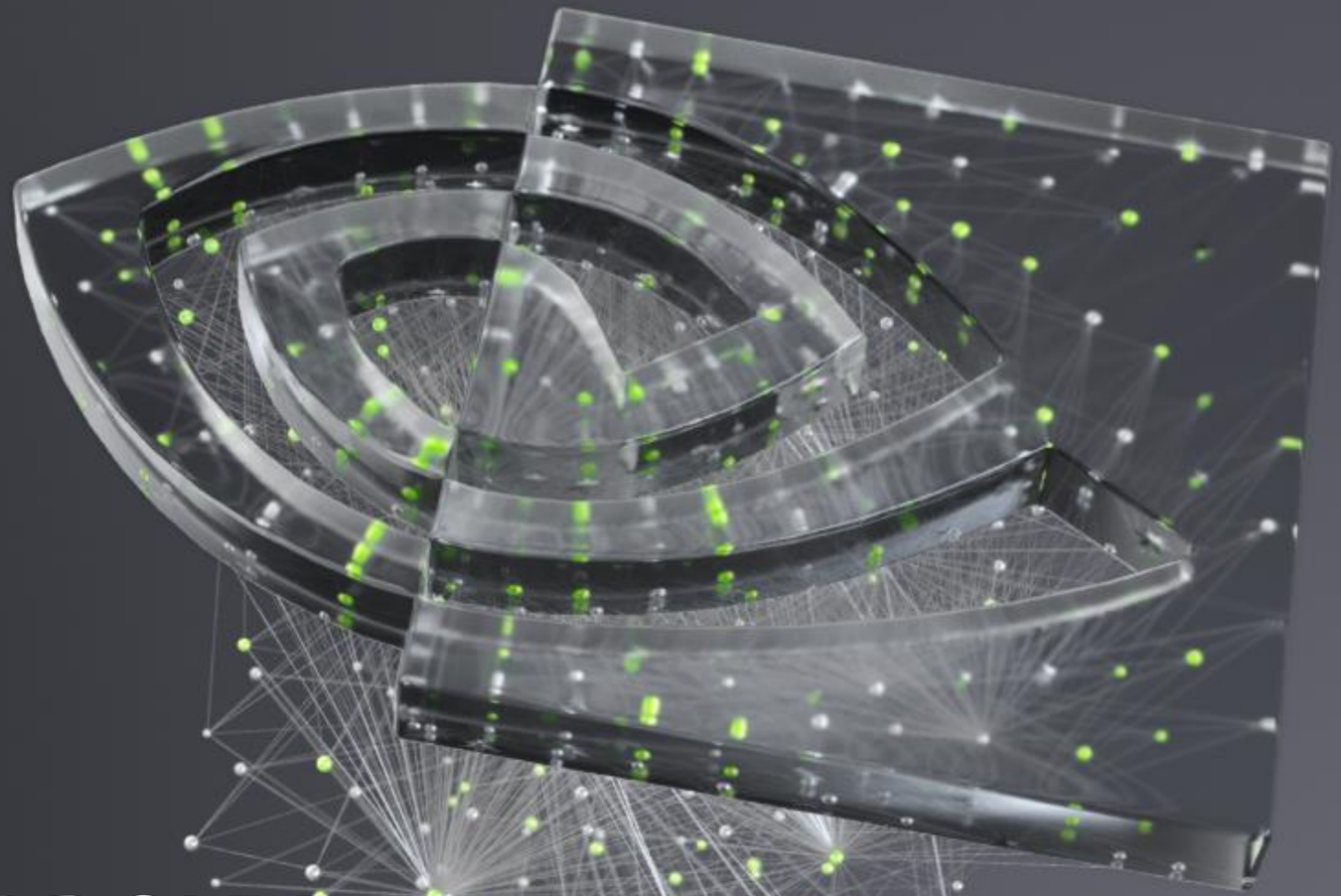


ACCELERATING SYNPROXY WITH XDP

Maxim Mikityanskiy, 2021

Architecture: Rony Efraim, Yossi Kuperman

Performance: Noam Kuehnberg, Yossi Kuperman, Maxim Mikityanskiy

AGENDA

Threat model, attack, mitigation

Describes the SYN flood attack, its dangers and a classical way to mitigate it.

Synproxy

Introduces the firewall based on iptables module synproxy that protects the server from the SYN flood attack.

XDP acceleration of synproxy

Shows our solution that speeds up SYN cookie generation, making the firewall more sustainable to the attack.

THREAT MODEL, ATTACK, MITIGATION

THREAT MODEL

TCP server is exposed to the Internet.

Attacker is not a man-in-the-middle (between the server and legit clients).

Attacker wants to disrupt the service by exhausting its resources using DoS/DDoS attacks.

(Exploiting programming vulnerabilities is out of scope of this talk.)

Bandwidth saturation attacks can generally be mitigated by rate-limiting per source IP address.

TCP SYN flood is kind of an attack that can’t be mitigated by rate-limiting connections.

This talk will focus on protection from TCP SYN flood.

TERMINOLOGY

Legit client – one who opens a TCP connection in a compliant way, i.e. by sending a SYN, receiving a SYNACK and sending an ACK, without violating the TCP protocol or spoofing the source IP address.

Malicious client (attacker) – one who attempts to deploy a SYN flood or disrupt the SYN flood protection mechanism described in this talk. It can send SYN packets without waiting for reply or making further action. It can spoof its source IP address. It can send ACK packets trying to circumvent SYN cookies.

Note that legit/malicious here apply to clients only with respect to the SYN flood attack. Other kinds of attacks exist, for example, opening a lot of connections (doing the whole 3-way handshake). Such clients will fall under the category of *legit* clients, because such attacks don't have properties of SYN flood, and mitigations for them are different (e.g., rate-limiting the number of connections per source IP address).

