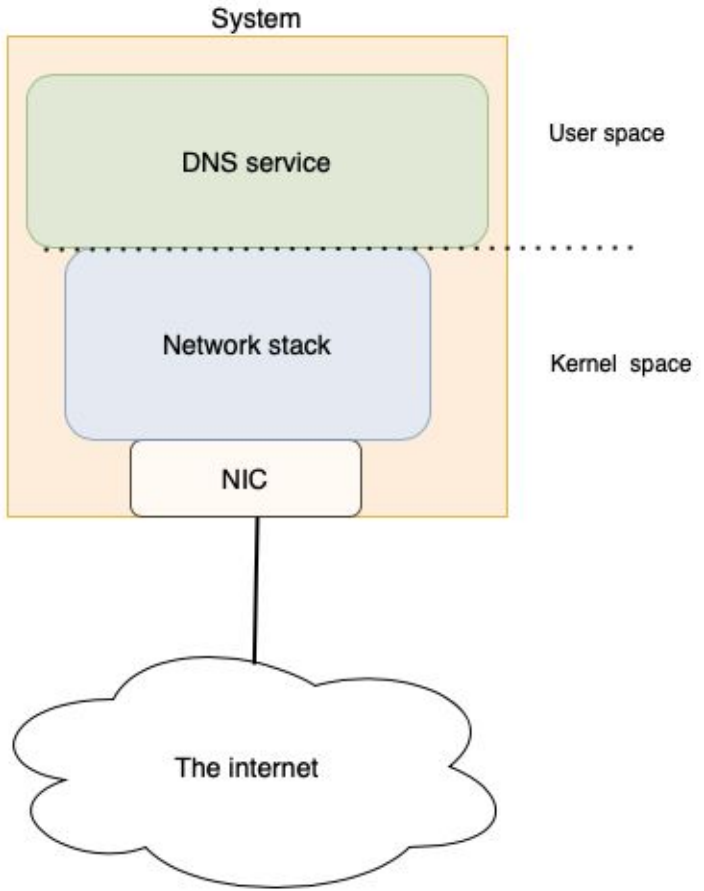
Server agnostic DNS augmentation

By Tom Carpay

Supervisors: Willem Toorop & Luuk Hendriks

Intro

- No DNS handling available low in the network stack, which is desirable for high volume authoritative servers
- Focus on DNS service agnostic
- Extended Berkeley Packet filter (eBPF)
- We don't fully know the possibilities of this technology



eBPF

- eBPF
  - Runs natively in Linux VM kernel space
  - Executes verified code
  - Limited instruction set
  - Execution limit (1 million instructions)
  - Different execution hooks

