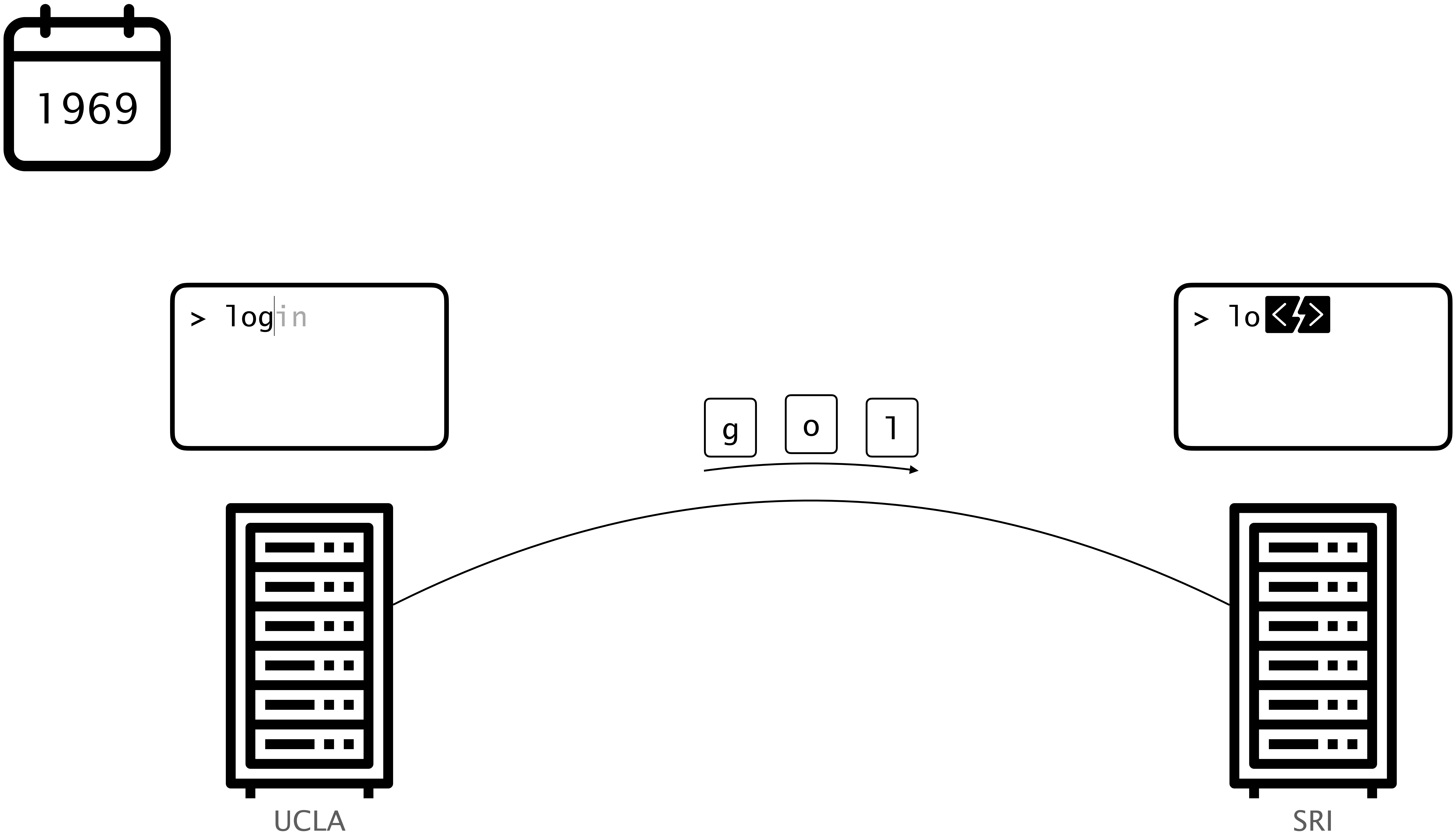
Improving Network Security through Obfuscation

Roland Meier

PhD Defense

Sept 23, 2022

ETH zürich


1969
> login
g
o
l
> lo
UCLA
SRI

Today, the Internet has more than 4 billion users — and not all of them have good intentions

Human traffic
36%

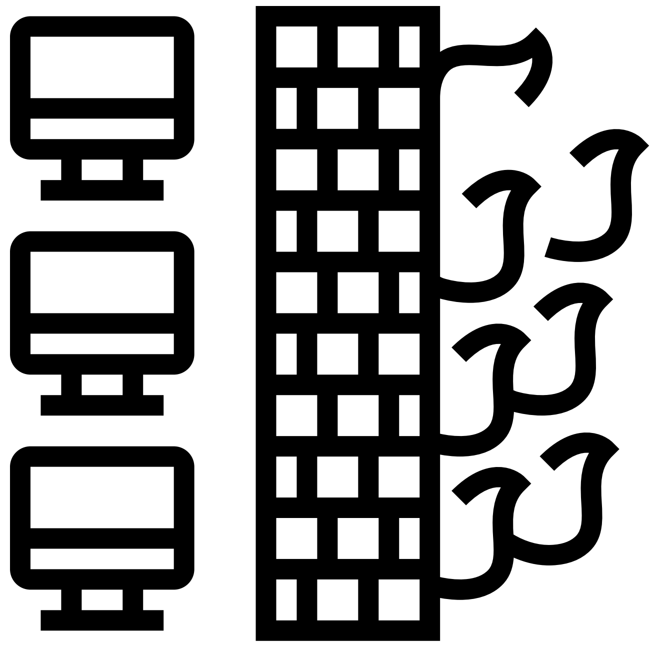
Good bots
25%

Bad bots
39%

[Barracuda]

When the threat landscape changed, new security features were added to existing protocols and algorithms

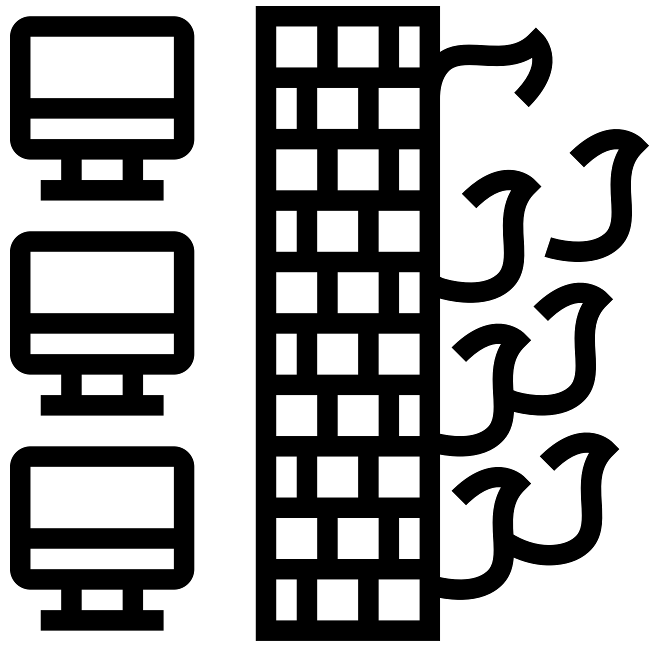
When the threat landscape changed, new security features were added to existing protocols and algorithms



Firewall

Prevents malicious traffic from reaching hosts

When the threat landscape changed, new security features were added to existing protocols and algorithms



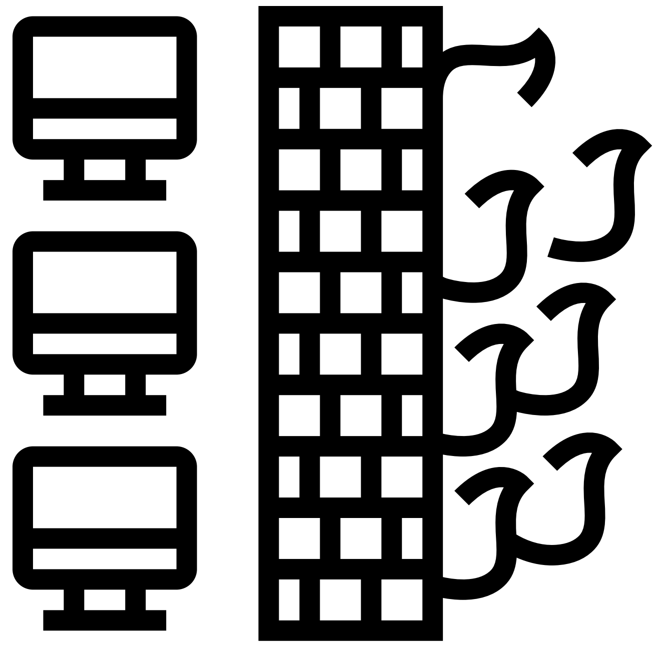
Firewall

Prevents malicious traffic from reaching hosts

Encryption

Prevents eavesdroppers from seeing the packet contents

When the threat landscape changed, new security features were added to existing protocols and algorithms



Firewall

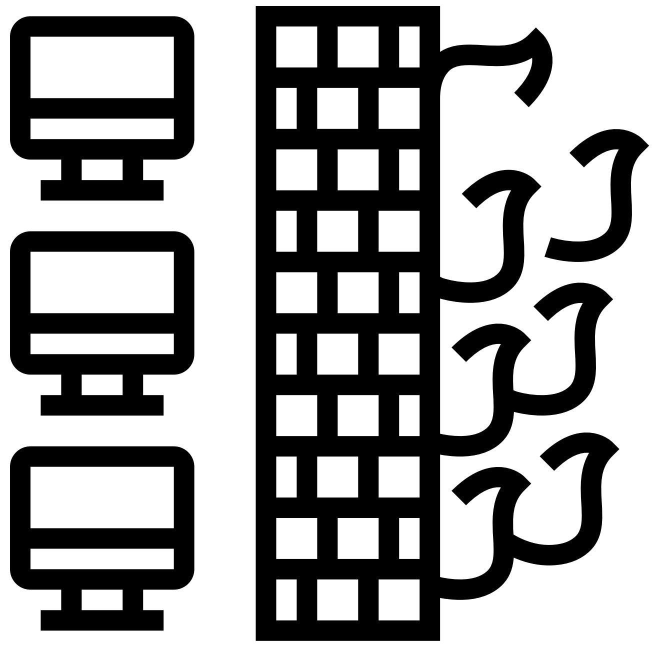
Prevents malicious traffic from reaching hosts

But malicious traffic can impair a host without reaching it

Encryption

Prevents eavesdroppers from seeing the packet contents

When the threat landscape changed, new security features were added to existing protocols and algorithms



Firewall

Prevents malicious traffic from reaching hosts

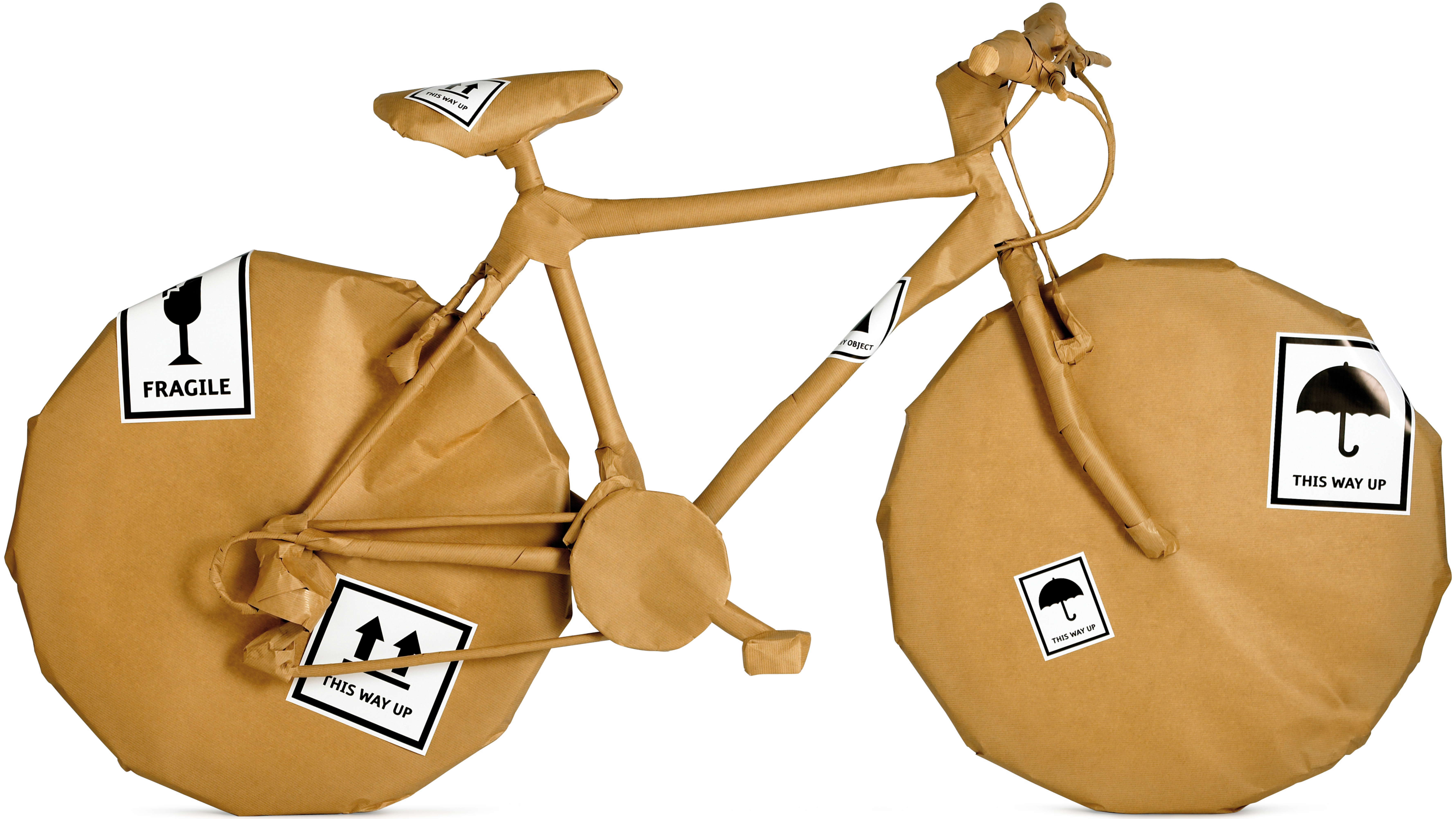
But malicious traffic can impair a host without reaching it

Encryption

Prevents eavesdroppers from seeing the packet contents

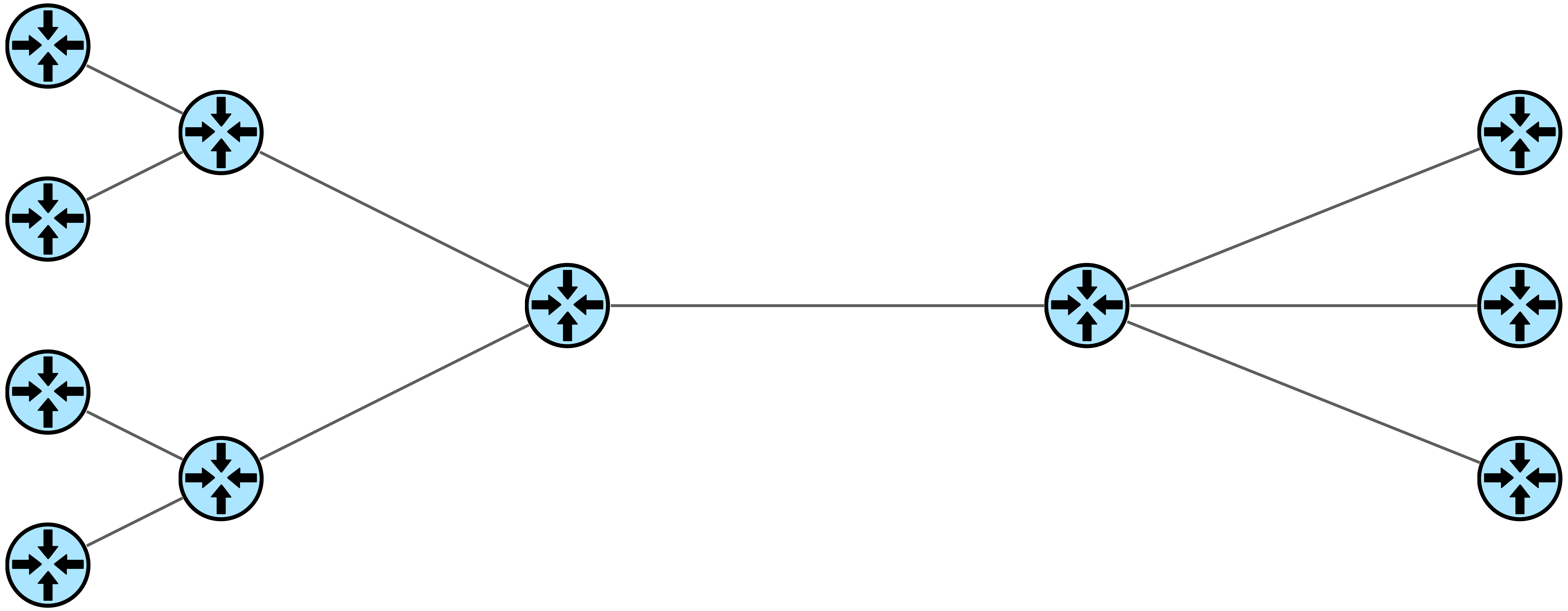
But metadata still reveals information about contents

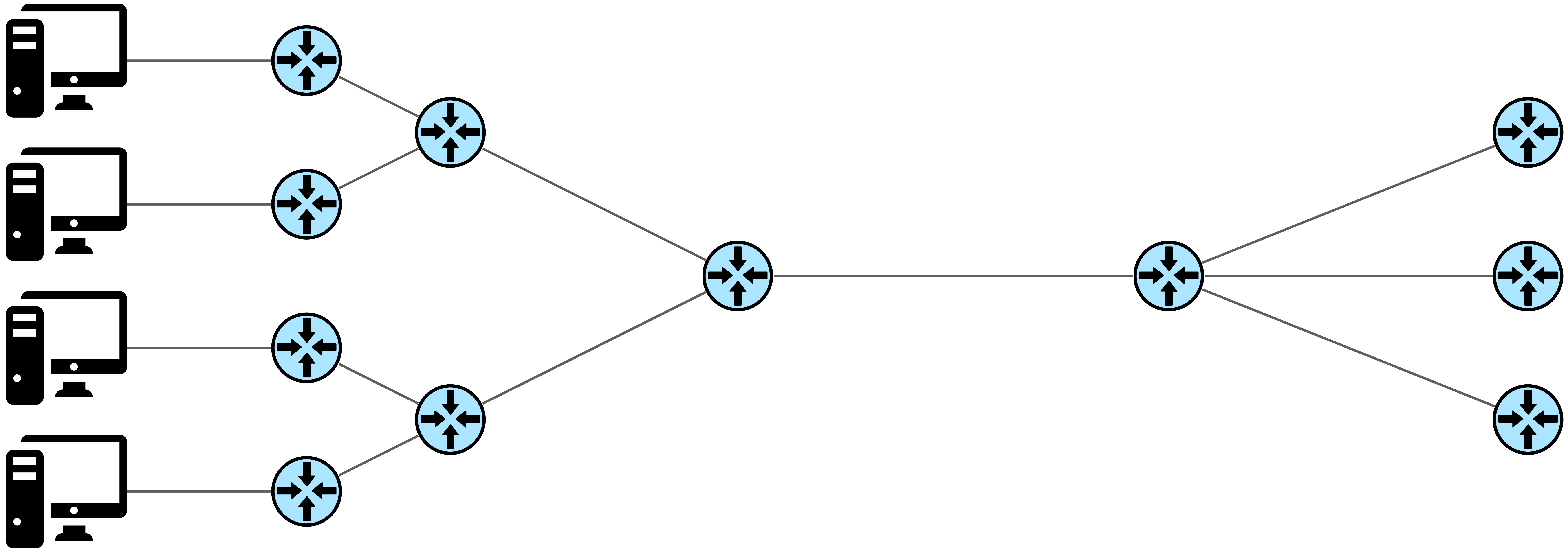
Encryption often hides the contents as much as this package does

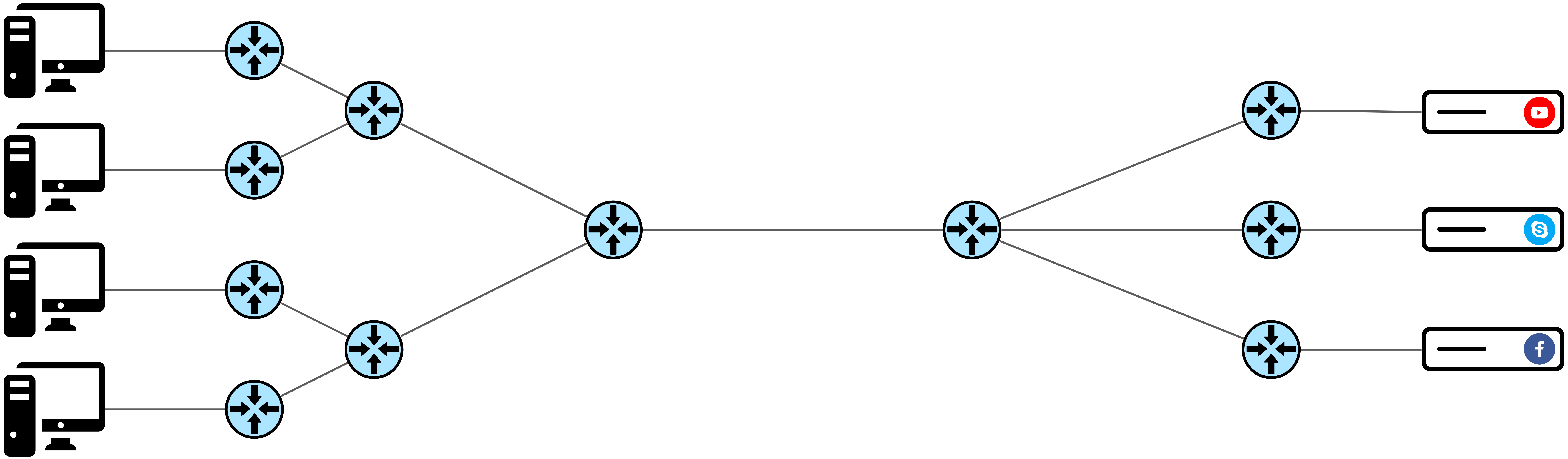


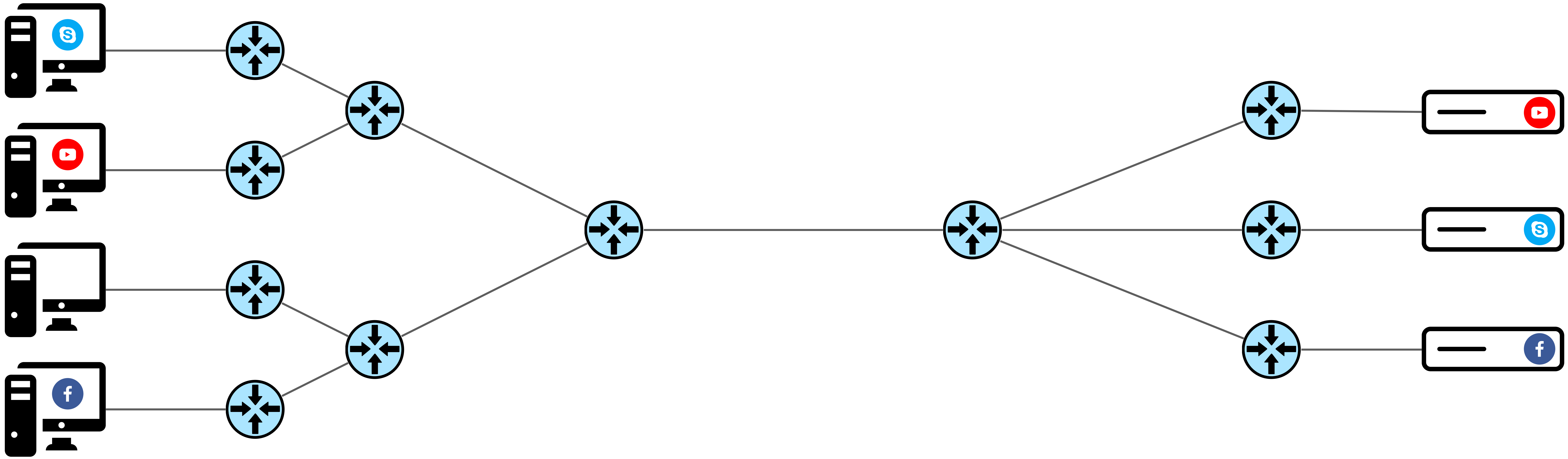
What we would like to have is rather something like this

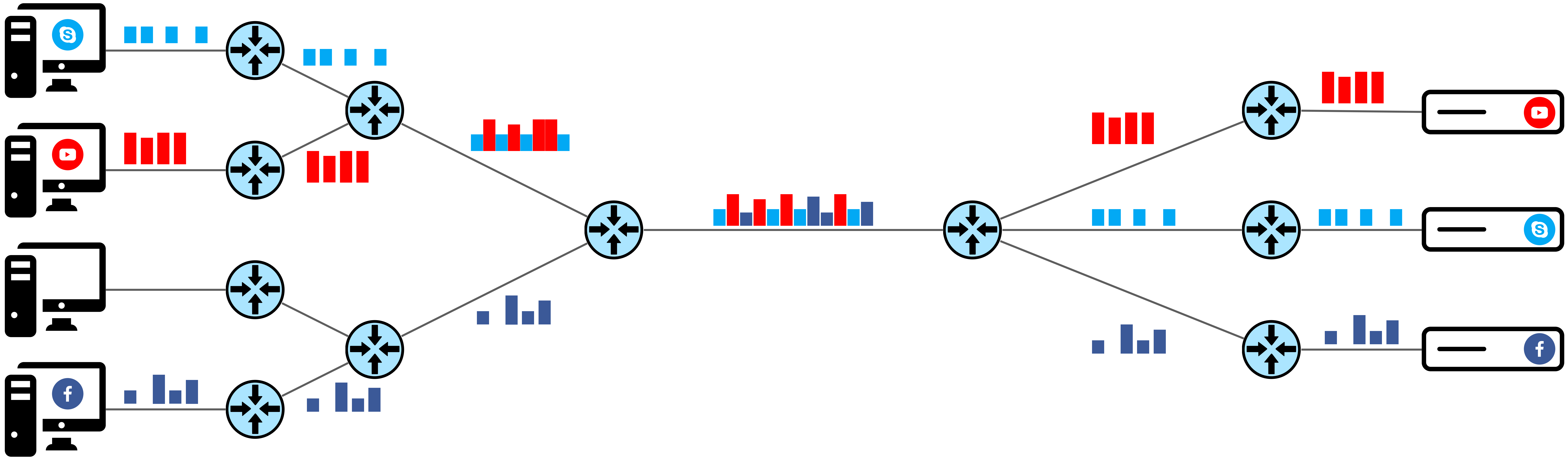
We can add obfuscation to change from one packaging to the other







