Improving PF's performance and reliability

Lessons learned from bashing pf with DDoS attacks and other malicious traffic.

By Kajetan Staszkiewicz, Senior Network Enginneer @ InnoGames

Let me introduce myself

About me

- Created a neighbourhood computer network in 1999. 115kb/s for 15 people! Wohoo!
- Worked for various local ISPs in Kraków, Poland.
- Some other company with not that big network. Sadly.
- Currently Директор Интернета at InnoGames GmbH.

1. How is FreeBSD used at IG?
2. What is a DDoS Attack?
3. How does DDoS influence pf?
4. How to make pf more resilient?
5. What performance can we expect?
6. Further work on pf.
7. Conclusions.

How is FreeBSD used at IG?

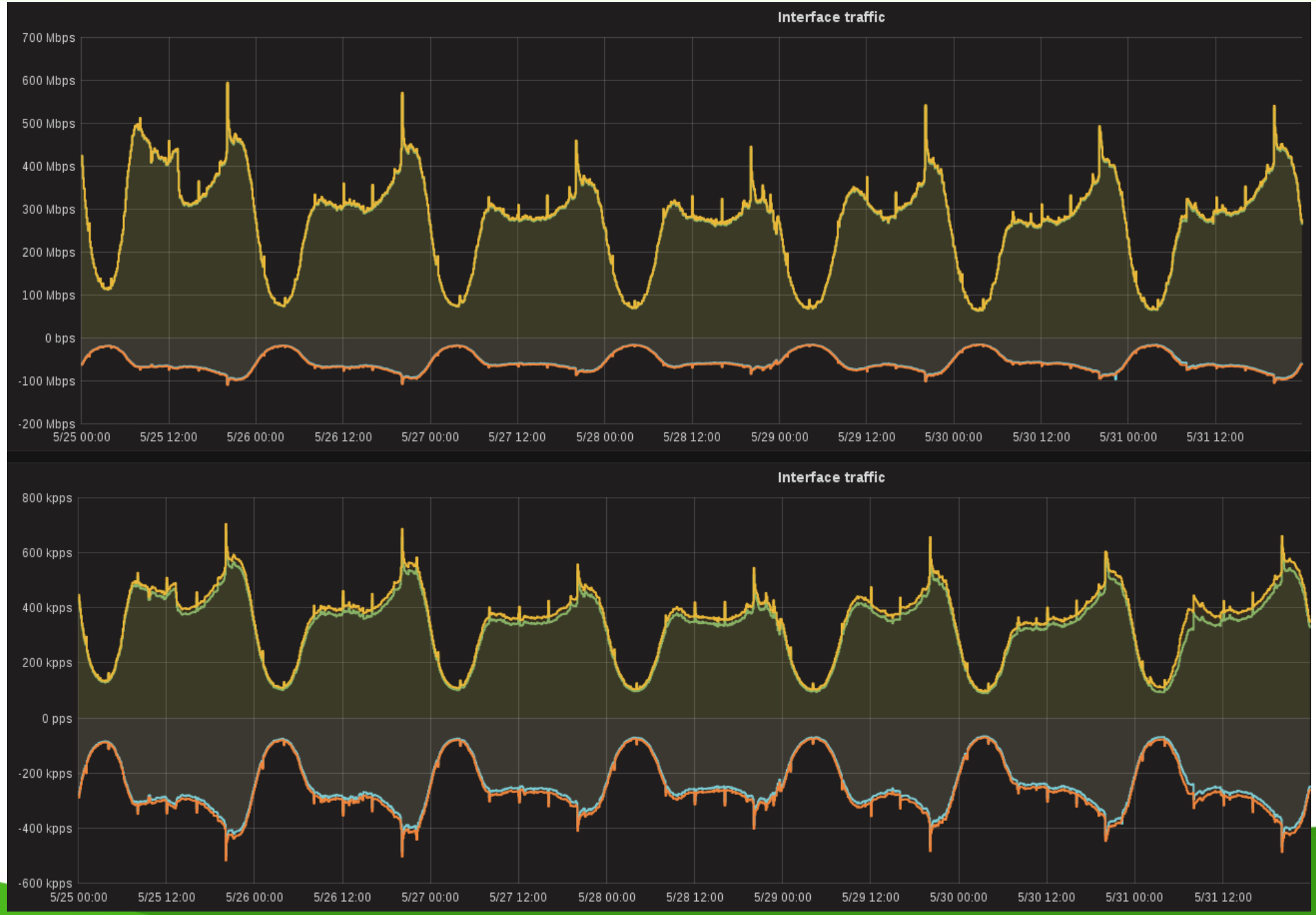
Load Balancers

- 36 HWLBs running FreeBSD 10.1 and pf.
- Kernel with some custom patches on pf.
- 4×1Gb or 2×10Gb NICs2.
- 10Gb NICs support multiple queues (no msi-x on bce on FreeBSD)