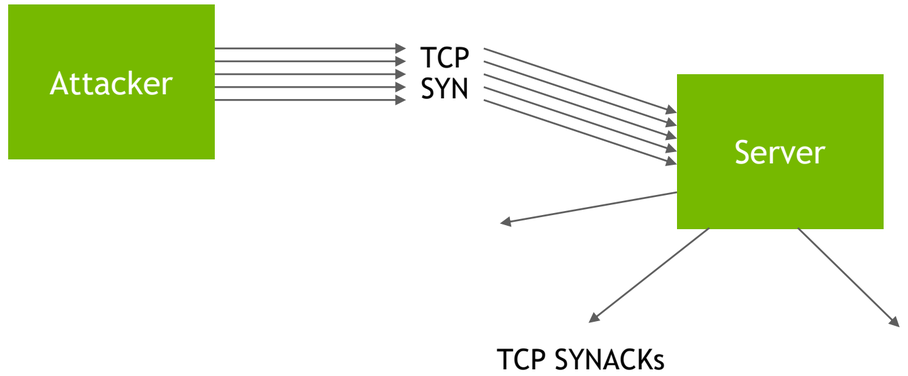
SYN FLOOD ATTACK

TCP connection establishment is stateful.

Incoming SYN leads to resource allocation.

Source IP address may be spoofed.

Simpler kind of an attack.



OUTCOME OF SYN FLOOD

Dangers of SYN flood:

Not possible to block or rate limit the attacker if it spoofs IP addresses.

After a SYN, the server sends 6 SYNACKs for 31 seconds (by default in Linux).

Half-open connections easily fill the backlog queue, leading to denial of service for legit clients.

Approach to mitigation:

The goal is to distinguish the legit clients from malicious and avoid allocating resources for malicious ones.

The criterion: a malicious client doesn’t respond to the SYNACK.

The solution must not expose new vulnerabilities.

SYN COOKIES SOLUTION

Make the handshake stateless until the server proves the client is legit.

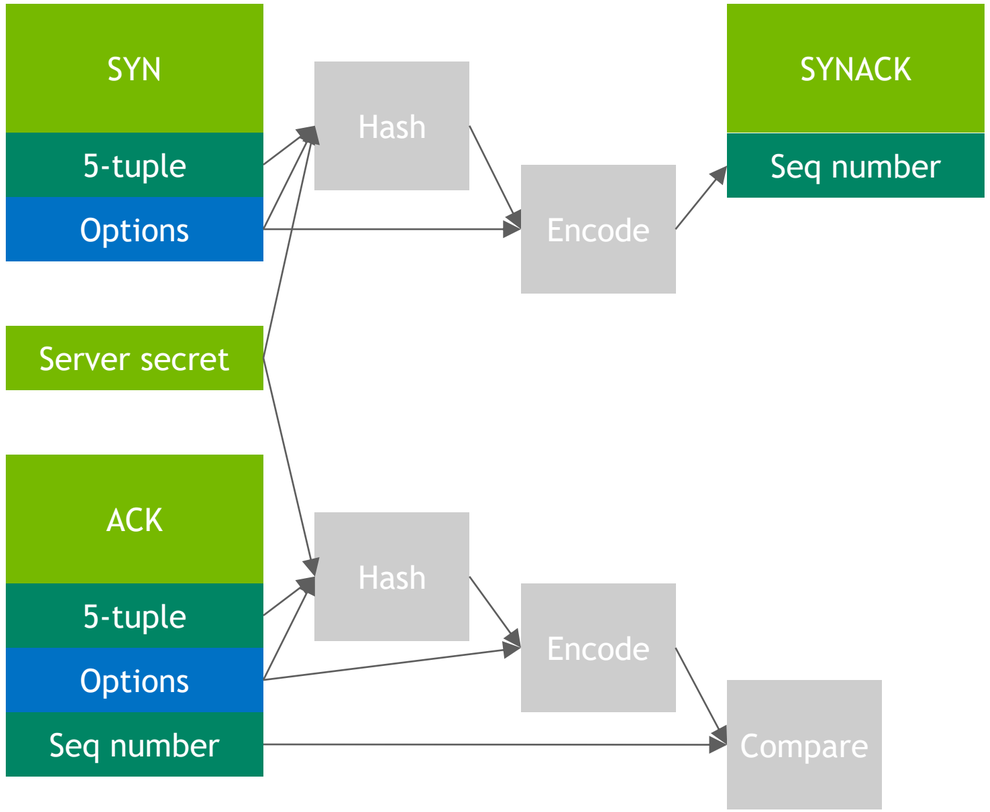
Encode the state in the sequence number (SYN cookie) and timestamp (TS cookie) when sending a SYNACK.

Legit clients will send the state back with the first ACK.

The SYN cookie includes a cryptographically secure hash of the 5-tuple, slowly increasing timer and the secret value.

Upon receiving the first ACK, the server can calculate that hash again and verify the client.

Resources are allocated after verifying the client and restoring the state from the SYN and TS cookies.



ATTACK SURFACE ON SYN COOKIES

SYNACKs are sent to potentially spoofed IP addresses, but there is no amplification.

The server no longer requires a SYN to open a connection.

The secret value prevents the attacker from guessing a cookie and opening the connection with ACK, skipping SYN. It makes IP address spoofing useless.

Timer prevents the attacker from reusing stored ACK packets.

The bottom line: SYN cookies successfully mitigate the dangers of SYN flood, making attackers switch to other attacks, which can be mitigated by other means (e.g., connection rate limiting per source IP address).

