XDP Production Usage: DDoS Protection and L4LB

Huapeng Zhou (hzhou@fb.com)
Nikita (tehnerd@fb.com)
Martin Lau (kafai@fb.com)
Load Balancing

**SHIV (L4 LB)**

**Proxy (L7 LB)**

**Proxy (L7 LB)**

**Proxy (L7 LB)**
SHIV (L4 LB)
An IPVS to XDP_TX Transition
SHIV: Map Lookups

LRU Map (/Cache)

Sharding Map (Proxy IPs)

XDP_TX
XDP_TX to Proxy

bpf_xdp_adjust_head()

[Diagram showing the flow of packets from XDP_TX to Proxy, with fields such as Client IP, VIP, sport, dport, and payload.]

XDP_TX
Droplet
Traffic Spike at SHIV?
Droplet (Dreamlist):

- Fast packet drop
- Earliest stage in the networking stack
- Programmability and Flexibility
  - Easy to develop and quick to deploy (No kernel reboot)

Realized by XDP_DROP
Drop at HW limited-rate
Droplet: DDoS Protection Framework

- BPF program written in C
- Runtime compilation using bcc
- Loaded in kernel + Map Setup
- Run in the NIC Driver (XDP)
Chaining Multiple BPF_PROG

Using `bpf_tail_call` + `BPF_MAP_TYPE_PROG_ARRAY`
Questions?
SHIV (L4 LB)
**Droplet: DDoS Protection Framework**

Programmability: abstract away interactions with user space

- **Droplet handler**: handles the dirty work
  - Runtime compilation
  - Kernel load/hook
- **Different types of handlers**
  - GenericHandler
  - IPHandler
  - PrefixHandler ...
- The user only needs to write BPF code in C
Lab tests (w/ pktgen)

Under 99% cache hit: 3x to 6x improvement

Under 0% cache hit: 10x
(up to 25x w/o session tracking)
TCP/IP stack processing on recv

IPVS is too generic

Poor DDOS survivability

Hard to add new features
XDP vs IPVS