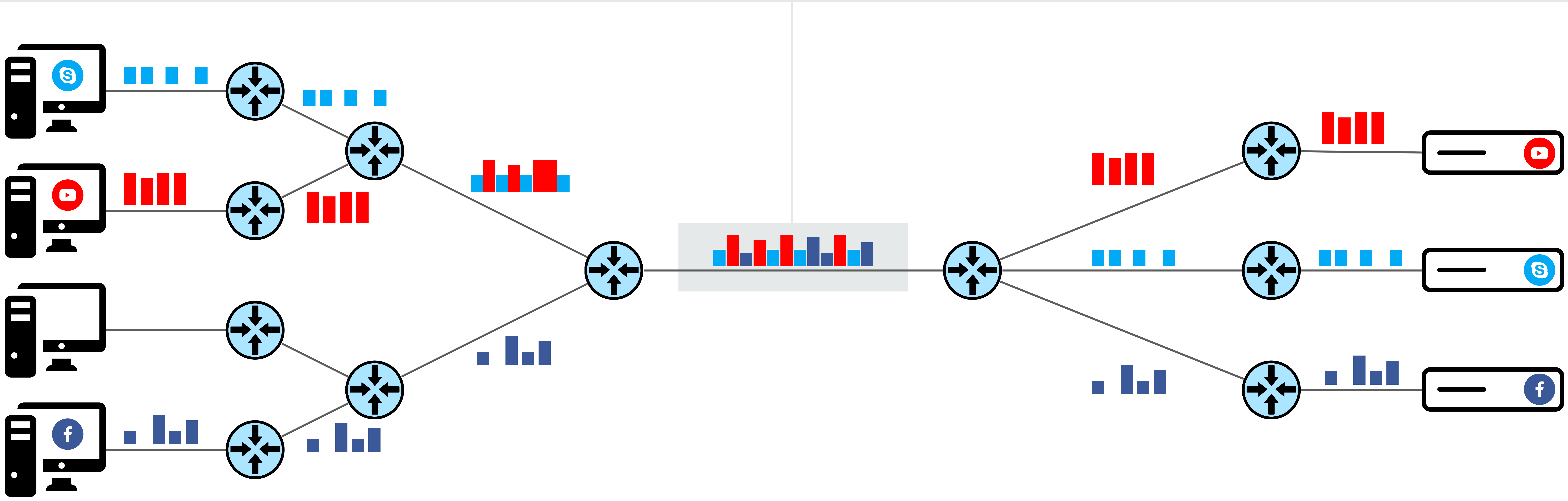




Problem #1

Traffic concentrates on one link

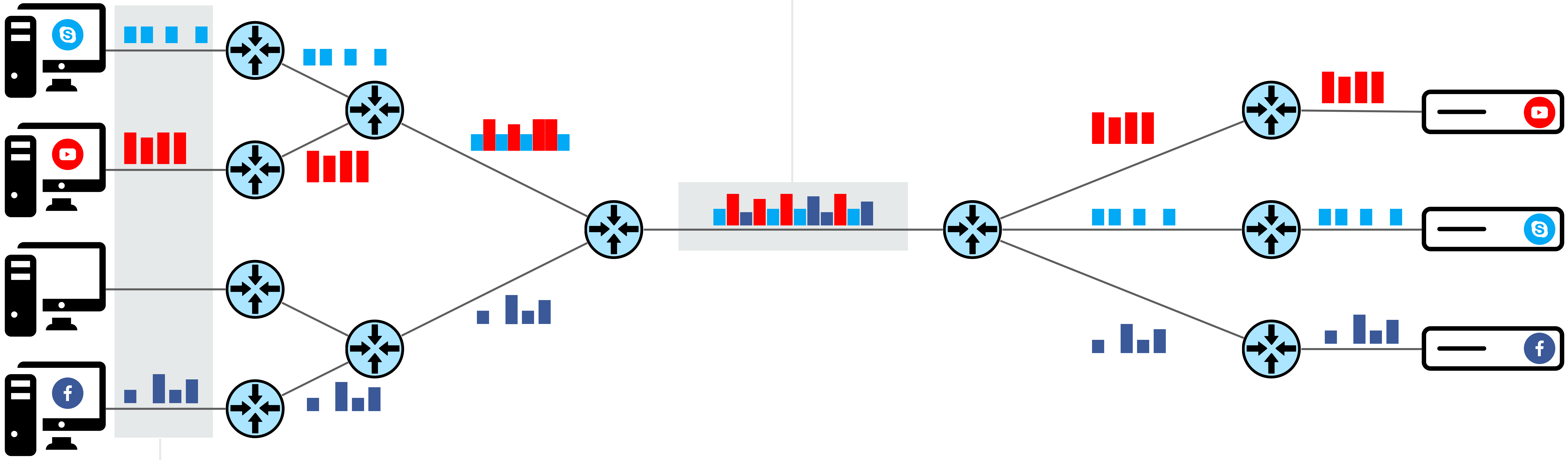
Vulnerable to denial-of-service attacks



Problem #1

Traffic concentrates on one link

Vulnerable to denial-of-service attacks



Encryption does not hide packet sizes and timings

Vulnerable to traffic-analysis attacks

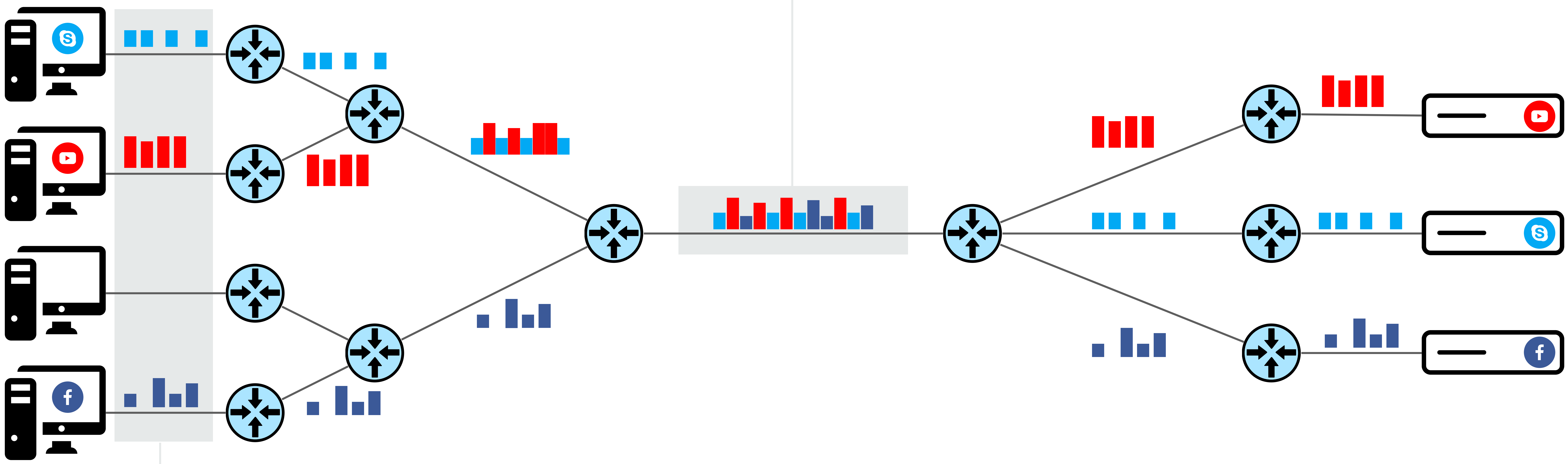
Problem #2

Problem #1

Traffic concentrates on one link

Vulnerable to denial-of-service attacks

NetHide prevents these attacks by obfuscating the topology



Encryption does not hide packet sizes and timings

Vulnerable to traffic-analysis attacks

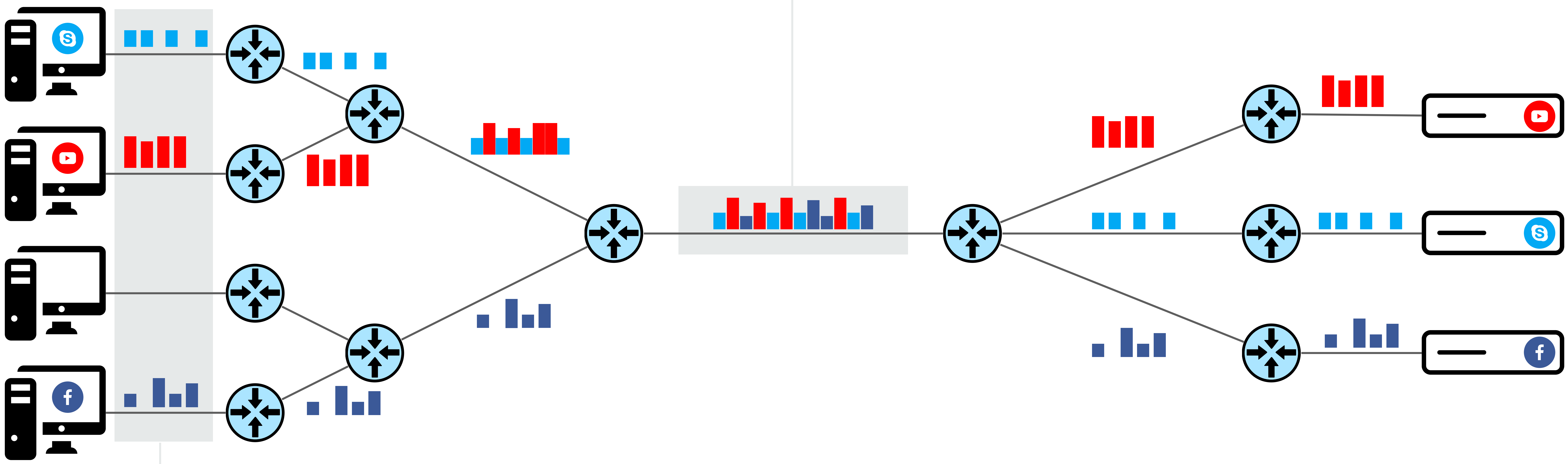
Problem #2

Problem #1

Traffic concentrates on one link

Vulnerable to denial-of-service attacks

NetHide prevents these attacks by obfuscating the topology



Encryption does not hide packet sizes and timings

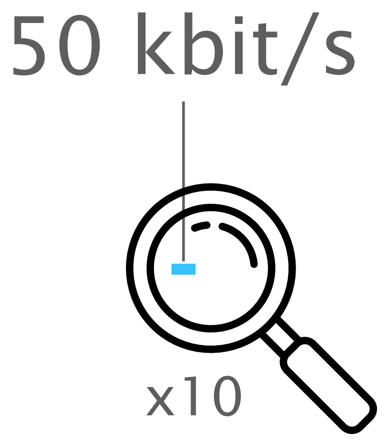
Vulnerable to traffic-analysis attacks

ditto prevents these attacks by obfuscating the traffic

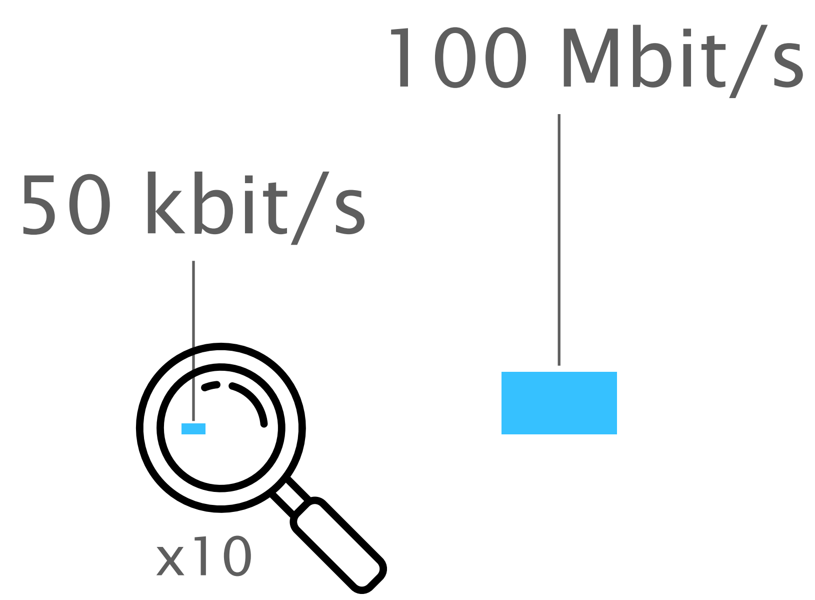
Problem #2

Not only the number of network users has changed, the traffic volume has too

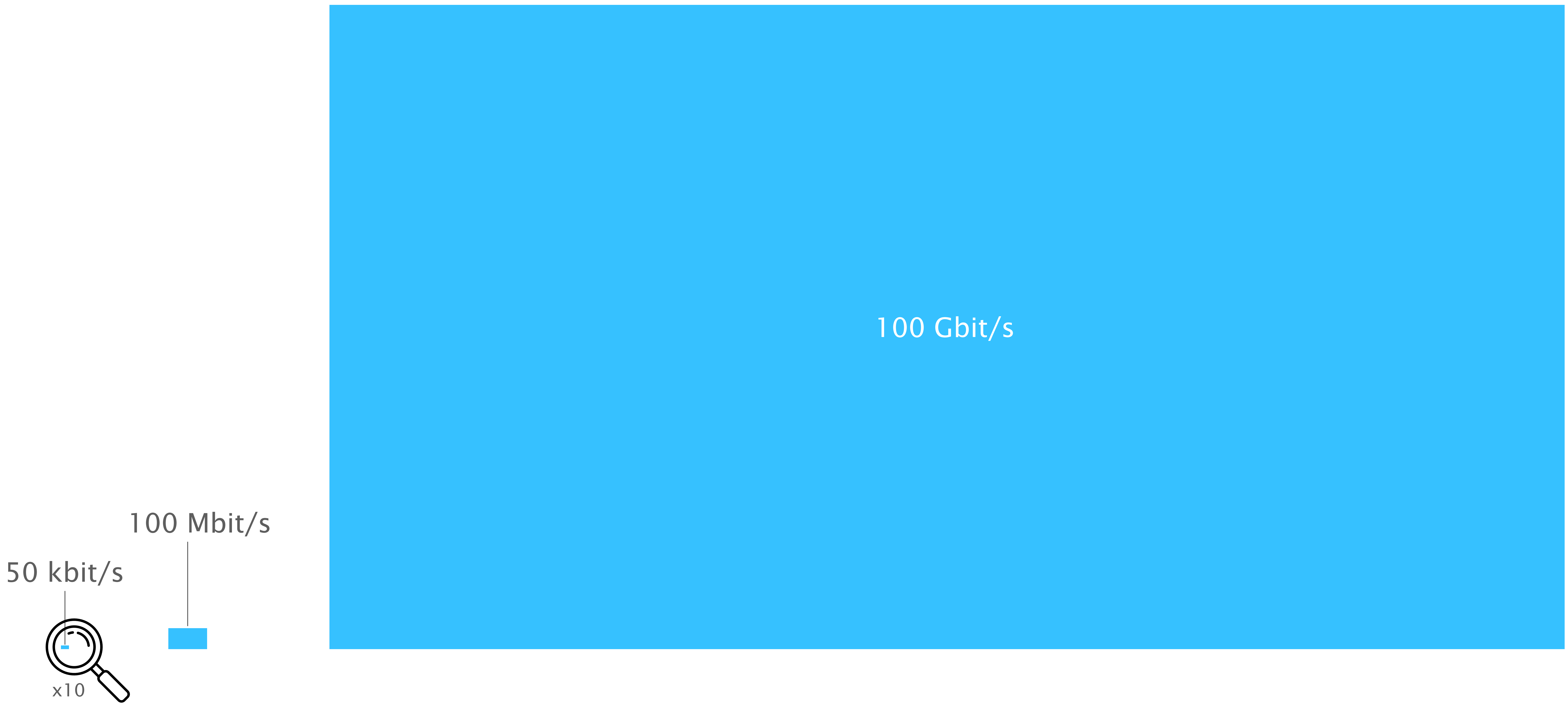
Not only the number of network users has changed, the traffic volume has too



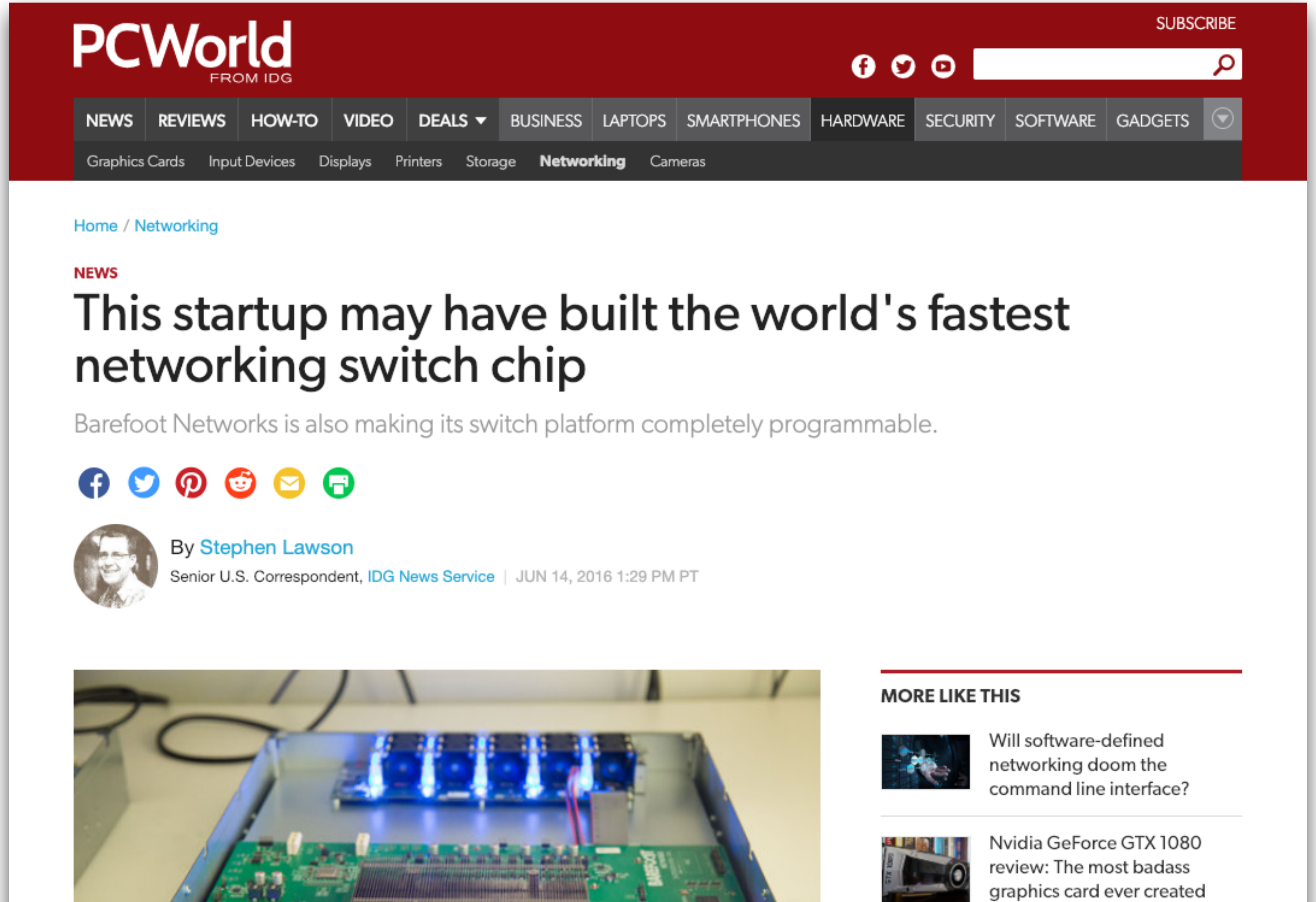
Not only the number of network users has changed, the traffic volume has too



Not only the number of network users has changed, the traffic volume has too



Our goal is to develop systems that work in these high-throughput networks using programmability


PCWorld FROM IDG

SUBSCRIBE

NEWS | REVIEWS | HOW-TO | VIDEO | DEALS | BUSINESS | LAPTOPS | SMARTPHONES | HARDWARE | SECURITY | SOFTWARE | GADGETS

Graphics Cards | Input Devices | Displays | Printers | Storage | **Networking** | Cameras

Home / Networking

NEWS

This startup may have built the world's fastest networking switch chip

Barefoot Networks is also making its switch platform completely programmable.

By Stephen Lawson

Senior U.S. Correspondent, IDG News Service | JUN 14, 2016 1:29 PM PT

MORE LIKE THIS

Will software-defined networking doom the command line interface?

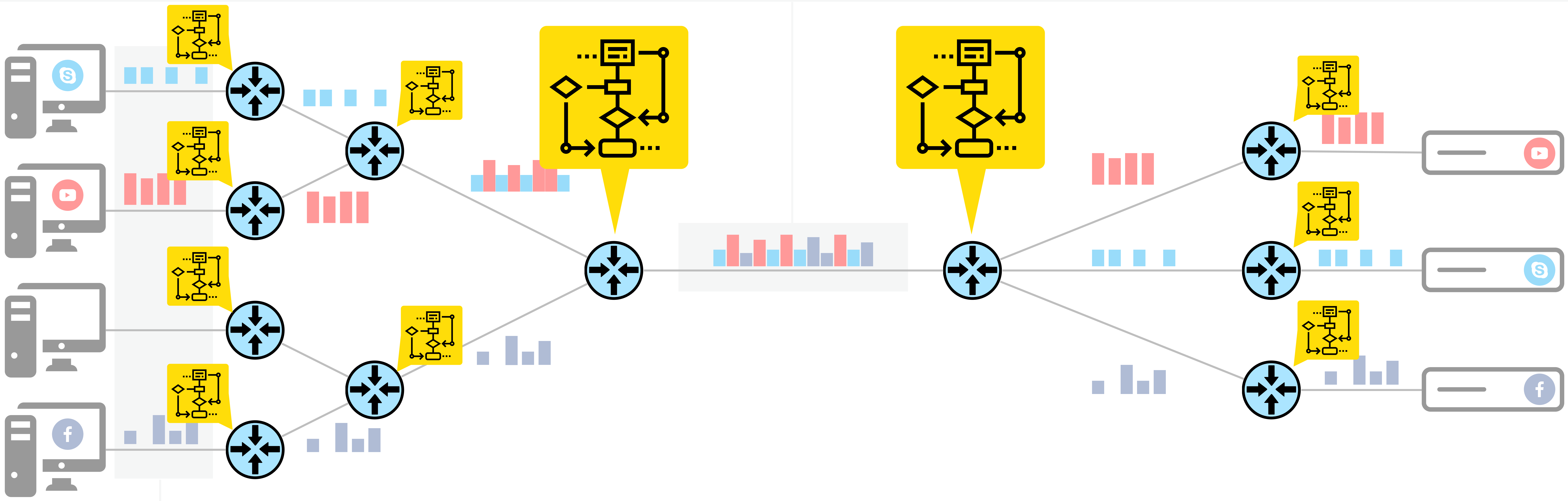
Nvidia GeForce GTX 1080 review: The most badass graphics card ever created

Problem #1

Traffic concentrates on one link

Vulnerable to denial-of-service attacks

NetHide prevents these attacks by obfuscating the topology



Encryption does not hide packet sizes and timings

Vulnerable to traffic-analysis attacks

ditto prevents these attacks by obfuscating the traffic

Problem #2

This thesis

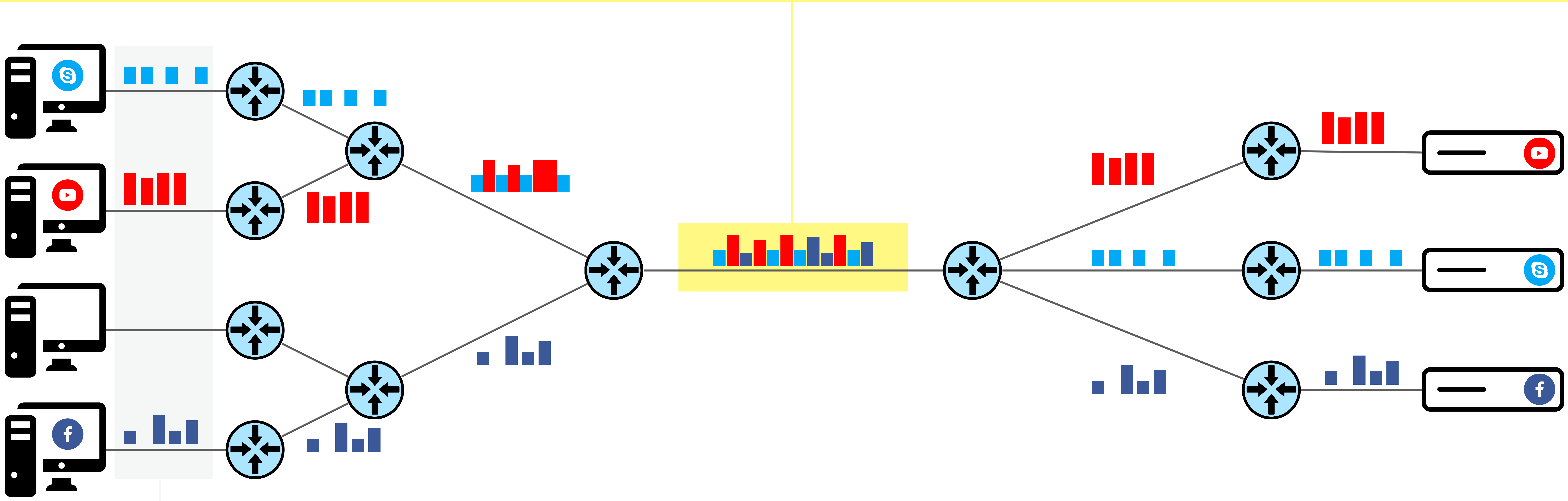
How can obfuscation and data-plane programmability increase the security of networks without degrading their performance?

Problem #1

Traffic concentrates on one link

Vulnerable to denial-of-service attacks

NetHide prevents these attacks by obfuscating the topology

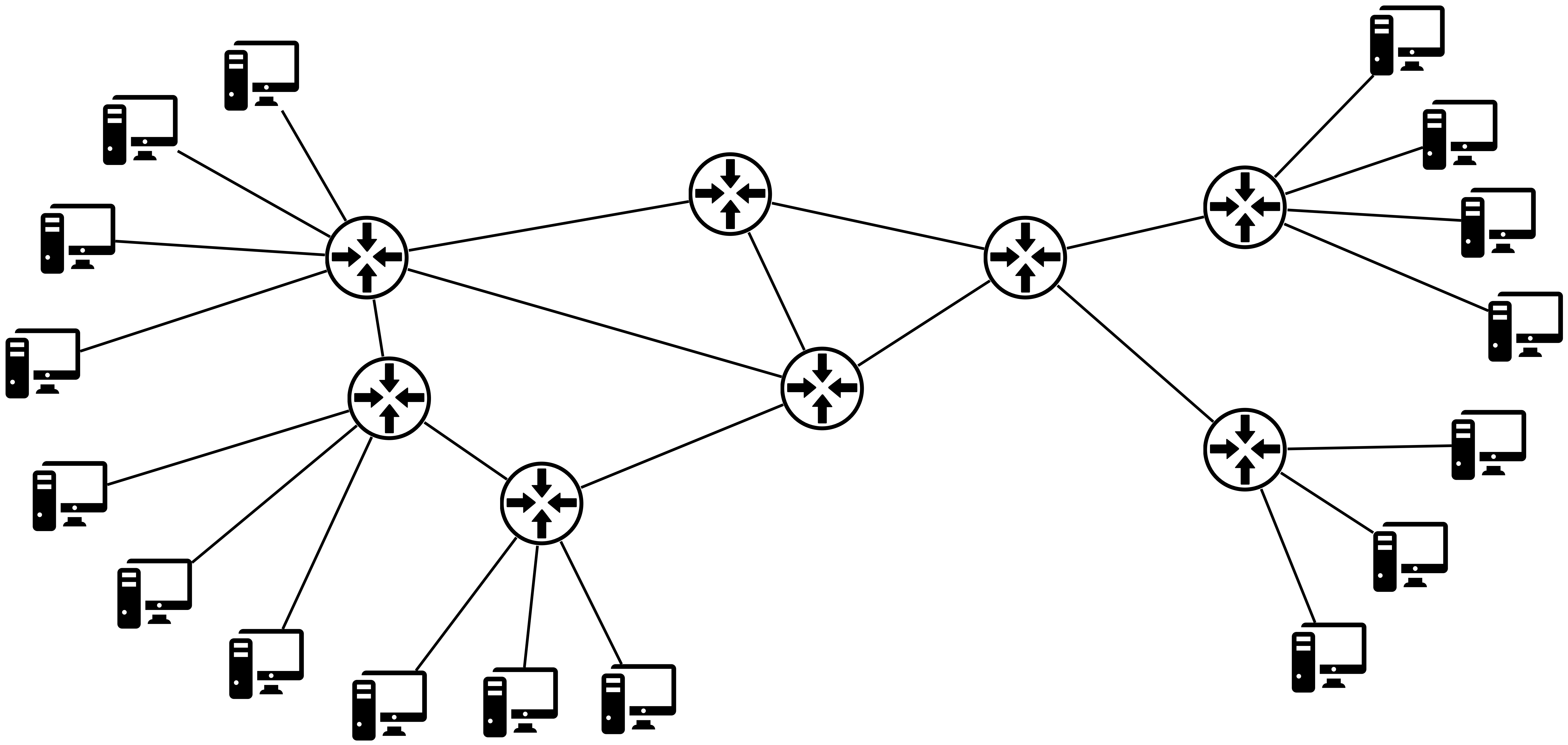


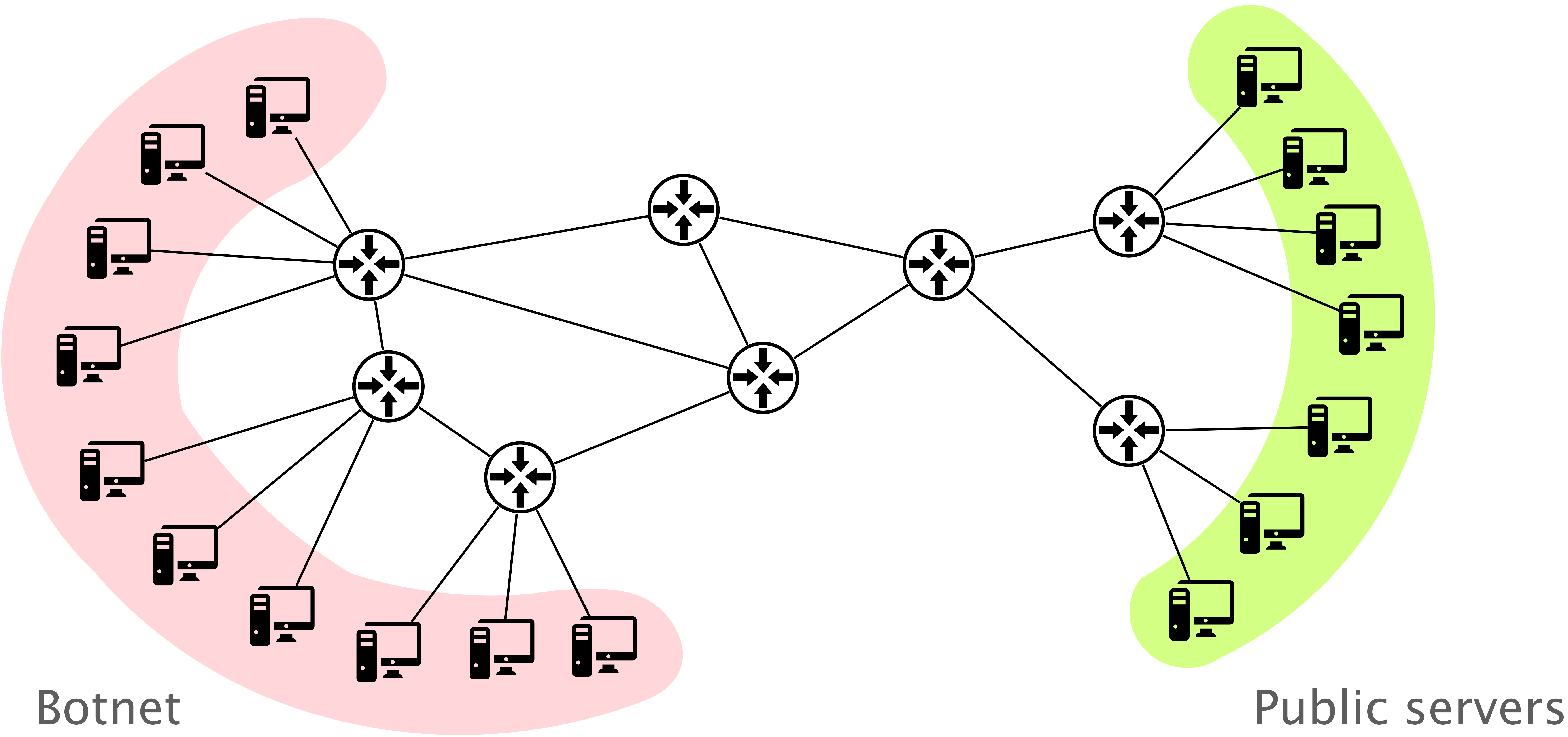
Encryption does not hide packet sizes and timings