DRAWBACKS OF SYN COOKIES

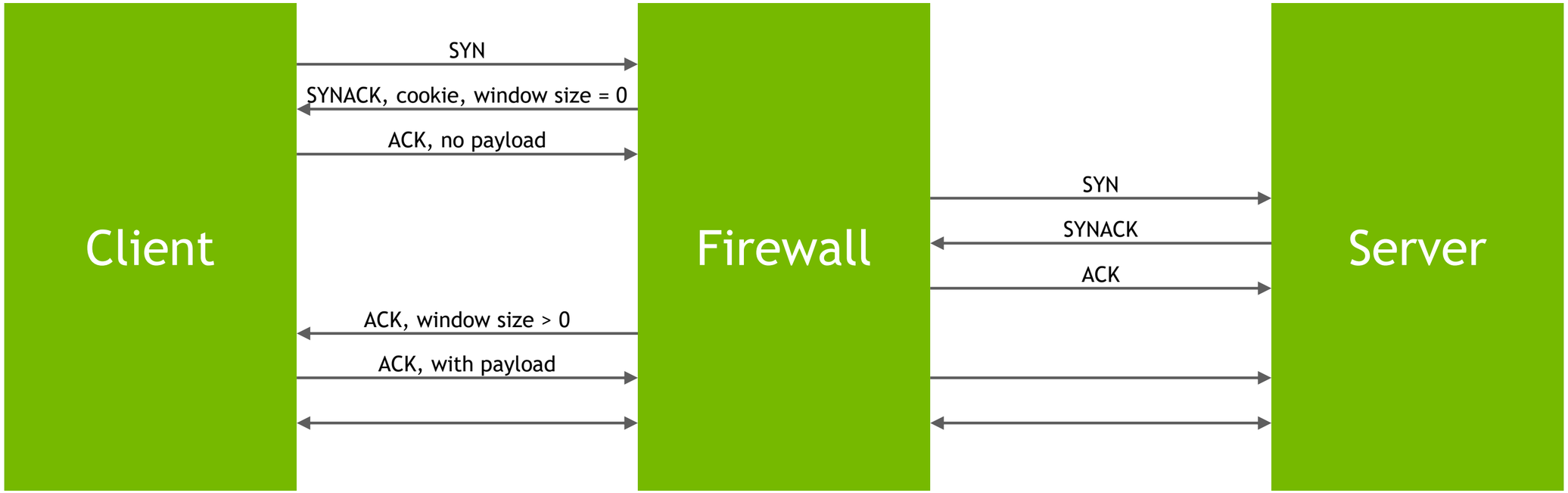
- SYNACKs are not retransmitted. Not a big deal, a legit client will retransmit a SYN.
- Some information is lost: precise value of MSS, client's initial sequence number, …
- If the client doesn't support TCP timestamps, more important information is lost: SACK permission, window scale, ECN support.
- If the first packet of the client is small (1..3 bytes), it may be dropped unnoticed, also damaging the client's MSS stored in the SYN cookie.
  If the first packet is bigger and gets dropped, the connection resets.
  Applies to Linux implementation of SYN cookies.
  More details [1].
  Not applicable to synproxy.

[1]: https://kognitio.com/blog/syn-cookies-ate-my-dog-breaking-tcp-on-linux/

SYNPROXY

SYNPROXY

Firewall that stands before the server, generates SYN cookies on server’s behalf and swallows the SYN flood attack.



SYNPROXY

Synproxy is an iptables module:

```
sysctl -w net.netfilter.nf_conntrack_tcp_loose=0
iptables -t raw -t PREROUTING -i eth0 -p tcp -m tcp --syn --dport 80 -j CT --notrack
iptables -A FORWARD -i eth0 -p tcp -m tcp --dport 80 -m state --state INVALID,UNTRACKED -j SYNPROXY \
        --timestamp --sack-perm --wscale 7 --mss 1460
iptables -A FORWARD -m state --state INVALID -j DROP
```

The server doesn’t spend its resources on malicious connections.

The firewall emits SYN cookies on behalf of the server. The same drawbacks apply, except the adverse effects of dropping the first ACK no longer apply, because the window size is set to 0 in the SYNACK.

Synproxy increases latency of connection establishment.

Synproxy needs to know in advance the server’s MSS and window scale.

WHY ACCELERATE WITH XDP?

Synproxy optimizes out heavy operations like conntrack modification [2].

Still, more code can be optimized out with XDP:

- Creating SKB.
- Going through networking stack and iptables rules.

The faster the firewall processes packets, the more resilient it is to stronger attacks.

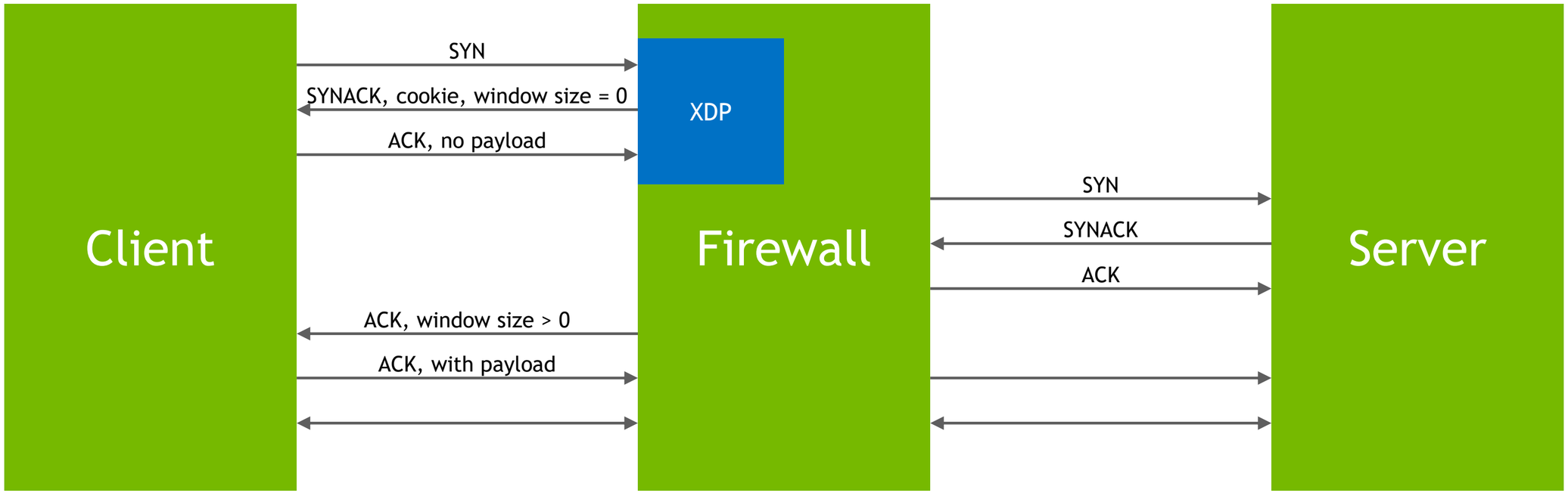
Firewall can be a low-end machine, such as DPU.