- Extensive high and low stack toolset used in many tracing tools

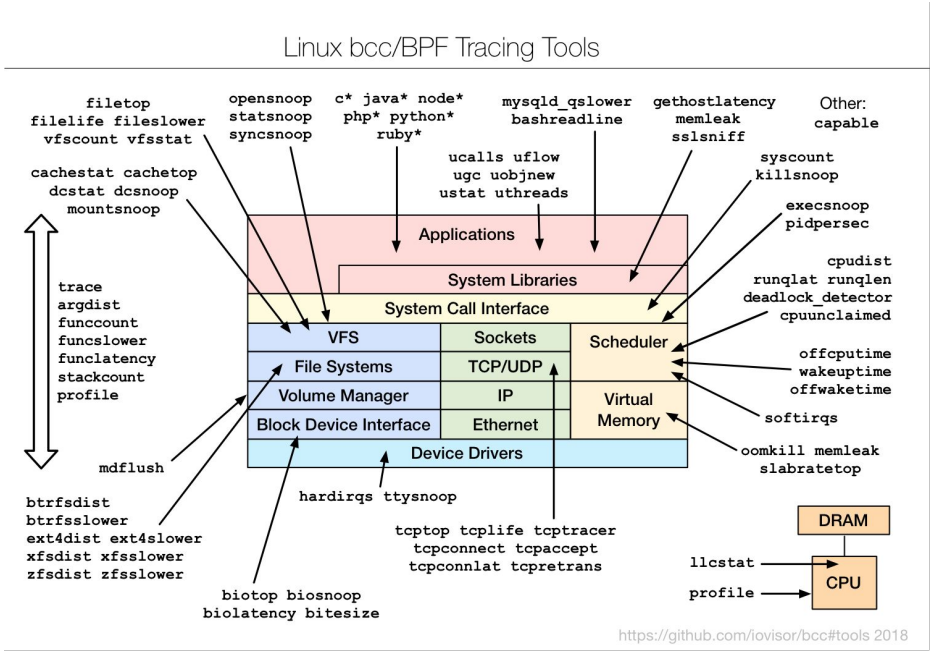
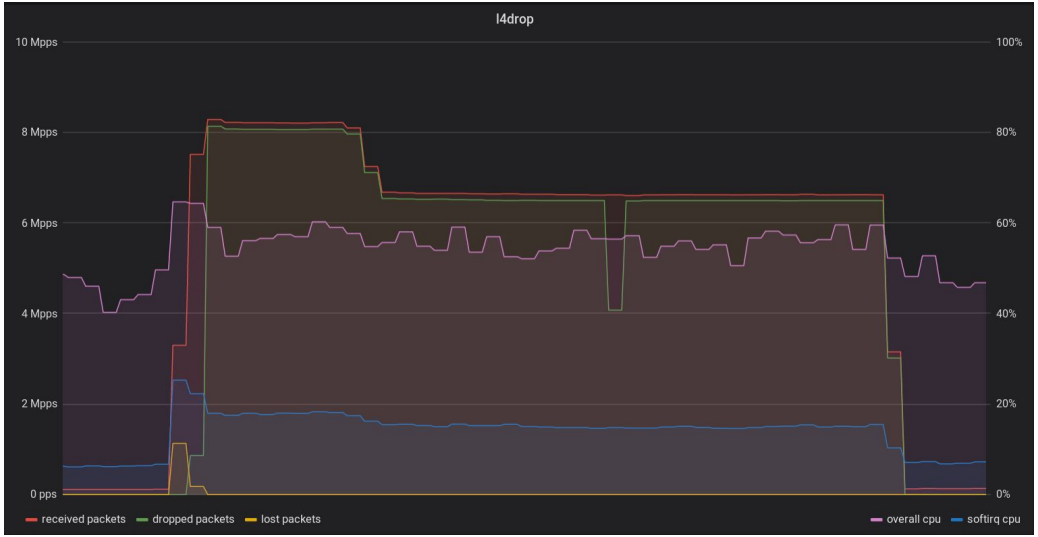


Fig. Linux tracing tools using eBPF.
Brendan Gregg 2018

Related work

- Knot DNS - Bypass the TCP/IP stack
- Cloudflare: L4 Drop - XDP DDOS protection
- Various papers evaluating eBPF performance



Cloudflare's L4Drop in action. 2018.

