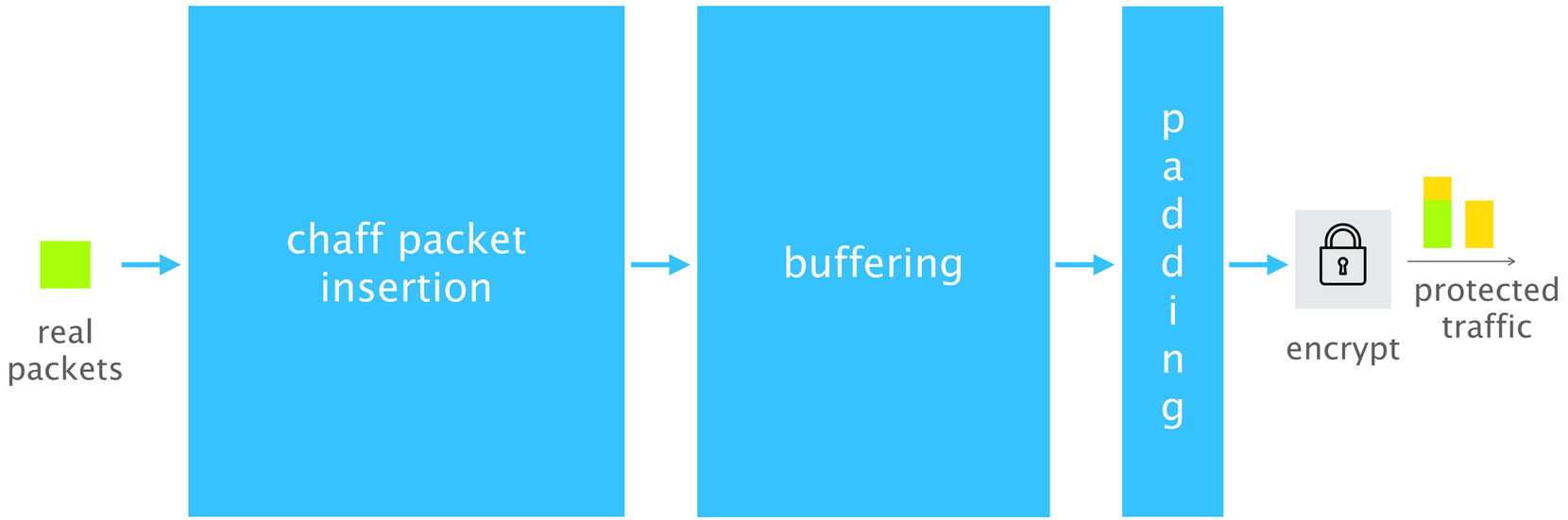


Computing efficient traffic patterns

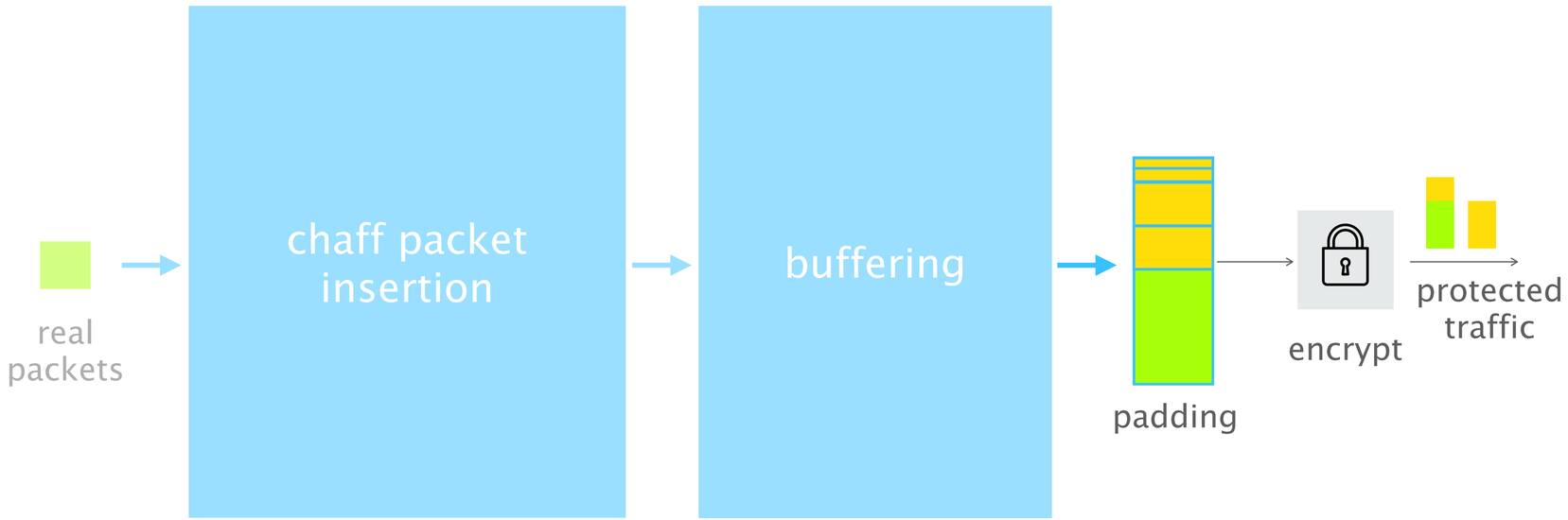
Traffic shaping in the data plane

Experimental results

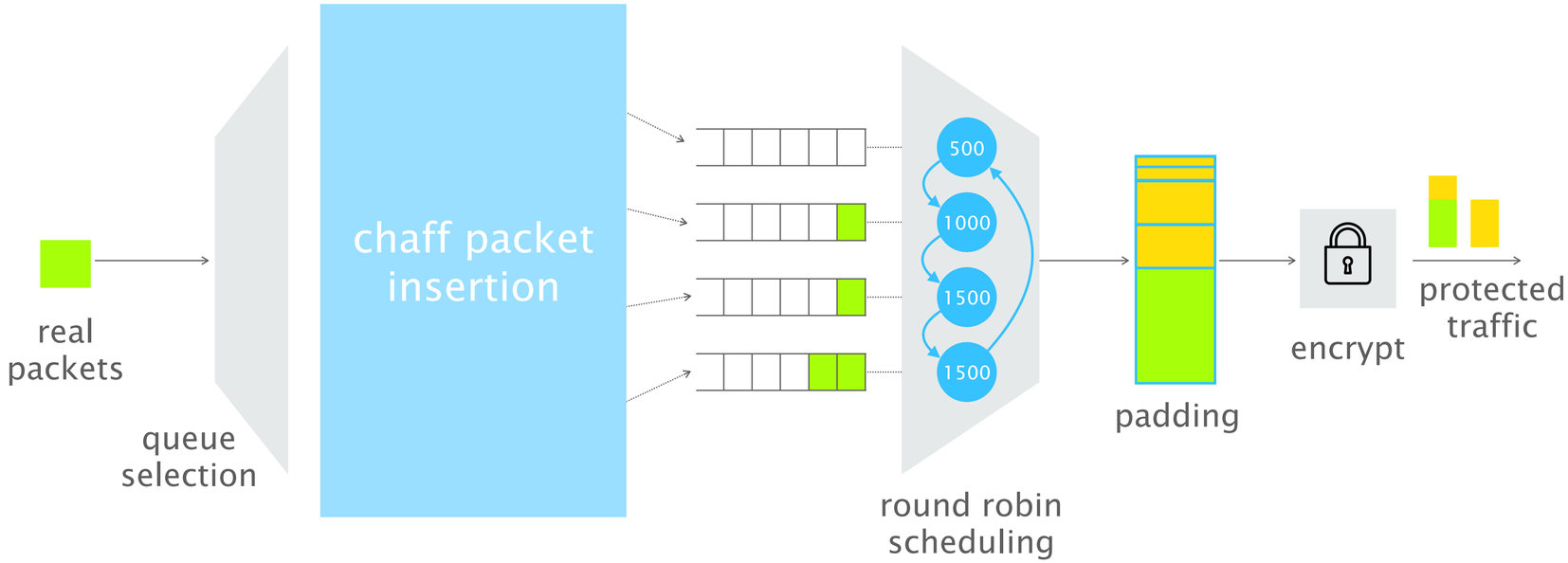
At a high level,
ditto consists of 3 building blocks



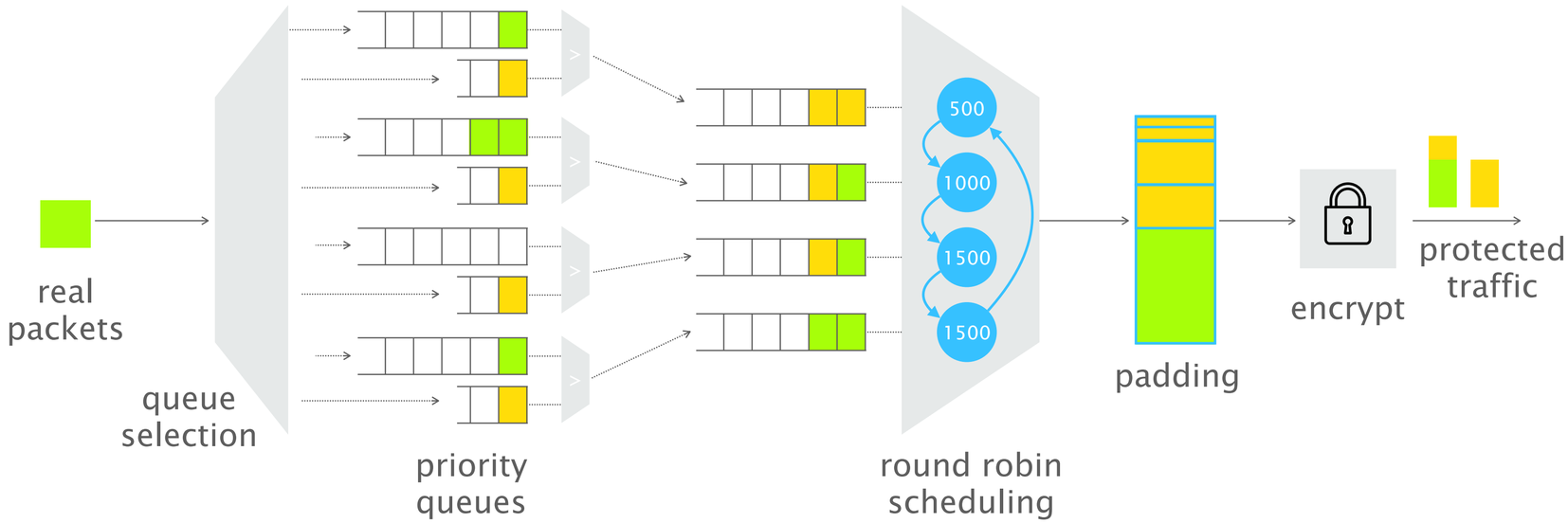
ditto pads packets by adding custom headers



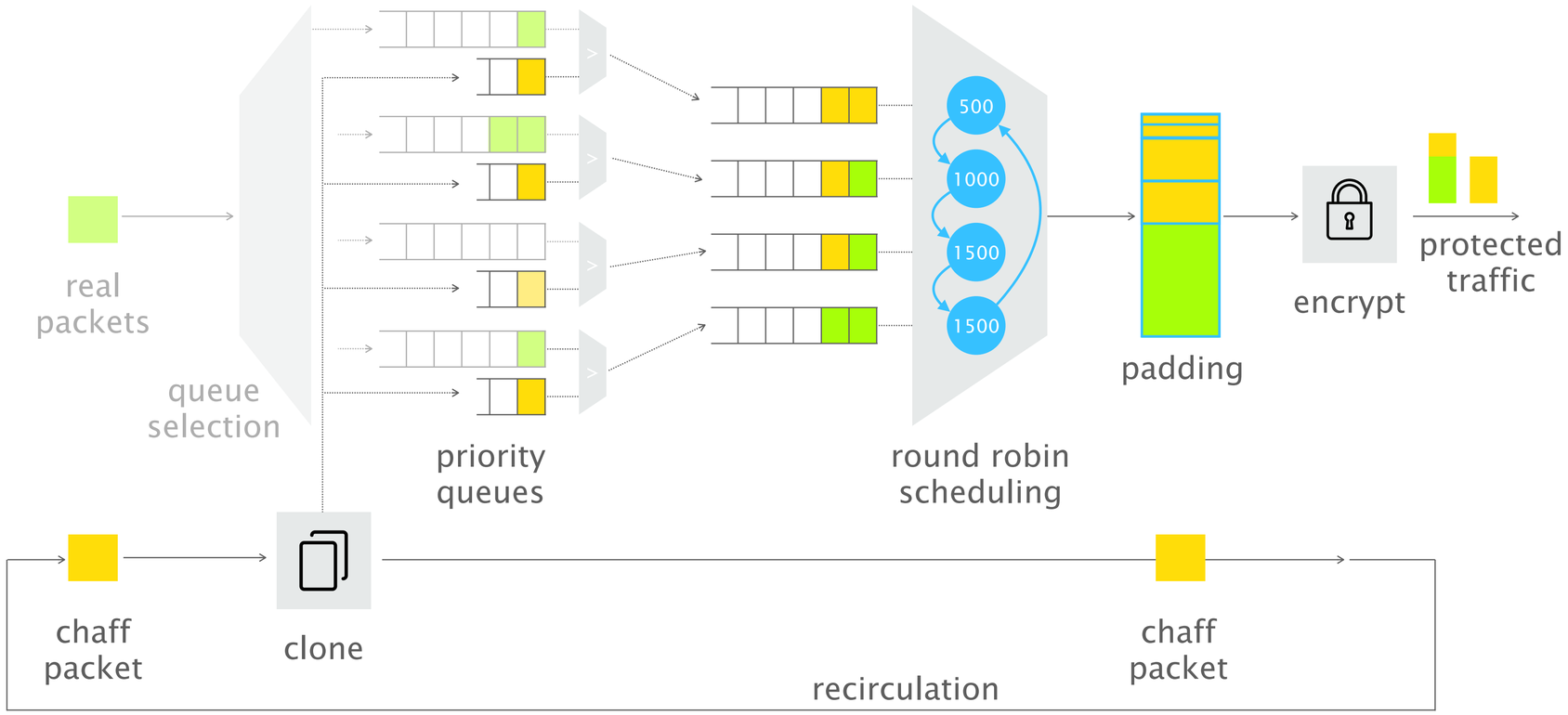
ditto uses round-robin scheduling to enforce the pattern



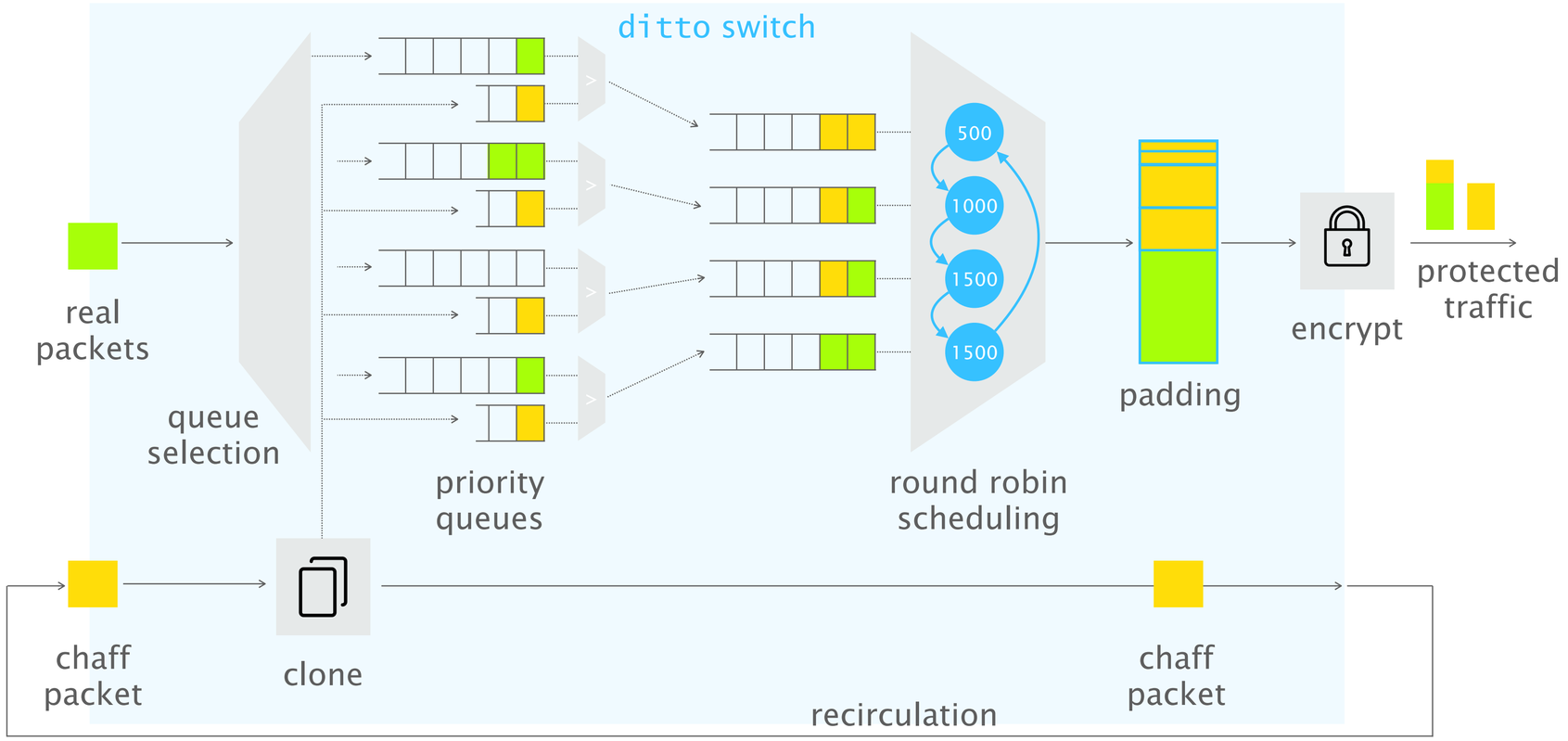
ditto uses priority queues to mix real and chaff packets



