IP Spoofing

(what is it, how it allows largest attacks and how to fix it)

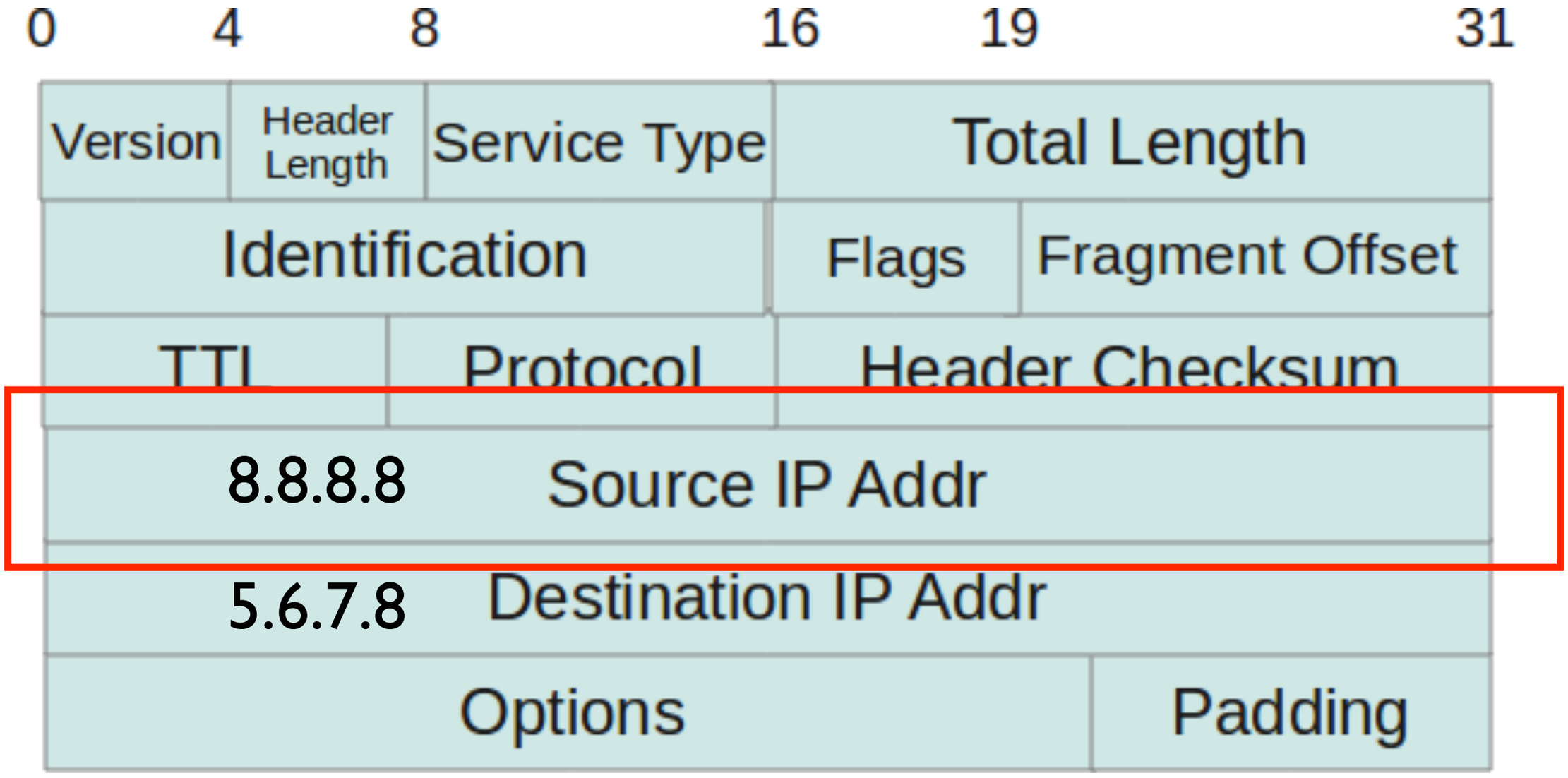
Marek Majkowski

What is it?

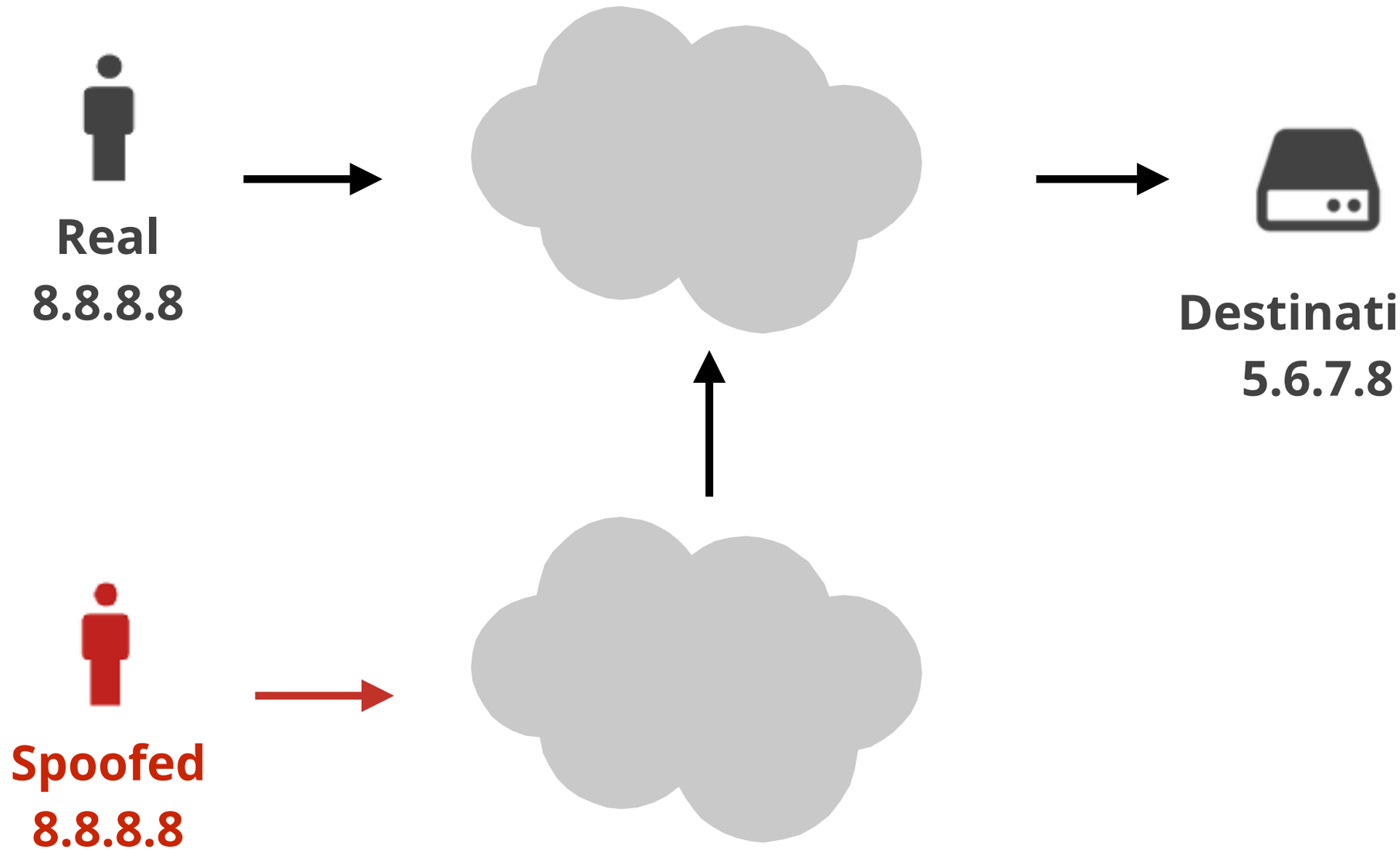
IP Spoofing

(source: DaPuglet)

IP Spoofing


0
4
8
16
19
31
Version
Header Length
Service Type
Total Length
Identification
Flags
Fragment Offset
TTL
Protocol
Header Checksum
8.8.8.8
Source IP Addr
5.6.7.8
Destination IP Addr
Options
Padding

Enables impersonation

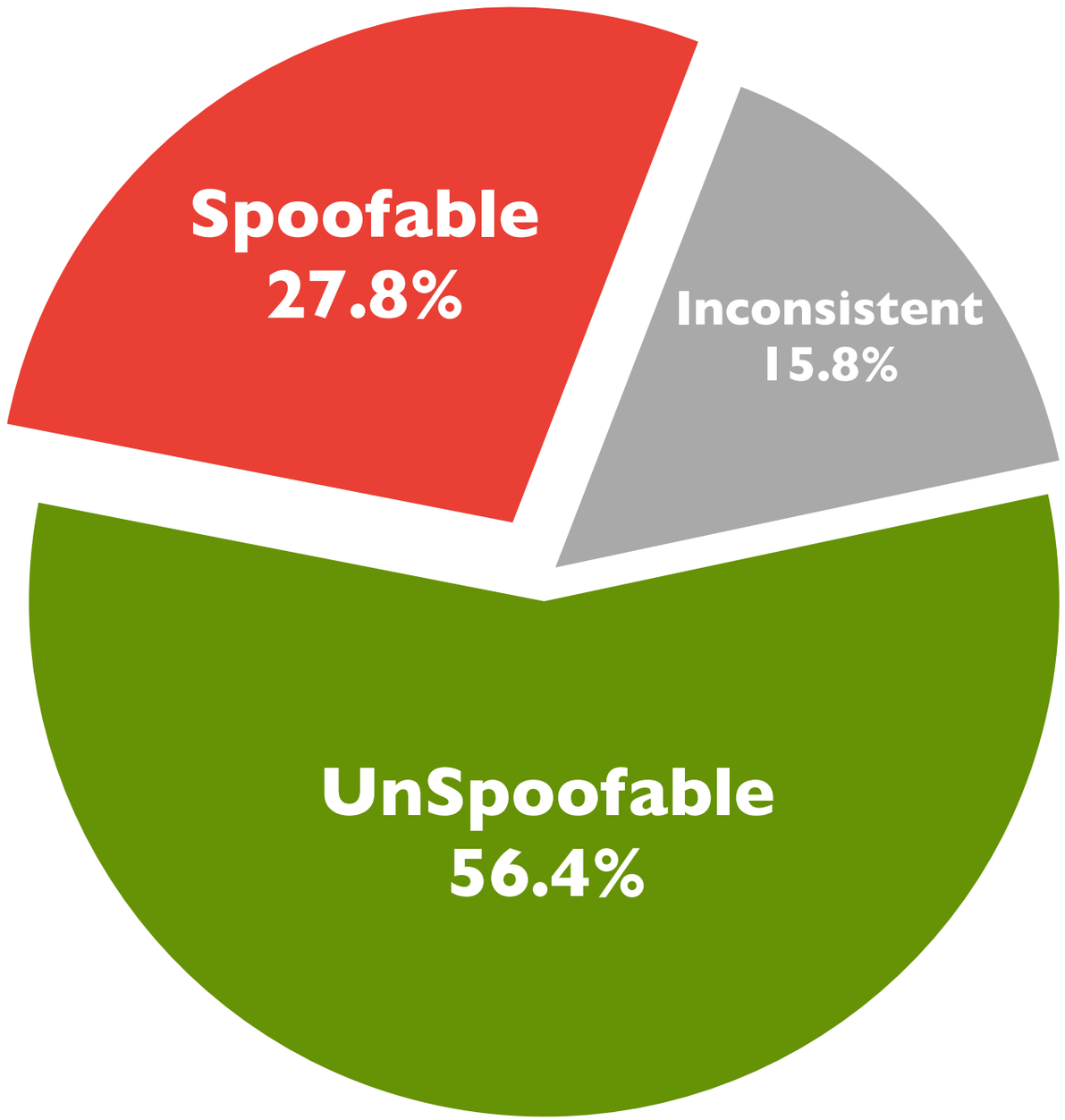


IP Spoofing is still a problem

May 2000: BCP38

spoofer.caida.org

Measured Autonomic Systems

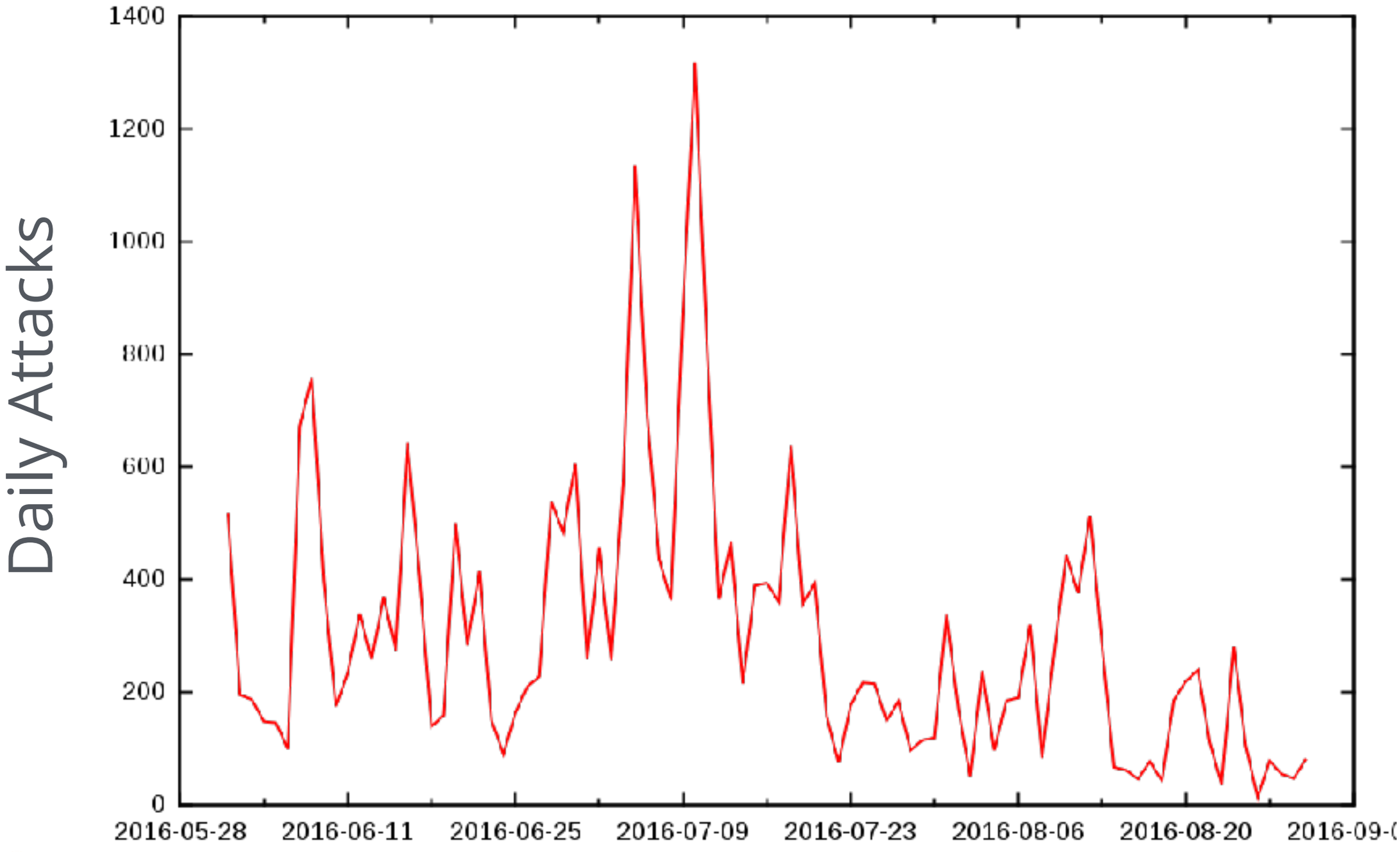


CLOUDFLARE

It allows largest attacks

Global network

Daily attacks

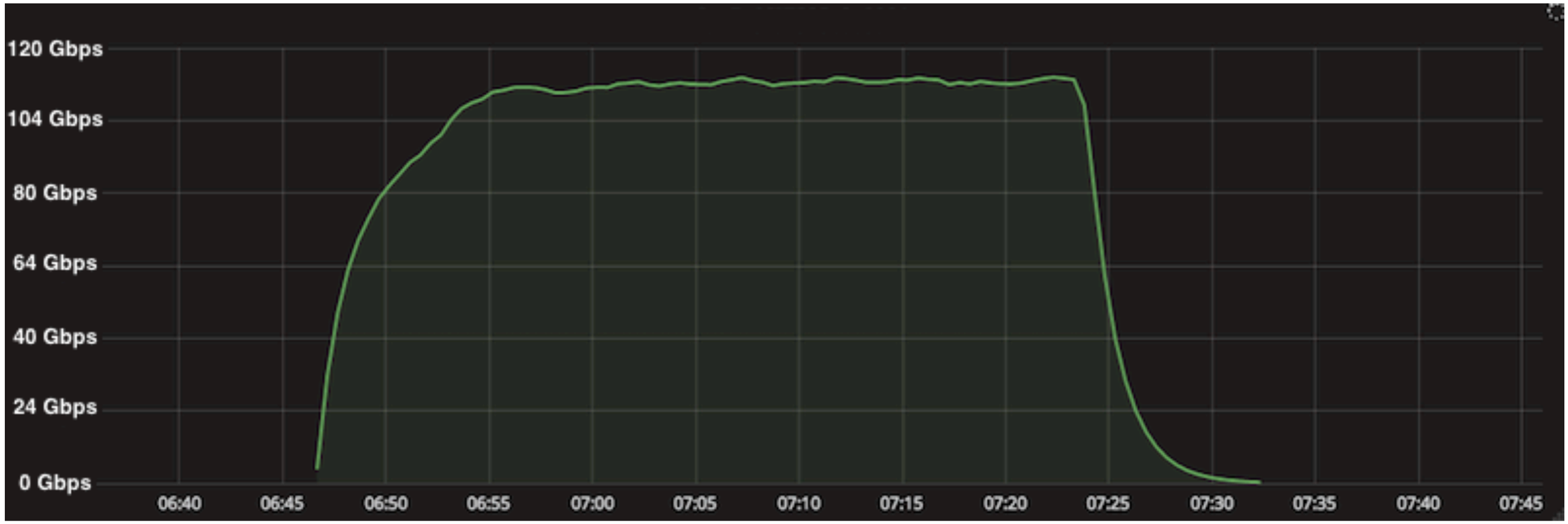


Some are super large

Direct: SYN Flood

"winter of attacks"