Vulnerable to traffic-analysis attacks

ditto prevents these attacks by obfuscating the traffic

Problem #2

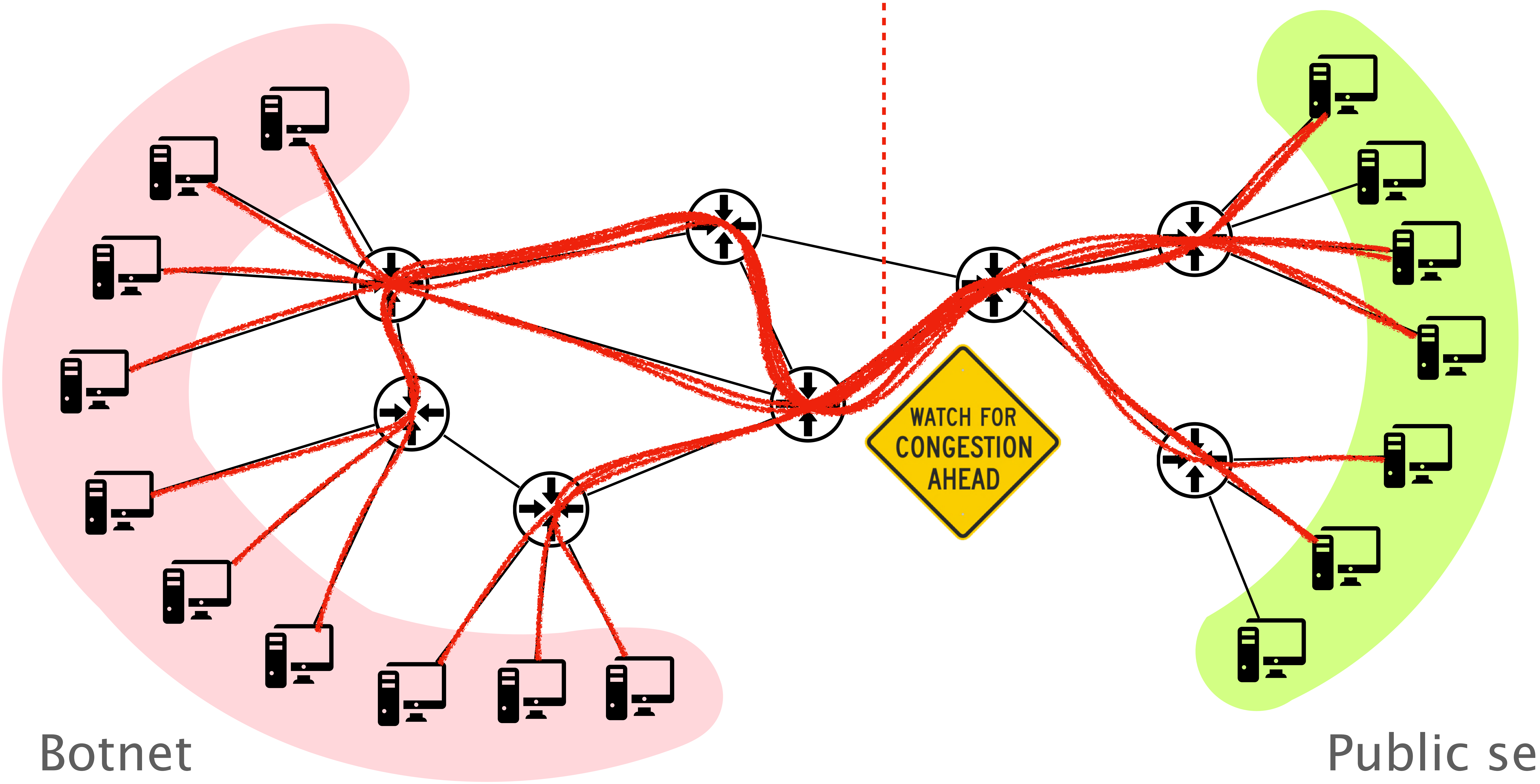



Botnet
Public servers

Link-flooding attacks (LFAs)

target the infrastructure

Low-rate, legitimate flows
spread over many endpoints

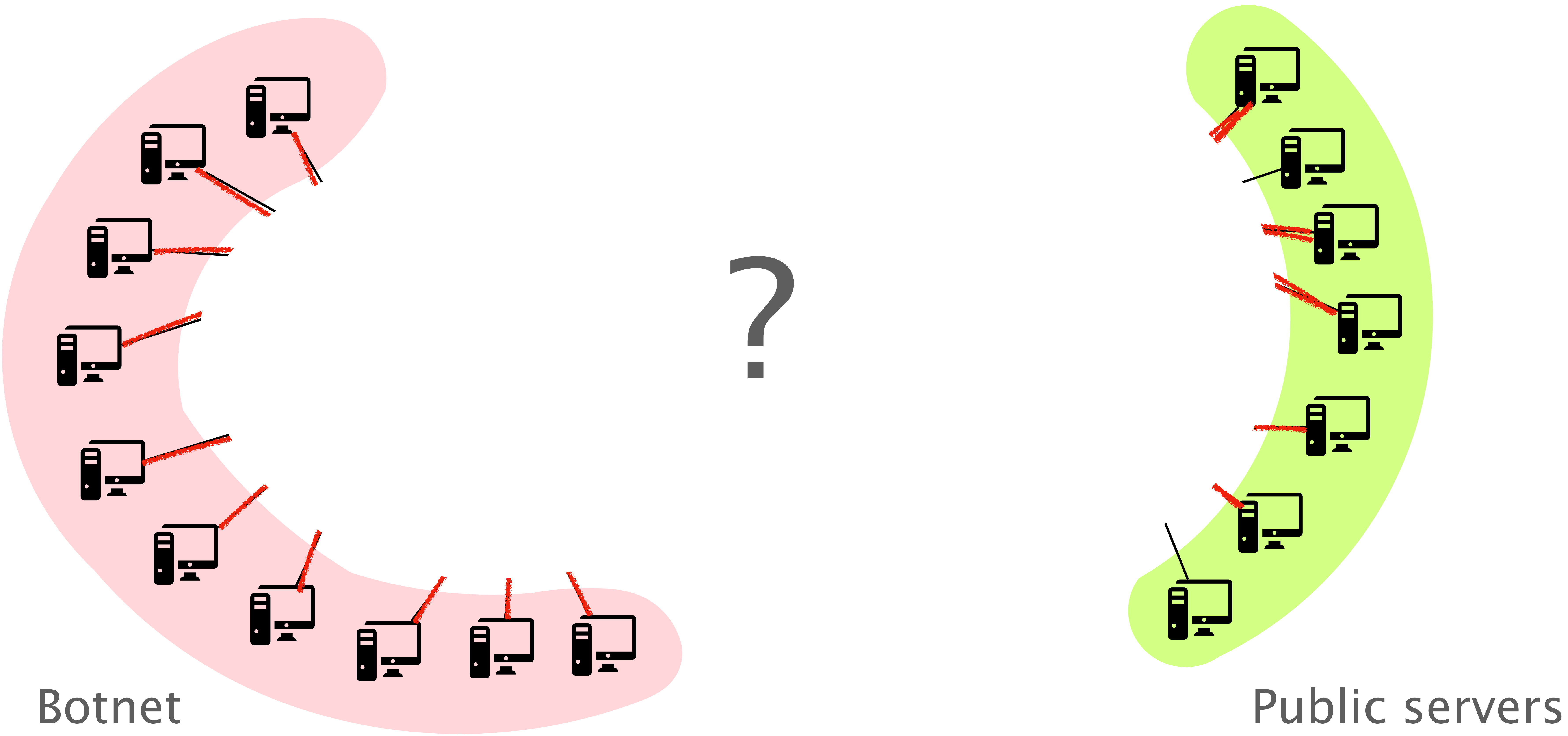


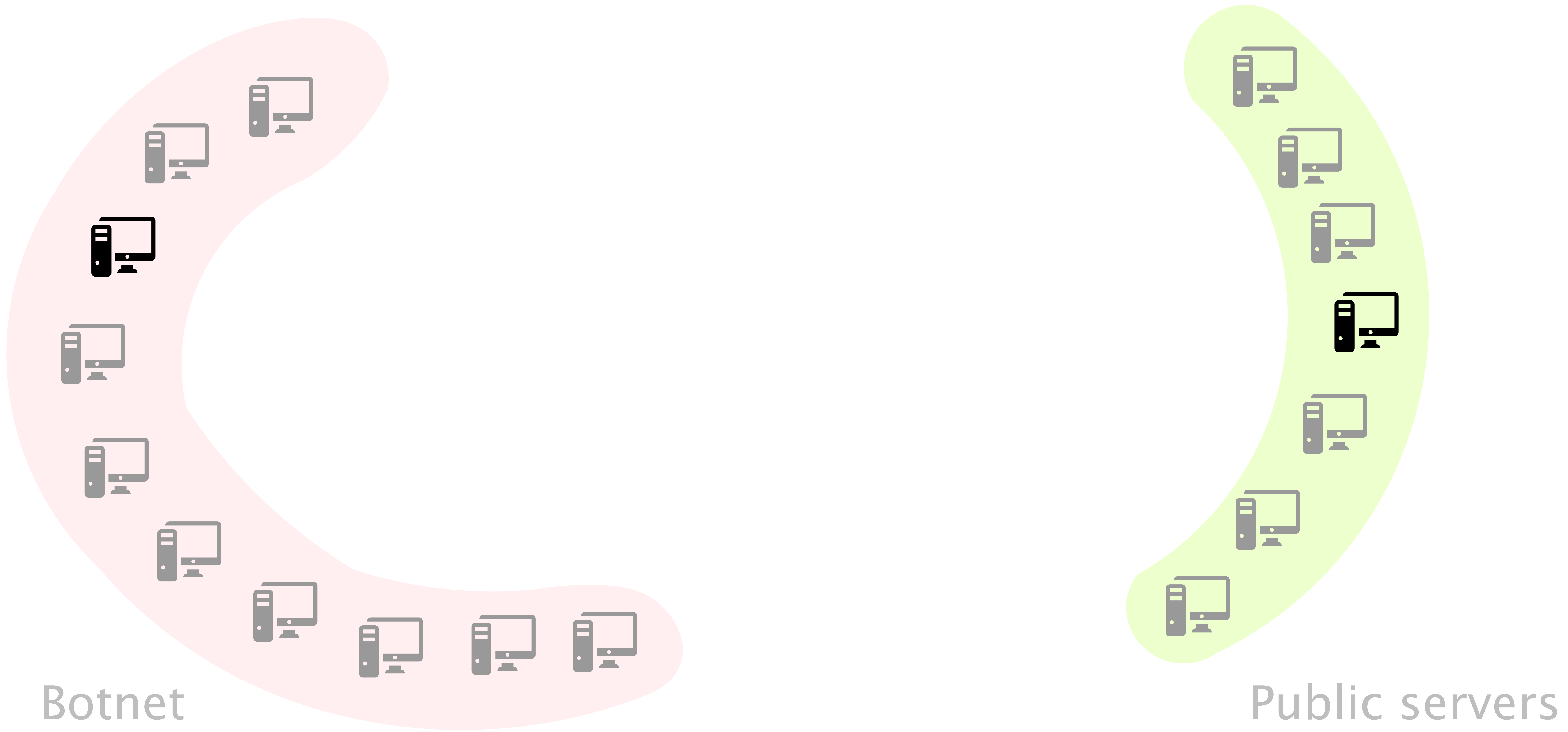
Botnet

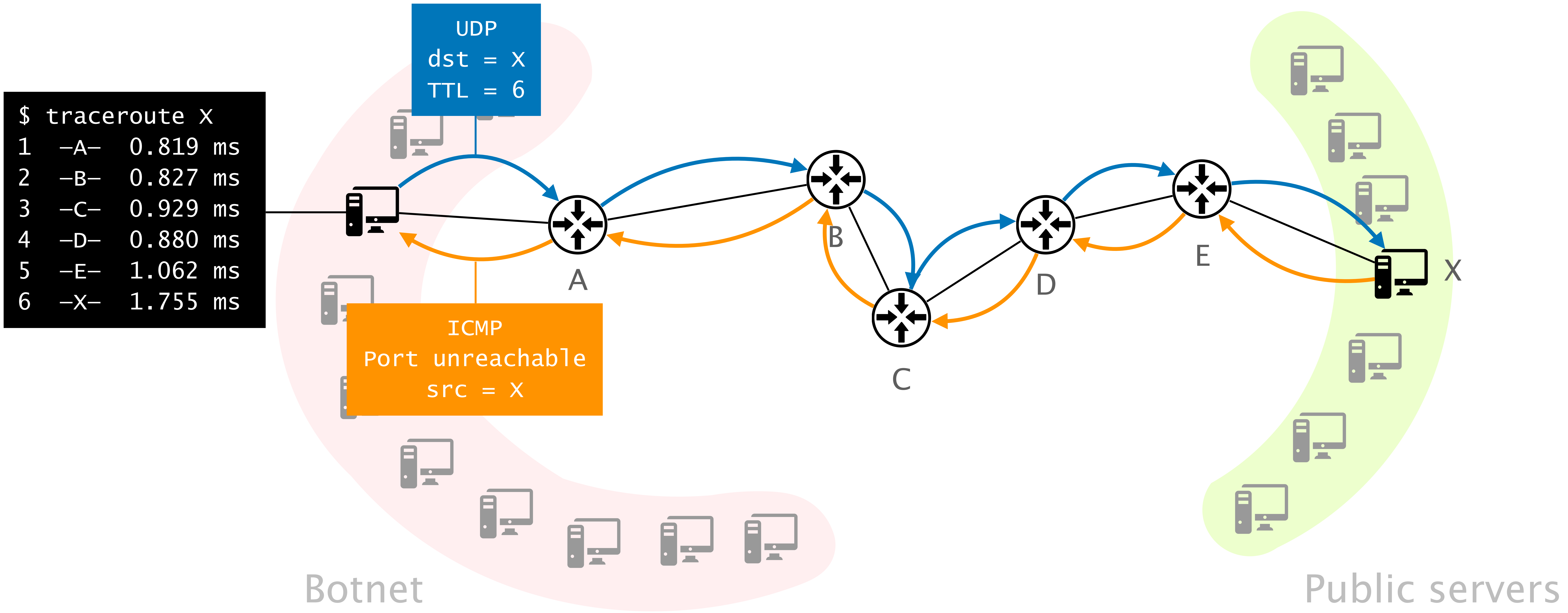
Public servers

Link-flooding attacks (LFAs)

require knowing the topology




Botnet
Public servers



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

So the solution is to hide the topology?

So the solution is to hide the topology?

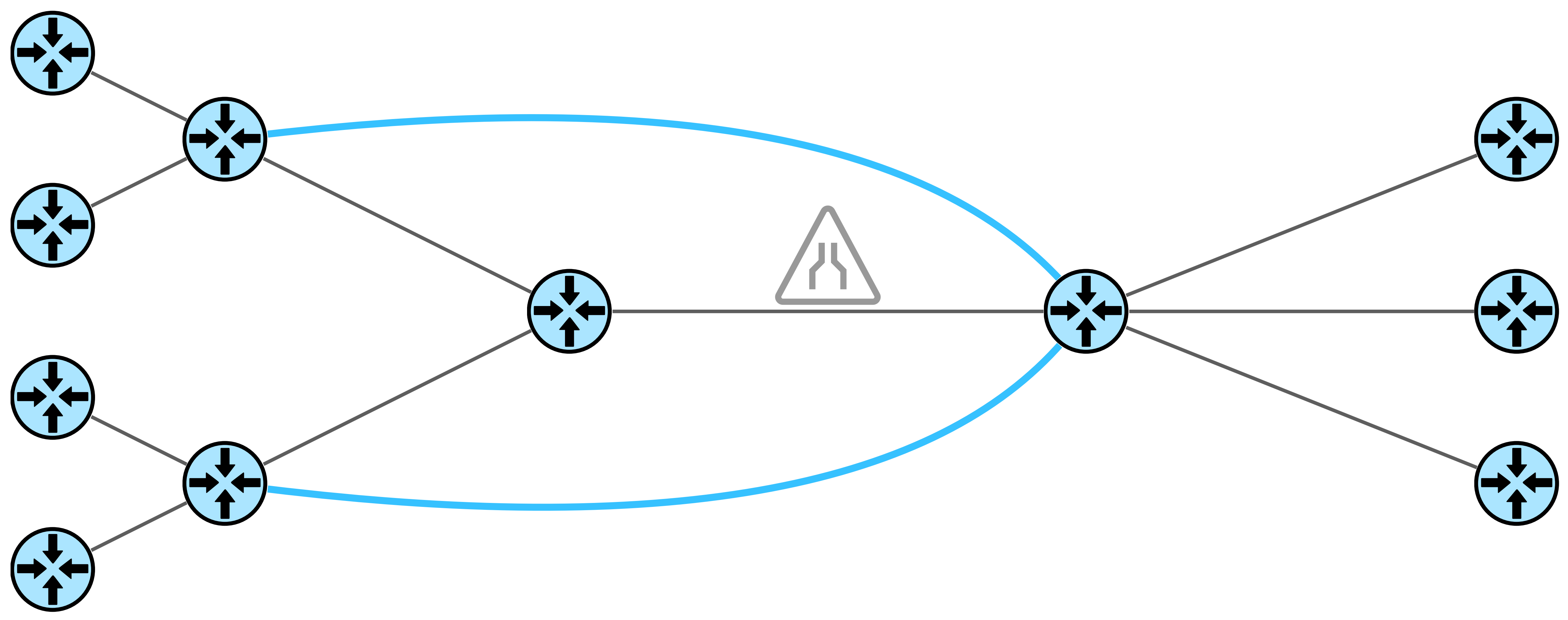
yes, but…

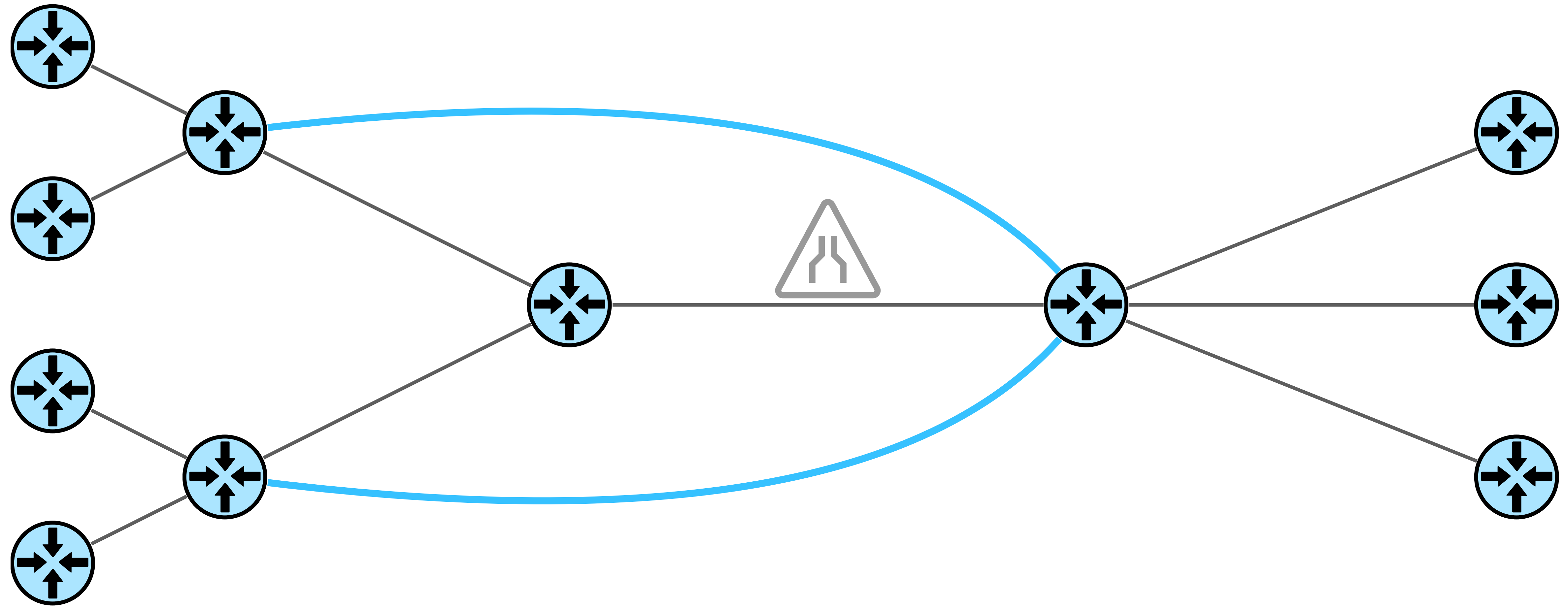
parts of

So the solution is to hide the topology?

parts of

So the solution is to hide the topology?

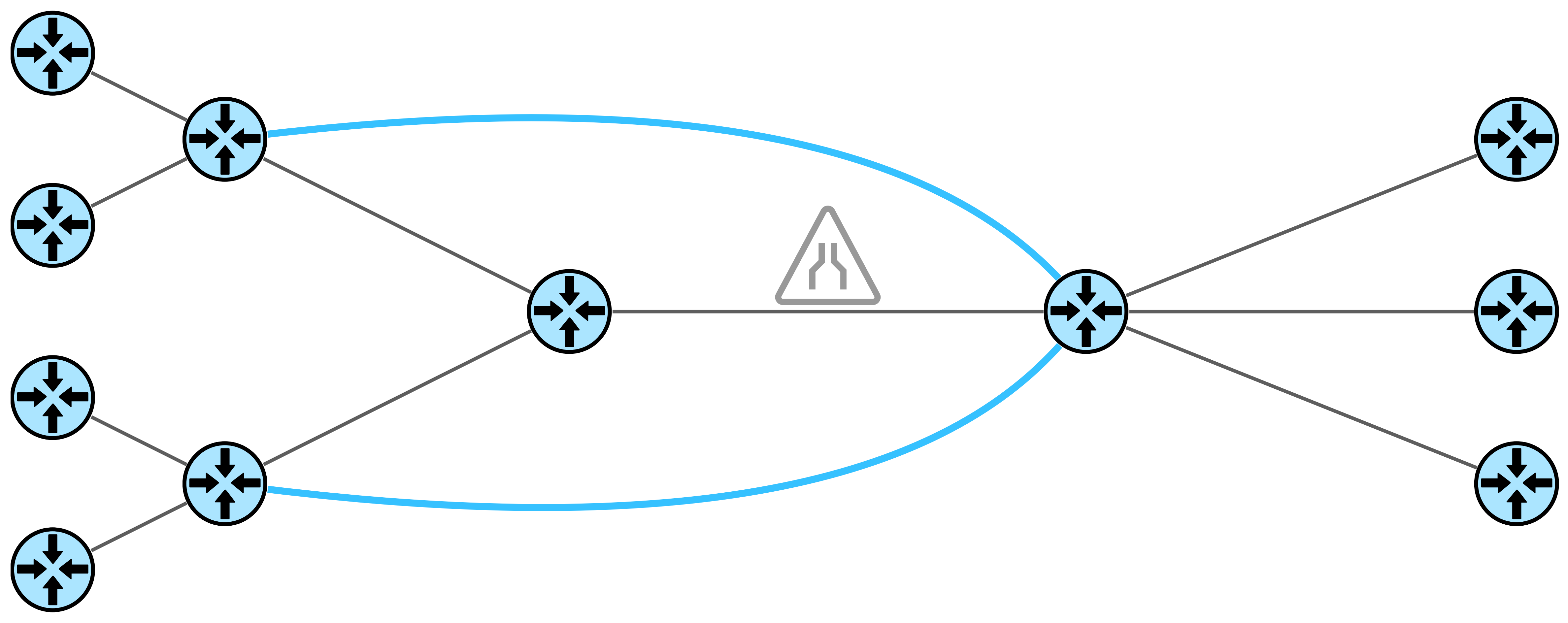




Computing the virtual topology

Deploying the virtual topology

Experimental results



Computing the virtual topology

Deploying the virtual topology

Experimental results

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

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

Attacker can run flows between pairs of routers

Attacker can run flows between pairs of routers

Attacker

- controls a set of hosts
  i.e., a botnet
- has a budget of flows to run
  flows between nodes (routers)
- has no prior knowledge about topology
  learns topology e.g., through traceroute

A topology is robust against LFAs,
if the flow density of each link does not exceed its capacity

$$\forall l \in L' :$$

Links in **V**

A topology is robust against LFAs,
if the flow density of each link does not exceed its capacity

Flow density of the link
(# of flows using it)

$$\forall l \in L' : \mathrm{fd}(l)$$

Links in **V**

A topology is robust against LFAs, if the flow density of each link does not exceed its capacity

Flow density of the link
(# of flows using it)

$$\forall l \in L' : \mathrm{fd}(l) \leq \mathrm{c}(l)$$

Links in **V**

Capacity of the link
(max # of flows)