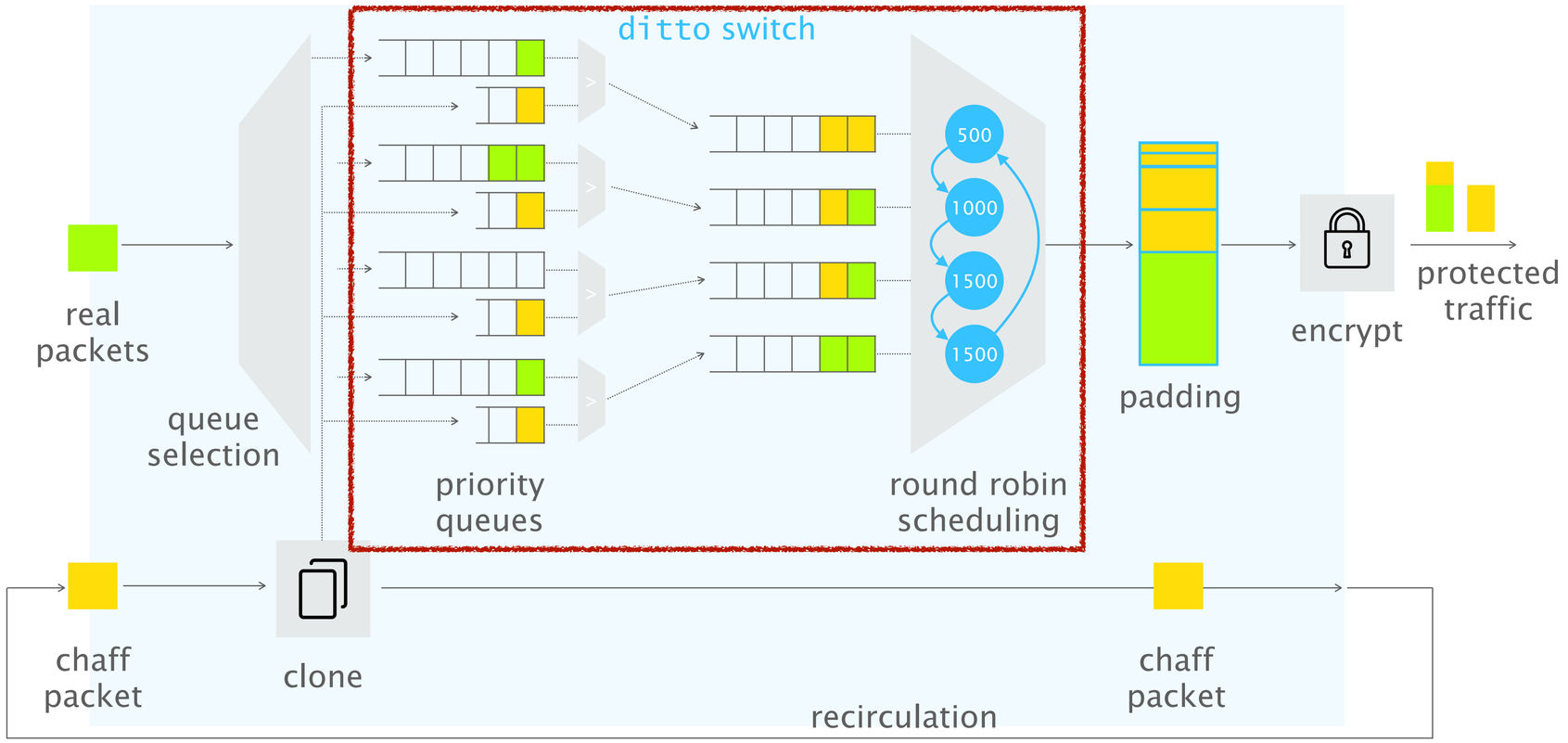
ditto generates chaff packets by recirculating and cloning them

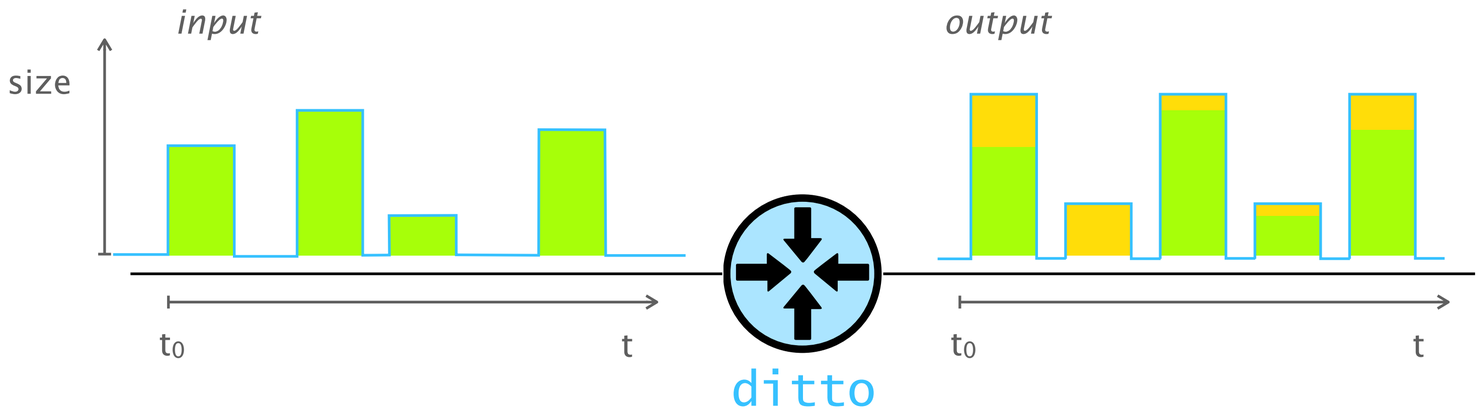


ditto runs entirely in the data plane of programmable switches



Current switches do not support 2-level queueing — the paper explains how we solved it





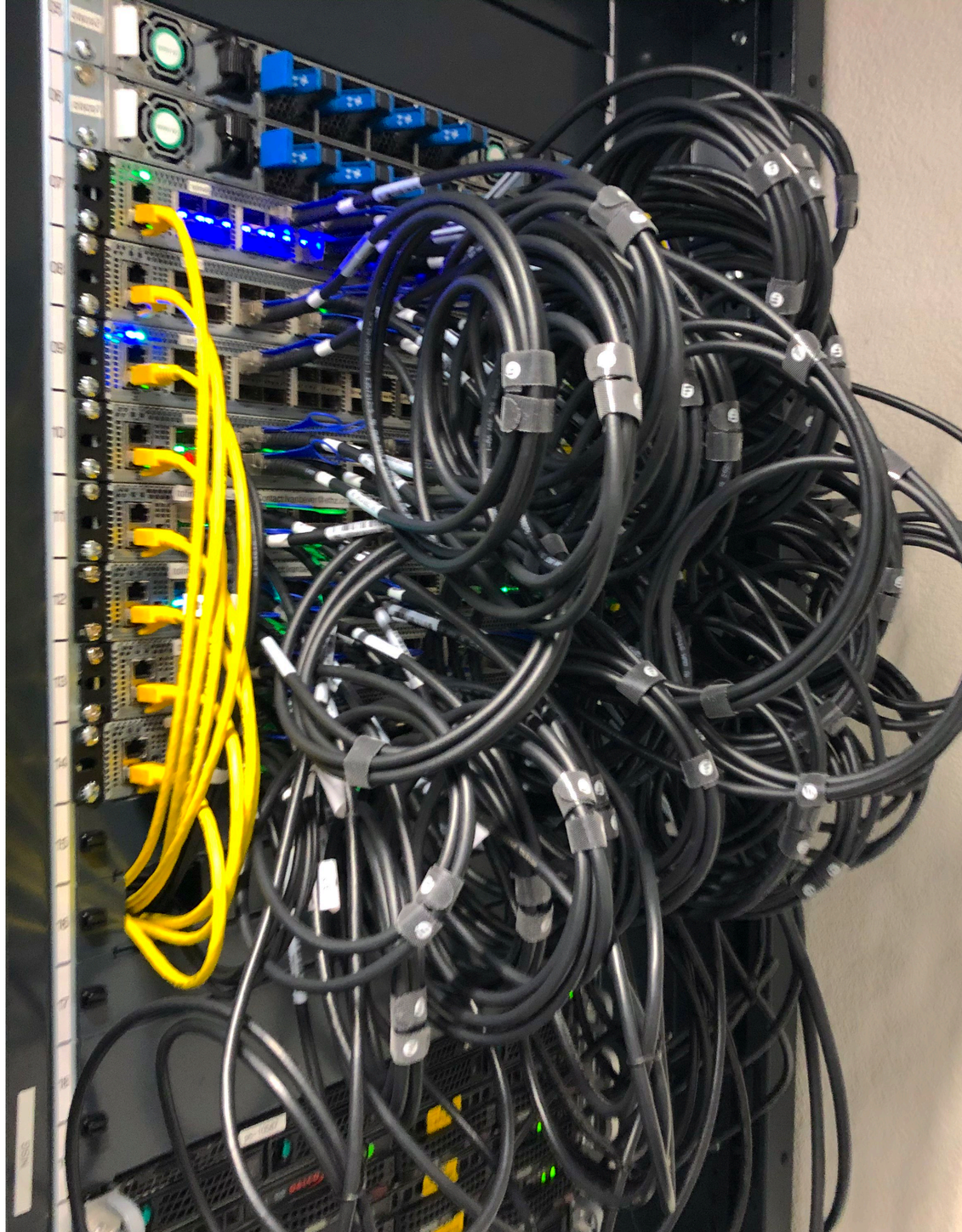
Computing efficient traffic patterns

Traffic shaping in the data plane

Experimental results

Our evaluation shows that ditto performs well and is secure

- Experiments on hardware
  Intel Tofino switches
- Simulations in software
  to show potential of future hardware



Our evaluation shows that ditto performs well and is secure

- Hardware
  - Patterns reduce overhead
  - Up to 80 Gbps throughput (100 Gbps link)
  - No performance loss for interactive applications
  - ditto is secure
- Simulations
  - The same pattern can be used for months
  - Up to 99% efficiency with 1MB buffer space
  - Less than 8% of the packets are reordered

NetHide: Secure and Practical Network Topology Obfuscation

Roland Meier[1], Petar Tsankov[1], Vincent Lenders[2], Laurent Vanbever[1], Martin Vechev[1]

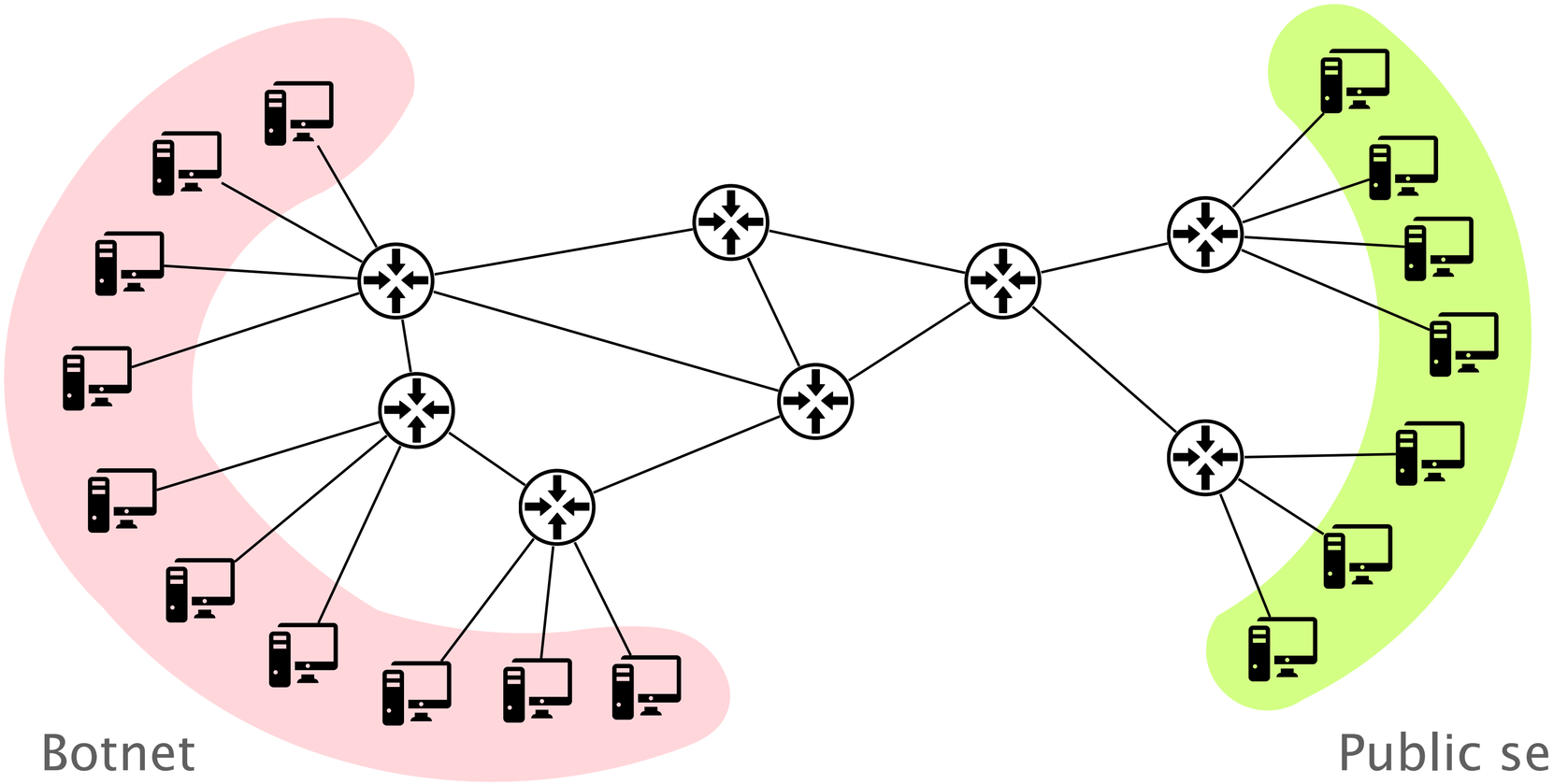
nethide.ethz.ch

USENIX Security 2018

(1) ETH zürich

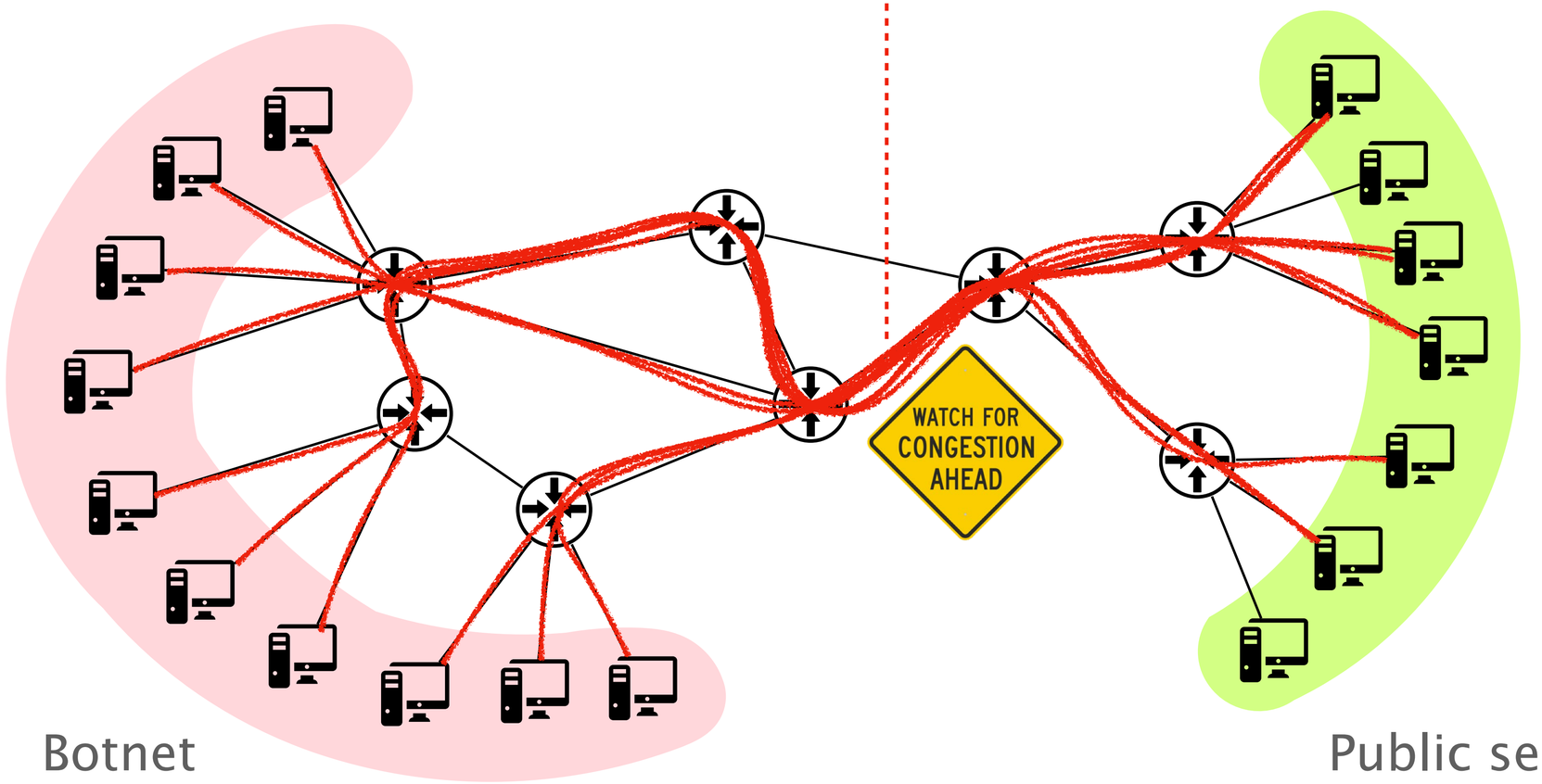
(2) Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse


Botnet
Public servers

Link-flooding attacks (LFAs)
target the infrastructure

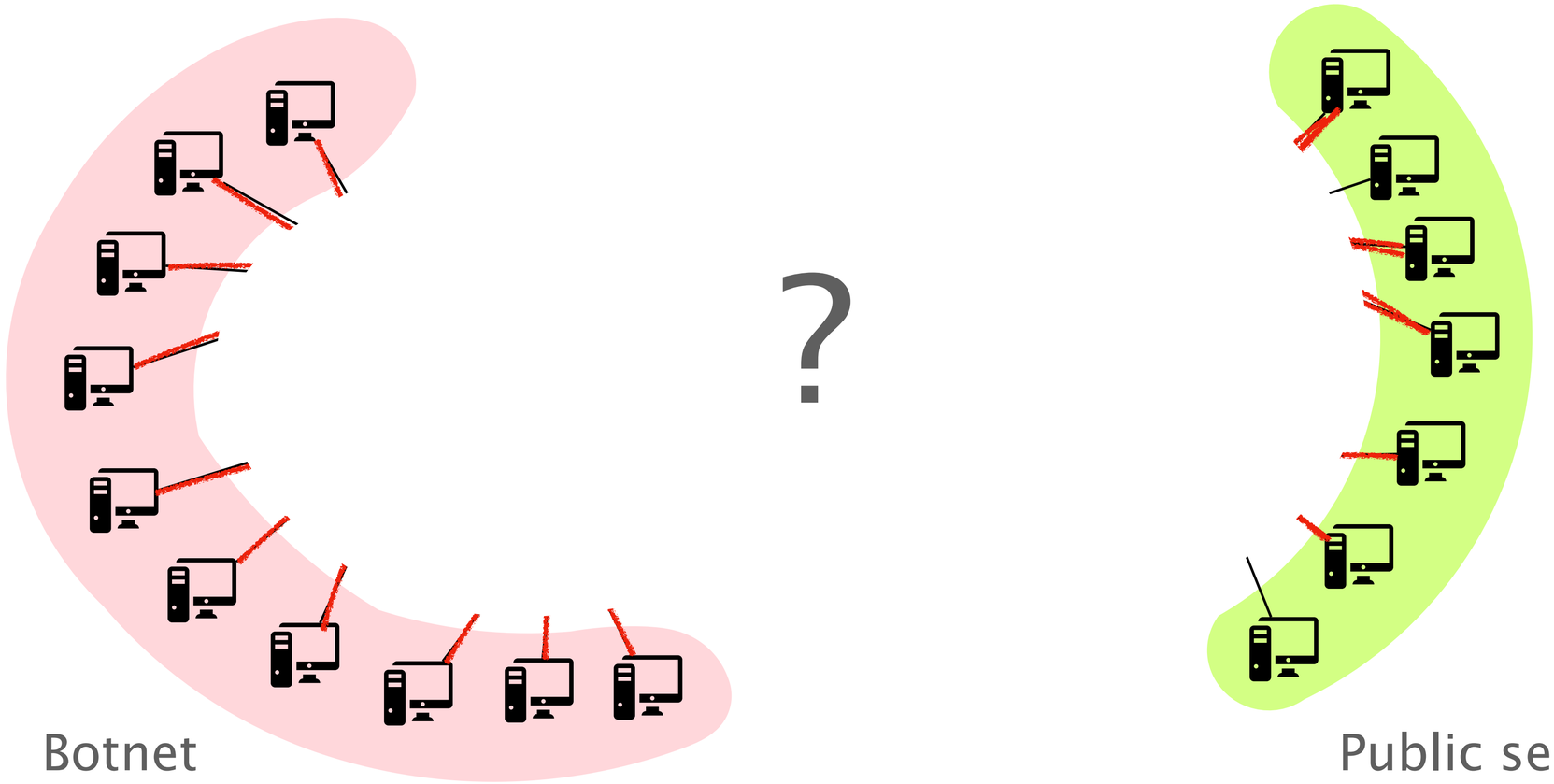
Low-rate, legitimate flows
spread over many endpoints

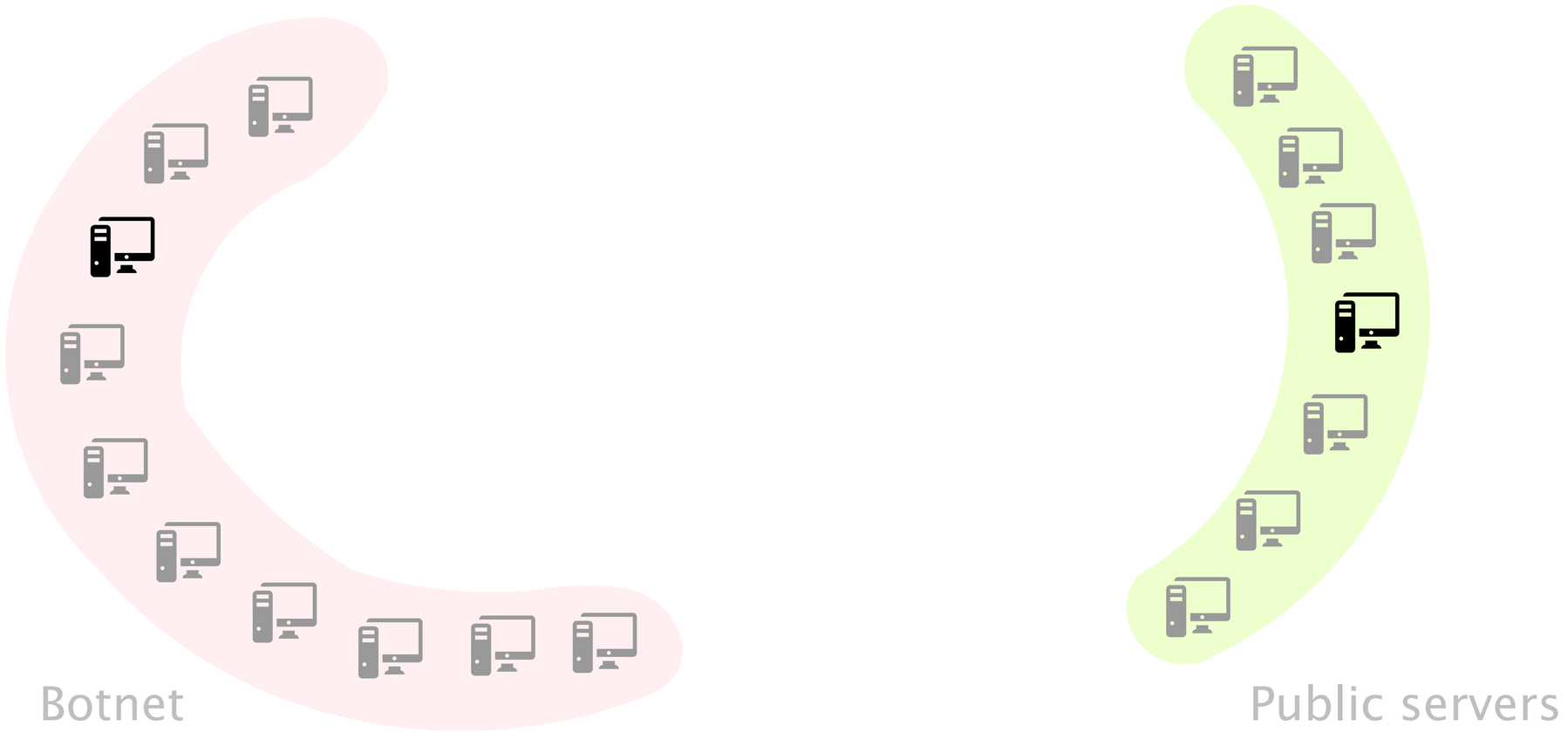


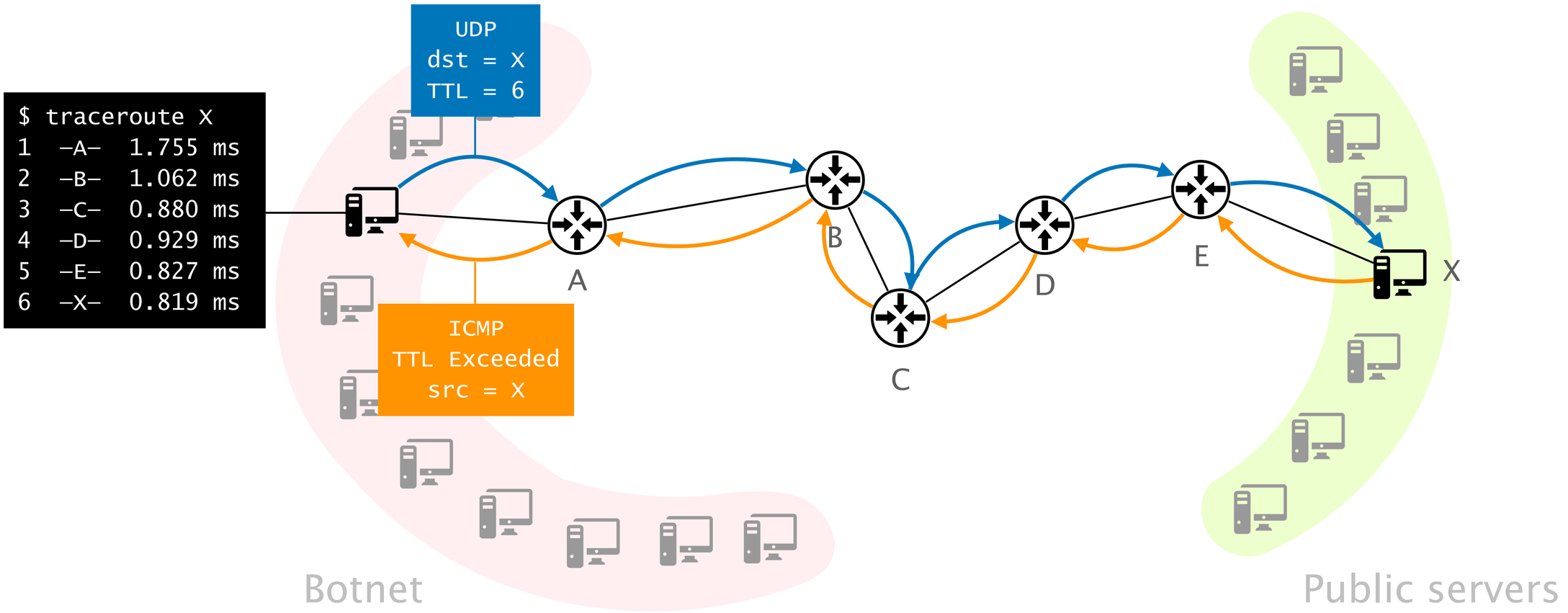
Botnet

Public servers

Link-flooding attacks (LFAs)
require knowing the topology




Botnet
Public servers


$ traceroute X
1 –A– 1.755 ms
2 –B– 1.062 ms
3 –C– 0.880 ms
4 –D– 0.929 ms
5 –E– 0.827 ms
6 –X– 0.819 ms
UDP
dst = X
TTL = 6
ICMP
TTL Exceeded
src = X
A
B
C
D
E
X
Botnet
Public servers

Learning large topologies by combining many path measurements

Measuring ISP Topologies with Rocketfuel

Neil Spring Ratul Mahajan David Wetherall

{nspring,ratul,djw}@cs.washington.edu
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350

ABSTRACT

To date, realistic ISP topologies have not been accessible to the research community, leaving work that depends on topology on an uncertain footing. In this paper, we present new Internet mapping techniques that have enabled us to directly measure router-level ISP topologies. Our techniques reduce the number of required traces compared to a brute-force, all-to-all approach by three orders of magnitude without a significant loss in accuracy. They include the use of BGP routing tables to focus the measurements, exploiting properties of IP routing to eliminate redundant measurements, better alias resolution, and the use of DNS to divide each map into POPs and backbone. We collect maps from ten diverse ISPs using our techniques, and find that our maps are substantially more complete than those of earlier Internet mapping efforts. We also report on properties of these maps, including the size of POPs, distribution of router outdegree, and the inter-domain peering structure. As part of this work, we release our maps to the community.

Categories and Subject Descriptors

C.2.1 [**Communication Networks**]: Architecture and Design—*topology*

General Terms

Measurement

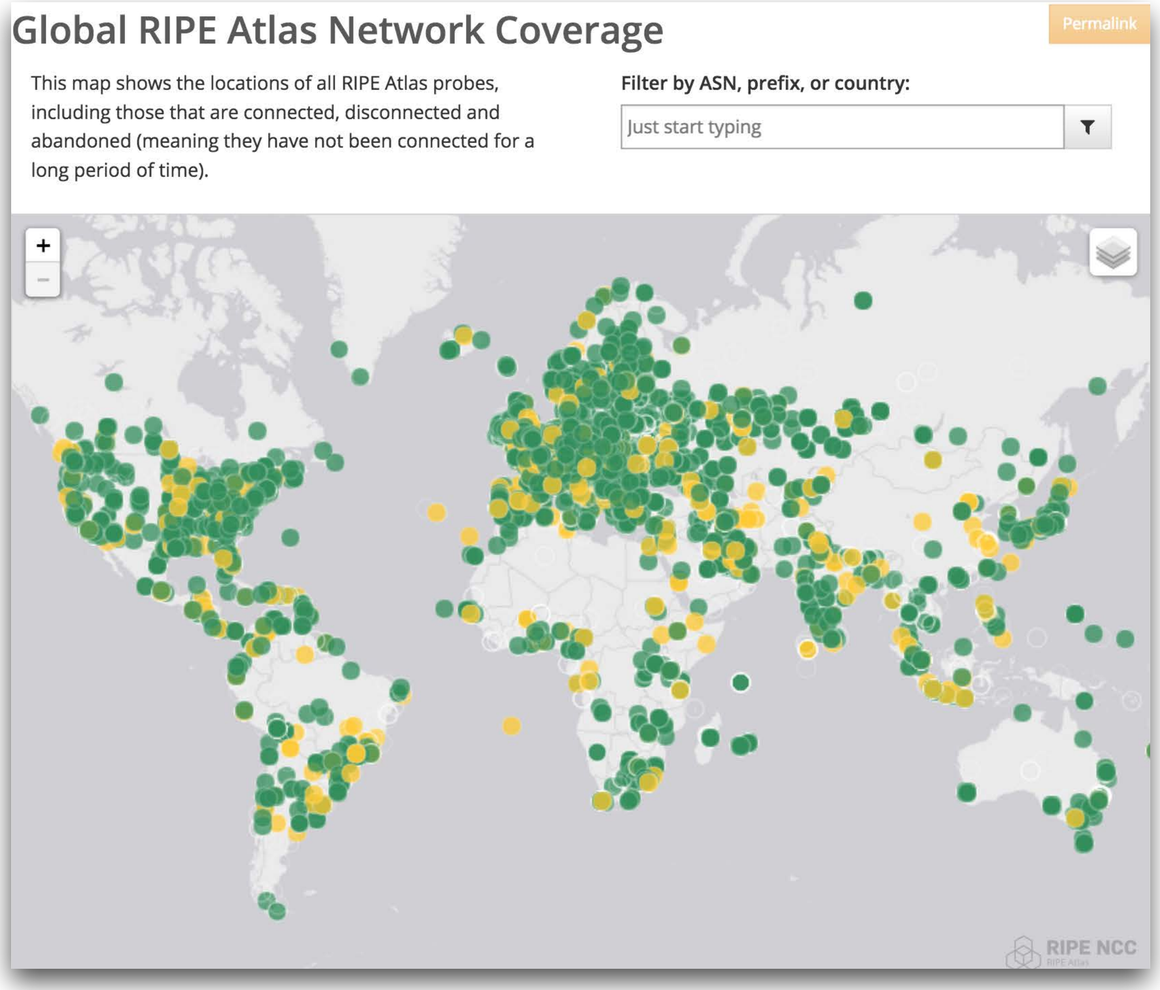
1. INTRODUCTION

Realistic Internet topologies are of considerable importance to network researchers. Topology influences the dynamics of routing protocols [2, 10], the scalability of multicast [17], the efficacy of proposals for denial-of-service tracing and response [16, 11, 21, 22], and other aspects of protocol performance [18]. topologies generated by tools such as GT-ITM [26] or Brite [12] are representative [25].

The main contribution of this paper is to present new measurement techniques to infer high quality ISP maps while using as few measurements as possible. Our insight is that routing information can be exploited to select the measurements that are most valuable. One technique, *directed probing*, uses BGP routing information to choose only those traceroutes that are likely to transit the ISP being mapped. A second technique, *path reductions*, suppresses traceroutes that are likely to follow redundant paths through the ISP network. These two techniques reduce the number of traces required to map an ISP by three orders of magnitude compared to a brute-force, all-to-all approach, and we show that the savings do not come at a high cost in terms of accuracy. We also describe a new solution to the *alias resolution* problem of clustering the interface IP addresses listed in a traceroute into their corresponding routers. Our new, pair-wise alias resolution procedure finds three times as many aliases as prior techniques. Additionally, we use DNS information to break the ISP maps into backbone and POP components, complete with their geographical location.

We used our techniques to map ten diverse ISPs – Abovenet, AT&T, Ebone, Exodus, Level3, Sprint, Telstra, Tiscali (Europe), Verio, and VSNL (India) – by using over 750 publicly available traceroute sources as measurement vantage points. These maps are summarized in the paper.

Three ISPs, out of the ten we measured, helped to validate our maps. We also estimate the completeness of our maps by scanning ISP IP address ranges for routers that we might have missed, and by comparing the peering links we find with those present in BGP routing tables. Our maps reveal more complete ISP topologies compared to earlier efforts; we find roughly seven times more routers and links in our area of focus than Skitter [6].



So the solution is to hide the topology?

yes, but…

traceroute from XO network?

Traceroute from within Colombia?