Amplification: SSDP



IP Spoofing

(source: DaPuglet)

IP Spoofing

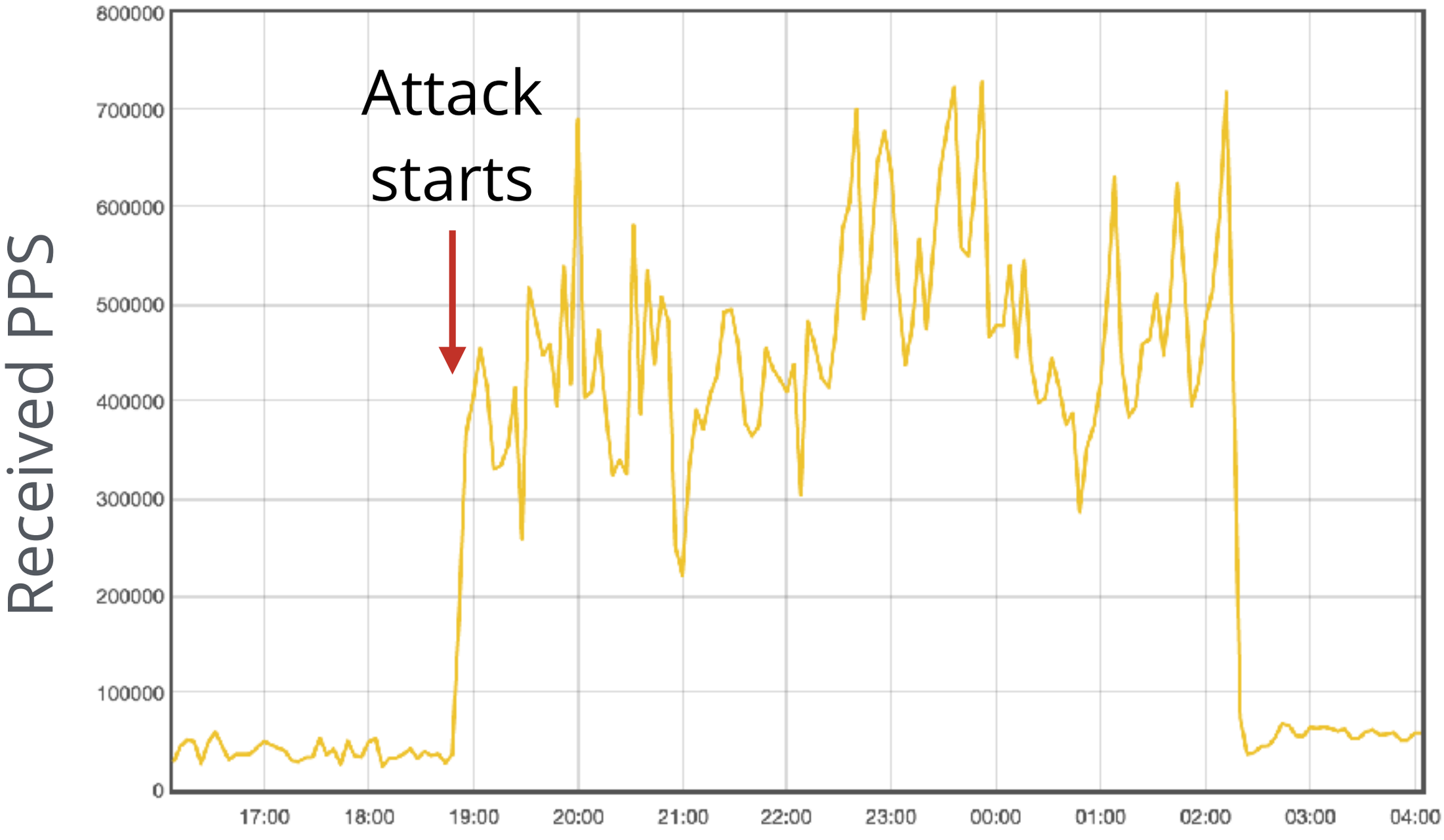
1. Tracing back is impossible

2. Allows sophisticated attacks

Direct: SYN Flood

2. “Winter of attacks”

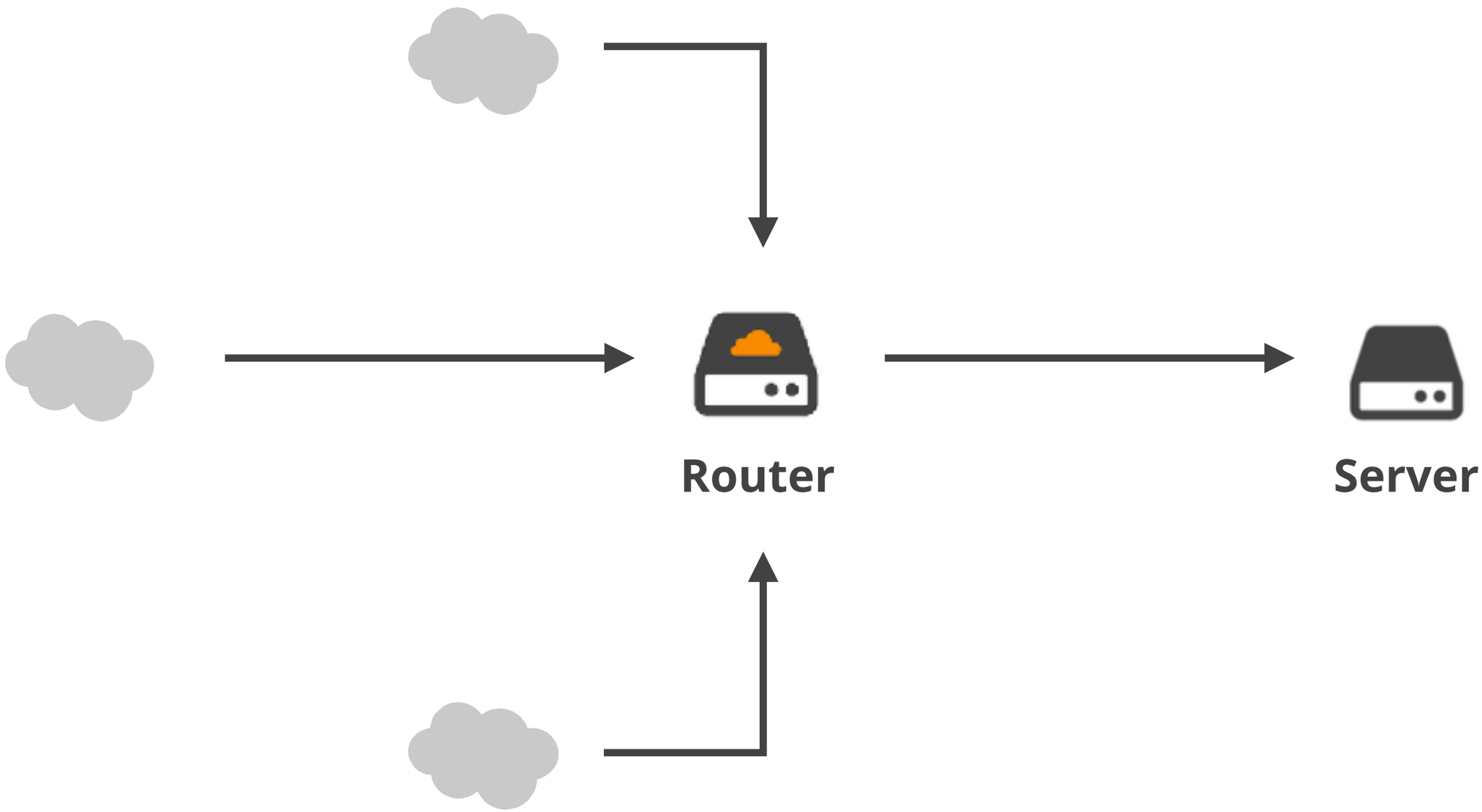
Tracing the attack



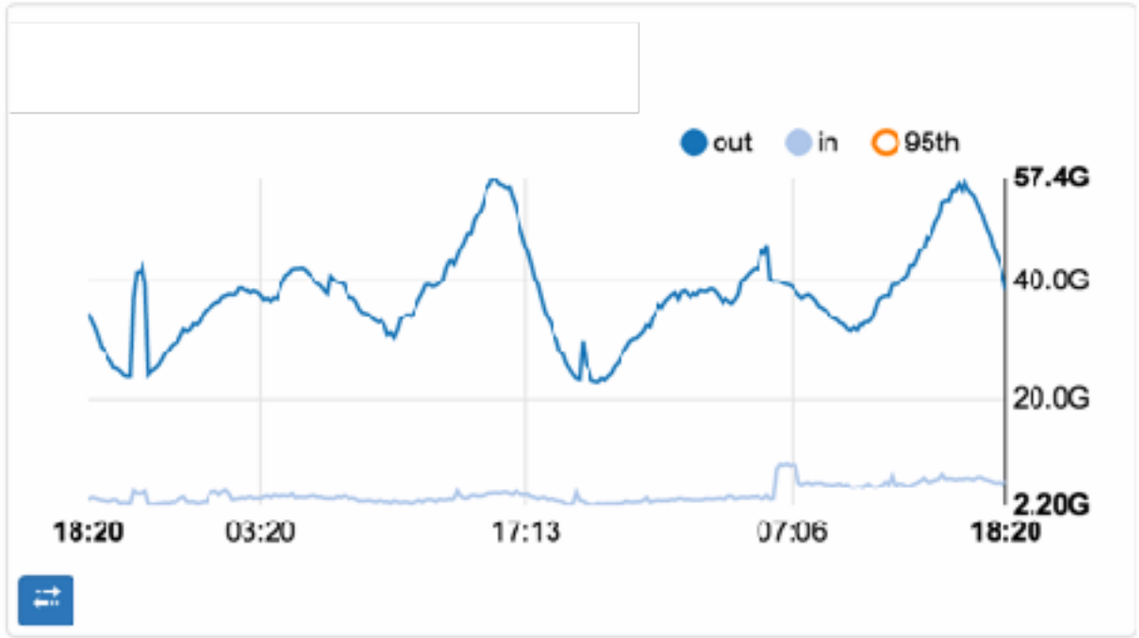
Tcpdump

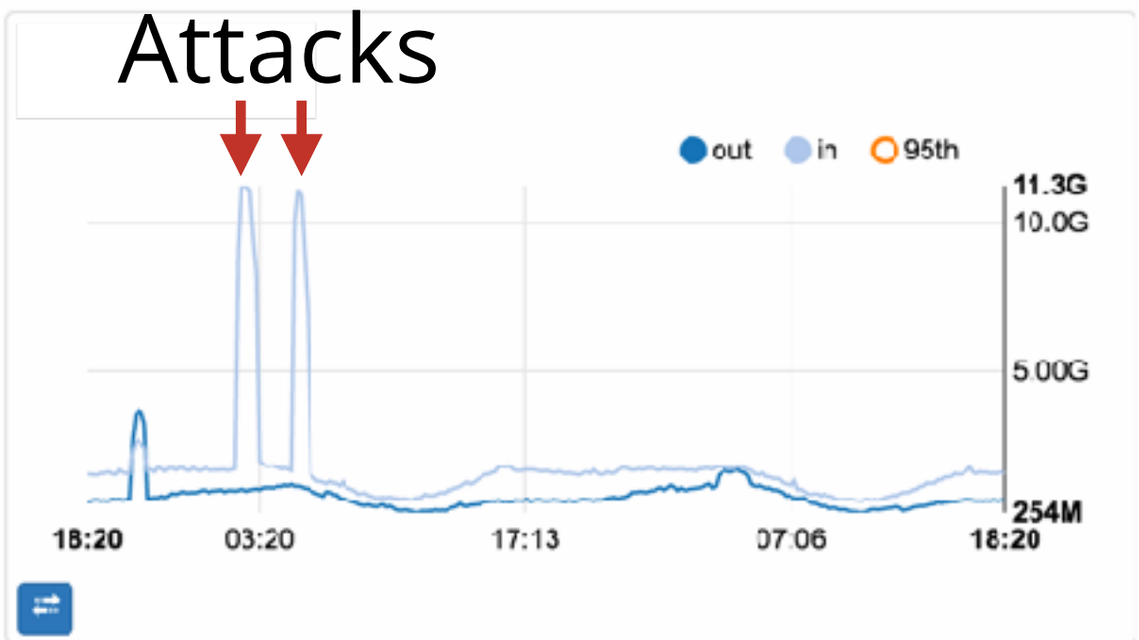
```
$ tcpdump -ni eth0 -c 100

IP 94.242.250.109.47330 > 1.2.3.4:80: Flags [S], seq 1444613291, win 63243
IP 188.138.1.240.61454 > 1.2.3.4:80: Flags [S], seq 1995637287, win 60551
IP 207.244.90.205.17572 > 1.2.3.4:80: Flags [S], seq 1523683071, win 61607
IP 94.242.250.224.65127 > 1.2.3.4:80: Flags [S], seq 928944042, win 61778
IP 207.244.90.205.43074 > 1.2.3.4:80: Flags [S], seq 137074667, win 63891
IP 64.22.81.44.23865 > 1.2.3.4:80: Flags [S], seq 838596928, win 63808
IP 188.138.1.137.23373 > 1.2.3.4:80: Flags [S], seq 593106072, win 60272
IP 207.244.90.205.39653 > 1.2.3.4:80: Flags [S], seq 47289666, win 63210
IP 208.66.78.204.64197 > 1.2.3.4:80: Flags [S], seq 1850809890, win 62714
IP 207.244.90.205.33108 > 1.2.3.4:80: Flags [S], seq 319707959, win 63351
IP 207.244.90.205.6937 > 1.2.3.4:80: Flags [S], seq 1591500126, win 63902
IP 213.152.180.151.60560 > 1.2.3.4:80: Flags [S], seq 1902119375, win 62511
IP 64.22.79.127.11061 > 1.2.3.4:80: Flags [S], seq 1456438676, win 62148
```

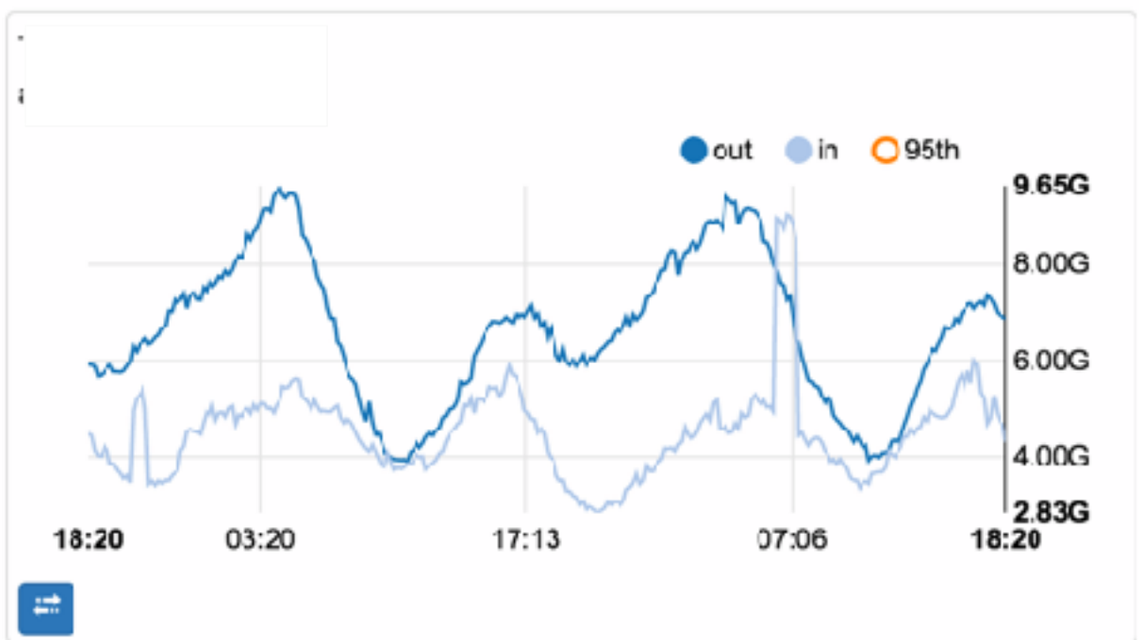
Which router iface is it from?



