ditto: WAN 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

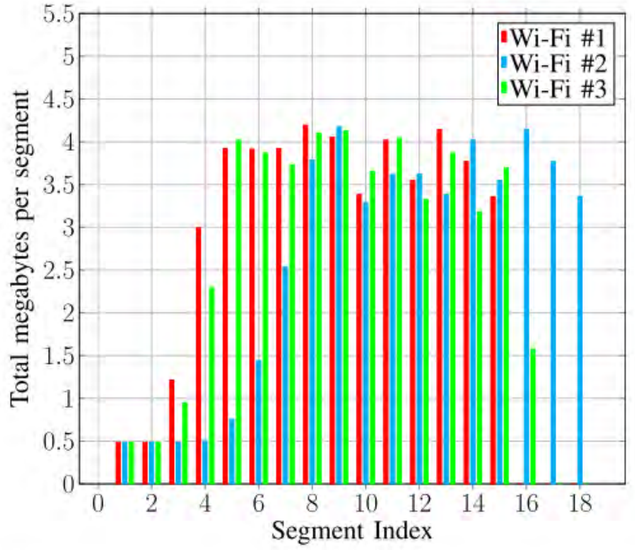
to determine which video somebody is watching

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

… and there are many 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

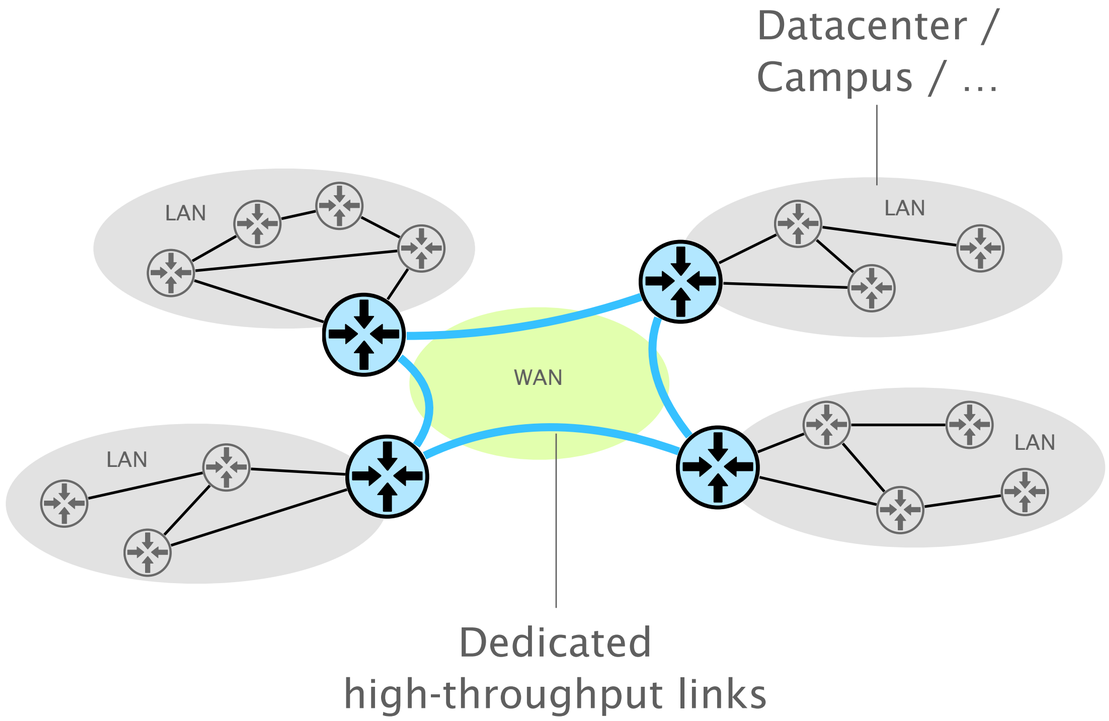
Jae Won Chung
Verizon Labs
Waltham, MA, 02145

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]

This kind of attacks is concerning for Wide Area Network operators too

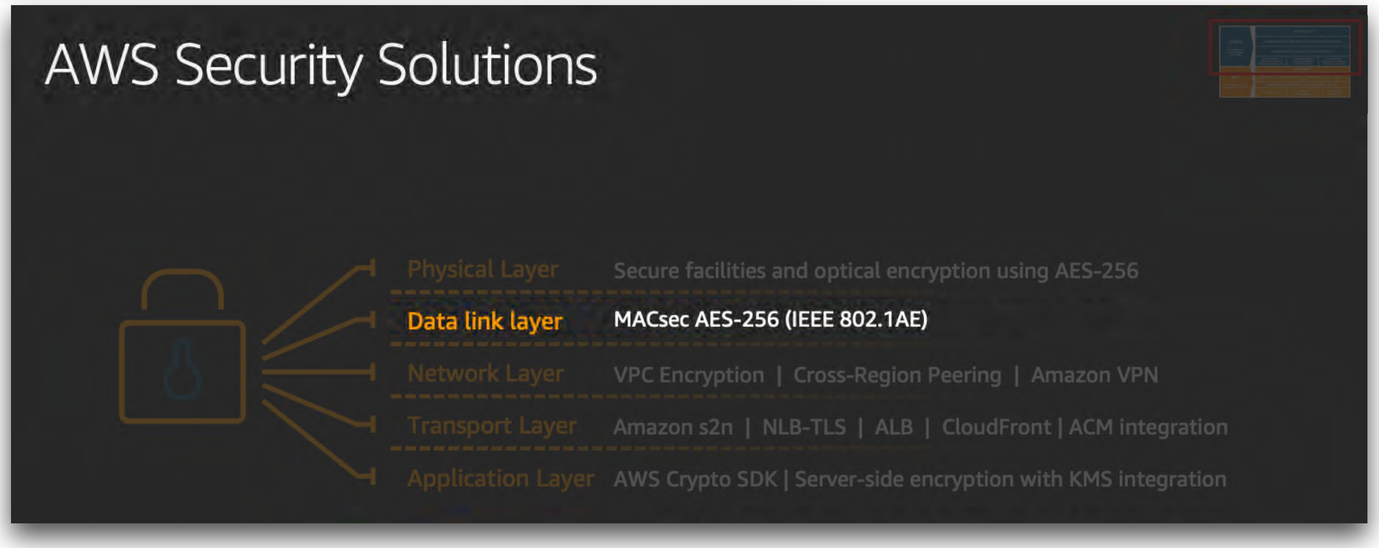


[nbg.tech]

Major WAN operators acknowledge the risk and already use link-layer encryption

Data-link Layer encryption in Azure

Whenever Azure Customer traffic moves between datacenters -- outside physical boundaries not controlled by Microsoft (or on behalf of Microsoft)-- a data-link layer encryption method using the IEEE 802.1AE MAC Security Standards (also known as MACsec) is applied from point-to-point across the underlying network hardware. The packets are encrypted and decrypted on the devices before being sent, preventing physical "man-in-the-middle" or snooping/wiretapping attacks. Because this technology is integrated on the network hardware itself, it provides line rate encryption on the network hardware with no measurable link latency increase. This MACsec encryption is on by default for all Azure traffic traveling within a region or between regions, and no action is required on customers' part to enable.



Three challenges for a practical WAN traffic-analysis prevention system

- Security
  Traffic does not leak information

  ditto makes observed traffic independent from the actual traffic

- Performance
  WANs run at 100s of Gbps

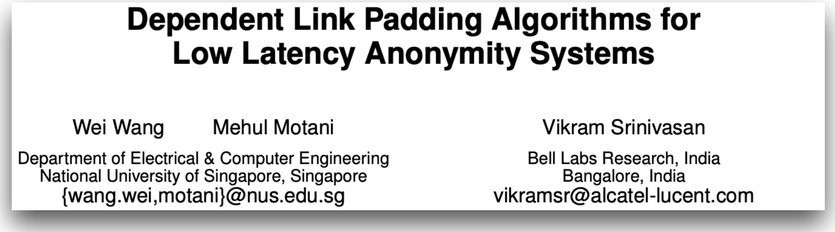
  ditto reduces overhead by using efficient traffic patterns

- Deployability
  Infeasible to change all servers

  ditto runs in the network data plane at line rate

Existing countermeasures do not satisfy the requirements of WANs

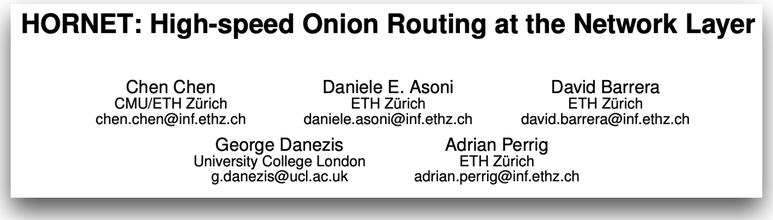
- Incomplete security
  Still allow traffic analysis attacks
- Low throughput
  Megabits to few gigabits per second
- Difficult to deploy
  Require changes at end-hosts



Dependent Link Padding Algorithms for Low Latency Anonymity Systems

Wei Wang, Mehul Motani
Department of Electrical & Computer Engineering
National University of Singapore, Singapore
{wang.wei,motani}@nus.edu.sg

Vikram Srinivasan
Bell Labs Research, India
Bangalore, India
vikramsr@alcatel-lucent.com



HORNET: High-speed Onion Routing at the Network Layer

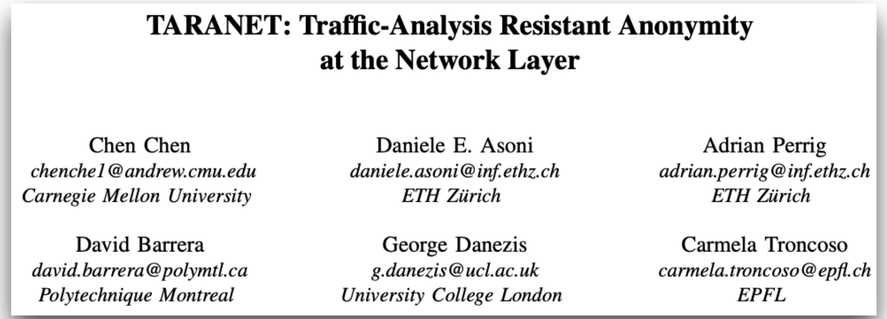
Chen Chen
CMU/ETH Zürich
chen.chen@inf.ethz.ch

Daniele E. Asoni
ETH Zürich
daniele.asoni@inf.ethz.ch

David Barrera
ETH Zürich
david.barrera@inf.ethz.ch

George Danezis
University College London
g.danezis@ucl.ac.uk

Adrian Perrig
ETH Zürich
adrian.perrig@inf.ethz.ch



TARANET: Traffic-Analysis Resistant Anonymity at the Network Layer

Chen Chen
chenche1@andrew.cmu.edu
Carnegie Mellon University

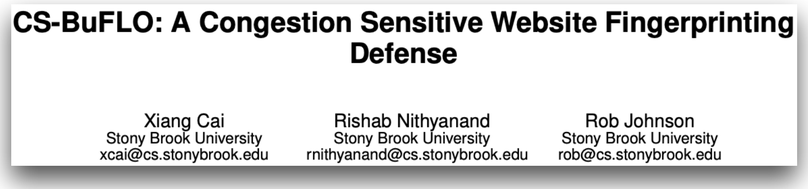
Daniele E. Asoni
daniele.asoni@inf.ethz.ch
ETH Zürich