Bill Woodcock woody at

Cloudflare 1.1.1.1 public DNS broken w/ AT&T CPE

Paul Rolland (=?UTF-8?B?44Od44O844Or44O744Ot44Op44Oz?=) rol at witbe.net
Tue Apr 3 06:22:04 UTC 2018

- Previous message (by threa
- Next message (by thread): (
- **Messages sorted by:** [date

Hello,

On Mon, 2 Apr 2018 16:26:13
Marty Strong via NANOG <nanc

> *So far we know about a few*
>
> *- Pace 5268*
> *- Calix GigaCenter*
> *- Various Cisco Wifi acces*
>
> *If you know of others plea*

It seems that in France, Ora
way...

```
215 [6:20] rol at riri:~> tr
traceroute to 1.1.1.1 (1.1.1
 1  * * *
 2  * * *
```

Can anyone check this routing agains
WI?

Shawn L shawnl at up.net
Sat Jun 14 17:04:58 UTC 2014

- Previous message: Can anyone check this routing against Charter in WI?
- Next message: Can anyone check this routing against Charter in WI?
- **Messages sorted by:** [date] [thread] [subject] [author]

It seems ok from here

```
traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets

 4  dtr02rhnlwi-bue-1.rhnl.wi.charter.com (96.34.16.250)  20.843 r
ms  21.616 ms
 5  crr02euclwi-bue-7.eucl.wi.charter.com (96.34.17.32)  27.662 ms
ms  35.623 ms
 6  bbr02euclwi-bue-4.eucl.wi.charter.com (96.34.2.6)  29.236 ms
ms  22.945 ms
```

Has Level3 done away with traceroute??

Mel Beckman mel at beckman.org
Thu Sep 21 17:57:51 CST 2017

- Previous message (by thread): Has Level3 done away with traceroute??
- Next message (by thread): Bell Canada (AS 577) and NTT (AS 2914) routing
- **Messages sorted by:** [date] [thread] [subject] [author]

I am able to traceroute in and out of Level3, but it seems like some L3 internal hops are missing, as I appear to go straight from the L3 edge router to my L3 customer agg router:

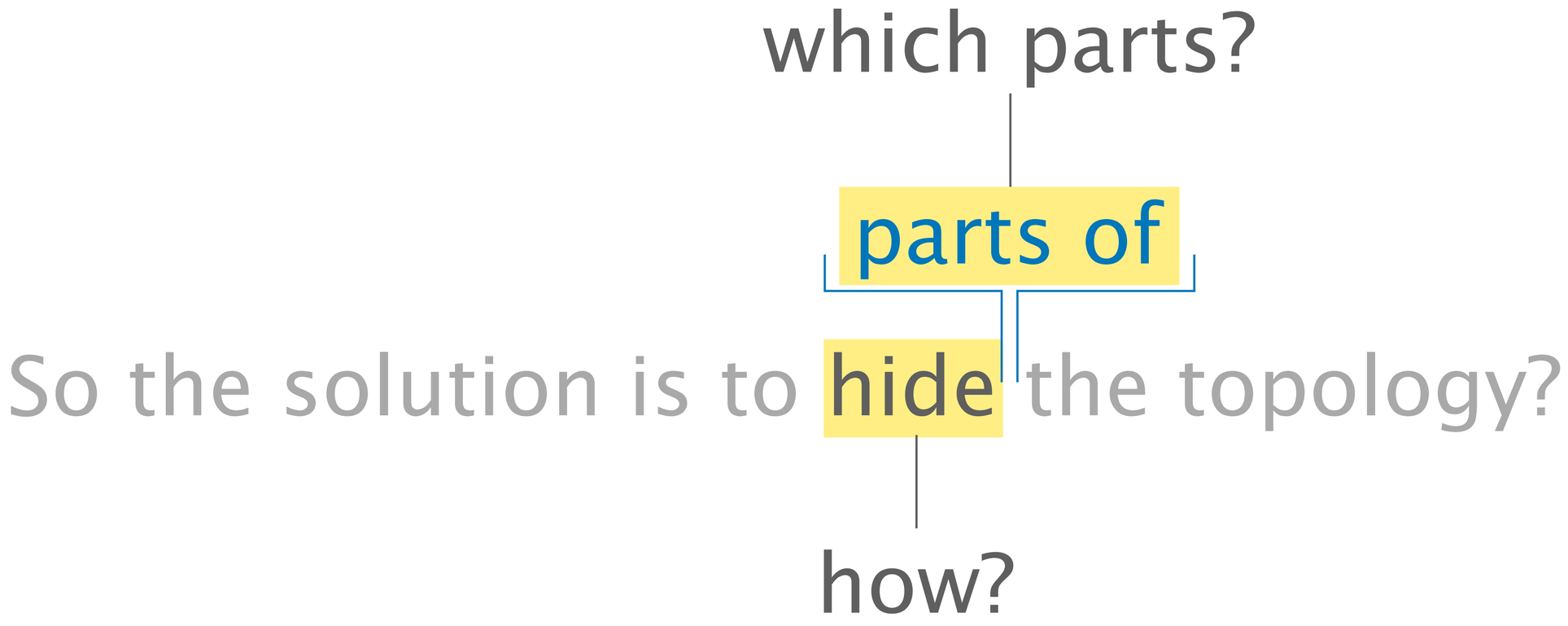
```
traceroute to 206.83.0.42 (206.83.0.42), 64 hops max, 52 byte packets
 1  47.155.227.1 (47.155.227.1)  4.318 ms  4.661 ms  4.715 ms
 2  172.102.107.166 (172.102.107.166)  10.202 ms  7.521 ms
    172.102.102.240 (172.102.102.240)  7.240 ms
 3  ae8---0.scr02.lsan.ca.frontiernet.net (74.40.3.49)  9.609 ms  7.501 ms
    ae8---0.scr01.lsan.ca.frontiernet.net (74.40.3.37)  5.298 ms
 4  ae0---0.cbr01.lsan.ca.frontiernet.net (74.40.3.198)  7.169 ms  7.015 ms  7.277 ms
 5  * * *
 6  ae-4-90.edge1.losangeles9.level3.net (4.69.144.202)  8.384 ms  7.012 ms
    ae-2-70.edge1.losangeles9.level3.net (4.69.144.74)  9.865 ms
 7  ae-2-70.edge1.losangeles9.level3.net (4.69.144.74)  7.471 ms
    ae-3-80.edge1.losangeles9.level3.net (4.69.144.138)  6.551 ms
    ae-2-70.edge1.losangeles9.level3.net (4.69.144.74)  10.520 ms
 8  4.68.111.22 (4.68.111.22)  9.010 ms  7.445 ms  7.314 ms
 9  sba1-ar1-xe-11-0-0-0.us.twtelecom.net (35.248.2.6)  9.788 ms  9.536 ms  10.266 ms
10  206-190-77-10.static.twtelecom.net (206.190.77.10)  14.739 ms  12.525 ms  9.979 ms
11  iris1.jet.net (206.83.0.42)  10.285 ms  9.880 ms  10.063 ms

traceroute to 47.155.227.1 (47.155.227.1), 64 hops max, 40 byte packets
```

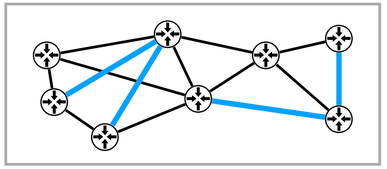
traceroute is an essential debugging tool

parts of

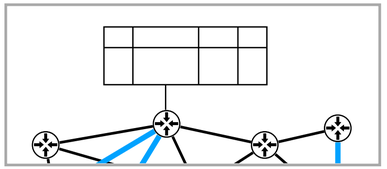
So the solution is to hide the topology?


which parts?
parts of
So the solution is to hide the topology?
how?

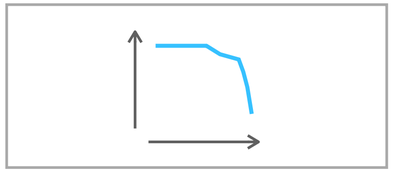
NetHide: Secure and Practical Network Topology Obfuscation



NetHide computes a secure virtual topology that is similar to the physical topology

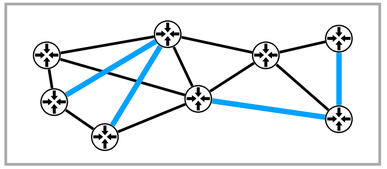


NetHide deploys the virtual topology using programmable networks

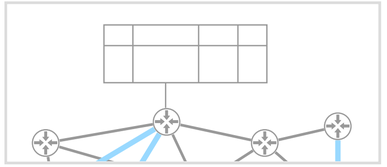


NetHide works for realistic topologies and maintains the utility of debugging tools

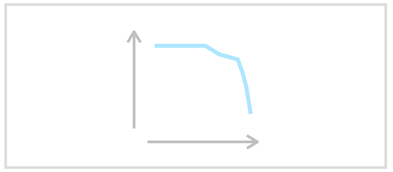
NetHide: Secure and Practical Network Topology Obfuscation



NetHide computes a secure virtual topology that is similar to the physical topology



NetHide deploys the virtual topology using programmable networks



NetHide works for realistic topologies and maintains the utility of debugging tools

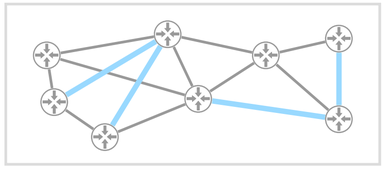
Topology obfuscation
as an optimization problem

Given the **physical topology P**,

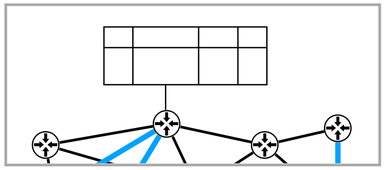
compute a **virtual topology V**, such that

- **V** is robust against link-flooding attacks
- **V** has maximal practicality

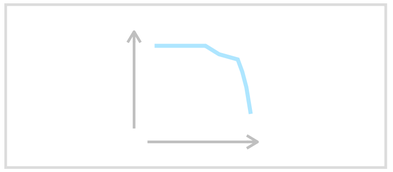
NetHide: Secure and Practical Network Topology Obfuscation



NetHide computes a secure virtual topology that is similar to the physical topology



NetHide deploys the virtual topology using programmable networks



NetHide works for realistic topologies and maintains the utility of debugging tools

Utility-preserving topology deployment

Deploy the **virtual topology V**, such that

- debugging tools still work
- network performance is not impacted
- it scales to large networks

Utility-preserving topology deployment

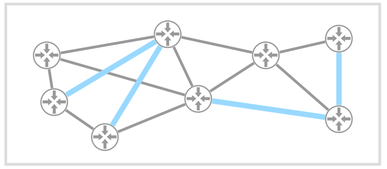
Deploy the **virtual topology V**, such that

- debugging tools still work
- network performance is not impacted
- it scales to large networks

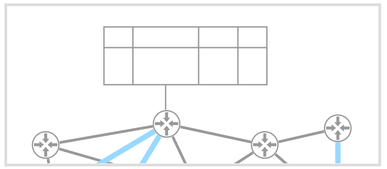
Maintaining the utility of debugging tools requires sending packets through the actual network

- ⛔ Answer from a central controller
- ⛔ Answer at the edge
- ⛔ Answer in a virtual clone of the network
- ➡ Answer from the correct device that appears on the path

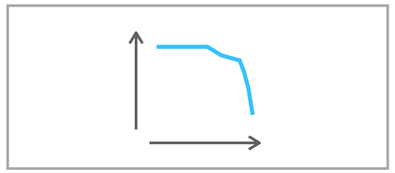
NetHide: Secure and Practical Network Topology Obfuscation



NetHide computes a secure virtual topology that is similar to the physical topology

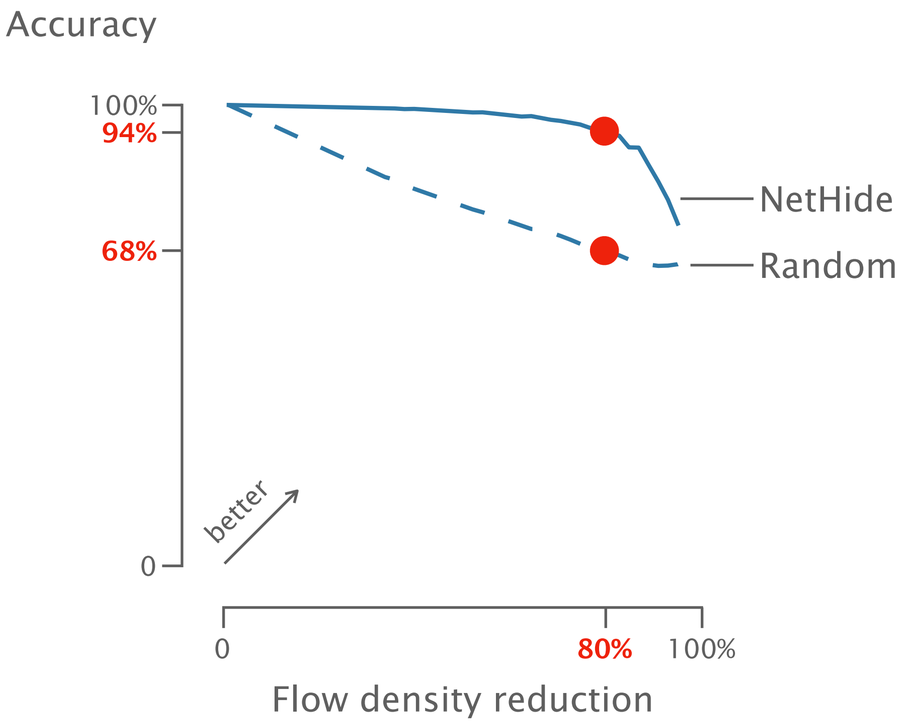


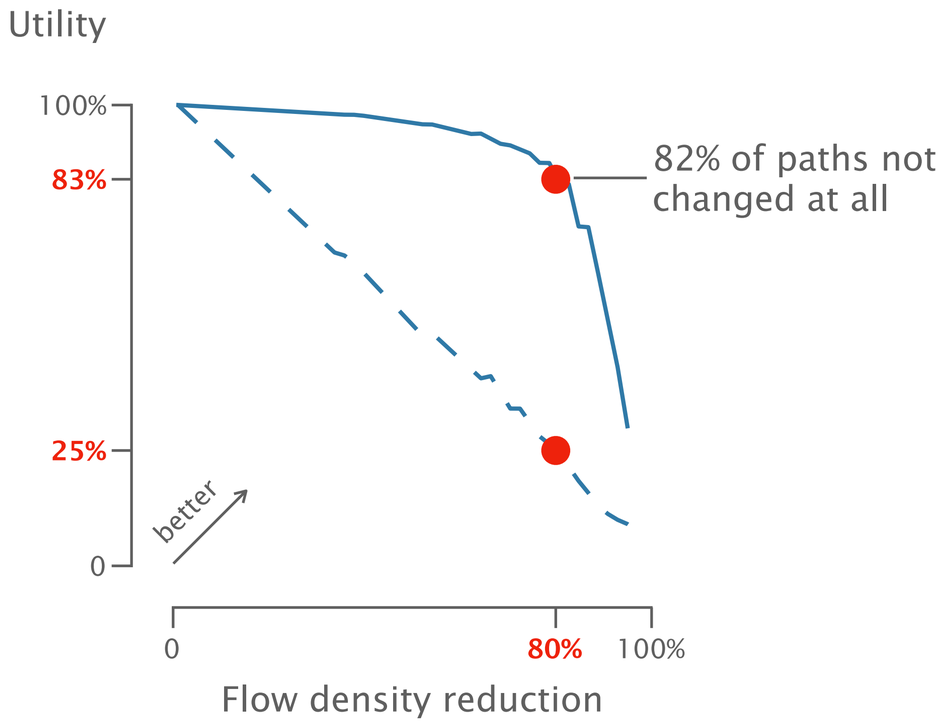
NetHide deploys the virtual topology using programmable networks



NetHide works for realistic topologies and maintains the utility of debugging tools

High protection
with small impact on accuracy and utility





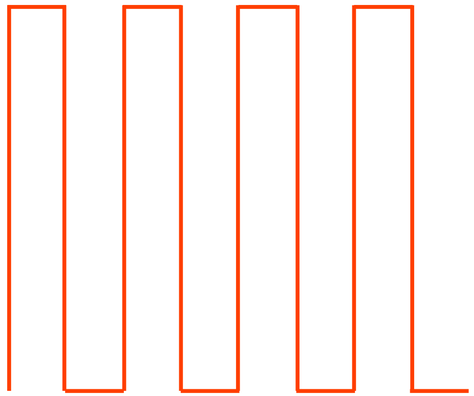
Protecting users
with programmable hardware

Proactively

Reactively

neutralizing next-gen DDoS attacks
without impacting production traffic

Aggregate-Based Congestion Control for Pulse-Wave DDoS Defense



Albert Gran Alcoz [*]
Martin Strohmeier [◇]
Vincent Lenders [◇]
Laurent Vanbever [*]

SIGCOMM
August 26 2022

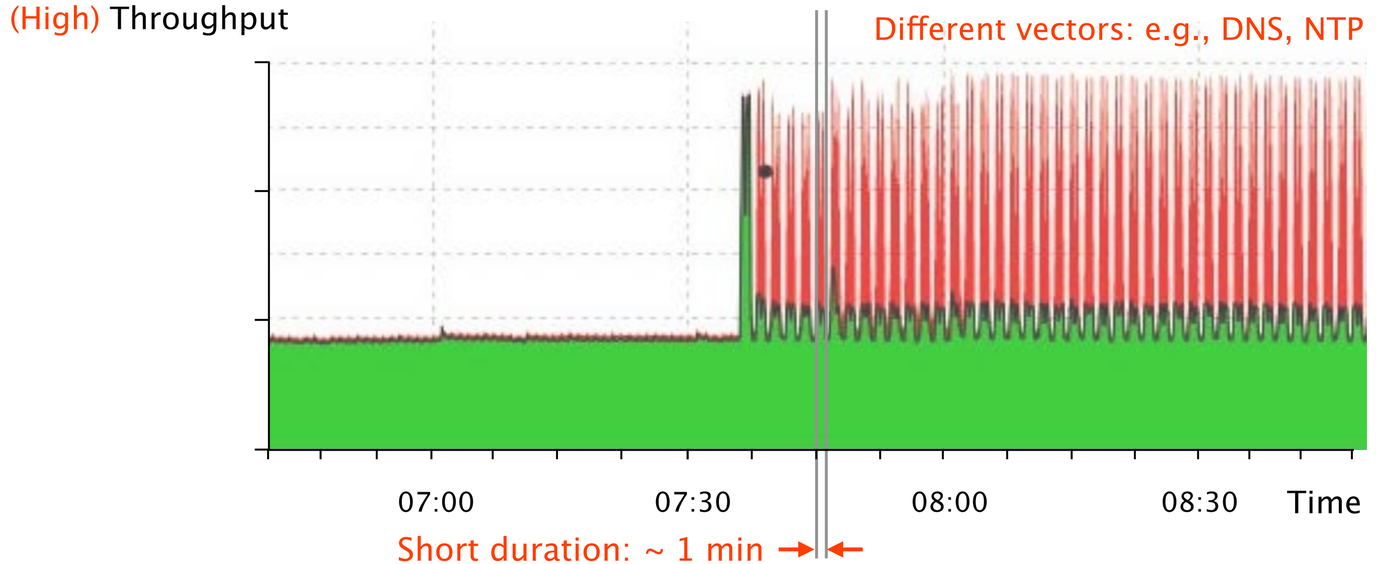
[*] ETH zürich

[◇] Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse

Don't miss Albert's talk on Friday @9am!