Identifying interface

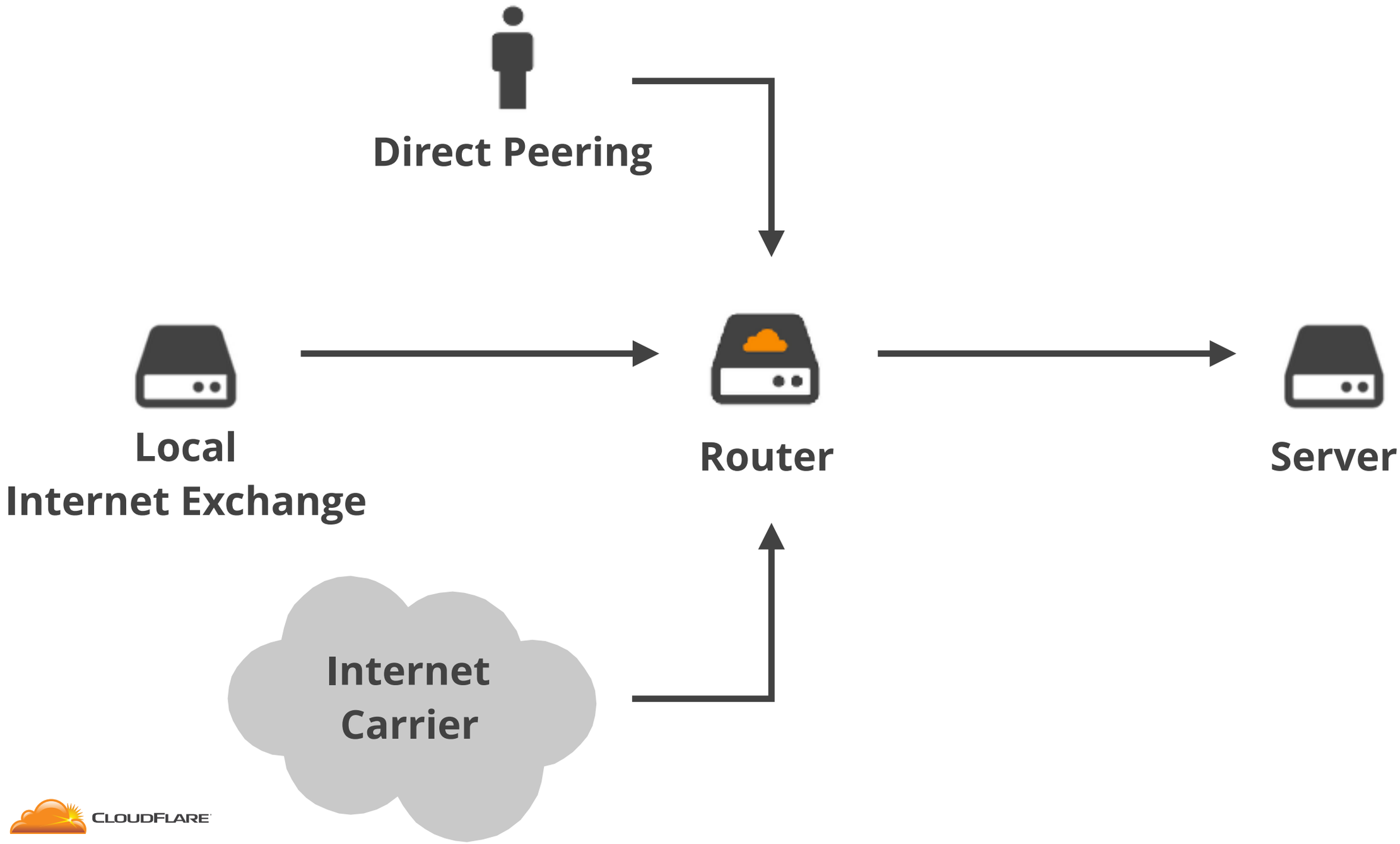






Identifying the interface

Other side of the cable



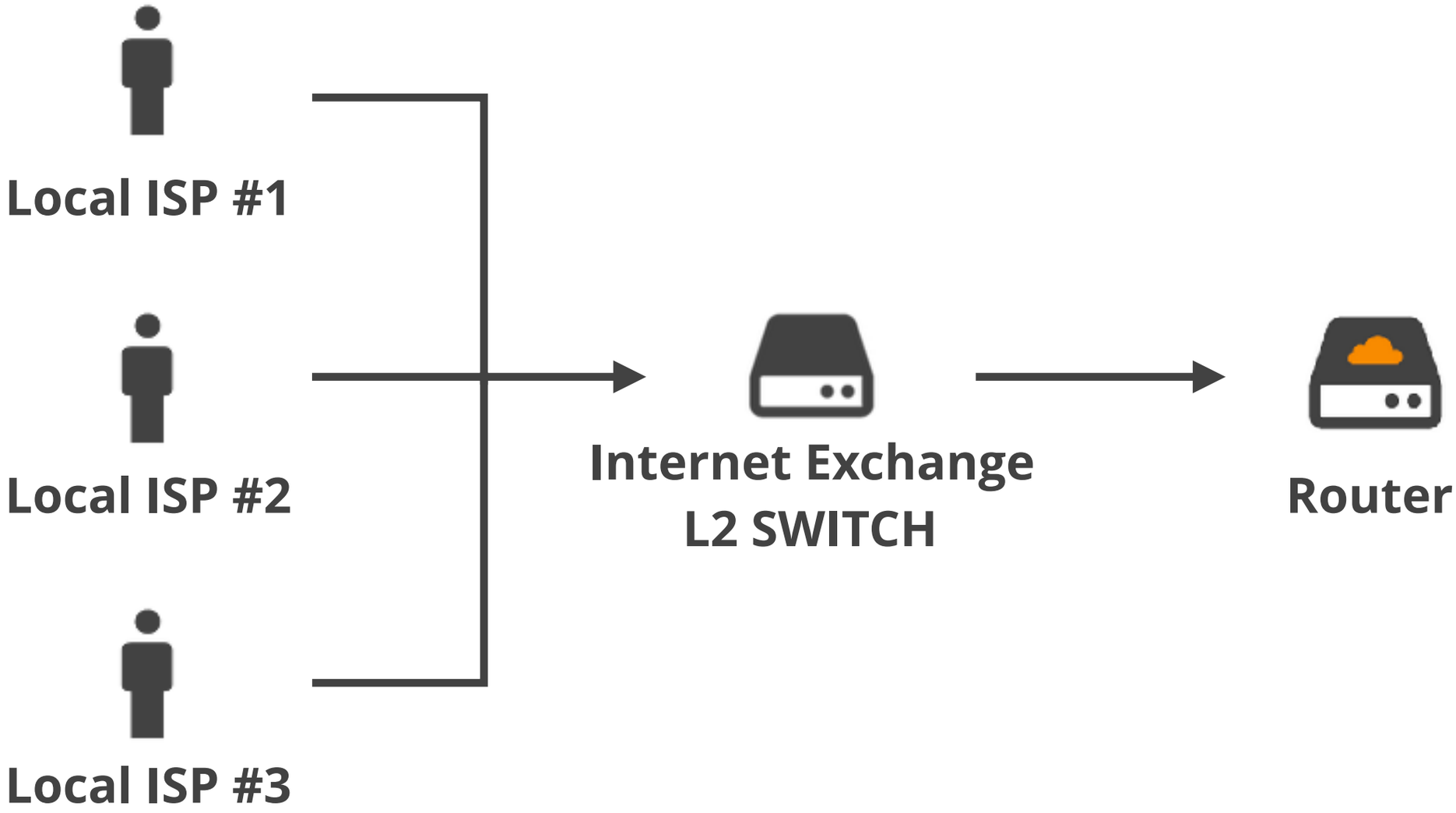
1. Direct Peering

2. Internet Exchange

3. Internet Carrier

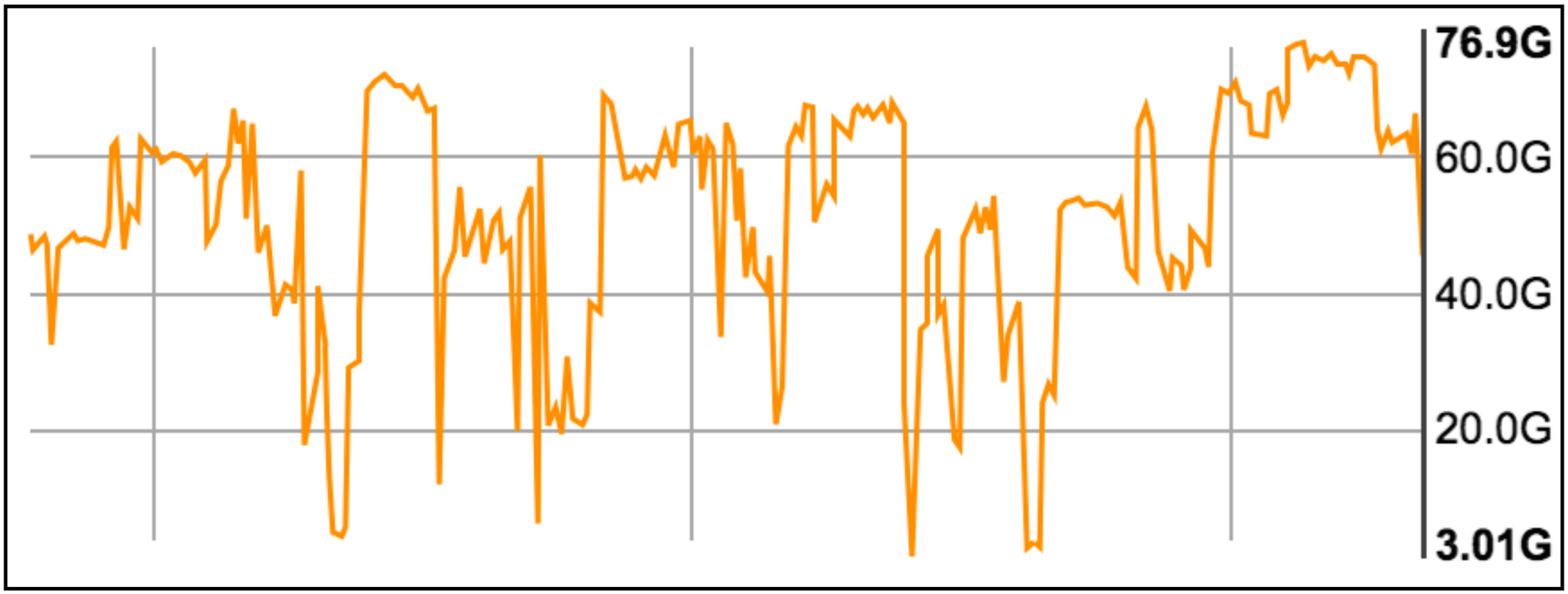
2. Internet Exchanges

2. Internet Exchanges

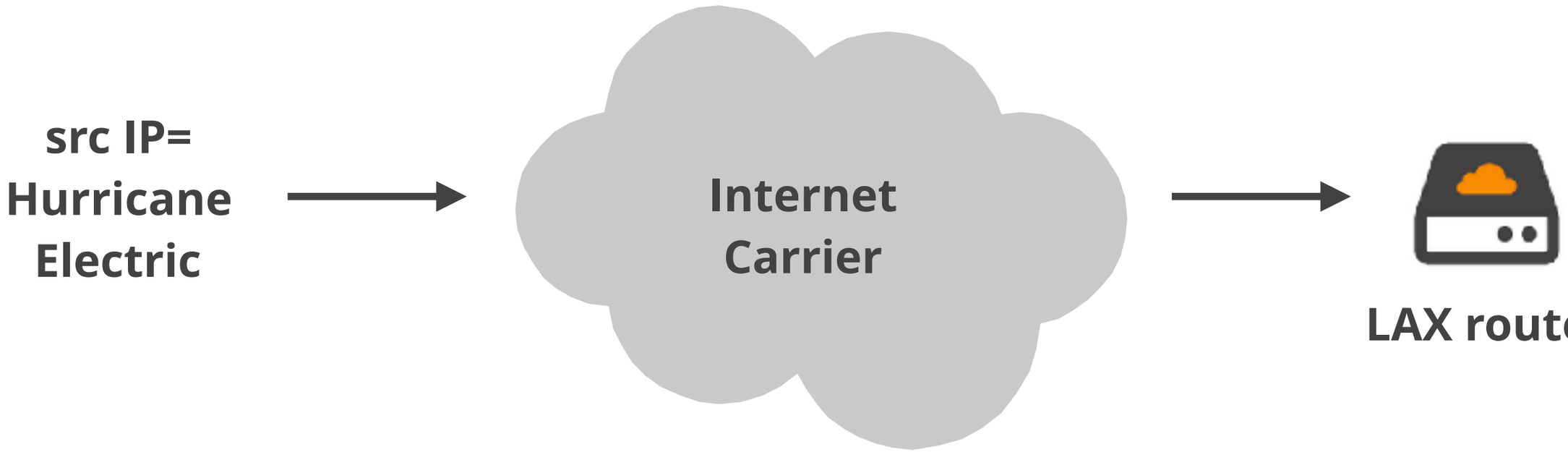


3. Internet Carriers

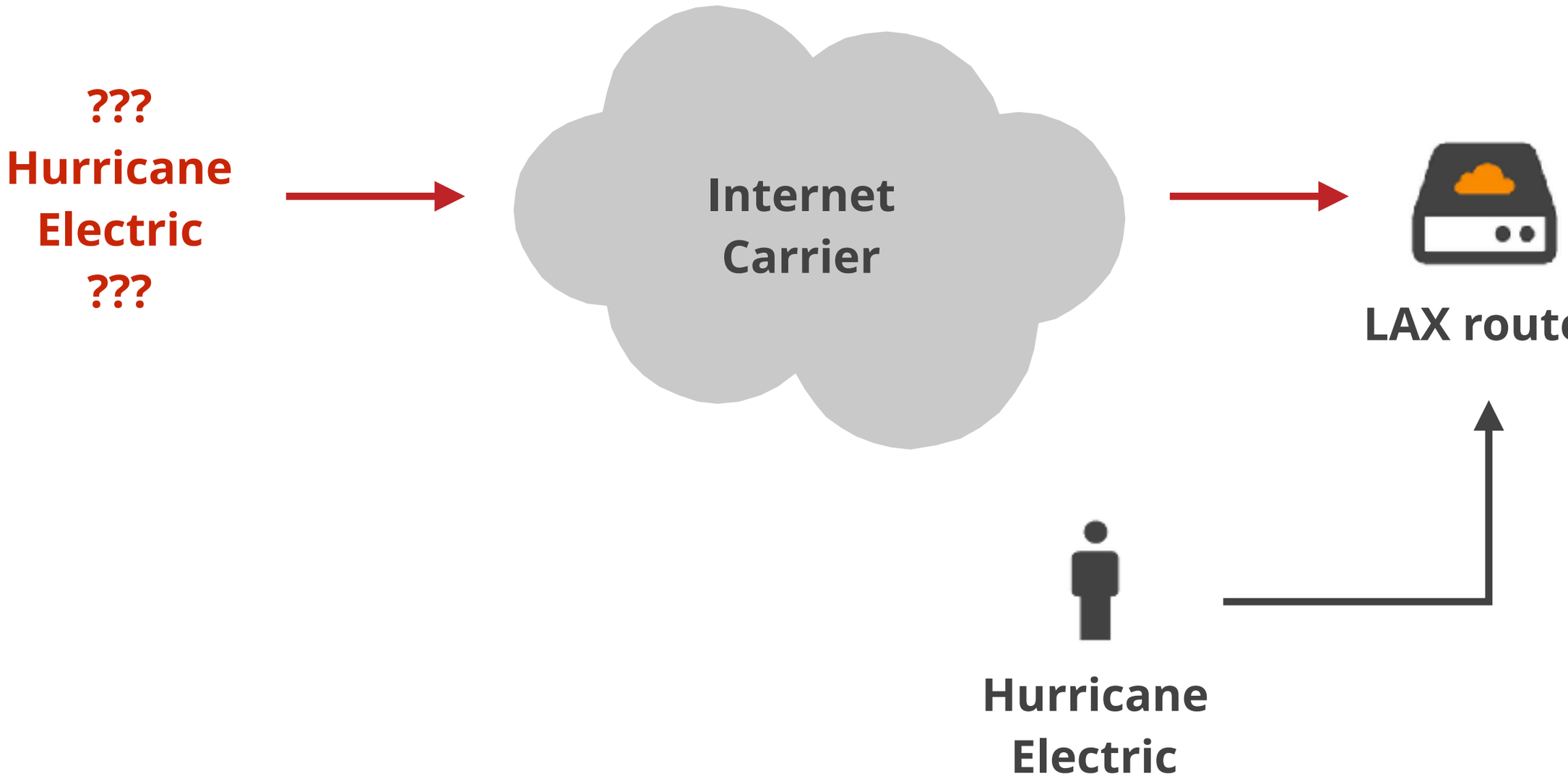
“Winter of attacks”



“Winter of attacks”



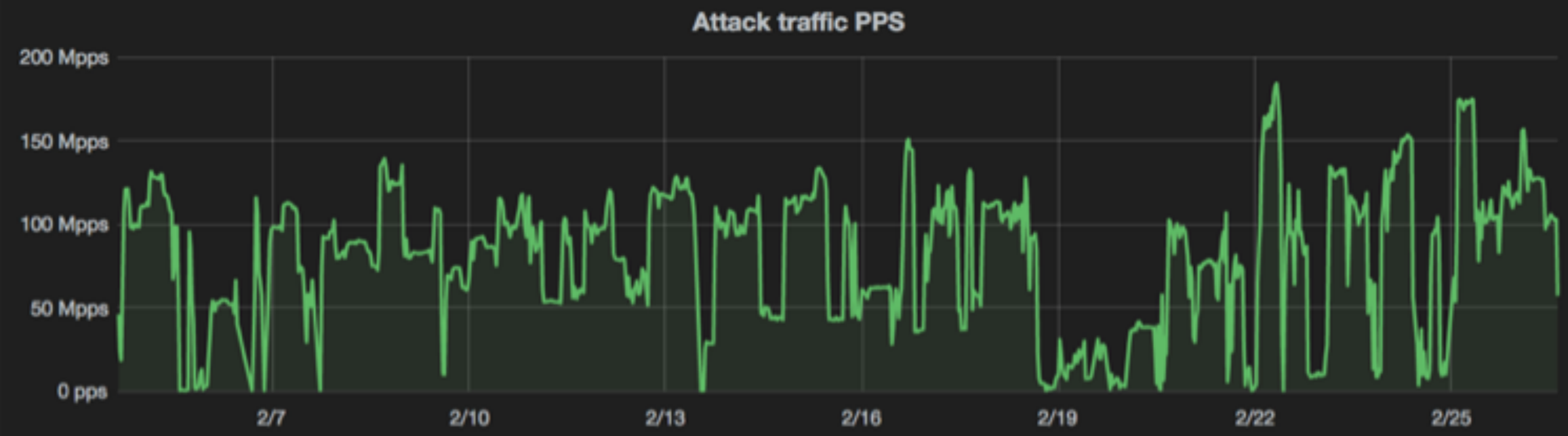
“Winter of attacks”

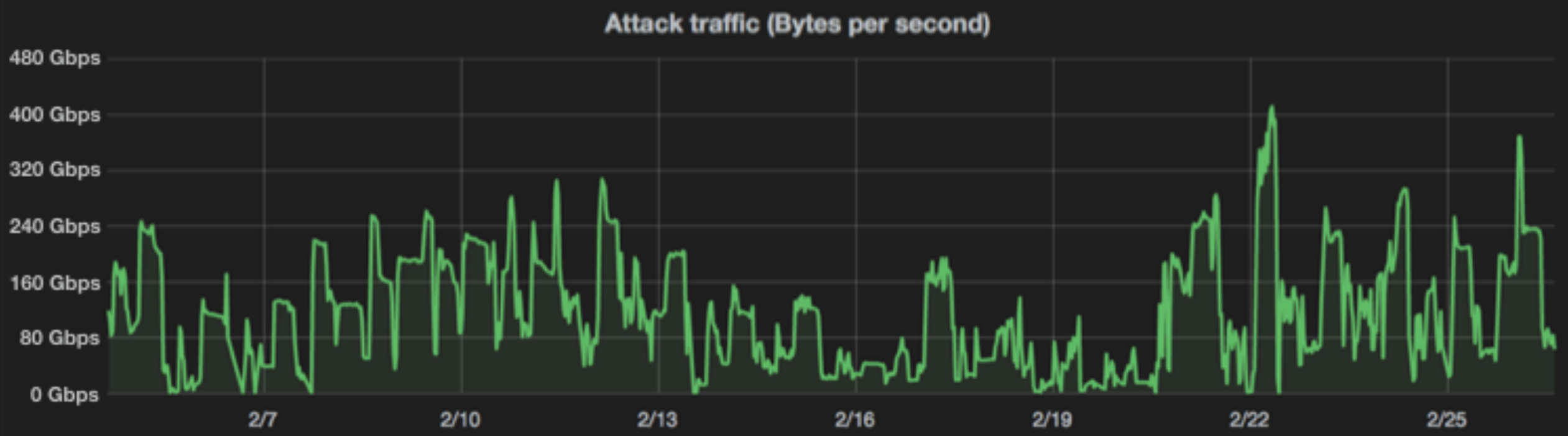


Lack of attribution

==

impossible to fight


Attack traffic PPS
200 Mpps
150 Mpps
100 Mpps
50 Mpps
0 pps
2/7
2/10
2/13
2/16
2/19
2/22
2/25


Attack traffic (Bytes per second)
480 Gbps
400 Gbps
320 Gbps
240 Gbps
160 Gbps
80 Gbps
0 Gbps
2/7
2/10
2/13
2/16
2/19
2/22
2/25

Blocked with BPF

```
iptables -A INPUT \
    --dst 1.2.3.4 \
    -p udp --dport 53 \
    -m bpf --bytecode "14,0 0 0 20,177 0 0 0,12 0 0 0,7
0 0 0,64 0 0 0,21 0 7 124090465,64 0 0 4,21 0 5
1836084325,64 0 0 8,21 0 3 56848237,80 0 0 12,21 0 1
0,6 0 0 1,6 0 0 0" \
   -j DROP
```

BPF bytecode

```
    ldx 4*([14]&0xf)
    ld #34
    add x
    tax
lb_0:
    ldb [x + 0]
    add x
    add #1
    tax
    ld [x + 0]
    jneq #0x07657861, lb_1
    ld [x + 4]
    jneq #0x6d706c65, lb_1
    ld [x + 8]
    jneq #0x03636f6d, lb_1
    ldb [x + 12]
    jneq #0x00, lb_1
    ret #1
lb_1:
    ret #0
```

Introducing the BPF Tools

03 Jul 2014 by Marek Majkowski.

In a recent article I described the basic concepts behind the use of Berkeley Packet Filter (aka BSD Packet filter or BPF) bytecode for high performance packet filtering, and the xt_bpf iptables module. In this post I'll explain how we use BPF and xt_bpf as one tool to deal with large scale DDoS attacks.