Adrian Perrig
adrian.perrig@inf.ethz.ch
ETH Zürich

David Barrera
david.barrera@polymtl.ca
Polytechnique Montreal

George Danezis
g.danezis@ucl.ac.uk
University College London

Carmela Troncoso
carmela.troncoso@epfl.ch
EPFL



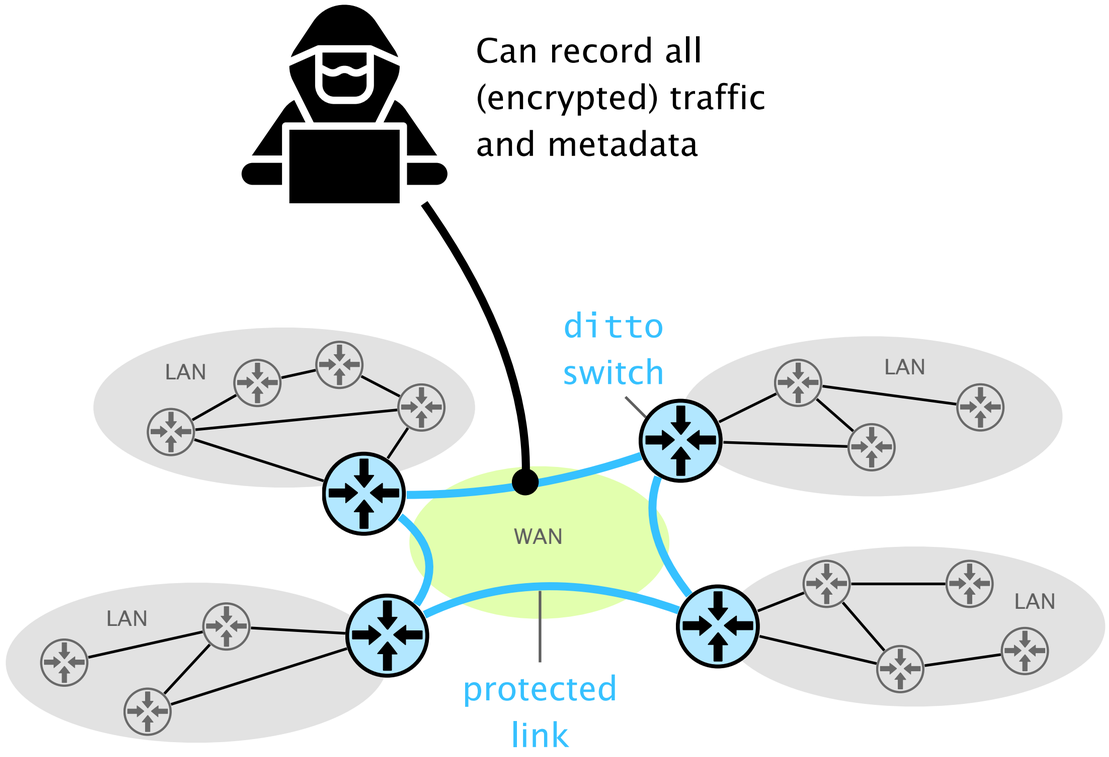
CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense

Xiang Cai
Stony Brook University
xcai@cs.stonybrook.edu

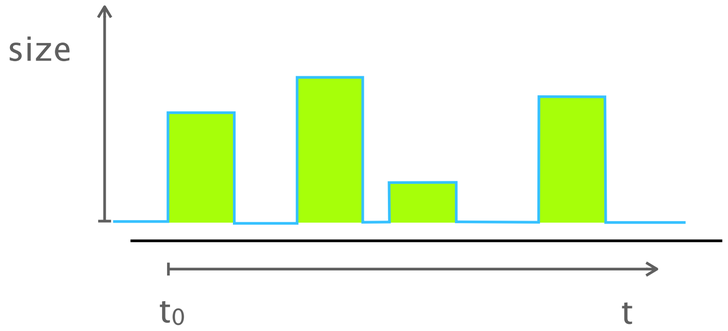
Rishab Nithyanand
Stony Brook University
rnithyanand@cs.stonybrook.edu

Rob Johnson
Stony Brook University
rob@cs.stonybrook.edu

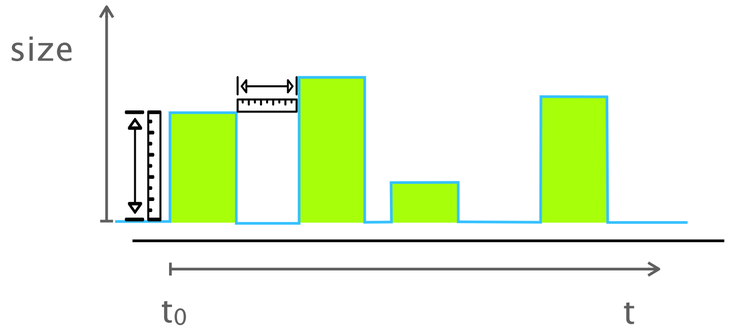
ditto protects against an eavesdropper and provides three security properties

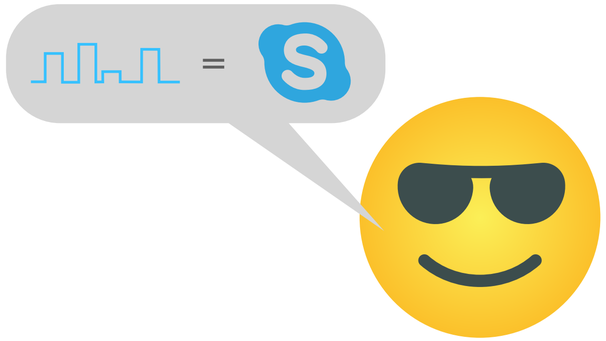


- Volume anonymity
  Attacker cannot determine the size of packets and flows
- Timing anonymity
  Attacker cannot determine the timing between two packets
- Path anonymity
  Attacker cannot track packets across multiple protected links

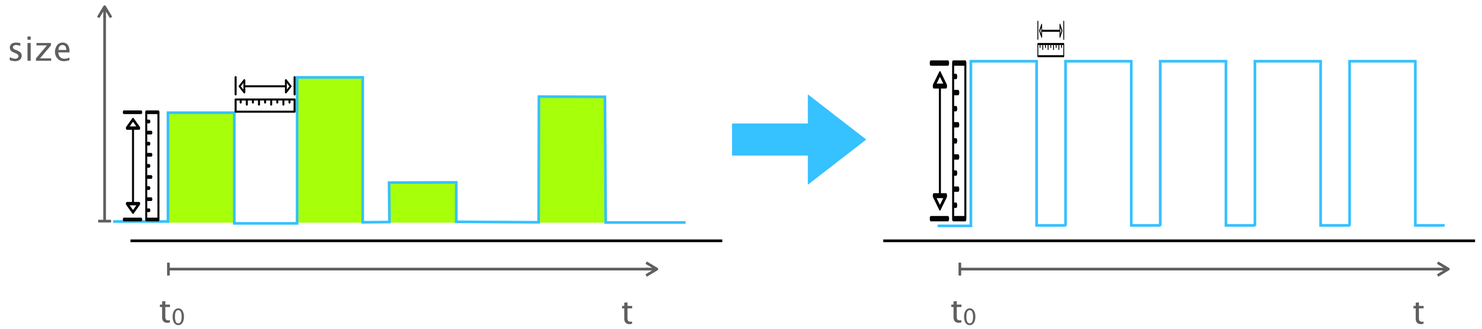

size
t_0
t

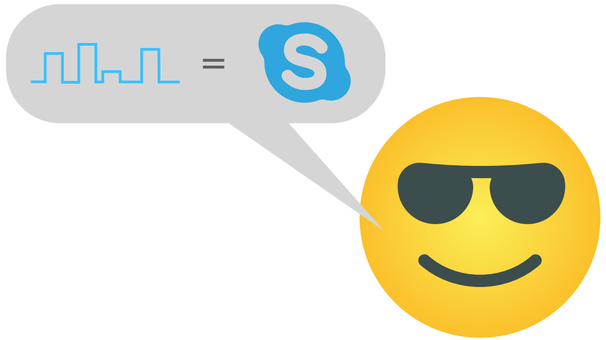
Packet sizes and timing allow traffic-analysis attacks in unprotected traffic



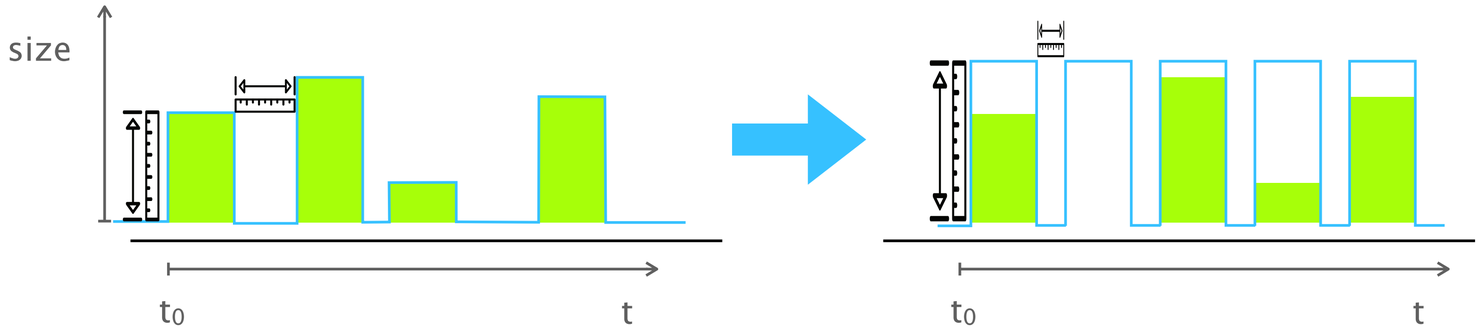


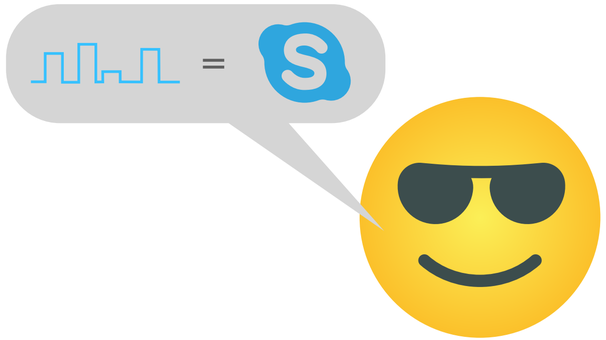
The high-level idea behind ditto is to make the observed traffic independent from the real traffic



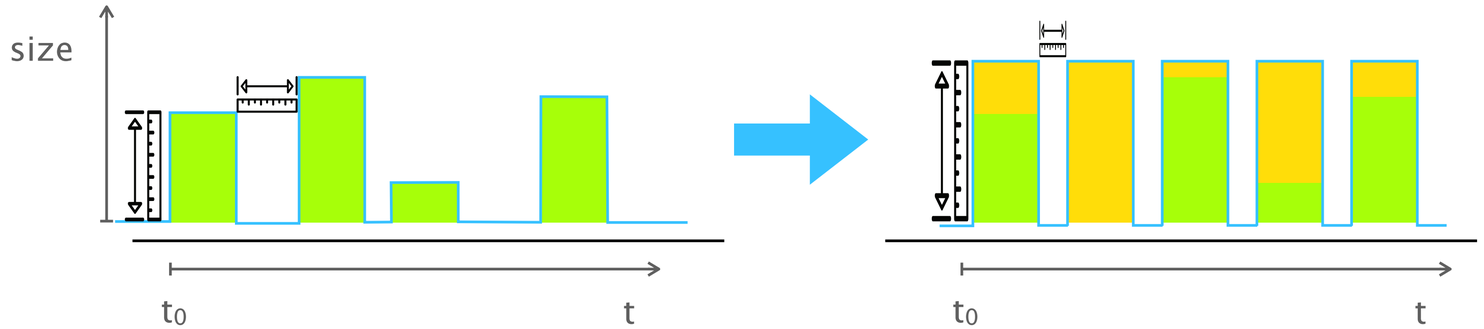


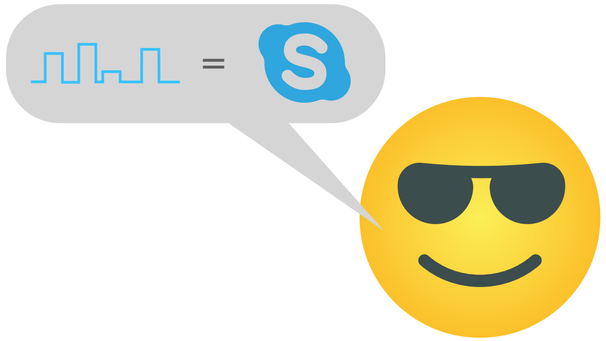
The high-level idea behind ditto is to make the observed traffic independent from the real traffic