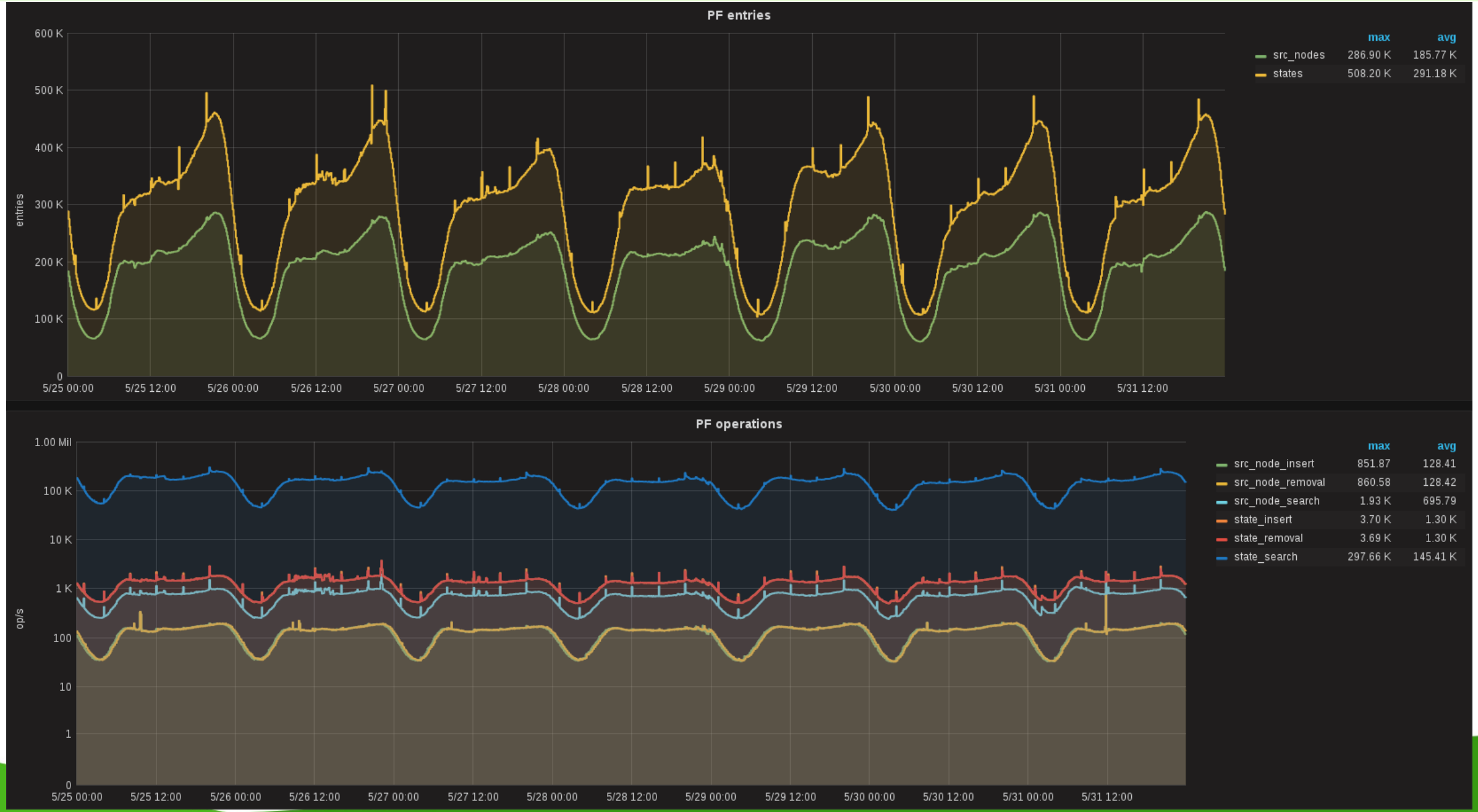
Load Balancers

- Route-to target, so we could use DSR in future (if we really want).
- No pfsync (does not work with route-to).
- Pf also used to restrict access to the Internet.
- HWLB sharing:
  - Big games – 2×active + 1×backup HWLB per game.
  - Smaller projects: 1×active + 1×backup per multiple VLANs.

How is FreeBSD used at IG?



How is FreeBSD used at IG?



Why FreeBSD?

- Historical reasons – was already there.
- LVS from Linux was told to be “slow” (whatever it means).
- With a bit of tuning it does the job.

Routers

- 6 Routers, 2× per datacenter (3 datacenters).
- Handling internal VLANs.
- Routing to other datacenters with IPsec.
- Transport mode + GRE + OSPF.

What is a DDoS attack?

What is a DDoS attack?

