Super Fast Packet Filtering with eBPF and XDP

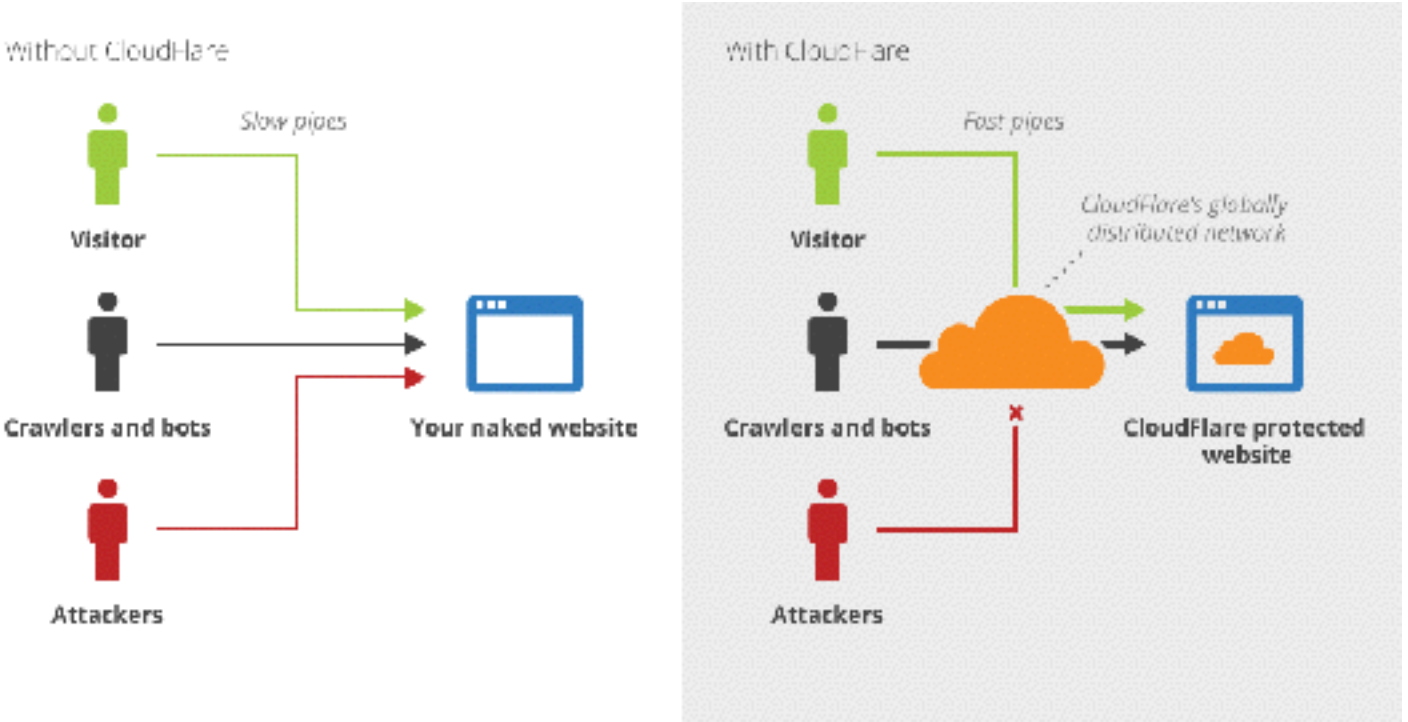
Helen Tabunshchyk, Systems Engineer, Cloudflare

@advance_lunge

Agenda

1. Background.
2. A tiny bit of theory about routing.
3. Problems that have to be solved.
4. Overview of existing solutions.
5. DDoS mitigation pipeline.
6. eBPF and XDP.
7. Bonus part.

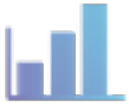
A bit of context about the work I do


Without CloudHare
Slow pipes
Visitor
Crawlers and bots
Attackers
Your naked website
With CloudHare
Fast pipes
CloudFlare's globally distributed network
Visitor
Crawlers and bots
Attackers
CloudFlare protected website

What does Cloudflare do?