SYN cookie generation is stateless, so XDP is a good fit.

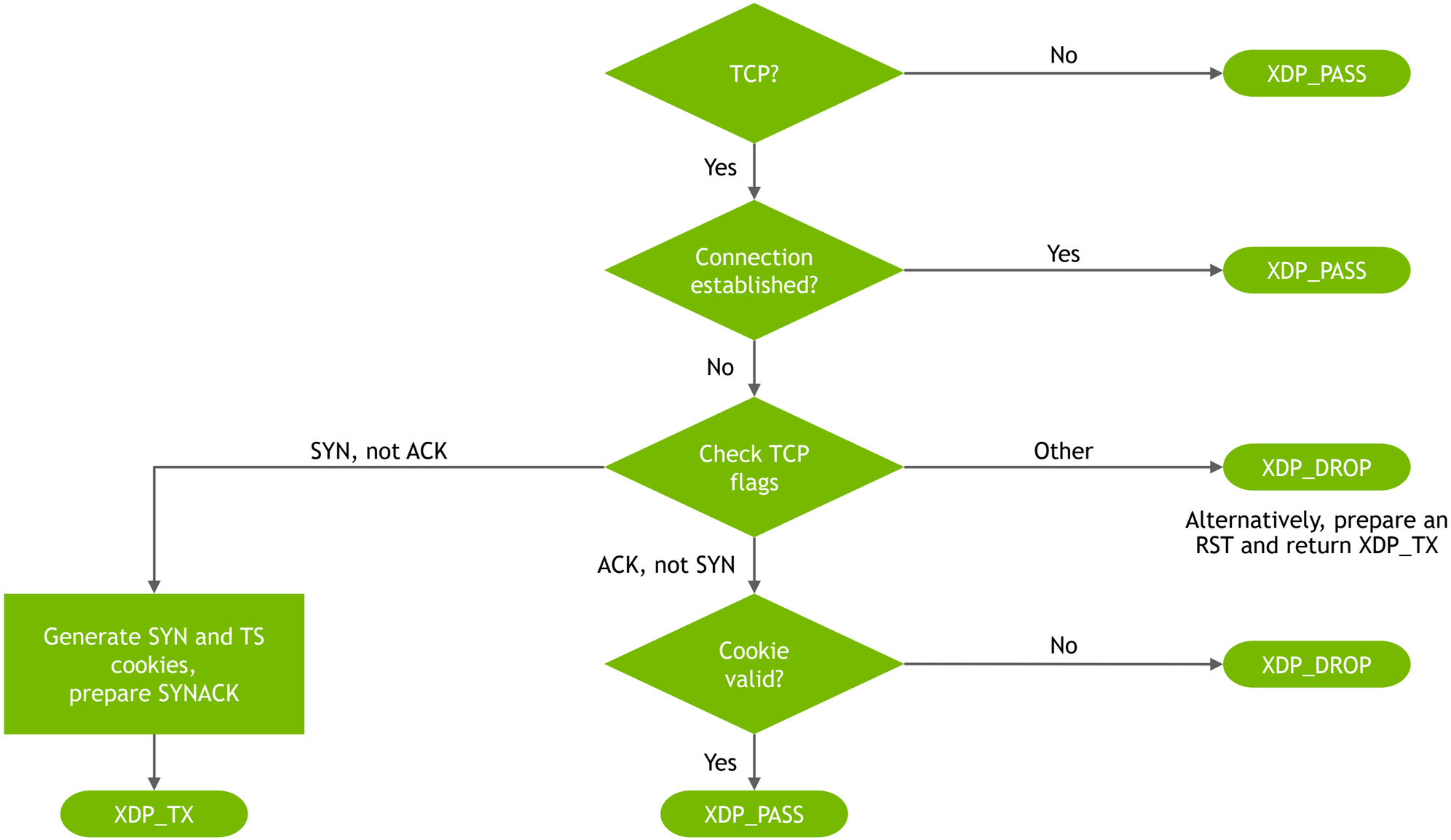
[2]: https://people.netfilter.org/hawk/presentations/devconf2014/iptables-ddos-mitigation_JesperBrouer.pdf

XDP ACCELERATION OF SYNPROXY

ACCELERATING SYNPROXY WITH XDP



XDP PROGRAM ALGORITHM



MISSING PARTS

Kernel has existing BPF helpers to generate and check SYN cookies: bpf_tcp_gen_syncookie and bpf_tcp_check_syncookie [3].

Limitations:

- Listening socket is required – not suitable for synproxy.
- No timestamp cookie support.

There are no BPF helpers to query conntrack.

[3]: https://legacy.netdevconf.info/0x14/pub/slides/50/Issuing%20SYN%20Cookies%20in%20XDP.pdf

NEW BPF HELPERS

Helpers needed to implement the algorithm with synproxy:

- bpf_ct_lookup_tcp to lookup connection status in conntrack.
- bpf_tcp_raw_gen_syncookie to generate a SYN cookie without depending on a socket.
- bpf_tcp_raw_check_syncookie to check a SYN cookie without depending on a socket.
- bpf_tcp_raw_gen_tscookie to generate a TS cookie compatible with synproxy.

New helpers were implemented and are being prepared for upstream submission.