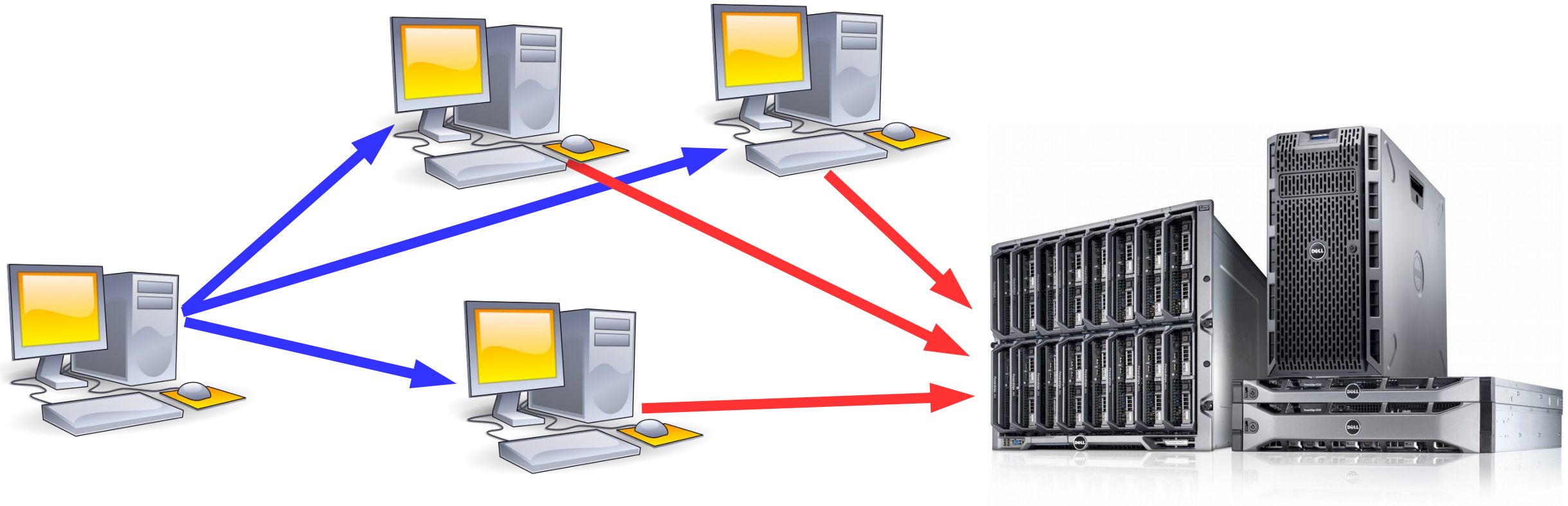
- Not really “hacking”.
- No data being stolen.
- Legitimate customers denied access to service.
  - ➔ Block other players from defending their villages. Yes, really.

What is a DDoS attack?

Direct attack:

- Attack force (pps, rps) limited by computer and Internet connection of attacker.
- Source Datacenter might filter attack.

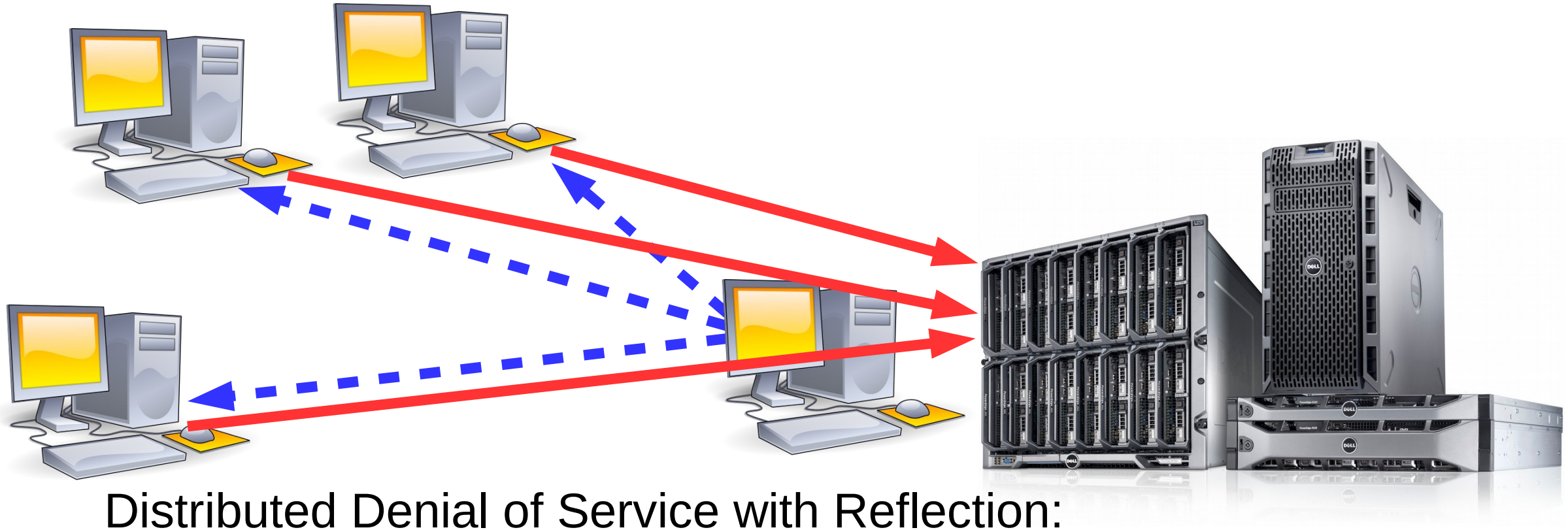
What is a DDoS attack?