And, today, CloudFlare is open sourcing the tools we've created to generate and deploy BPF rules.

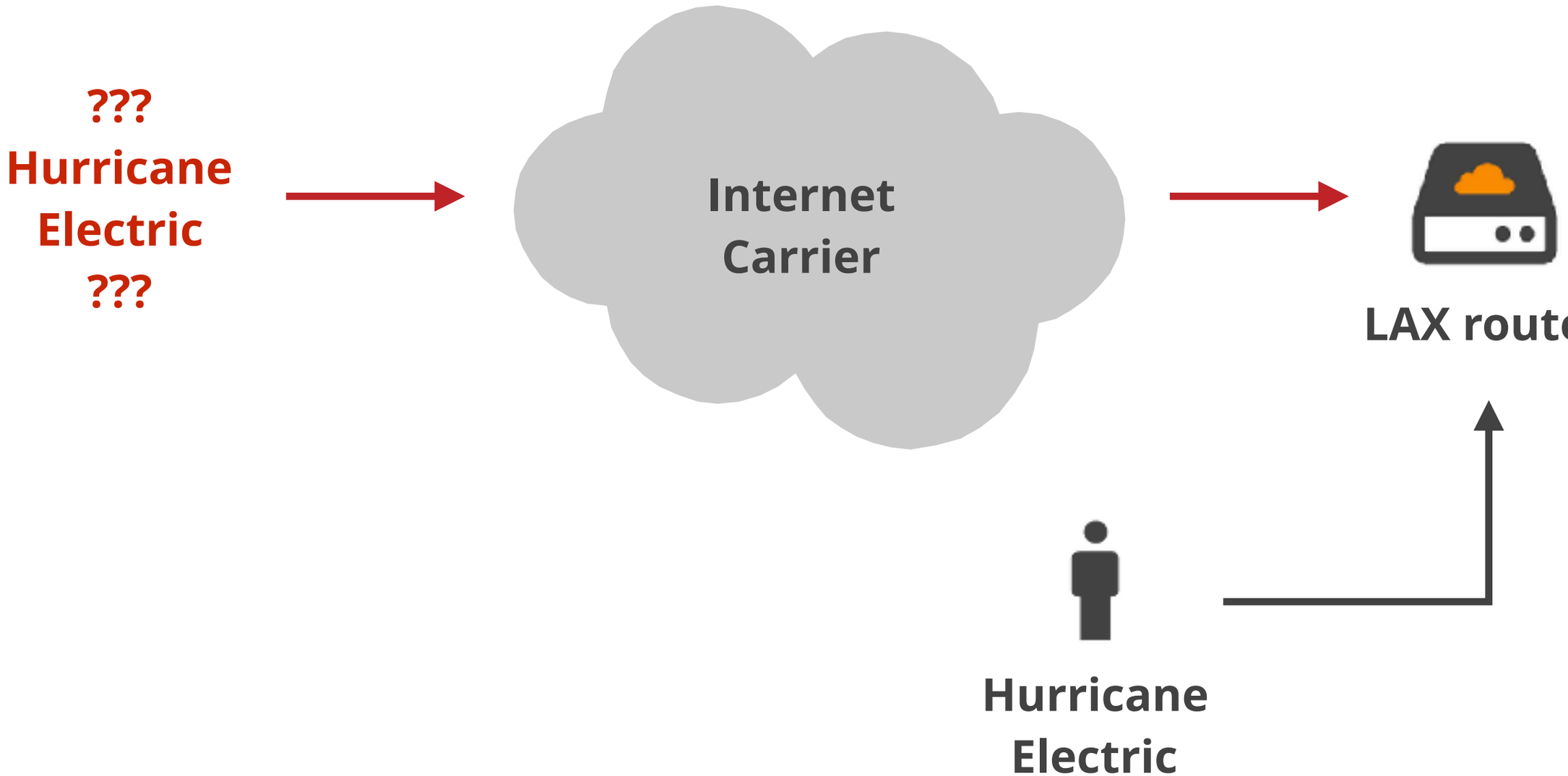
The Code

Our BPF Tools are now available on the CloudFlare Github:
https://github.com/cloudflare/bpftools

For installation instructions review the README, but typing `make` should do most of the work:

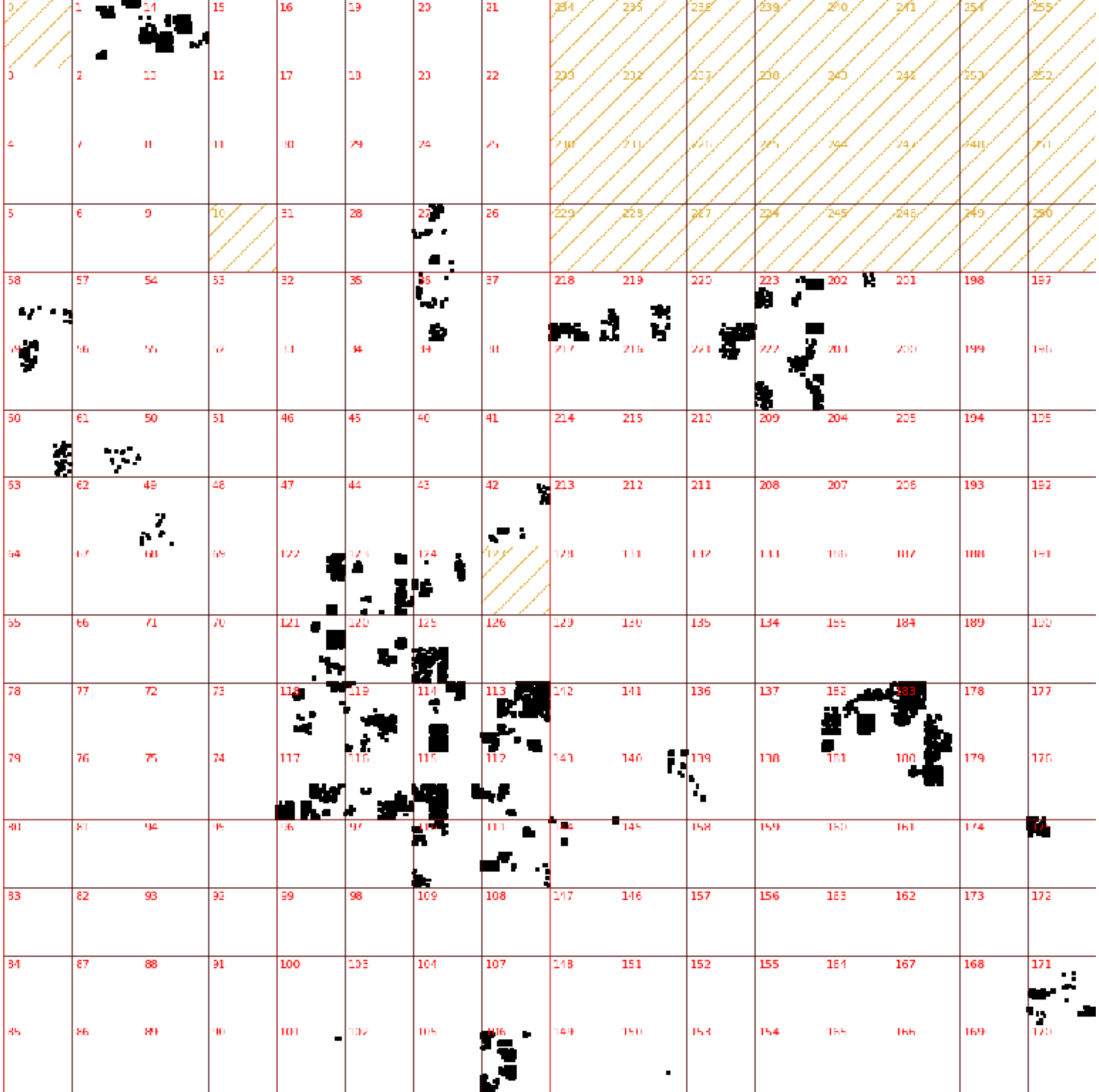
```
$ git clone https://github.com/cloudflare/bpftools.git
$ cd bpftools
$ make
```