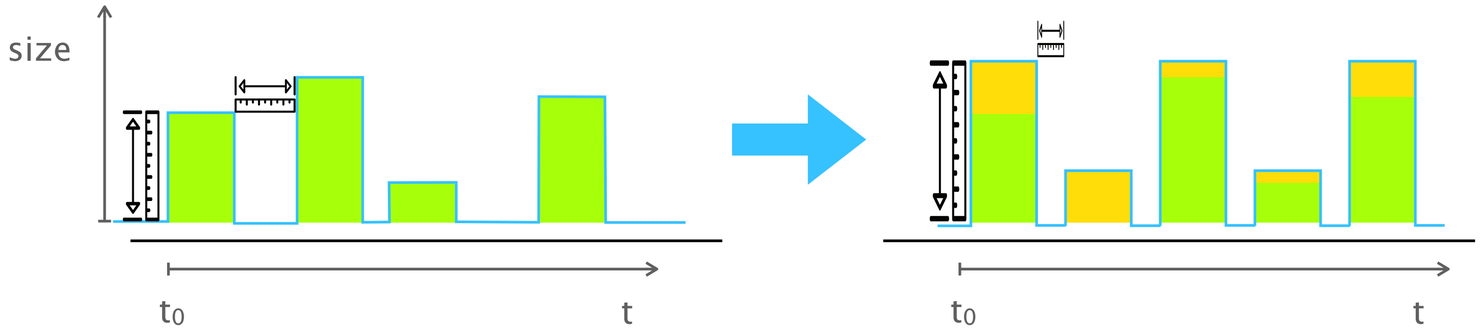


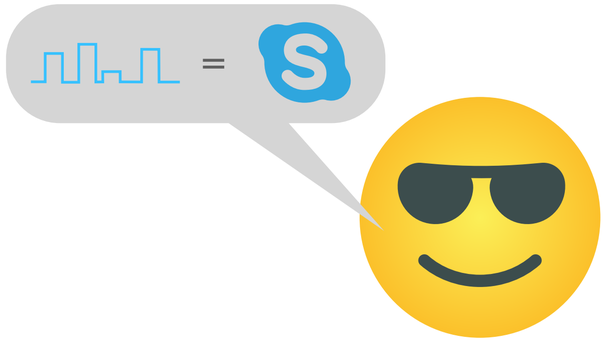
While secure, “constant” traffic can be inefficient



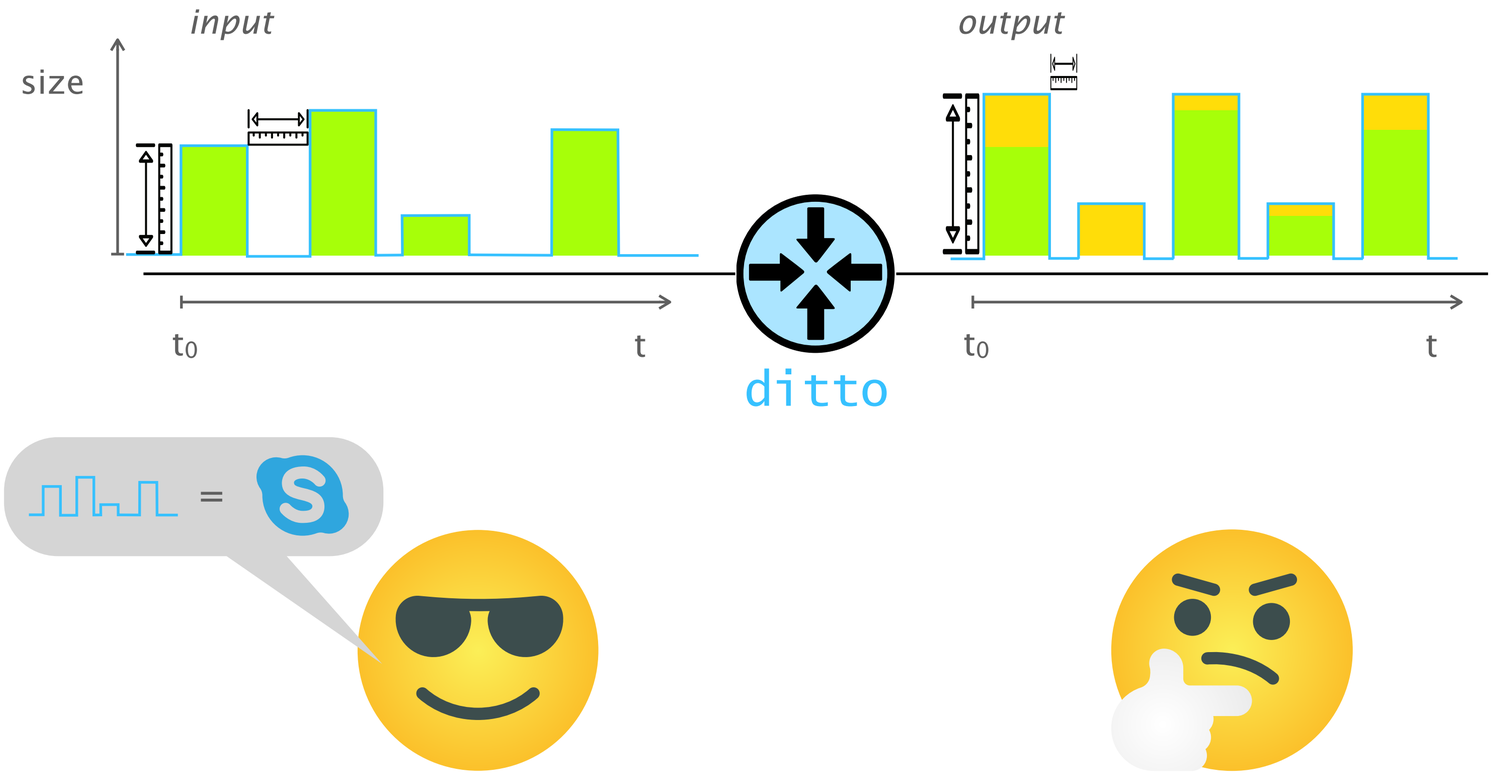


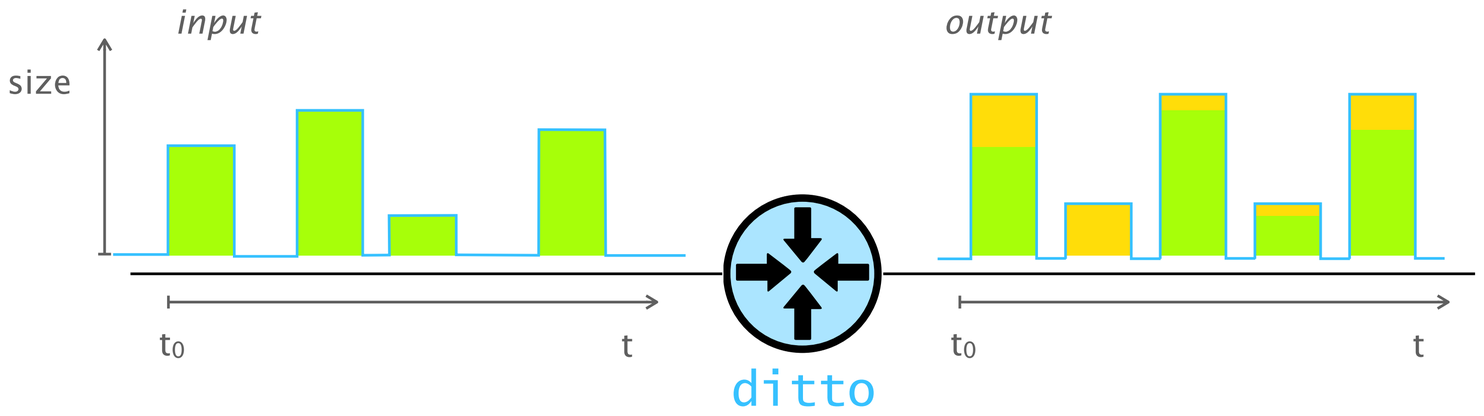
ditto shapes traffic according to an efficient pattern