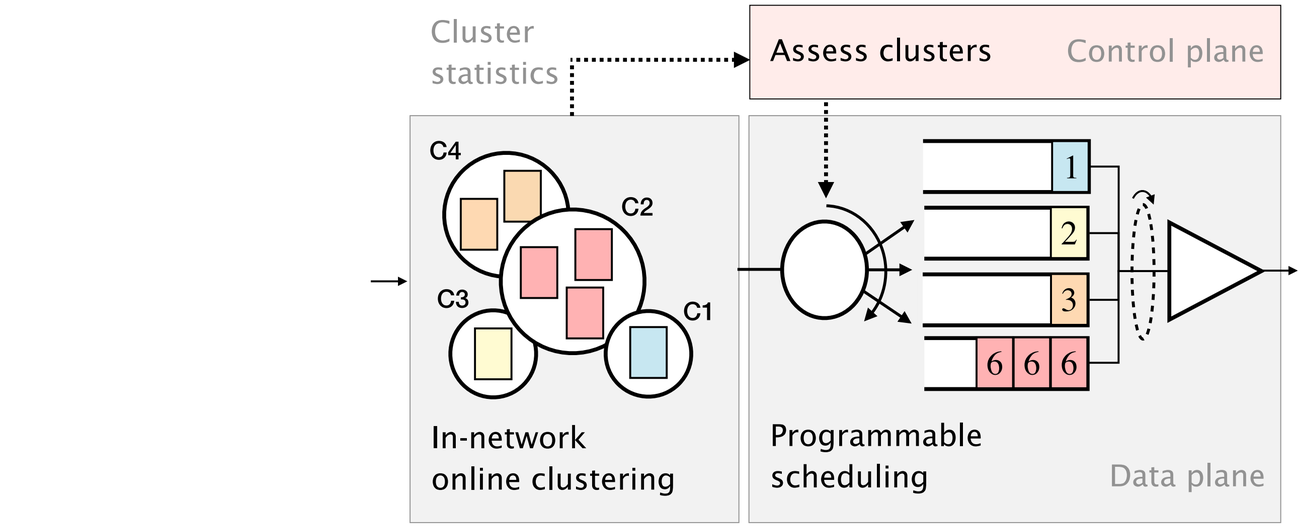
Pulse-wave DDoS attacks are composed of short-duration high-rate traffic pulses



Damian Menscher, Google 2021

Don't miss Albert's talk on Friday @9am!

ACC-Turbo combines in-network online clustering with programmable scheduling



Don't miss Albert's talk on Friday @9am!

When network programmability meets network security

Protecting users

The good

2 Being attacked

The bad

Attacking others

The ugly

(Self) Driving Under the Influence: Intoxicating Adversarial Network Inputs

Roland Meier[1], Thomas Holterbach[1], Stephan Keck[1], Matthias Stähli[1], Vincent Lenders[2], Ankit Singla[1], Laurent Vanbever[1]

ACM HotNets 2019

(1) ETH zürich

(2) Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse

Advances in network programability allow to perform many decisions in the data plane

P4: Programming Protocol-Independent Packet Processors

Pat Bosshart[†], Dan Daly[*], Glen Gibb[†], Martin Izzard[†], Nick McKeown[‡], Jennifer Rexford[**], Cole Schlesinger[**], Dan Talayco[†], Amin Vahdat[¶], George Varghese[§], David Walker[**]
[†]Barefoot Networks [*]Intel [‡]Stanford University [**]Princeton University [¶]Google [§]Microsoft Research

Blink: Fast Connectivity Recovery Entirely in the Data Plane

Thomas Holterbach,* Edgar Costa Molero,* Maria Apostolaki*
Alberto Dainotti,† Stefano Vissicchio,‡ Laurent Vanbever*

*ETH Zurich, †CAIDA / UC San Diego, ‡University College London

Hardware-Accelerated Network Control Planes

Edgar Costa Molero
ETH Zürich
cedgar@ethz.ch

Stefano Vissicchio
University College London
s.vissicchio@cs.ucl.ac.uk

Laurent Vanbever
ETH Zürich
lvanbever@ethz.ch

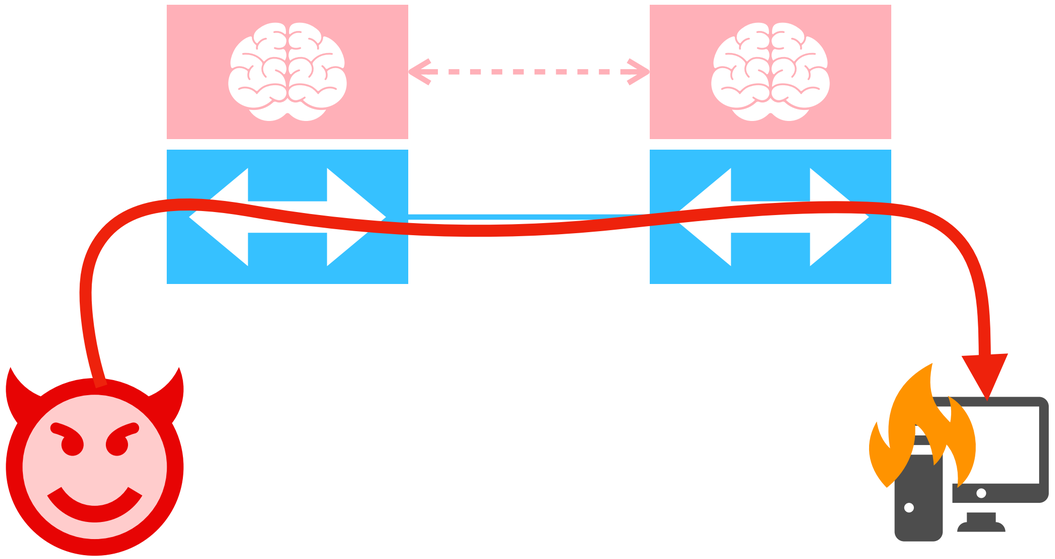
In-network Neural Networks

Giuseppe Siracusano, Roberto Bifulco
NEC Laboratories Europe

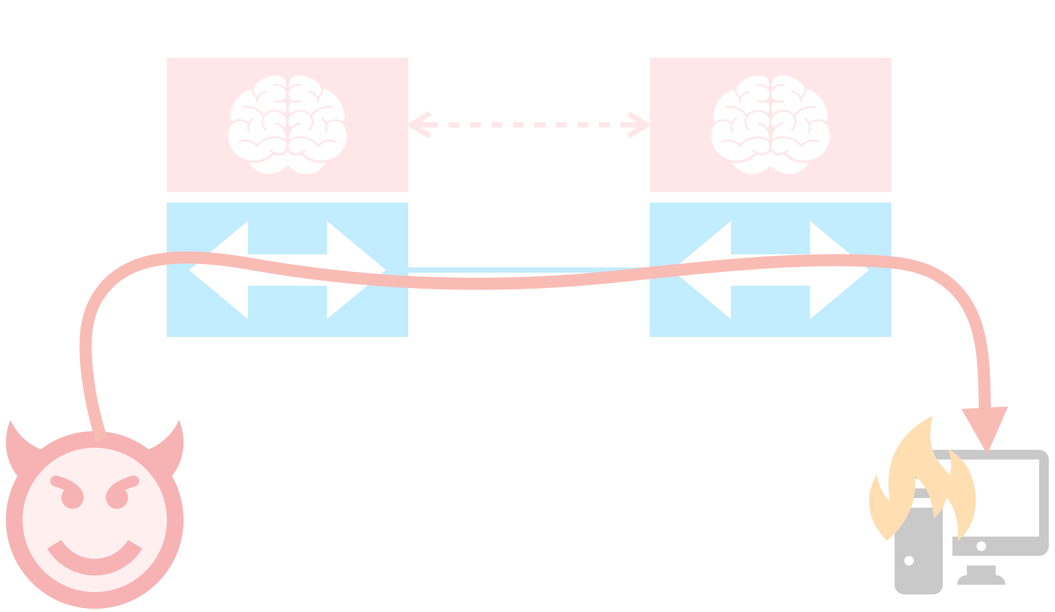
Contra: A Programmable System for Performance-aware Routing

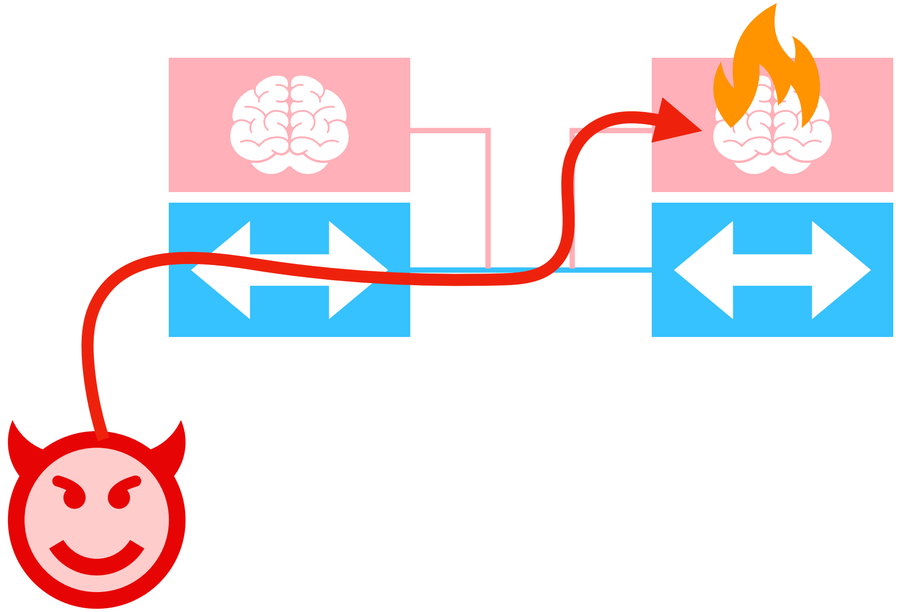
Kuo-Feng Hsu[†], Ryan Beckett[⋆], Ang Chen[†], Jennifer Rexford[‡], Praveen Tammana[‡], David Walker[‡]
[†]Rice University, [⋆]Microsoft Research, [‡]Princeton University

Traditional networks separate data and control channels



Self-driving networks merge data and control channels





In-band-signaling in the telephony system allowed “hackers” free long-distance calls

[Wikimedia]

Adversarial inputs to data-driven networks can have big consequences

- Privacy violations
  e.g., traffic hijacking
- Performance degradation
  e.g., choosing longer paths
- Reachability problems
  e.g., disconnected network
- Revenue loss
  e.g., bad QoE for clients

Advances in network programability allow to perform many decisions in the data plane

P4: Programming Protocol-Independent Packet Processors

Pat Bosshart[†], Dan Daly[*], Glen Gibb[†], Martin Izzard[†], Nick McKeown[‡], Jennifer Rexford[**], Cole Schlesinger[**], Dan Talayco[†], Amin Vahdat[¶], George Varghese[§], David Walker[**]
[†]Barefoot Networks [*]Intel [‡]Stanford University [**]Princeton University [¶]Google [§]Microsoft Research

Blink: Fast Connectivity Recovery Entirely in the Data Plane

Thomas Holterbach,* Edgar Costa Molero,* Maria Apostolaki*
Alberto Dainotti,† Stefano Vissicchio,‡ Laurent Vanbever*

*ETH Zurich, †CAIDA / UC San Diego, ‡University College London

Hardware-Accelerated Network Control Planes

Edgar Costa Molero
ETH Zürich
cedgar@ethz.ch

Stefano Vissicchio
University College London
s.vissicchio@cs.ucl.ac.uk

Laurent Vanbever
ETH Zürich
lvanbever@ethz.ch

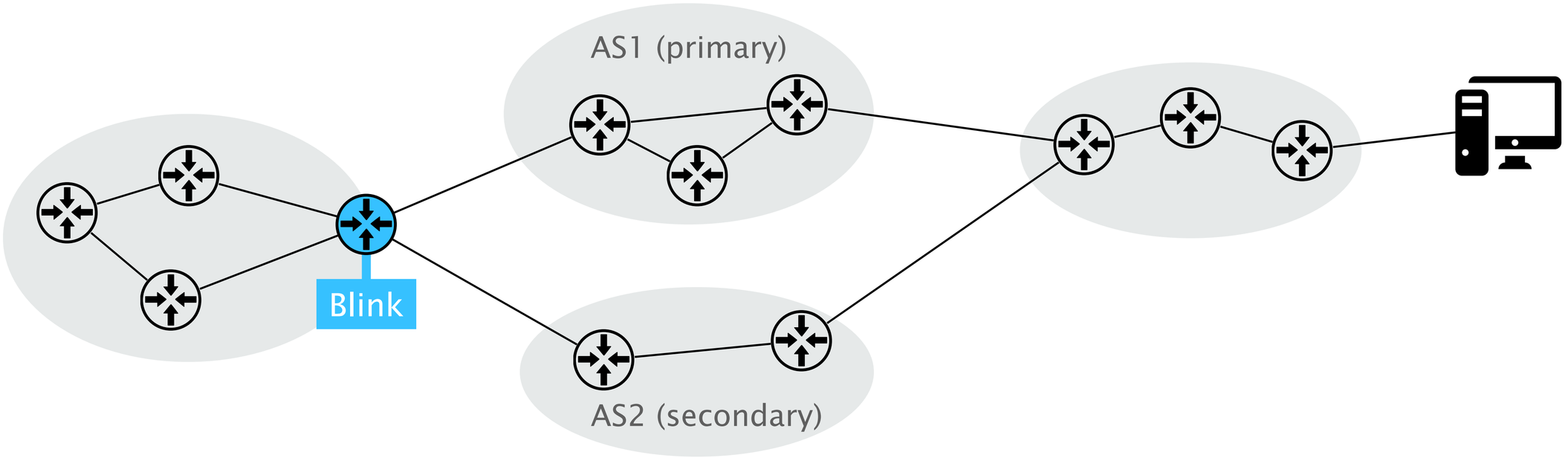
In-network Neural Networks

Giuseppe Siracusano, Roberto Bifulco
NEC Laboratories Europe

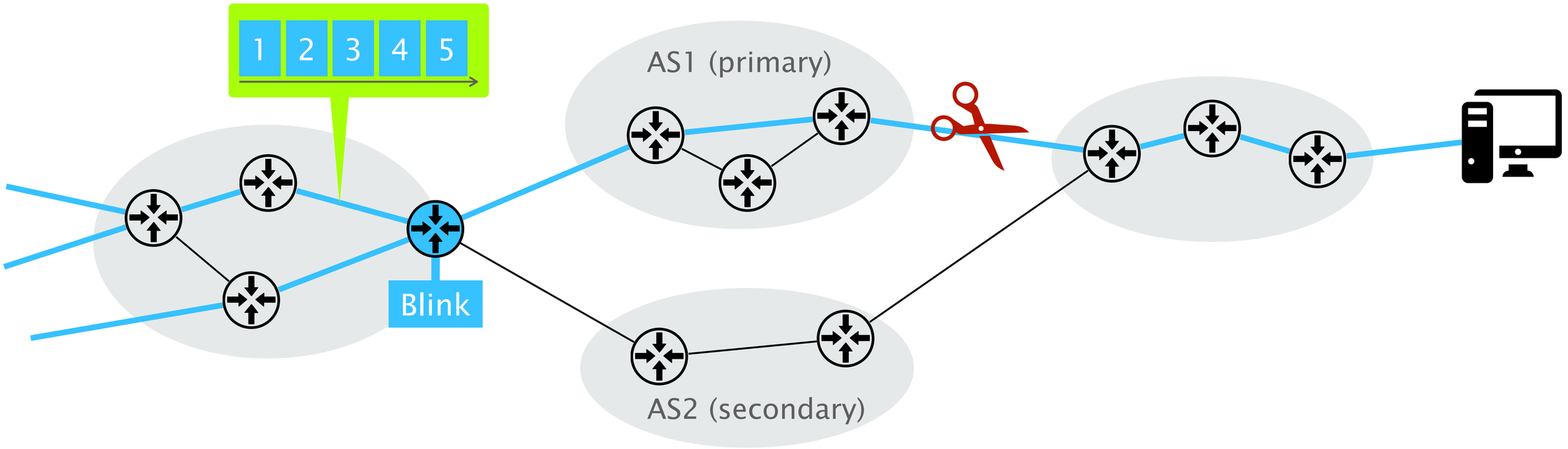
Contra: A Programmable System for Performance-aware Routing