CDN

- Moving content physically closer to visitors with our CDN
- Intelligent caching
- Unlimited DDOS mitigation
- Unlimited bandwidth at flat pricing with free plans



Website Optimisation

- Making web fast and up to date for everyone.
- TLS 1.3 (with 0-RTT)
- HTTP/2 + QUIC
- Server push
- AMP
- Origin load-balancing
- Smart routing
- Workers
- Post quantum crypto
- Many more

DNS

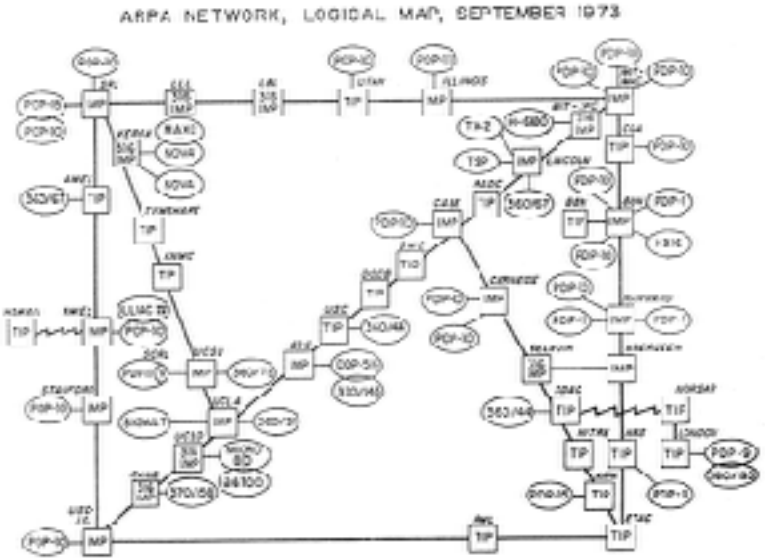
- Cloudflare is the fastest managed DNS providers in the world.
- 1.1.1.1
- 2606:4700:4700::1111
- DNS over TLS

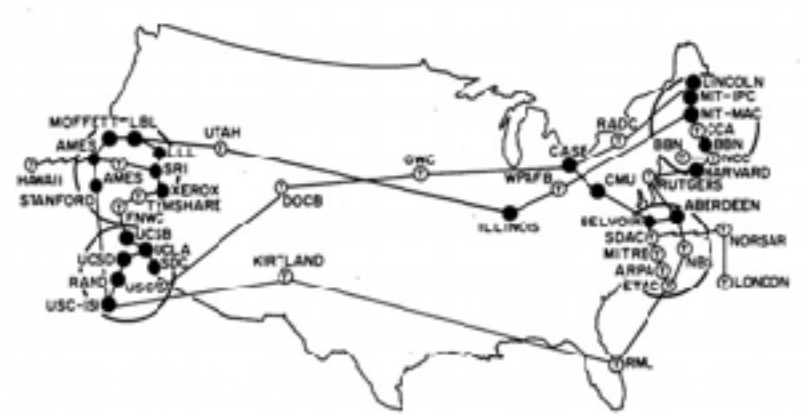
Scale

- 154 data centres in 74 countries
- More than 10 million domains
- 10% of all Internet requests
- 7.5M requests per second on average, 10M at peak
- 1.6M DNS queries per second
- 2.8 billion people served each month
- Biggest DDoS attack - 942 Gbps
- 20 Tbps network capacity and growing