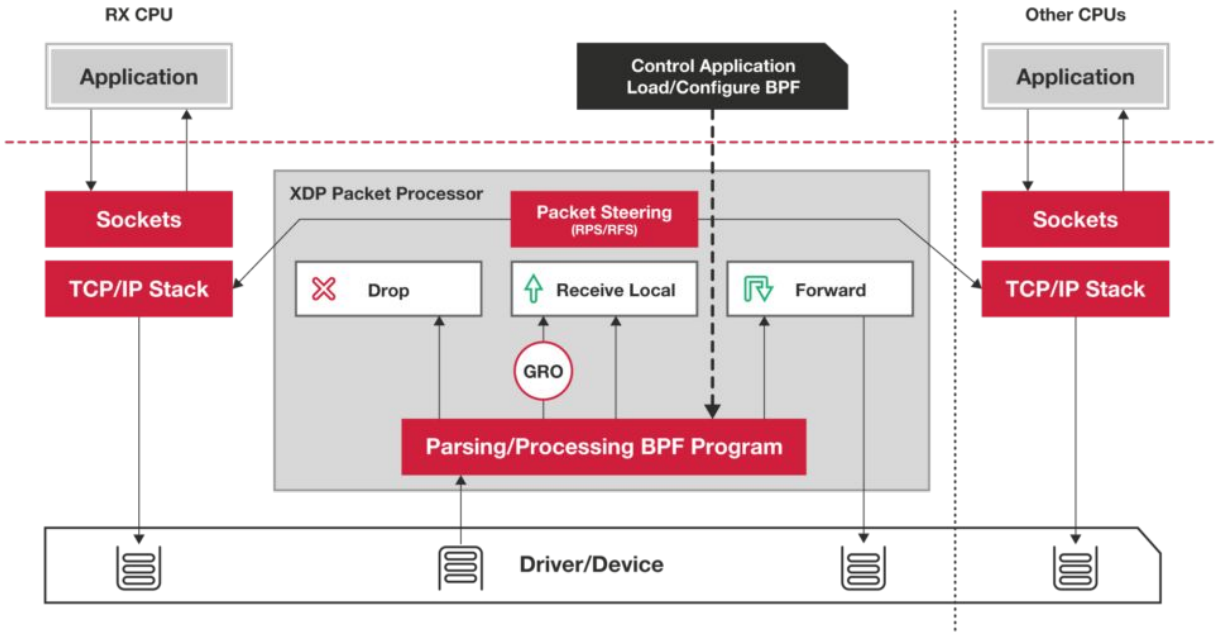
Research questions

How can XDP eBPF be used to augment and improve DNS software?

- Which features from XDP eBPF could be used to augment DNS software?
- How can DNS augmentations be implemented based upon these XDP eBPF features?
- How do these implementations impact performance?

The eXpress Data Path hook

- XDP actions
  - XDP_PASS
  - XDP_DROP
  - XDP_ABORTED
  - XDP_TX
  - XDP_REDIRECT



XDP IoVisor, 2018.

XDP eBPF features

- XDP & Traffic Control (TC) hooks
- Change packet size and contents
- Bypass network stack, XDP offloading
- Userspace “maps” and configuration e.g.
  - ARRAY
  - HASHMAP
  - PERCPU_ARRAY
  - PERCPU_HASHMAP
  - LPM_TRIE

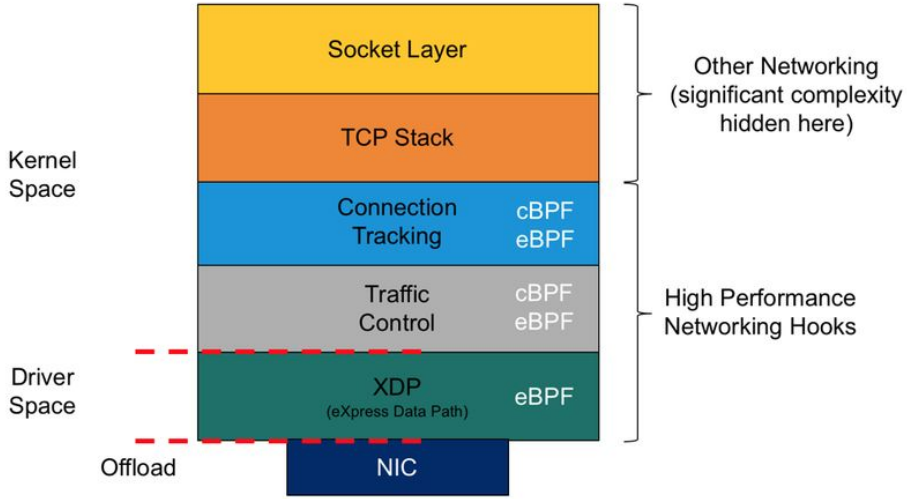


Fig. XDP in the network stack. Adapted from Quentin Monet, Netronome, 2018

```
struct bpf_map_def SEC("maps") state_map = {
    .type = BPF_MAP_TYPE_PERCPU_ARRAY,
    .key_size = sizeof(uint32_t),
    .value_size = sizeof(struct bucket),
    .max_entries = 1
};
```