Topology obfuscation as an optimization problem

Given the **physical topology P**,

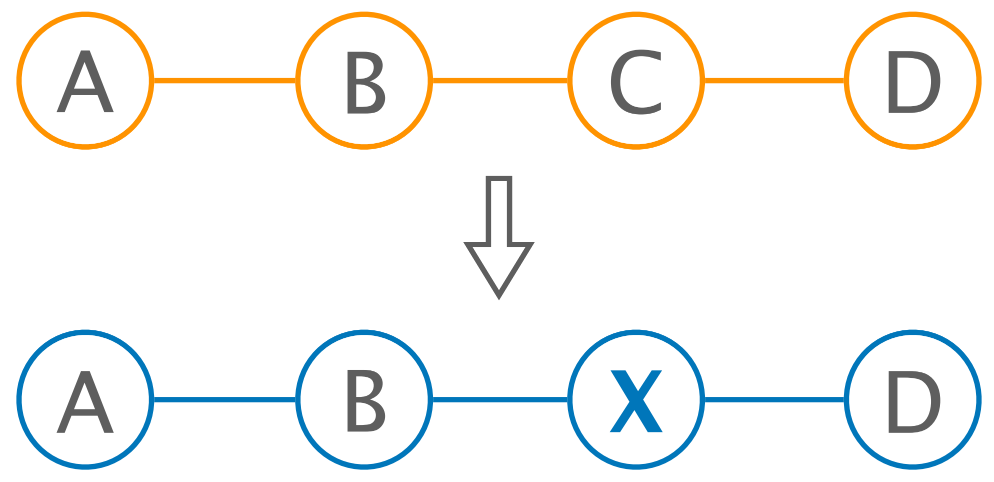
compute a **virtual topology V**, such that

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

Accuracy and utility measure the closeness of P and V

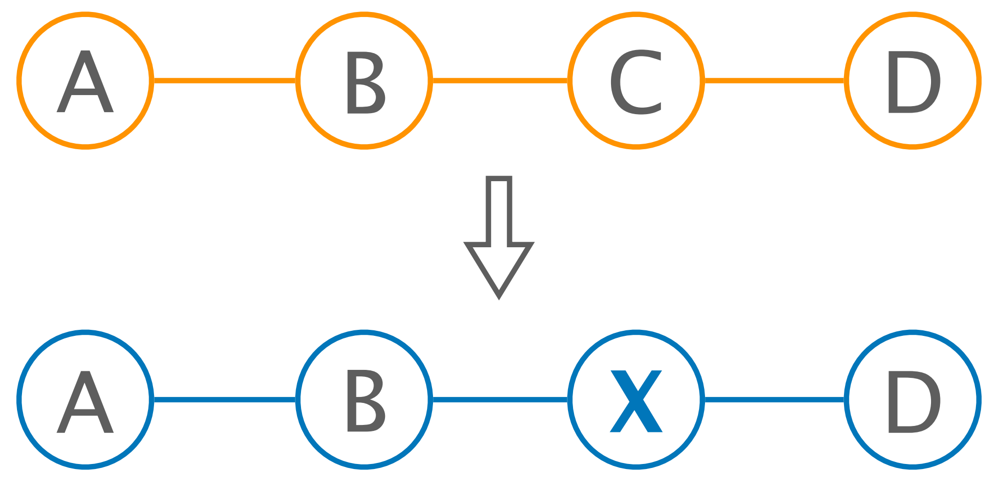
Accuracy and utility measure the closeness of P and V

- *Accuracy:* Virtual paths are similar to physical paths

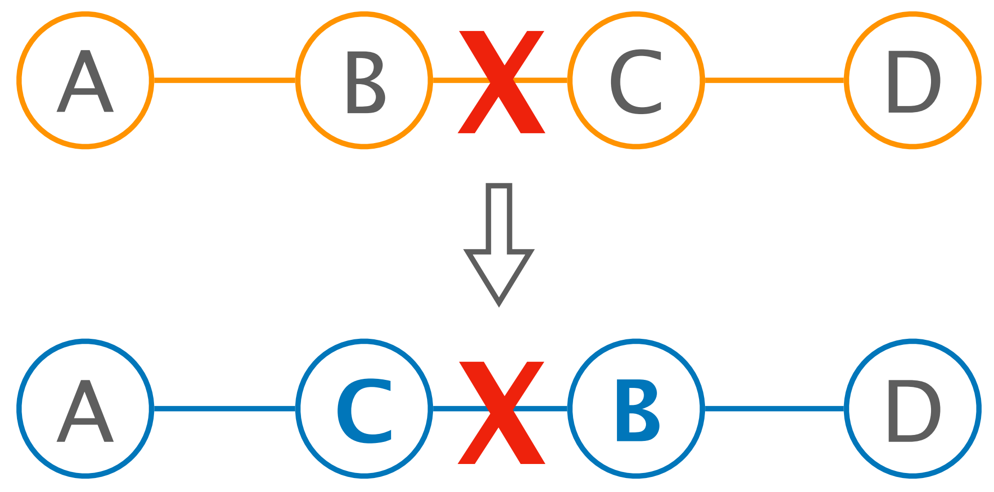


Accuracy and utility measure the closeness of **P** and **V**

- *Accuracy:* Virtual paths are similar to physical paths



- *Utility:* Failures in **P** are reflected in **V**



Topology obfuscation as an optimization problem

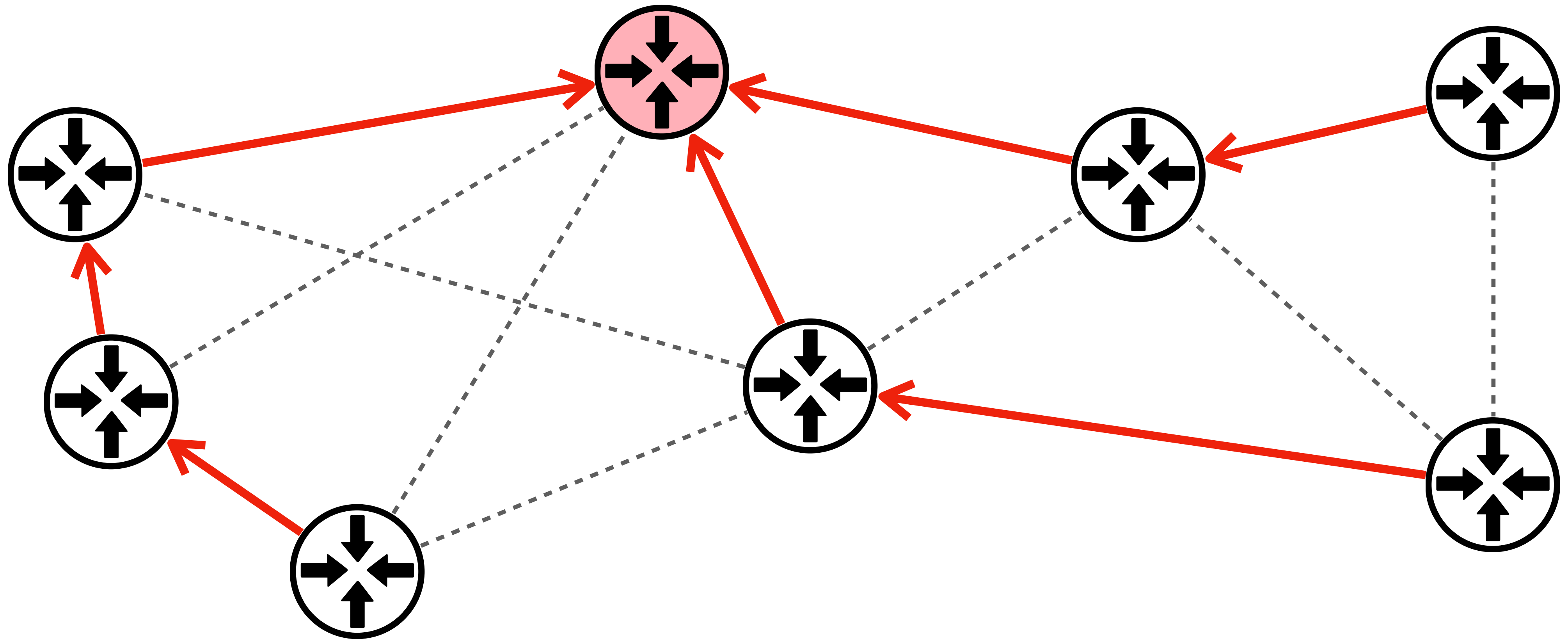
Given the **physical topology P**,

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

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

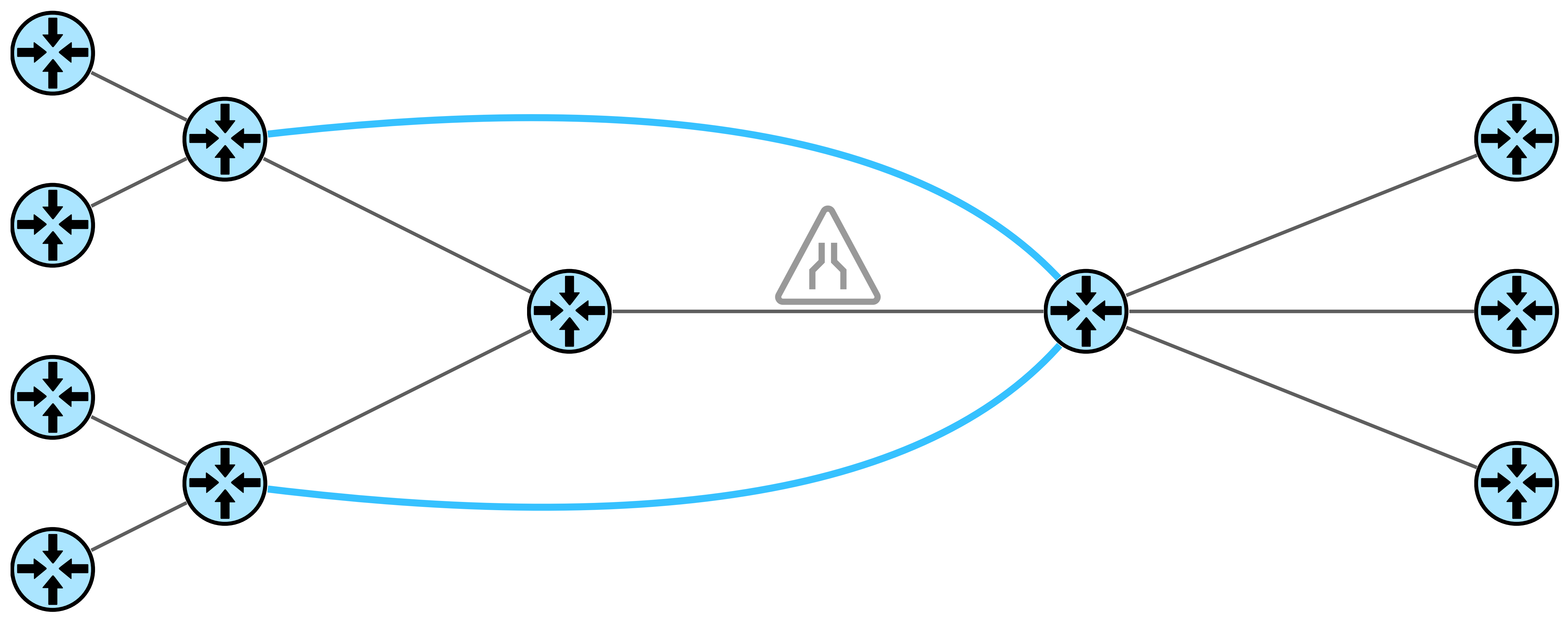
NetHide finds the virtual topology as the best combination of forwarding trees

NetHide finds the virtual topology as the best combination of forwarding trees



For each node *n*:

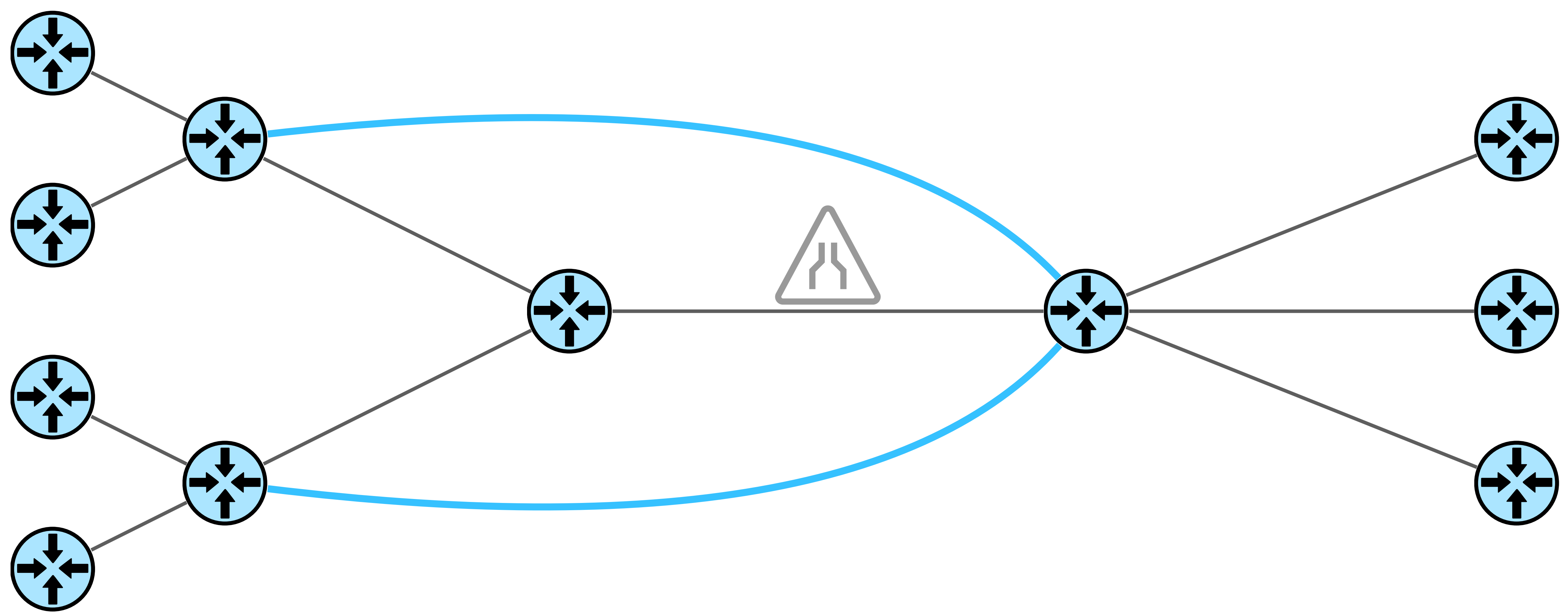
tree rooted at *n*
that specifies forwarding paths
from each other node to *n*