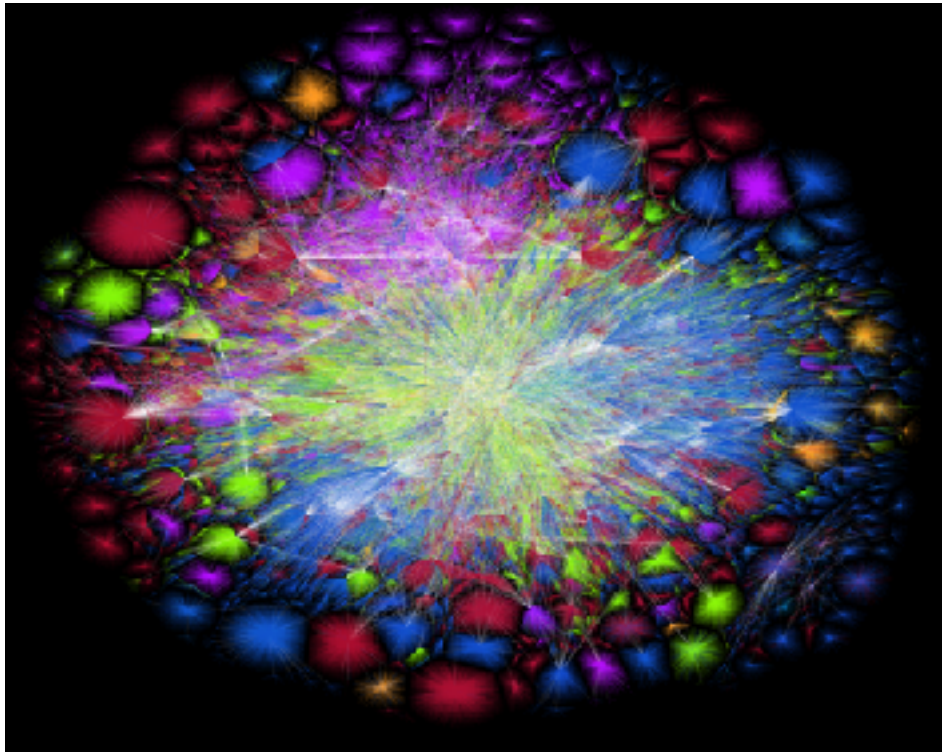
Life of a packet

A long time ago in a galaxy far, far away...





The OPTE Project Internet 2015 Map



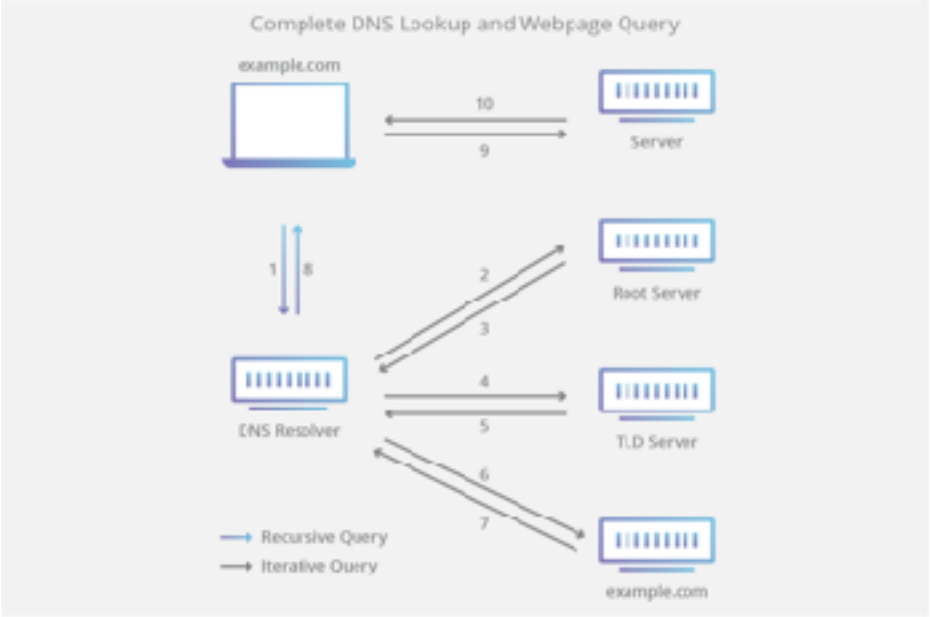
North America (ARIN)
Europe (RIPE)
Latin America (LACNIC)
Asia Pacific (APNIC)
Africa (AFRINIC)

“Backbone” (highly connected networks)