Distributed Denial of Service:

- Attack force (pps, rps) is multiplied.
- Attacker needs control over attacking machines.

What is a DDoS attack?



Distributed Denial of Service with Reflection:

- Attack force (pps, rps) is multiplied.
- Innocent, uninfected machines perform attack for the attacker.
- Attack sources think that we attack them – extra paperwork.

How is it possible?

- Internet providers and Datacenters don't always verify source addresses of packets from their networks.
- For reflection attack connectionless, UDP-based services (DNS, SNMP, NTP) are used:
  - DNS – up to 54×, 179× with DNSSEC
  - NTP – up to 556×
  - 300Gb/s attack can be performed

What is targeted?

- UDP-based reflection: volumetric attack saturating network links.
- TCP SYN flood: web, mail servers, loadbalancers, firewalls
  - ➔ resource depletion.
- ICMP traffic: no idea, probably side effect.
- Other possibilities:
  - ➔ TCP fragments?

SYN Flood

- For each SYN pf creates a state (rather 2) and source node entry.
- State and source node table resource shortage.
- Huge amount of state creation and state removal events.
- Each state lives for some time and occupies resources.

UDP Flood

- Saturation of links.
- No state creation for TCP services.
- But still causes load on firewall.

How do we protect ourselves?

- Bigger links (10Gb/s).
- Dynamic QoS on attack detection (see my other presentation).
- Permanent QoS for UDP floods.
- External attack mitigation service (you can't live without them).
- Improving pf.

How a DDoS influences pf?

DDoS attacks will probably kill your router

- Too many states are created, global state limit is reached.
- Amount of states created is double because of internal states.

```
pass  in quick on $pub route-to ($int <targets>) to $pub_ip
pass out quick on $int
```

- Src_nodes are also created and will hit limit.
- Performance drops with amount of states (does it?).

Bugs discovered thanks to DDoS

- O(n*m) algorithms in pf – total kernel freeze.
- Packets still being forwarded after hitting per-rule state limit.
- Network hiccups during pf_purge operation.
- Passing of traffic via route_to with empty target table.
- Source node searched 2 times.

How to make pf more resilient?

Reduce timers!

- 30% drop of state number after tcp.established=3600
- tcp.{first,opening} = 15 – probably still way to high
- Same for closing, finwait, closed.
  - Nobody complained and we got less states.

Limit damage!

- Limit amount of states per rule:
  - ✗ FreeBSD 9 – based idea.
  - ✓ Could we just allow to have many states on FreeBSD 10?
- Attack limited to a single game world.
  - ✗ Not applicable to every service.

Have states!

- It is faster to find a state than to check thousands of rules.
- Have pass on internal interfaces even with no blocking rules.

Don't have states!

- Syslog/logstash over UDP with many source ports.
- Pings from the Internet.
- Set skip will not really help for pf with global locking.

Use FreeBSD 10!

Performance is greatly improved.

What performance can we expect?