ditto runs in the network data plane

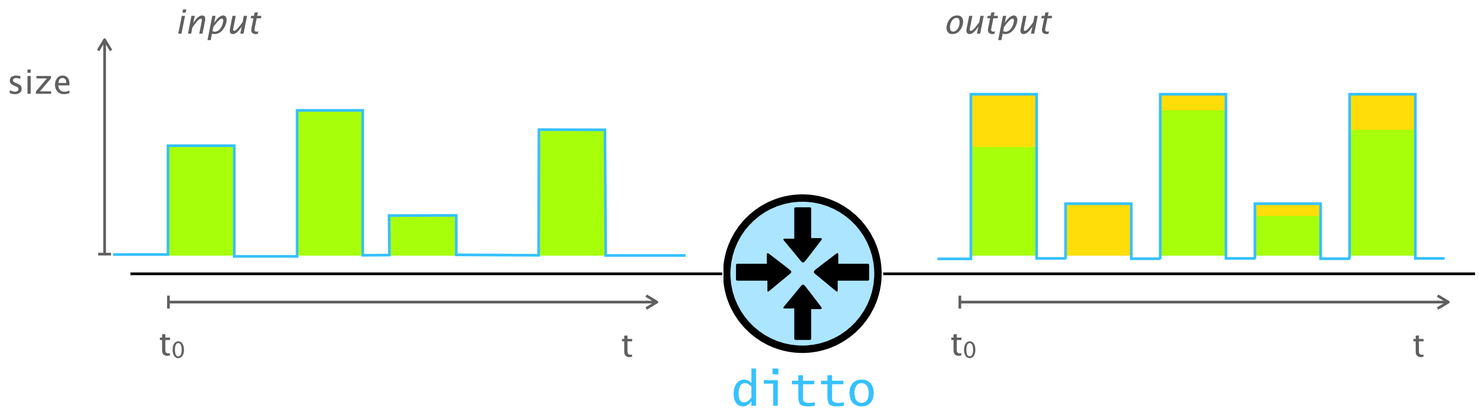




Computing efficient traffic patterns

Traffic shaping in the data plane

Experimental results

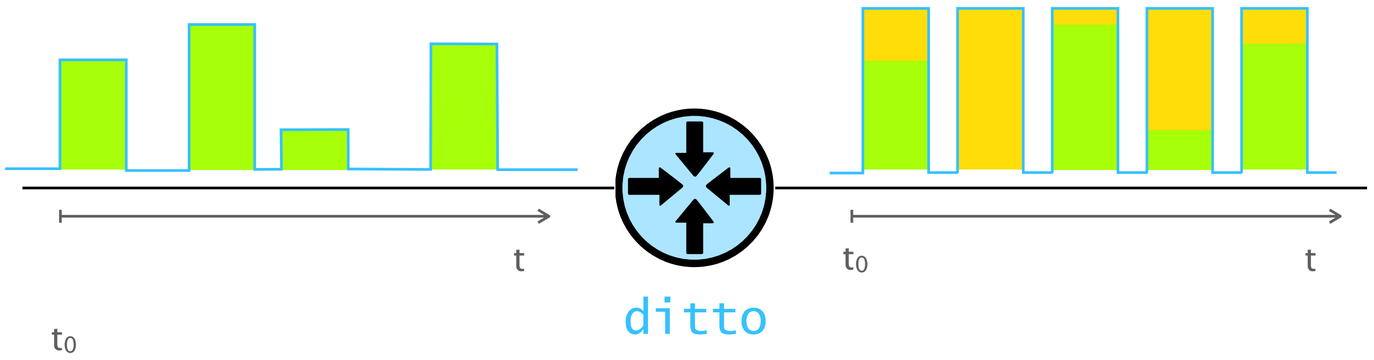


Computing efficient traffic patterns

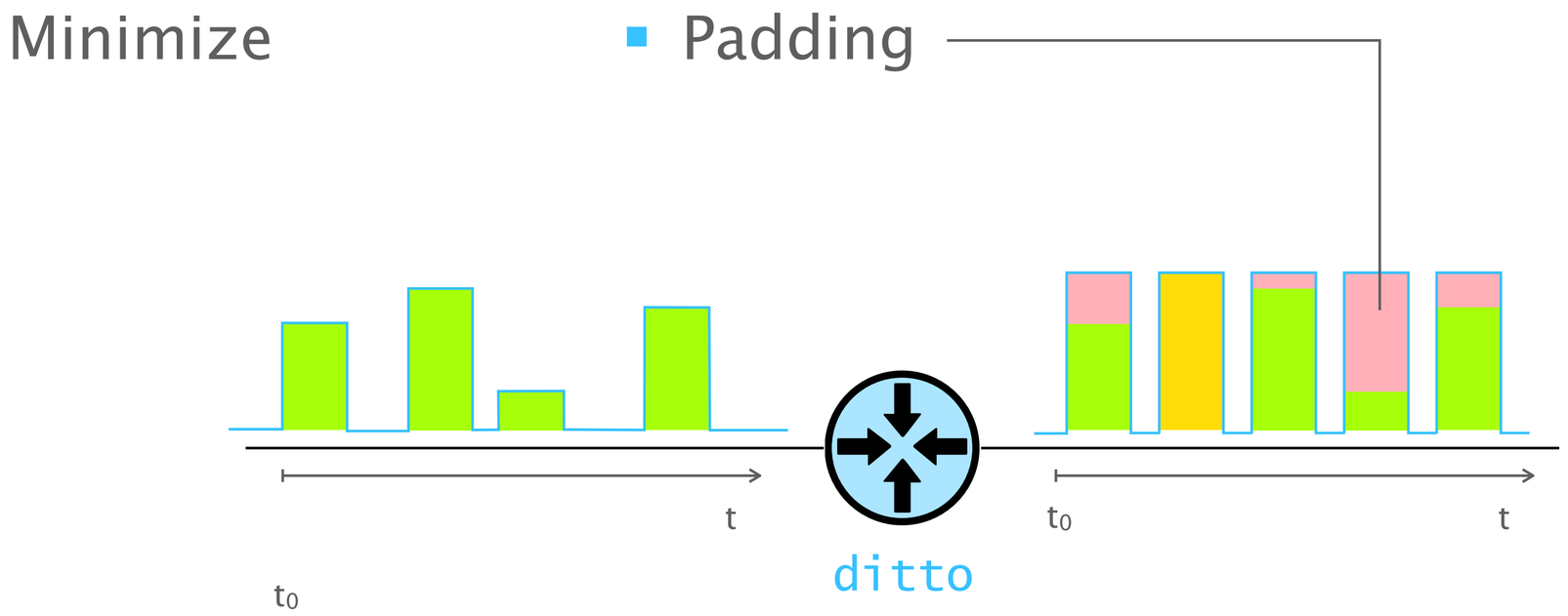
Traffic shaping in the data plane

Experimental results

An efficient pattern minimizes padding and chaff packets



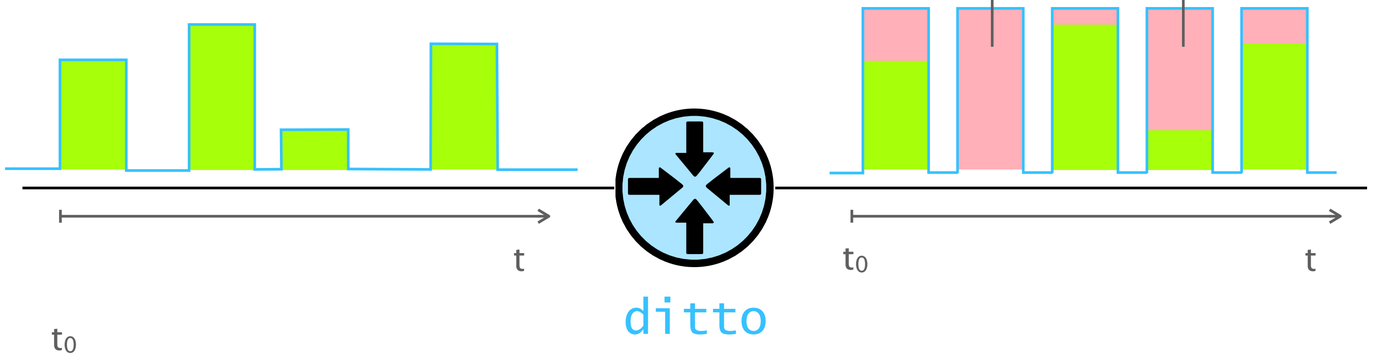
An efficient pattern minimizes padding and chaff packets



An efficient pattern minimizes padding and chaff packets

Minimize

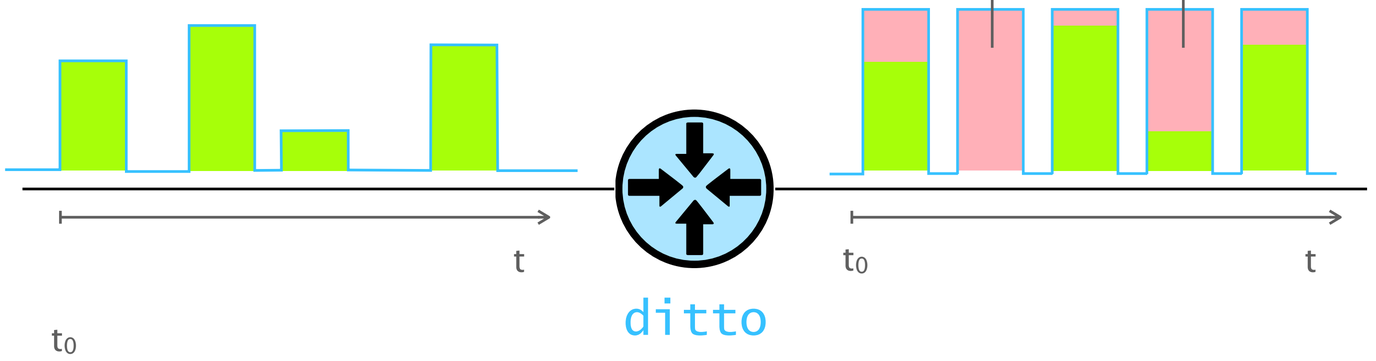
- Padding
- Chaff packets



An efficient pattern minimizes padding and chaff packets

Minimize

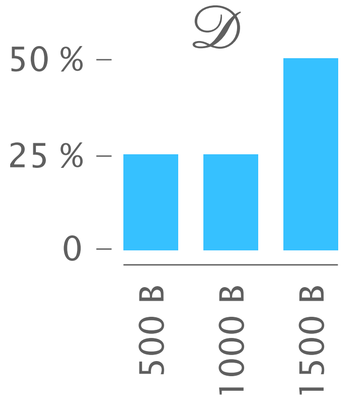
- Padding
- Chaff packets



with a repeating pattern

The pattern should fit to the traffic that is expected in the protected network

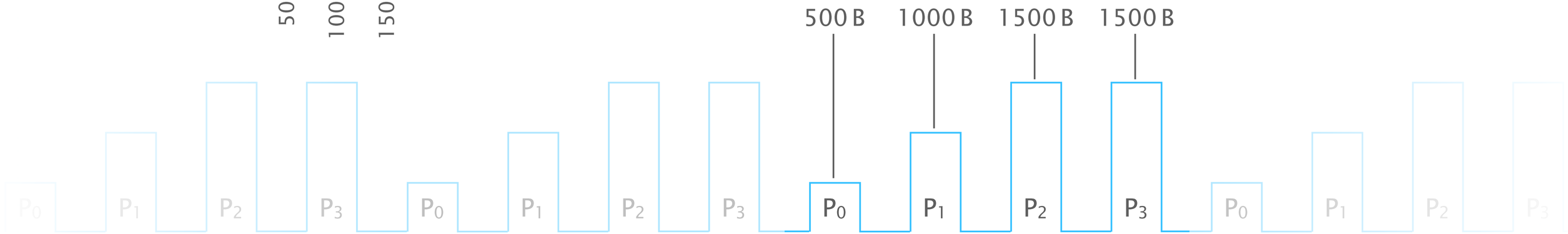
- Input: Traffic distribution
  e.g., from own recording