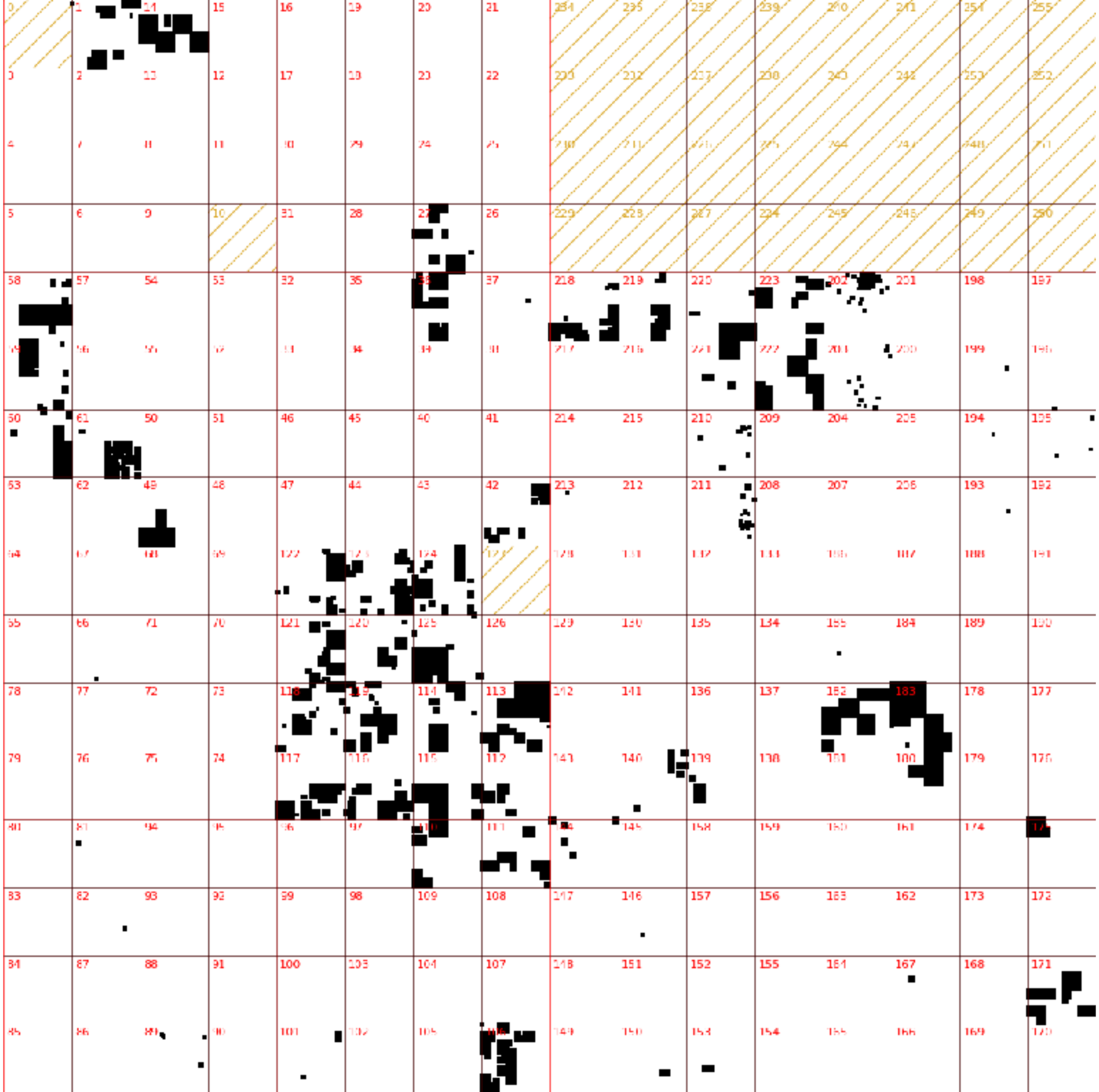
Source IP addresses

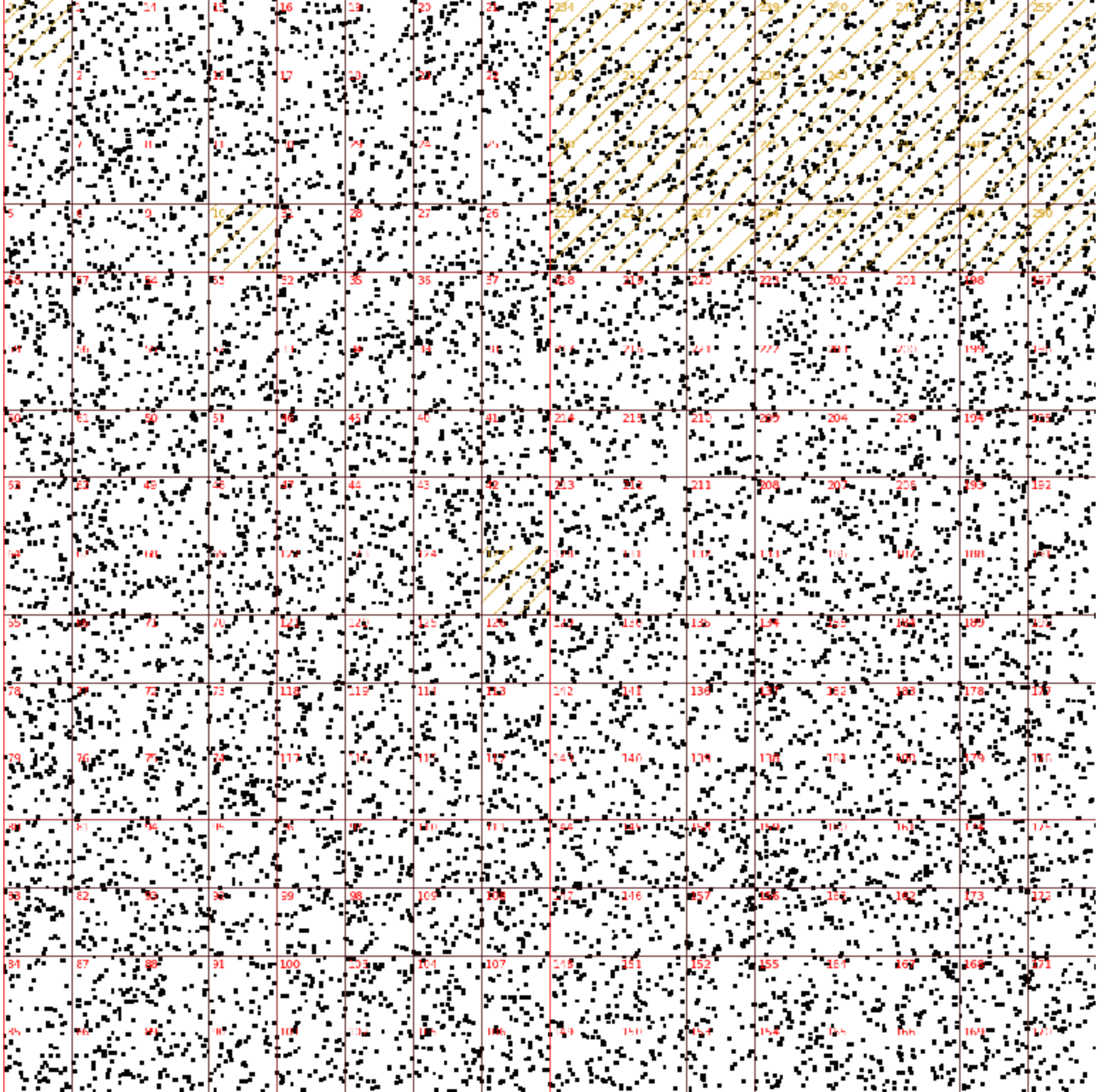


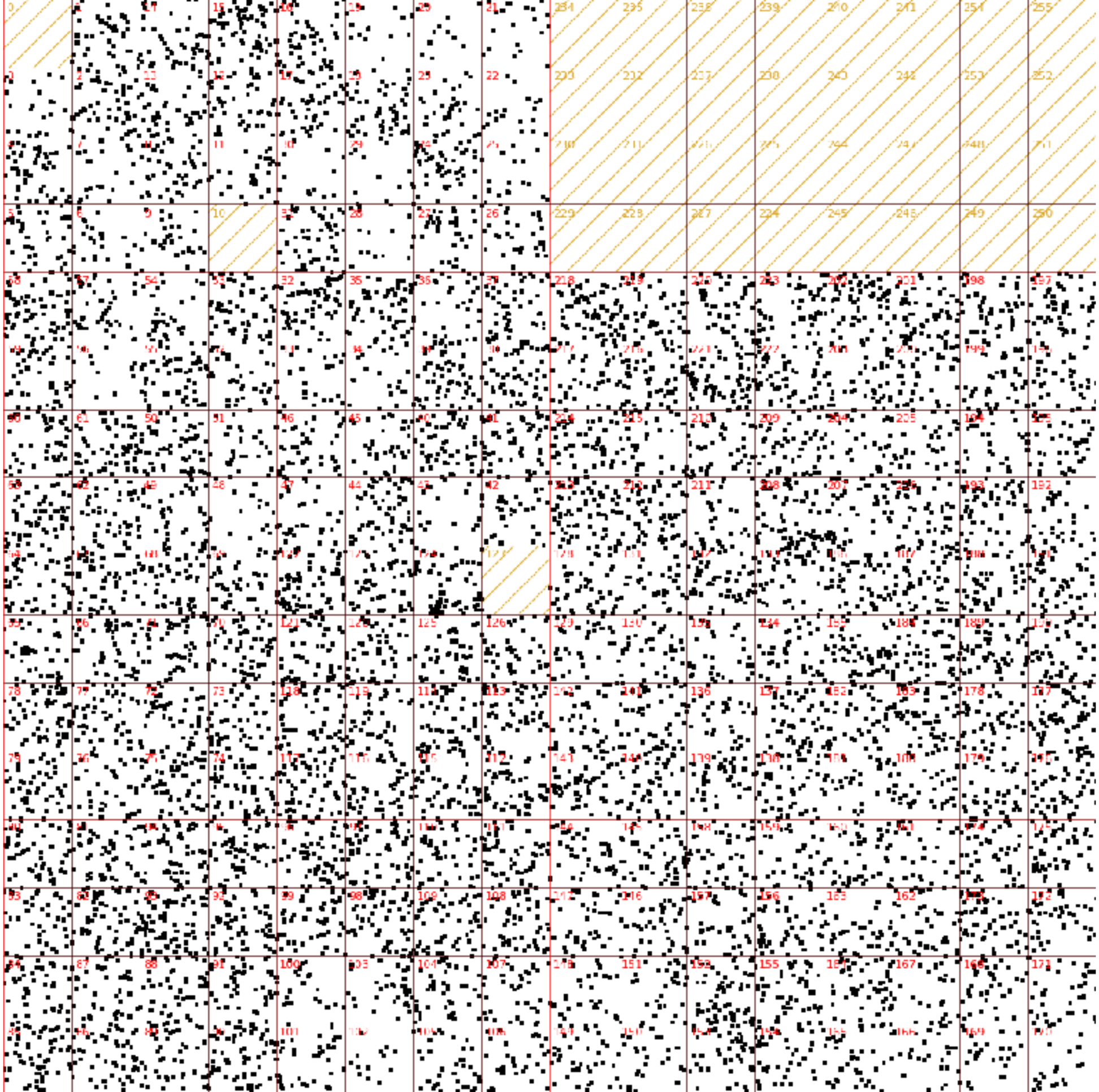
MAP OF THE INTERNET

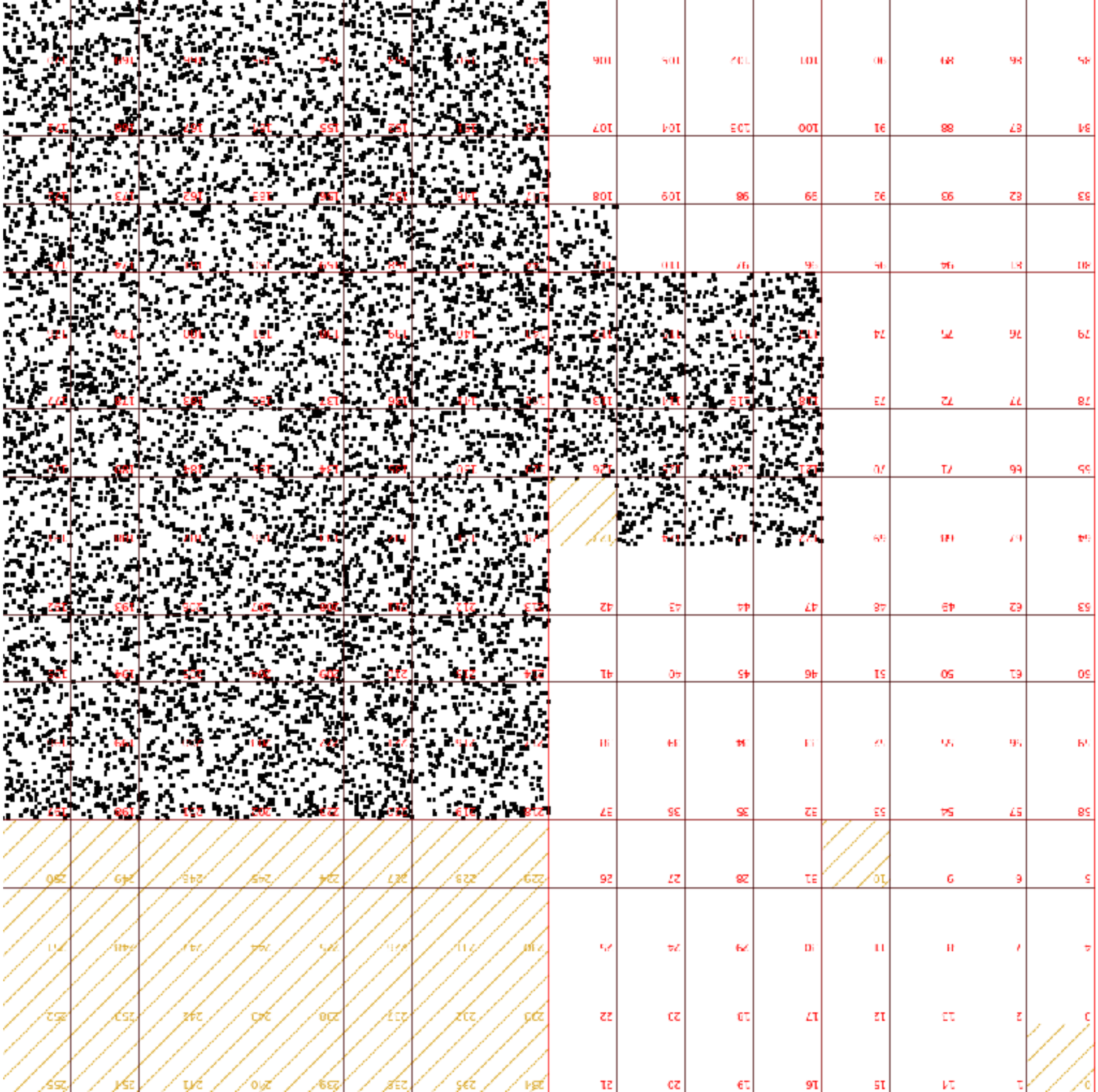
THE IPv4 SPACE, 2006

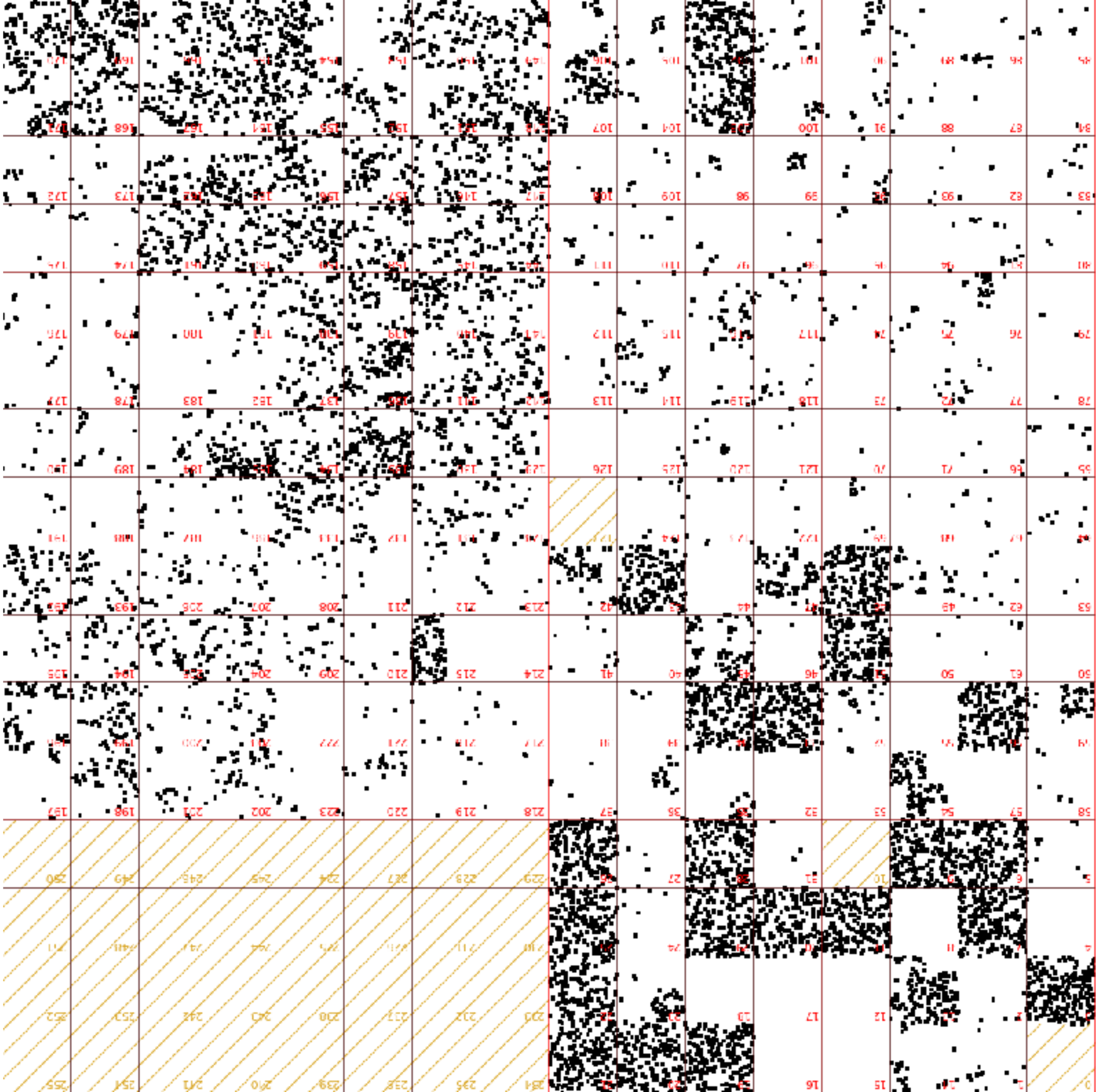










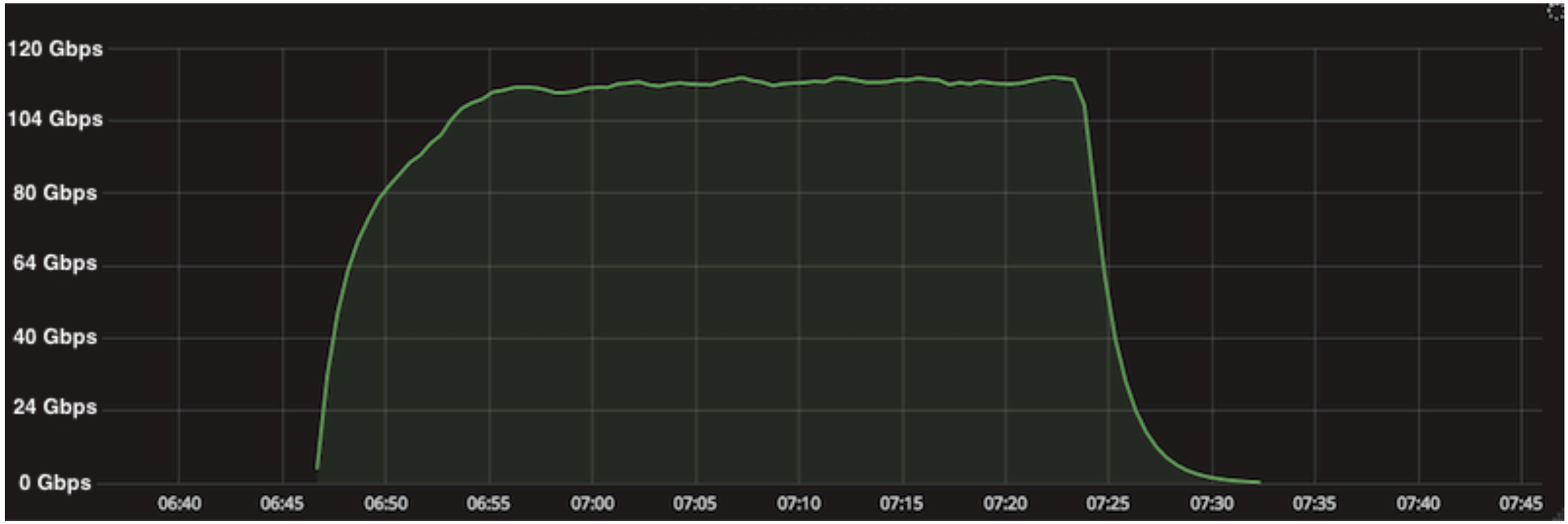


IP Spoofing

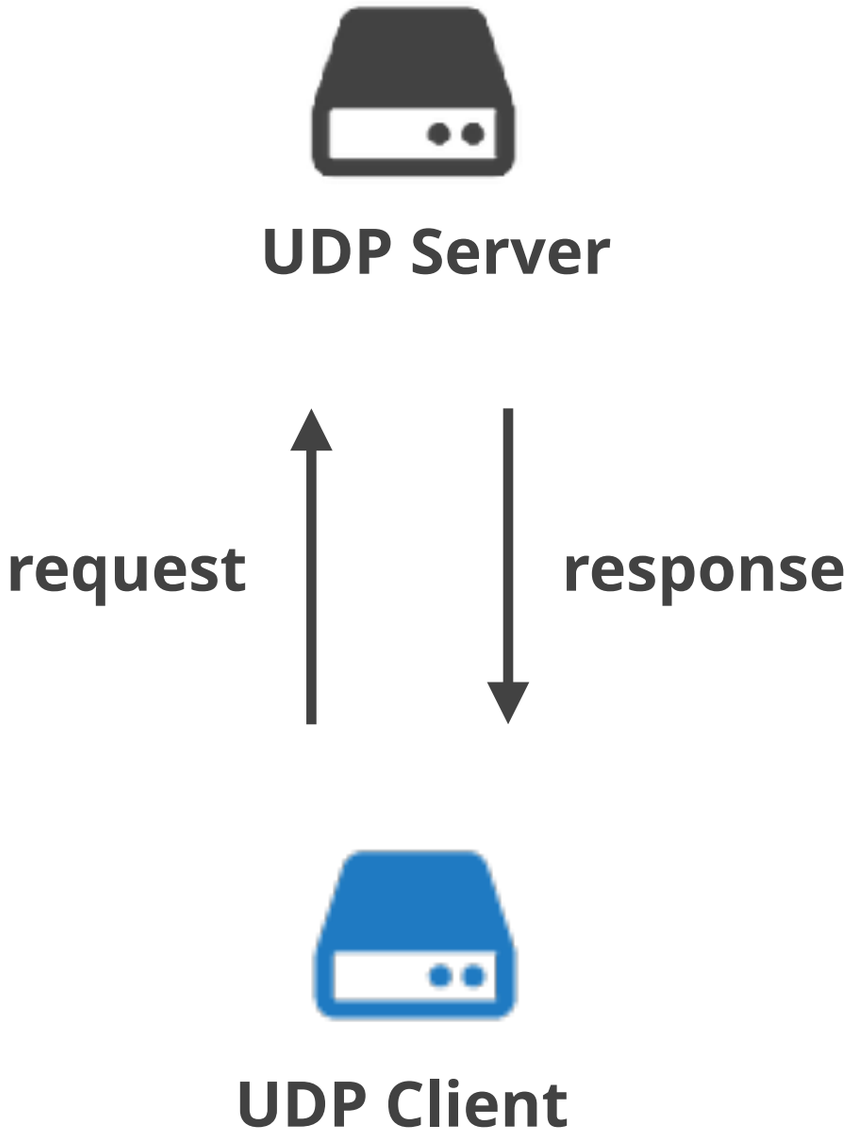
1. Tracing back is impossible

2. Allows sophisticated attacks