Base conditions - hardware

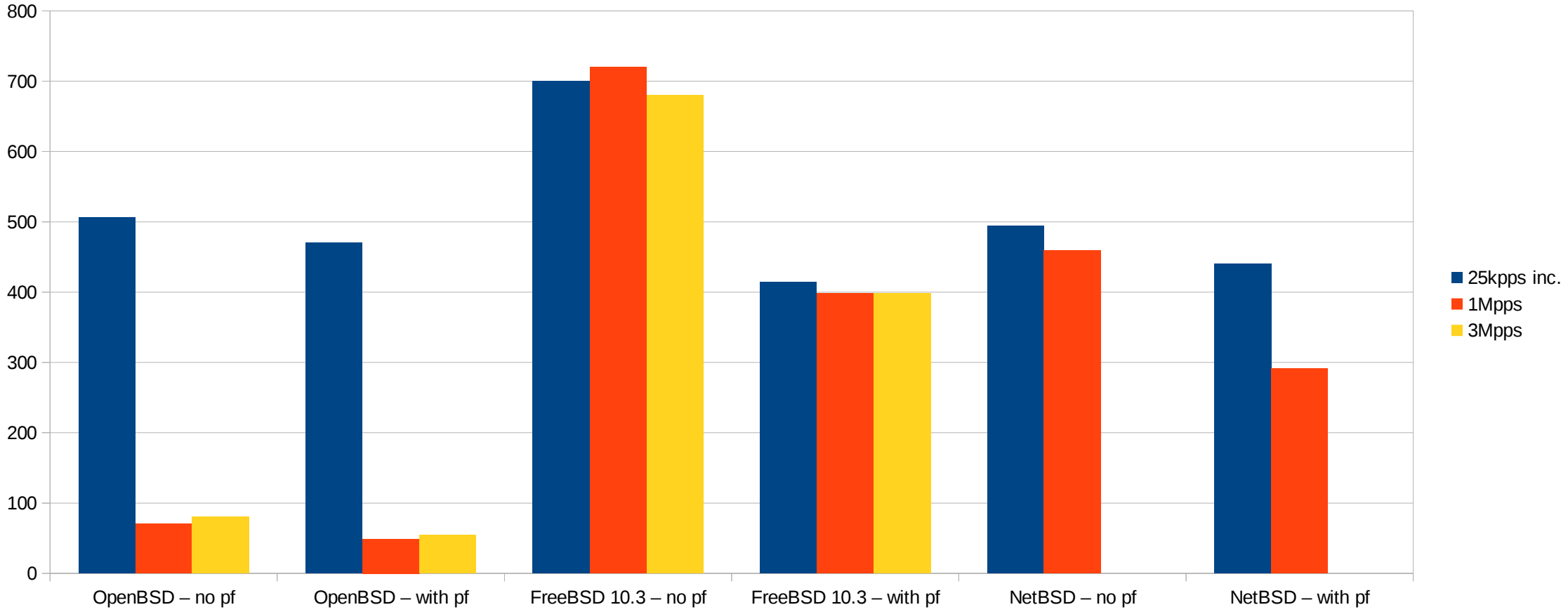
- 2× Dell PowerEdge R510 (DDoS-attacker and pf-victrim)
- 2× X5650 (12MiB L3 cache) @ 2.67GHz
- 6 cores per CPU
- HT disabled (slows things down)
- 128GiB memory
- 2-port Intel 82599 (up to 16 queues)
- 2-port Broadcom BCM5716 for management (set skip on)

Measurements

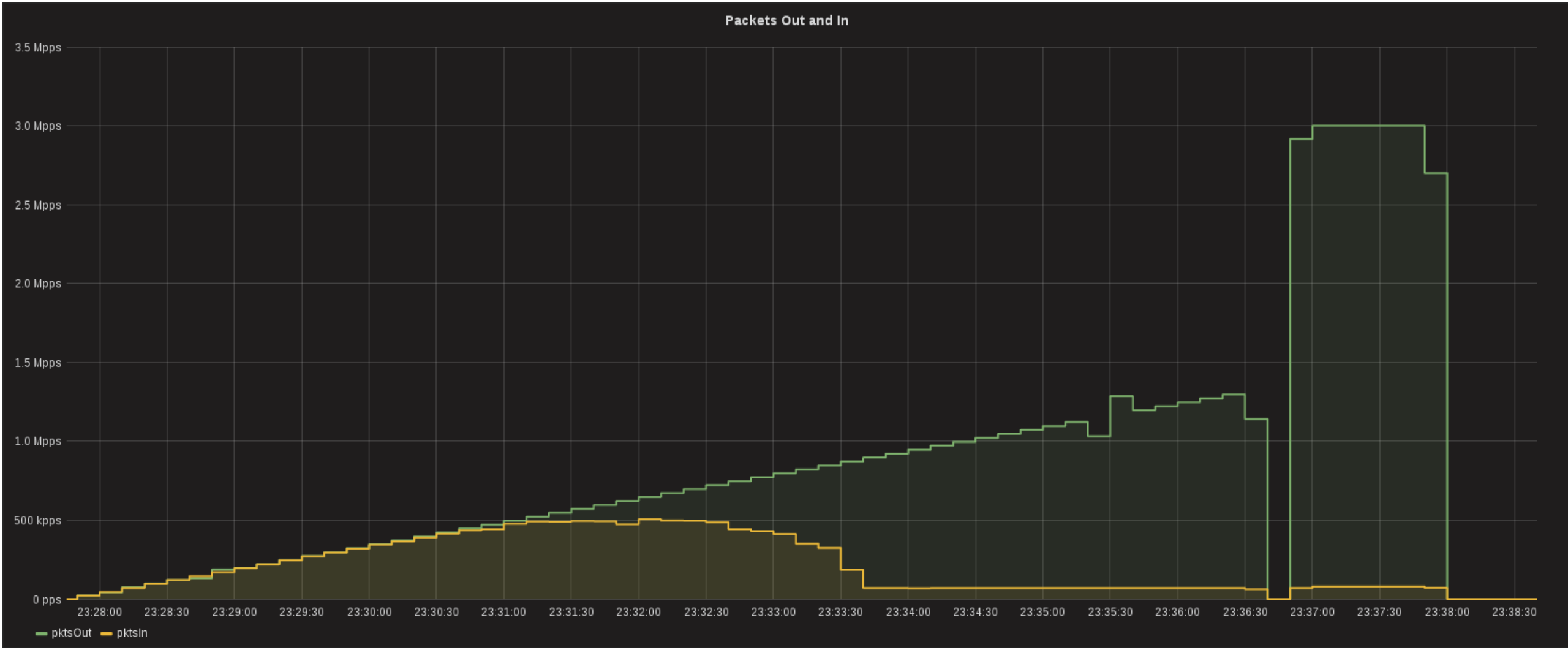
- Ipgen + netmap used for packet generation.
- Packets per second, not bytes.
- Minimal packet size.

