# 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

Know your enemy,

and you can fight 100 battles without fearing defeat.

— Sun Tzu

Know your DDoS enemy,

and you can fight 100 battles without fearing defeat.

— Sun Tzu

Introducing…

# Pulse-wave DDoS attacks

Pulse-wave DDoS attacks are

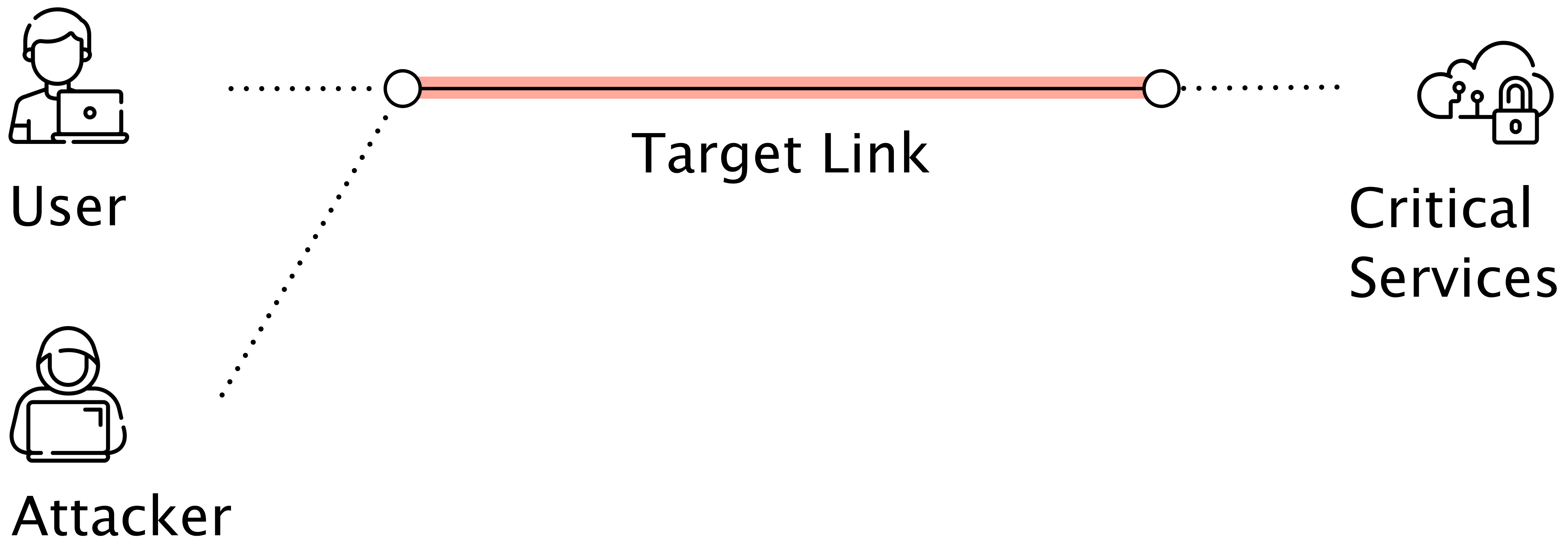
a new type of network-layer DDoS attack

Pulse-wave DDoS attacks are

a new type of network-layer DDoS attack

Target

a critical link

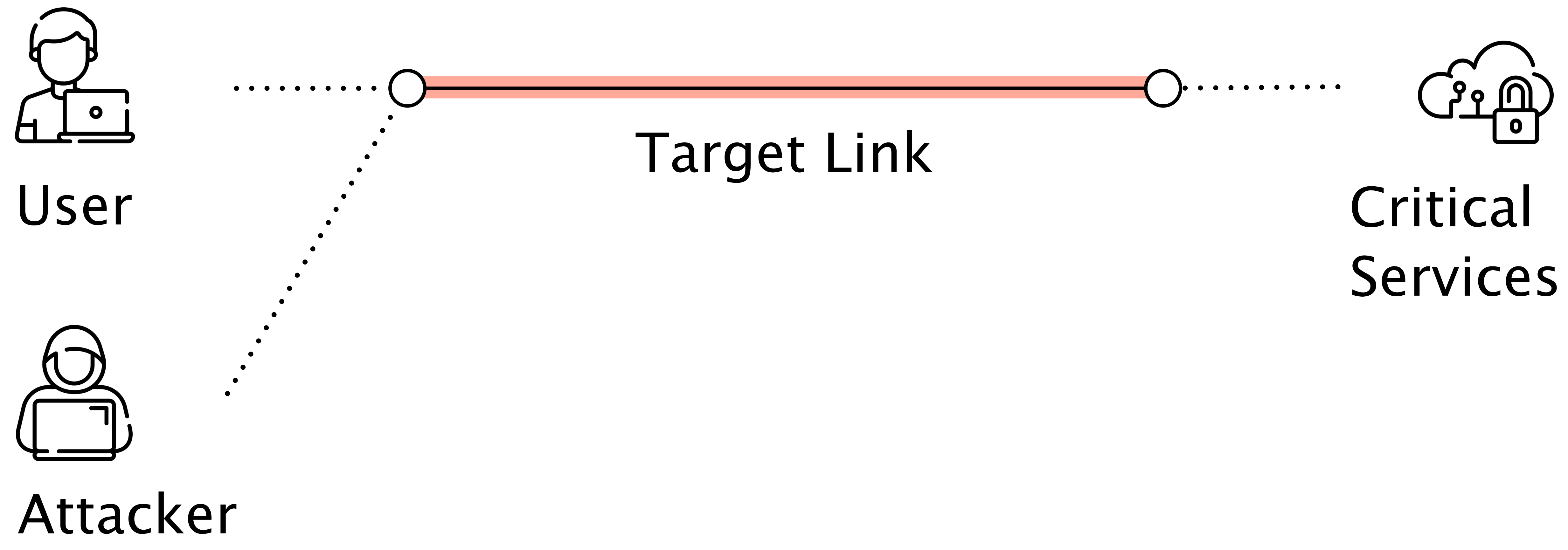


# Pulse-wave DDoS attacks are a new type of network-layer DDoS attack

Target
a critical link

Volumetric
(Gbps)

Multiple
attack vectors

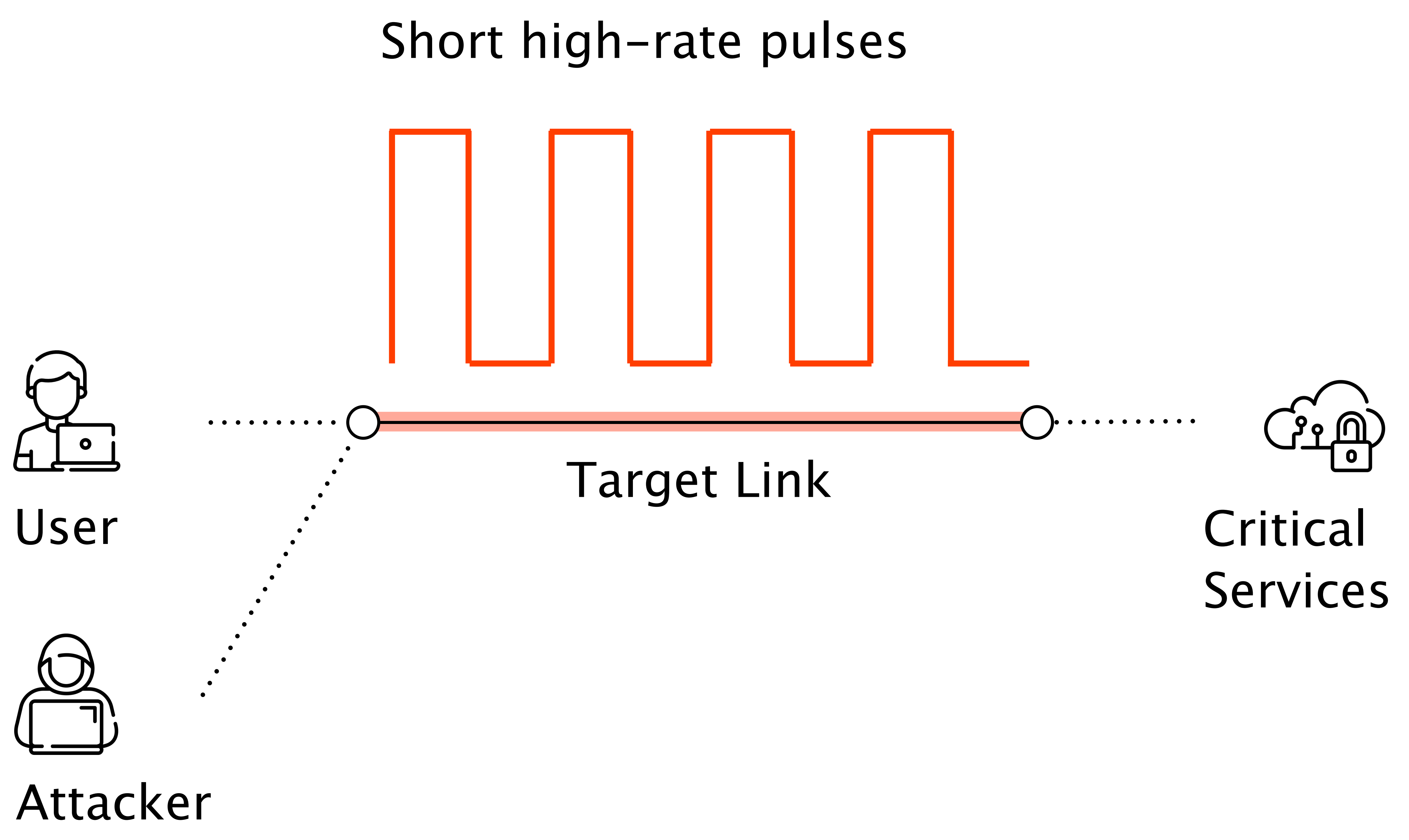


# Pulse-wave DDoS attacks are a new type of network-layer DDoS attack

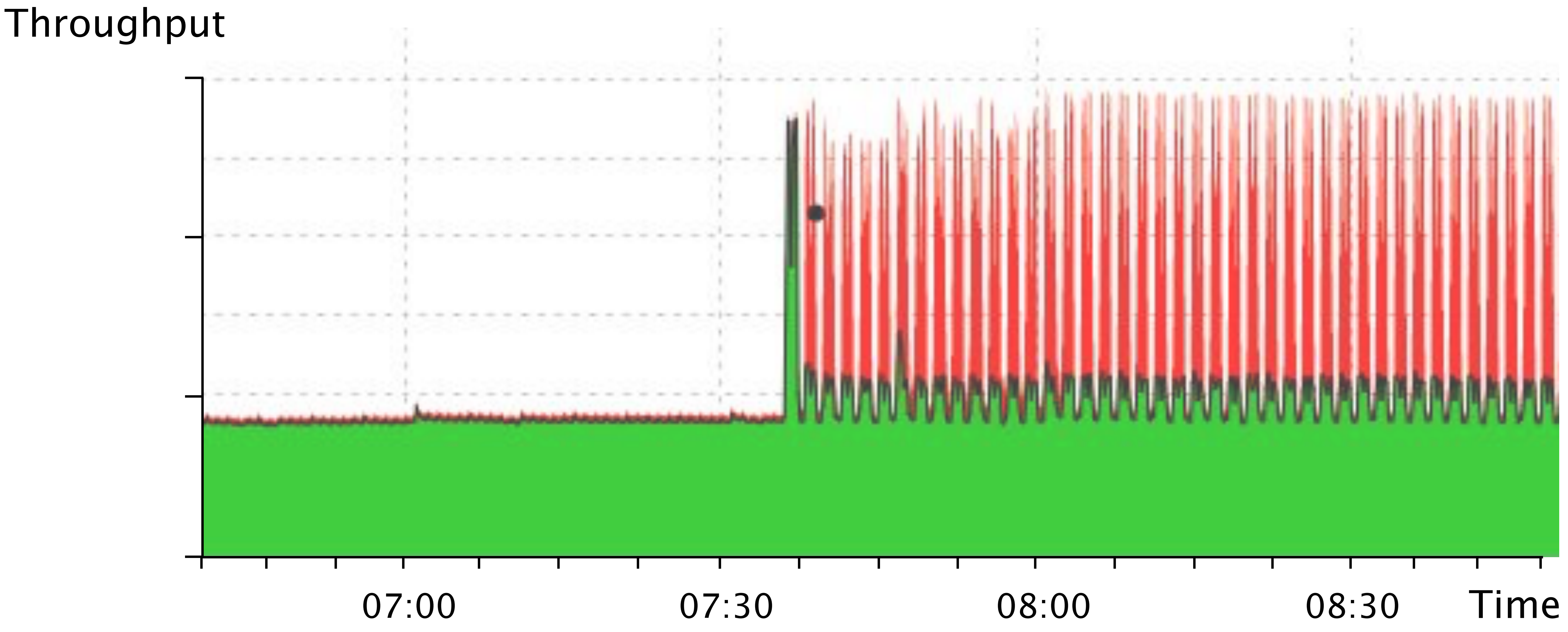
Target
a critical link

Volumetric
(Gbps)

Multiple
attack vectors

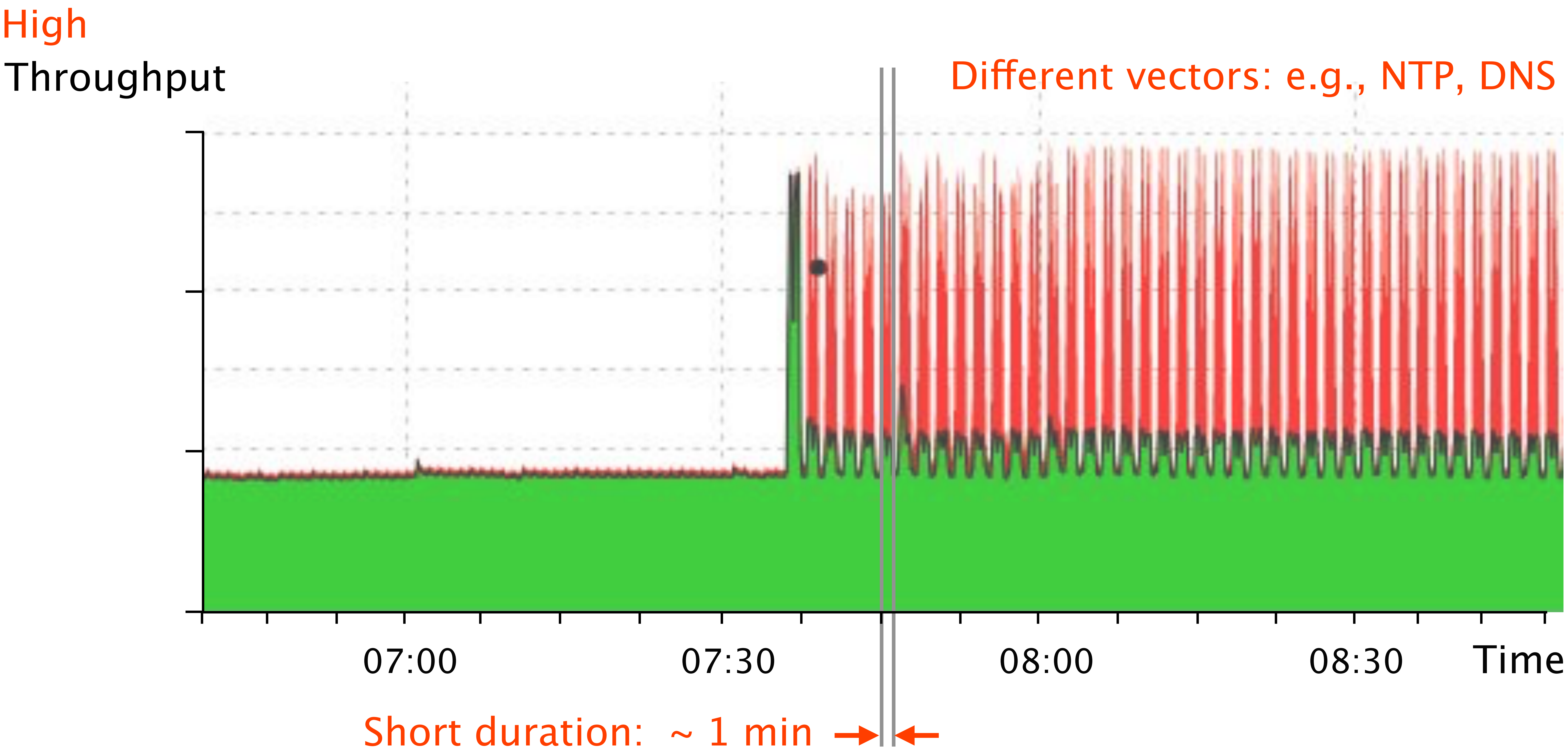


# Pulse-wave DDoS attacks are composed of short-duration high-rate traffic pulses




Damian Menscher, Google 2021

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



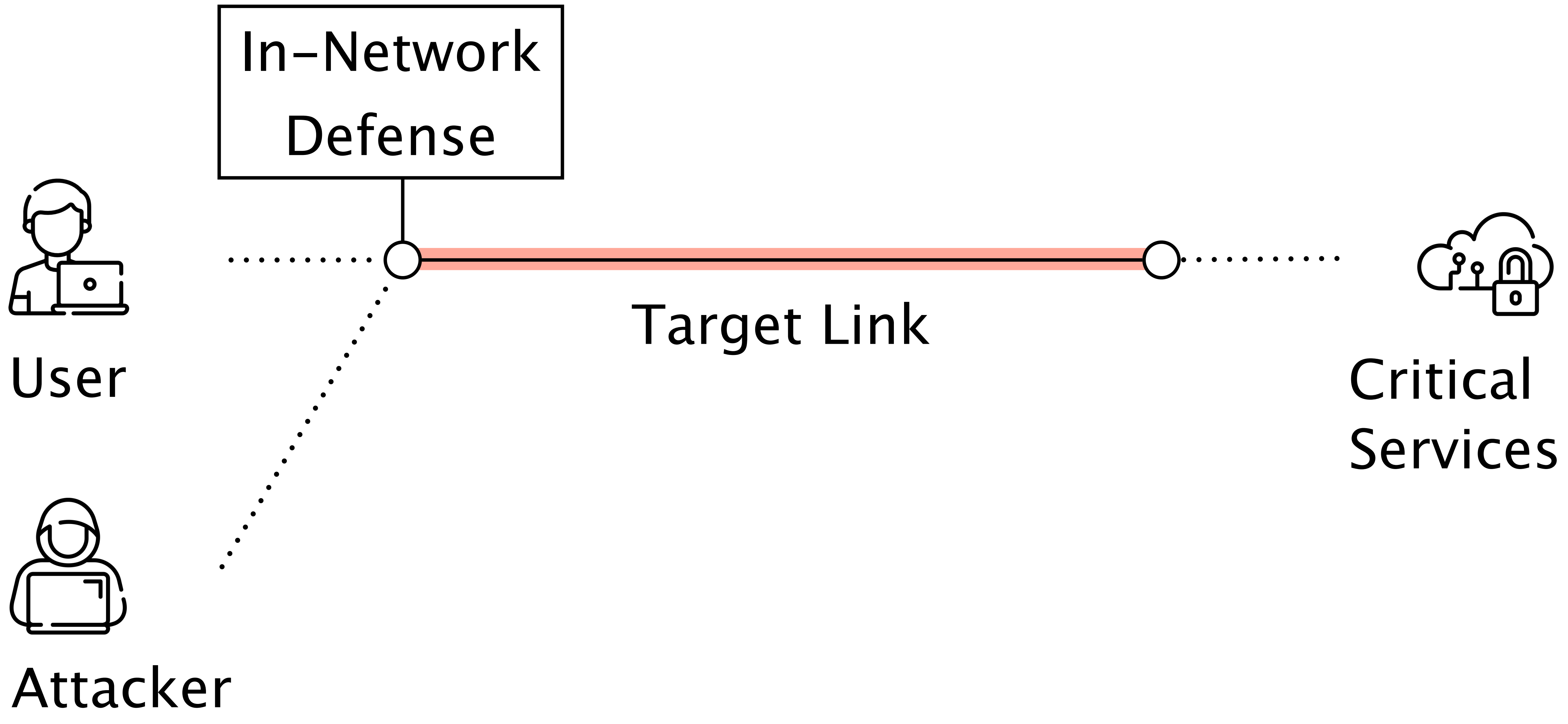

Damian Menscher, Google 2021

Pulse-wave DDoS attacks exploit the limitations of existing defenses

# Pulse-wave DDoS attacks exploit the limitations of existing defenses

Narrow
attack coverage

*Signature-based*
*Access-control lists*



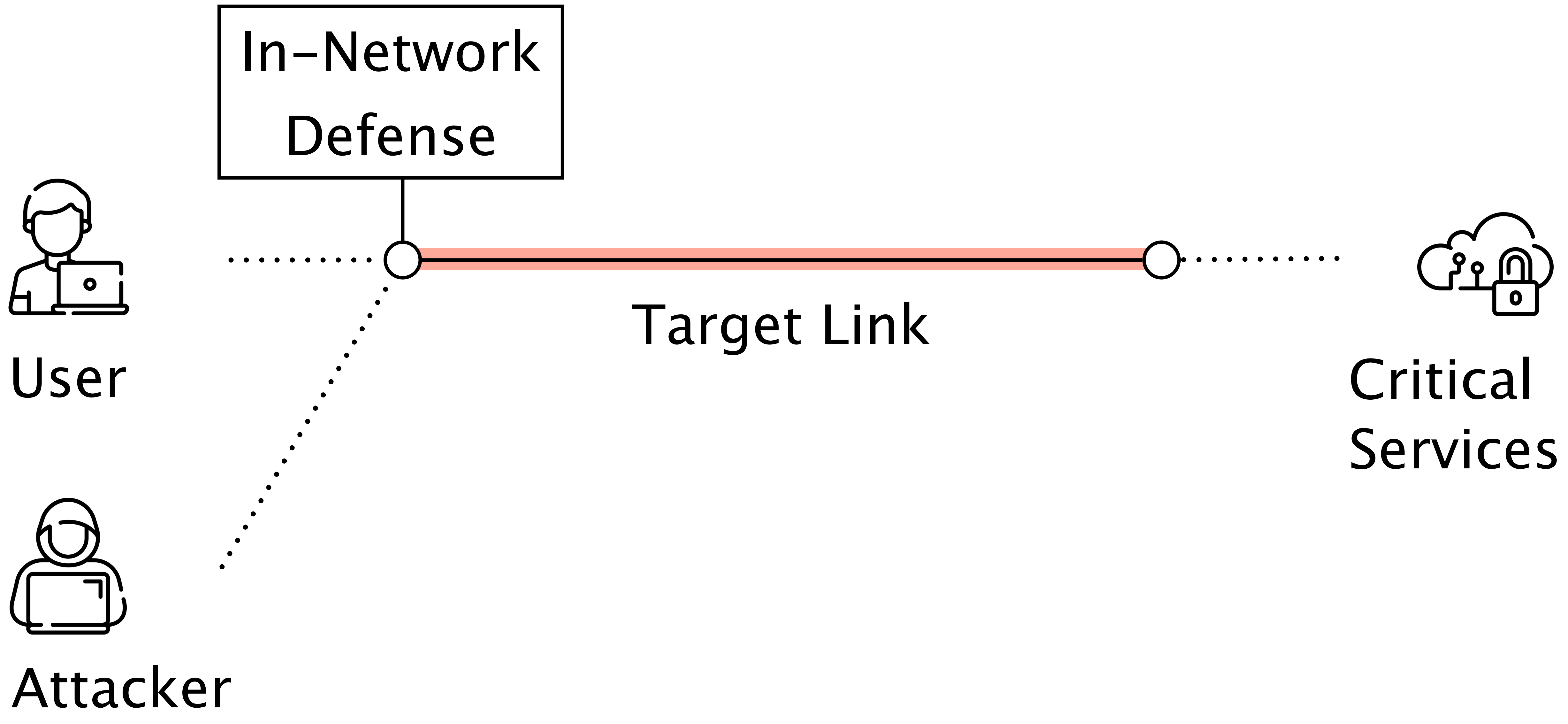
# Pulse-wave DDoS attacks exploit the limitations of existing defenses

Narrow attack coverage

*Filter-based*

*Rerouting-based*

Drastic mitigation

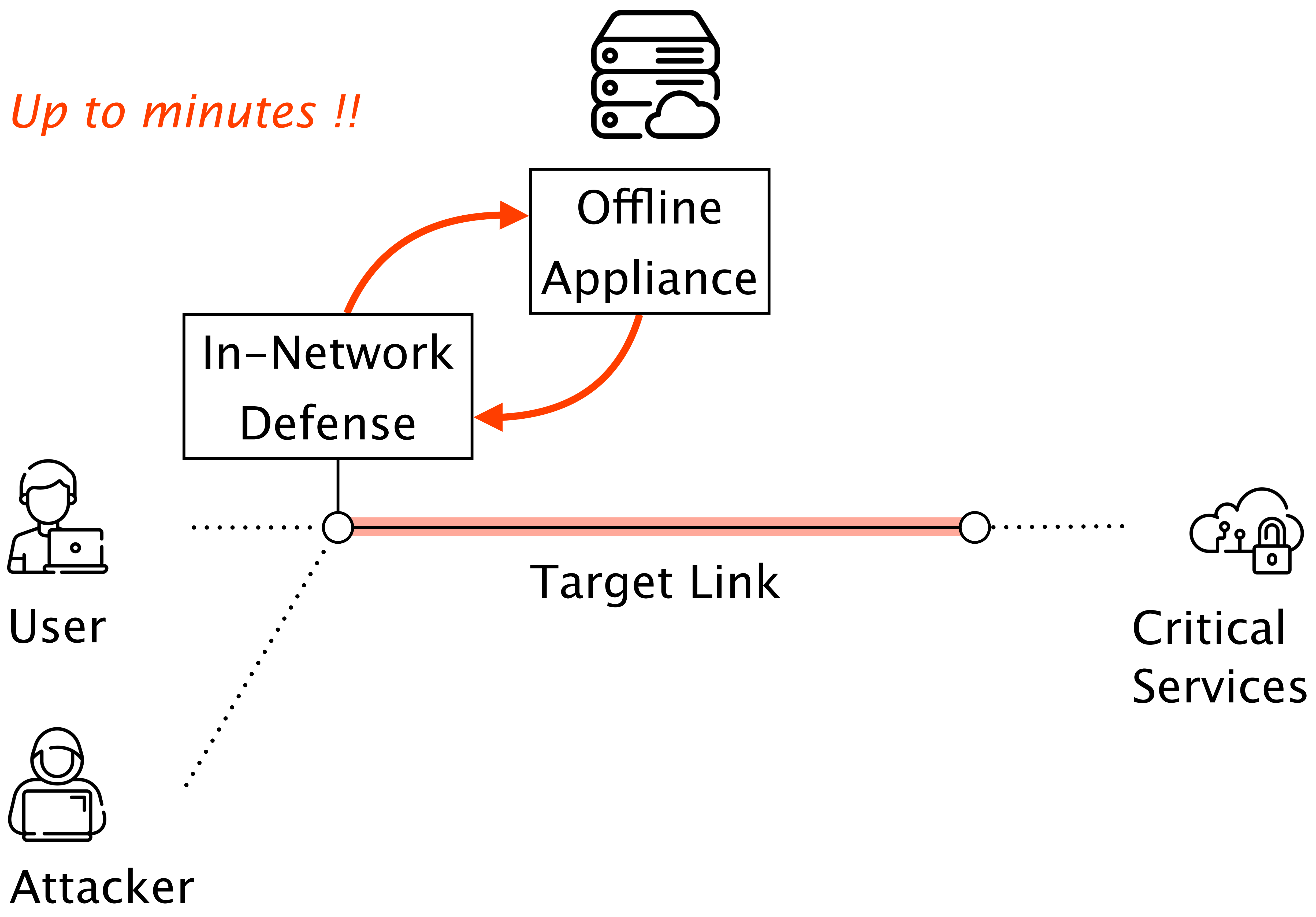


# Pulse-wave DDoS attacks exploit the limitations of existing defenses

Narrow attack coverage

Drastic mitigation

Slow reaction time



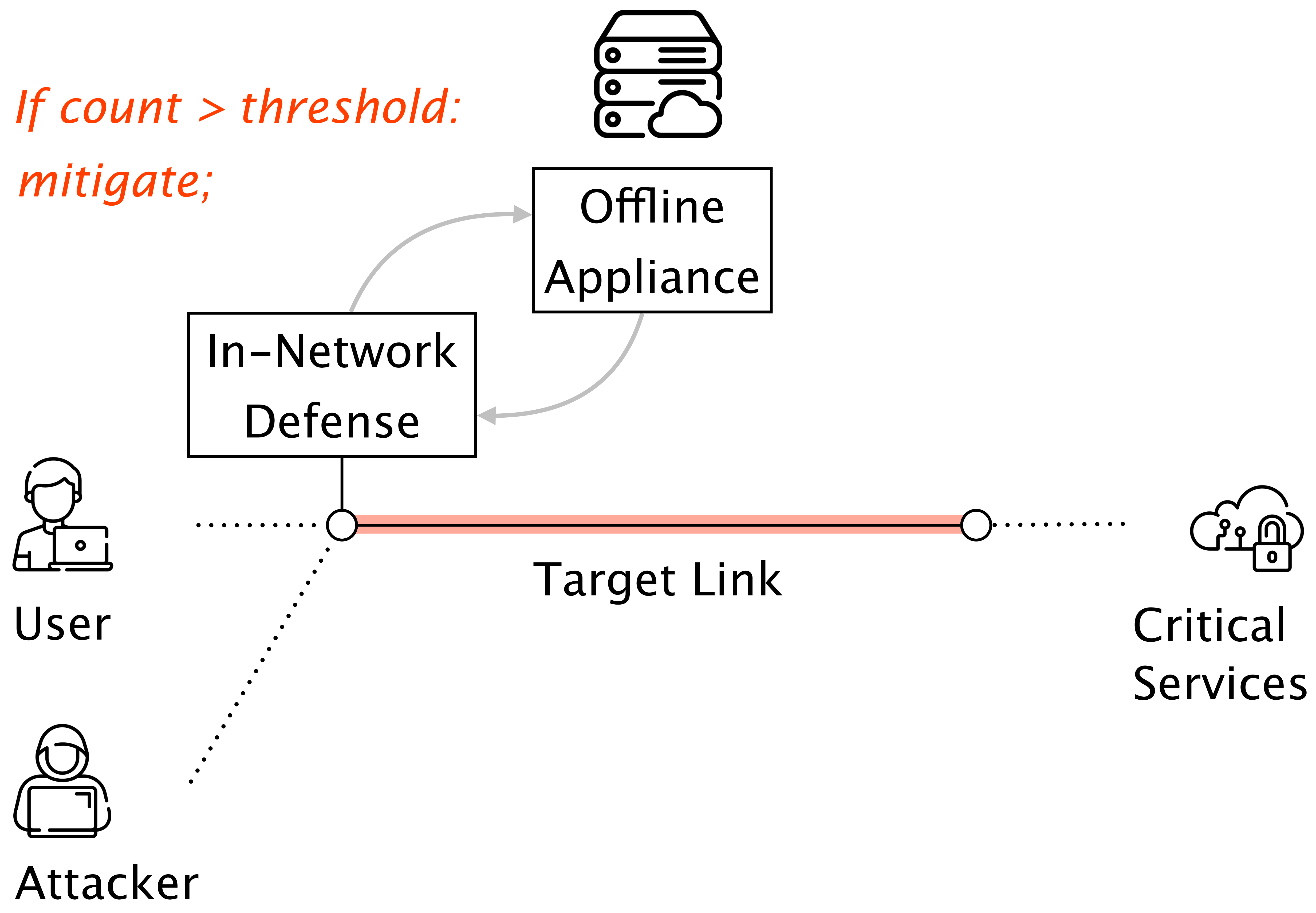
# Pulse-wave DDoS attacks exploit the limitations of existing defenses

Narrow attack coverage

Drastic mitigation

Slow reaction time

Risk of misconfiguration



# A pulse-wave DDoS defense needs to be …

| | |
|---|---|
| Narrow attack coverage | Generic detection |
| Drastic mitigation | Safe mitigation |
| Slow reaction time | Fast reaction |
| Risk of misconfiguration | Automated configuration |

# Can we build a pulse-wave DDoS defense … ?

Generic
detection

Safe
mitigation

Fast
reaction

Automated
configuration

# Can we build a pulse-wave DDoS defense … ?

| | *ACC-Turbo* 2022 |
| --- | --- |
| Generic detection | ✔ |
| Safe mitigation | ✔ |
| Fast reaction | ✔ |
| Automated configuration | ✔ |

| | ACC 2002 | *ACC-Turbo* 2022 |
| --- | --- | --- |
| Generic detection | ✔ | ✔ |
| Safe mitigation | ✔ | ✔ |
| Fast reaction | ✘ | ✔ |
| Automated configuration | ✘ | ✔ |

Generic

detection

Safe

mitigation

Fast

reaction

Automated

configuration

## Aggregate-based Congestion Control

2002

# Controlling High Bandwidth Aggregates in the Network

Ratul Mahajan, Steven M. Bellovin, Sally Floyd,
John Ioannidis, Vern Paxson, and Scott Shenker*

ICSI Center for Internet Research (ICIR) AT&T Labs Research
ratul@cs.washington.edu; smb,ji@research.att.com; floyd,vern,shenker@icir.org

## ABSTRACT

The current Internet infrastructure has very few built-in protection mechanisms, and is therefore vulnerable to attacks and failures. In particular, recent events have illustrated the Internet's vulnerability to both denial of service (DoS) attacks and flash crowds in which one or more links in the network (or servers at the edge of the network) become severely congested. In both DoS attacks and flash crowds the congestion is due neither to a single flow, nor to a general increase in traffic, but to a well-defined subset of the traffic – an *aggregate*. This paper proposes mechanisms for detecting and controlling such high bandwidth aggregates. Our design involves both a local mechanism for detecting and controlling an aggregate at a single router, and a cooperative *pushback* mechanism in which a router can ask upstream routers to control an aggregate. While certainly not a panacea, these mechanisms could provide some needed relief from flash crowds and flooding-style DoS attacks. The presentation in this paper is a first step towards a more rigorous evaluation of these mechanisms.

## 1. INTRODUCTION

In the current Internet, when a link is persistently overloaded all flows traversing that link experience significantly degraded service over an extended period of time. Persistent overloads can arise for several reasons. First, persistent overloads can result from a single flow not using end-to-end congestion control and continuing to transmit despite encountering a high packet drop rate. There is a substantial literature [6, 18, 27, 20] on mechanisms to cope with such *ill-behaved* flows (where, by *flow*, we mean a stream of packets sharing IP source and destination addresses, protocol field, and source and destination port numbers). Second, as was

of this are *denial of service* attacks (DoS) and *flash crowds*.

DoS attacks occur when a large amount of traffic from one or more hosts is directed at some resource of the network such as a link or a web server. This artificially high load denies or severely degrades service to legitimate users of that resource. The current Internet infrastructure has few protection mechanisms to deal with such DoS attacks, and is particularly vulnerable to distributed denial of service attacks (DDoS), in which the attacking traffic comes from a large number of disparate sites. A series of DDoS attacks occurred in February 2000 to considerable media attention, resulting in higher packet loss rates for several hours [12]. DDoS attacks have also been directed against network infrastructure rather than against individual web servers [21].

Flash crowds occur when a large number of users try to access the same server simultaneously. Apart from overloading the server itself, the traffic due to flash crowds can also overload the network links and thereby interfere with other, unrelated traffic. For example, degraded Internet performance was experienced during a Victoria's Secret webcast [2] and during the NASA Pathfinder mission. The "Slashdot effect" often leads to flash crowds.

While the intent and the triggering mechanisms for DoS attacks and flash crowds are quite different, from the network's perspective these two events are quite similar. The persistent congestion is neither due to a single well-defined flow, nor due to an *undifferentiated* overall increase in traffic. Instead, there is a particular *aggregate* of packets causing the overload, and these offending packets are usually spread across many flows.

Generic
detection

Safe
mitigation

Fast
reaction

Automated
configuration

## Aggregate-based Congestion Control

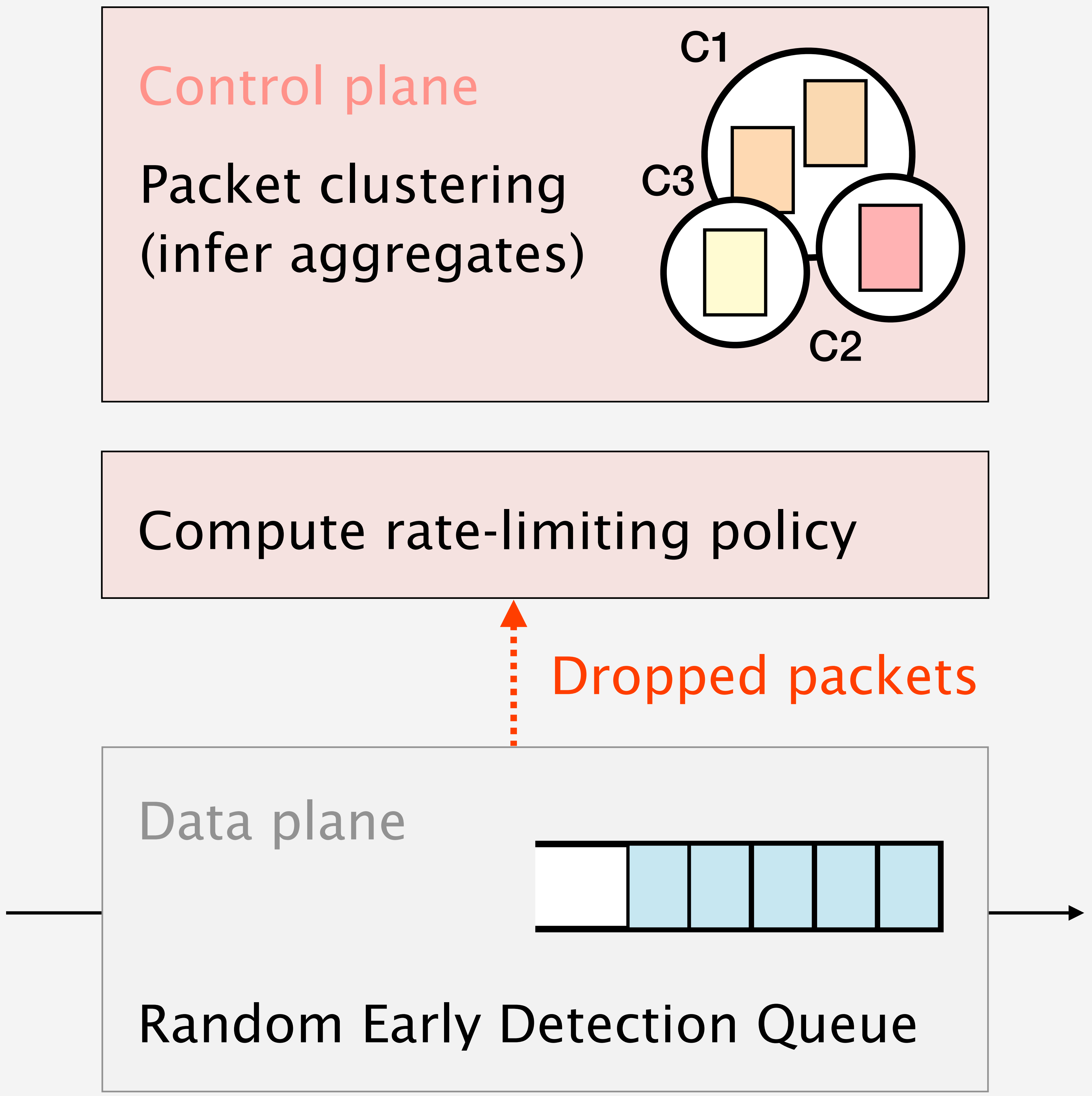
2002

Clustering ✔

Rate limiting ✔

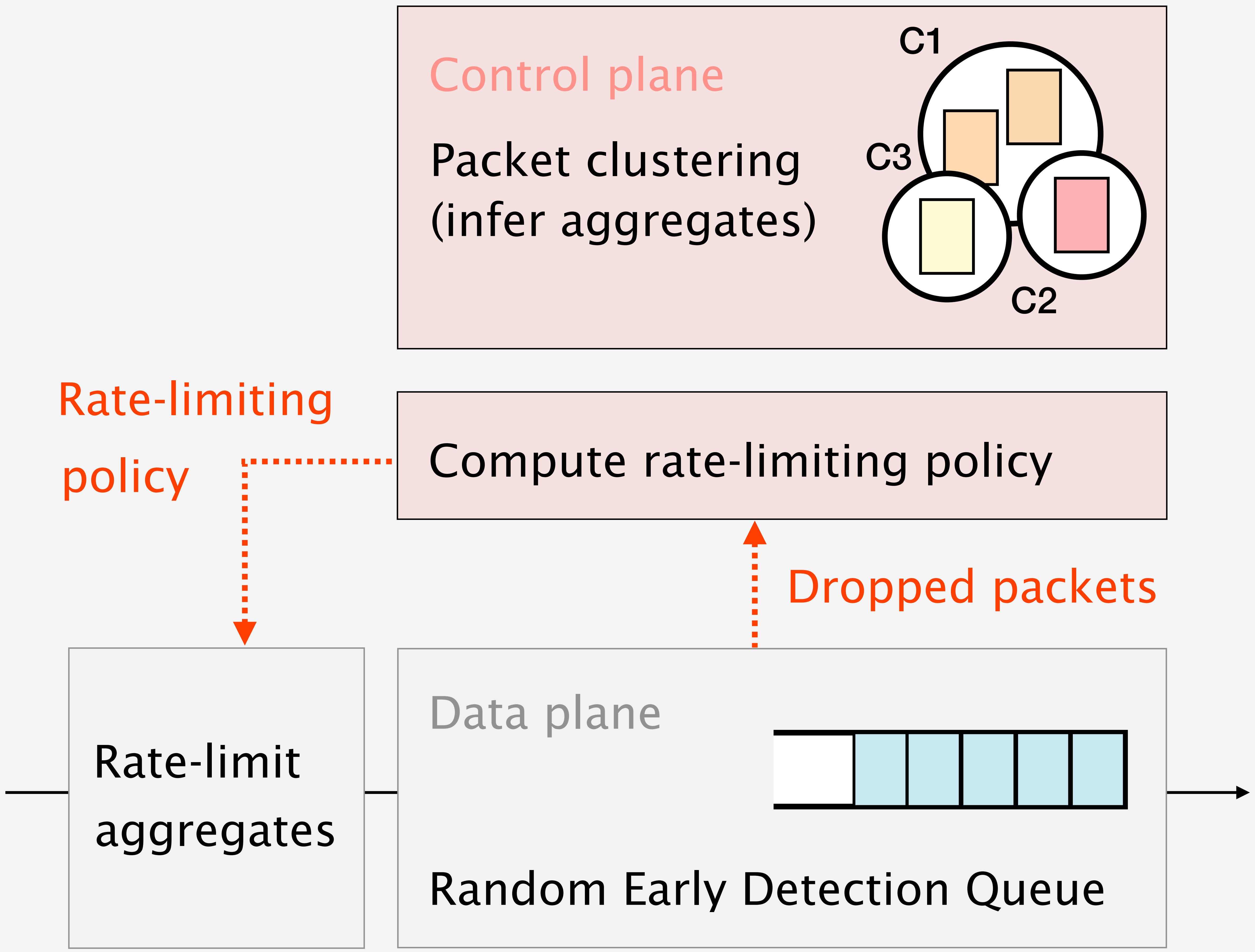
# Aggregate-based Congestion Control

2002



# Aggregate-based Congestion Control

2002



Generic
detection

Safe
mitigation

Fast
reaction

Automated
configuration

## Aggregate-based Congestion Control

2002

Clustering ✔

Rate limiting ✔

Offline inference ✘

Threshold-based ✘

| | ACC 2002 | | *ACC-Turbo* 2022 | |
|---|---|---|---|---|
| Generic detection | Clustering | ✔ | Clustering | ✔ |
| Safe mitigation | Rate limiting | ✔ | "Rate limiting" | ✔ |
| Fast reaction | Offline inference | ✗ | Online inference | ✔ |
| Automated configuration | Threshold-based | ✗ | Automated mitigation | ✔ |

| | ACC 2002 | | *ACC-Turbo* 2022 | |
|---|---|---|---|---|
| Generic detection | Clustering | ✔ | Clustering | ✔ |
| Safe mitigation | Rate limiting | ✔ | “Rate limiting” | ✔ |
| Fast reaction | Offline inference | ✗ | Online inference | ✔ |
| Automated configuration | Threshold-based | ✗ | Automated mitigation | ✔ |

# How to infer traffic aggregates online, in the data plane?

Hardware limitations

Packets processed only once

Restricted computations

Limited state access

…

# How to infer traffic aggregates online, in the data plane?

Hardware limitations

Packets processed only once

Restricted computations

Limited state access

…

Online clustering

Endless stream of data

Each point processed only once

Irrevocable action at each point’s arrival

# How to infer traffic aggregates online, in the data plane?

Hardware limitations

Packets processed only once

Restricted computations

Limited state access

…

Online (**packet**) clustering

For each arriving packet:

Map it to closest cluster

*or*

Merge two clusters and
create new cluster for the packet

# How to infer traffic aggregates online, in the data plane?

How to represent packets?

How to represent clusters?

What distance to use?

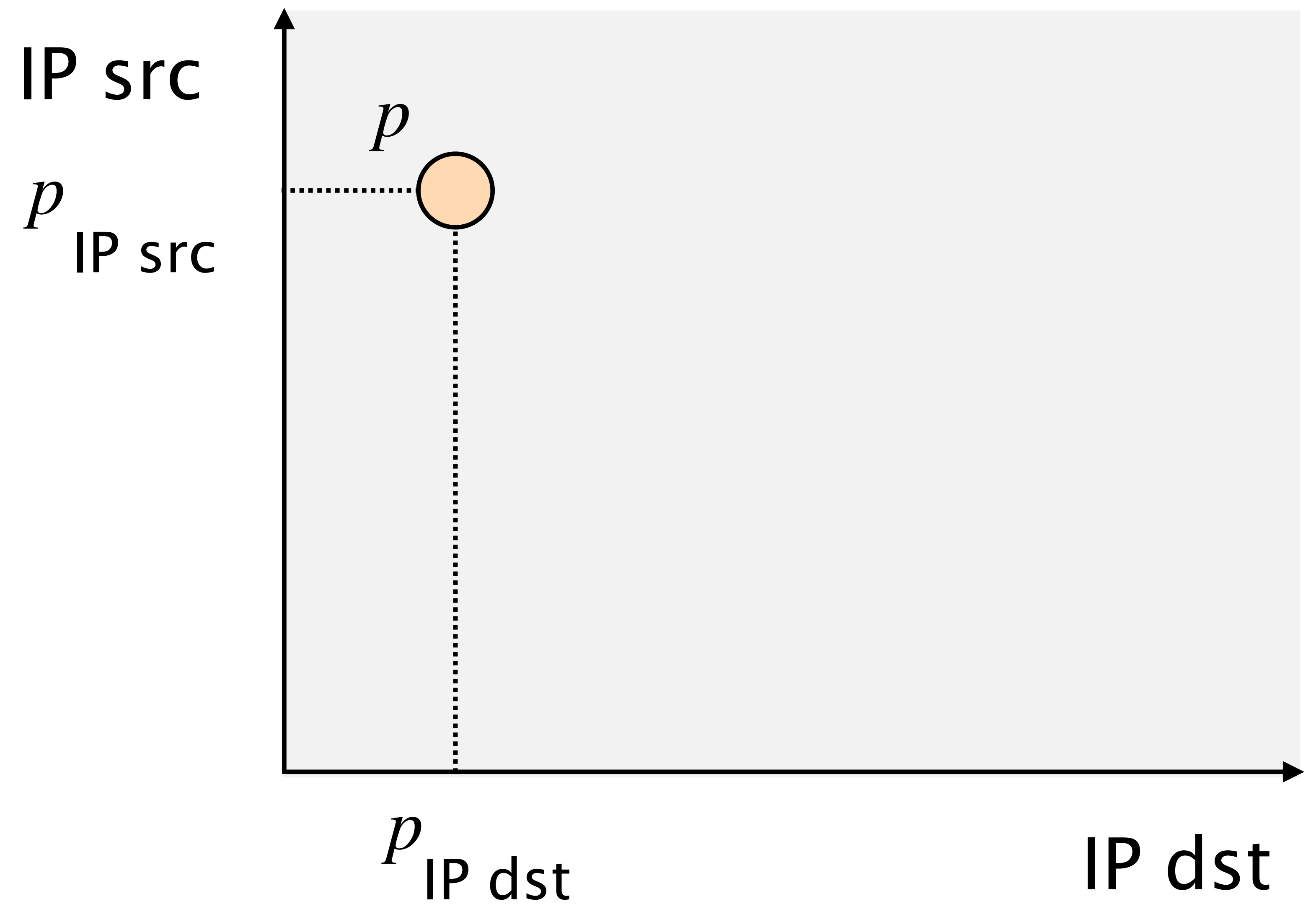
# How to infer traffic aggregates online, in the data plane?

How to represent packets?

Points in the header space

How to represent clusters?

What distance to use?



# How to infer traffic aggregates online, in the data plane?

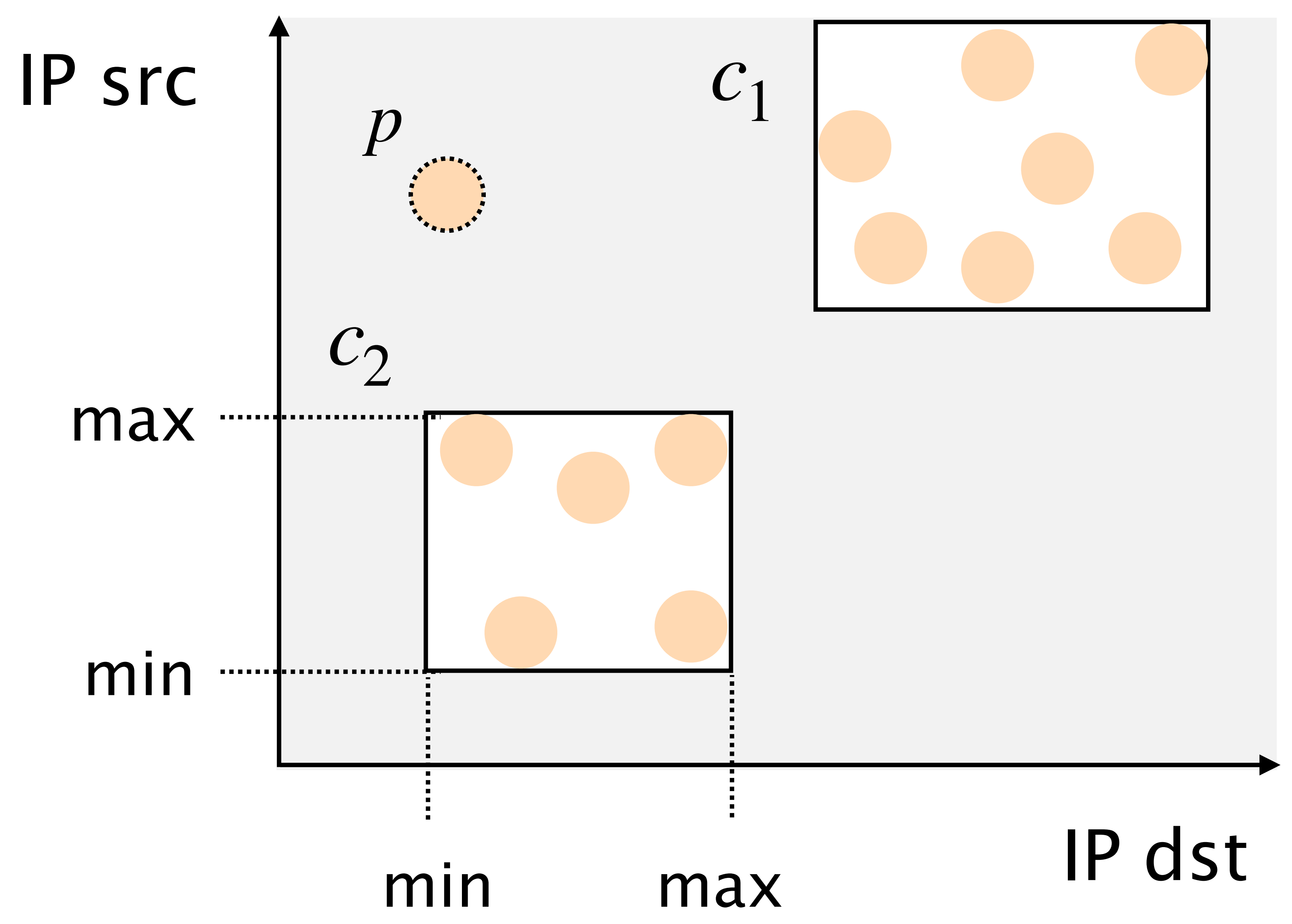
### How to represent packets?

Points in the header space

### How to represent clusters?

Ranges (registers), sets (bloom filters)

### What distance to use?



# How to infer traffic aggregates online, in the data plane?

### How to represent packets?

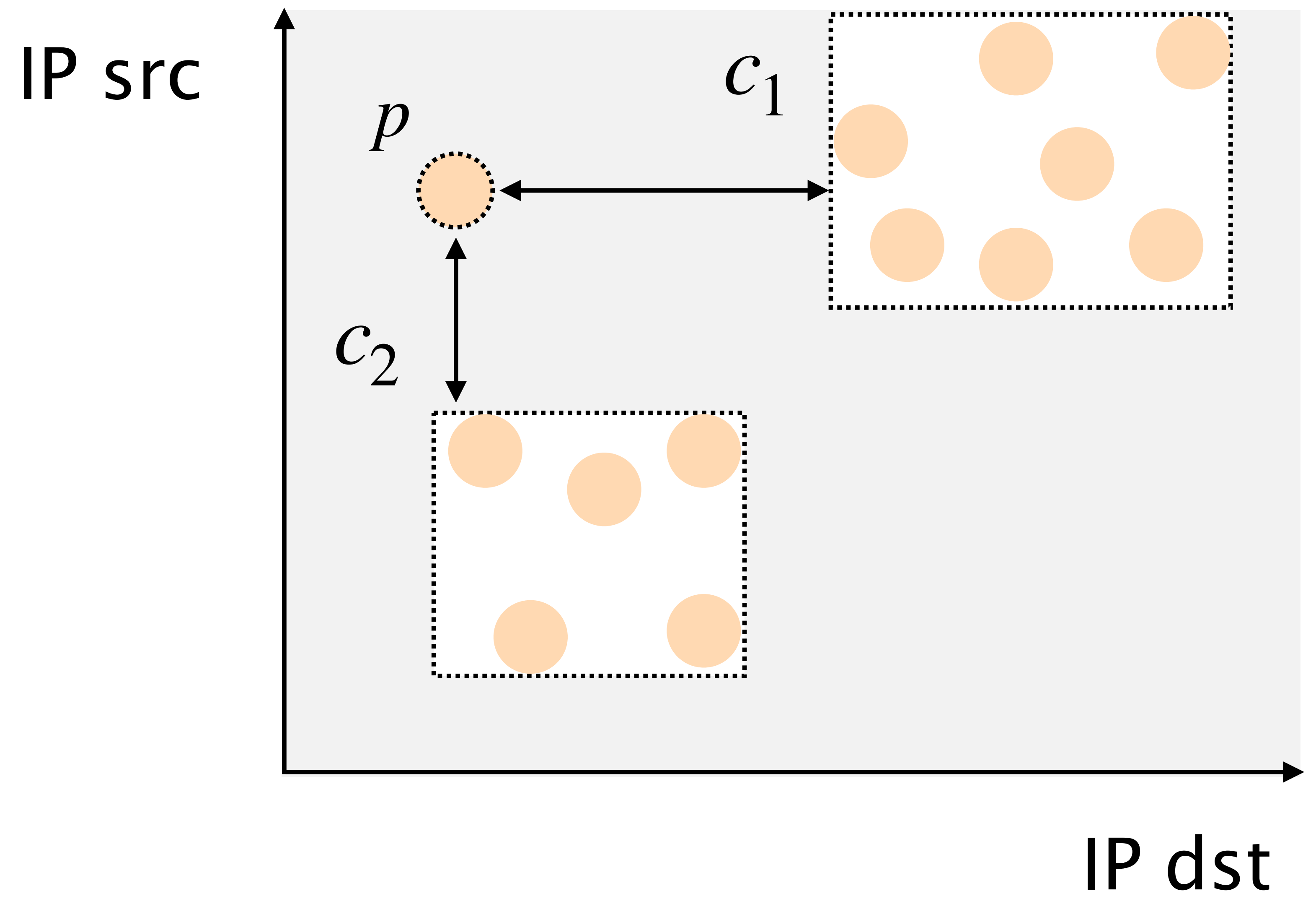
Points in the header space

### How to represent clusters?

Ranges (registers), sets (bloom filters)

### What distance to use?

Manhattan distance



| | ACC 2002 | *ACC-Turbo* 2022 |
|---|---|---|
| Generic detection | Clustering ✔ | Clustering ✔ |
| Safe mitigation | Rate limiting ✔ | “Rate limiting” ✔ |
| Fast reaction | Offline inference ✗ | Online inference ✔ |
| Automated configuration | Threshold-based ✗ | Automated mitigation ✔ |

# How to automatically mitigate inferred attacks?

## How to identify malicious clusters?

We can have false positives (e.g., flash crowd)

## How to mitigate them?

Filtering traffic is detrimental under misclassification

## When to activate the mitigation?

Threshold-based is vulnerable to pulse-wave

# How to automatically mitigate inferred attacks?

Programmable scheduling

*ACC-Turbo* deprioritizes malicious clusters

Extract cluster statistics

From the data plane, e.g. rate and size

Assess clusters' maliciousness

Narrow clusters with a higher rate, more malicious

Synthesize scheduling policy

Deprioritize most-malicious clusters

# How to automatically mitigate inferred attacks?

Programmable scheduling

*ACC-Turbo* deprioritizes malicious clusters

… leverages the whole uncertainty spectrum

with fine-grained scheduling policies

… is safe

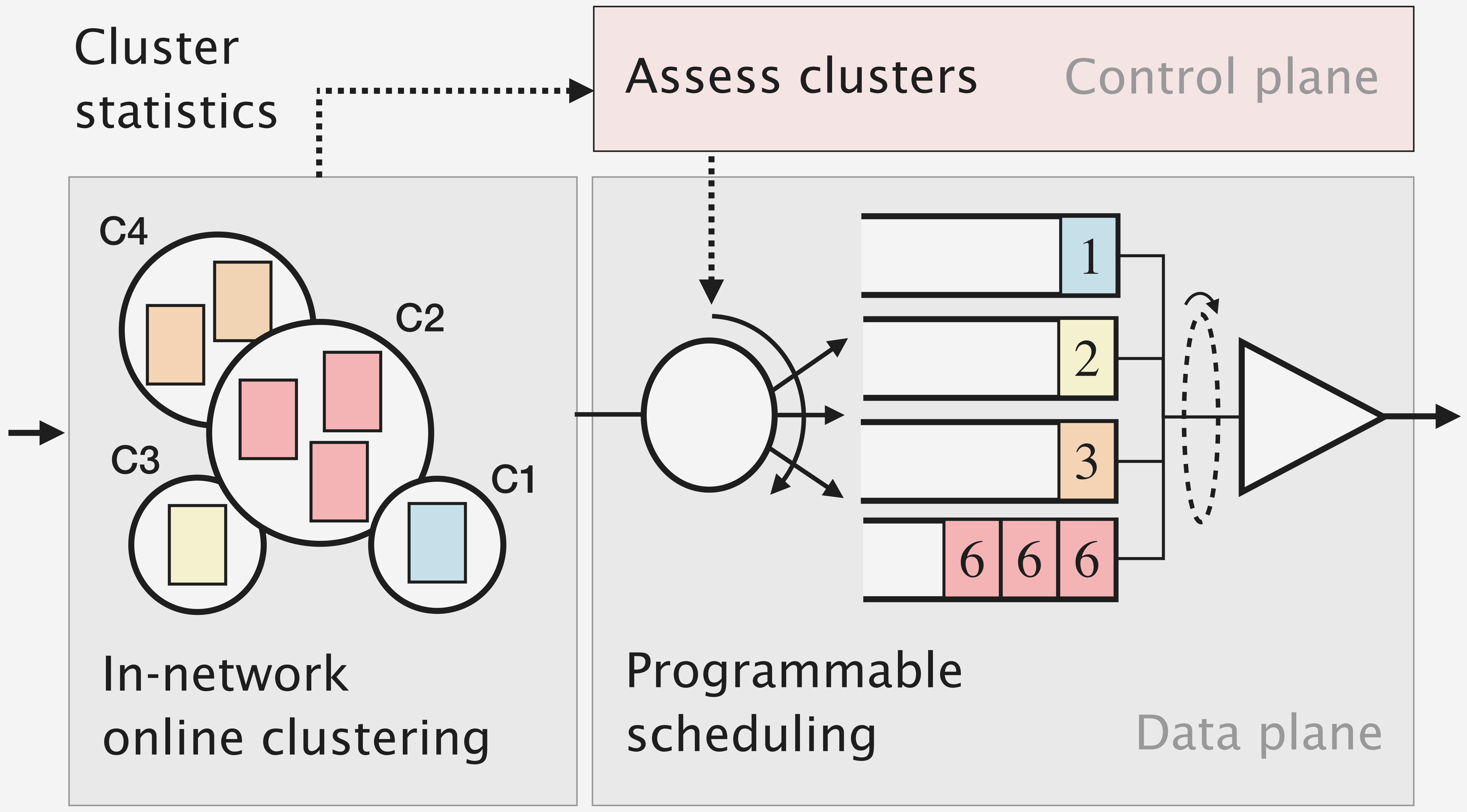
only drops under congestion

… does not require activation

can be always-on

# *ACC-Turbo*

2022



| | ACC<br>2002 | *ACC-Turbo*<br>2022 |
|---|---|---|
| **Generic** detection | ✔ | ✔ |
| **Safe** mitigation | ✔ | ✔ |
| **Fast** reaction | ✘ | ✔ |
| **Automated** configuration | ✘ | ✔ |

# We evaluated *ACC-Turbo* on hardware and simulations

Hardware evaluation (Tofino)

Pulse-wave DDoS mitigation

Comparison with state-of-the-art

Software evaluation (NetBench)

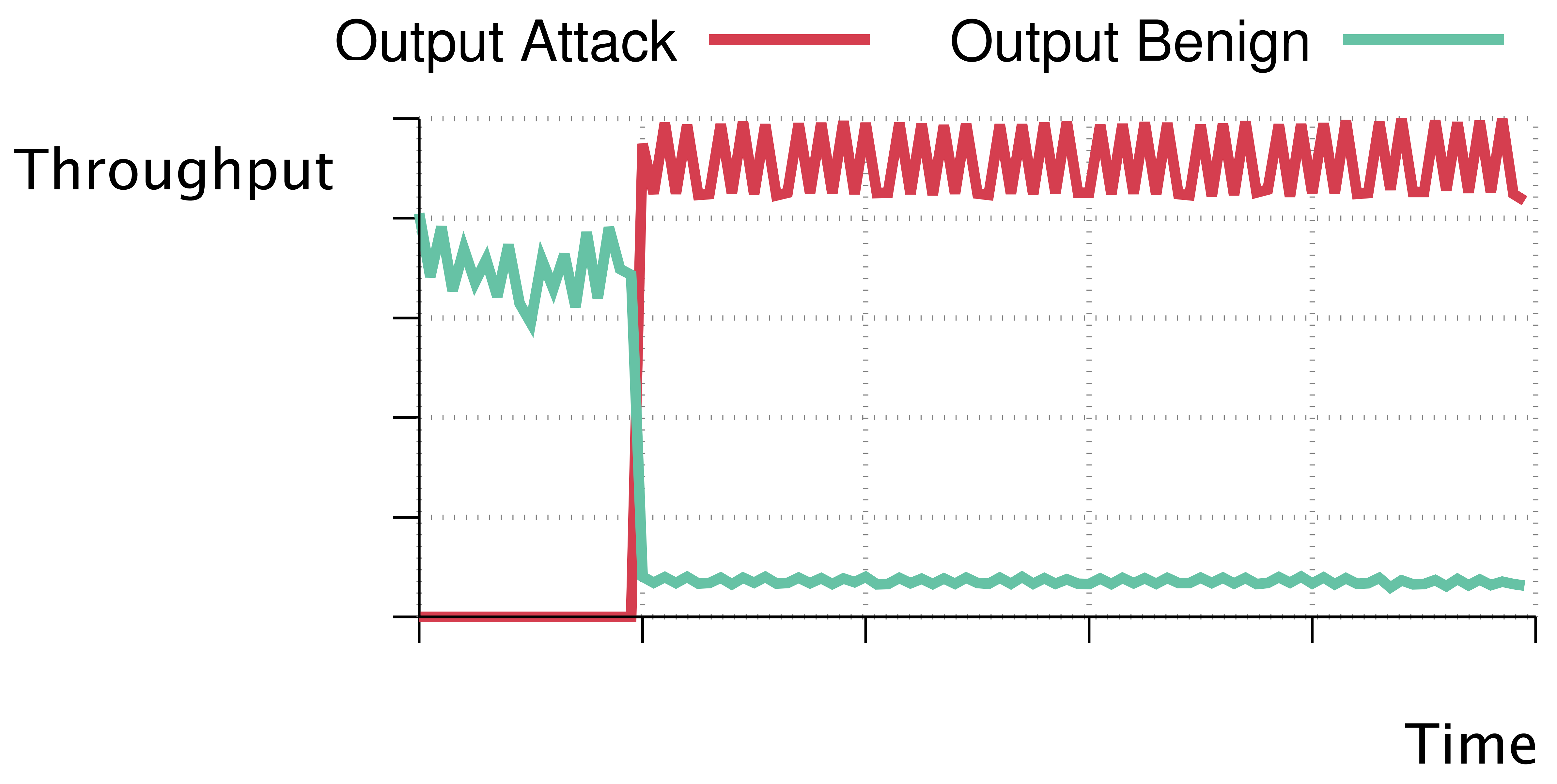
Impact of design decisions

Performance of more-complete versions

github.com/nsg-ethz/ACC-Turbo

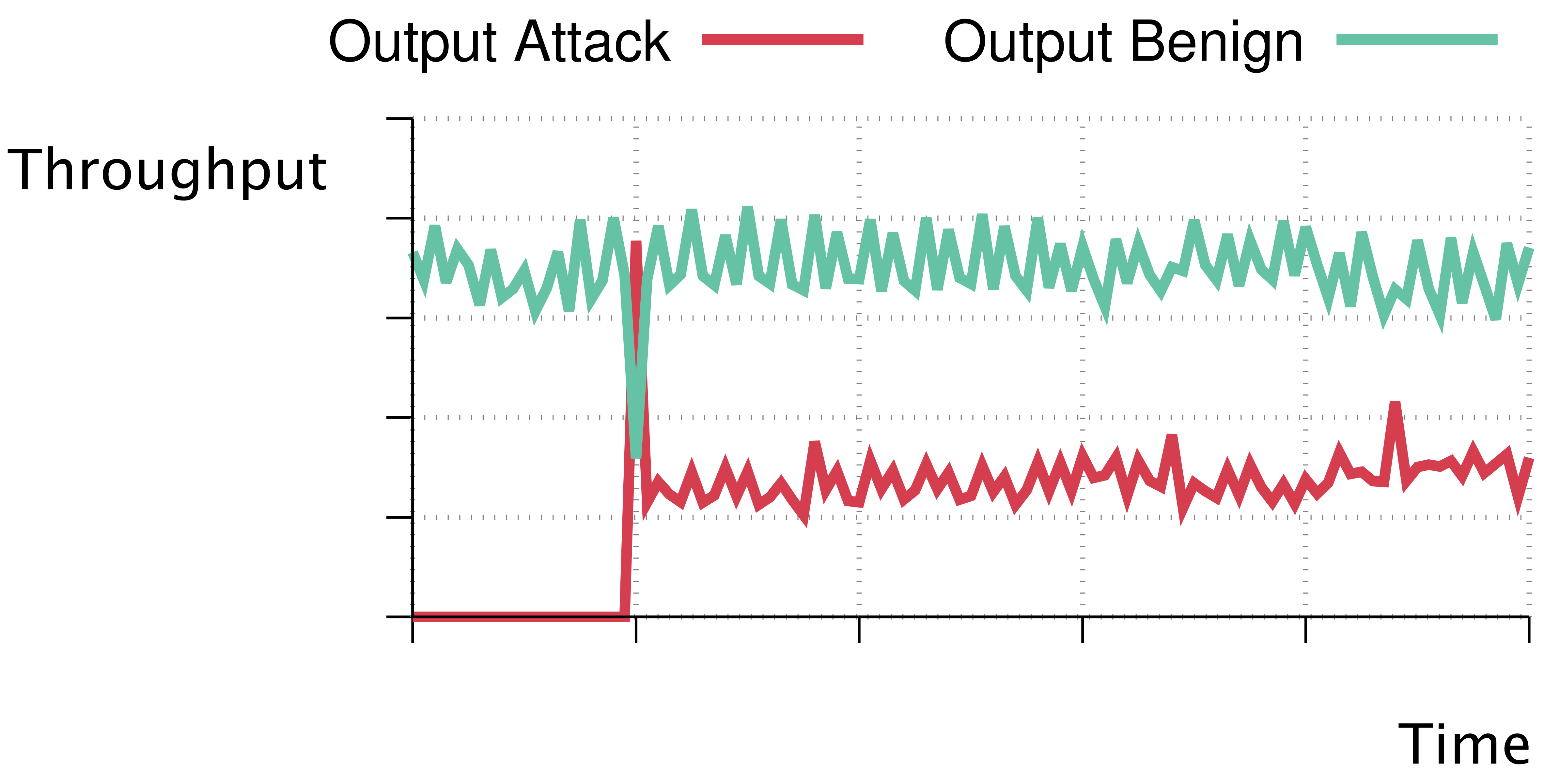
*ACC-Turbo* outperforms existing defenses and mitigates pulse-wave DDoS attacks

No defense

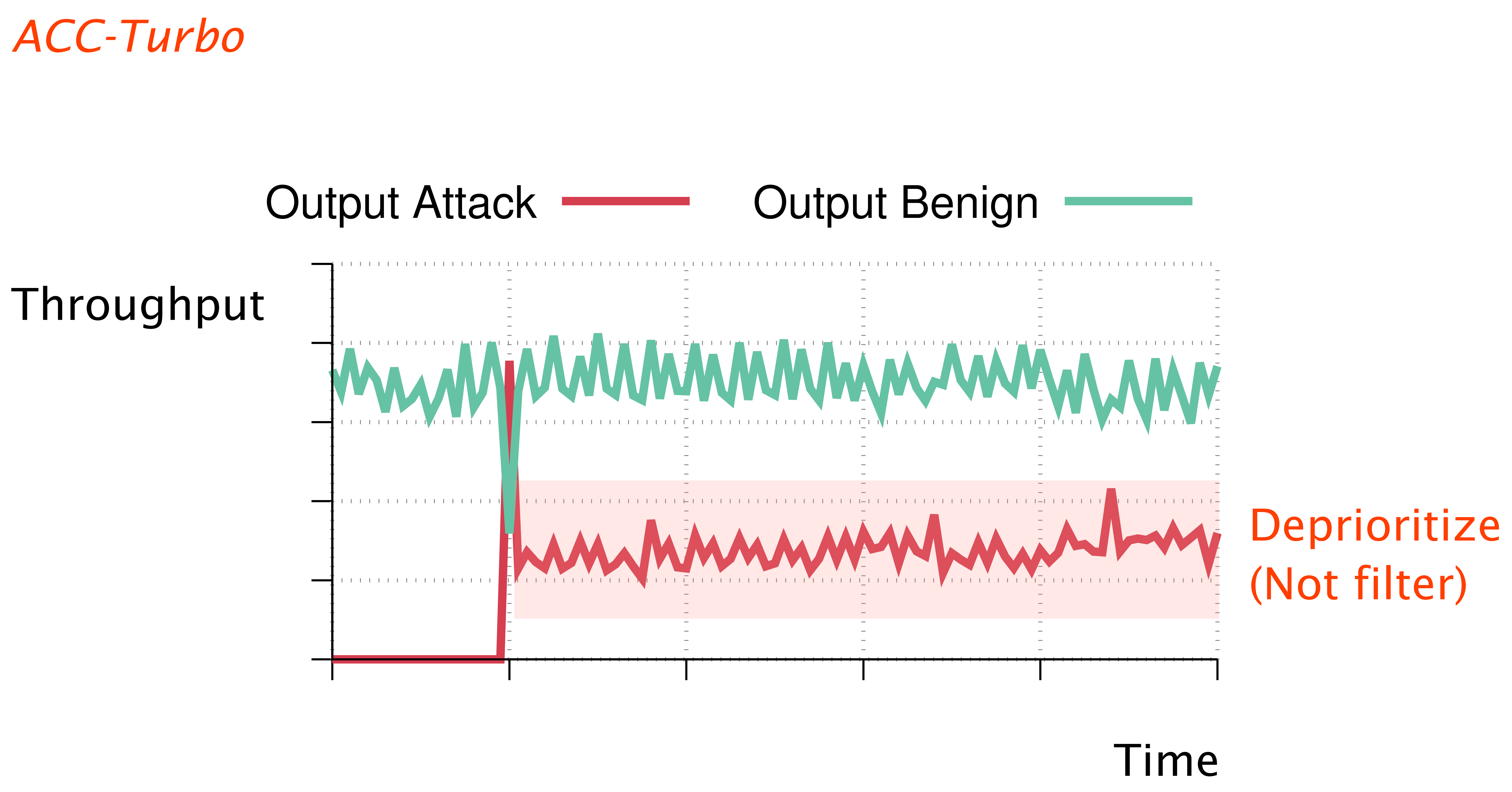


# *ACC-Turbo* outperforms existing defenses and mitigates pulse-wave DDoS attacks

*ACC-Turbo*

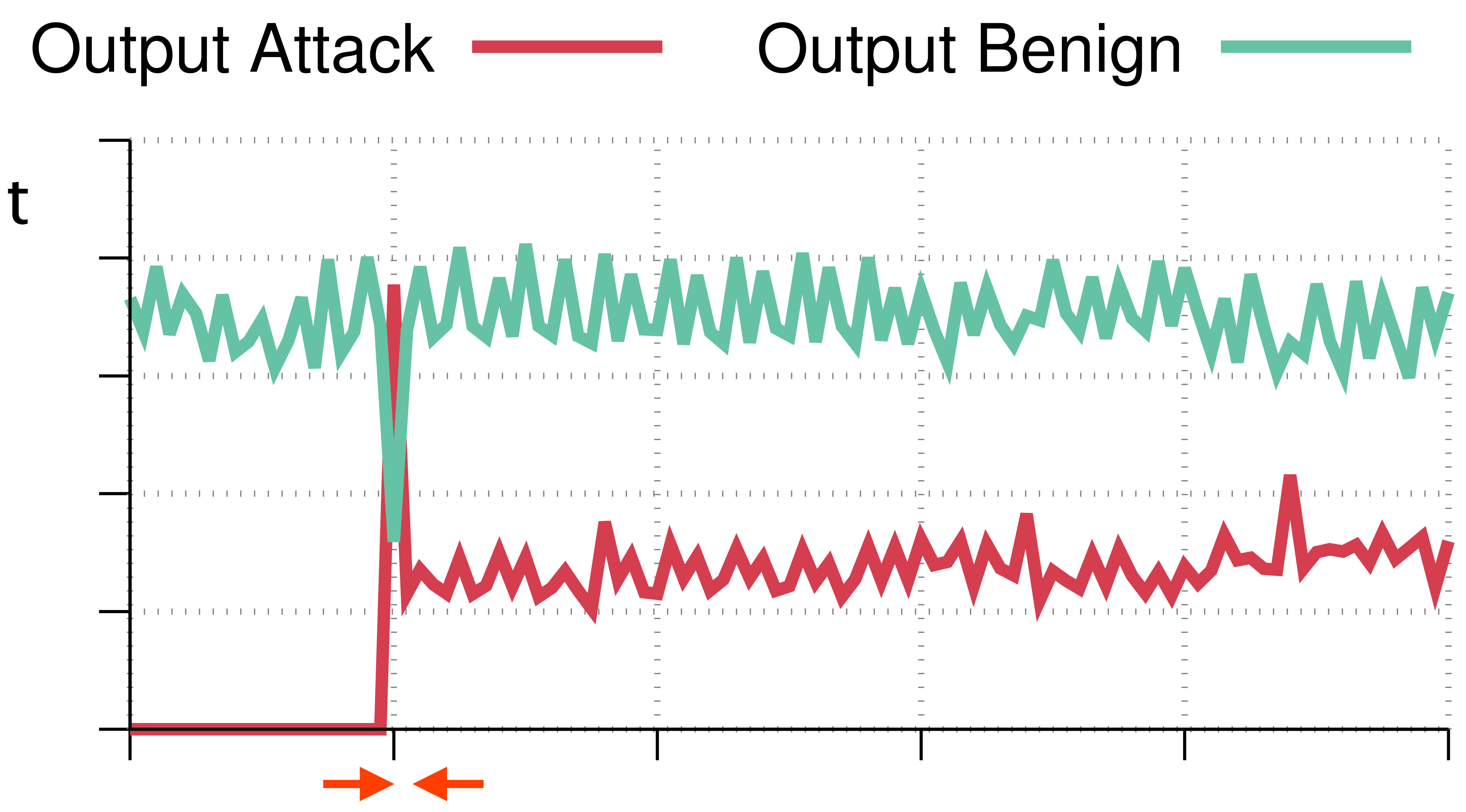


# *ACC-Turbo* outperforms existing defenses and mitigates pulse-wave DDoS attacks



# *ACC-Turbo* outperforms existing defenses and mitigates pulse-wave DDoS attacks

*ACC-Turbo*




< 1s reaction time

10x faster than state of the art

# Aggregate-Based Congestion Control for Pulse-Wave DDoS Defense

Pulse-wave DDoS attacks target existing defenses

by exploiting their limitations

*ACC-Turbo* mitigates pulse-wave DDoS attacks

at line rate, on programmable switches

*ACC-Turbo* combines online clustering
and programmable scheduling

github.com/nsg-ethz/ACC-Turbo