Computing the virtual topology

Deploying the virtual topology

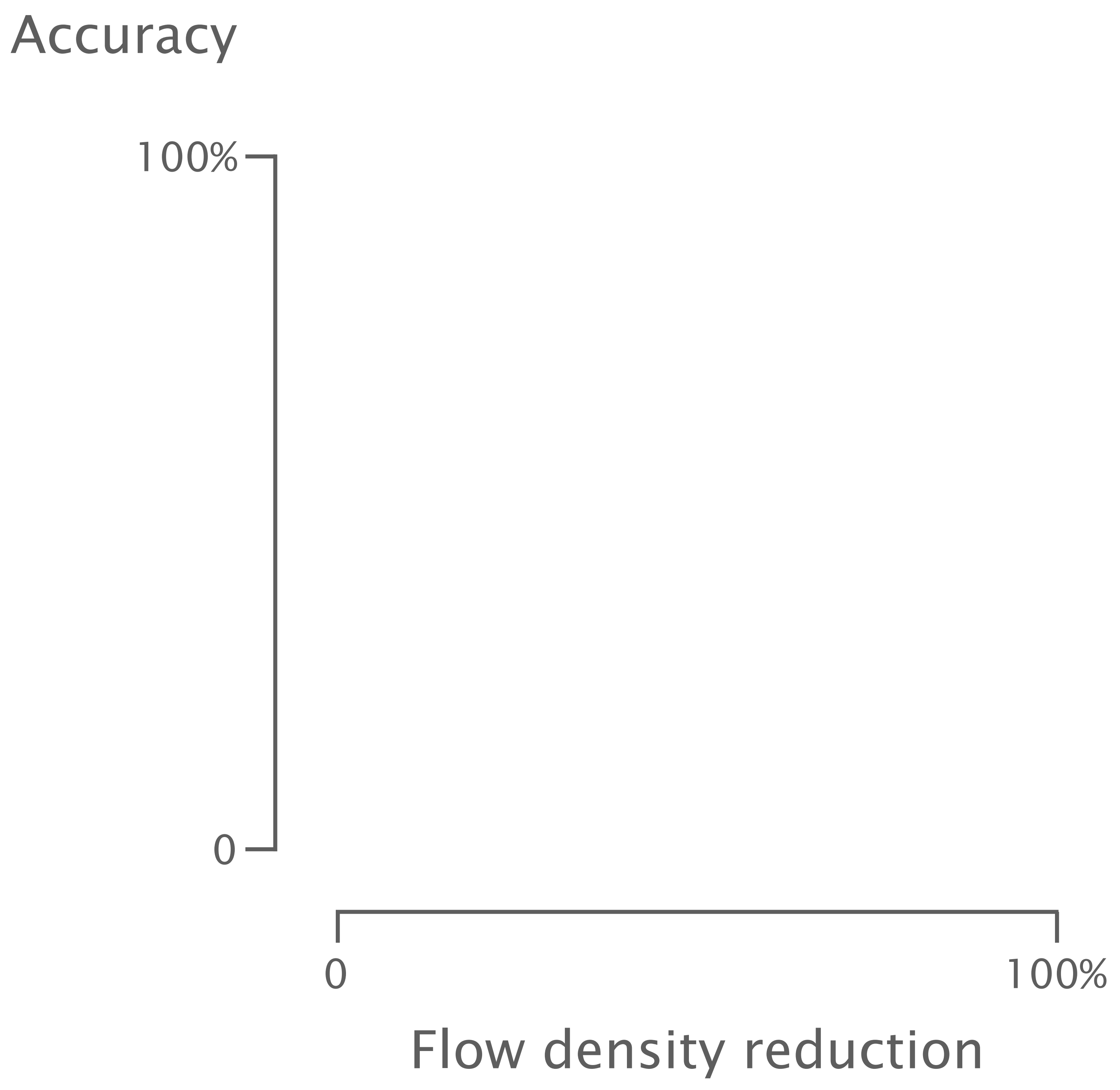
Experimental results

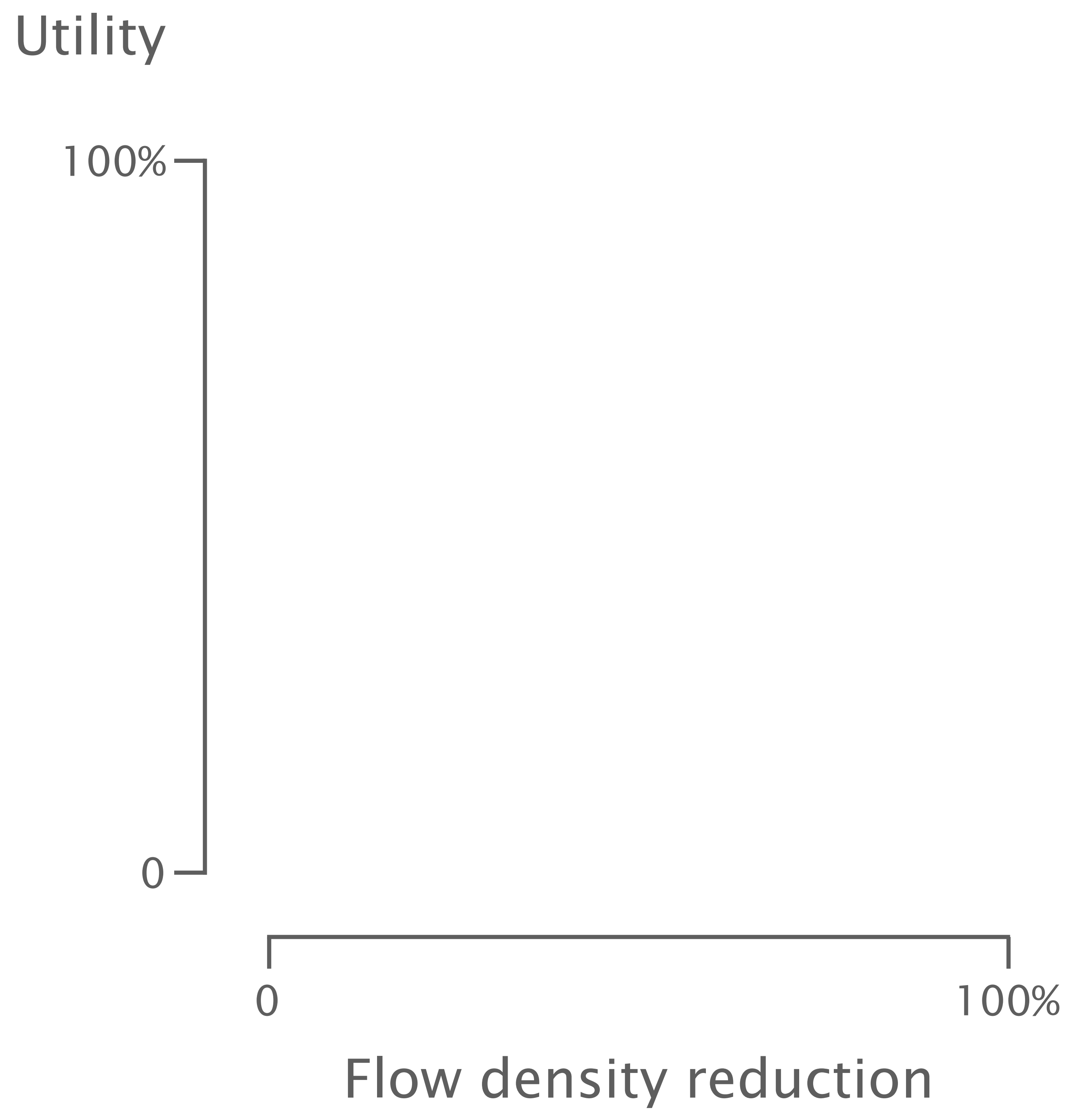


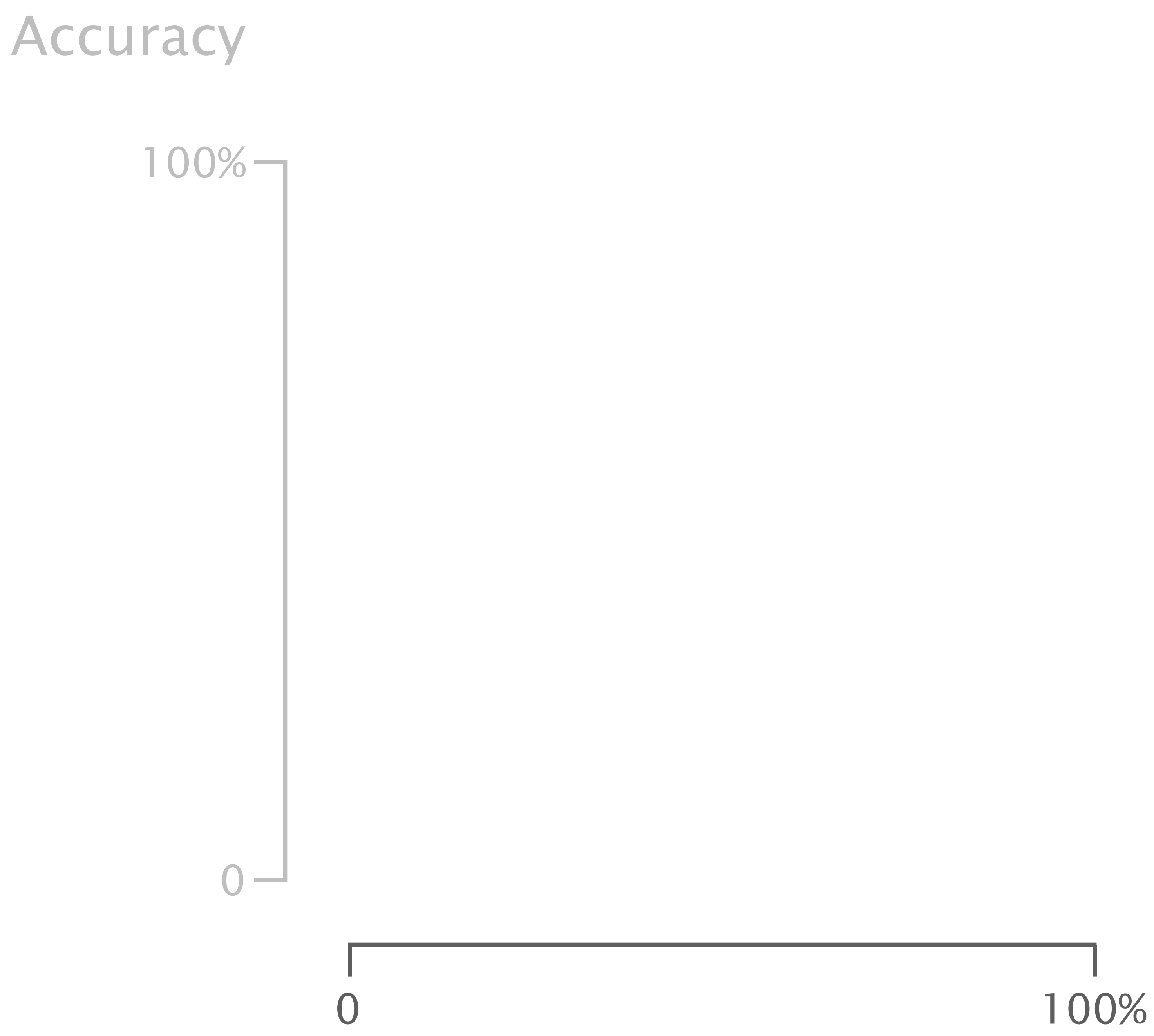
Computing the virtual topology

Deploying the virtual topology

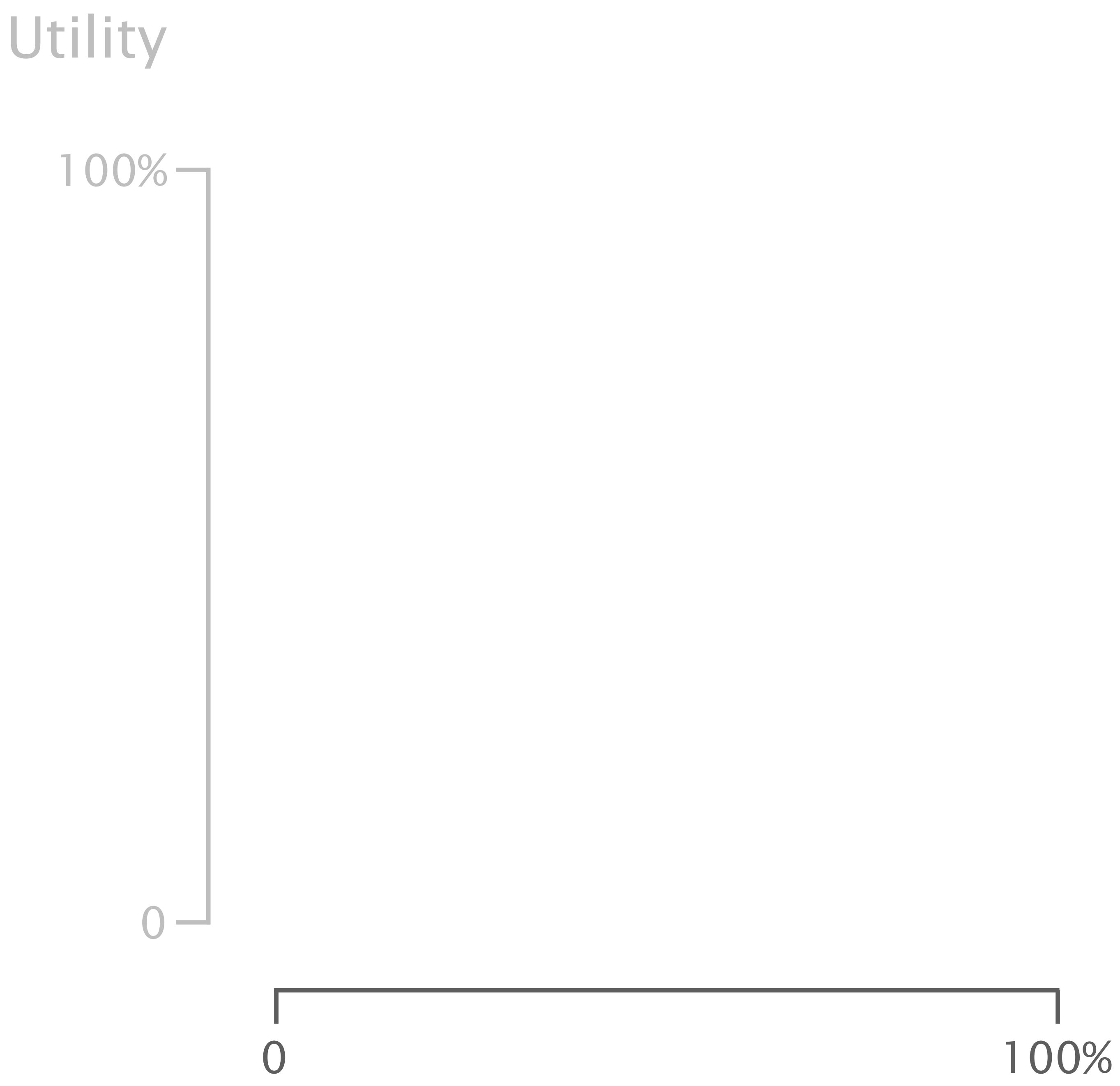
Experimental results


Accuracy
100%
0
0
100%
Flow density reduction


Utility
100%
0
0
100%
Flow density reduction



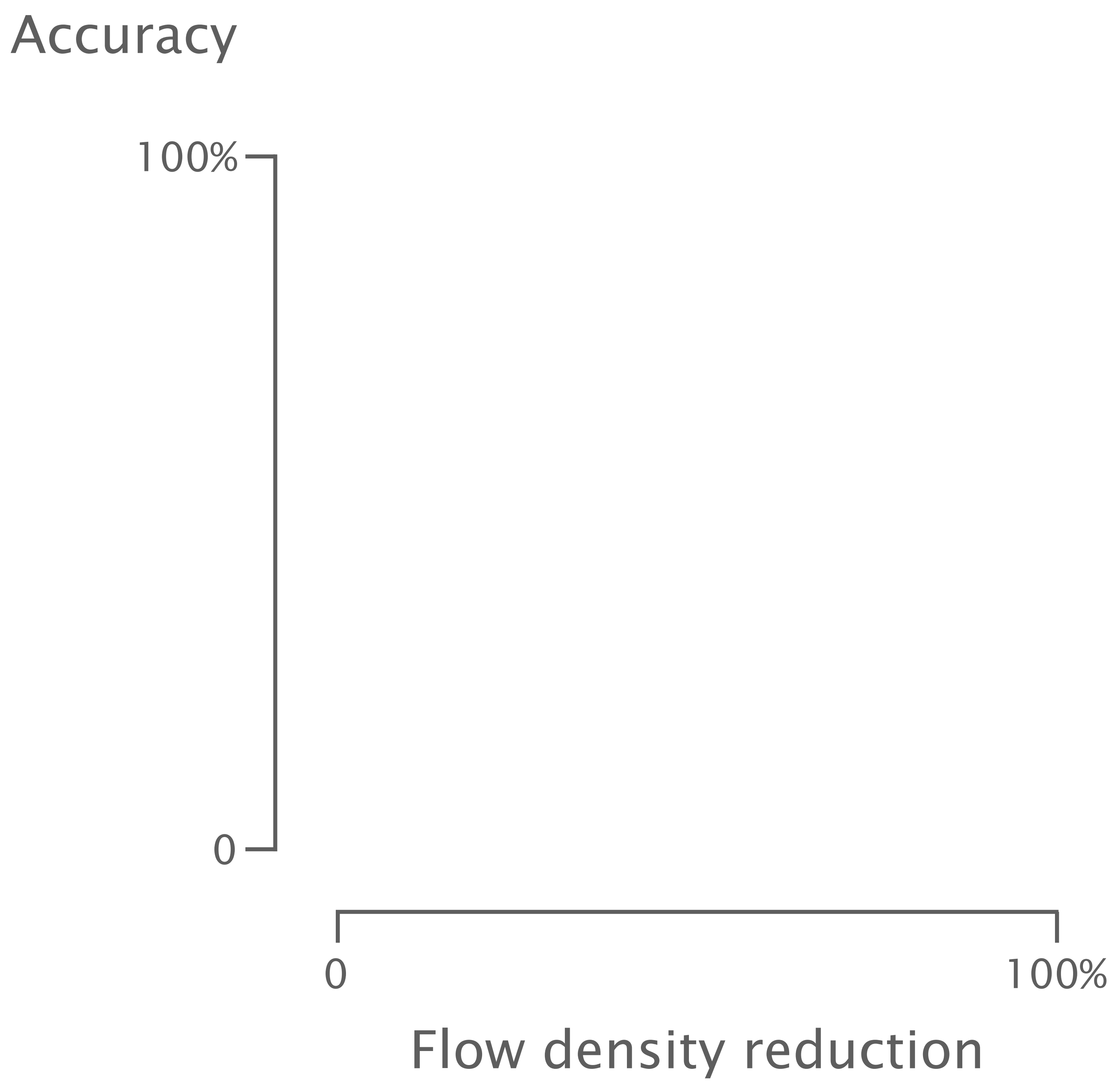
Flow density reduction

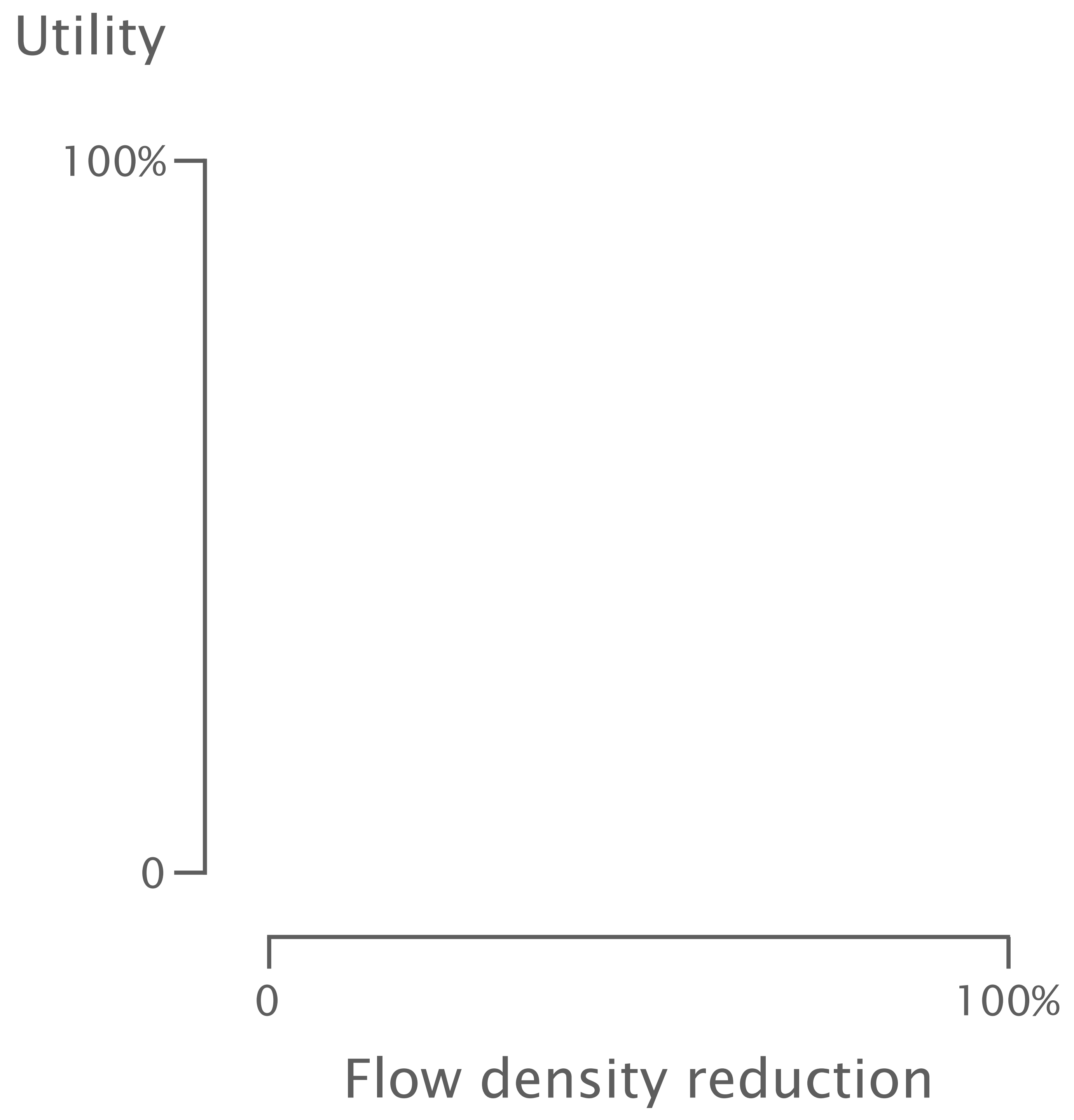


Flow density reduction

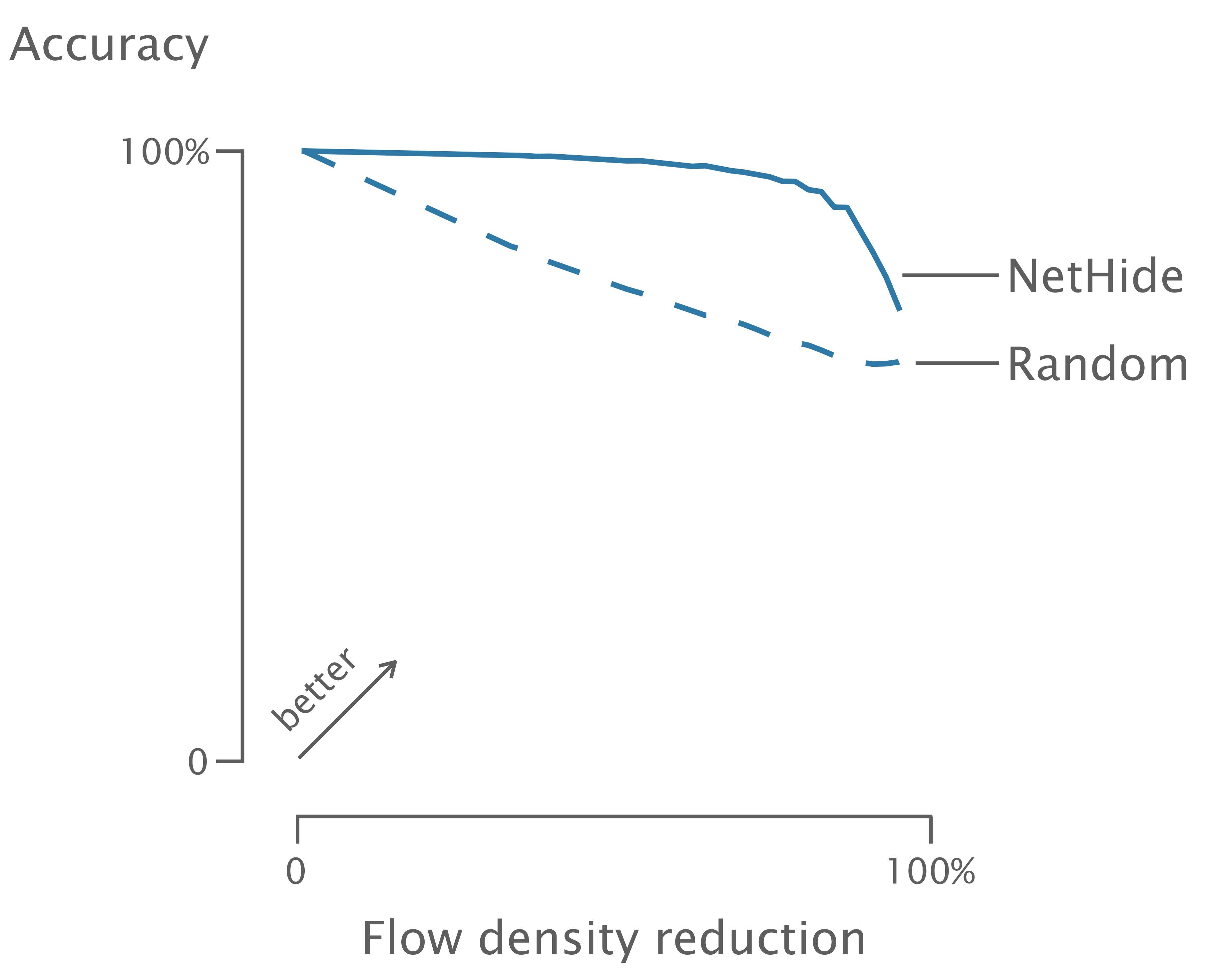
Ratio between the flow density
in the physical and the virtual topology

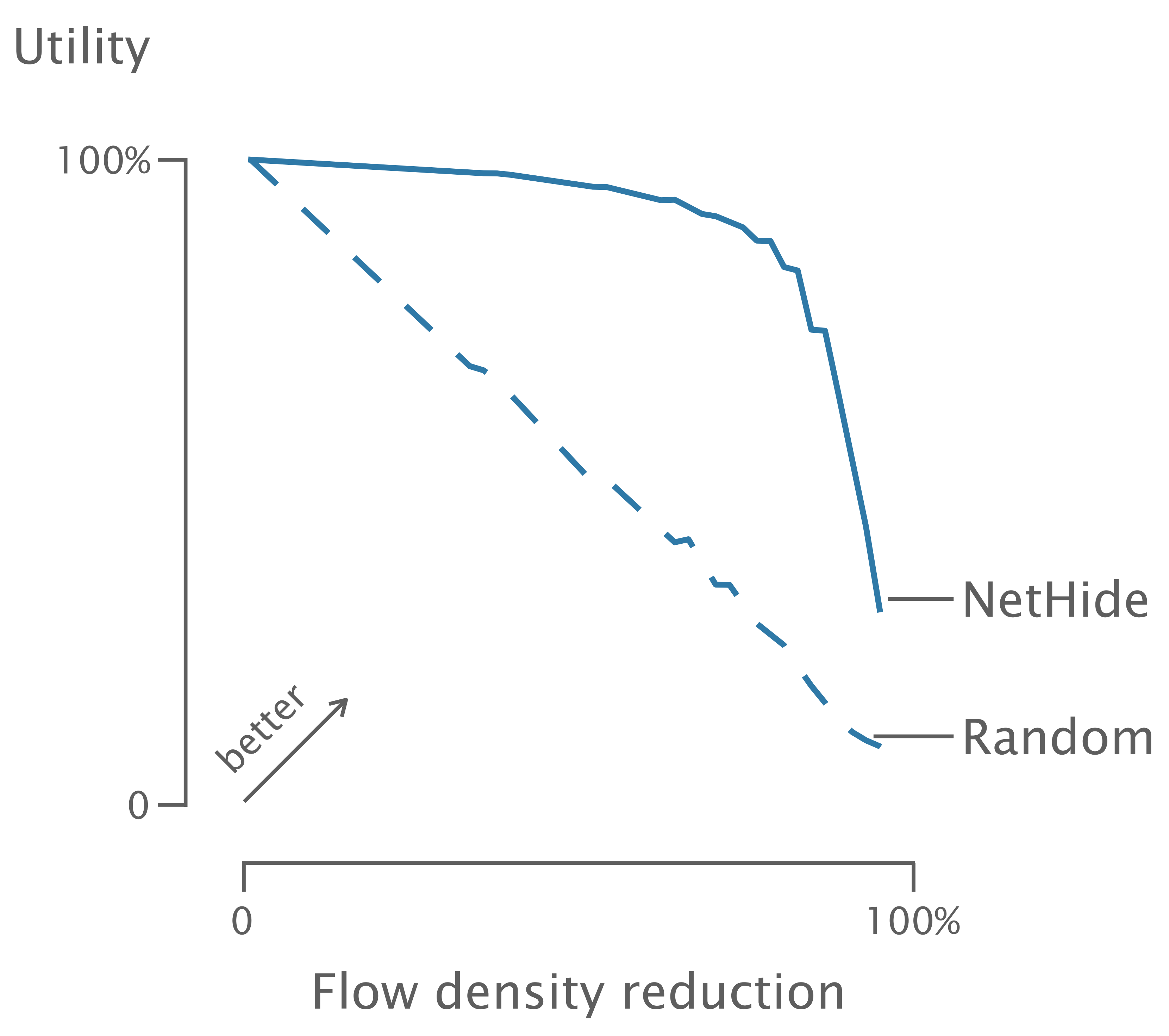
High protection
with small impact on accuracy and utility