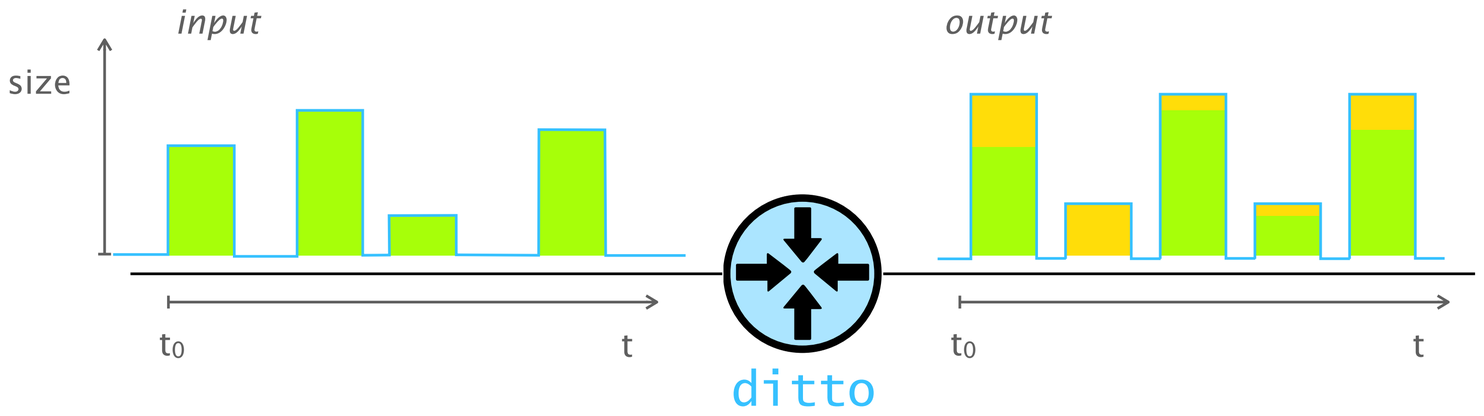


- Output: Pattern states
  for a given pattern length L

$$P_i = \text{percentile}_{(i+1)\cdot 100/L}\,\mathscr{D}$$

$$i \in [0,1,\ldots,L-1]$$





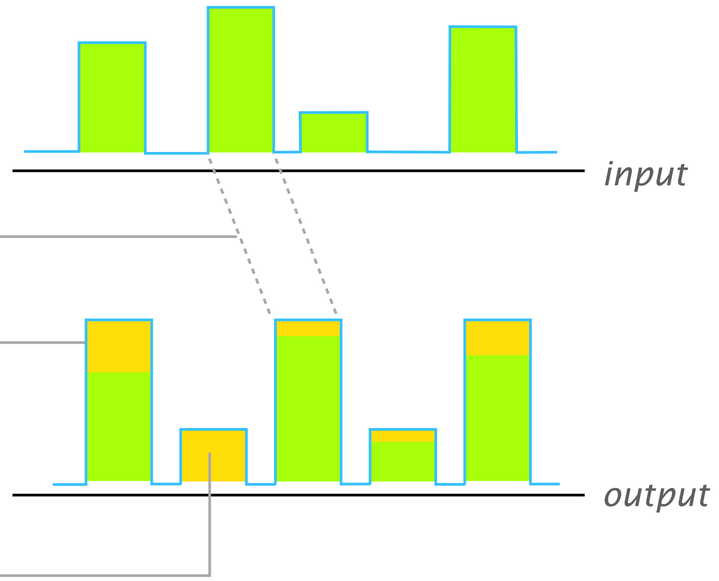
Computing efficient traffic patterns

Traffic shaping in the data plane

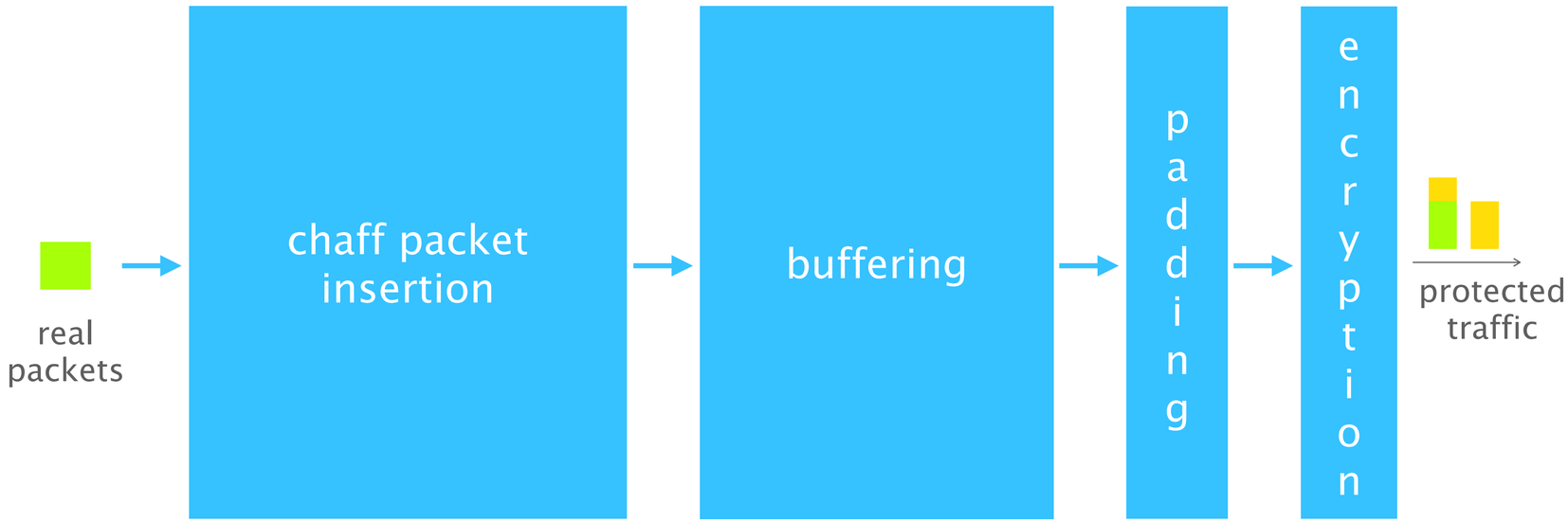
Experimental results

ditto uses three operations to enforce the pattern at line rate

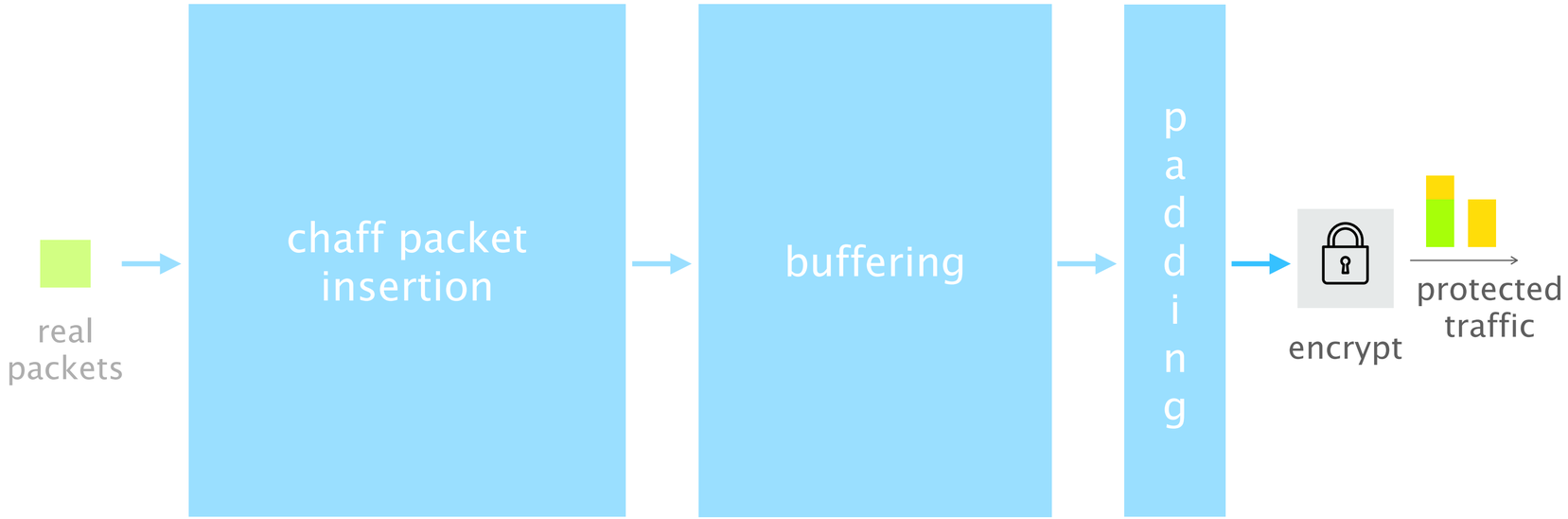
- **Buffering**
  until a packet fits in the pattern
- **Padding**
  to make packets larger
- **Chaff packet insertion**
  to fill gaps without real traffic



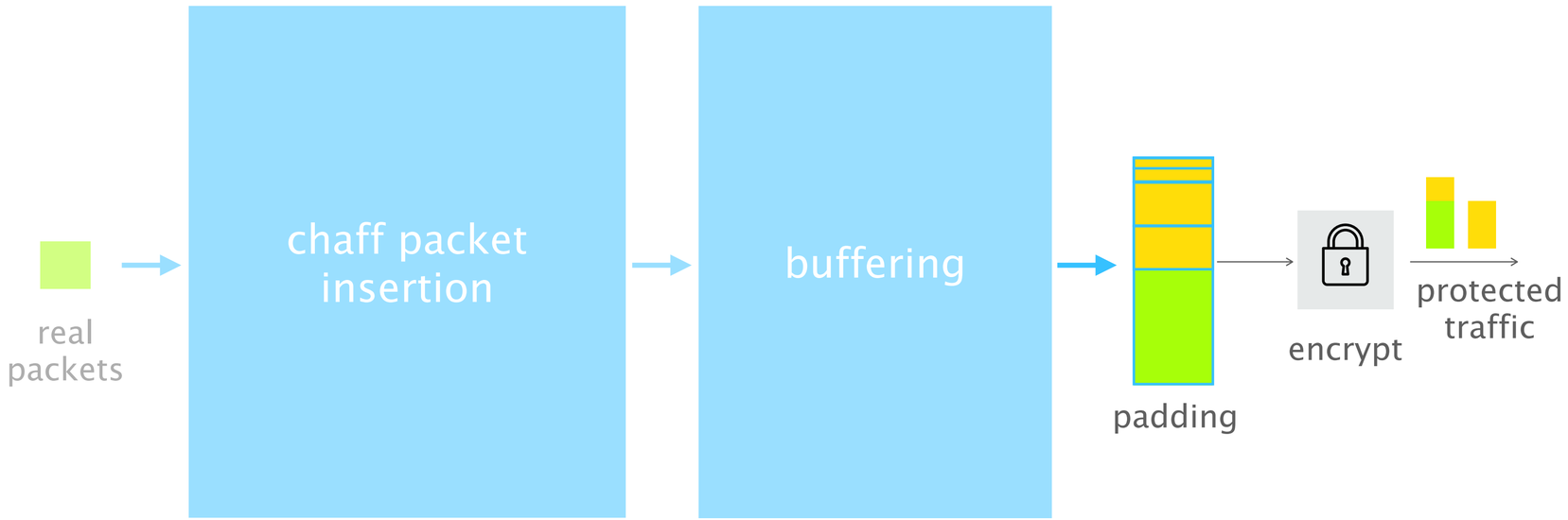
At a high level, ditto consists of 4 building blocks



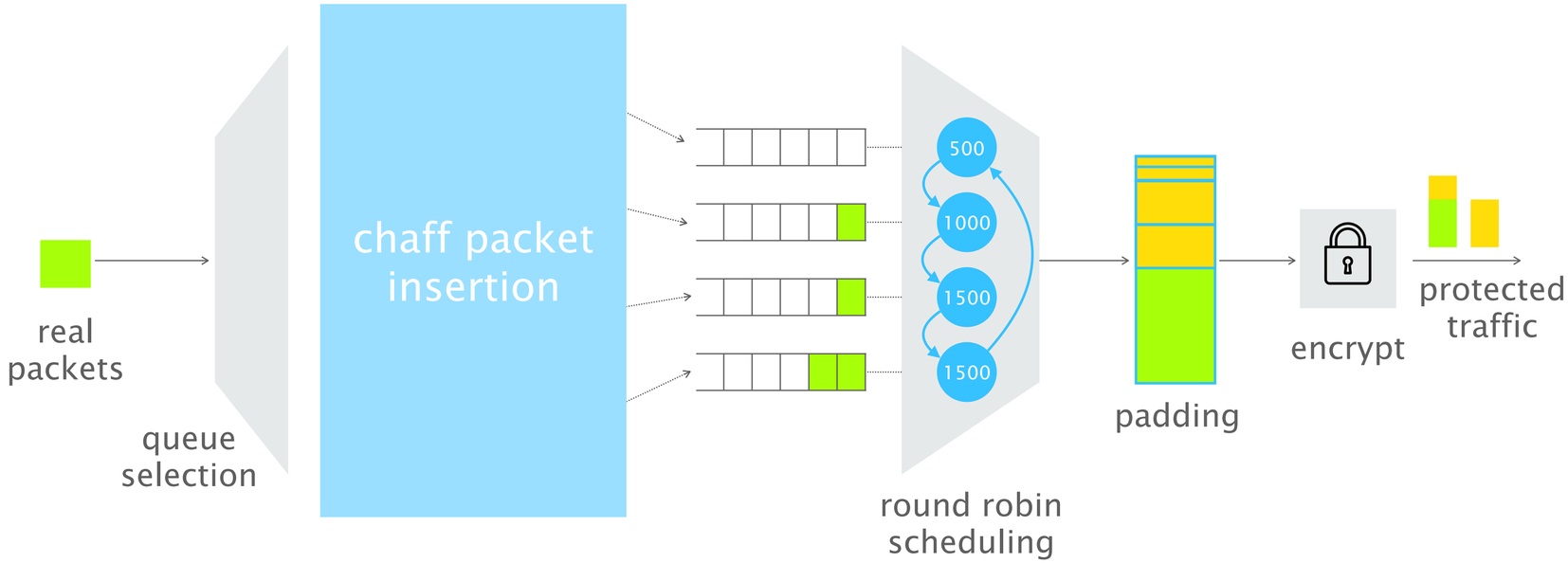
ditto sends traffic over encrypted tunnels (e.g., using MACsec)



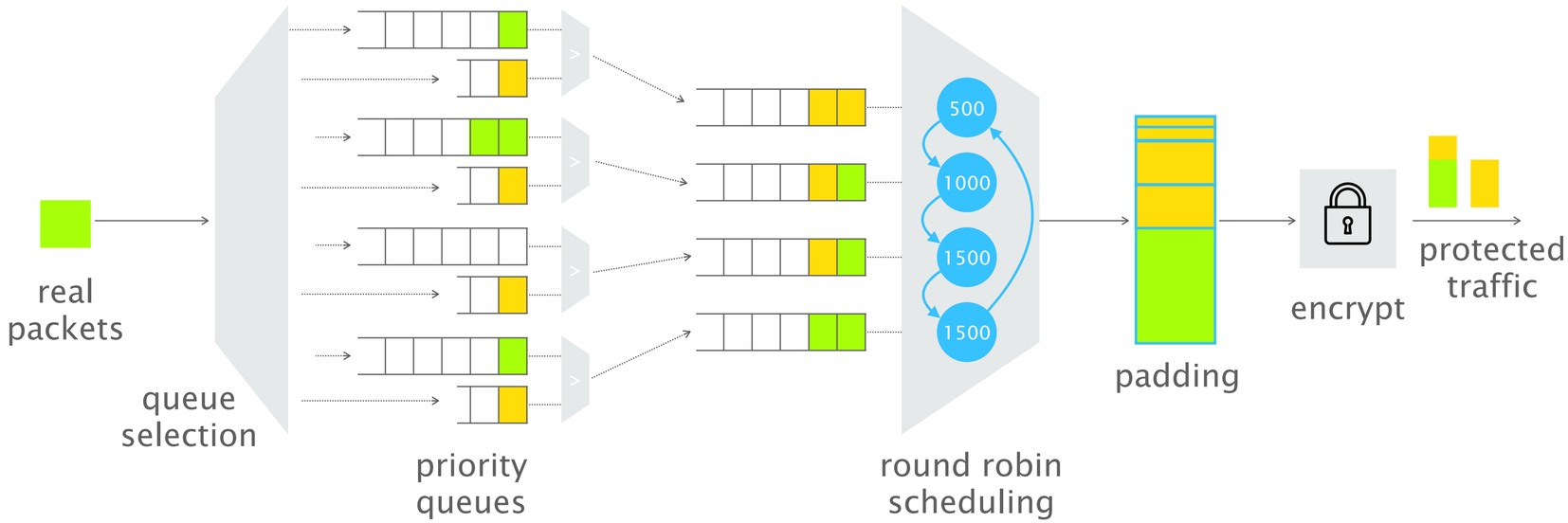
ditto pads packets by adding custom headers