PERFORMANCE TEST

Malicious traffic: TRex

Legit traffic: httperf (10000 connections per second)

Server: nginx

CPU: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
(confined to single core)

NIC: Mellanox ConnectX-5 Ex

XDP_TX baseline: 10.8 Mpps

MALICIOUS TRAFFIC ONLY

SYN flood

synproxy: 406 kpps

XDP: 2.96 Mpps (7.3x more)

XDP_TX baseline: 10.8 Mpps

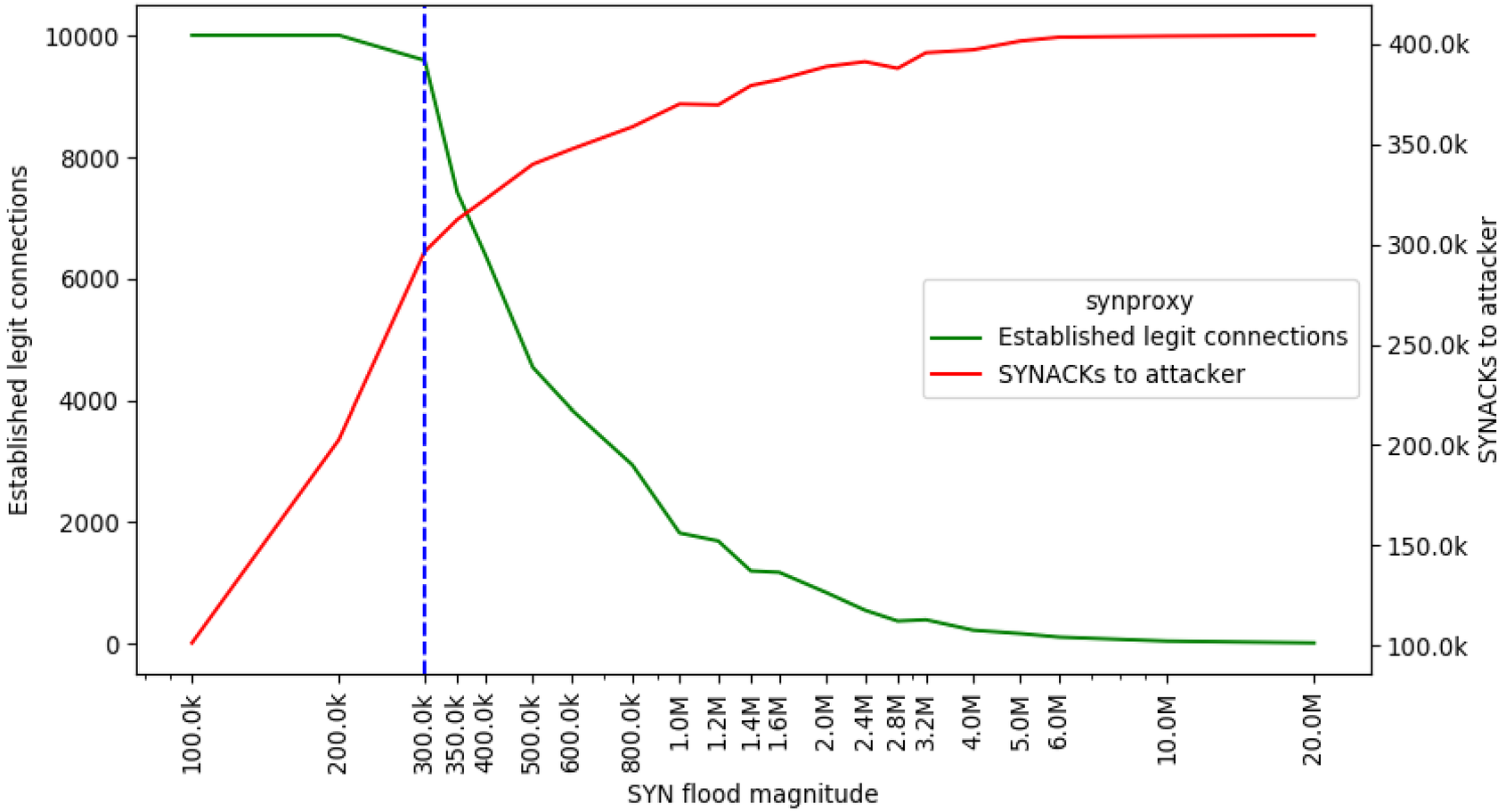
ACK flood

synproxy: 900 kpps

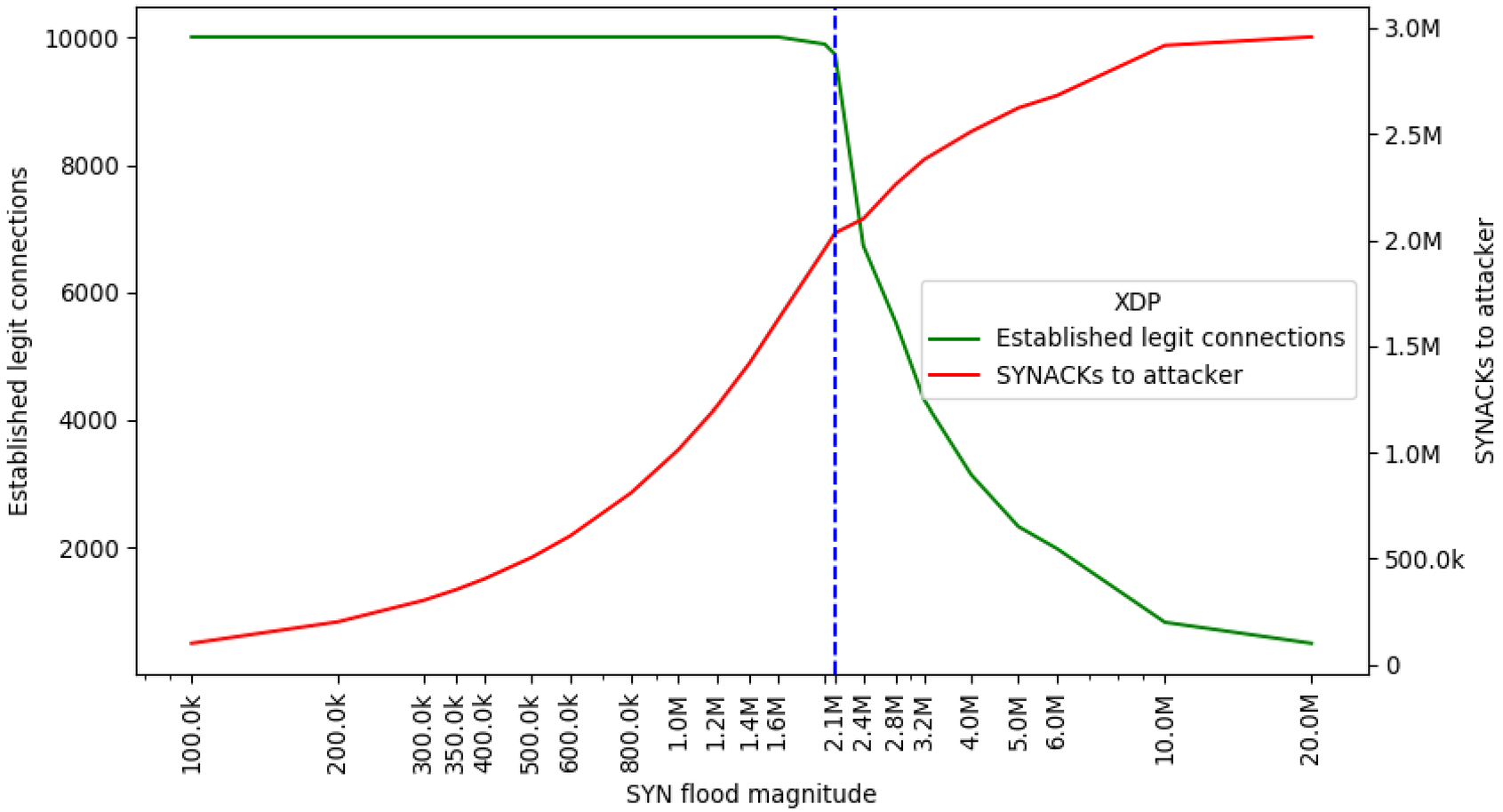
XDP: 5.6 Mpps (6.2x more)