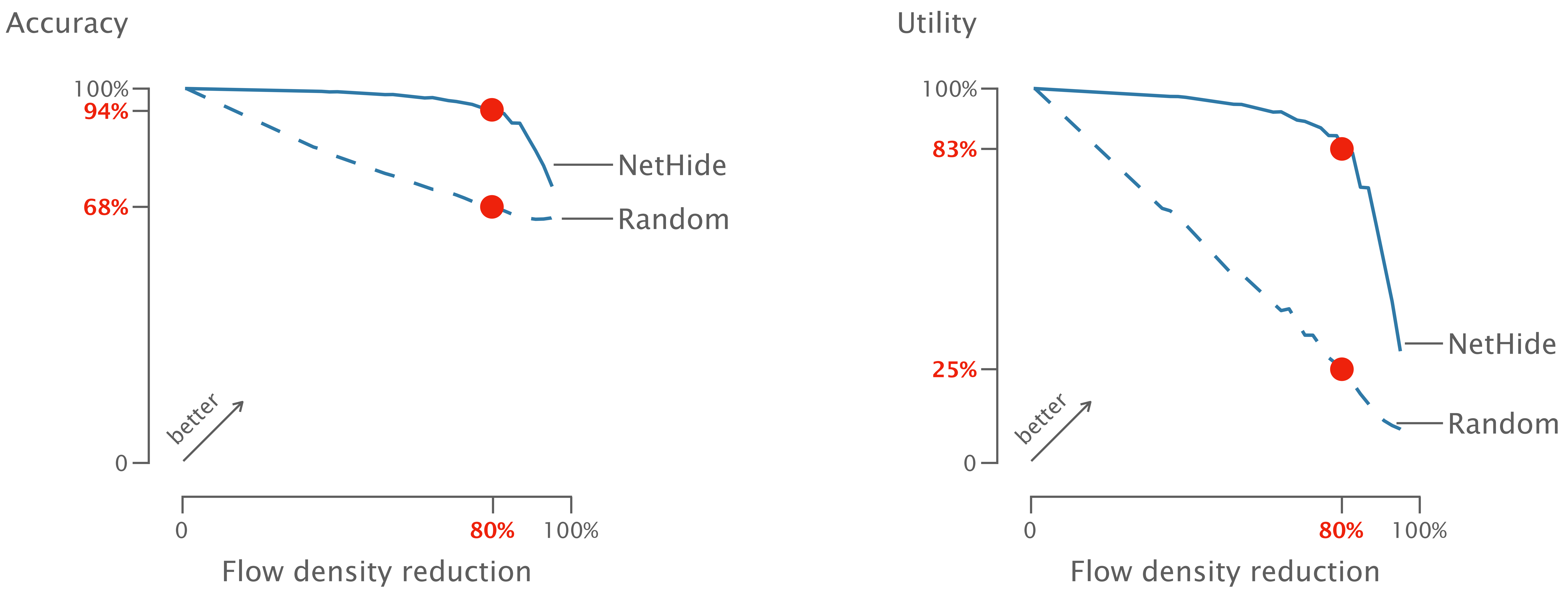


High protection with small impact on accuracy and utility

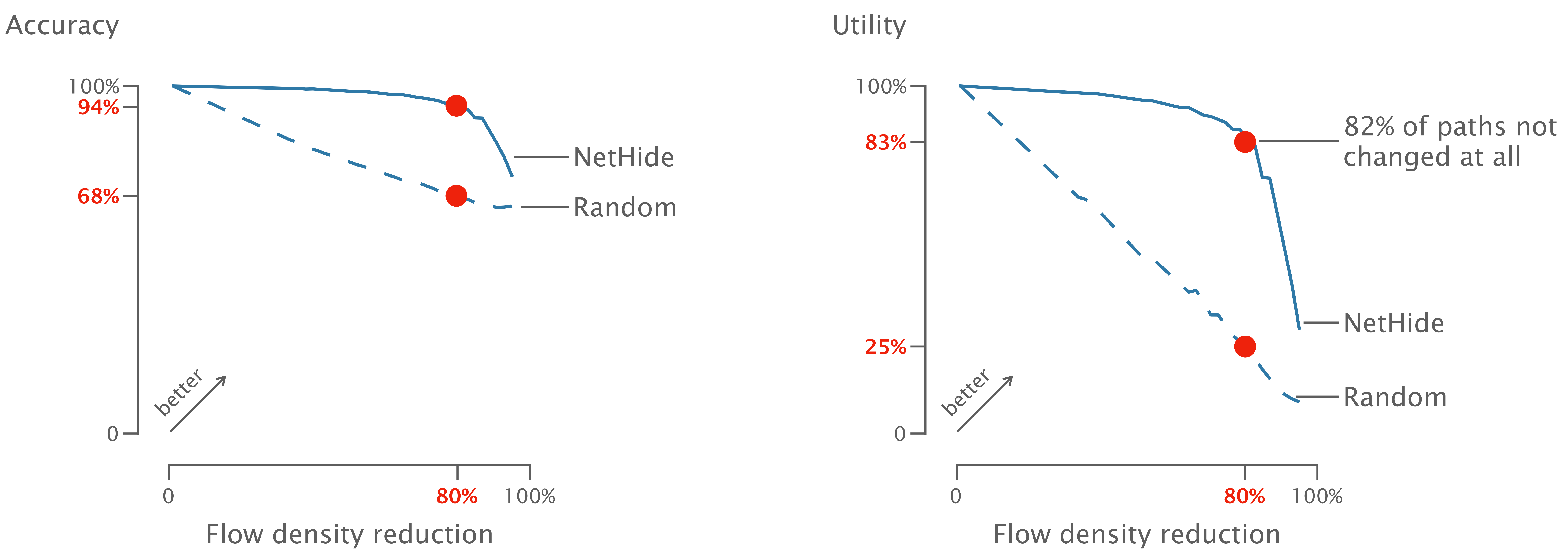




High protection with small impact on accuracy and utility



High protection
with small impact on accuracy and utility

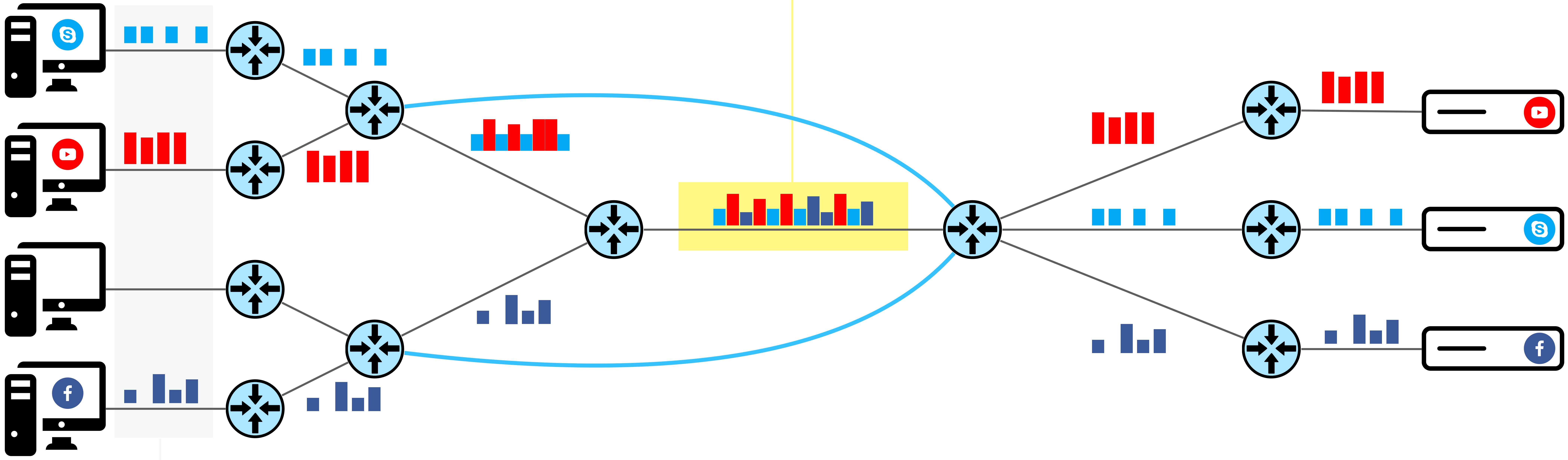


Problem #1

Traffic concentrates on one link

Vulnerable to denial-of-service attacks

NetHide prevents these attacks by obfuscating the topology



Encryption does not hide packet sizes and timings

Vulnerable to traffic-analysis attacks

ditto prevents these attacks by obfuscating the traffic

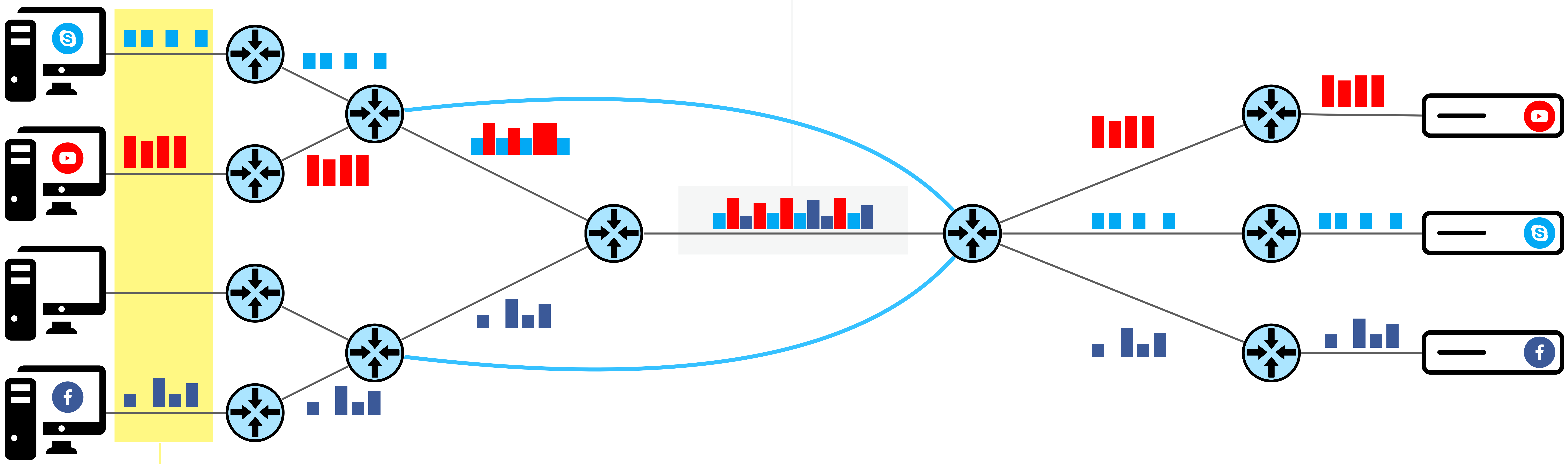
Problem #2

Problem #1

Traffic concentrates on one link

Vulnerable to denial-of-service attacks

NetHide prevents these attacks by obfuscating the topology

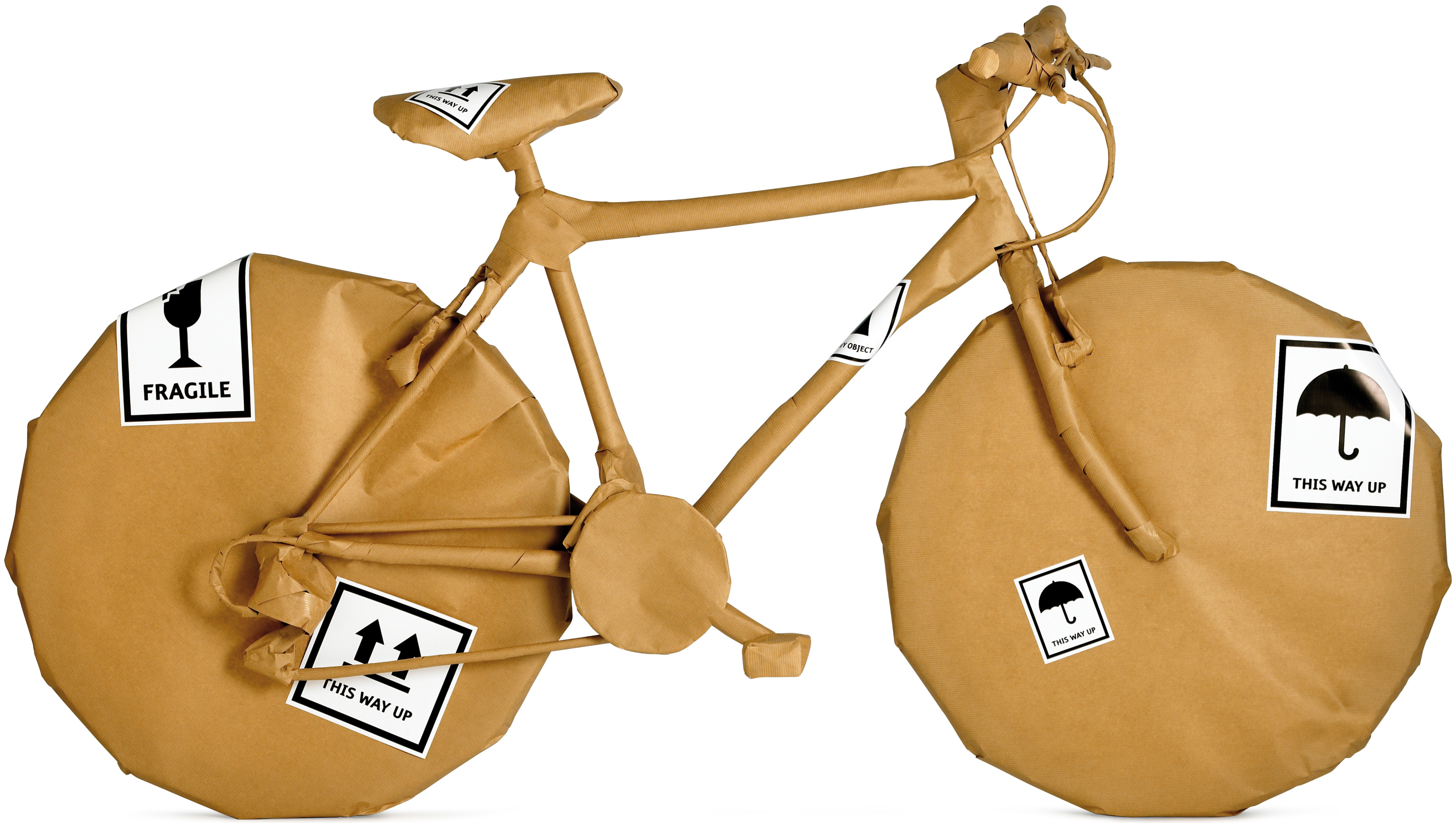


Encryption does not hide packet sizes and timings

Vulnerable to traffic-analysis attacks

ditto prevents these attacks by obfuscating the traffic

Problem #2


FRAGILE
THIS WAY UP
THIS WAY UP
THIS WAY UP

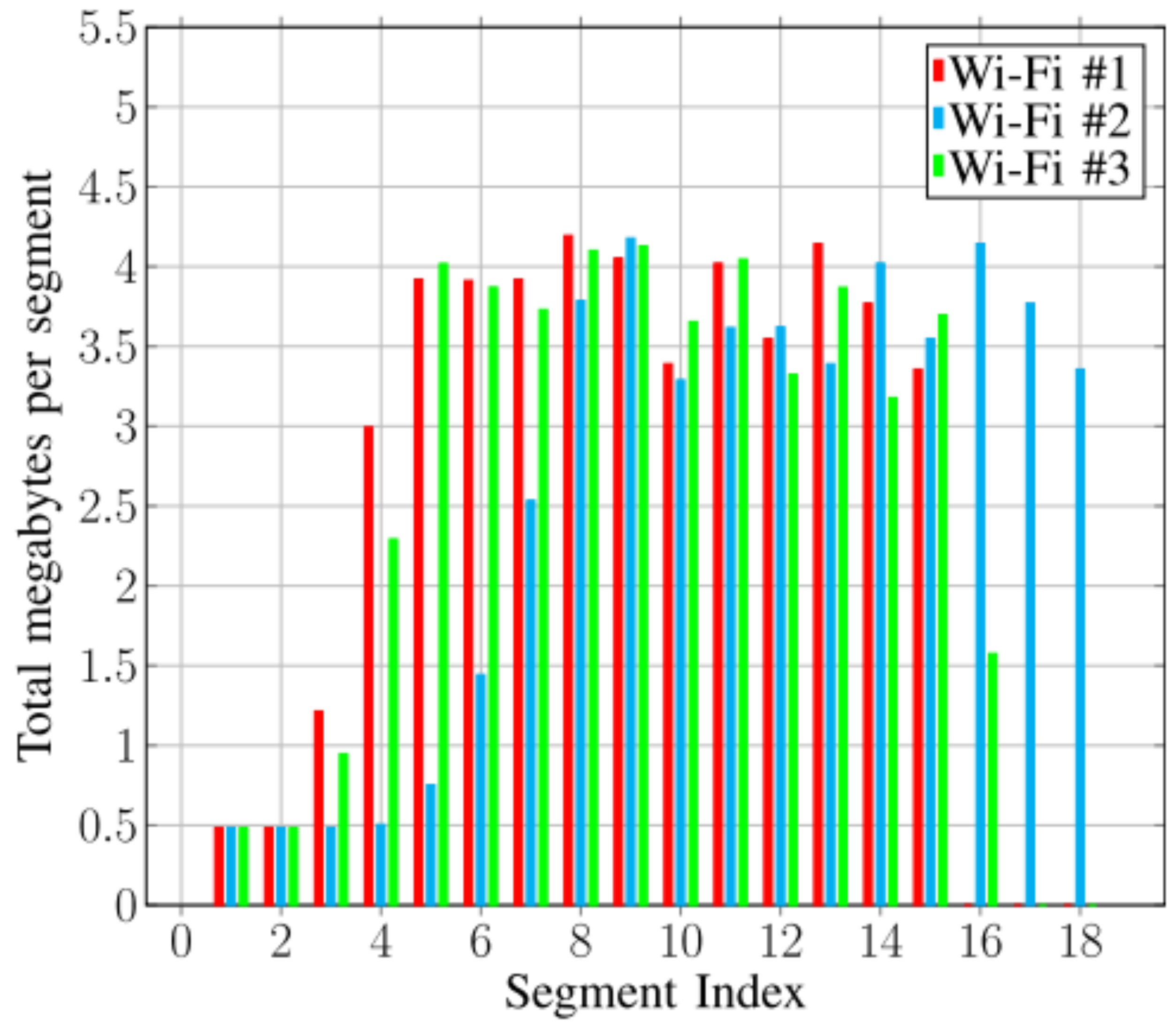
Traffic volume and timing allows to determine which video somebody is watching

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

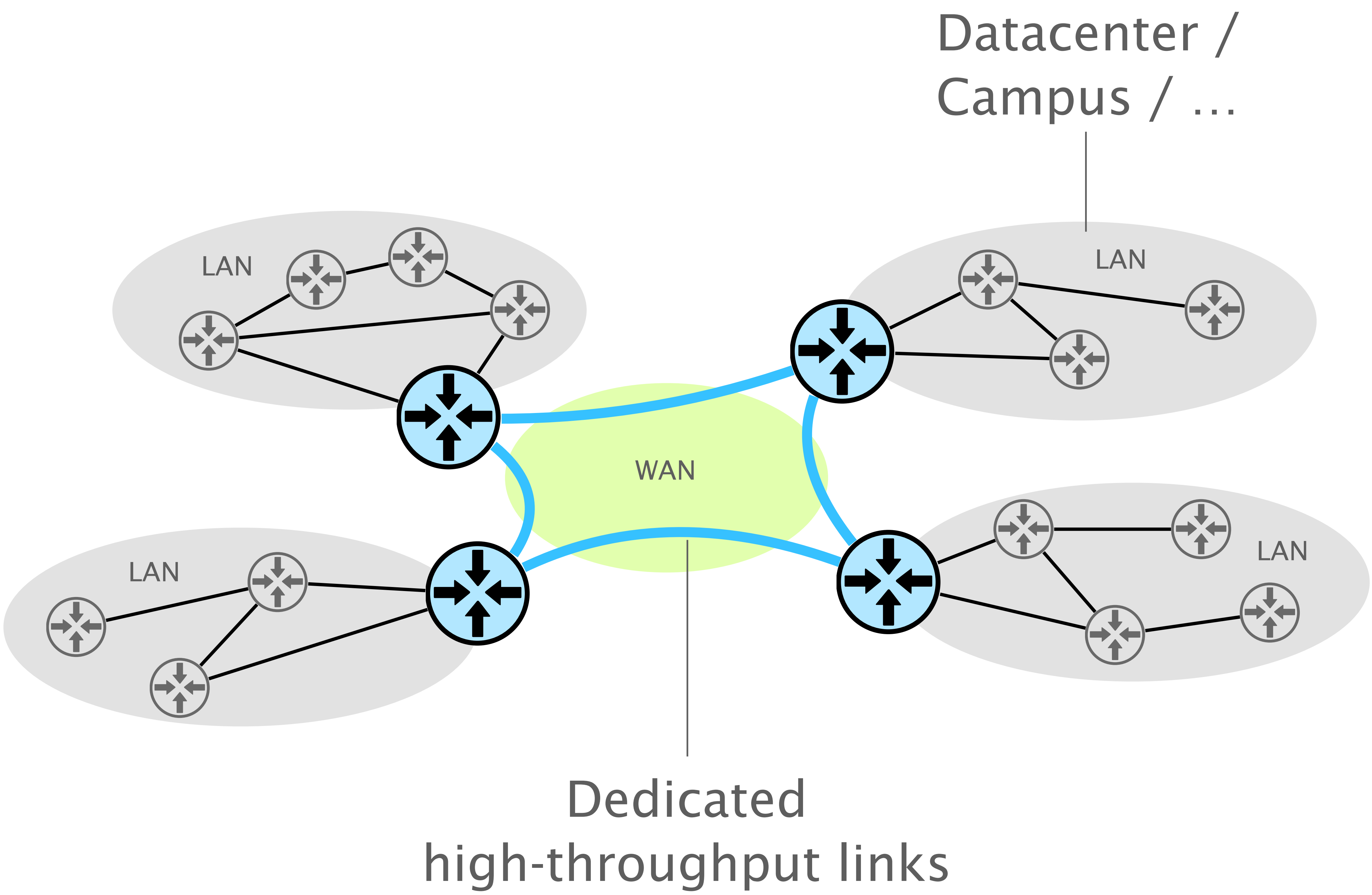
I Know What You Saw Last Minute—Encrypted HTTP Adaptive Video Streaming Title Classification

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

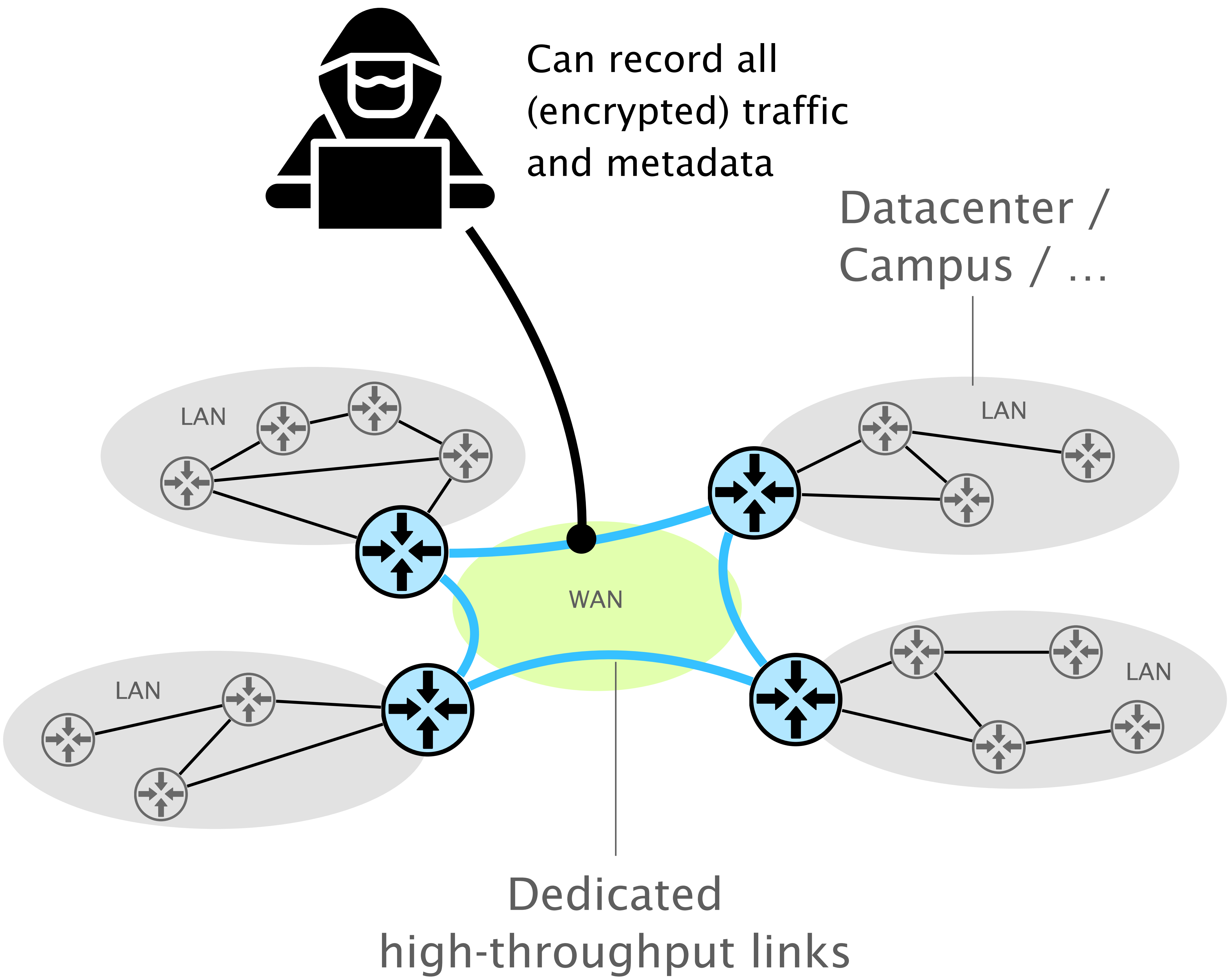
***Abstract*—Desktops can be exploited to violate privacy. There are two main types of attack scenarios: active and passive. We consider the passive scenario where the adversary does not interact actively with the device, but is able to eavesdrop on the network traffic of the device from the network 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,**



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



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



Three challenges for a practical WAN traffic-analysis prevention system

Three challenges for a practical WAN traffic-analysis prevention system

- Security
  Traffic does not leak information

Three challenges for a practical WAN traffic-analysis prevention system

- Security
  Traffic does not leak information

- Performance
  WANs run at 100s of Gbps

Three challenges for a practical WAN traffic-analysis prevention system

- Security
  Traffic does not leak information
- Performance
  WANs run at 100s of Gbps
- Deployability
  Infeasible to change all servers

Three challenges for a practical WAN traffic-analysis prevention system

- Security
  Traffic does not leak information
- Performance
  WANs run at 100s of Gbps
- Deployability
  Infeasible to change all servers

ditto makes observed traffic independent from the actual traffic

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

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

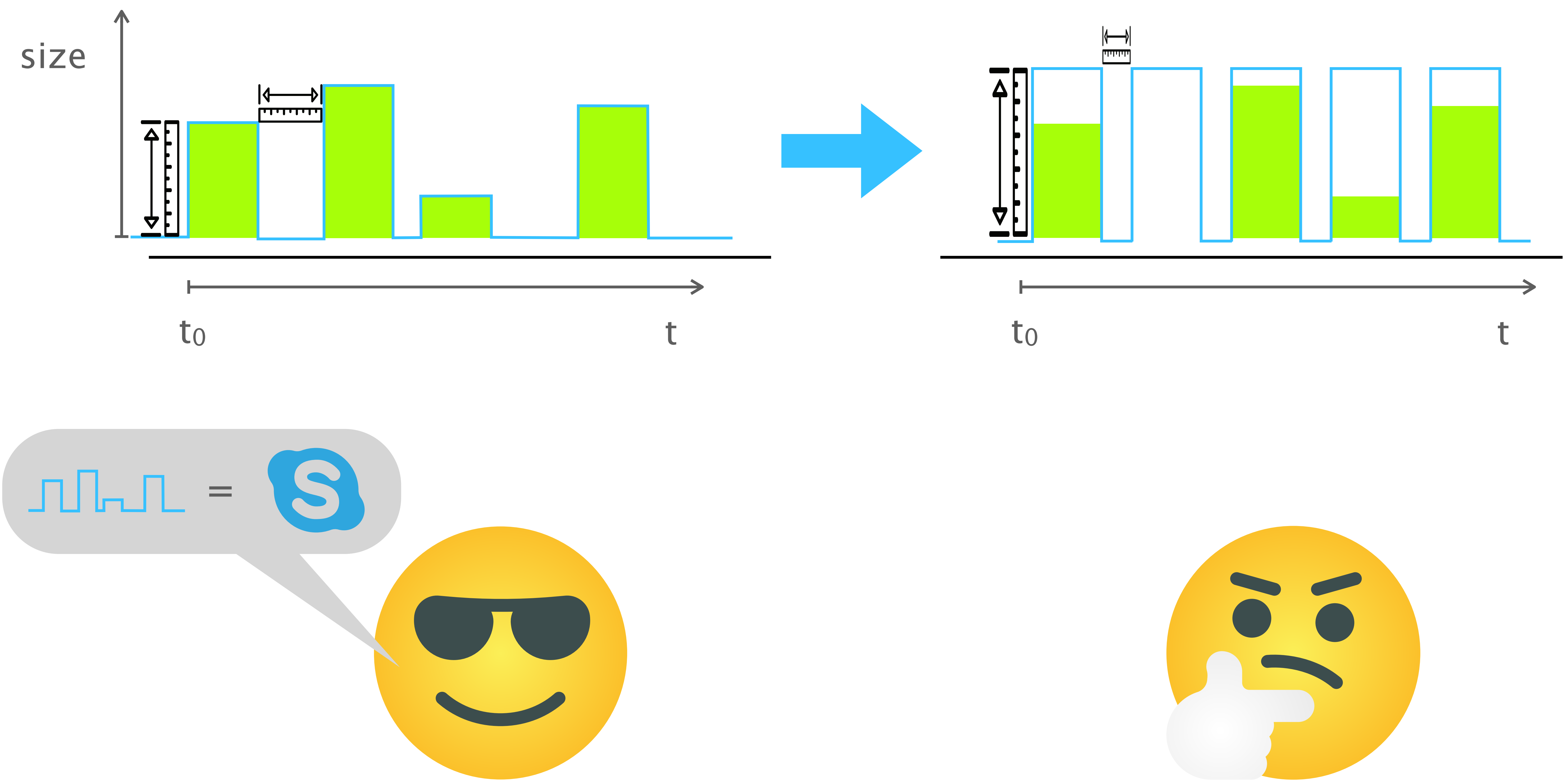
- Performance
  WANs run at 100s of Gbps

  ditto reduces overhead by using efficient traffic patterns

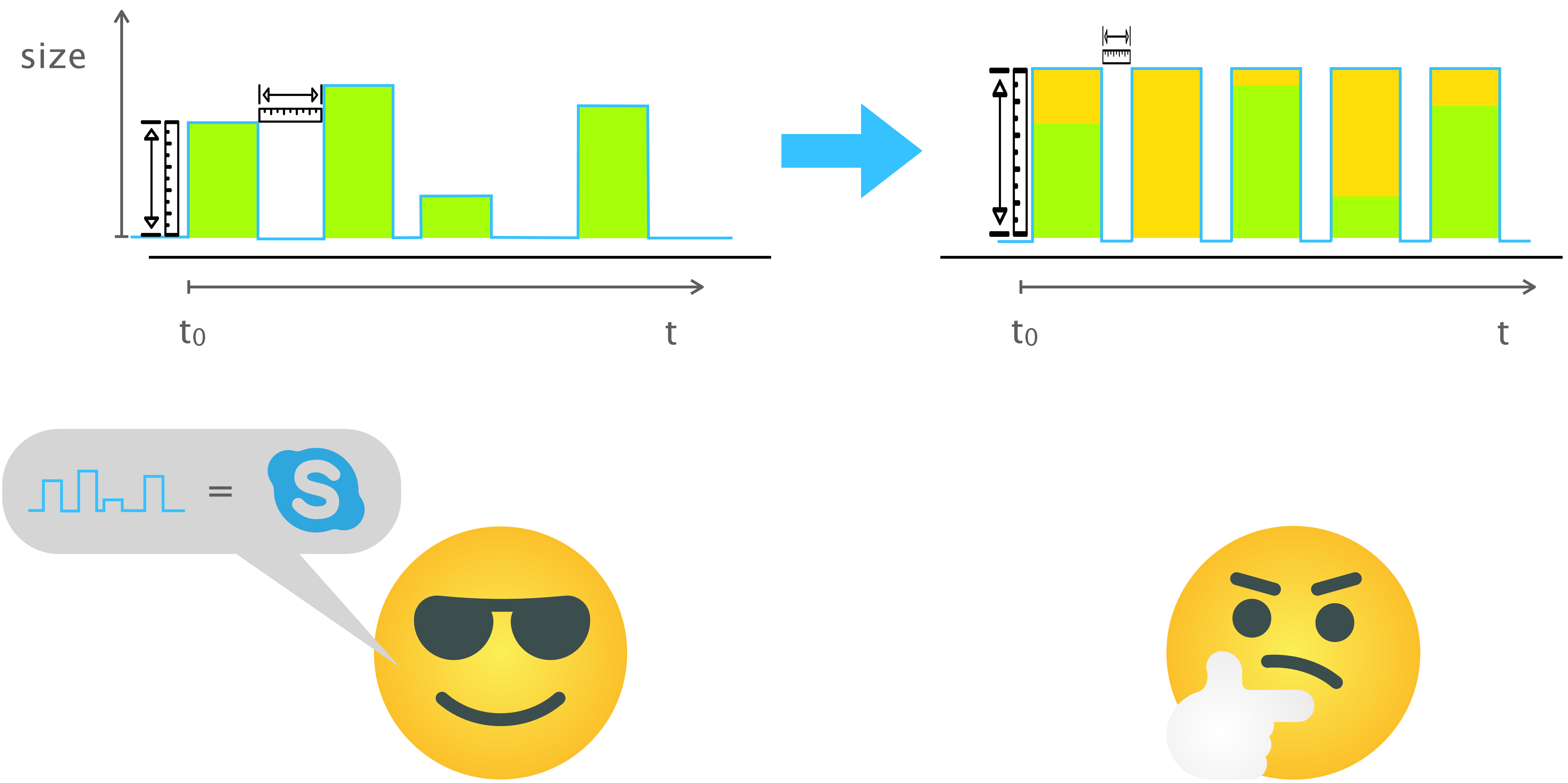
- Deployability
  Infeasible to change all servers

  ditto runs in the network data plane at line rate

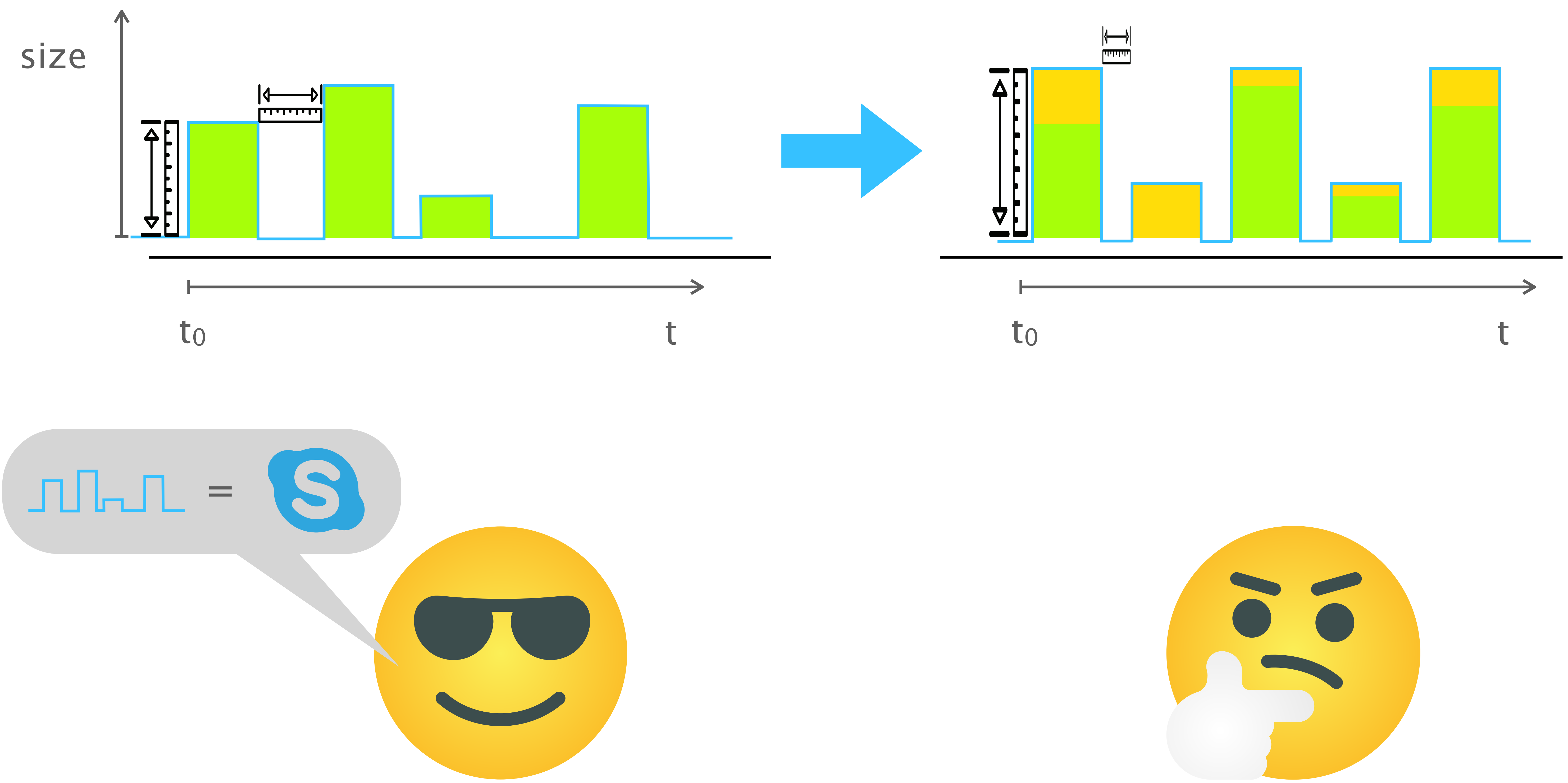
The high-level idea behind ditto is to make the observed traffic independent from the real traffic