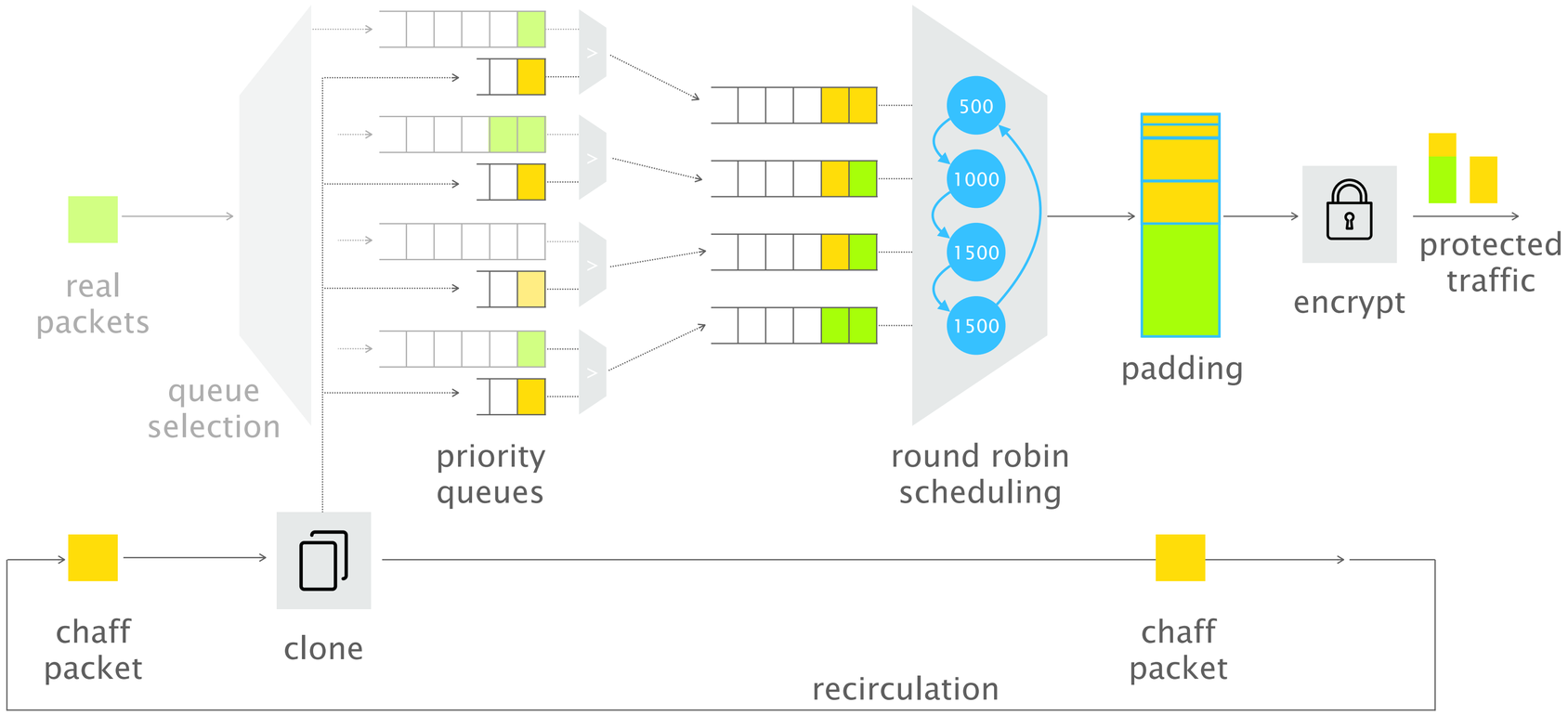
ditto uses round-robin scheduling to enforce the pattern



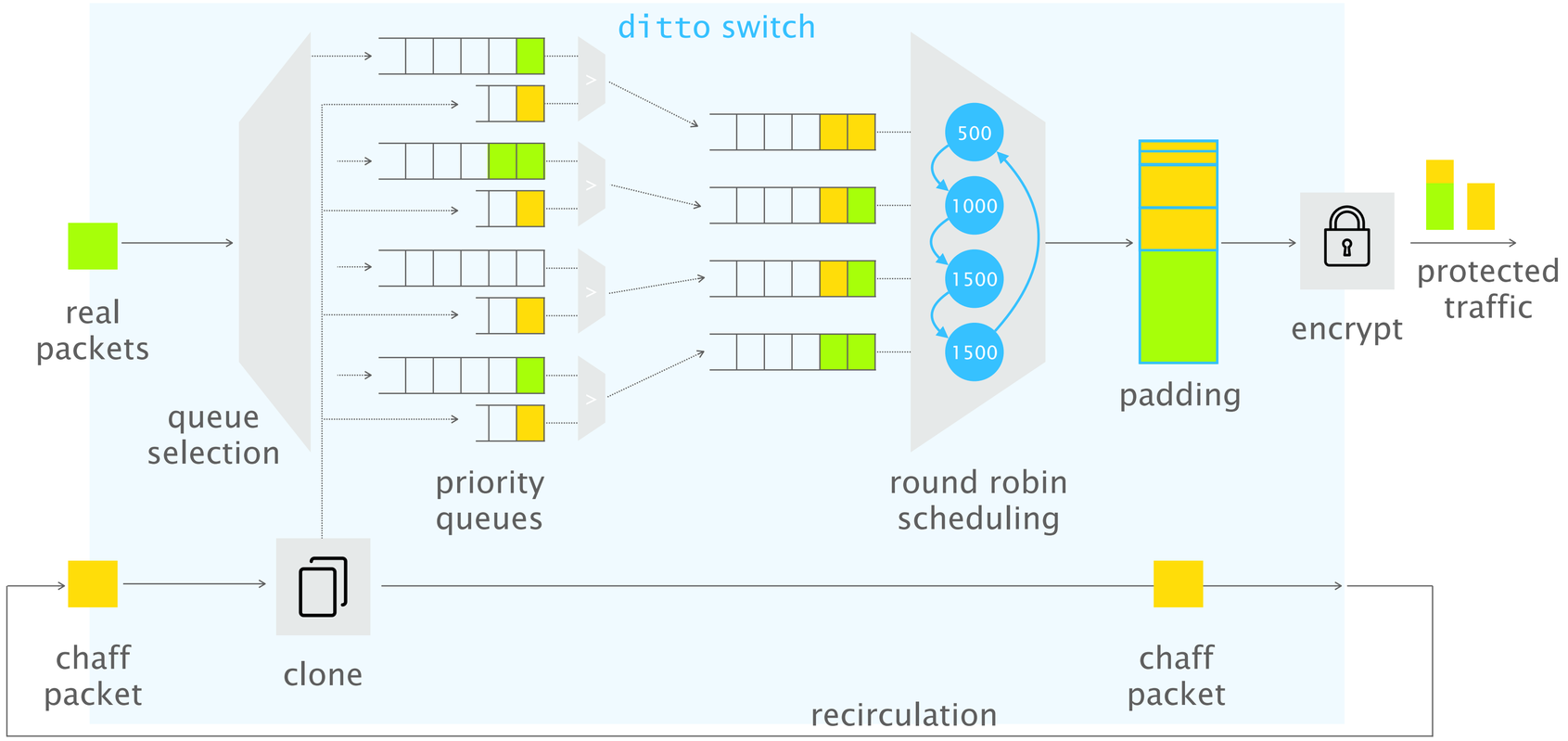
ditto uses priority queues to mix real and chaff packets



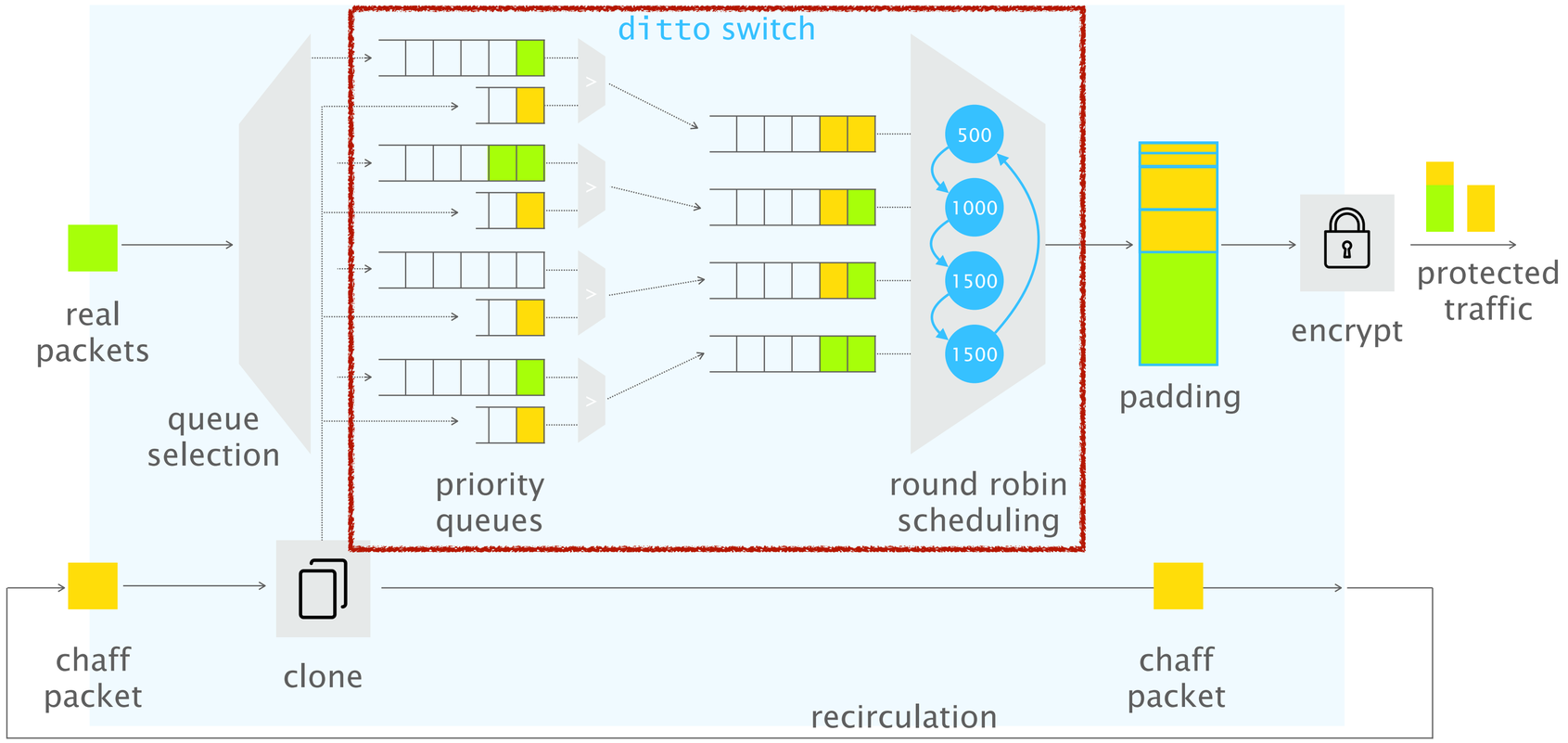
ditto generates chaff packets by recirculating and cloning them