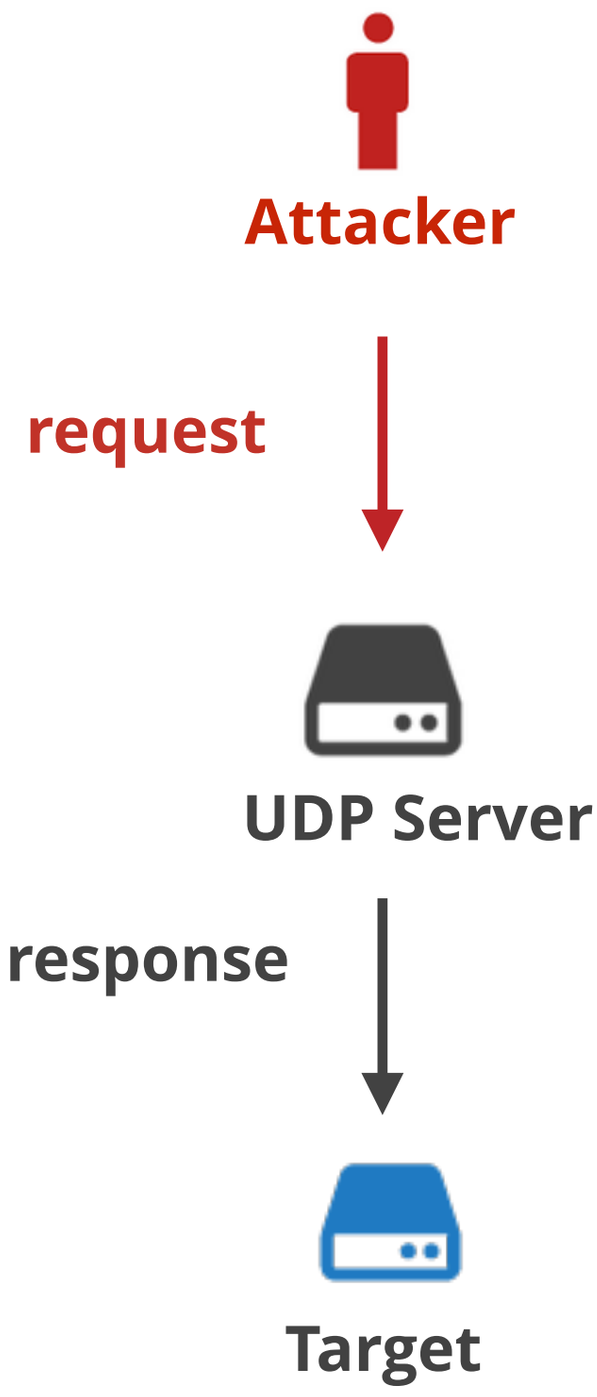
Sophistication



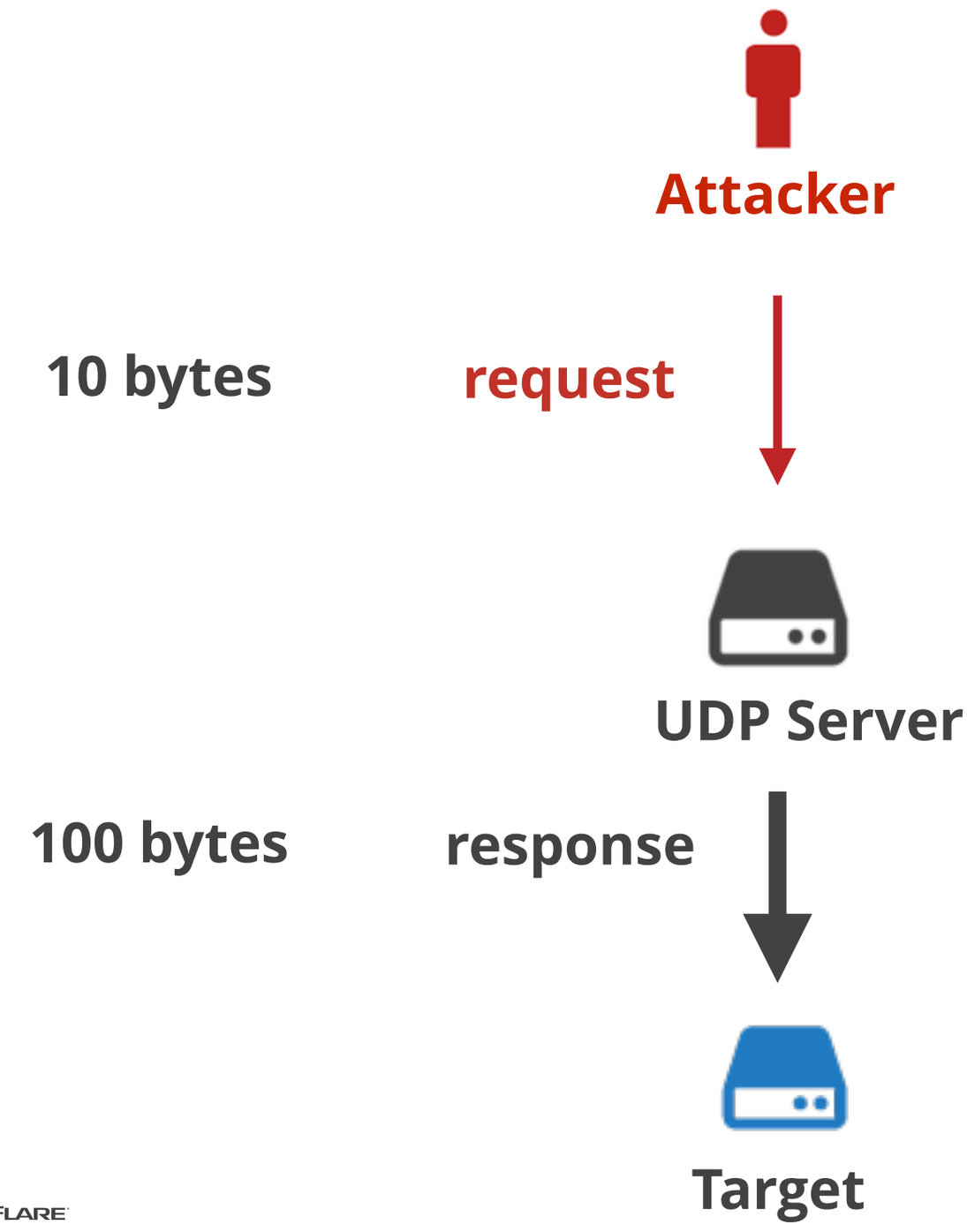
1. UDP request-response



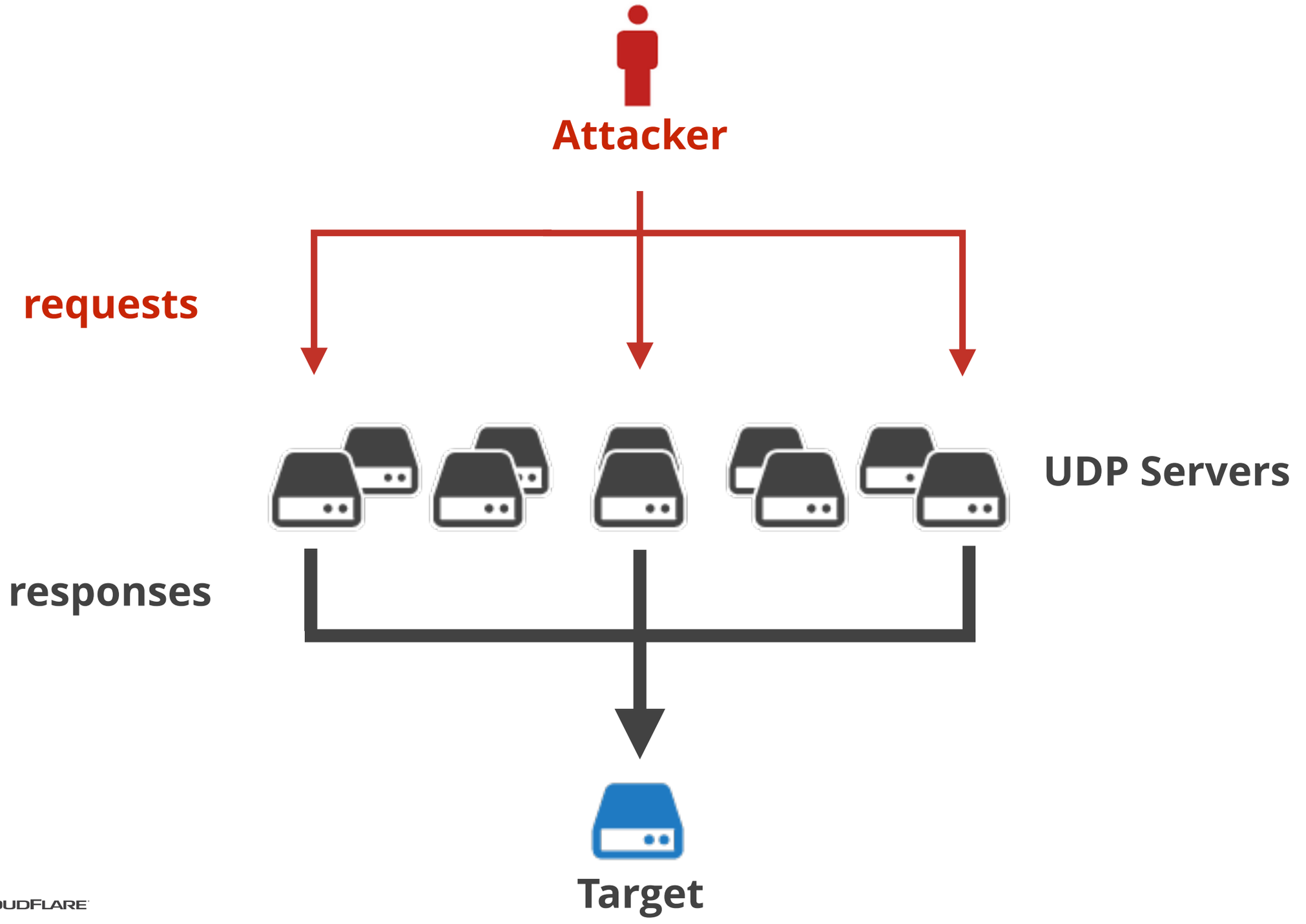
1. Amplification



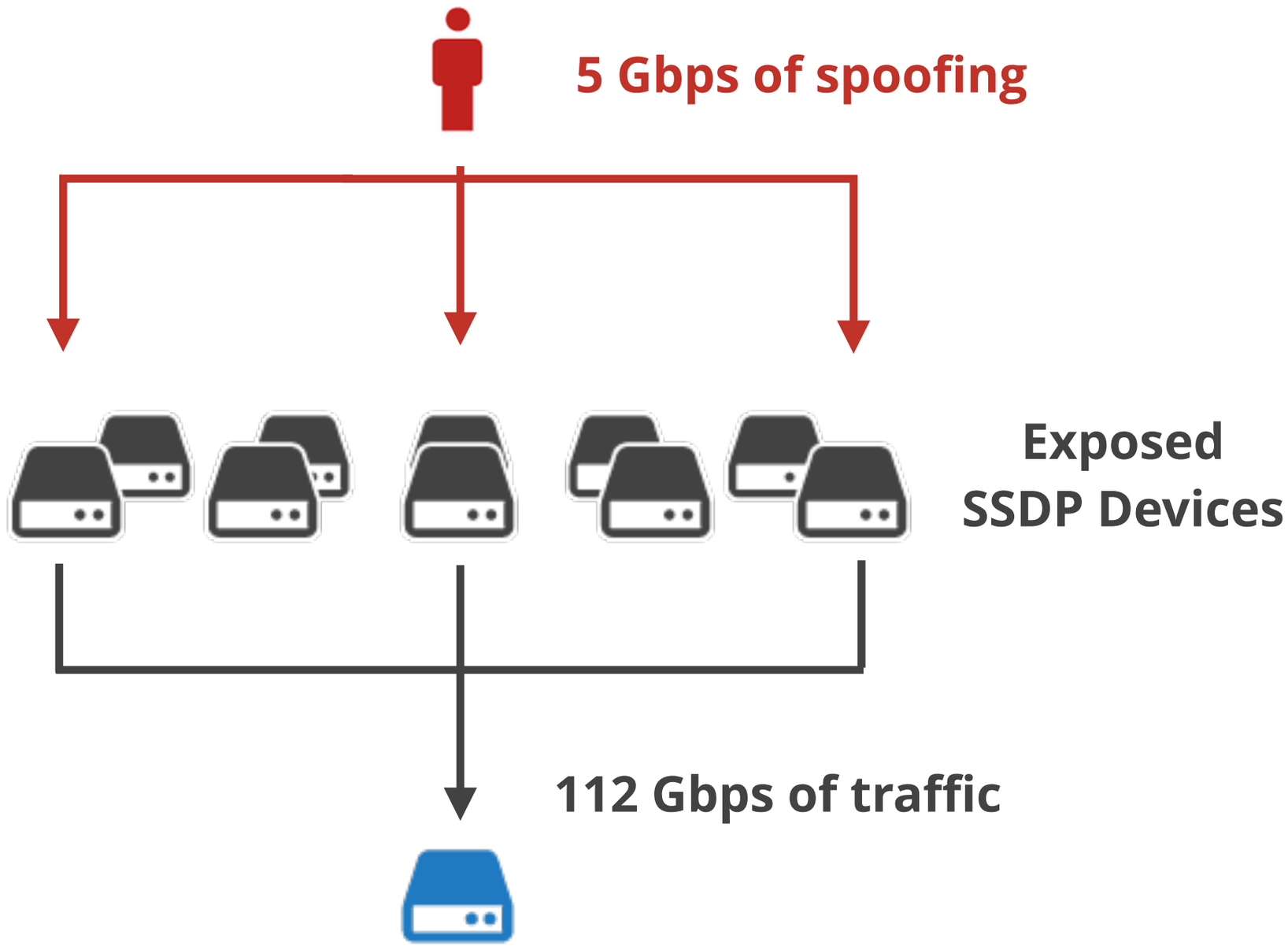
1. Amplification factor



1. Scale up!



June 2017: SSDP



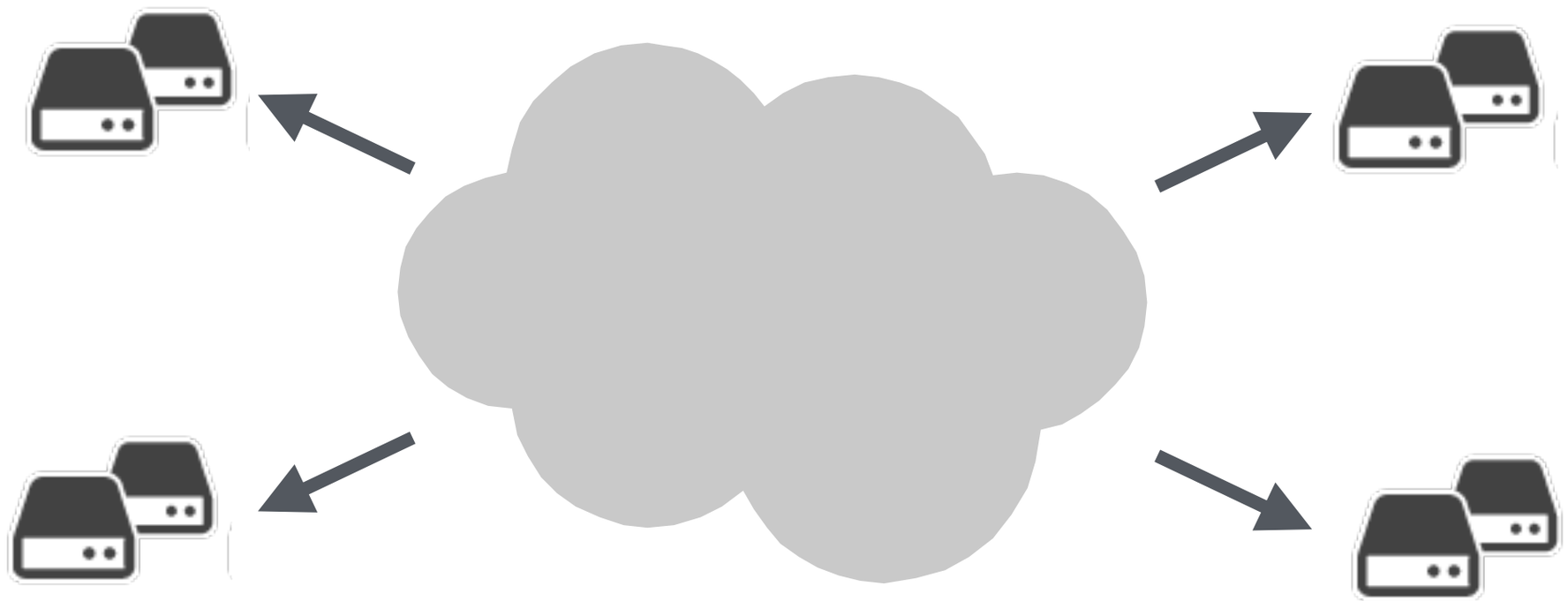
Amplification easy to block

- Easy to block on firewall
  - udp and src port 1900
- Nicely dispersed geographically
- But tracing is impossible