What performance can we expect?

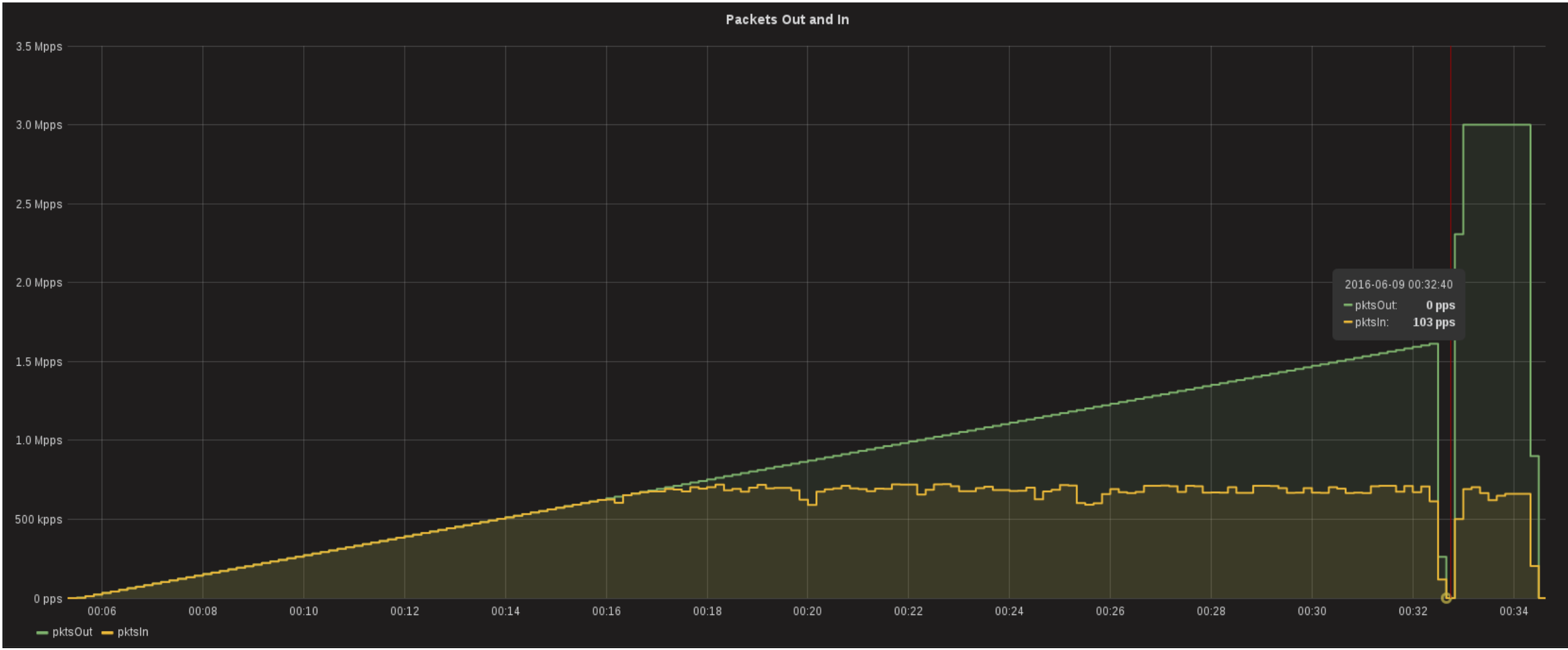
Without NIC, netisr or any other tuning.



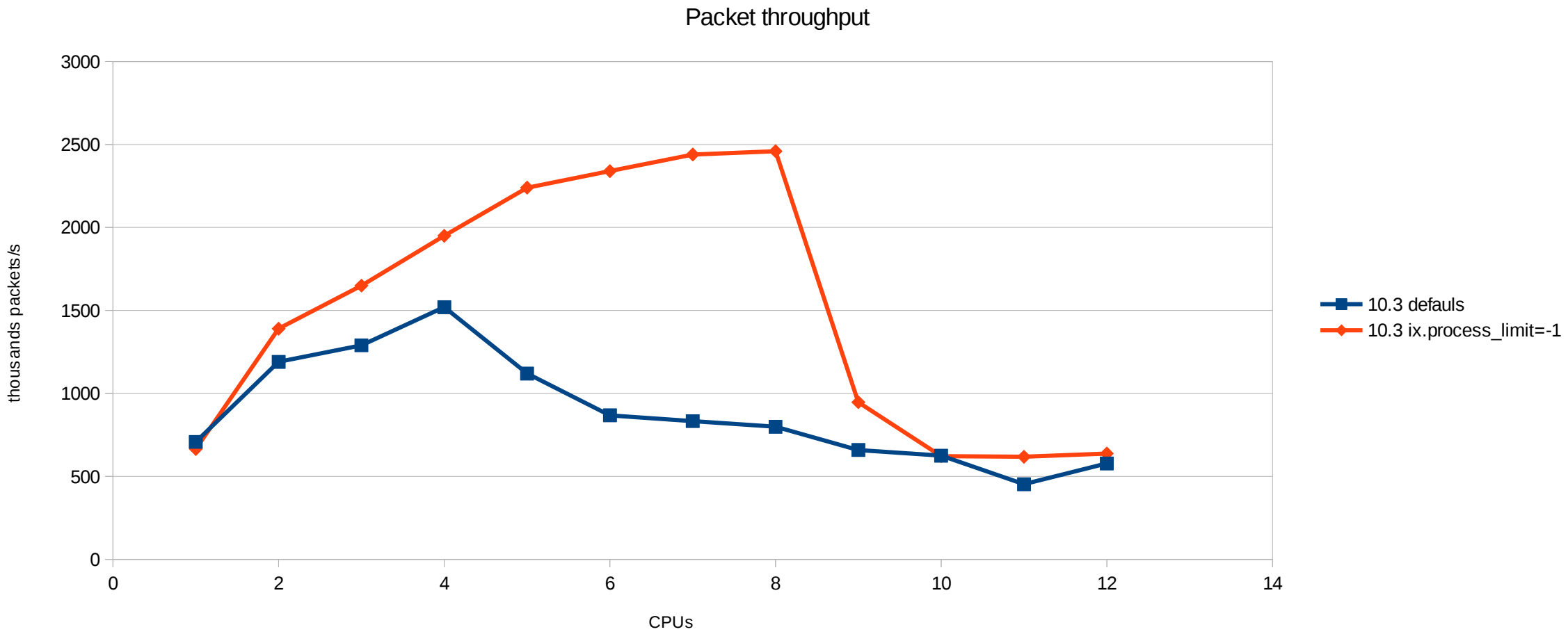
What performance can we expect?



What performance can we expect?

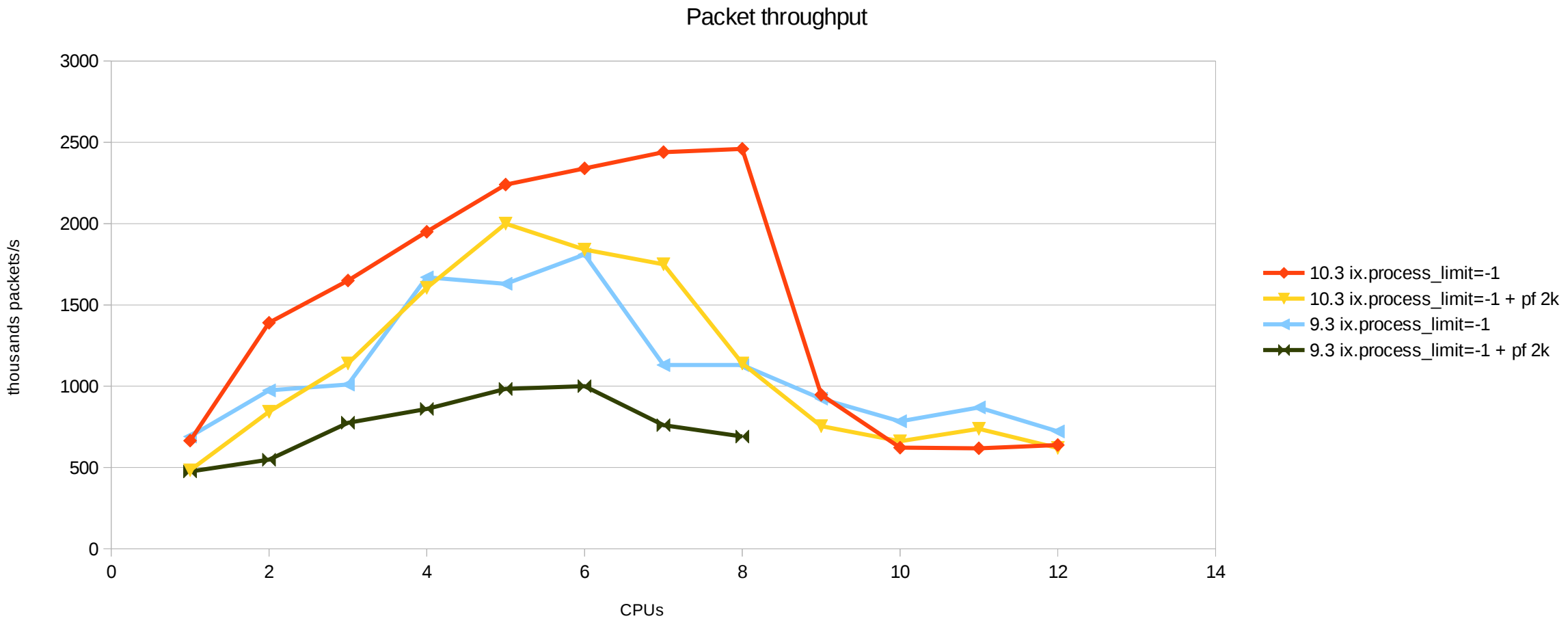


Be sure to configure your NIC!