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



ditto protects against an eavesdropper and provides three security properties

- Volume anonymity
  Attacker cannot determine the size of packets and flows

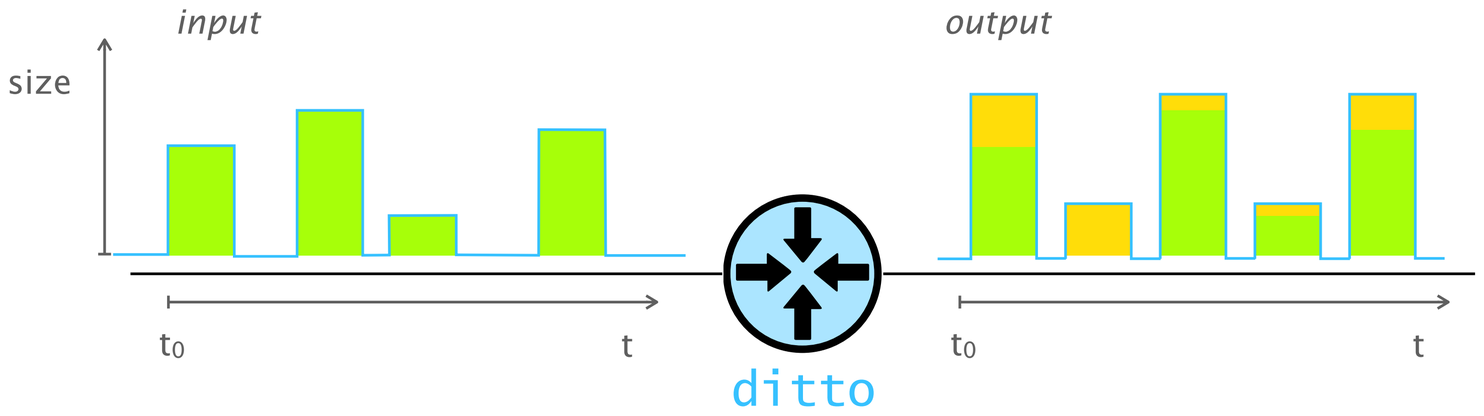
  Traffic always follows the pattern, which makes the volume constant

- Timing anonymity
  Attacker cannot determine the timing between two packets

  Traffic is sent at a fixed rate according to the pattern

- Path anonymity
  Attacker cannot track packets across multiple protected links

  Traffic is encrypted per link and the volume is always the same



Computing efficient traffic patterns

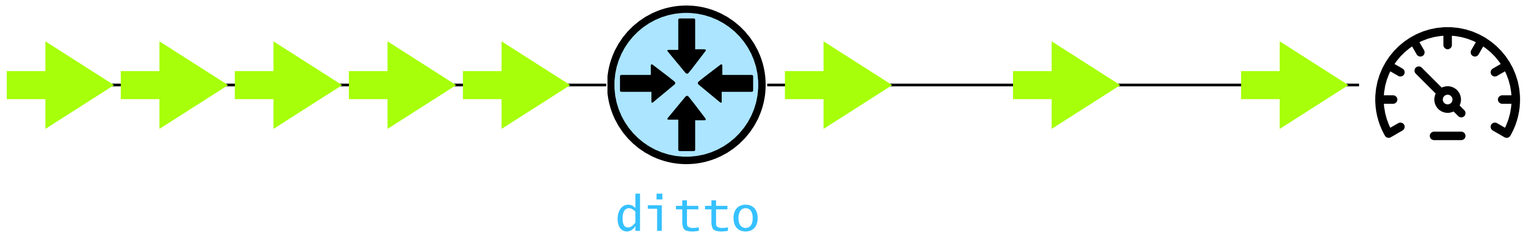
Traffic shaping in the data plane

Experimental results

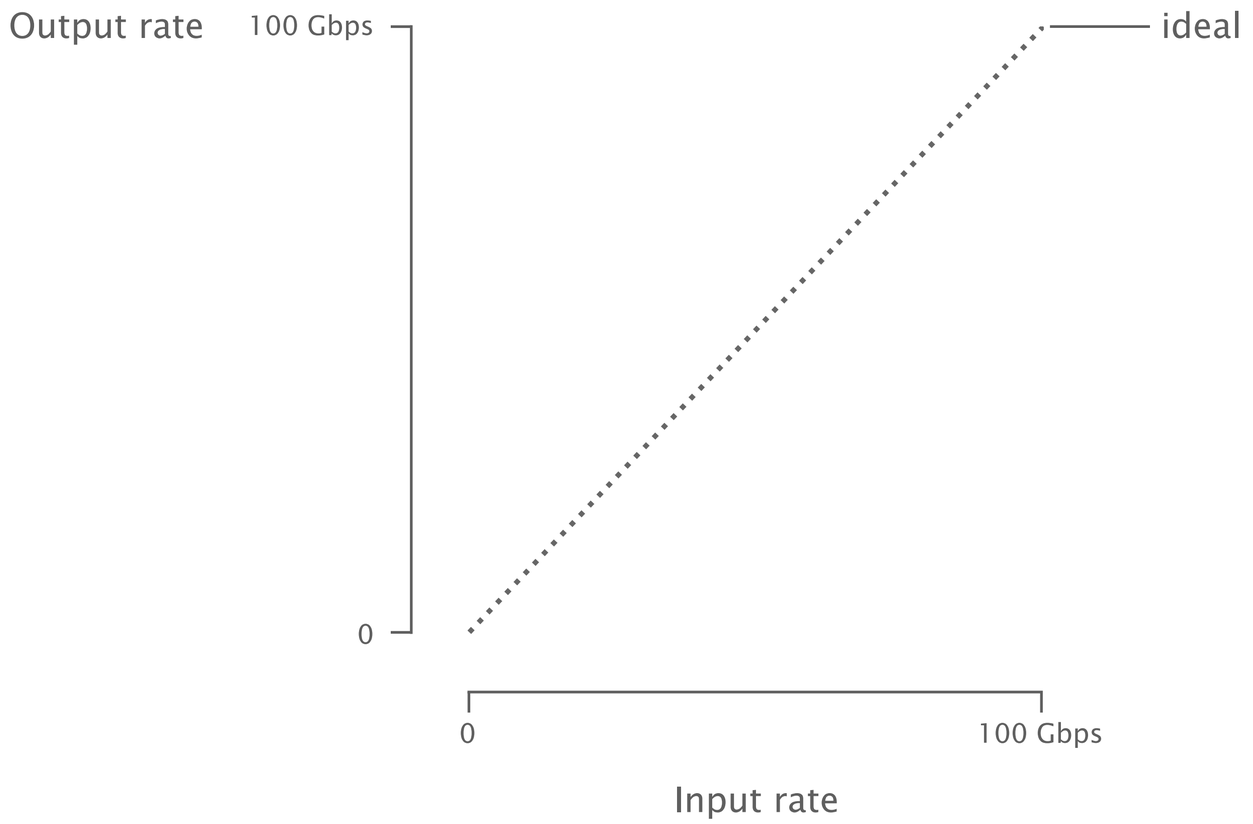
Our evaluation shows that ditto performs well and is secure

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

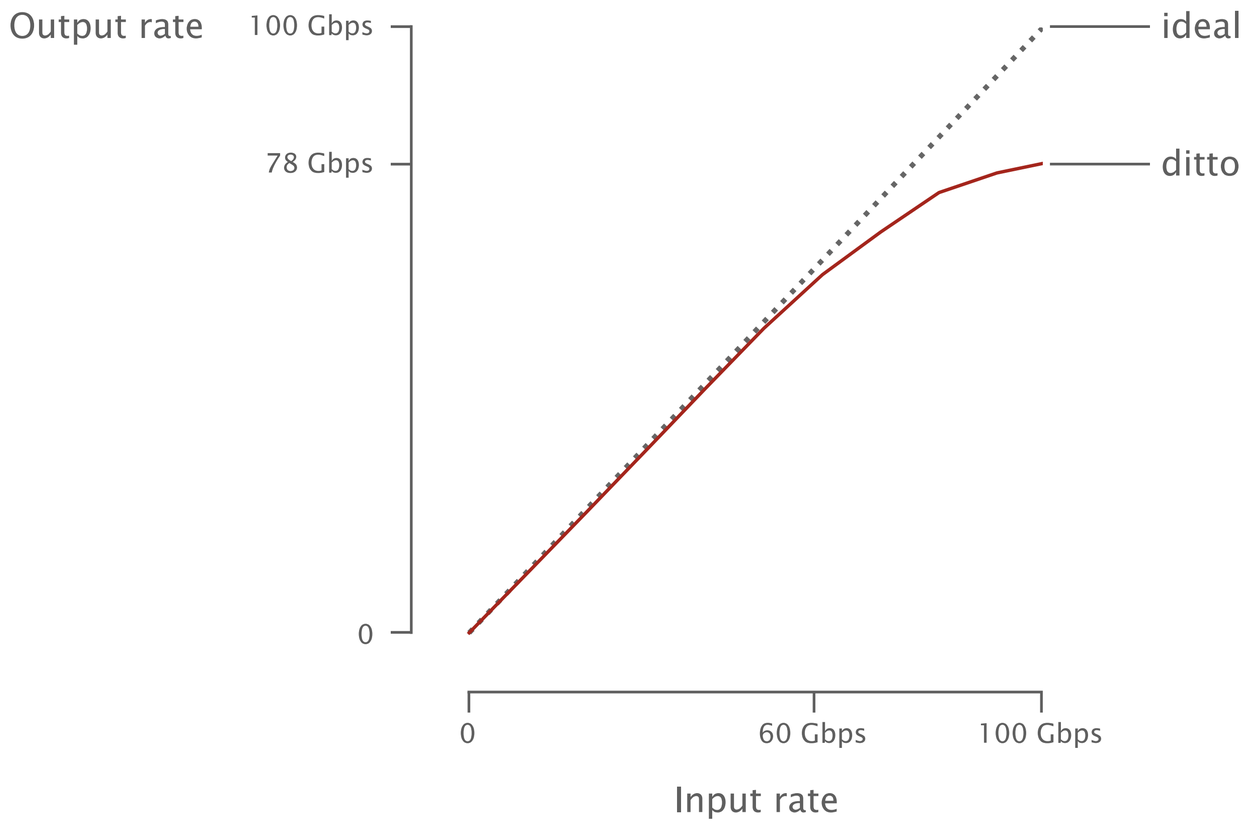
This experiment measures how much throughput ditto can achieve



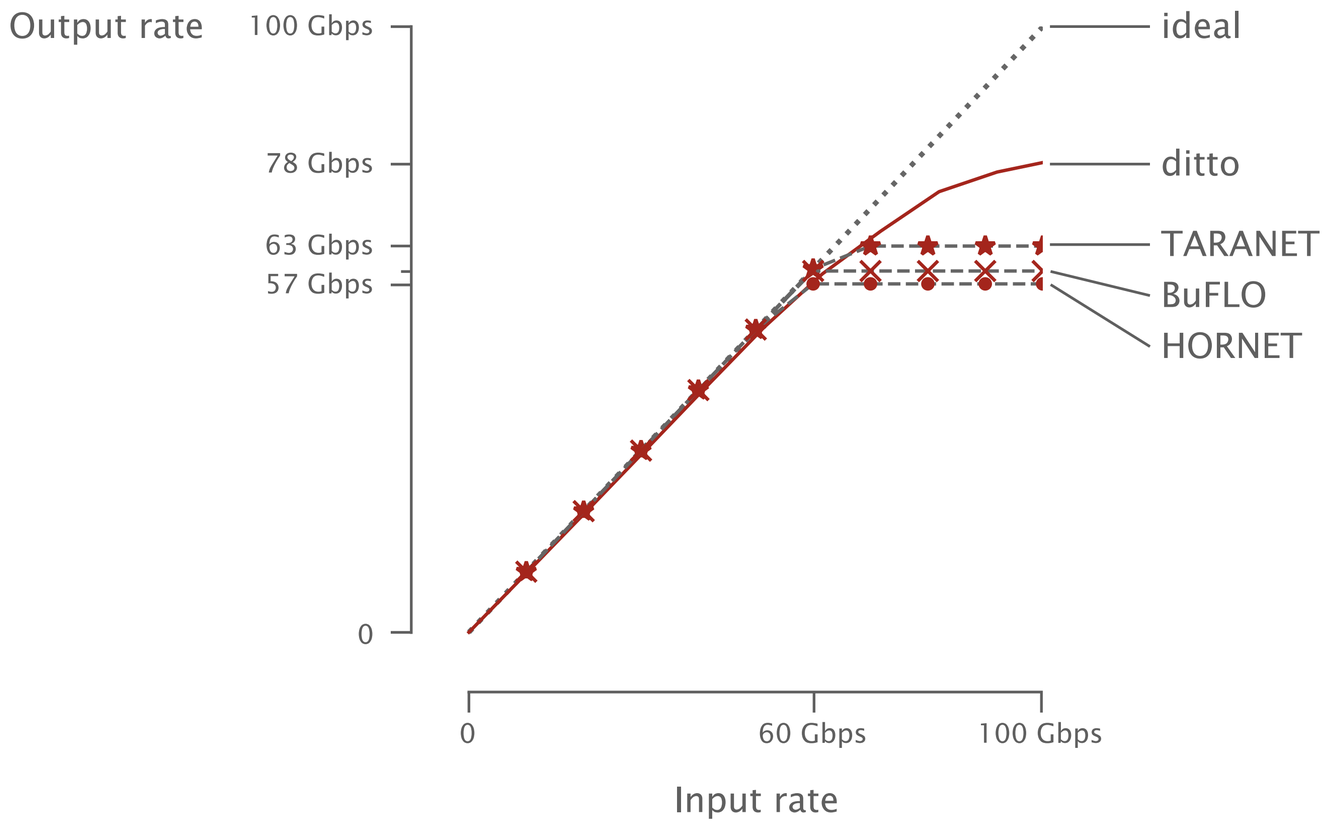
Ideally, the output rate equals the input rate