Kuo-Feng Hsu[†], Ryan Beckett[⋆], Ang Chen[†], Jennifer Rexford[‡], Praveen Tammana[‡], David Walker[‡]
[†]Rice University, [⋆]Microsoft Research, [‡]Princeton University

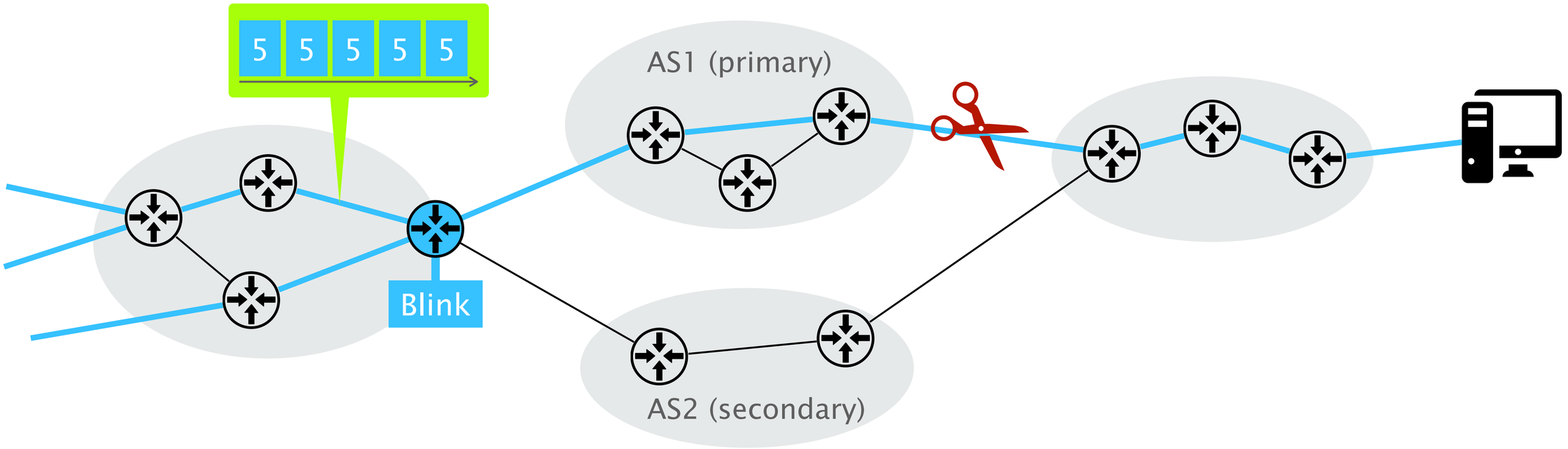
Blink monitors TCP retransmissions to detect failed paths



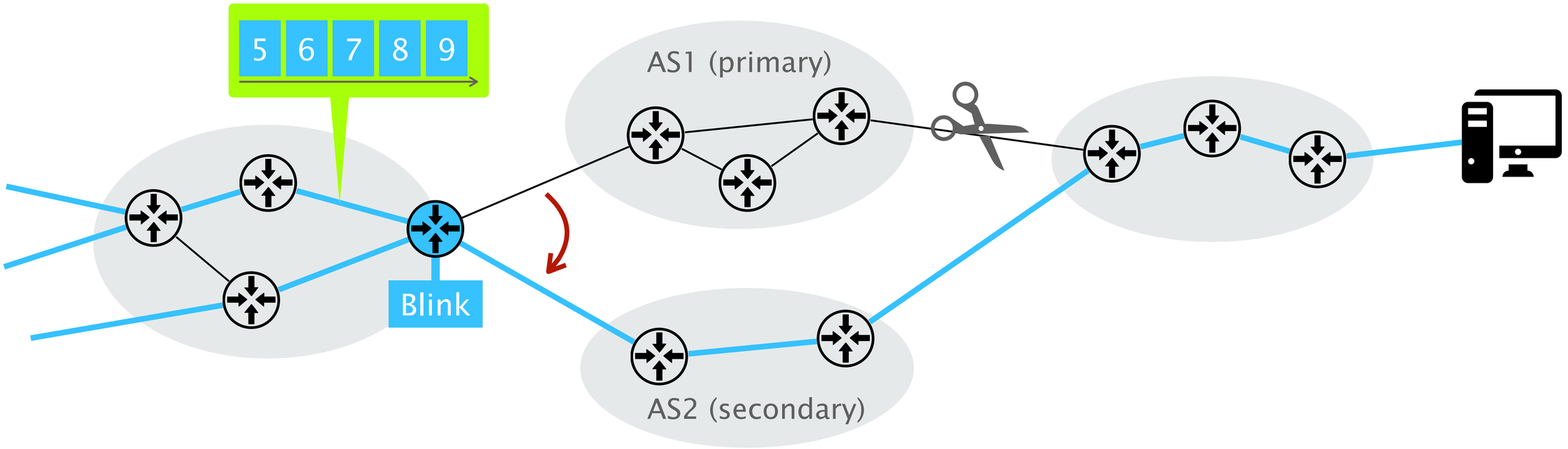
Blink monitors TCP retransmissions to detect failed paths



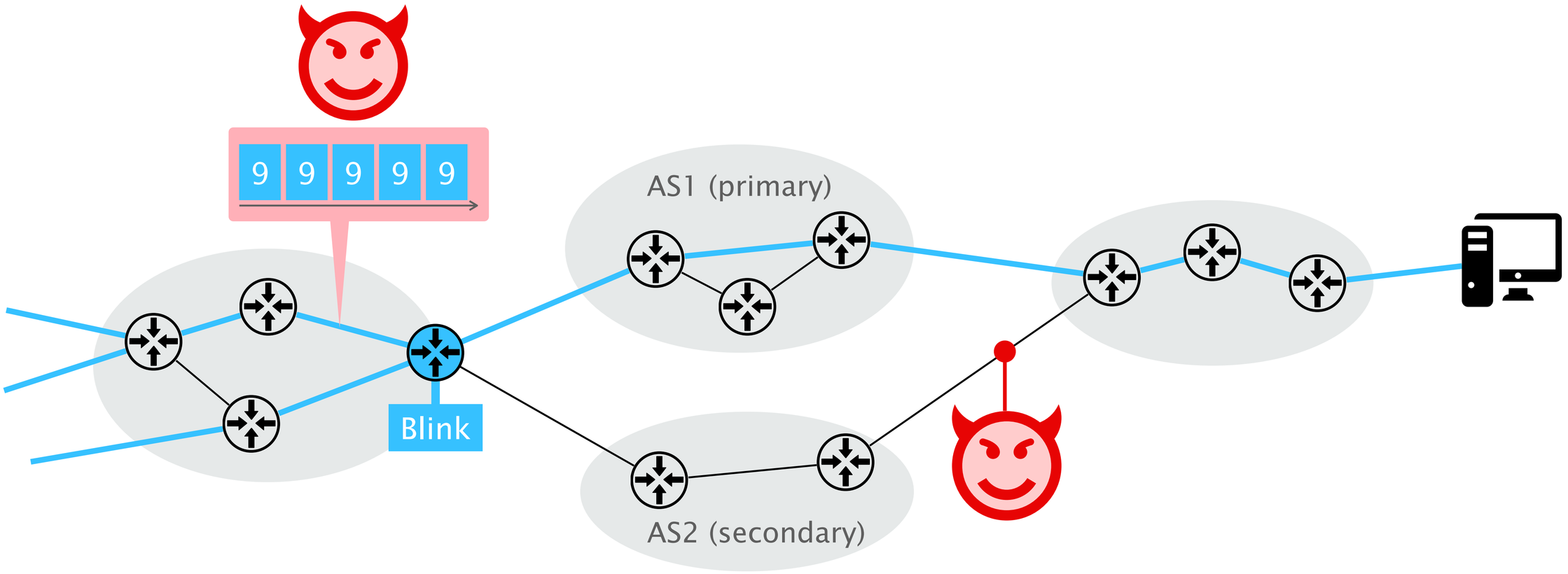
Blink monitors TCP retransmissions to detect failed paths



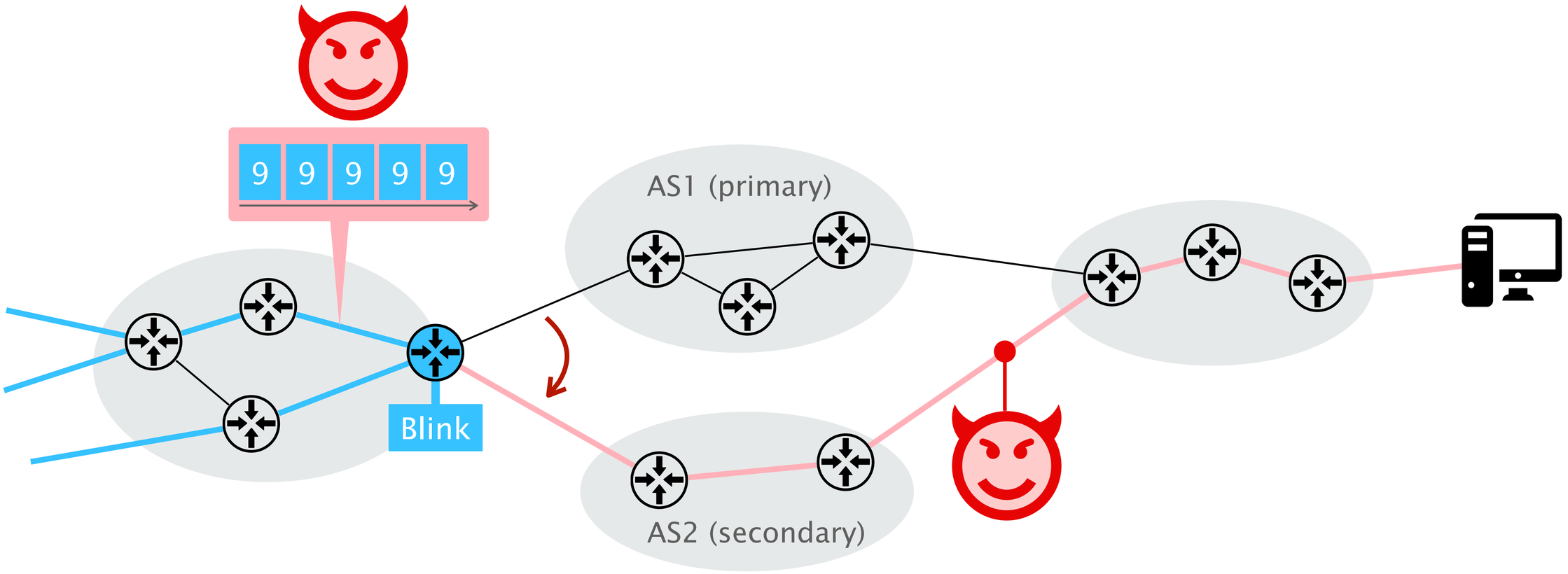
Blink monitors TCP retransmissions to detect failed paths



Blink monitors TCP retransmissions to detect failed paths



Blink monitors TCP retransmissions to detect failed paths



When network programmability meets network security

Protecting users

The good

Being attacked

The bad

3 Attacking others

The ugly

What if you compromise…

one or few programmable switches?

all of them?

Order P4-66*: Characterizing and mitigating surreptitious programmable network device exploitation

Simon Kassing[1], Hussain Abbas[1], Laurent Vanbever[1], Ankit Singla[1]
[1]*ETH Zürich*

Abstract

Substantial efforts are invested in improving network security, but the threat landscape is rapidly evolving, particularly with the recent interest in programmable network hardware. We explore a new security threat, from an attacker who has gained control of such devices. While it should be obvious that such attackers can trivially cause substantial damage, the challenge and novelty are in doing so while preventing quick diagnosis by the operator.

We find that compromised programmable devices can easily degrade networked applications by orders of magnitude, while evading diagnosis by even the most sophisticated network diagnosis methods in deployment. Two key observations yield this result: (a) targeting a small number of packets is often enough to cause disproportionate performance degradation; and (b) new programmable hardware is an effective enabler of careful, selective targeting of packets. Our results also point to recommendations for minimizing the damage from such attacks, ranging from known, easy to implement techniques like encryption and redundant requests, to more complex considerations that would potentially limit some intended uses of programmable hardware. For data center contexts, we also discuss application-aware monitoring and response as a potential mitigation.

1 Introduction

The critical role of networks in our computing ecosystem has made them a high-value target for malicious actors, with high-profile attacks reported with alarming frequency and increasing severity. While the most common attacks target end systems or services like DNS, increasingly, networking devices themselves are coming under attack. The motivation is transparent: compromising network routers and switches provides visibility across large swathes of individual end devices. For many types of attackers, being able to compromise network devices is thus a higher priority [43].

While such attacks on network devices are not new, they have typically focused on either espionage or disruption. We explore a new type of threat on the horizon, posed by an attacker who has obtained control over programmable switches in a target network: long-term deterioration in the network's service that is hard to diagnose. While it should be obvious that an attacker controlling powerful network devices can easily degrade performance to the point of denial-of-service, what distinguishes our work from well-known past attacks is the ability to operate while preventing diagnosis. Preventing or encumbering diagnosis is key to the attack's longevity; if the network operator could easily identify compromised devices, they would be removed or patched. We shall refer to such attacks, conducted with the objective of maximizing long-term, clearly visible damage while preventing diagnosis, as UnderRadar attacks.

A successful UnderRadar attack would result in long-term poor performance, and necessitate time-consuming manual intervention and analysis to eventually fix. Such attacks could be lucrative as "ransomware" – attacking a popular public or private data center network, or ISP or enterprise network, in this way could cost the operator a substantial fraction of their revenue and hurt their reputation if not fixed, thus giving the attacker leverage. They could be broadly attractive to actors who today deploy denial-of-service attacks for "hacktivism", nuisance value, vendetta, etc.

Such attacks are non-trivial, even if the attacker controls powerful devices within a network. Network operators can deploy a variety of monitoring systems that aid in identifying anomalous network behavior and pinpointing faulty devices. Recent research, particularly in the data center context, has developed increasingly sophisticated methods to detect both total and partial failures, *e.g.,* Microsoft's 007 [4], Everflow [47], NetBouncer [42], and Pingmesh [16], and a passive realtime monitoring system (henceforth, referred to as FB-mon) tested in a Facebook data center [36]. These systems are a natural adversary for an UnderRadar attacker.

We develop several UnderRadar attacks and evaluate

*In the Star Wars universe, "Order 66" activated a secret program within the Republic's clone troopers, turning them against their former masters.

arXiv 2021

Mass Surveillance of VoIP Calls in the Data Plane

Ege Cem Kirci
ETH Zürich
Zürich, Switzerland
ekirci@ethz.ch

Maria Apostolaki
Princeton University
Princeton, USA
apostolaki@princeton.edu

Roland Meier
ETH Zürich
Zürich, Switzerland
meierrol@ethz.ch

Ankit Singla
ETH Zürich
Zürich, Switzerland
ankit.singla@inf.ethz.ch

Laurent Vanbever
ETH Zürich
Zürich, Switzerland
lvanbever@ethz.ch

ABSTRACT

Popular Voice-over-IP (VoIP) services such as Signal, Skype, Telegram, and WhatsApp, are considered privacy-preserving due to end-to-end encryption and the establishment of an unmediated communication channel between the callers. However, we identify such applications' call streams in-path, at line-rate, inside terabits of Internet traffic per second, revealing which users communicate with each other.

This work introduces DELTA: a side-channel attack executed by programmable switches. DELTA exploits the inherent network footprint of the VoIP protocol suite, that is, the establishment of direct communication between two users succeeding their peer-discovery routine with publicly-known application servers. We implement DELTA on programmable data planes, enabling tracking of this network footprint at scale and in real-time.

DELTA demonstrates that popular VoIP services are vulnerable to mass surveillance, particularly from specific vantage points in the Internet. We implement DELTA on existing hardware running on terabits per second and conduct high-throughput tests based on representative traffic.

1 INTRODUCTION

Since its inception in 2014, data-plane programmability [8] and P4 language [7] have powered innovation in various networking areas, including Internet security and privacy [5, 14, 31, 60]. Researchers have already demonstrated different use cases where commercial off-the-shelf programmable switches can potentially replace or surpass proprietary network middleboxes [22, 37, 65]. Now that the entry barrier to in-network active intelligence is more accessible than ever, it is time to study the other side of the coin: how does in-line programmability facilitate potentially oppressive activity?

This work studies VoIP call identification, a well-known example of surveillance and censorship [15, 23, 42]. Past investigations reveal that NSA's PRISM program explicitly targeted calls for several popular VoIP applications [24, 35, 46], and various countries had been spying on their residents' calls [39, 50, 51]. The hosting VoIP application of a given call is an intelligence asset, too: reports show that certain dissident groups choose specific VoIP applications to hold their internal communication [1, 12, 41, 43, 57].

Given this interest, it is unsurprising that vendors supply equipment capable of monitoring VoIP calls for on-demand policing, blocking, and recording []. The task is fortunately non-trivial since modern VoIP applications encrypt calls end-to-end [53–55]. End-to-end encryption prevents agencies from directly intercepting the calls. However, it does not protect calls' metadata, such as the IP addresses of the communicating parties,[1] packet sizes, inter-packet delays, or flow durations. Middleboxes such as Sandvine's [] analyze this metadata to eventually "show who is talking to whom, for how long, and try to discover online anonymous identities" [19].

A closer look into the technicalities of these proprietary middleboxes justifies their high cost []. First, they are resource-intensive since they analyze traffic in their CPU clusters off-path []. Second, they require traffic mirroring or advanced state distribution to synchronize in the face of asymmetric routing []. Third, they necessitate continual vendor support since traffic analysis depends on statistical learning and thus regular model updates []. However, this complexity has a silver lining in malicious use cases: it keeps the functionality reasonably out of reach as vendor dependence, dedicated middleboxes, and monetary costs are all limiting factors. We argue that programmable switches challenge these assumptions and demand a reassessment of this technology's possible malicious use cases.

We introduce DELTA a novel network-level side-channel attack that can efficiently identify VoIP calls in-path, in real-time, and at scale. DELTA differentiates from traffic analysis

SOSR '22, October 18–20, 2022, Virtual Event
2022. ACM ISBN 978-1-4503-XXXX-X/18/06.

[1]VoIP applications, by default, establish direct connections between the caller and the callee for better performance [53–55].

SOSR 22

What if you compromise…

one or few programmable switches?

all of them?

Order P4-66*: Characterizing and mitigating surreptitious programmable network device exploitation

Simon Kassing[1], Hussain Abbas[1], Laurent Vanbever[1], Ankit Singla[1]
[1]*ETH Zürich*

Abstract

Substantial efforts are invested in improving network security, but the threat landscape is rapidly evolving, particularly with the recent interest in programmable network hardware. We explore a new security threat, from an attacker who has gained control of such devices. While it should be obvious that such attackers can trivially cause substantial damage, the challenge and novelty are in doing so while preventing quick diagnosis by the operator.

We find that compromised programmable devices can easily degrade networked applications by orders of magnitude, while evading diagnosis by even the most sophisticated network diagnosis methods in deployment. Two key observations yield this result: (a) targeting a small number of packets is often enough to cause disproportionate performance degradation; and (b) new programmable hardware is an effective enabler of careful, selective targeting of packets. Our results also point to recommendations for minimizing the damage from such attacks, ranging from known, easy to implement techniques like encryption and redundant requests, to more complex considerations that would potentially limit some intended uses of programmable hardware. For data center contexts, we also discuss application-aware monitoring and response as a potential mitigation.

1 Introduction

The critical role of networks in our computing ecosystem has made them a high-value target for malicious actors, with high-profile attacks reported with alarming frequency and increasing severity. While the most common attacks target end systems or services like DNS, increasingly, networking devices themselves are coming under attack. The motivation is transparent: compromising network routers and switches provides visibility across large swathes of individual end devices. For many types of attackers, being able to compromise network devices is thus a higher priority [43].

While such attacks on network devices are not new, they have typically focused on either espionage or disruption. We explore a new type of threat on the horizon, posed by an attacker who has obtained control over programmable switches in a target network: long-term deterioration in the network's service that is hard to diagnose. While it should be obvious that an attacker controlling powerful network devices can easily degrade performance to the point of denial-of-service, what distinguishes our work from well-known past attacks is the ability to operate while preventing diagnosis. Preventing or encumbering diagnosis is key to the attack's longevity; if the network operator could easily identify compromised devices, they would be removed or patched. We shall refer to such attacks, conducted with the objective of maximizing long-term, clearly visible damage while preventing diagnosis, as UnderRadar attacks.

A successful UnderRadar attack would result in long-term poor performance, and necessitate time-consuming manual intervention and analysis to eventually fix. Such attacks could be lucrative as "ransomware" – attacking a popular public or private data center network, or ISP or enterprise network, in this way could cost the operator a substantial fraction of their revenue and hurt their reputation if not fixed, thus giving the attacker leverage. They could be broadly attractive to actors who today deploy denial-of-service attacks for "hacktivism", nuisance value, vendetta, etc.

Such attacks are non-trivial, even if the attacker controls powerful devices within a network. Network operators can deploy a variety of monitoring systems that aid in identifying anomalous network behavior and pinpointing faulty devices. Recent research, particularly in the data center context, has developed increasingly sophisticated methods to detect both total and partial failures, *e.g.,* Microsoft's 007 [4], Everflow [47], NetBouncer [42], and Pingmesh [16], and a passive realtime monitoring system (henceforth, referred to as FB-mon) tested in a Facebook data center [36]. These systems are a natural adversary for an UnderRadar attacker.

We develop several UnderRadar attacks and evaluate

*In the Star Wars universe, "Order 66" activated a secret program within

arXiv 2021

Mass Surveillance of VoIP Calls in the Data Plane

Ege Cem Kirci
ETH Zürich
Zürich, Switzerland
ekirci@ethz.ch

Maria Apostolaki
Princeton University
Princeton, USA
apostolaki@princeton.edu

Roland Meier
ETH Zürich
Zürich, Switzerland
meierrol@ethz.ch

Ankit Singla
ETH Zürich
Zürich, Switzerland
ankit.singla@inf.ethz.ch

Laurent Vanbever
ETH Zürich
Zürich, Switzerland
lvanbever@ethz.ch

ABSTRACT

Popular Voice-over-IP (VoIP) services such as Signal, Skype, Telegram, and WhatsApp, are considered privacy-preserving due to end-to-end encryption and the establishment of an unmediated communication channel between the callers. However, we identify such applications' call streams in-path, at line-rate, inside terabits of Internet traffic per second, revealing which users communicate with each other.

This work introduces DELTA: a side-channel attack executed by programmable switches. DELTA exploits the inherent network footprint of the VoIP protocol suite, that is, the establishment of direct communication between two users succeeding their peer-discovery routine with publicly-known application servers. We implement DELTA on programmable data planes, enabling tracking of this network footprint at scale and in real-time.

DELTA demonstrates that popular VoIP services are vulnerable to mass surveillance, particularly from specific vantage points in the Internet. We implement DELTA on existing hardware running on terabits per second and conduct high-throughput tests based on representative traffic.

1 INTRODUCTION

Since its inception in 2014, data-plane programmability [8] and P4 language [7] have powered innovation in various networking areas, including Internet security and privacy [5, 14, 31, 60]. Researchers have already demonstrated different use cases where commercial off-the-shelf programmable switches can potentially replace or surpass proprietary network middleboxes [22, 37, 65]. Now that the entry barrier to in-network active intelligence is more accessible than ever, it is time to study the other side of the coin: how does in-line programmability facilitate potentially oppressive activity?

This work studies VoIP call identification, a well-known example of surveillance and censorship [15, 23, 42]. Past investigations reveal that NSA's PRISM program explicitly targeted calls for several popular VoIP applications [24, 35, 46], and various countries had been spying on their residents' calls [39, 50, 51]. The hosting VoIP application of a given call is an intelligence asset, too: reports show that certain dissident groups choose specific VoIP applications to hold their internal communication [1, 12, 41, 43, 57].

Given this interest, it is unsurprising that vendors supply equipment capable of monitoring VoIP calls for on-demand policing, blocking, and recording []. The task is fortunately non-trivial since modern VoIP applications encrypt calls end-to-end [53–55]. End-to-end encryption prevents agencies from directly intercepting the calls. However, it does not protect calls' metadata, such as the IP addresses of the communicating parties,[1] packet sizes, inter-packet delays, or flow durations. Middleboxes such as Sandvine's [] analyze this metadata to eventually "show who is talking to whom, for how long, and try to discover online anonymous identities" [19].

A closer look into the technicalities of these proprietary middleboxes justifies their high cost []. First, they are resource-intensive since they analyze traffic in their CPU clusters off-path []. Second, they require traffic mirroring or advanced state distribution to synchronize in the face of asymmetric routing []. Third, they necessitate continual vendor support since traffic analysis depends on statistical learning and thus regular model updates []. However, this complexity has a silver lining in malicious use cases: it keeps the functionality reasonably out of reach as vendor dependence, dedicated middleboxes, and monetary costs are all limiting factors. We argue that programmable switches challenge these assumptions and demand a reassessment of this technology's possible malicious use cases.

We introduce DELTA a novel network-level side-channel attack that can efficiently identify VoIP calls in-path, in real-time, and at scale. DELTA differentiates from traffic analysis

SOSR '22, October 18–20, 2022, Virtual Event
2022. ACM ISBN 978-1-4503-XXXX-X/18/06.

[1]VoIP applications, by default, establish direct connections between the caller and the callee for better performance [53–55].

SOSR 22

Turns out programmable hardware is *great* to perform a wide-variety of attacks, under the radar

Turns out programmable hardware is *great* to perform

a wide-variety of attacks, under the radar

network & transport-level attacks
aimed at decreasing performance

Turns out programmable hardware is *great* to perform a wide-variety of attacks, under the radar

evading state-of-the-art network monitoring systems

Attacks
SYN Drop
Same Sequence Drop
SYN Flood
RST Tinker
ACK and Drop
ECN Tinker
CWND Tinker
Coordinated Incast

see paper for more attacks

Attacks	Monitoring Systems						
	sFlow	NetFlow	007	NetBouncer	Pingmesh	FB-Mon	Everflow
SYN Drop							
Same Sequence Drop							
SYN Flood							
RST Tinker							
ACK and Drop							
ECN Tinker							
CWND Tinker							
Coordinated Incast							

Attacks	Monitoring Systems							Damage
	sFlow	NetFlow	007	NetBouncer	Pingmesh	FB-Mon	Everflow	
SYN Drop								
Same Sequence Drop								
SYN Flood								
RST Tinker								
ACK and Drop								
ECN Tinker								
CWND Tinker								
Coordinated Incast								

Attackers can deteriorate performance for targeted flows by several times, up to a complete denial of service

Attacks	Monitoring Systems							Damage
	sFlow	NetFlow	007	NetBouncer	Pingmesh	FB-Mon	Everflow	
SYN Drop	✓	✓	✓	✓	✓	✓[a,e]	✗	64x Conn. Est. Time [c]
Same Sequence Drop	✓	✓	✓[d]	✓	✓	✓[a,e]	✓[b]	50x FCT [c]
SYN Flood	✓	✓	✓	✓	✓	✓[a,e]	✗	DoS
RST Tinker	✓	✓	✓	✓	✓	✓[a,e]	✗	Disconnect
ACK and Drop	✓	✓	✓[d]	✓	✓	✓[a,e]	✓[b]	Disconnect
ECN Tinker	✓	✓	✓	✓	✓	✓[a,e]	✓[b]	6x FCT
CWND Tinker	✓	✓	✓	✓	✓	✓[a,e]	✓[b]	245x FCT
Coordinated Incast	✓	✓	✓[a]	✓[a]	✓	✓[a]	✓[a]	1.5x FCT

Attackers can deteriorate performance for targeted flows

by several times… while going undetected

Attacks	Monitoring Systems							Damage
	sFlow	NetFlow	007	NetBouncer	Pingmesh	FB-Mon	Everflow	
SYN Drop	✓	✓	✓	✓	✓	✓[a,e]	✗	64x Conn. Est. Time [c]
Same Sequence Drop	✓	✓	✓[d]	✓	✓	✓[a,e]	✓[b]	50x FCT [c]
SYN Flood	✓	✓	✓	✓	✓	✓[a,e]	✗	DoS
RST Tinker	✓	✓	✓	✓	✓	✓[a,e]	✗	Disconnect
ACK and Drop	✓	✓	✓[d]	✓	✓	✓[a,e]	✓[b]	Disconnect
ECN Tinker	✓	✓	✓	✓	✓	✓[a,e]	✓[b]	6x FCT
CWND Tinker	✓	✓	✓	✓	✓	✓[a,e]	✓[b]	245x FCT
Coordinated Incast	✓	✓	✓[a]	✓[a]	✓	✓[a]	✓[a]	1.5x FCT

What if you compromise…

one or few programmable switches?

all of them?

Order P4-66*: Characterizing and mitigating surreptitious programmable network device exploitation

Simon Kassing[1], Hussain Abbas[1], Laurent Vanbever[1], Ankit Singla[1]
[1]*ETH Zürich*

Abstract

Substantial efforts are invested in improving network security, but the threat landscape is rapidly evolving, particularly with the recent interest in programmable network hardware. We explore a new security threat, from an attacker who has gained control of such devices. While it should be obvious that such attackers can trivially cause substantial damage, the challenge and novelty are in doing so while preventing quick diagnosis by the operator.

We find that compromised programmable devices can easily degrade networked applications by orders of magnitude, while evading diagnosis by even the most sophisticated network diagnosis methods in deployment. Two key observations yield this result: (a) targeting a small number of packets is often enough to cause disproportionate performance degradation; and (b) new programmable hardware is an effective enabler of careful, selective targeting of packets. Our results also point to recommendations for minimizing the damage from such attacks, ranging from known, easy to implement techniques like encryption and redundant requests, to more complex considerations that would potentially limit some intended uses of programmable hardware. For data center contexts, we also discuss application-aware monitoring and response as a potential mitigation.

1 Introduction

The critical role of networks in our computing ecosystem has made them a high-value target for malicious actors, with high-profile attacks reported with alarming frequency and increasing severity. While the most common attacks target end systems or services like DNS, increasingly, networking devices themselves are coming under attack. The motivation is transparent: compromising network routers and switches provides visibility across large swathes of individual end devices. For many types of attackers, being able to compromise network devices is thus a higher priority [43].

While such attacks on network devices are not new, they have typically focused on either espionage or disruption. We explore a new type of threat on the horizon, posed by an attacker who has obtained control over programmable switches in a target network: long-term deterioration in the network's service that is hard to diagnose. While it should be obvious that an attacker controlling powerful network devices can easily degrade performance to the point of denial-of-service, what distinguishes our work from well-known past attacks is the ability to operate while preventing diagnosis. Preventing or encumbering diagnosis is key to the attack's longevity; if the network operator could easily identify compromised devices, they would be removed or patched. We shall refer to such attacks, conducted with the objective of maximizing long-term, clearly visible damage while preventing diagnosis, as UnderRadar attacks.

A successful UnderRadar attack would result in long-term poor performance, and necessitate time-consuming manual intervention and analysis to eventually fix. Such attacks could be lucrative as "ransomware" – attacking a popular public or private data center network, or ISP or enterprise network, in this way could cost the operator a substantial fraction of their revenue and hurt their reputation if not fixed, thus giving the attacker leverage. They could be broadly attractive to actors who today deploy denial-of-service attacks for "hacktivism", nuisance value, vendetta, etc.

Such attacks are non-trivial, even if the attacker controls powerful devices within a network. Network operators can deploy a variety of monitoring systems that aid in identifying anomalous network behavior and pinpointing faulty devices. Recent research, particularly in the data center context, has developed increasingly sophisticated methods to detect both total and partial failures, *e.g.,* Microsoft's 007 [4], Everflow [47], NetBouncer [42], and Pingmesh [16], and a passive realtime monitoring system (henceforth, referred to as FB-mon) tested in a Facebook data center [36]. These systems are a natural adversary for an UnderRadar attacker.

*In the Star Wars universe, "Order 66" activated a secret program within the Republic's clone troopers, turning them against their Jedi force providers.

arXiv 2021

Mass Surveillance of VoIP Calls in the Data Plane

Ege Cem Kirci
ETH Zürich
Zürich, Switzerland
ekirci@ethz.ch

Maria Apostolaki
Princeton University
Princeton, USA
apostolaki@princeton.edu

Roland Meier
ETH Zürich
Zürich, Switzerland
meierrol@ethz.ch

Ankit Singla
ETH Zürich
Zürich, Switzerland
ankit.singla@inf.ethz.ch

Laurent Vanbever
ETH Zürich
Zürich, Switzerland
lvanbever@ethz.ch

ABSTRACT

Popular Voice-over-IP (VoIP) services such as Signal, Skype, Telegram, and WhatsApp, are considered privacy-preserving due to end-to-end encryption and the establishment of an unmediated communication channel between the callers. However, we identify such applications' call streams in-path, at line-rate, inside terabits of Internet traffic per second, revealing which users communicate with each other.

This work introduces DELTA: a side-channel attack executed by programmable switches. DELTA exploits the inherent network footprint of the VoIP protocol suite, that is, the establishment of direct communication between two users succeeding their peer-discovery routine with publicly-known application servers. We implement DELTA on programmable data planes, enabling tracking of this network footprint at scale and in real-time.

DELTA demonstrates that popular VoIP services are vulnerable to mass surveillance, particularly from specific vantage points in the Internet. We implement DELTA on existing hardware running on terabits per second and conduct high-throughput tests based on representative traffic.

1 INTRODUCTION

Since its inception in 2014, data-plane programmability [8] and P4 language [7] have powered innovation in various networking areas, including Internet security and privacy [5, 14, 31, 60]. Researchers have already demonstrated different use cases where commercial off-the-shelf programmable switches can potentially replace or surpass proprietary network middleboxes [22, 37, 65]. Now that the entry barrier to in-network active intelligence is more accessible than ever, it is time to study the other side of the coin: how does in-line programmability facilitate potentially oppressive activity?

This work studies VoIP call identification, a well-known example of surveillance and censorship [15, 23, 42]. Past investigations reveal that NSA's PRISM program explicitly targeted calls for several popular VoIP applications [24, 35, 46], and various countries had been spying on their residents' calls [39, 50, 51]. The hosting VoIP application of a given call is an intelligence asset, too: reports show that certain dissident groups choose specific VoIP applications to hold their internal communication [1, 12, 41, 43, 57].

Given this interest, it is unsurprising that vendors supply equipment capable of monitoring VoIP calls for on-demand policing, blocking, and recording []. The task is fortunately non-trivial since modern VoIP applications encrypt calls end-to-end [53–55]. End-to-end encryption prevents agencies from directly intercepting the calls. However, it does not protect calls' metadata, such as the IP addresses of the communicating parties,[1] packet sizes, inter-packet delays, or flow durations. Middleboxes such as Sandvine's [] analyze this metadata to eventually "show who is talking to whom, for how long, and try to discover online anonymous identities" [19].

A closer look into the technicalities of these proprietary middleboxes justifies their high cost []. First, they are resource-intensive since they analyze traffic in their CPU clusters off-path []. Second, they require traffic mirroring or advanced state distribution to synchronize in the face of asymmetric routing []. Third, they necessitate continual vendor support since traffic analysis depends on statistical learning and thus regular model updates []. However, this complexity has a silver lining in malicious use cases: it keeps the functionality reasonably out of reach as vendor dependence, dedicated middleboxes, and monetary costs are all limiting factors. We argue that programmable switches challenge these assumptions and demand a reassessment of this technology's possible malicious use cases.

We introduce DELTA a novel network-level side-channel attack that can efficiently identify VoIP calls in-path, in real-time, and at scale. DELTA differentiates from traffic analysis

SOSR '22, October 18–20, 2022, Virtual Event
2022. ACM ISBN 978-1-4503-XXXX-X/18/06.

[1]VoIP applications, by default, establish direct connections between the caller and the callee for better performance [53–55].