XDP_DROP baseline: 24.5 Mpps

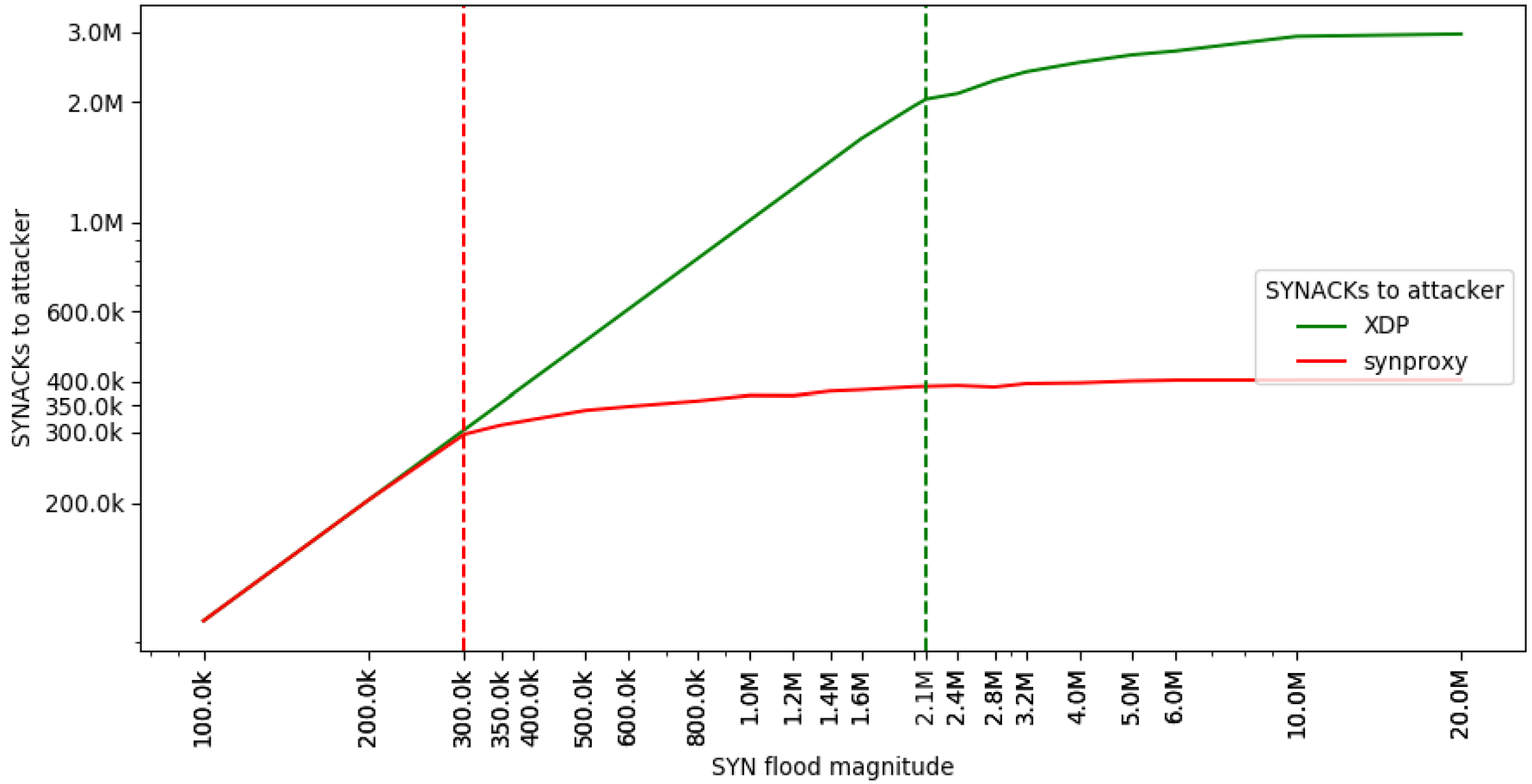
SYNPROXY BASELINE PERFORMANCE



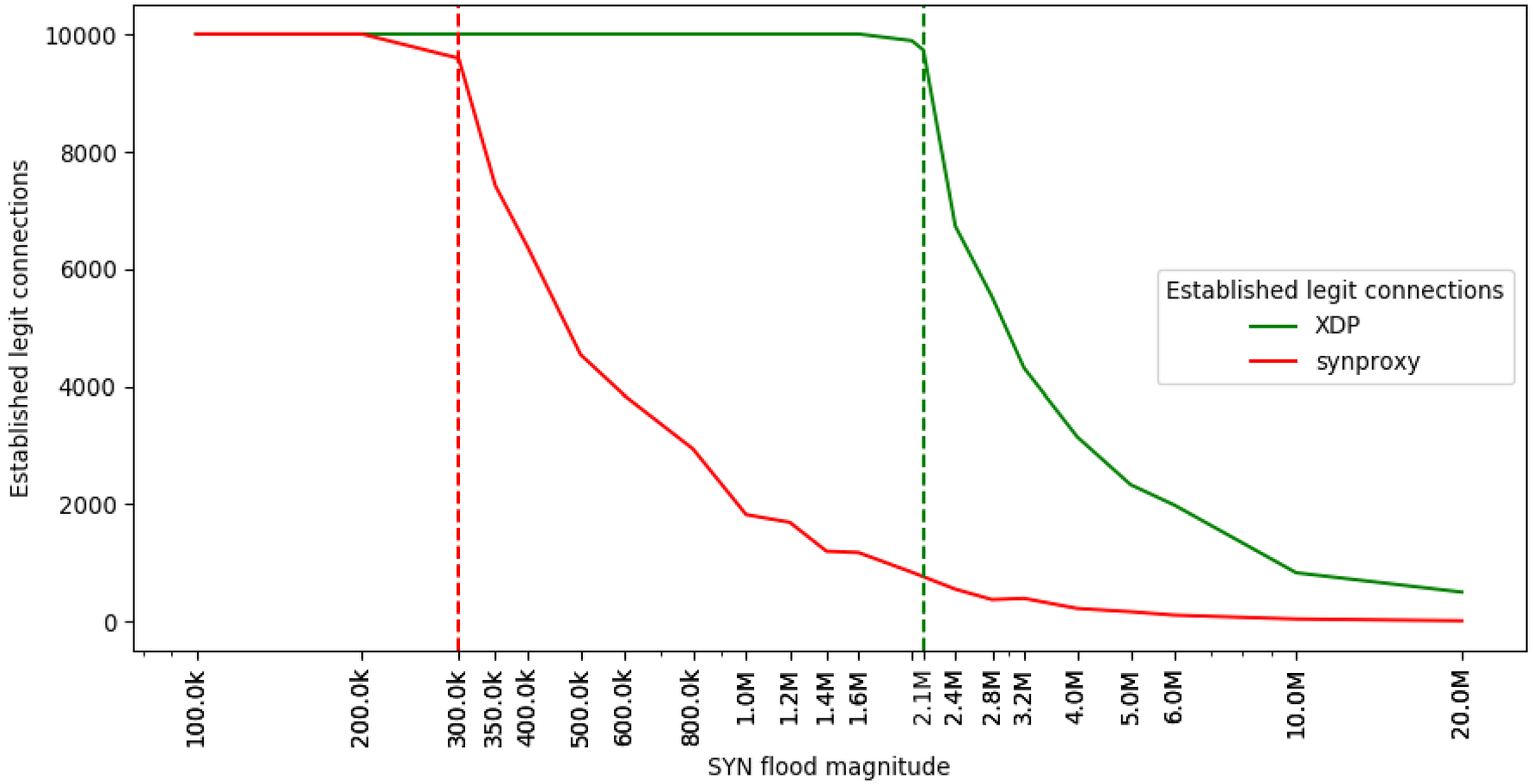
XDP ACCELERATION



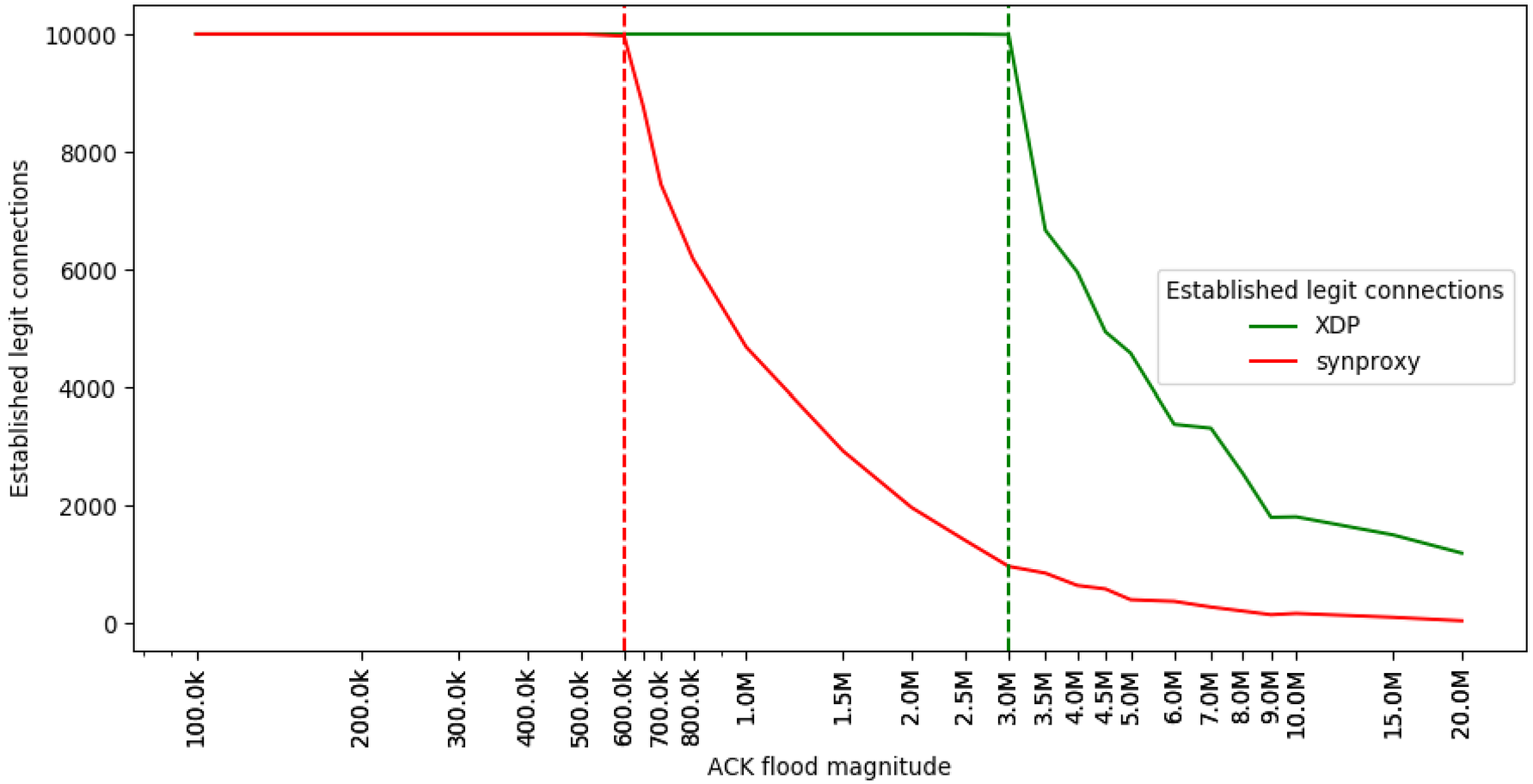
COMPARISON – NUMBER OF SYNACKS



COMPARISON – NUMBER OF CONNECTIONS



COMPARISON - ACK FLOOD



CONCLUSION

XDP allows to accelerate an existing synproxy-based solution.