Prototypes

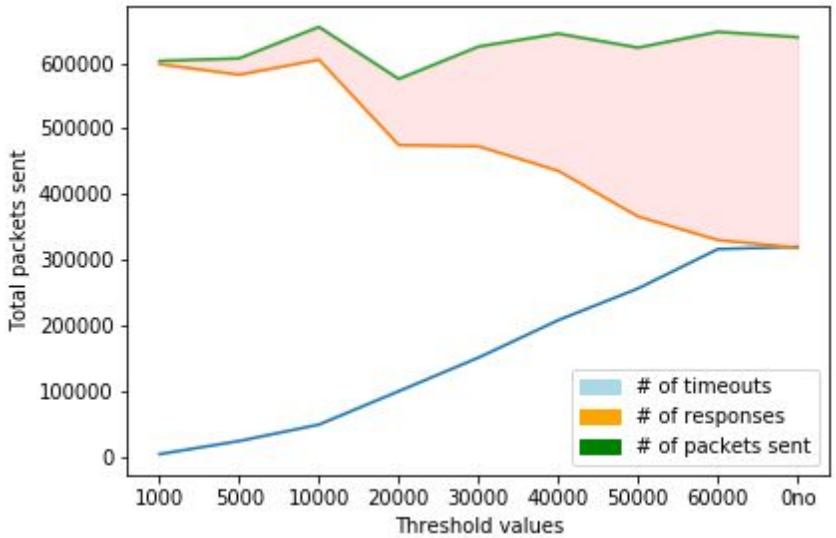
- QName rewrite (collaborative work)

- Response Rate Limiting (RRL)
  - Basic prototype
  - Per IP RRL
  - Unknown host RRL

Response Rate Limiting

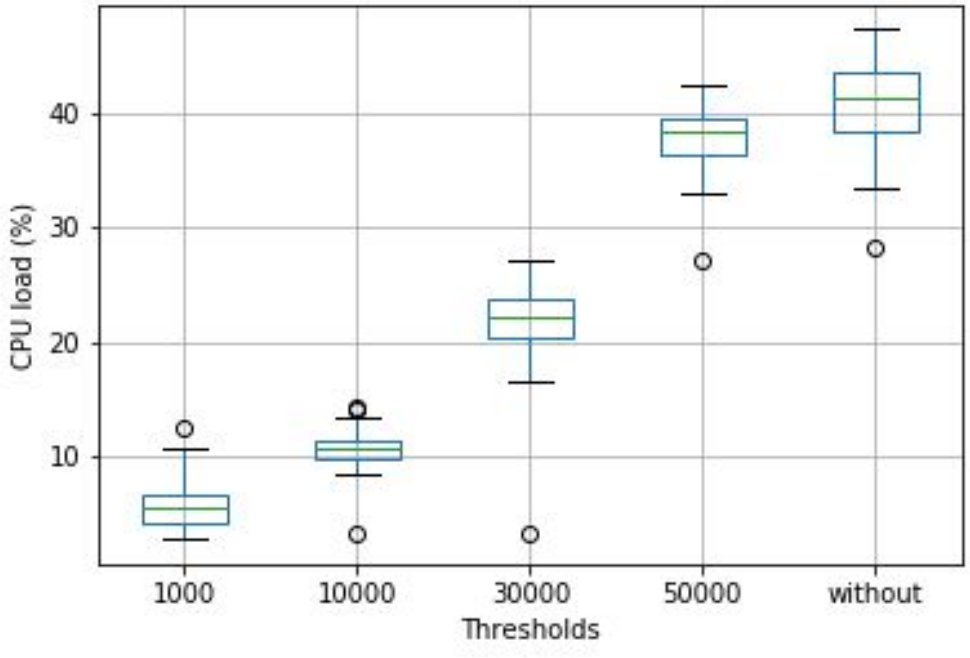
- How many packets have I seen in my current time frame? Cut off after threshold
- Check time frame a percentage of the time
- Flamethrower tool to query NSD
- Check rate of 50%, time frame of 1 second, 10 second bursts

Timeouts vs responses



Response Rate Limiting cont.

The combined CPU load per threshold



Discussion and future work

- Flamethrower measurements are subject to network variability
- RRL of NSD shows that the RRL prototype works, though it does not reduce timeouts

- CPU load dependent adaptive RRL
- DNS cookies

Summary

- Which features from XDP eBPF could be used to augment DNS software?
  - Literature study
- How can DNS augmentations be implemented based upon these XDP BPF features?
  - Prototypes
- How do these implementations impact performance?
  - Experiments to validate and quantify prototypes

How can XDP BPF be used to augment and improve DNS software?

- Offload and add functionalities regardless of the DNS service