SOSR 22

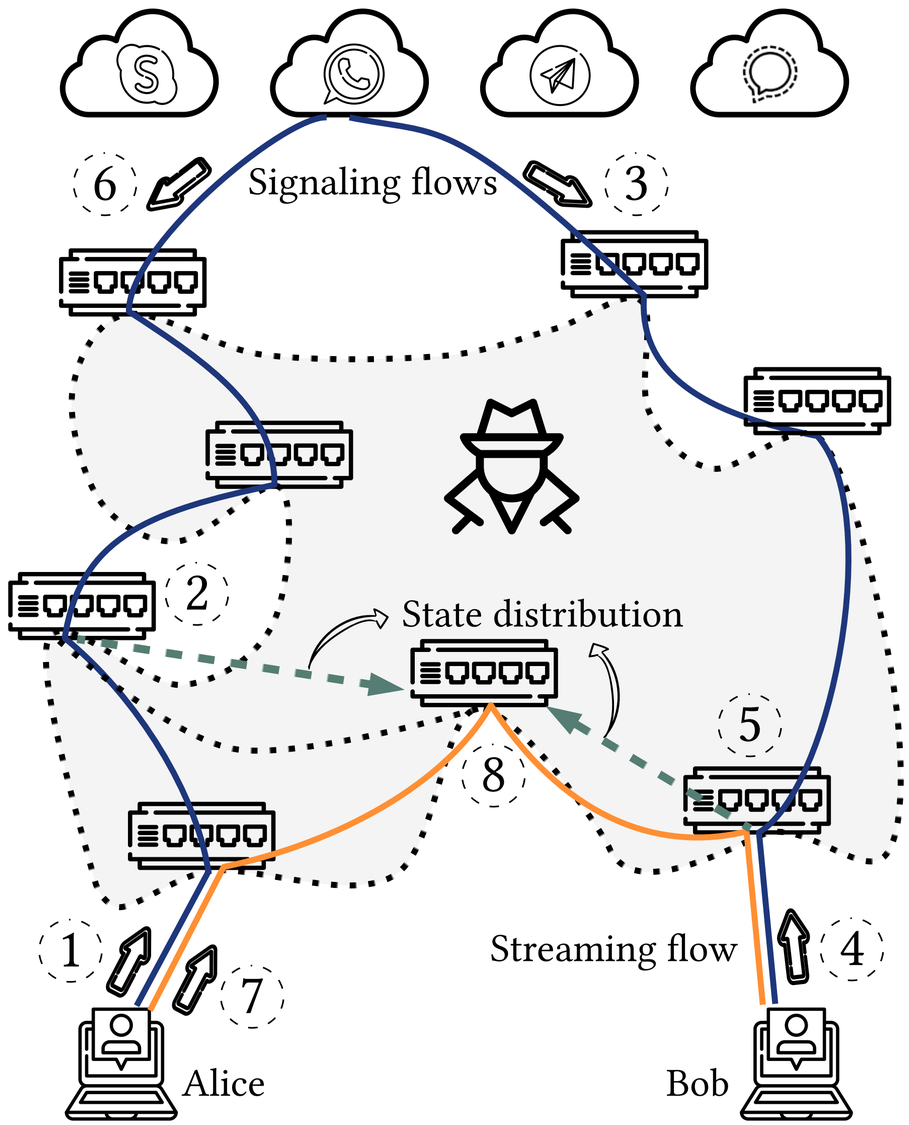
Malicious networks are powerful attackers (unsurprisingly) capable—among many others—of advanced side-channel attacks

Malicious networks are powerful attackers (unsurprisingly) capable—among many others—of advanced side-channel attacks

e.g. identifying VoIP calls

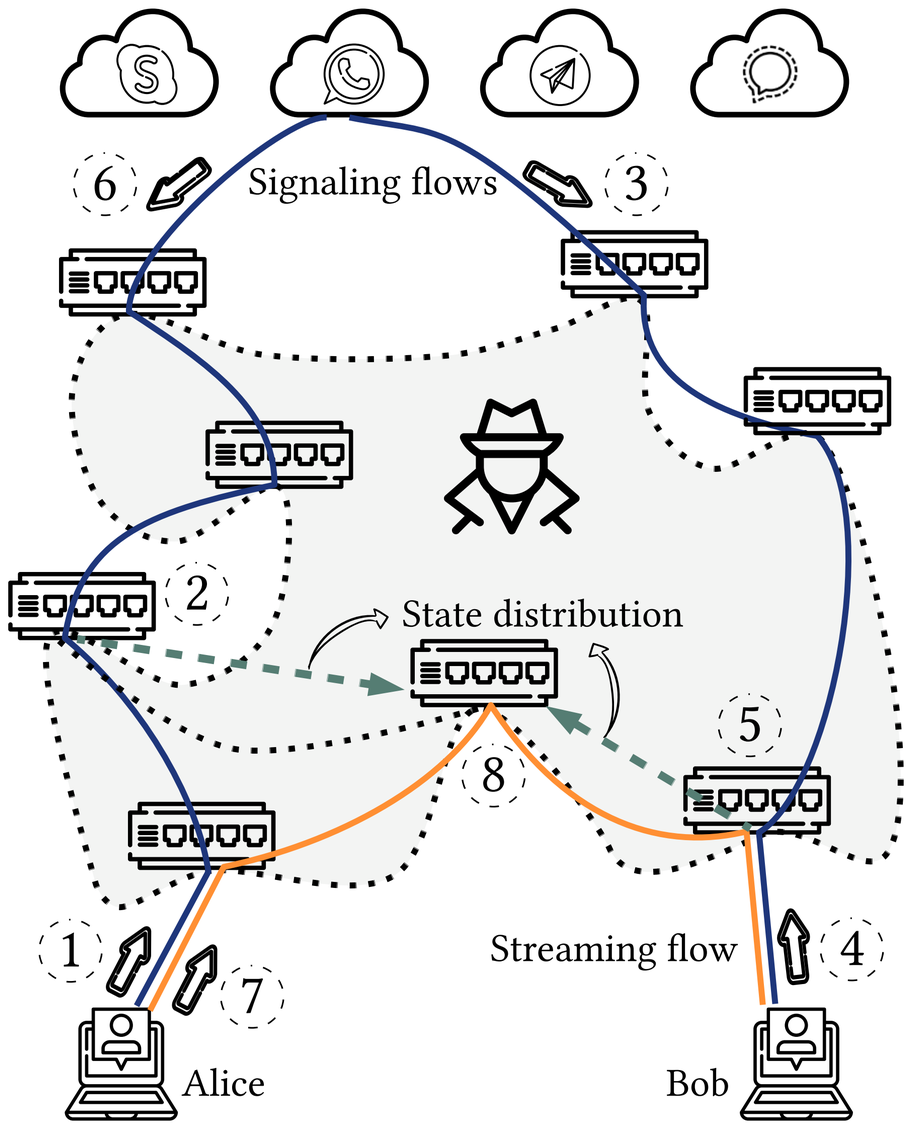
VoIP signaling: a 3-step process

VoIP signaling: a 3-step process



VoIP signaling: a 3-step process

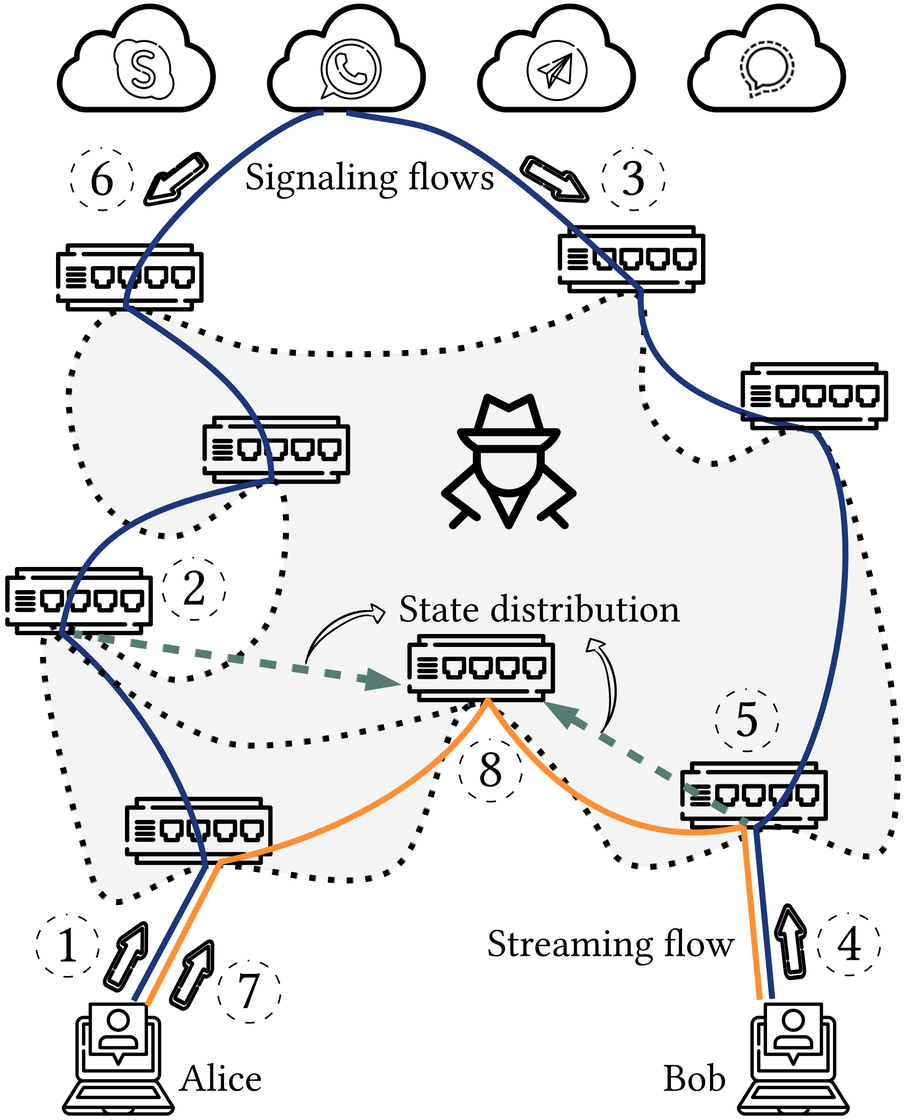
- caller contacts server
- server contacts callee
- callers calls callee



VoIP signaling: a 3-step process

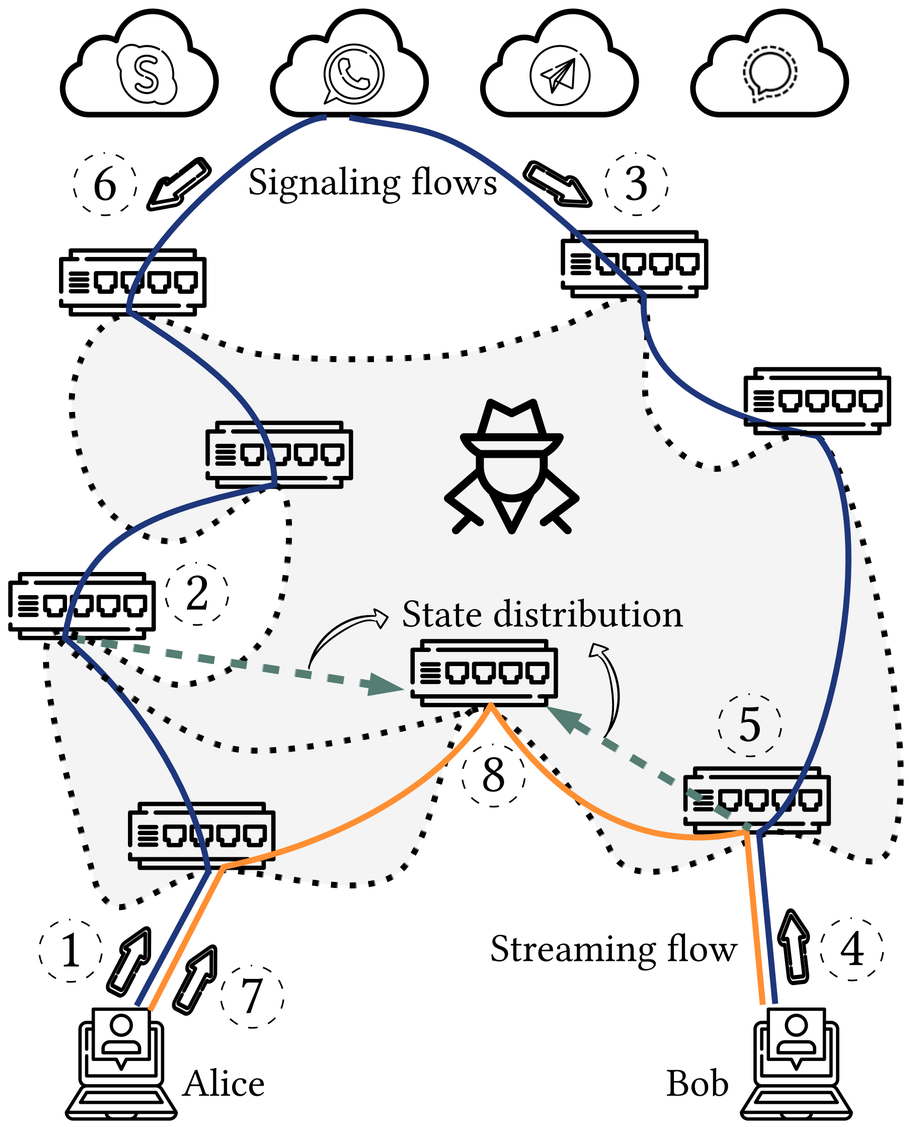
- caller contacts server
- server contacts callee
- callers calls callee

A programmable network can efficiently track these triangular patterns



Delta Attacks

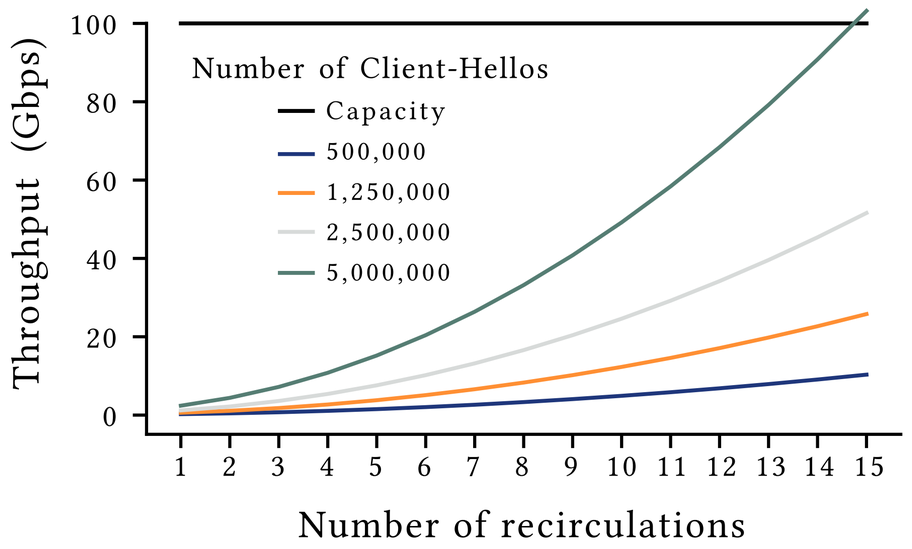
- match signaling connections
  using string matching capabilities
- keep track of triangle formation
  using a probabilistic data structure
- synchronize states across switches
  using a data-plane messaging protocol

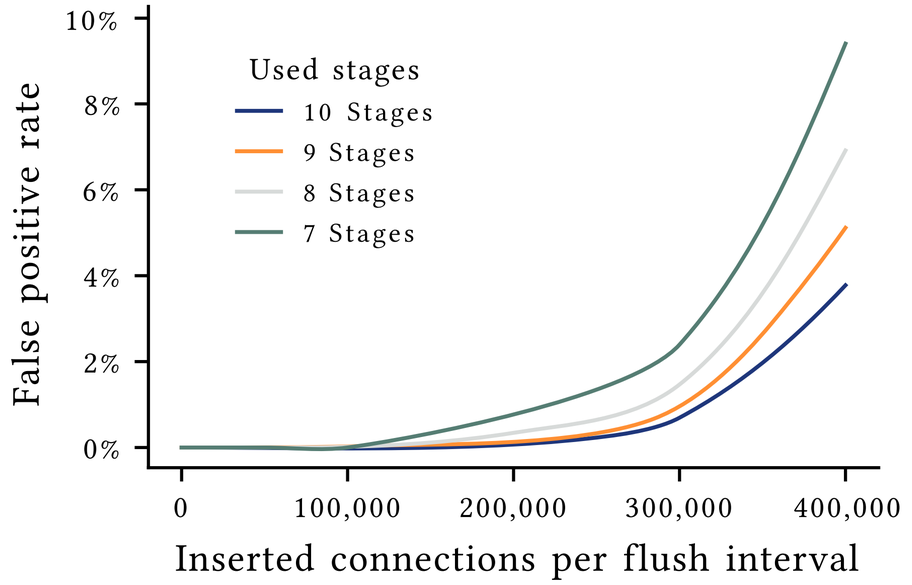


Delta Attacks

A single programmable switch can

- track 100k+ unique VoIP calls
- inspect 5M+ signaling packets/second





When network programmability meets network security

Protecting users

The good

Being attacked

The bad

Attacking others

The ugly

Some of these problems are inherent and can't be solved

Some of these problems are inherent and can't be solved.

For the rest, we'll need to take a hollistic approach

For the rest, we'll need to take a hollistic approach

Inputs

Program

Outputs

Inputs

Program

Outputs

How do we protect our network programs from adversarial inputs?

Similar problems in Machine Learning

Inputs

Program

Outputs

How do we guarantee that network programs *only* do what they are supposed to?

Similar problems in program verification

Inputs

Program

Outputs

How do we monitor and "silo" running net. programs?

Similar problems in virtualization

Interested? Join us!

We're hiring!

When network programmability meets security:

The good. The bad. And the ugly.

Laurent Vanbever
nsg.ee.ethz.ch

FFSPIN 2022
Mon Aug 22 2022