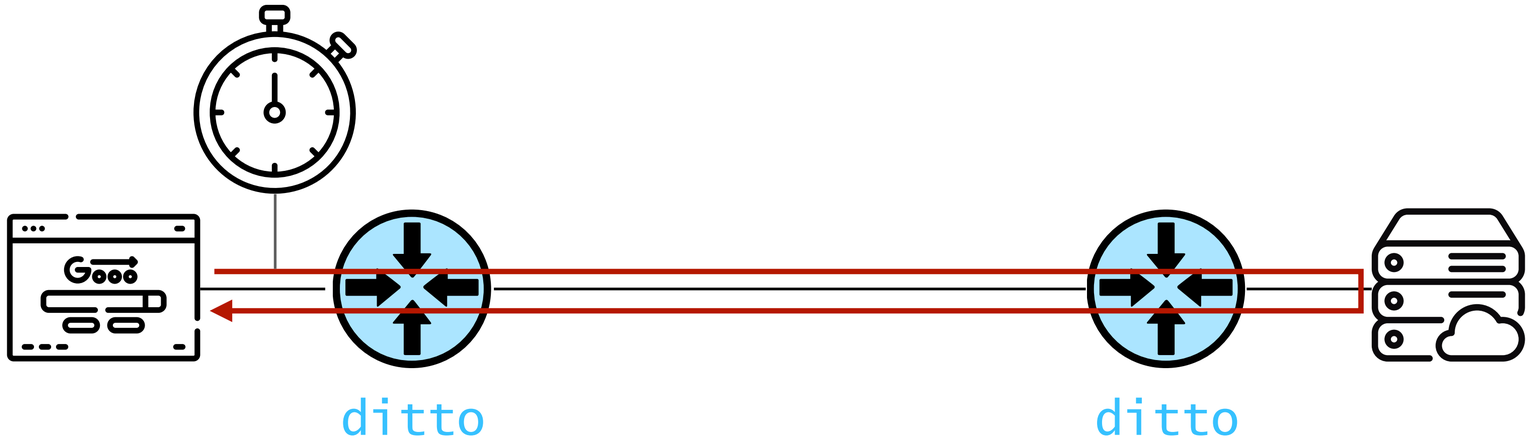
ditto reaches up to 78 Gbps with Internet backbone traffic on a 100 Gbps link



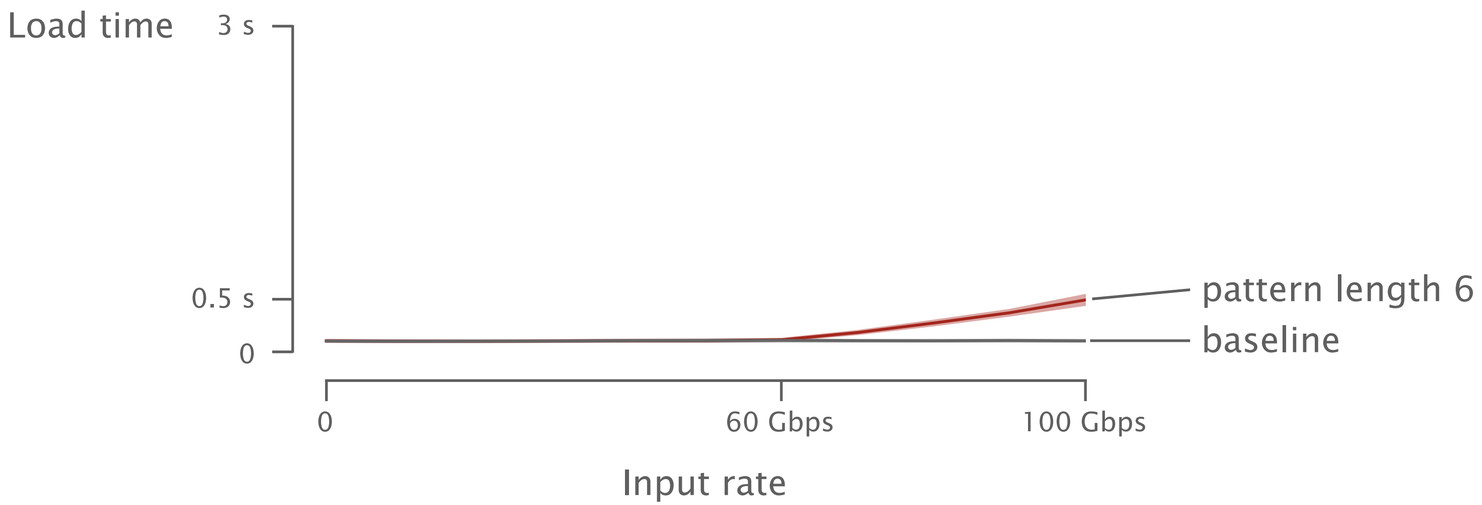
ditto performs significantly better than (idealized) related work



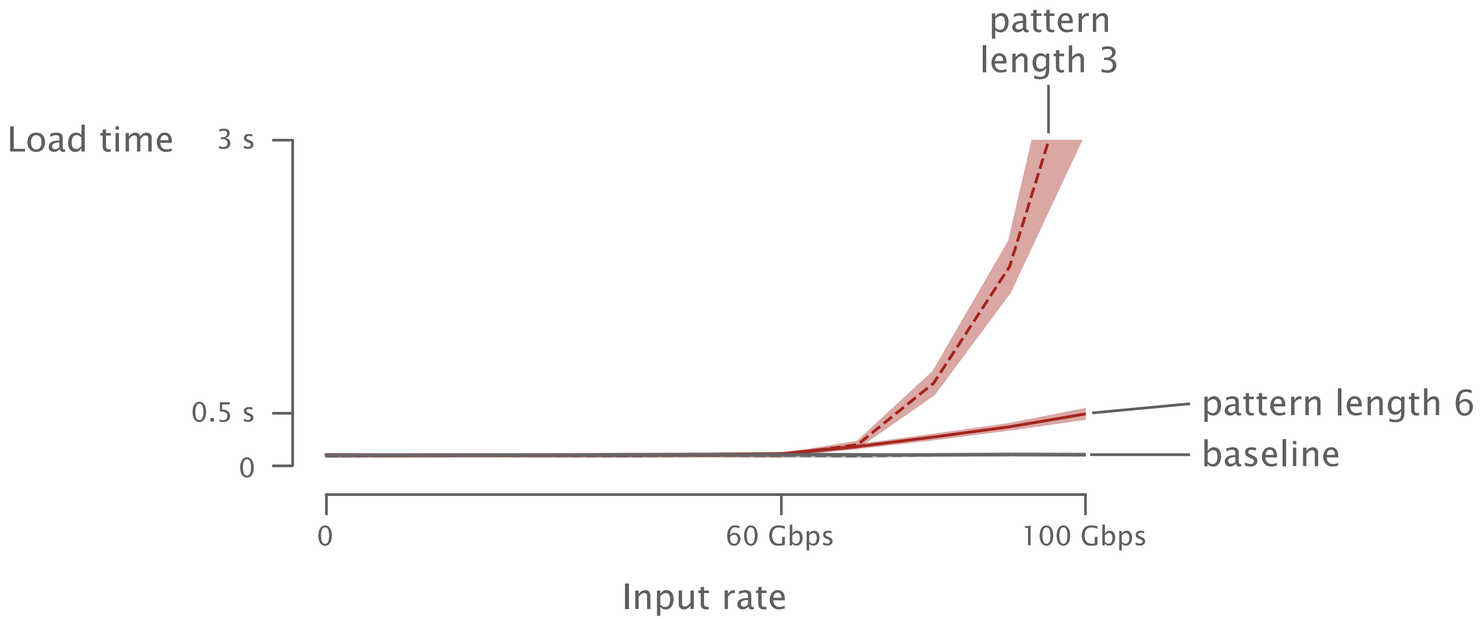
This experiment measures how much impact ditto has on applications



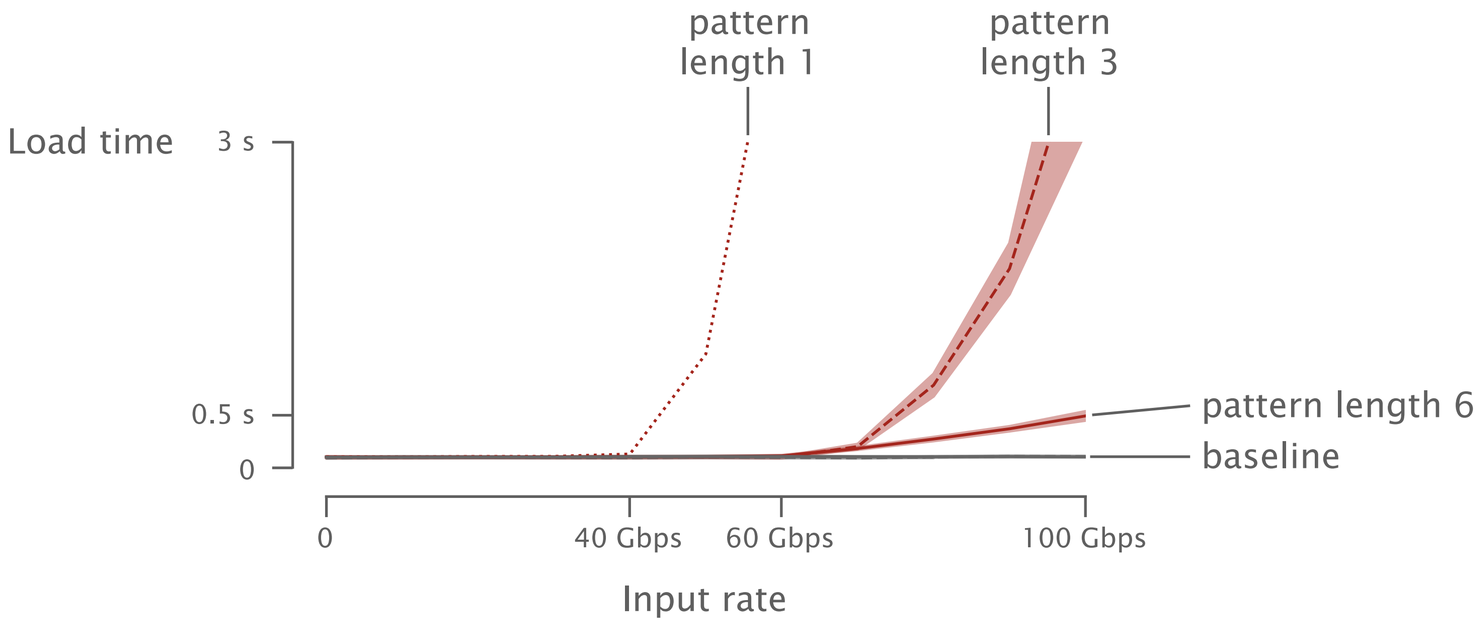
ditto does not affect the website load time up to 60 % load



Longer patterns achieve better performance



Longer patterns achieve better performance



ditto: WAN Traffic Obfuscation at Line Rate

github.com/nsg-ethz/ditto

- Security
  Traffic does not leak information

  ditto makes observed traffic independent from the actual traffic

- Performance
  WANs run at 100s of Gbps

  ditto reduces overhead by using efficient traffic patterns

- Deployability
  Infeasible to change all servers

  ditto runs in the network data plane at line rate

Roland Meier
meierrol@ethz.ch

ETH zürich

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse