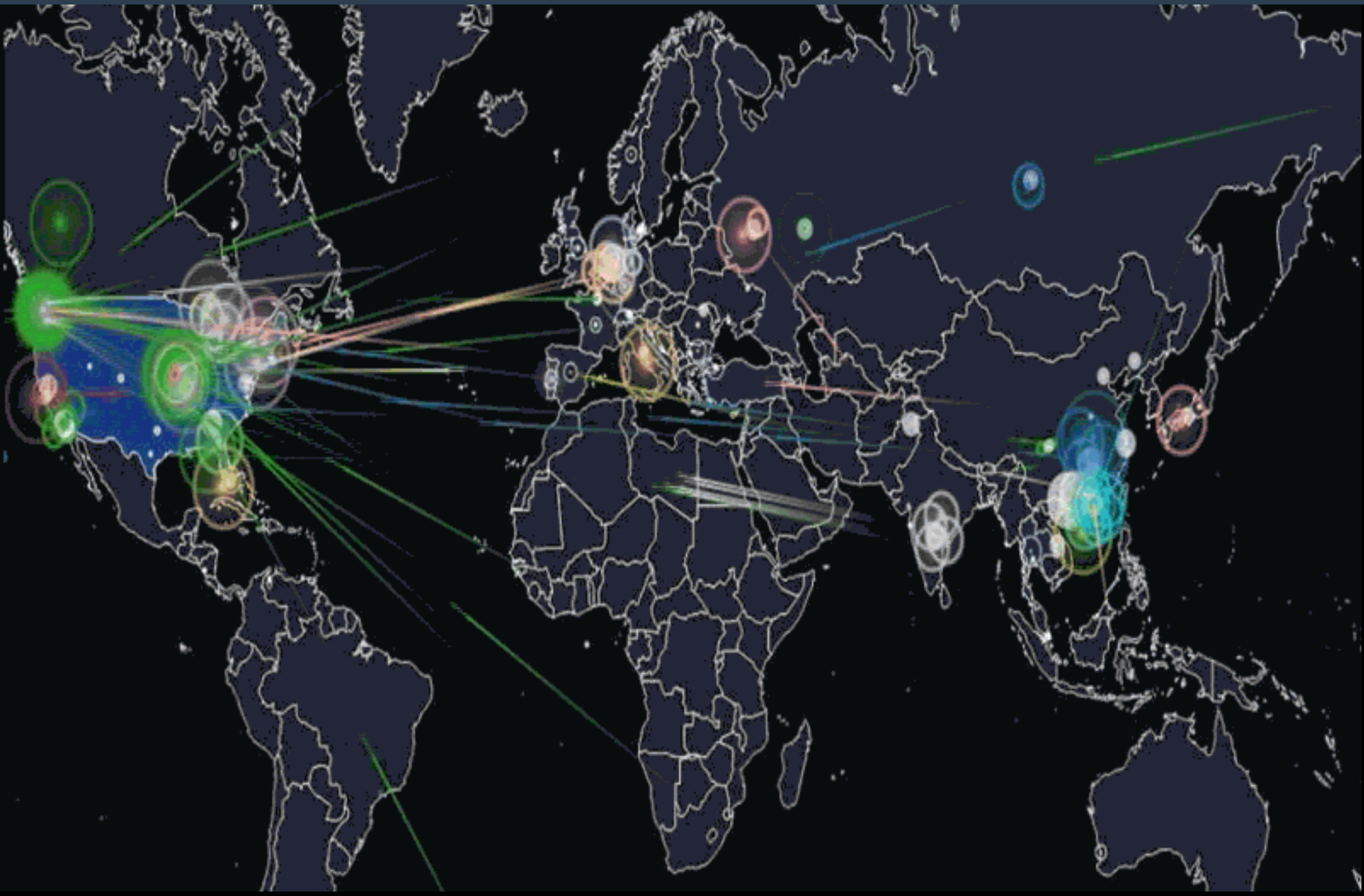
# Measuring the efficiency of SDN mitigations against cyber attacks

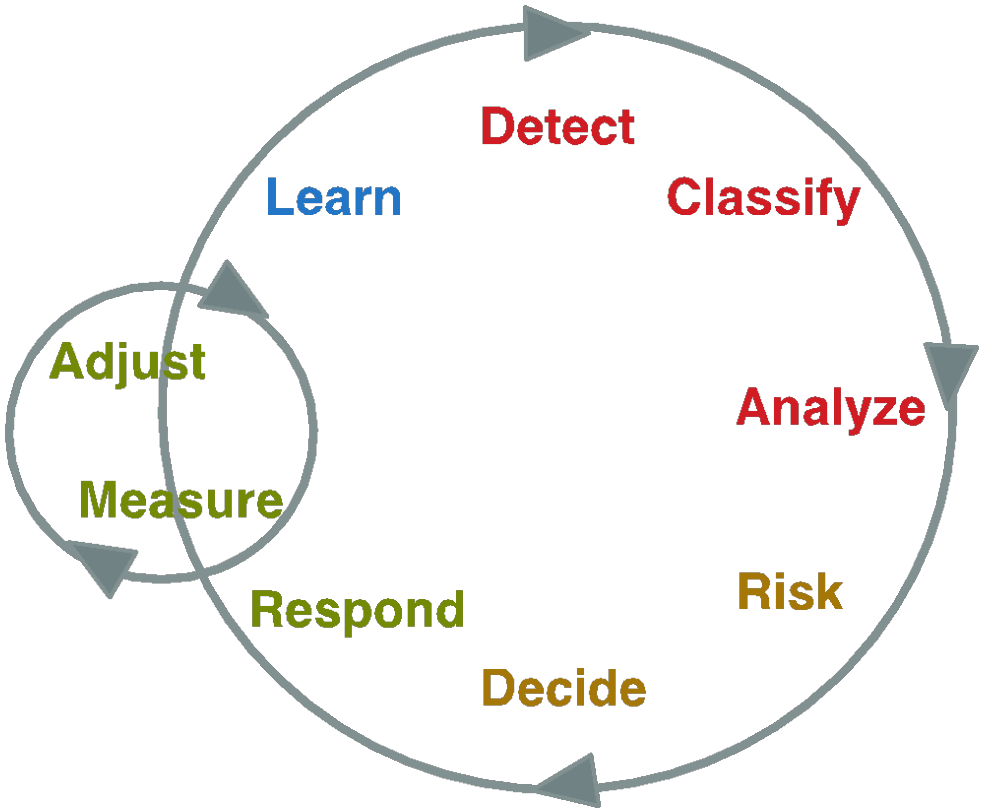
Ralph Koning
Ben de Graaff, Robert Meijer, Cees de Laat, Paola Grosso

System and Network Engineering research group
Universiteit van Amsterdam

# Context



# SARNET Control loop




**Detection phase:** Detect, Classify, Analyze

**Decision phase:** Risk, Decide

**Response phase:** Respond, Adjust, Measure

**Learn phase:** Learn (with input from other phases)

# Problem statement

**Detect** → ***Decide*** → **Respond**

There are multiple ways of responding to a single attack.

To select the right countermeasure the response phase executes:

- **Risk analysis** (we will omit this since we cannot quantify risk)
- **Response** selection

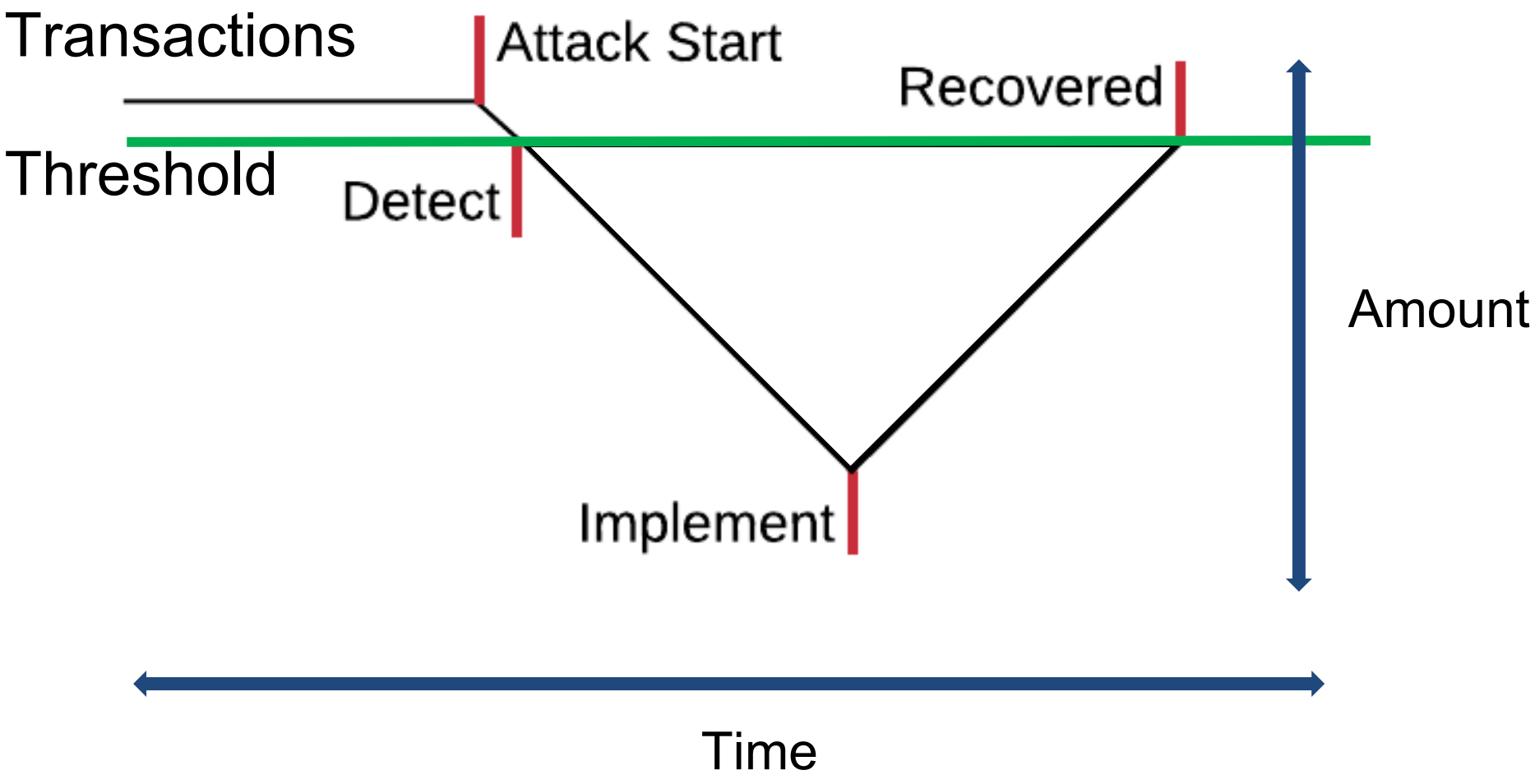
**How does one pick the most effective countermeasure to defend against an attack?**

# Response selection

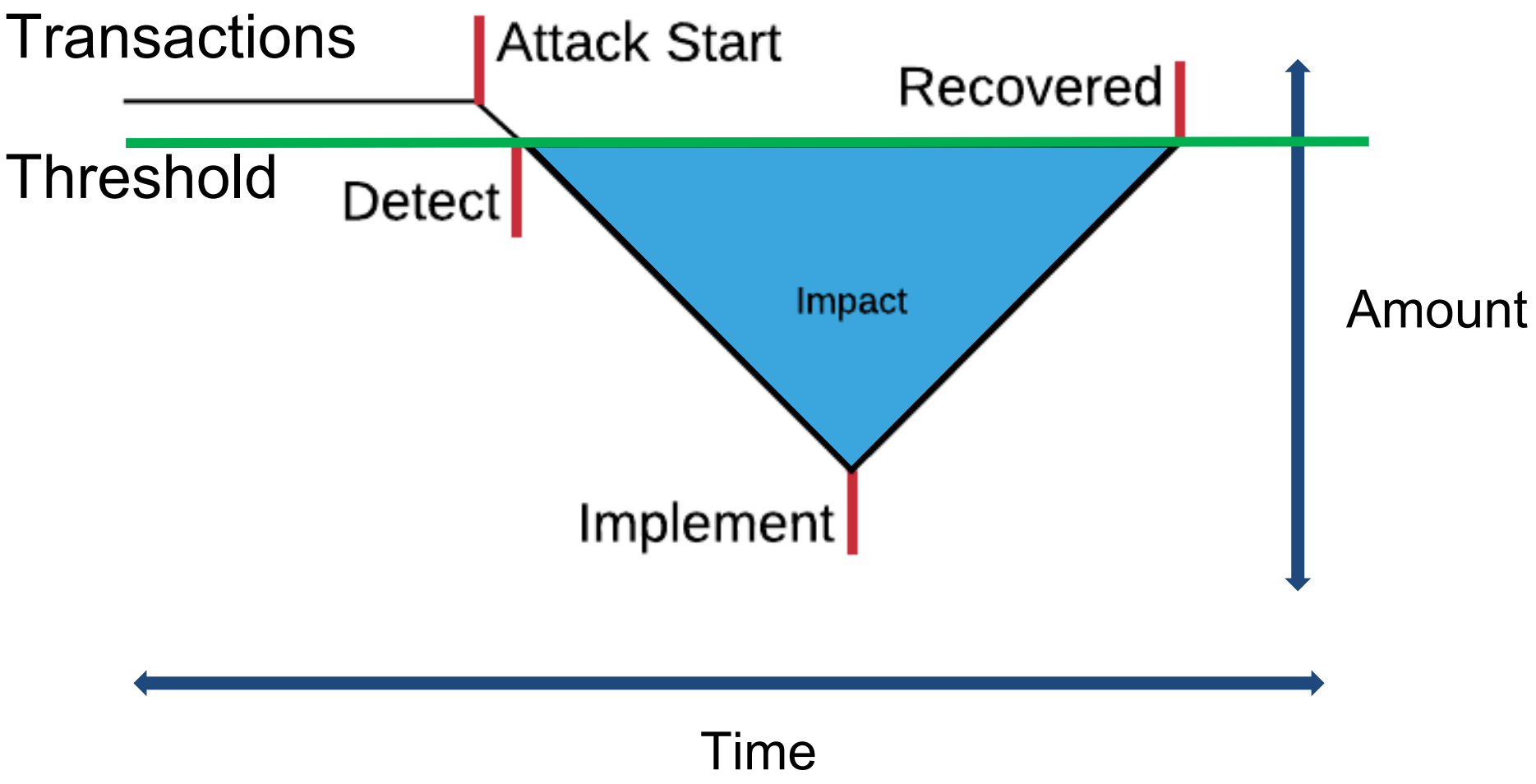
We can *rank* responses based on past behaviour. Therefore, we need a measure of the effect of the response.

- Success rate; did we recover?
- Response time; how fast did we recover?
- Impact; How much was lost?

# Detection points and thresholds



# Impact


Transactions
Attack Start
Recovered
Threshold
Detect
Impact
Amount
Implement
Time

# Efficiency

| | | | |
|---|---|---|---|
| Impact | $I$ | Importance | $\alpha$ |
| Timeout | $T$ | Not recovered boundary | $\beta$ |
| Threshold (Baseline) | $B$ | Cost integral | $Ct$ |
| | | Cost under normal circumstances | $C$ |

$$E(\text{recovered or not}, I, Ct) \triangleq \begin{cases} \beta + \alpha\frac{B\cdot T - I}{B\cdot T} + (1-\beta-\alpha)\frac{C\cdot T - Ct}{C\cdot T} & \\ = 1 - \frac{\alpha}{B\cdot T}I - \frac{1-\beta-\alpha}{C\cdot T}Ct & \text{Recovered,} \\ \alpha(\frac{\beta}{1-\beta})\frac{B\cdot T - I}{B\cdot T} + (1-\beta-\alpha)(\frac{\beta}{1-\beta})\frac{C\cdot T - Ct}{C\cdot T} & \\ = \beta - \alpha\frac{\beta}{(1-\beta)(B\cdot T)}I - (1-\beta-\alpha)\frac{\beta}{(1-\beta)(C\cdot T)}Ct & \text{otherwise,} \end{cases}$$

**Because we have no cost measurements we will ignore cost for now.**
$\beta = 0, \quad \alpha \approx 1$

**We are only interested in the 'Recovered' situation**

# Efficiency

Impact $I$
Timeout $T$
Threshold (baseline) $B$

Success rate $S$
Weight (Importance of metric) $\gamma$

Efficiency single metric:

$$E_m(recovered, I) \triangleq 1 - \frac{I}{B * T}$$

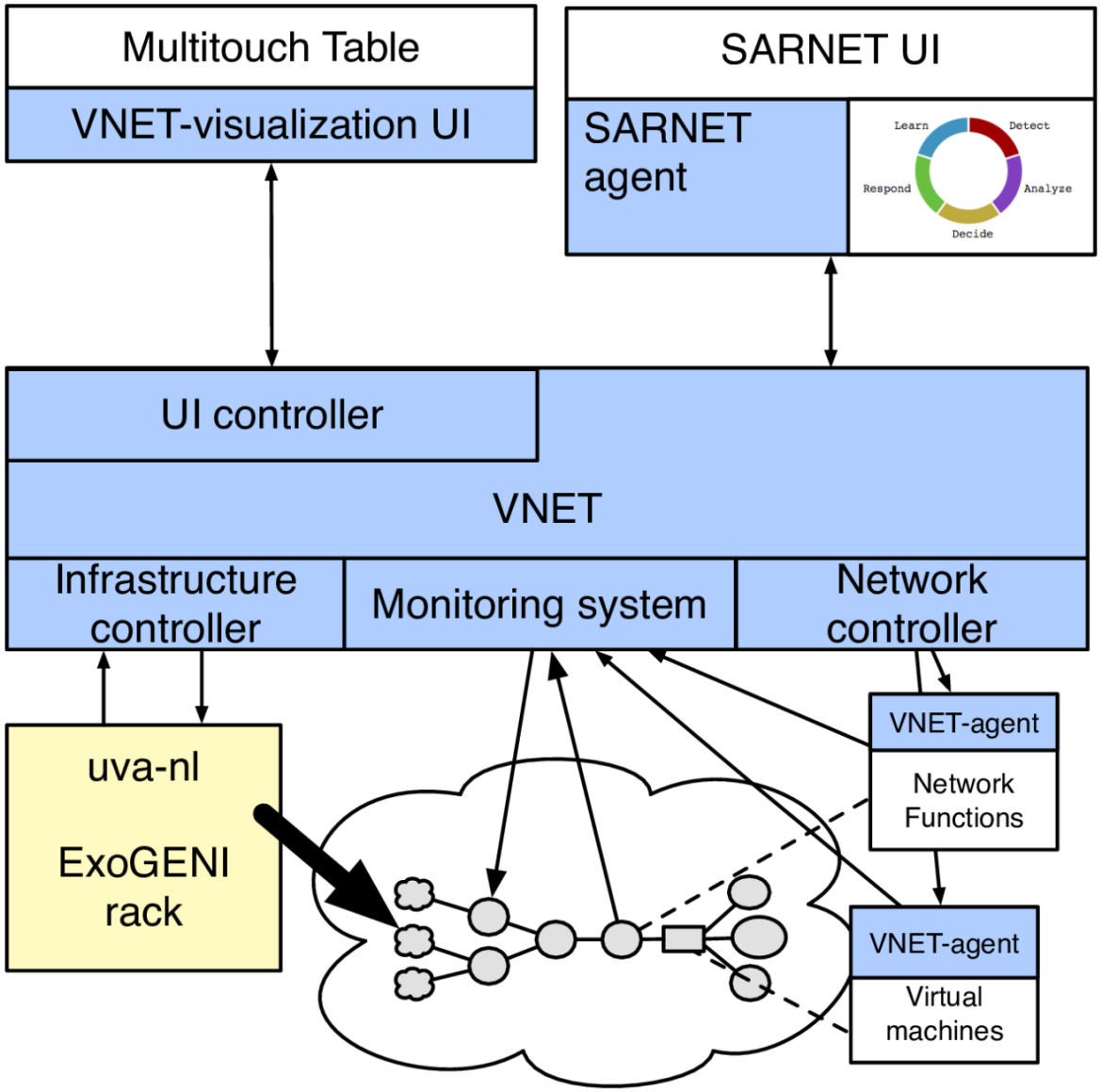
Efficiency defence:

$$E_d \triangleq \sum_{i=1}^{n} \gamma_i E_{m,i}$$

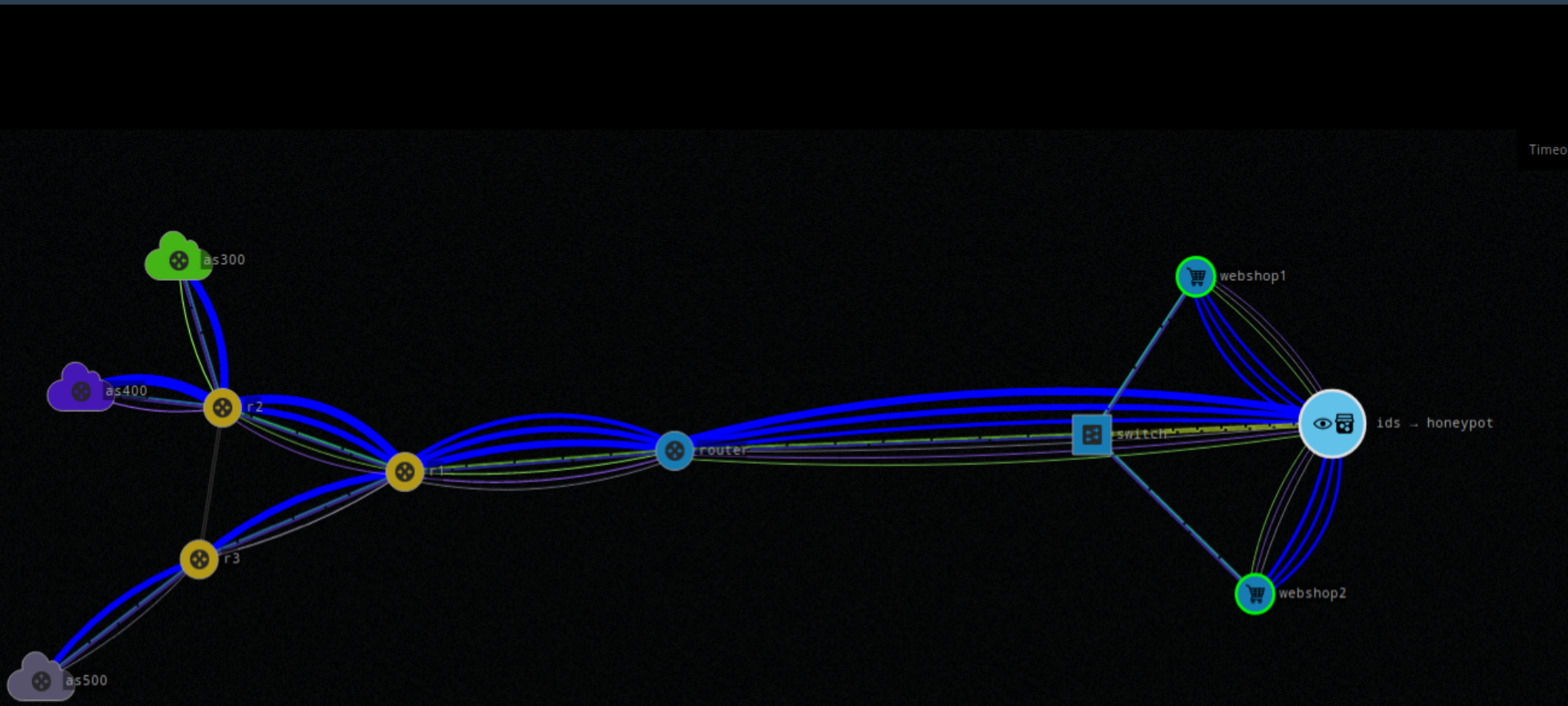
Efficiency countermeasure:

$$E_c \triangleq S_c * E_d$$

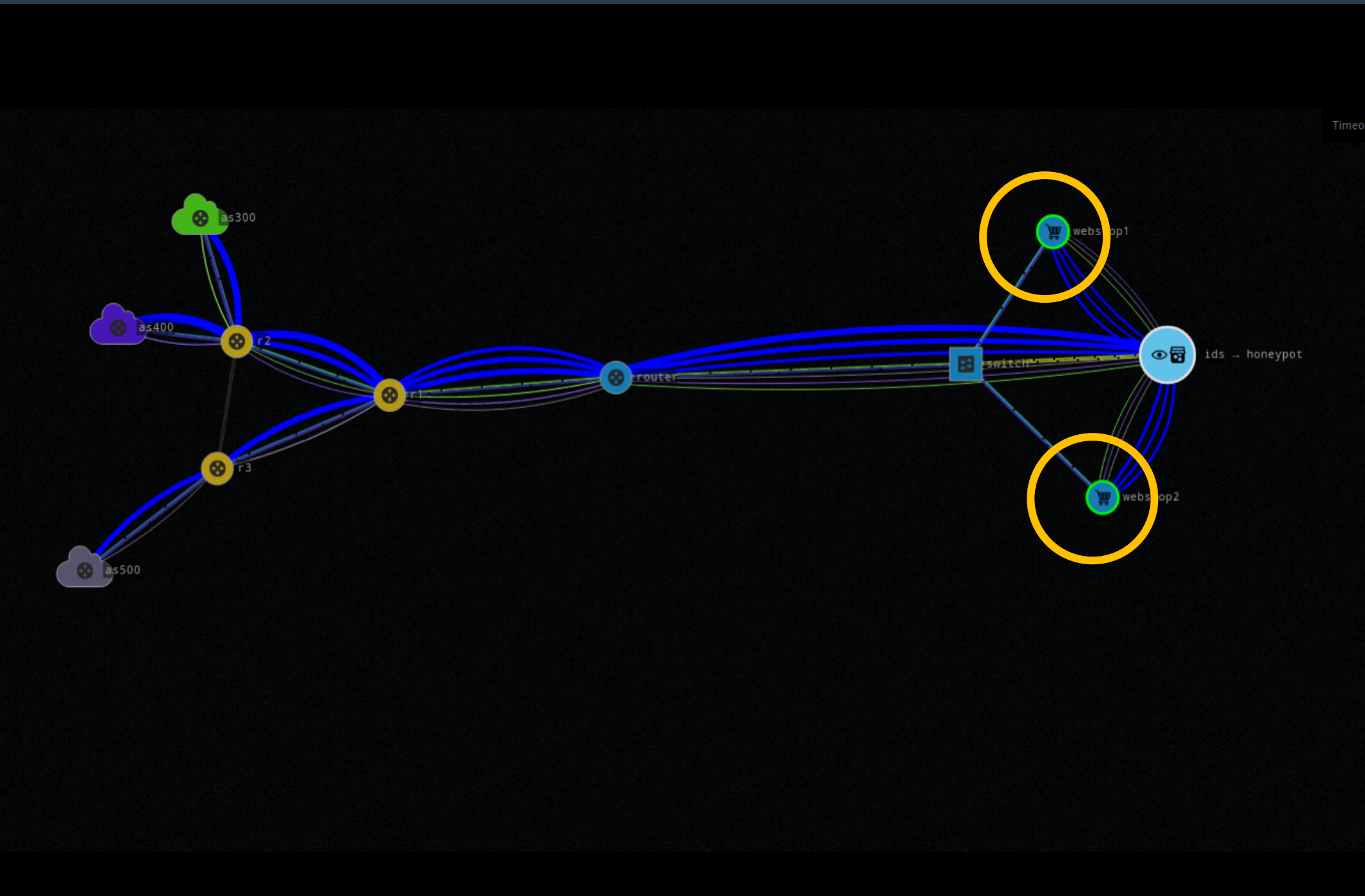
# Environment



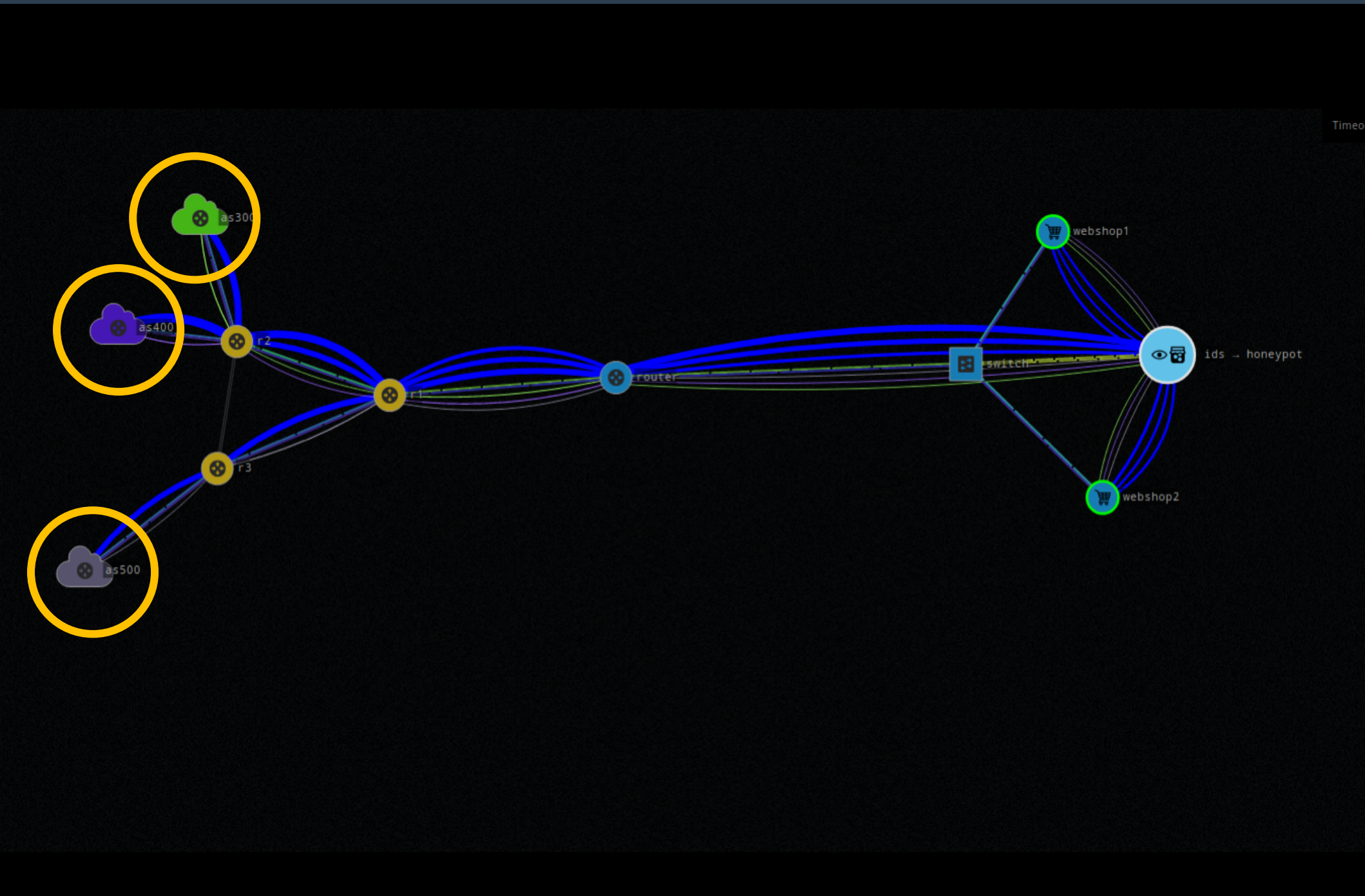
# Scenario


as300
as400
r2
r1
router
switch
webshop1
ids → honeypot
r3
webshop2
as500

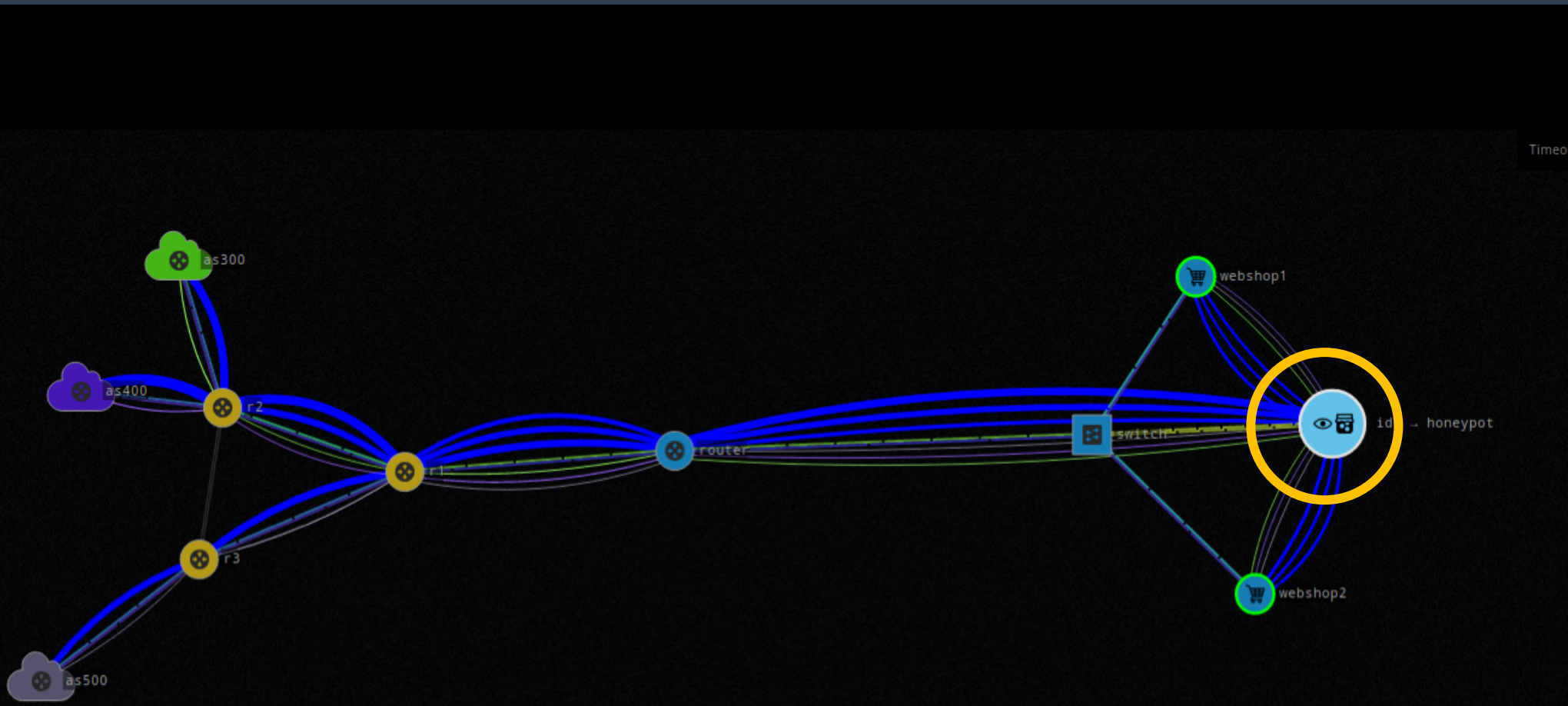
# Scenario



# Scenario


as300
as400
as500
r1
r2
r3
router
switch
webshop1
webshop2
ids → honeypot

# Scenario


as300
as400
r2
r1
router
switch
webshop1
id → honeypot
r3
webshop2
as500

# Observables

## Metrics

*Sales*: the combined amount of transactions to web services
*CPU*: combined CPU load on the web services
*logfail*: The amount of failed logins

## Attack classifications

| | | |
|---|---|---|
| ***DDoS_Attack*** | UDP based DDoS attack causes link congestion | if not *Sales* > 210 |
| ***pwd_bf_attack*** | Password brute force attack causes abnormal login failures | if not *logfail*< 20 |
| ***CPU_Attack*** | DoS attack focused on consuming CPU resources. | if not *CPU* < 85 and not *Sales* < 210 |

# Experiments

Each measurement is repeated 50 times.

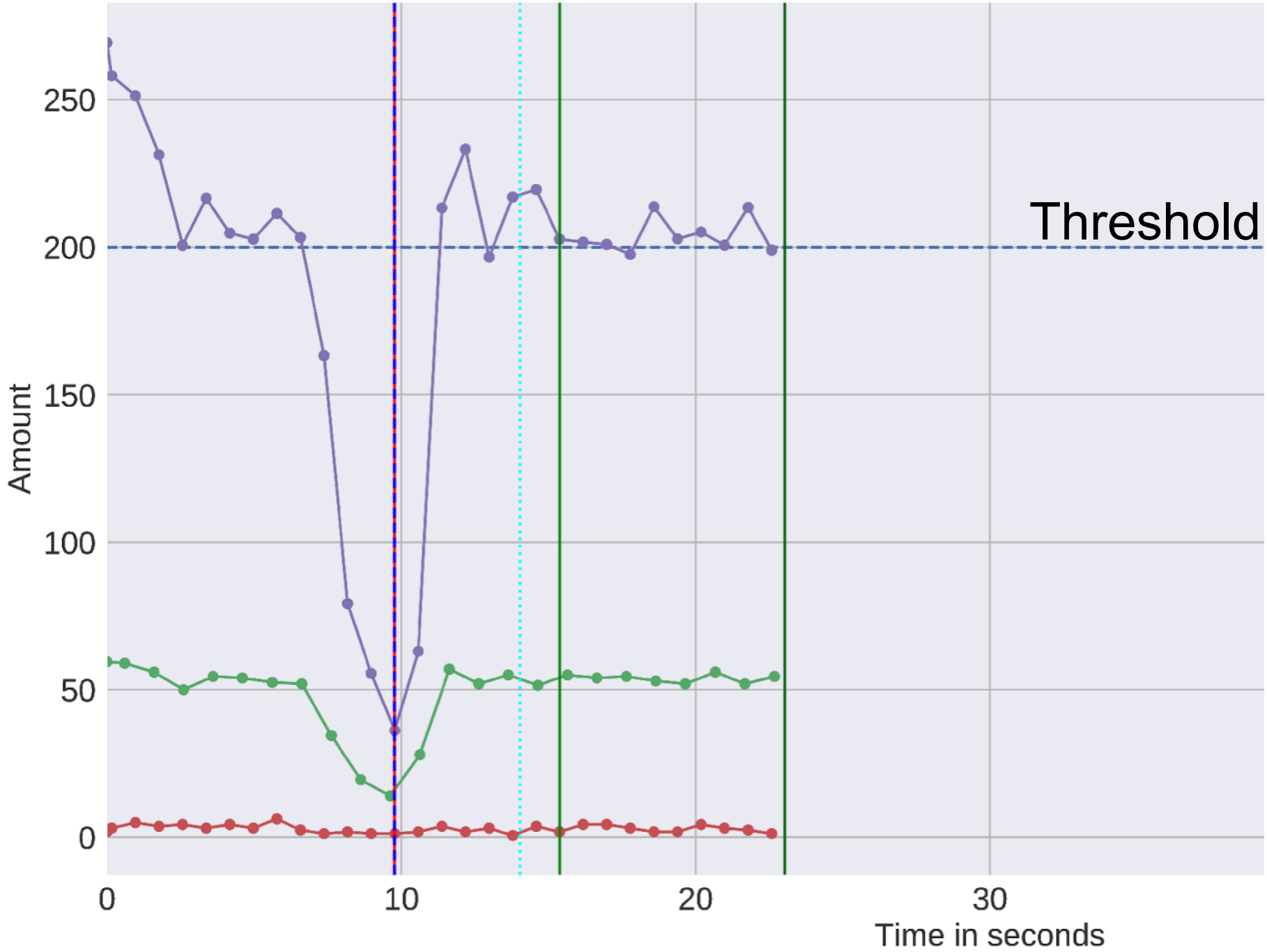
We vary the following parameters:

- Response timeout: 30-40-60s
- Attack size: Light, Medium, Heavy

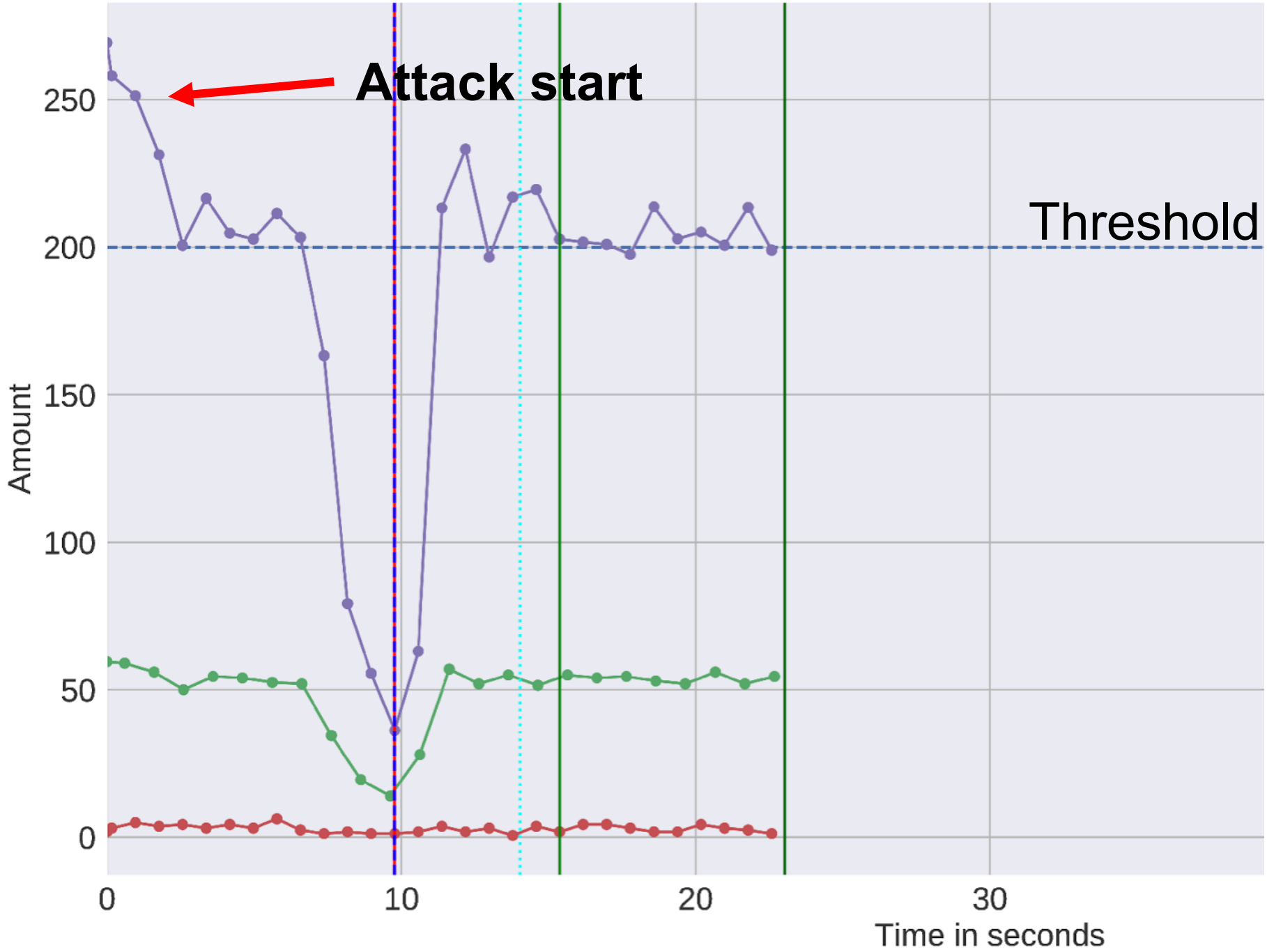
The recorded values are *averaged* over a moving window of 10 samples.

For impact we use a ceiling of 2 * threshold to prevent the impact of unbounded metrics from growing out of scale towards infinity.

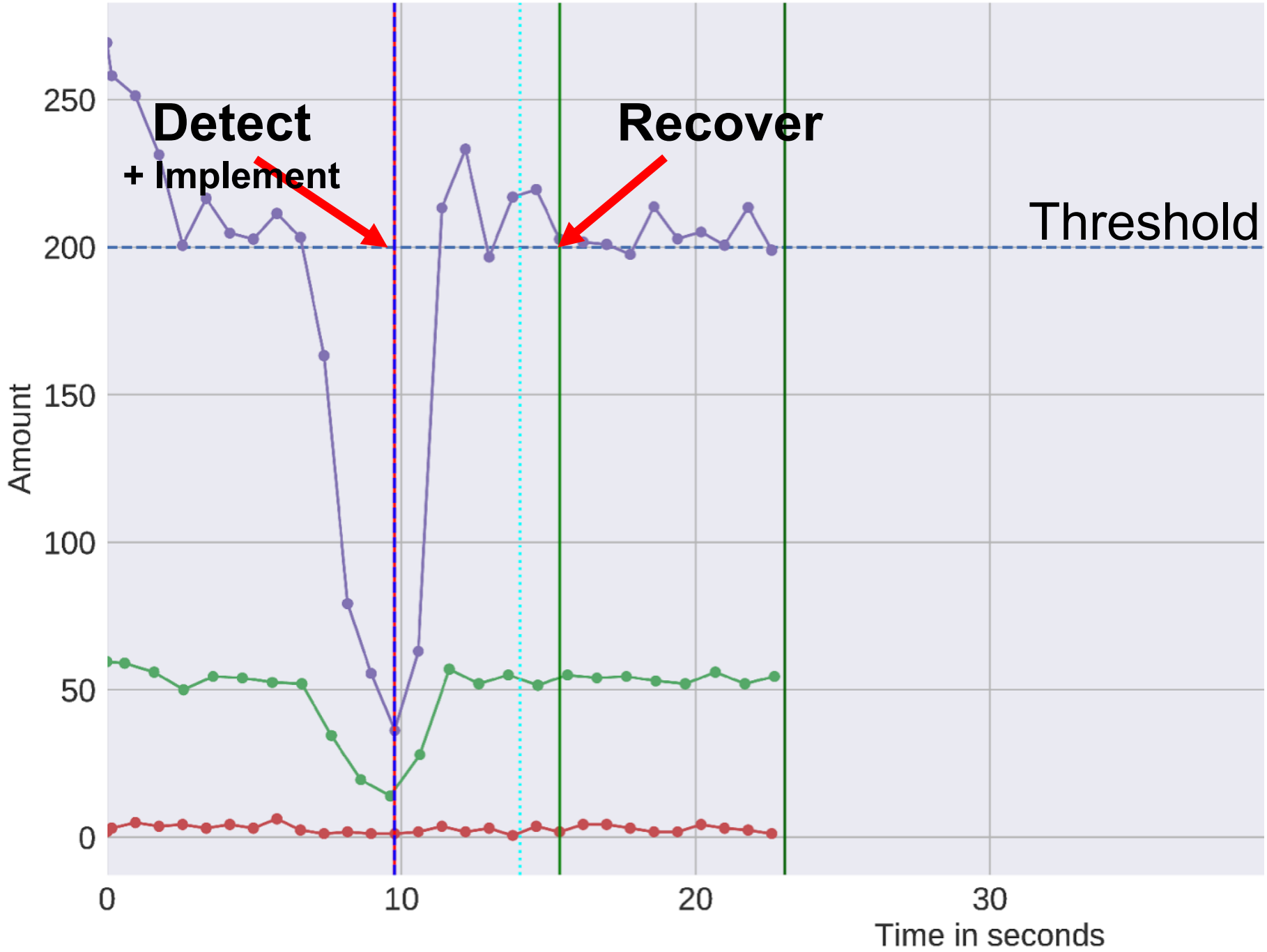
# Example: DDos attack



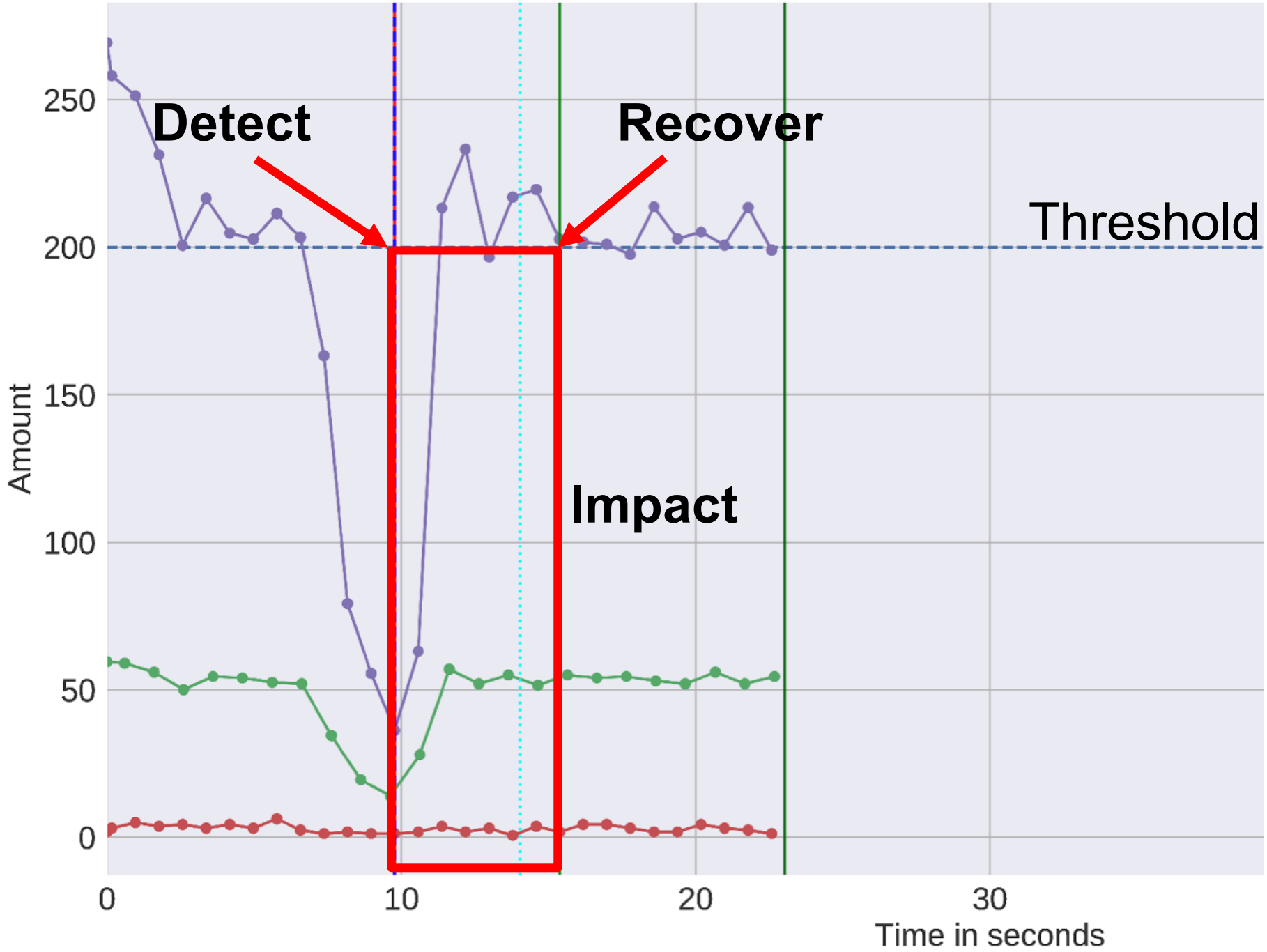
# Example: DDos attack



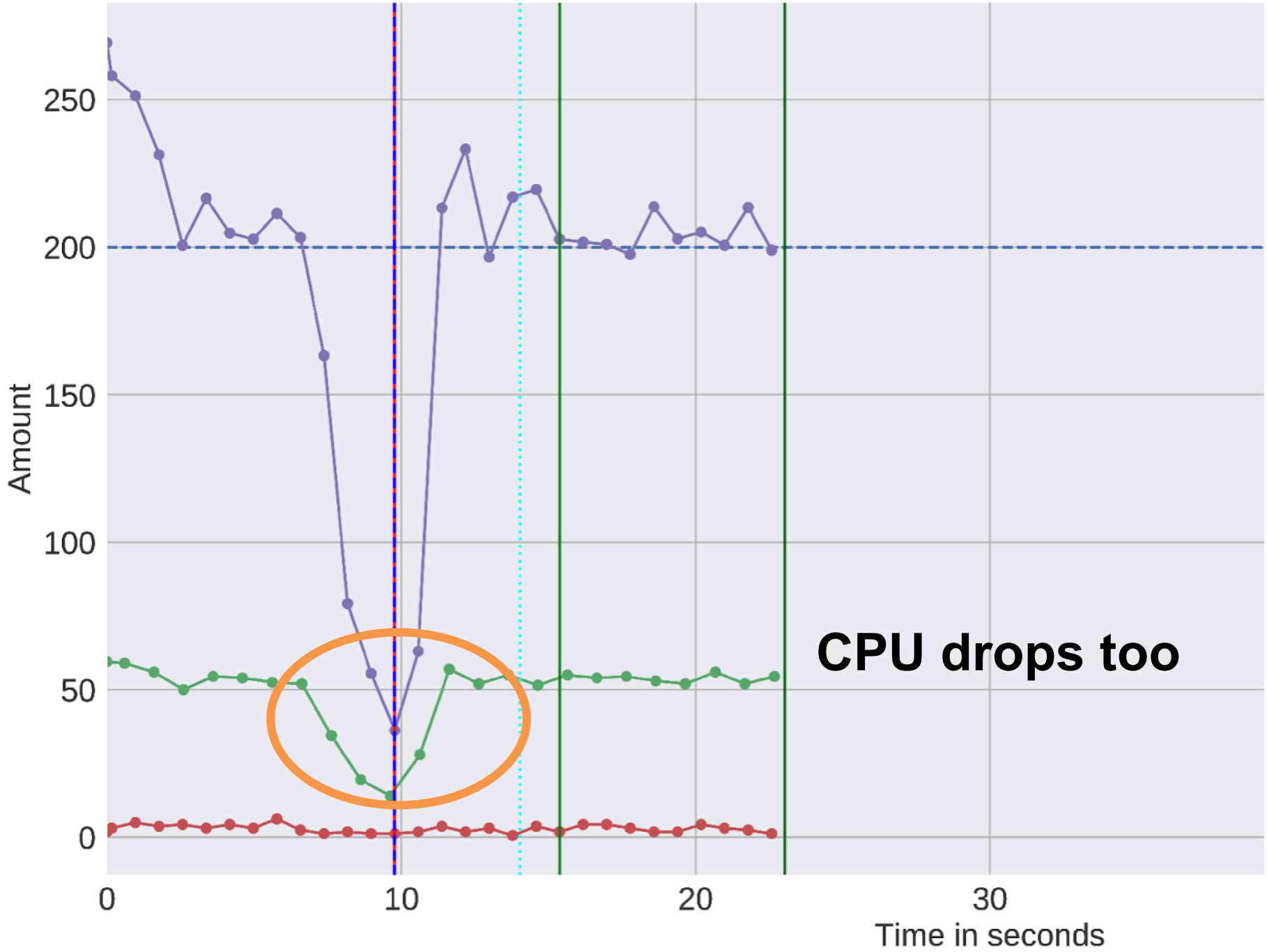
# Example: DDos attack



# Example: DDos attack



# Example: DDos attack (Bonus)



# Success rate

**Defence success rate in %** (50 attempts)

| Attack | Defence | | Size | |
|---|---|---|---|---|
| | | Light | Medium | Heavy |
| | | | | |
| ***CPU_attack*** | ***captcha*** | 68 | 10 | 0 |
| | ***honeypot*** | 100 | 100 | 100 |
| | ***udp-filter*** | 0 | 0 | 0 |
| | ***udp-rateup*** | 0 | 0 | 0 |
| ***pwd_bf_attack*** | ***captcha*** | 100 | 100 | 100 |
| | ***honeypot*** | 100 | 100 | 100 |
| | ***udp-filter*** | 0 | 0 | 0 |
| | ***udp-rateup*** | 0 | 0 | 0 |
| ***DDoS_attack*** | ***captcha*** | 0 | 0 | 0 |
| | ***honeypot*** | 6 | 0 | 0 |
| | ***udp-filter*** | 100 | 100 | 100 |
| | ***udp-rateup*** | 64 | 0 | 0 |

# Success rate

**Defence success rate in %** (50 attempts)

| Attack | Defence | | Size | |
|---|---|---|---|---|
| | | Light | Medium | Heavy |
| | | | | |
| ***CPU_attack*** | ***captcha*** | 68 | 10 | 0 |
| | ***honeypot*** | 100 | 100 | 100 |
| | ***udp-filter*** | 0 | 0 | 0 |
| | ***udp-rateup*** | 0 | 0 | 0 |
| ***pwd_bf_attack*** | ***captcha*** | 100 | 100 | 100 |
| | ***honeypot*** | 100 | 100 | 100 |
| | ***udp-filter*** | 0 | 0 | 0 |
| | ***udp-rateup*** | 0 | 0 | 0 |
| ***DDoS_attack*** | ***captcha*** | 0 | 0 | 0 |
| | ***honeypot*** | 6 | 0 | 0 |
| | ***udp-filter*** | 100 | 100 | 100 |
| | ***udp-rateup*** | 64 | 0 | 0 |

# Success rate

**Defence success rate in %** (50 attempts)

| Attack | Defence | | Size | |
|---|---|---|---|---|
| | | Light | Medium | Heavy |
| | | | | |
| ***CPU_attack*** | ***captcha*** | 68 | 10 | 0 |
| | ***honeypot*** | 100 | 100 | 100 |
| | ***udp-filter*** | 0 | 0 | 0 |
| | ***udp-rateup*** | 0 | 0 | 0 |
| ***pwd_bf_attack*** | ***captcha*** | 100 | 100 | 100 |
| | ***honeypot*** | 100 | 100 | 100 |
| | ***udp-filter*** | 0 | 0 | 0 |
| | ***udp-rateup*** | 0 | 0 | 0 |
| ***DDoS_attack*** | ***captcha*** | 0 | 0 | 0 |
| | ***honeypot*** | 6 | 0 | 0 |
| | ***udp-filter*** | 100 | 100 | 100 |
| | ***udp-rateup*** | 64 | 0 | 0 |

# Success rate

**Defence success rate in %** (50 attempts)

| Attack | Defence | | Size | |
|---|---|---|---|---|
| | | Light | Medium | Heavy |
| | | | | |
| ***CPU_attack*** | ***captcha*** | 68 | 10 | 0 |
| | ***honeypot*** | 100 | 100 | 100 |
| | ***udp-filter*** | 0 | 0 | 0 |
| | ***udp-rateup*** | 0 | 0 | 0 |
| ***pwd_bf_attack*** | ***captcha*** | 100 | 100 | 100 |
| | ***honeypot*** | 100 | 100 | 100 |
| | ***udp-filter*** | 0 | 0 | 0 |
| | ***udp-rateup*** | 0 | 0 | 0 |
| ***DDoS_attack*** | ***captcha*** | 0 | 0 | 0 |
| | ***honeypot*** | 6 | 0 | 0 |
| | ***udp-filter*** | 100 | 100 | 100 |
| | ***udp-rateup*** | 64 | 0 | 0 |

# Efficiency

## Efficiency of countermeasures
(50 attempts)

| Attack | Defence | | Size | |
|---|---|---|---|---|
| | | Light | Medium | Heavy |
| | | | | |
| ***CPU_attack*** | ***captcha*** | 0.44 | 0.04 | 0.00 |
| | ***honeypot*** | 0.98 | 0.98 | 0.97 |
| ***pwd_bf_attack*** | ***captcha*** | 0.94 | 0.94 | 0.94 |
| | ***honeypot*** | 0.94 | 0.94 | 0.94 |
| ***DDoS_attack*** | ***udp-filter*** | 1.00 | 1.00 | 0.99 |
| | ***udp-rateup*** | 0.35 | 0.00 | 0.00 |

# Ranking defences

**Based on the defence efficiency a SARNET will make the following choices**

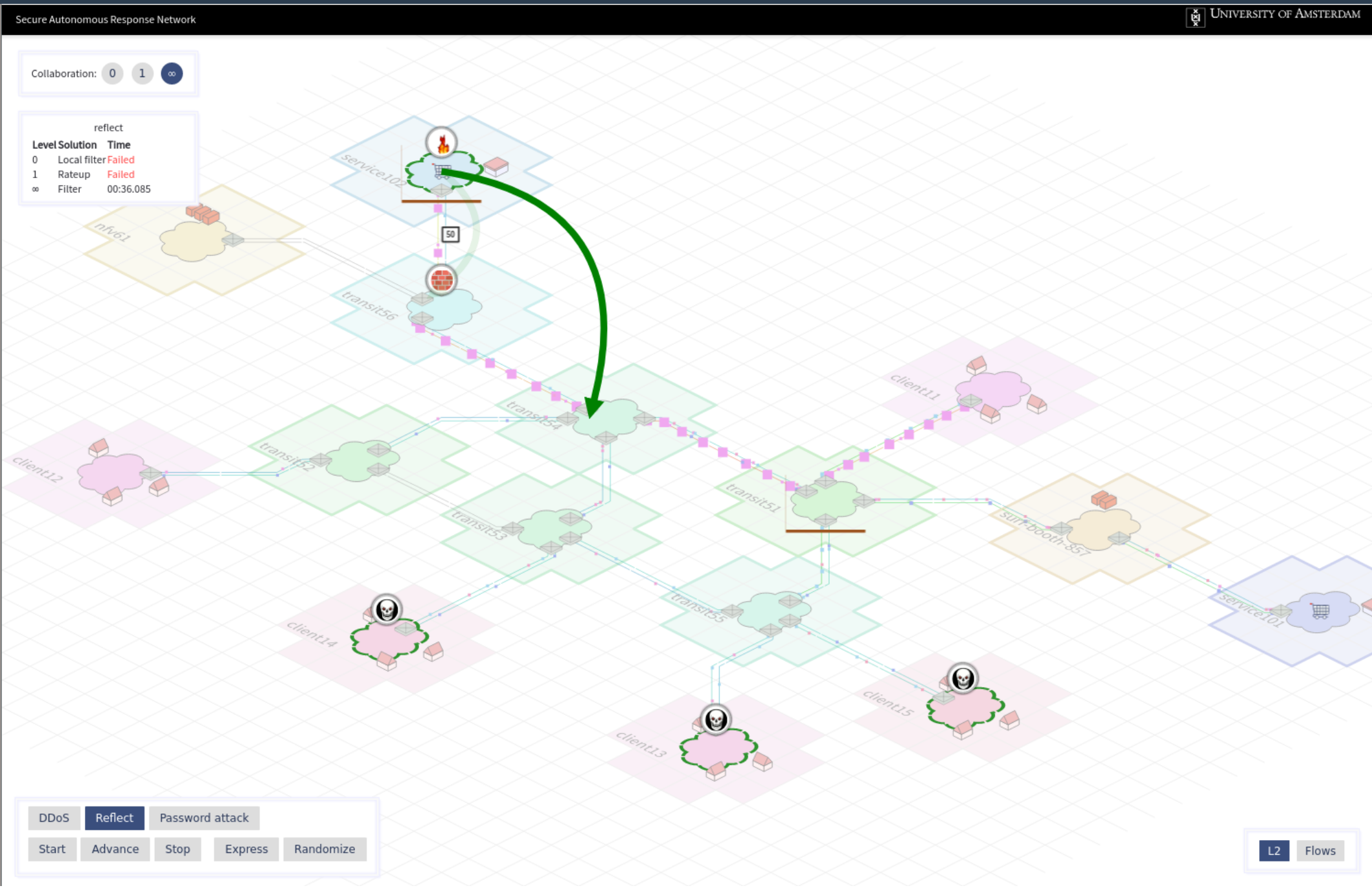
| | **First choice** | **Second choice** |
|---|---|---|
| ***CPU_attack*** | captcha | honeypot |
| ***pwd_bf_attack*** | honeypot/captcha | - |
| ***DDoS_attack*** | udp-filter | - |
| ***DDoS_attack*** **(light)** | udp-filter | udp-rateup |

# Conclusions

Determining efficiency is a first step towards ranking countermeasures and self learning.

Efficiency is *universal* enough to apply and compare on new countermeasures.

# Multi-Domain Demo at booth #1281 (Ciena)



**Contact:**

Ralph Koning
r.koning _at_ uva.nl

**SC17: @Ciena booth #1281**

**@SURF booth #857**

https://staff.fnwi.uva.nl/r.koning/

https://sarnet.uvalight.net

**Research made possible by:**

TNO

NWO Netherlands Organisation for Scientific Research

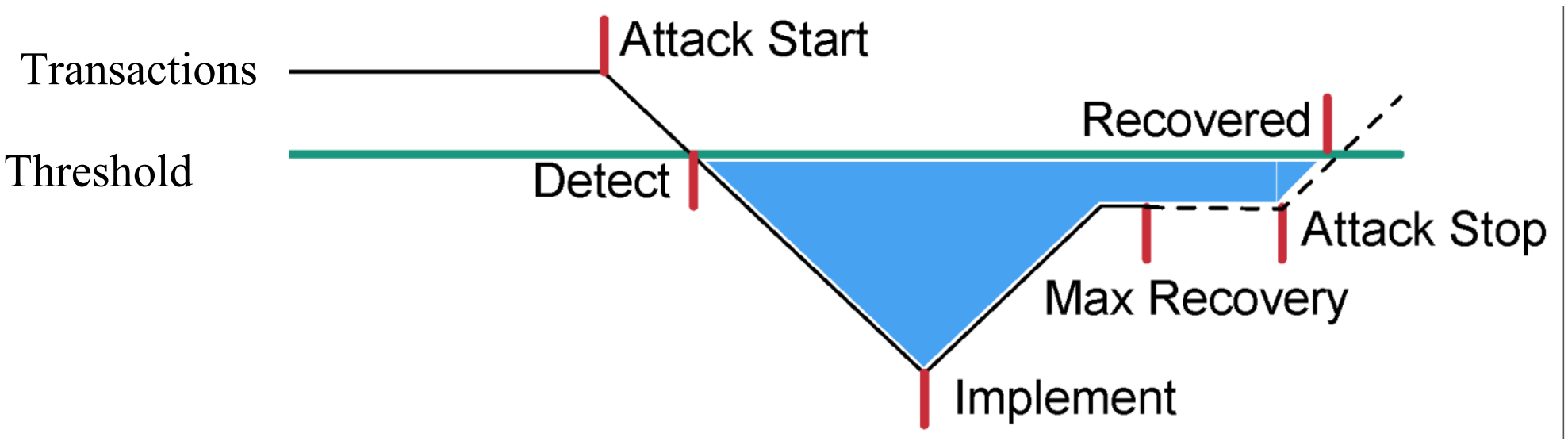
ciena

AIRFRANCE KLM

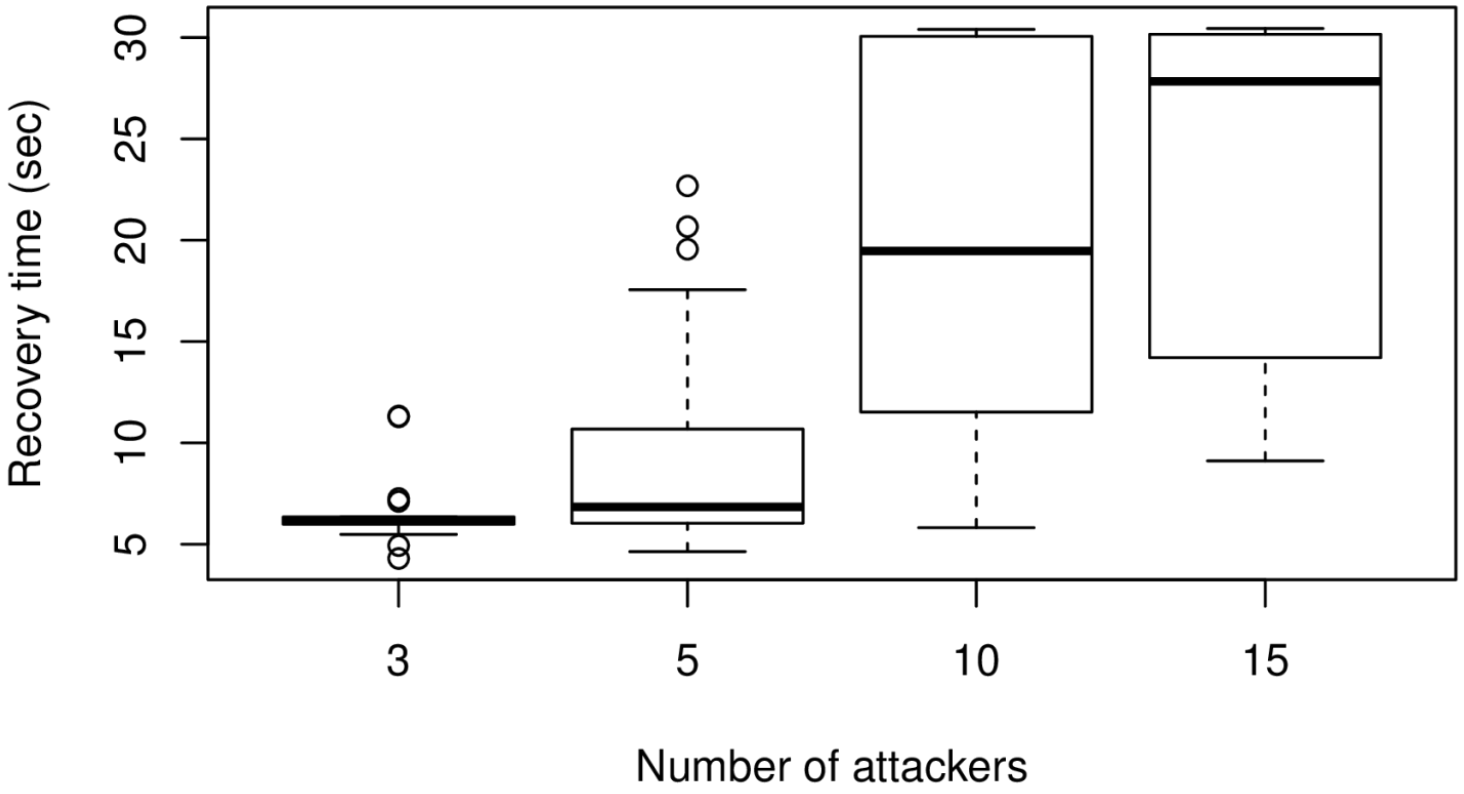
UNIVERSITEIT VAN AMSTERDAM

COMMIT/

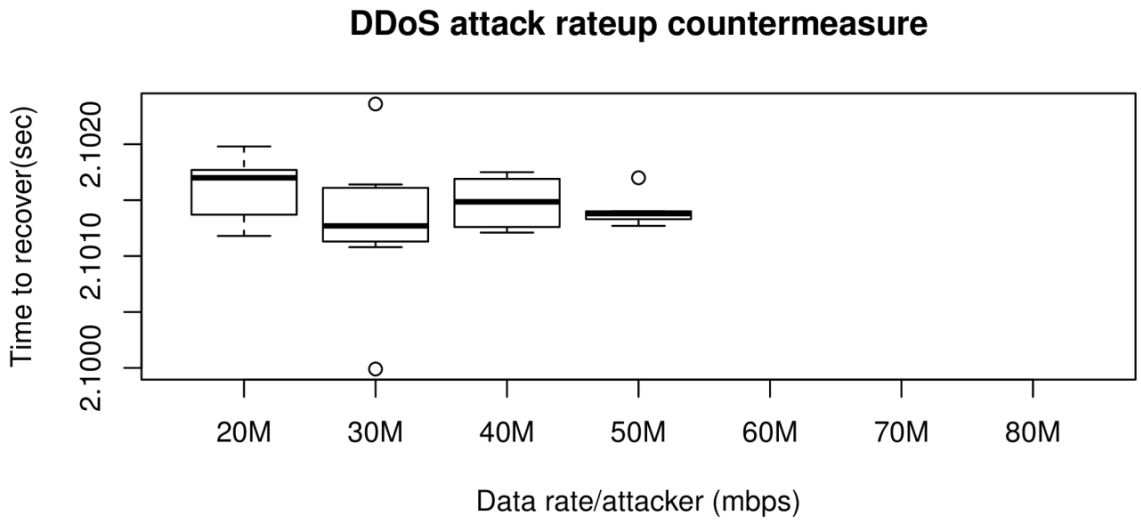
# Effectiveness and Impact (2)

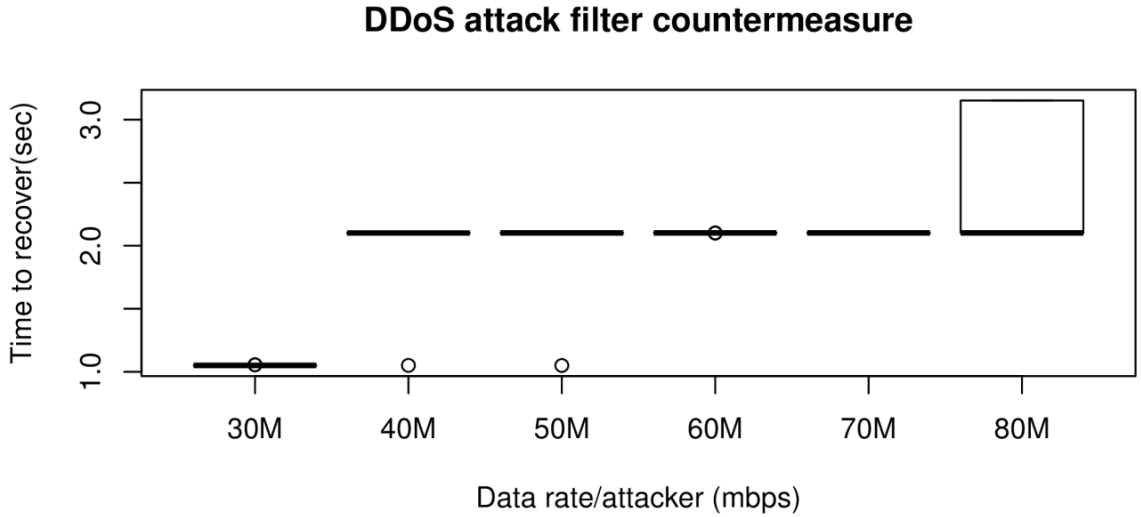


# CPU attack recovery time



# DDOS recovery time



# Weights

The values we use for $\gamma$ in the efficiency calculation.

| Classification | Sales | Logfail | CPU |
|---|---|---|---|
| ***CPU_attack*** | 0.75 | 0 | 0.25 |
| ***DDoS_attack*** | 1 | 0 | 0 |
| ***pwd_bf_attack*** | 0 | 0 | 1 |