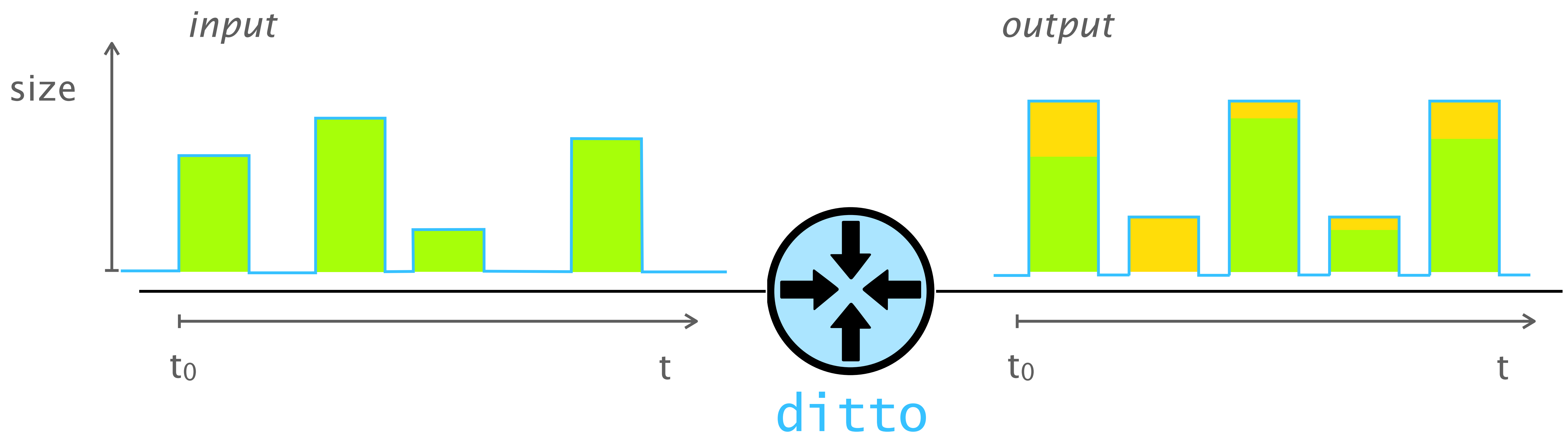


While secure, “constant” traffic can be inefficient



ditto shapes traffic according to an efficient pattern

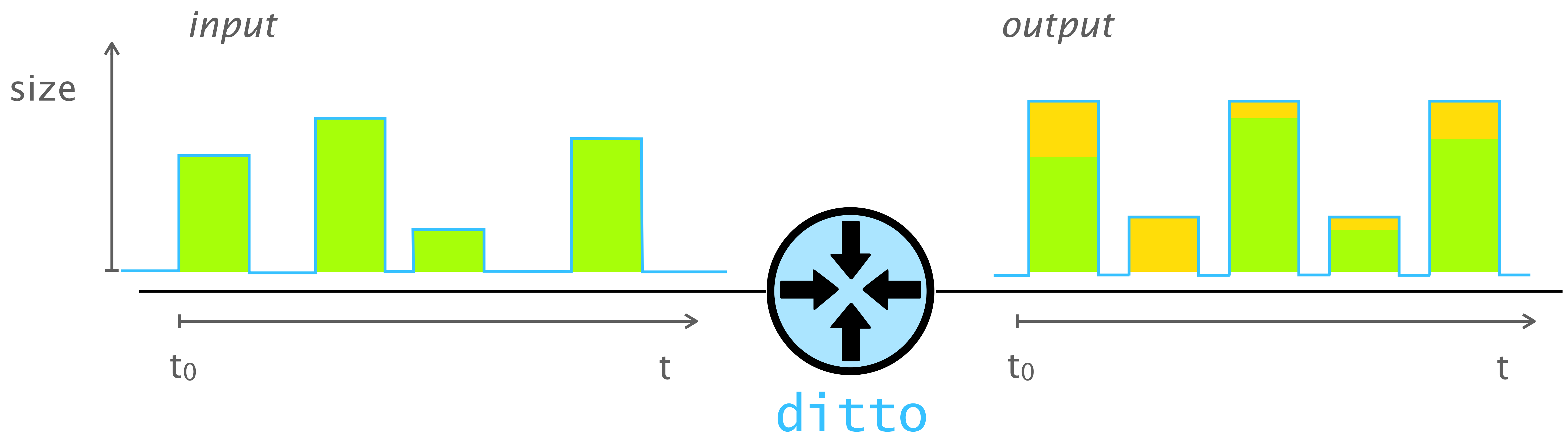




Computing efficient traffic patterns

Traffic shaping in the data plane

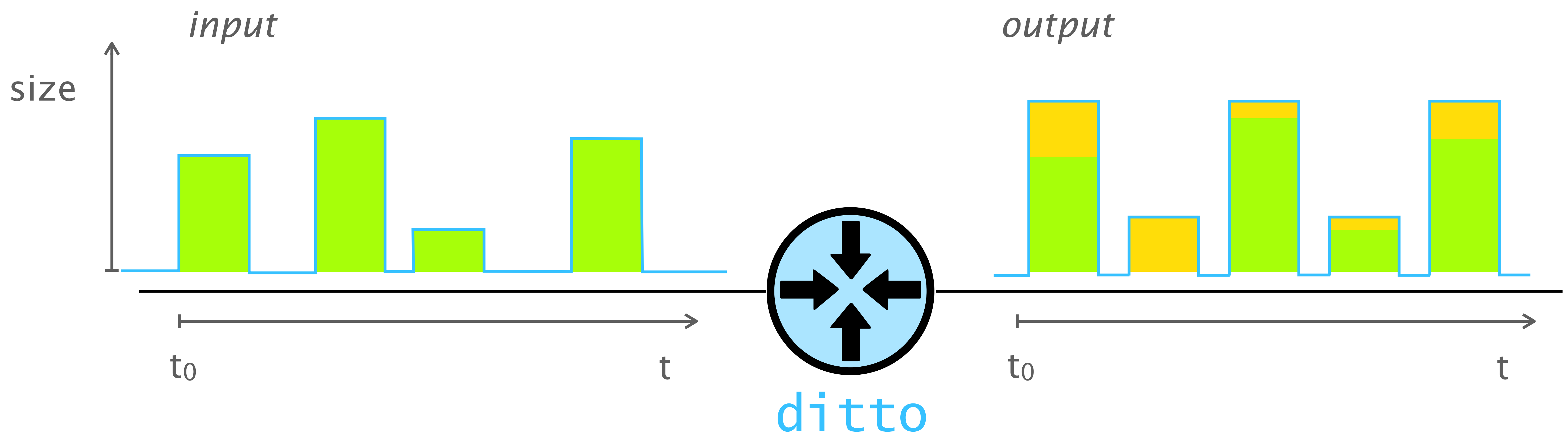
Experimental results



Computing efficient traffic patterns

Traffic shaping in the data plane

Experimental results

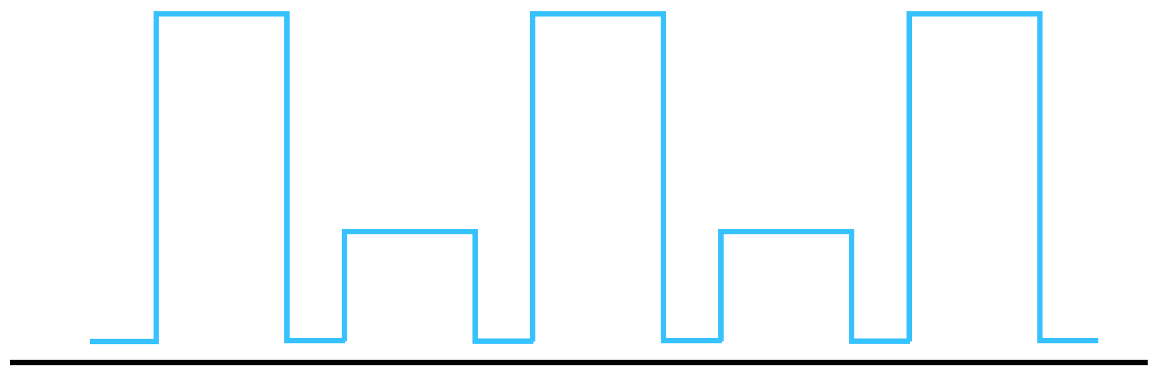


Computing efficient traffic patterns

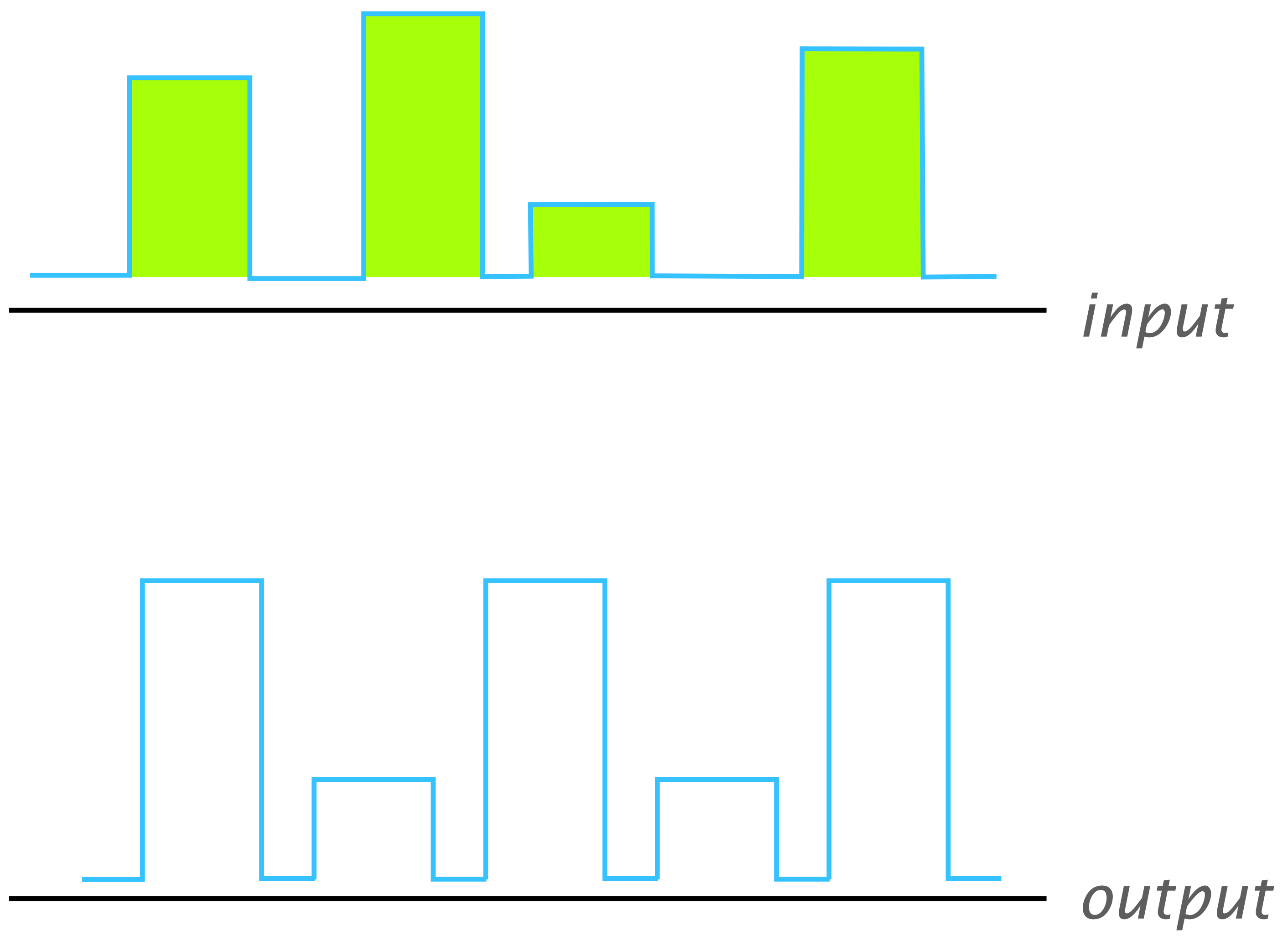
Traffic shaping in the data plane

Experimental results

ditto uses three operations to enforce the pattern at line rate

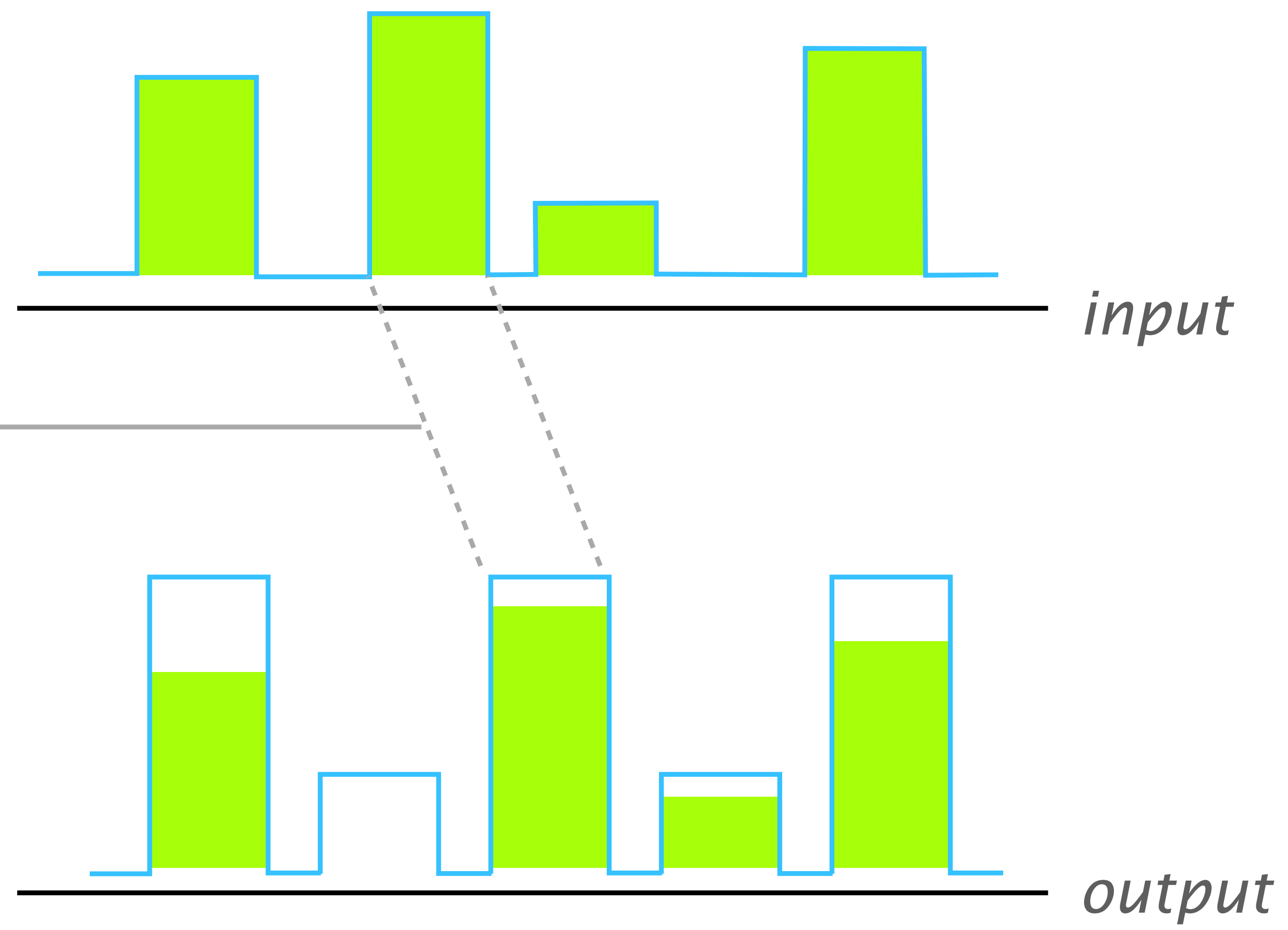


ditto uses three operations to enforce the pattern at line rate



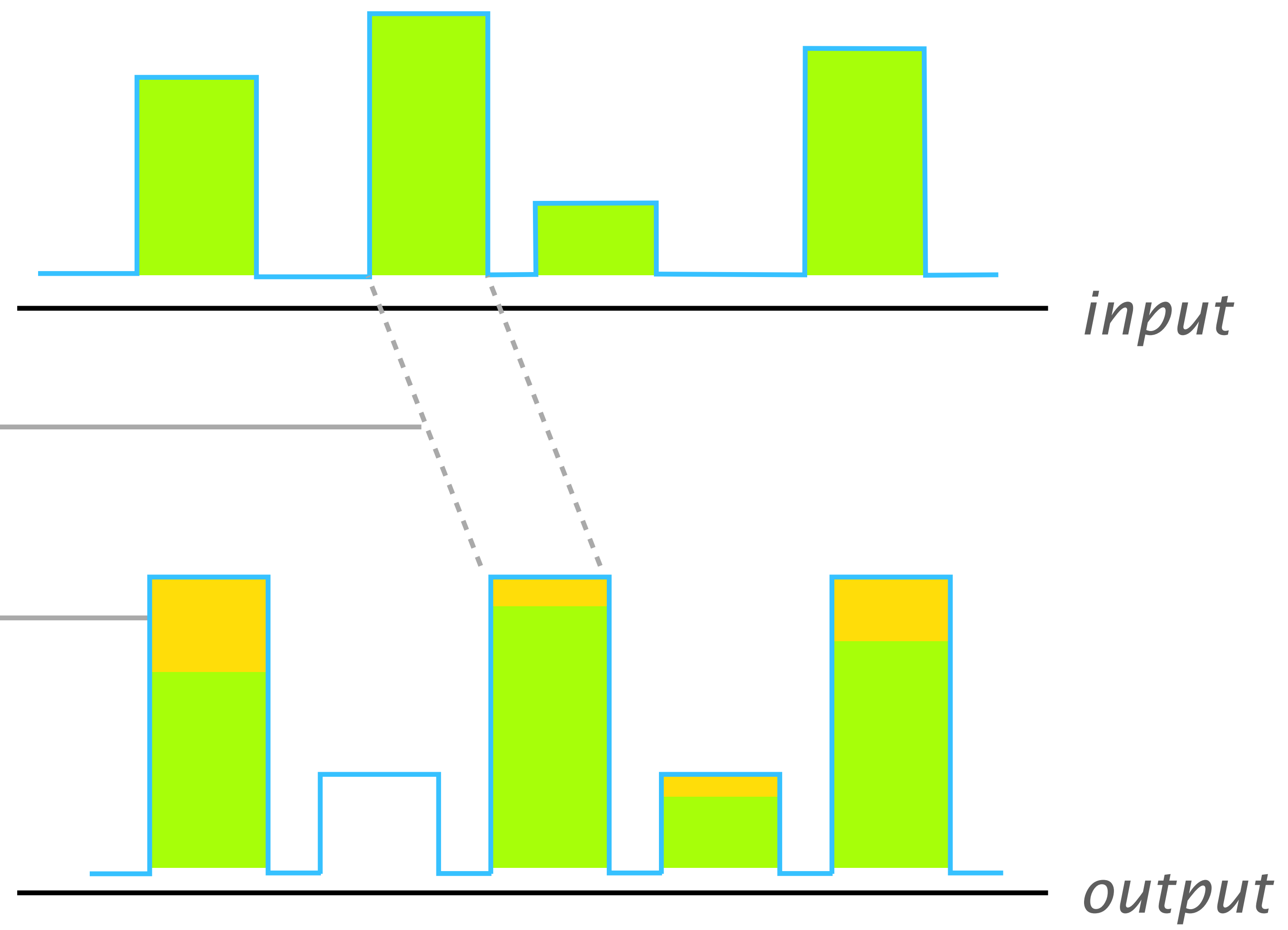
ditto uses three operations to enforce the pattern at line rate

- Buffering
  until a packet fits in the pattern



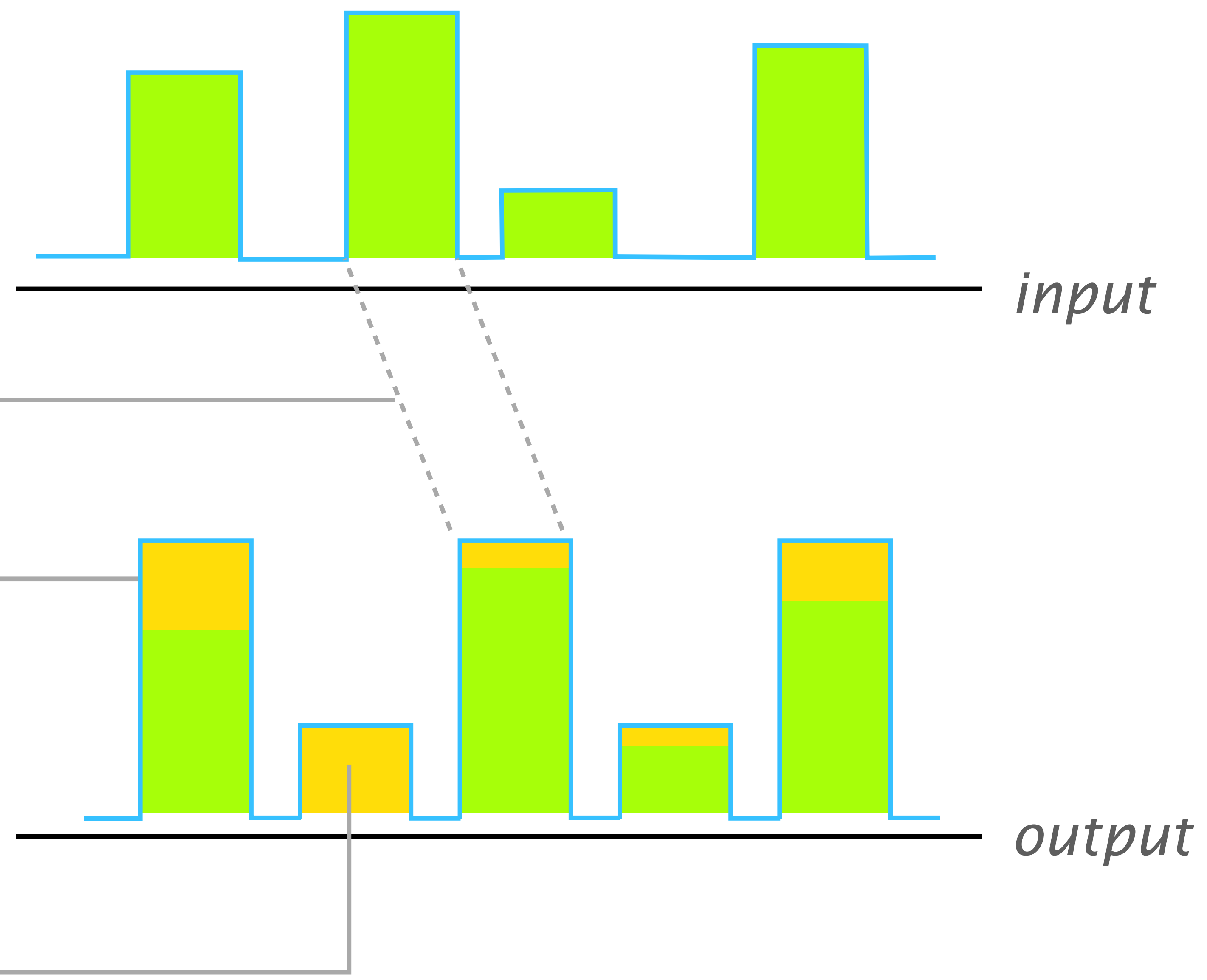
ditto uses three operations to enforce the pattern at line rate

- Buffering
  until a packet fits in the pattern
- Padding
  to make packets larger

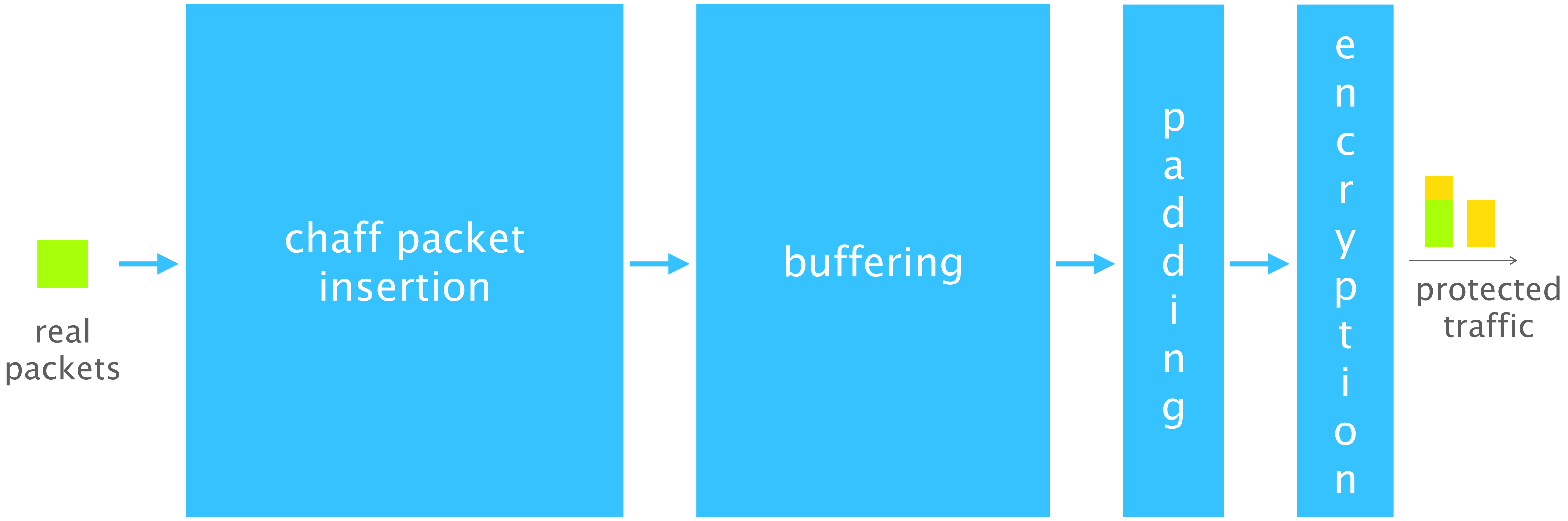


ditto uses three operations to enforce the pattern at line rate

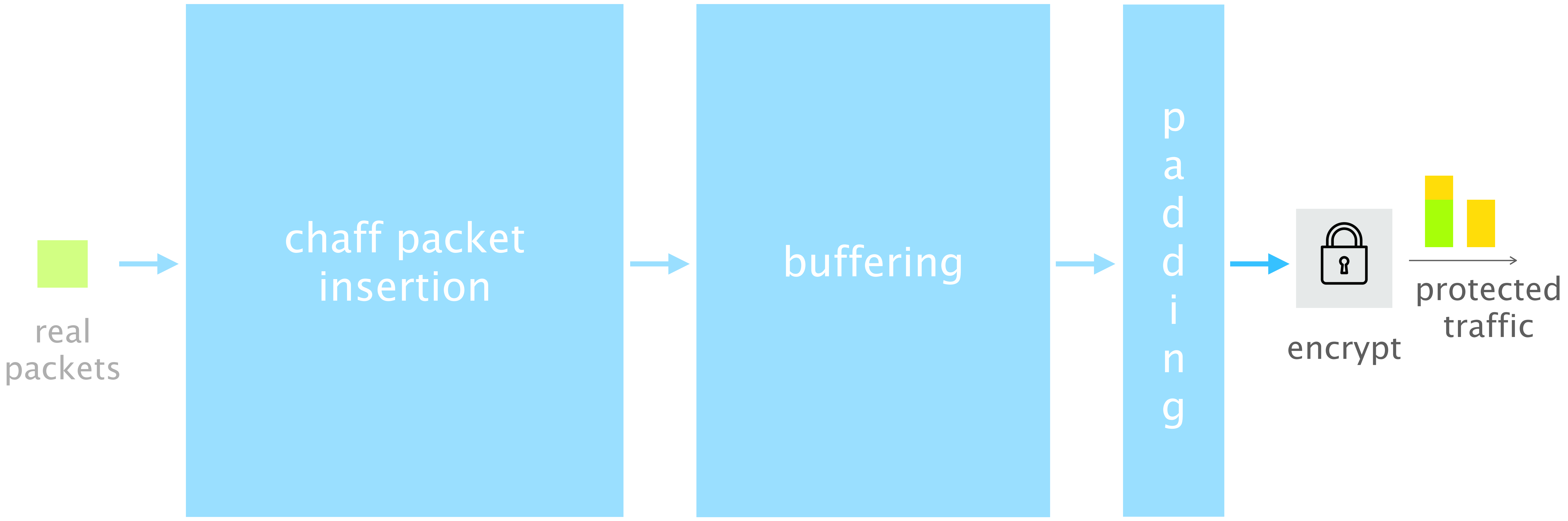
- **Buffering**
  until a packet fits in the pattern
- **Padding**
  to make packets larger
- **Chaff packet insertion**
  to fill gaps without real traffic



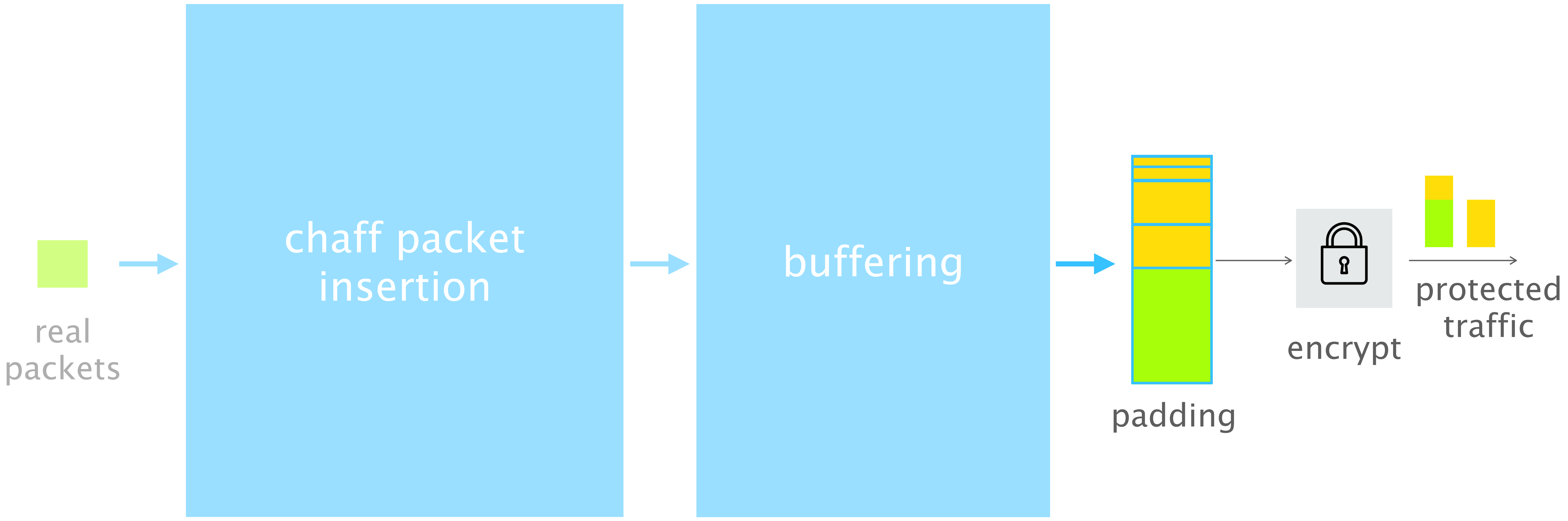
At a high level, ditto consists of 4 building blocks