The only way to keep online is to absorb the attack

Receive and process

Centralisation



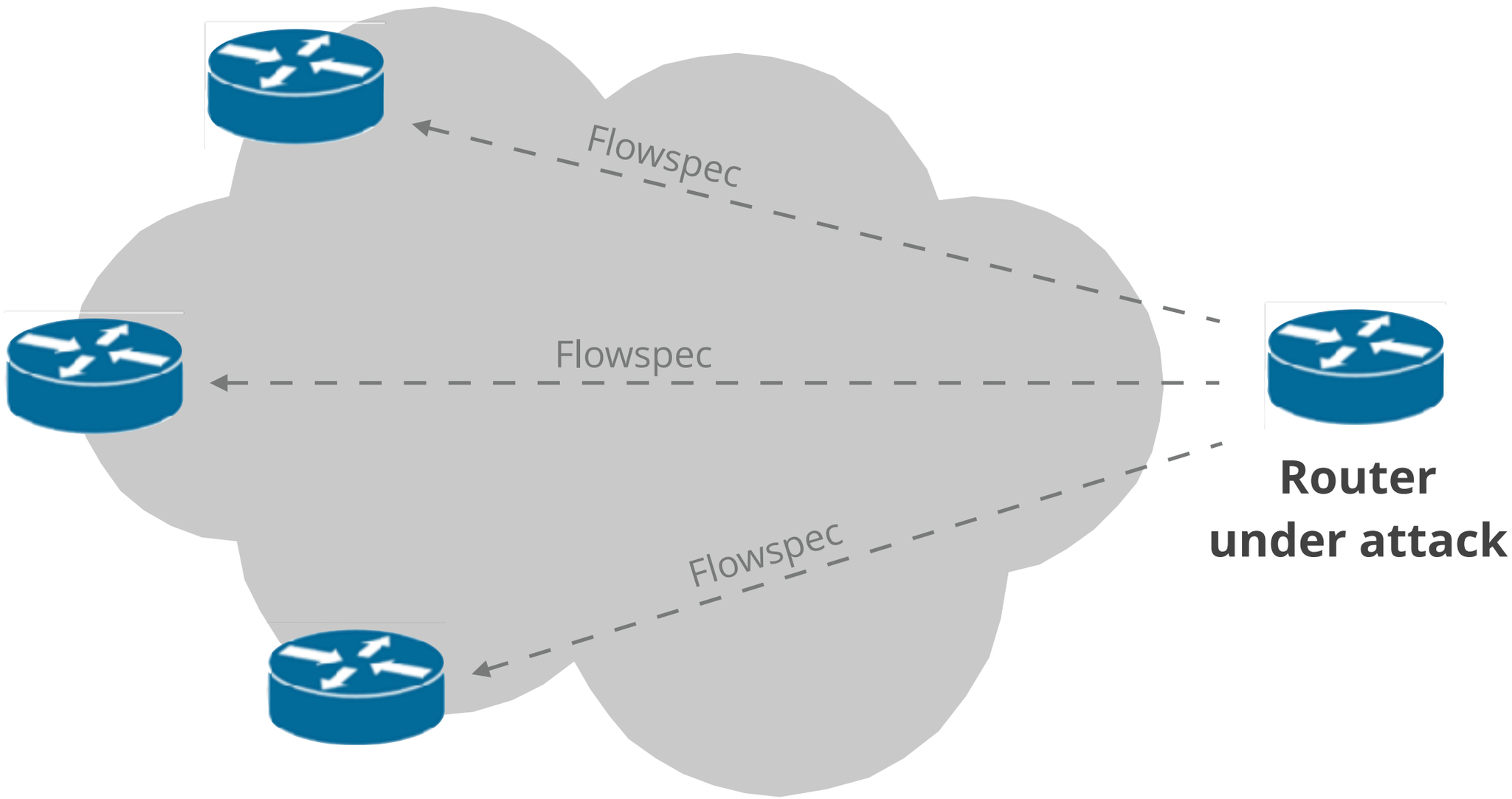
Solution

Technical solutions

- BGP Flowspec
- Prevent IP spoofing - BCP38
- Netflow

BGP Flowspec is awesome

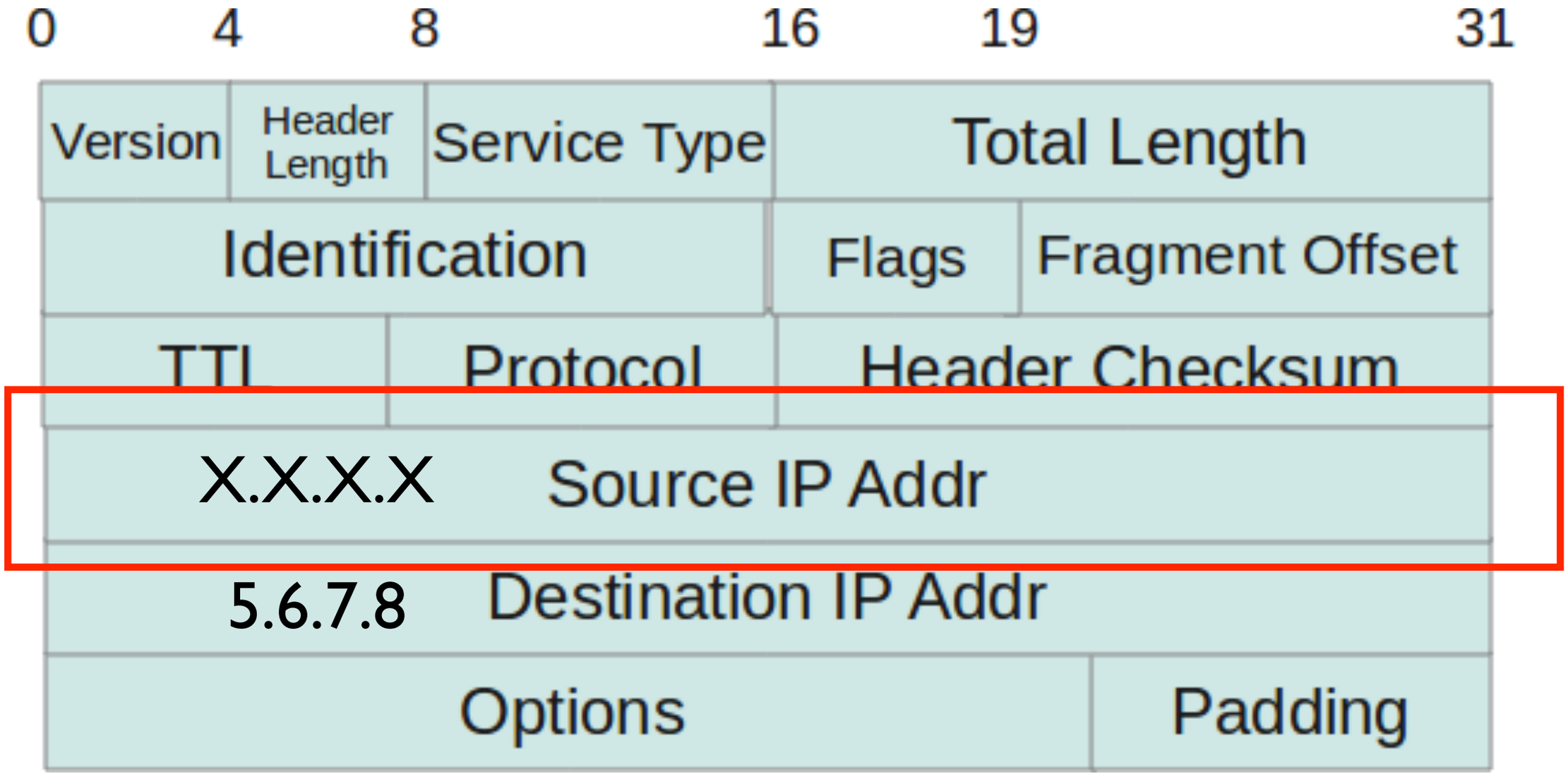
Flowspec



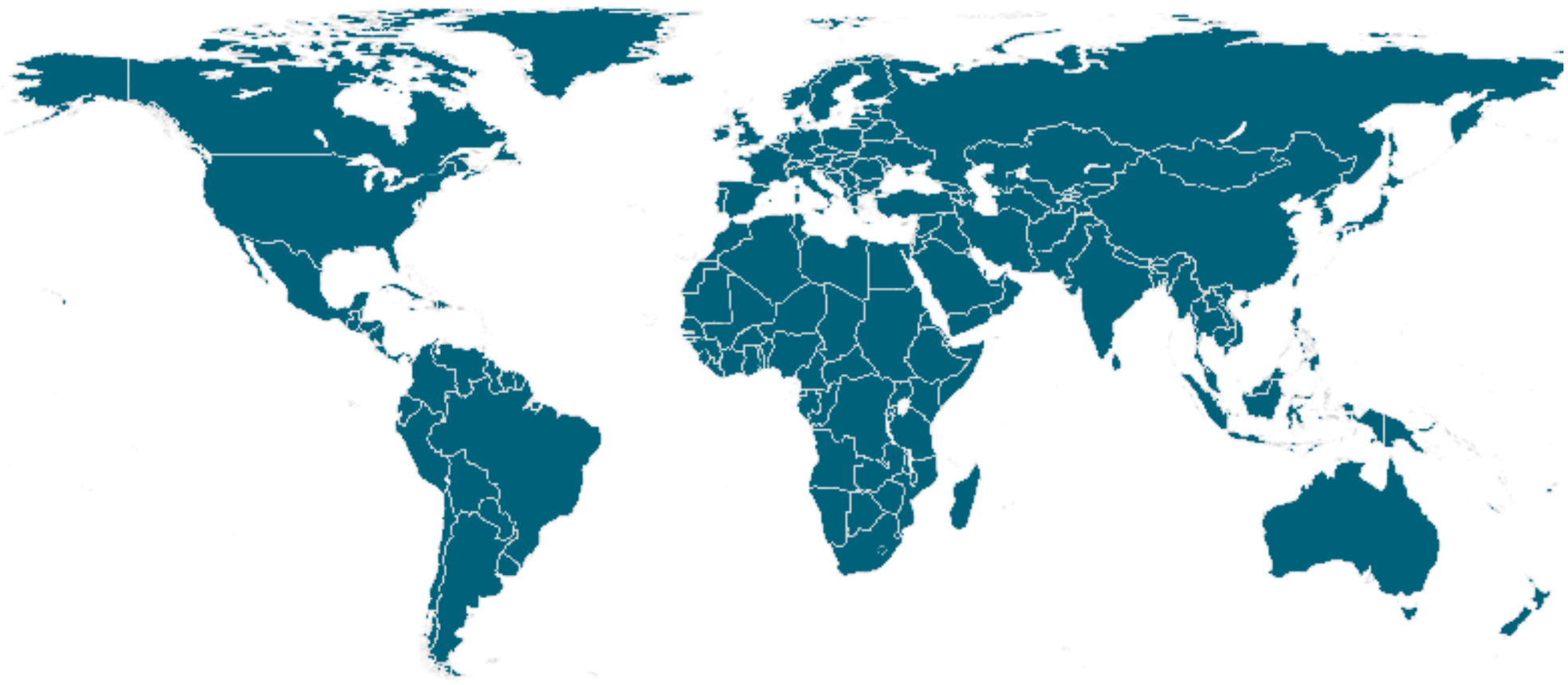
Adoption = nil

Prevent IP spoofing - BCP38

Failed! Live with it!



We're left with incompetent



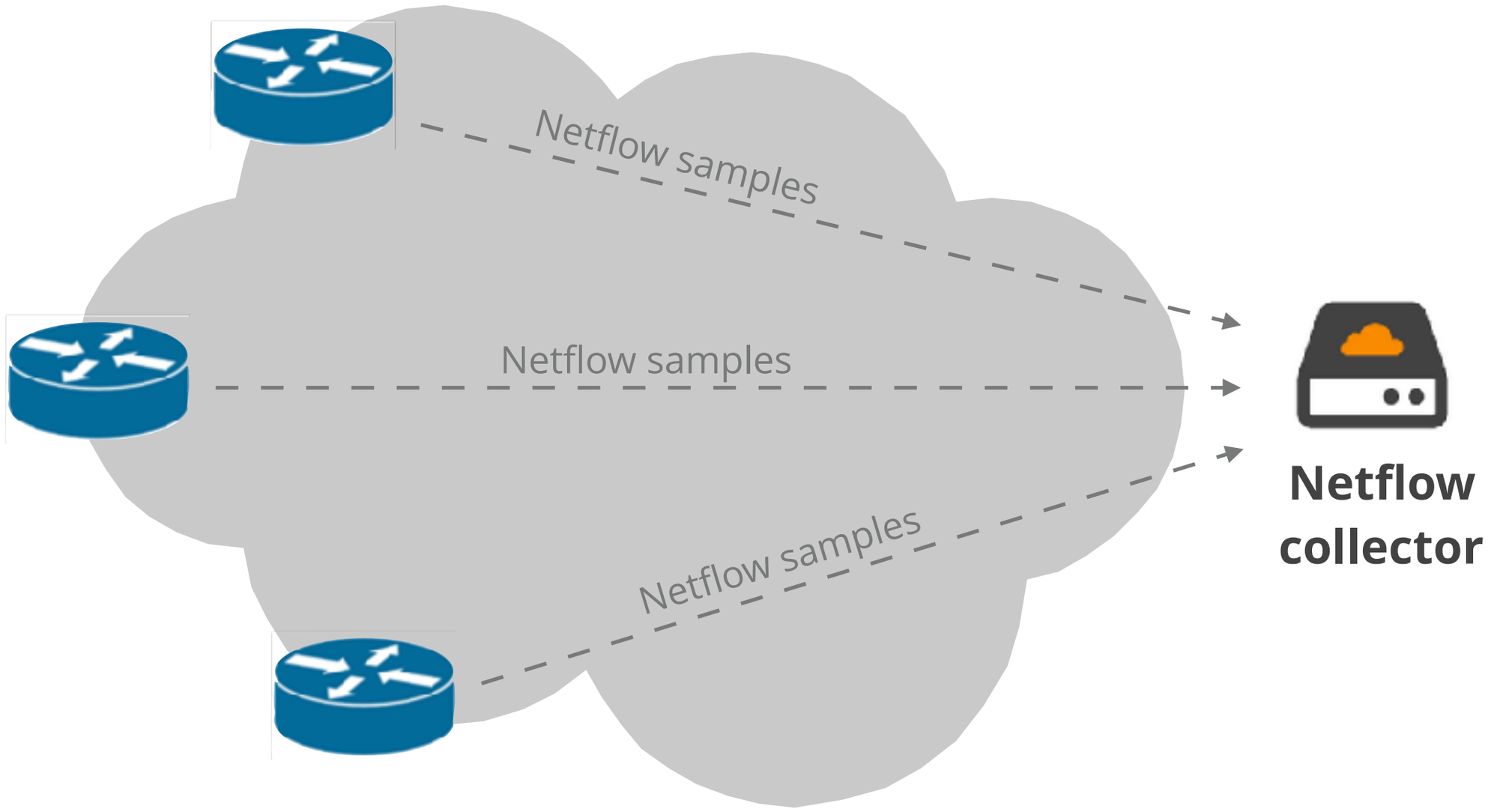
Don't solve the IP spoofing!

Solve the attribution!

Netflow


NETFLOW
ALL THE THINGS
memegenerator.net

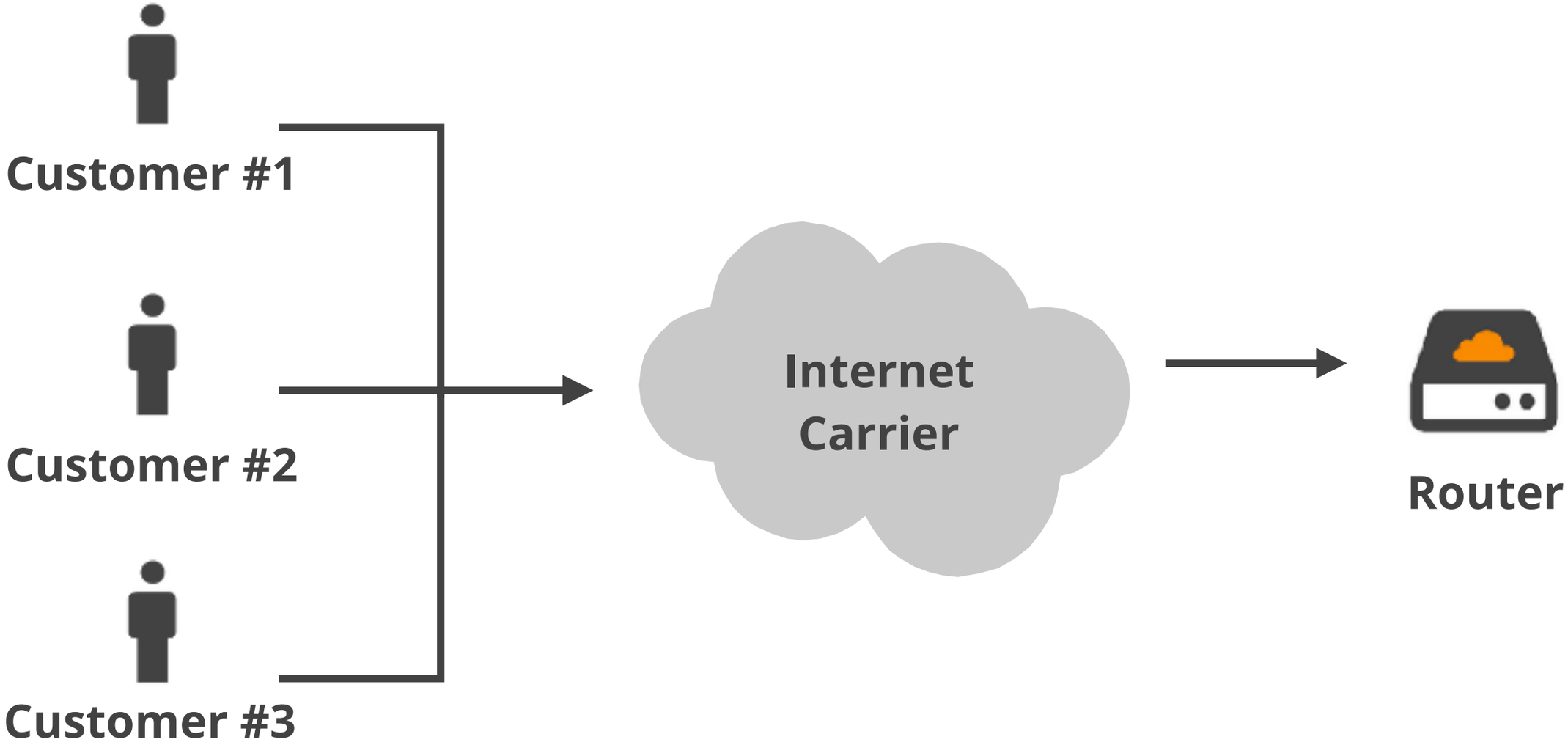
Netflow


Netflow samples
Netflow samples
Netflow samples
Netflow collector

Netflow

- Open source netflow toolchain is great
- Scales well
- To avoid privacy issues
  - Rotate logs often
  - Set high sampling rate - 1/64k connections

Internet Carriers



Netflow

```
(netops)# nfdump -M db/waw01:lhr01 -R . -n2 -t -300  -s dstip/packets "in if 731"
Top 2 Dst IP Addr ordered by packets:
Dst IP Addr     Flows(%)     Packets(%)      Bytes(%)         pps      bps   bpp
173.245.58.40  1.0 M(77.0)  17.6 G(75.8)    1.1 T(22.6)    59.0 M   30.7 G    65
173.245.59.15  54962( 4.0)  910.3 M( 3.9)   75.5 G( 1.5)    3.1 M    2.0 G    82

Summary: total flows: 1361108, total bytes: 5087980650496, total packets:
23271079936, avg bps: 135599480319, avg pps: 77524526, avg bpp: 218
Total flows processed: 2457140, Blocks skipped: 0, Bytes read: 177251772
Sys: 0.210s flows/second: 11700666.7 Wall: 0.210s flows/second: 11654603.2
```

Recap

- BGP flowspec firewall
  - A stop gap, useful for amplification attacks
- Prevent IP spoofing - BCP38
  - The root of all evil, but unfixable in short time
- Netflow sampling
  - Required for attack attribution
  - Supported by some competent ISP's / tier 1
  - Still not supported by majority!