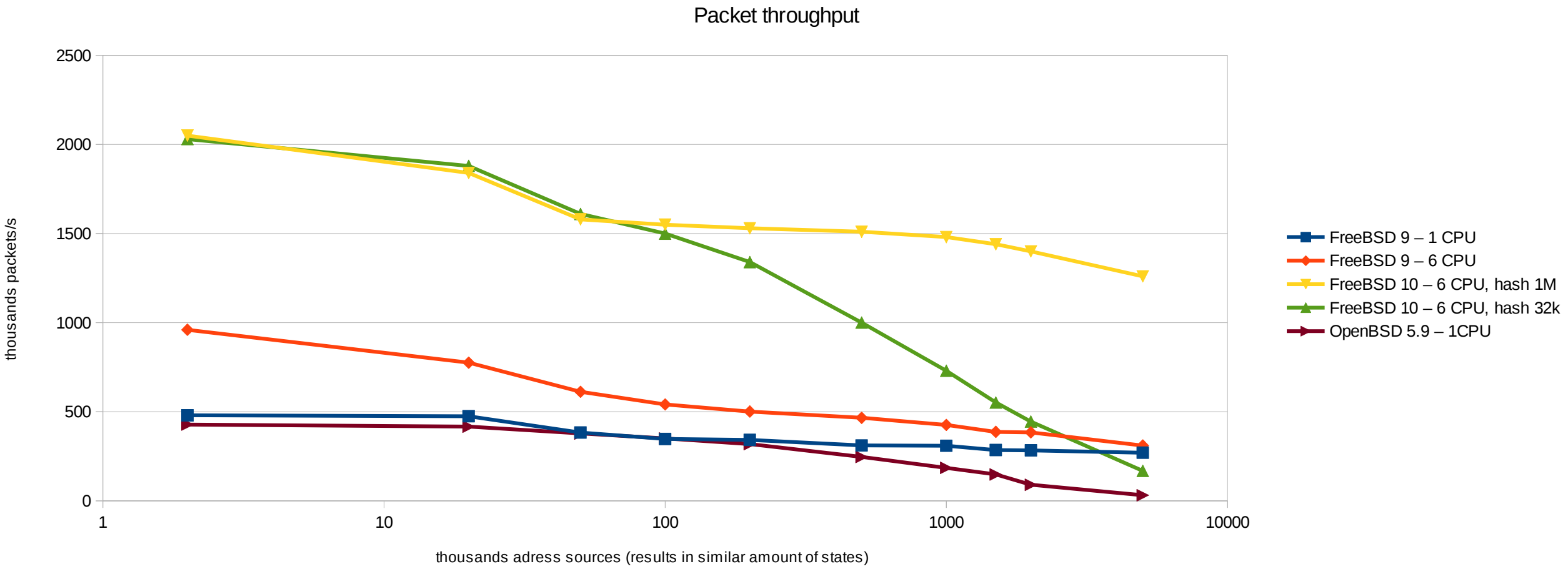


What performance can we expect?

Use FreeBSD 10!



Increase your hash size!



No graphs, but works better[1] in FreeBSD 10

- System load on carp/BGP failover.
- Recovery after DDoS is finished.

➔ New states can be created much faster.

[1] works better for me

Further work on pf

Ideas for performance

- Create source_node only after state gets fully established.
- Enable source_node deferring and/or synproxy only during attack, based on amount of states.
- SYN-Proxy without state creation would be awesome.
- Configurable rx-hashing would allow to bind attacked IP address to a single CPU.

Ideas for load balancing

- Have source_nodes not per rule, but per target table or label.
- Remove source_nodes per rule / table / label with linked states.
- Inject RSTs into states being removed.
- Make SYNProxy work with route_to.
- Make route_to work with pfsync.

Conclusions

You can use FreeBSD and pf for load balancing in 2016 with all the ugly stuff in the Internet.

Conclusions

- With FreeBSD 10 it should be possible to run SYN Proxy capable of dealing with DDoS[1]. On FreeBSD 9[2] SYN Flood = dead router.
- Tune your firewall's timeouts.
- Do you really need internal firewalling?
- Get a multi-queue NIC and multi-core CPU.

[1] Once SYN Proxy works with route-to.

[2] Or any other system having global-locking pf.

<applause>

Fin