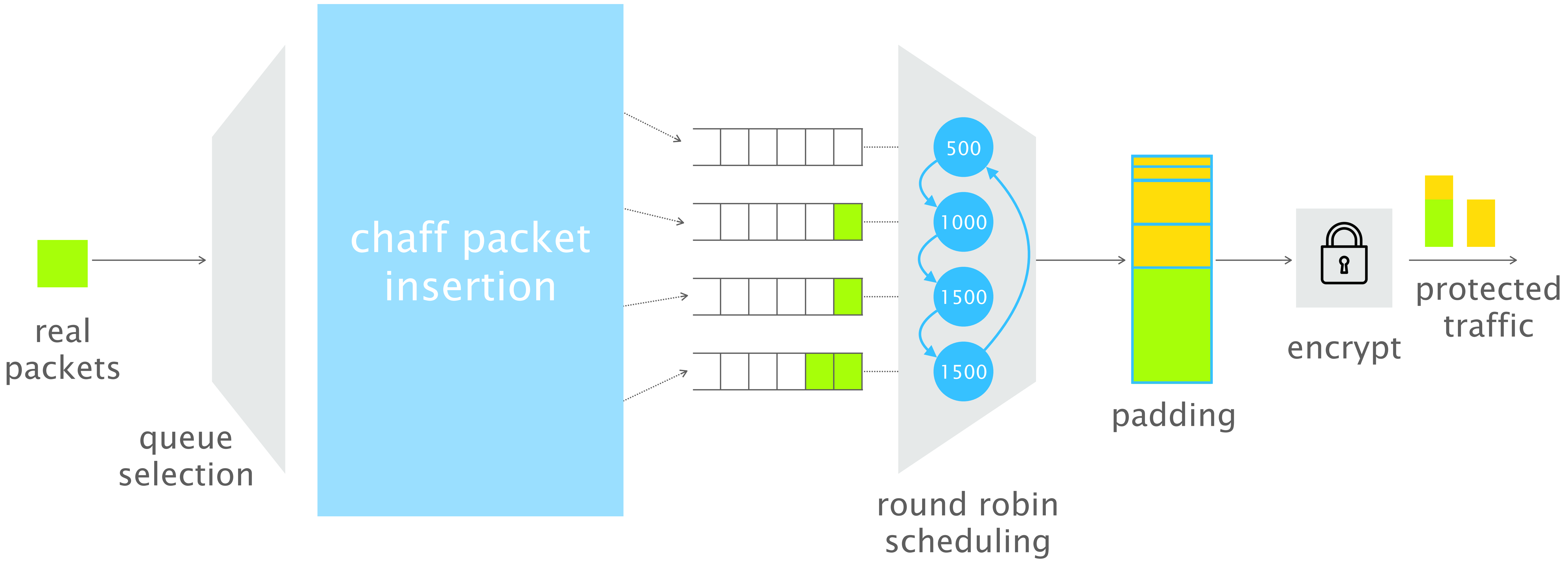
ditto sends traffic over encrypted tunnels (e.g., using MACsec)



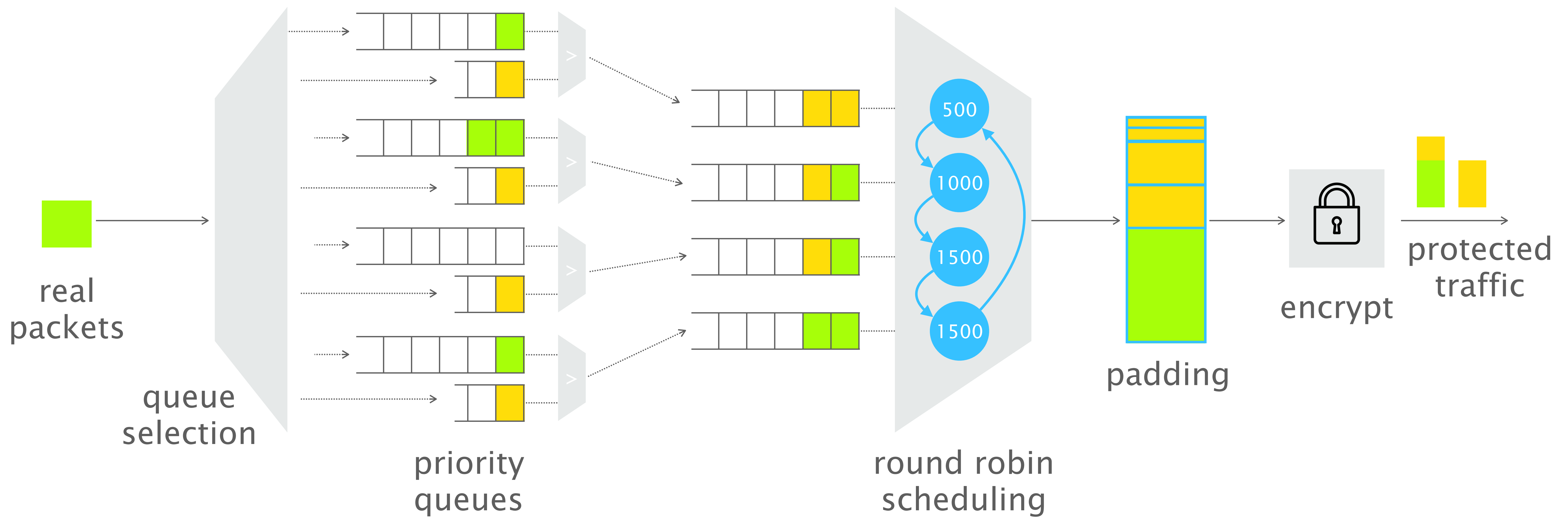
ditto pads packets by adding custom headers



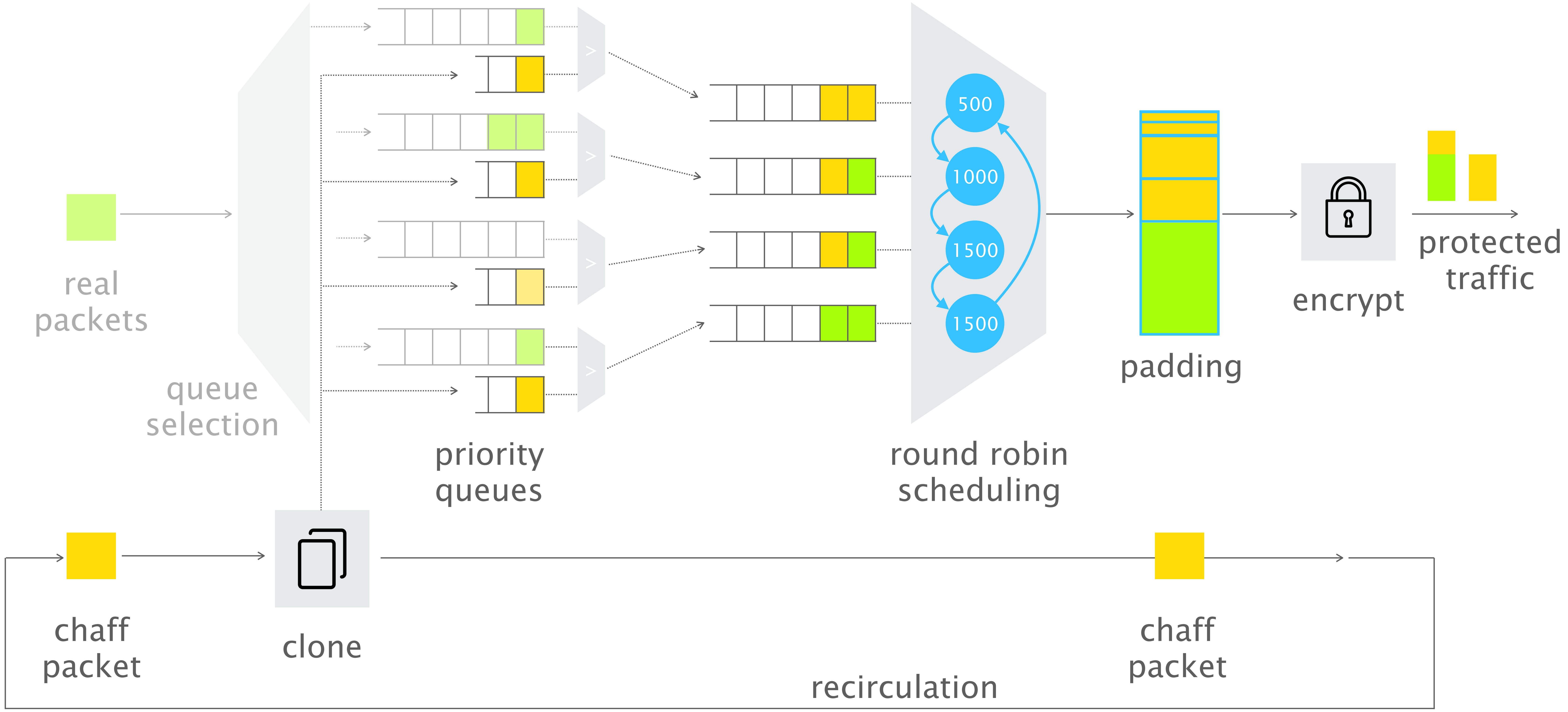
ditto uses round-robin scheduling to enforce the pattern



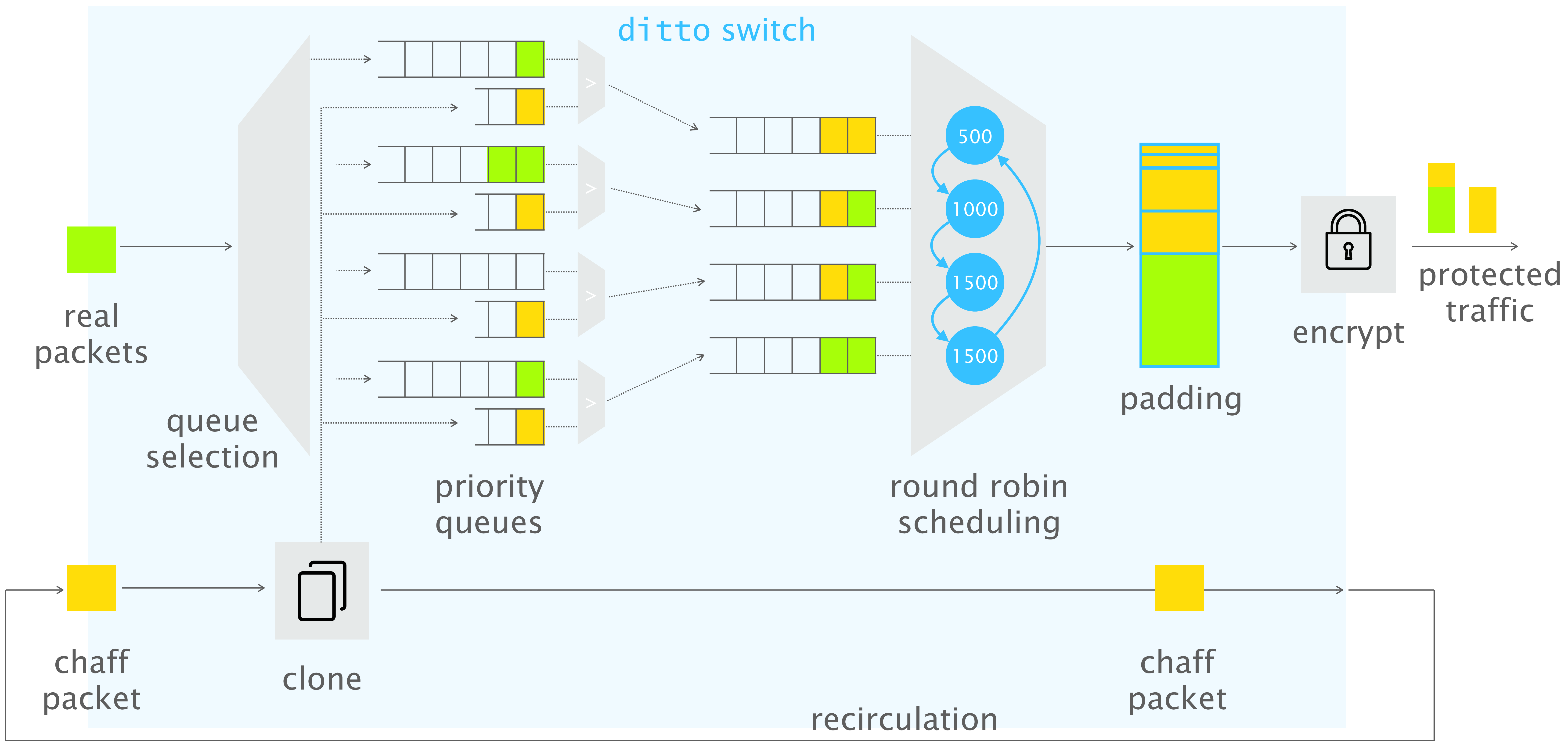
ditto uses priority queues to mix real and chaff packets

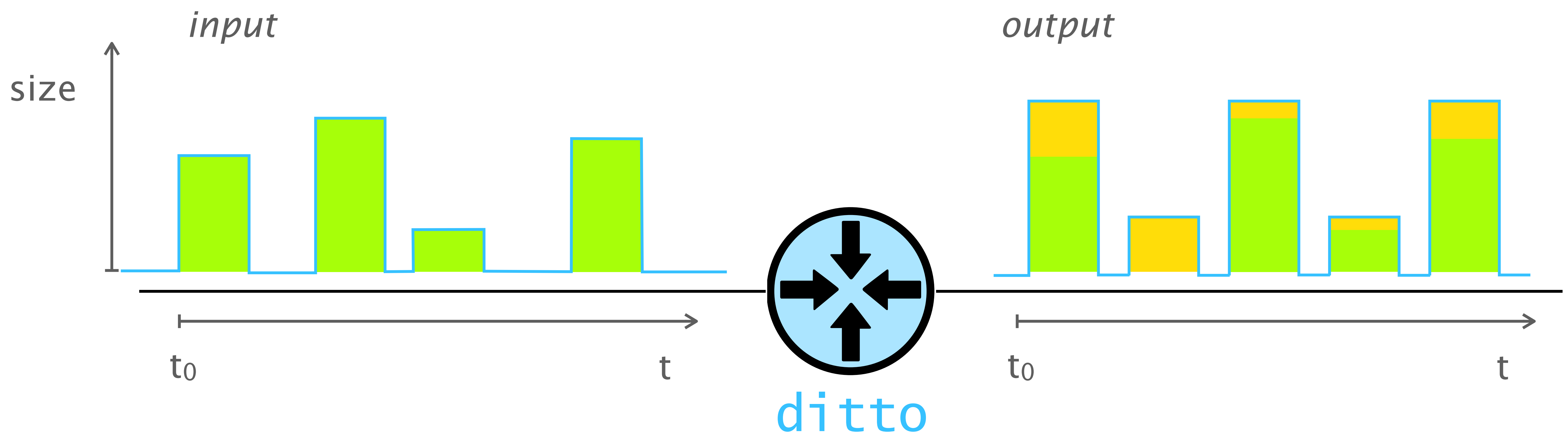


ditto generates chaff packets by recirculating and cloning them



ditto runs entirely in the data plane of programmable switches



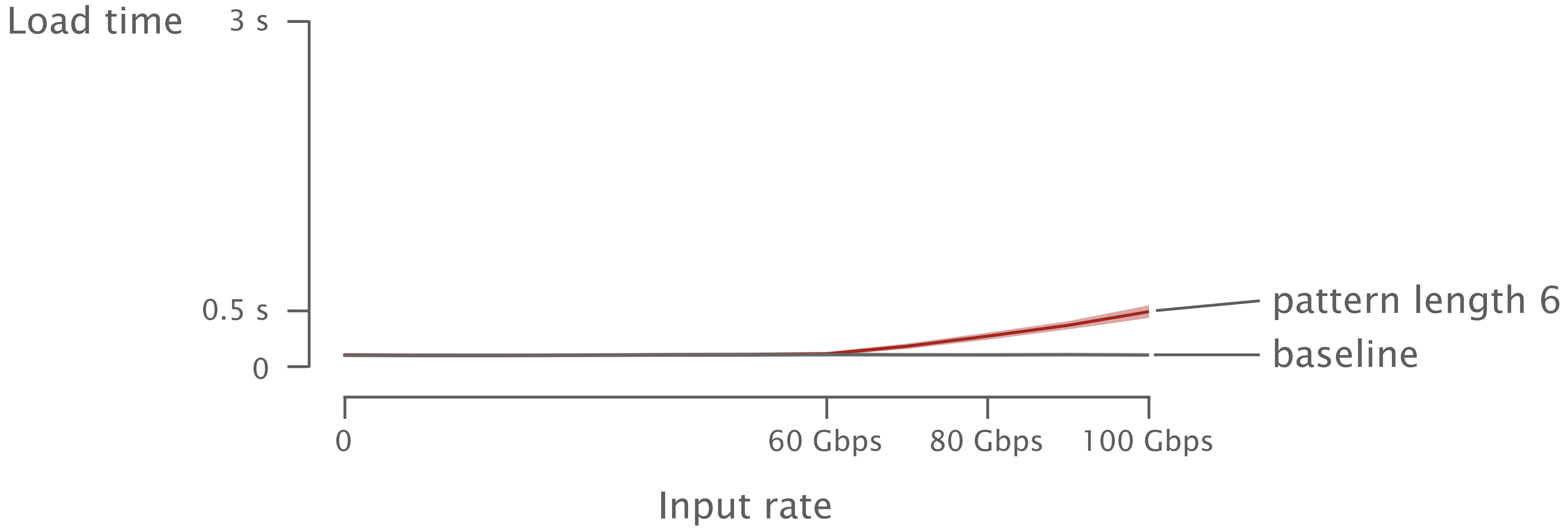


Computing efficient traffic patterns

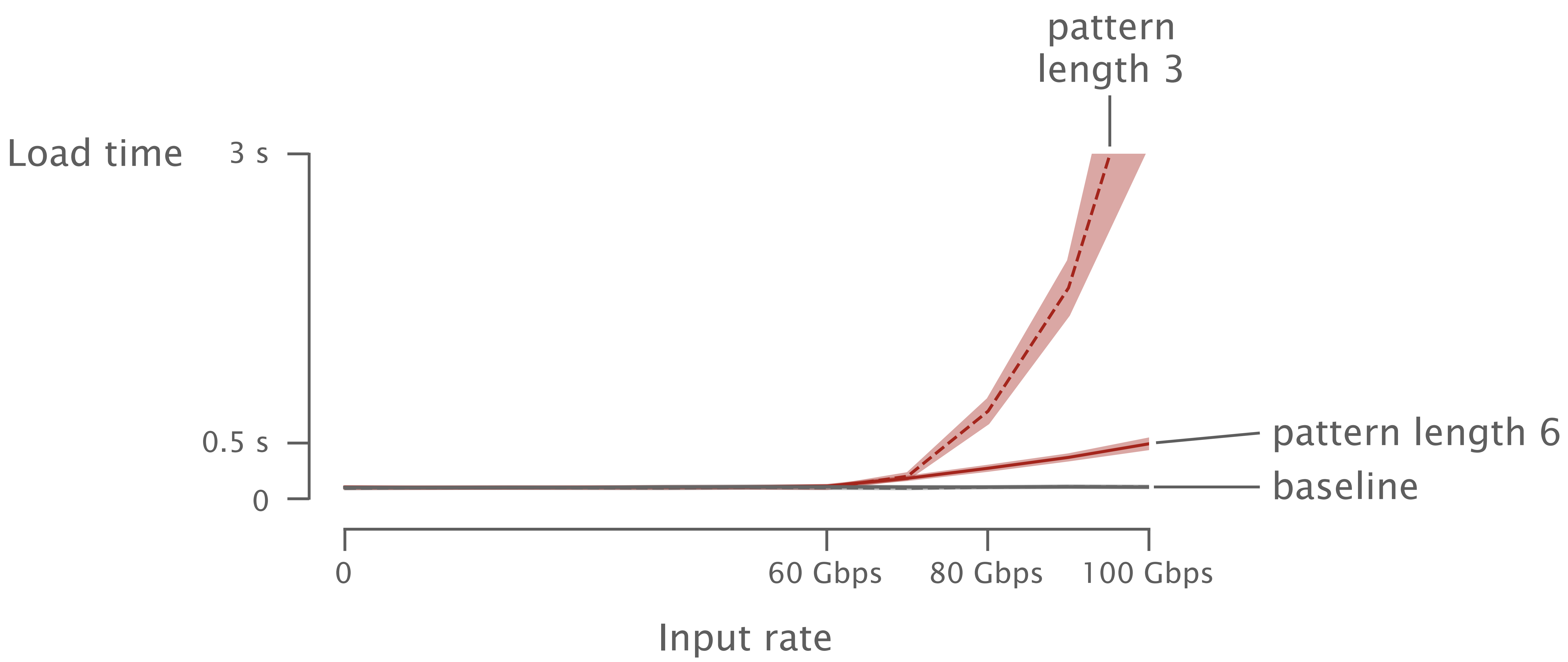
Traffic shaping in the data plane

Experimental results

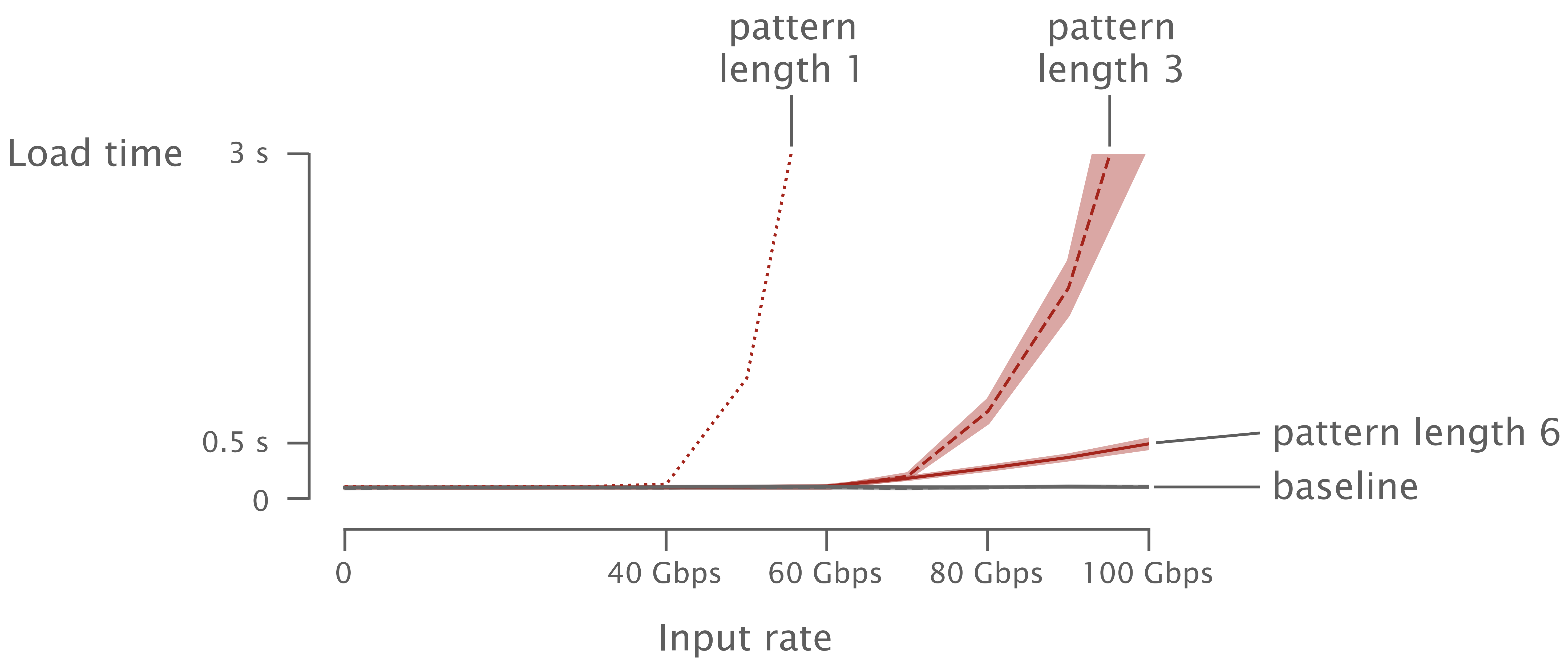
ditto does not affect the website load time up to 60 % load



Longer patterns achieve better performance



Longer patterns achieve better performance

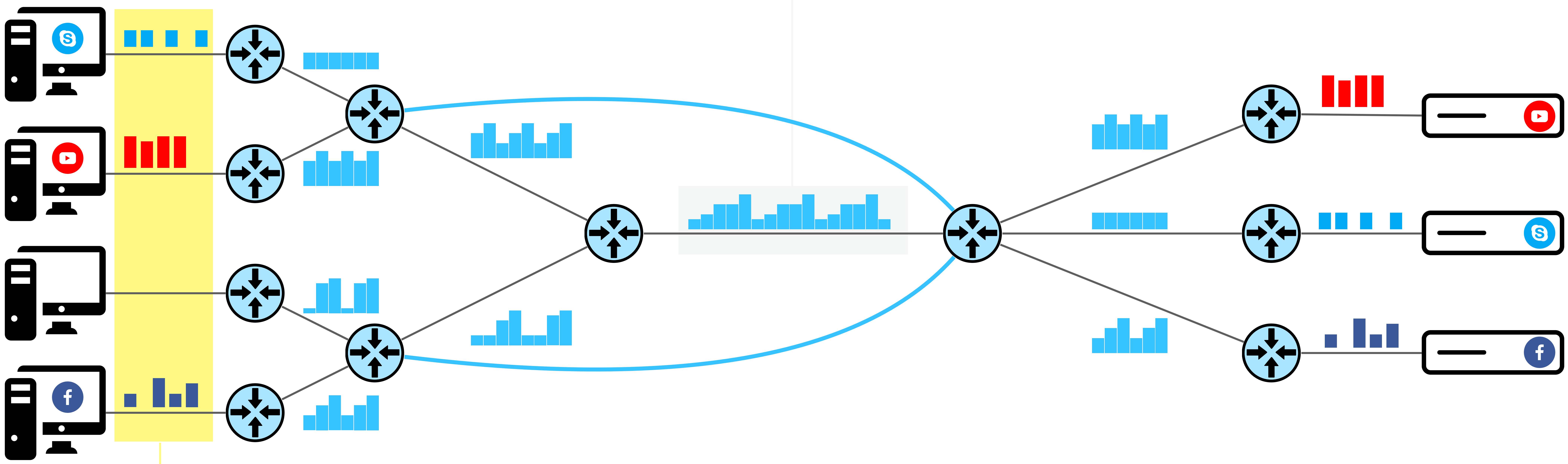


Problem #1

Traffic concentrates on one link

Vulnerable to denial-of-service attacks

NetHide prevents these attacks by obfuscating the topology



Encryption does not hide packet sizes and timings

Vulnerable to traffic-analysis attacks

ditto prevents these attacks by obfuscating the traffic

Problem #2

Topology obfuscation to prevent link-flooding attacks

NetHide

ditto

- Obfuscation through modified ICMP responses
  produced by programmable network devices
- Attacker cannot identify bottleneck links
  while debugging tools still work

Traffic obfuscation
to prevent traffic-analysis attacks

NetHide

ditto

- Obfuscation through traffic shaping
  at line rate in the data plane
- Attacker cannot identify real traffic
  because the observed traffic is independent

Traffic de-obfuscation
for benign and malicious purposes

NetHide

ditto

Traffic analysis

- Accelerating traffic-analysis attacks
  by extracting features in the data plane
- Finding participants of VoIP calls
  by identifying unique traffic signatures
- Classifying traffic at line rate
  by applying machine-learning models
- Identifying proxy servers
  by measuring response times

NetHide

Topology obfuscation
to prevent link-flooding attacks

[USENIX Security 2018]

ditto

Traffic obfuscation
to prevent traffic-analysis attacks

[NDSS 2022]

Traffic analysis

Traffic de-obfuscation
for benign and malicious purposes

[Arxiv 2019, SOSR 2022]