Attribution allows informed discussion

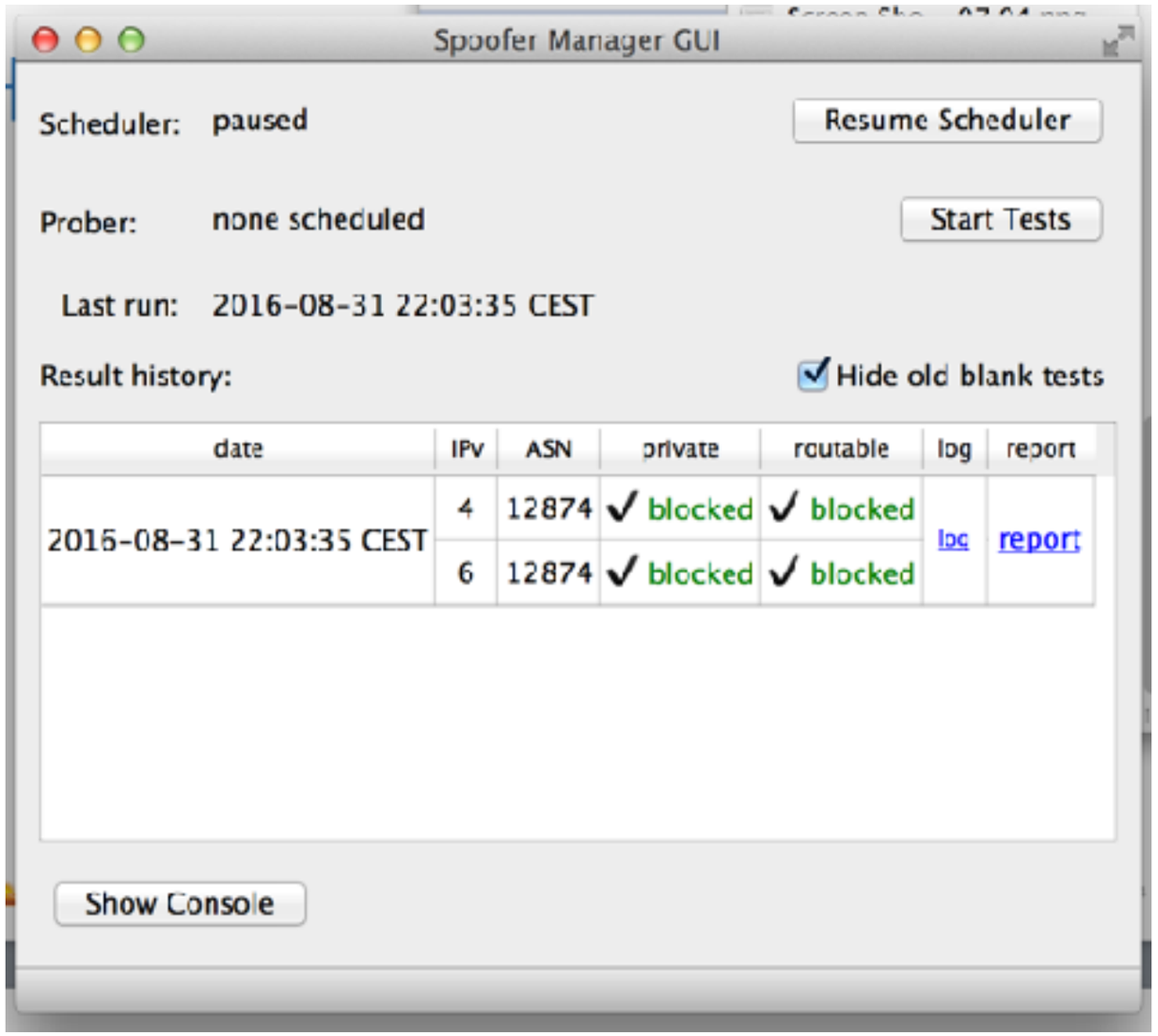
The internet will be better for everyone.

marek@cloudflare.com

How to help?

Help: report IP spoofing

- From spoofer.caida.org



Help: close NTP and DNS

- Scan your network for open NTP and DNS servers
- http://openntpproject.org/
- http://openresolverproject.org
- http://www.team-cymru.org/Open-Resolver-Challenge.html
- https://www.shodan.io/

Help: press for attribution

- When under attack
- Collect evidence
- Ask where the traffic came from!

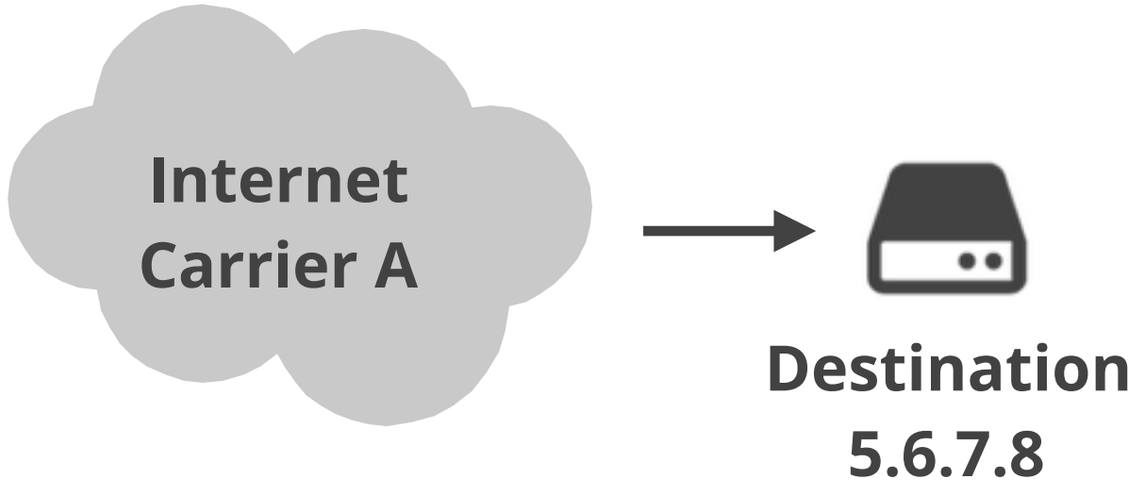
Is amplification in decline?

Is amplification in decline?

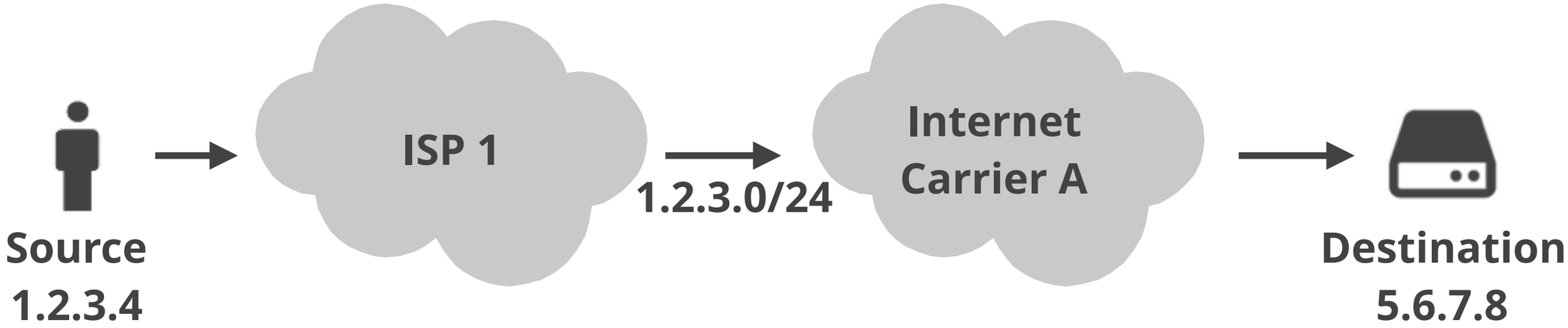
- Very easy to block on firewall
  - udp and src port 123 == NTP attack
  - udp and src port 53 == DNS attack
- DDoS mitigation vendors have FAT pipes
- Amplification is bouncing off *real* servers
- Therefore geographically distributed
- Not effective against anycast

Why IP Filtering must be on the edge

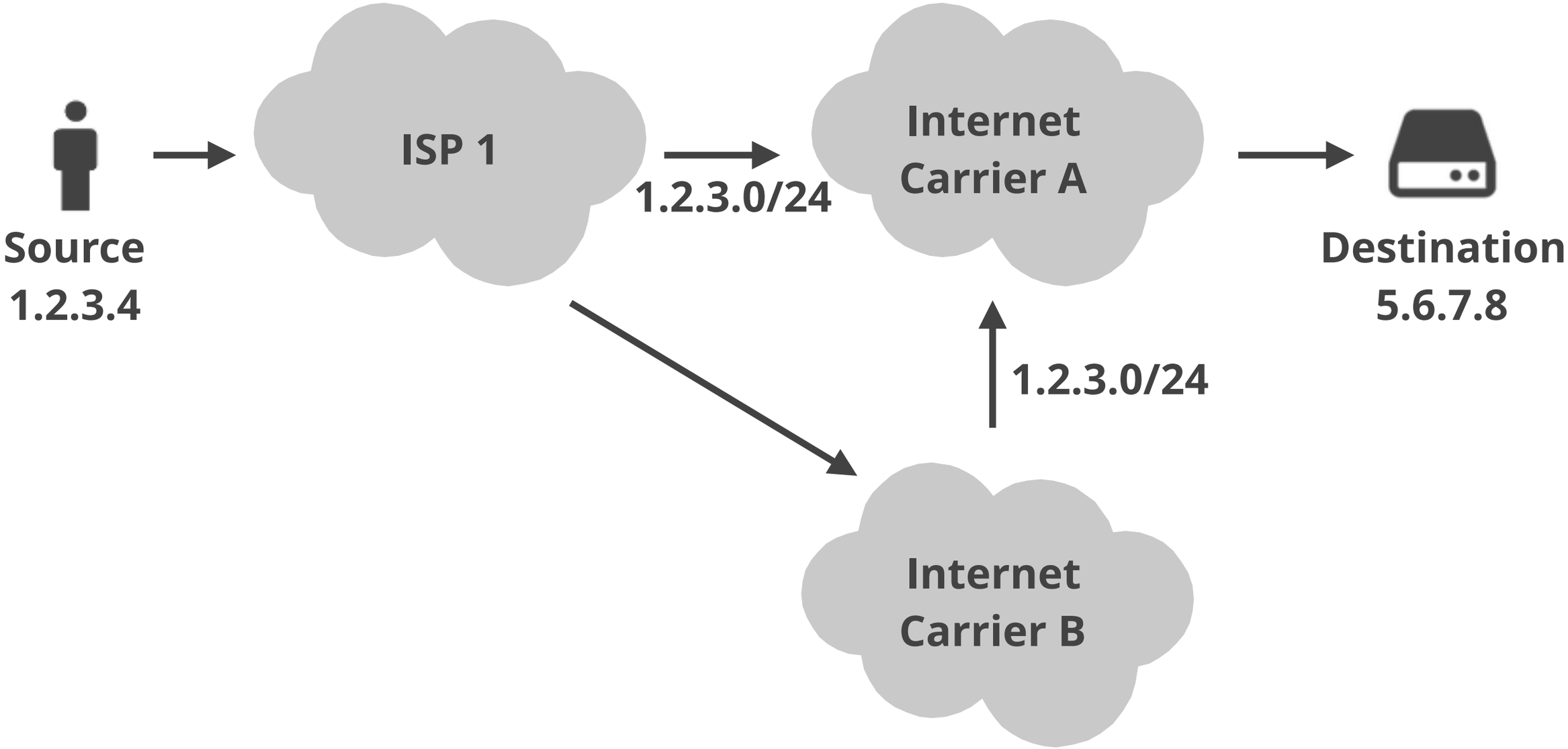
Filtering is hard



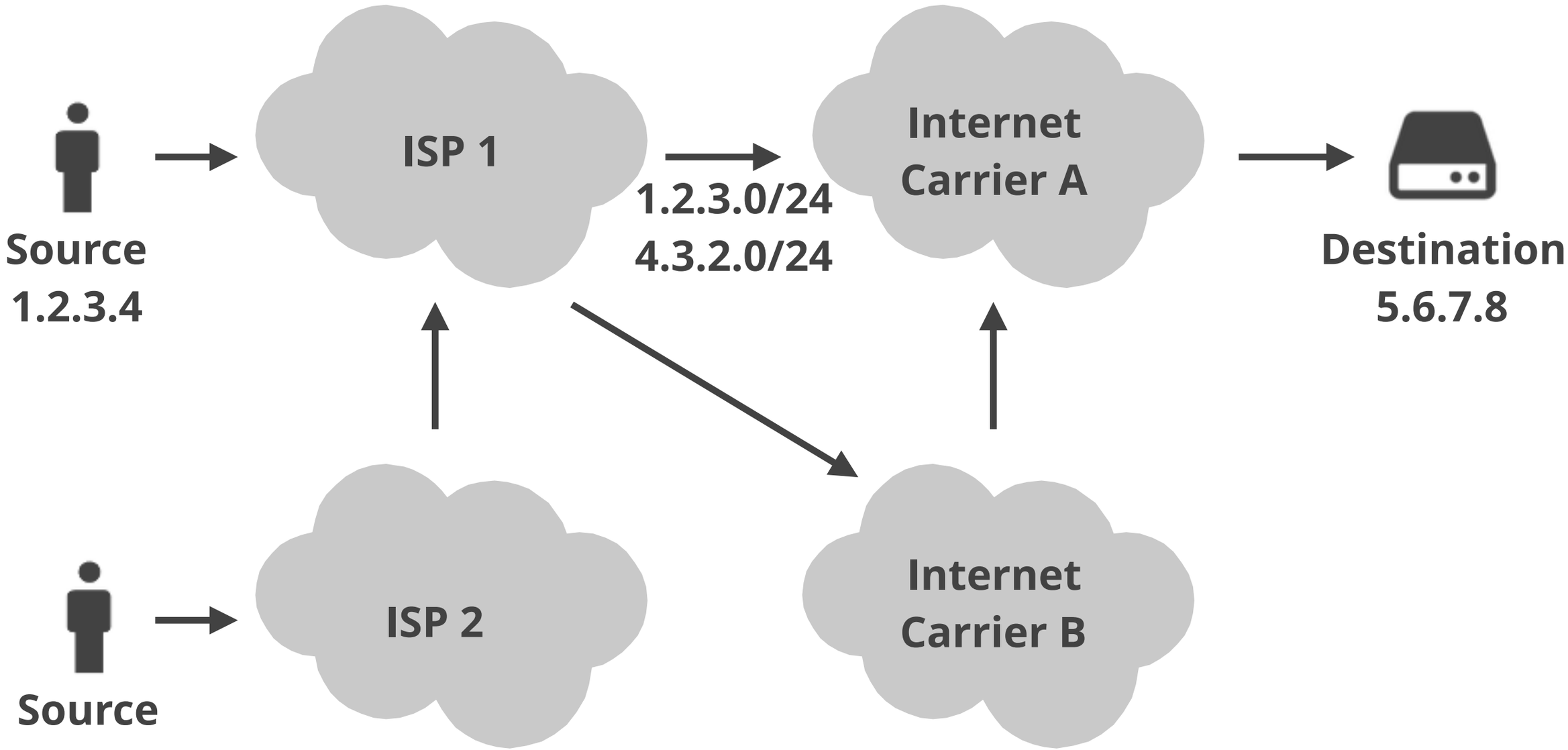
Filtering is hard



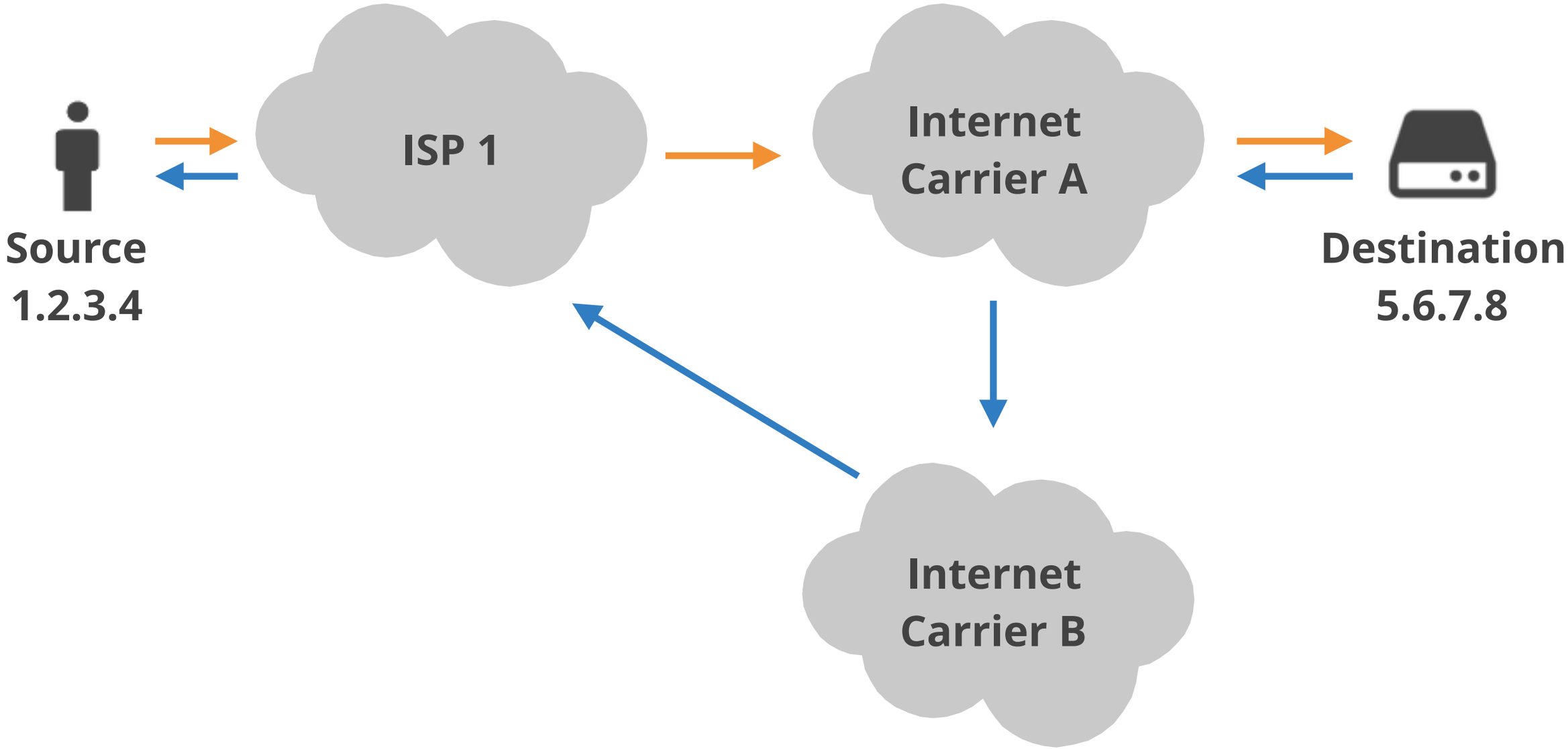
Filtering is hard



Filtering is hard



Internet is asymmetric



Filter close to the source