Integration is fairly simple and doesn’t require changes to synproxy itself.

XDP-accelerated solution withstands SYN and ACK flood attacks, several times bigger than just synproxy does.

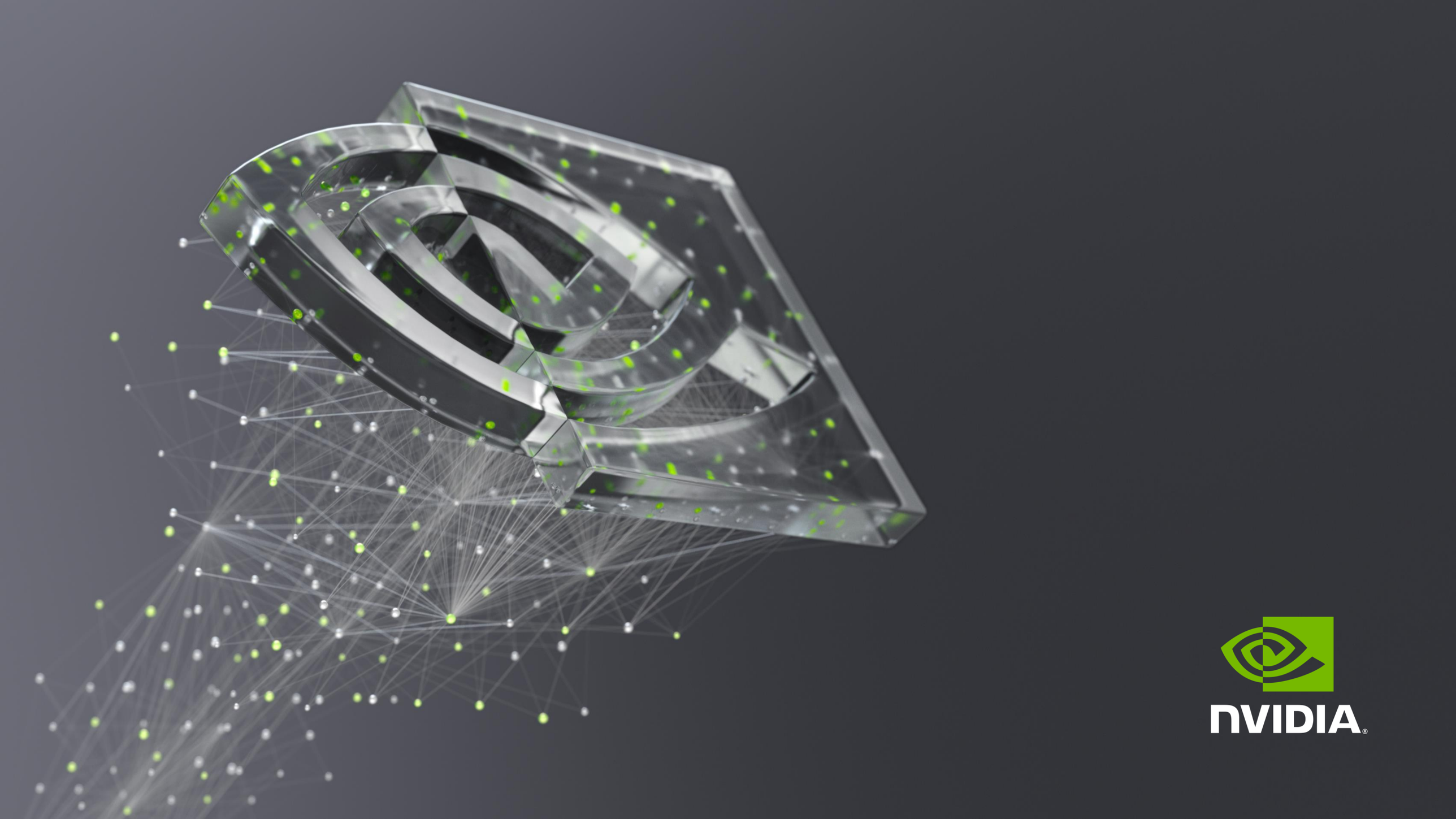
Work on preparing the patches for upstreaming is in progress.

REFERENCES

[1]: https://kognitio.com/blog/syn-cookies-ate-my-dog-breaking-tcp-on-linux/
SYN cookies ate my dog – breaking TCP on Linux
Graeme Cole

[2]: https://people.netfilter.org/hawk/presentations/devconf2014/iptables-ddos-mitigation_JesperBrouer.pdf
DDoS protection using Netfilter/iptables
Jesper Dangaard Brouer
RedHat

[3]: https://legacy.netdevconf.info/0x14/pub/slides/50/Issuing%20SYN%20Cookies%20in%20XDP.pdf
Issuing SYN cookies in XDP
Petar Penkov, Eric Dumazet, Stanislav Fomichev
Google


NVIDIA