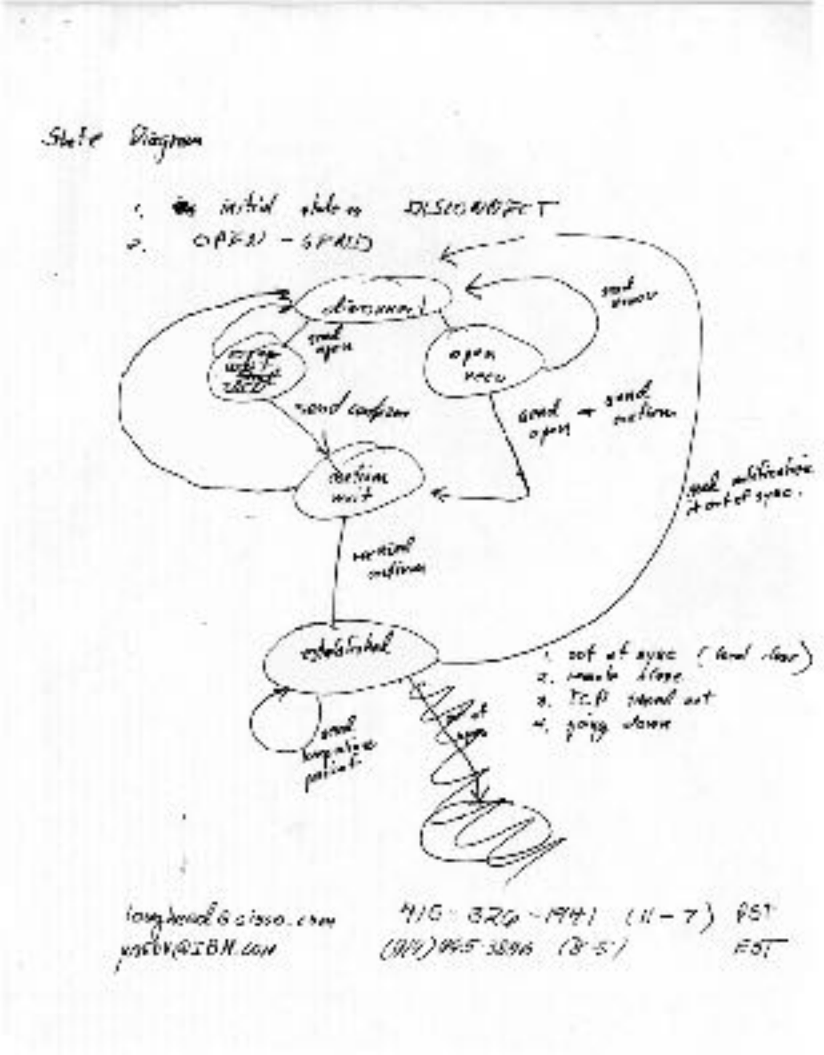
http://www.opte.org

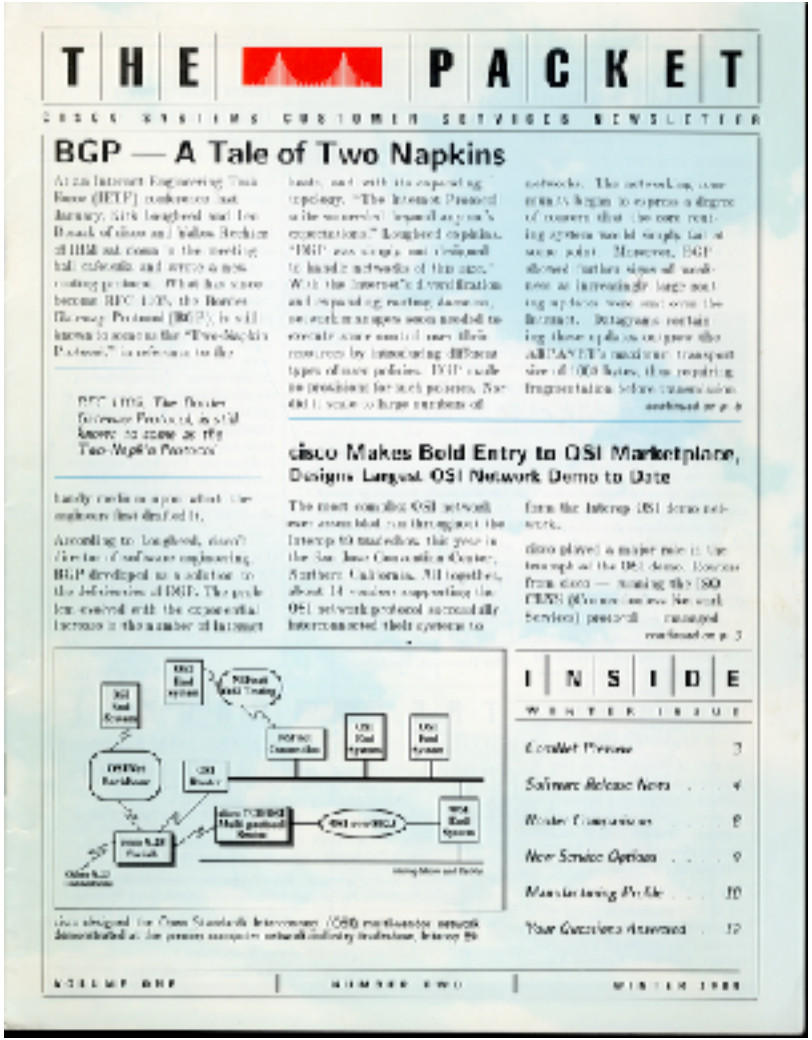
Load Balancing Between Data Centres

- Locality and congestion control
- DNS
- BGP
- Anycast



https://www.cloudflare.com/learning/dns/what-is-dns/




THE PACKET

BGP — A Tale of Two Napkins

RFC 1105, The Border Gateway Protocol, is still known to some as the Two-Napkin Protocol

cisco Makes Bold Entry to OSI Marketplace, Designs Largest OSI Network Demo to Date

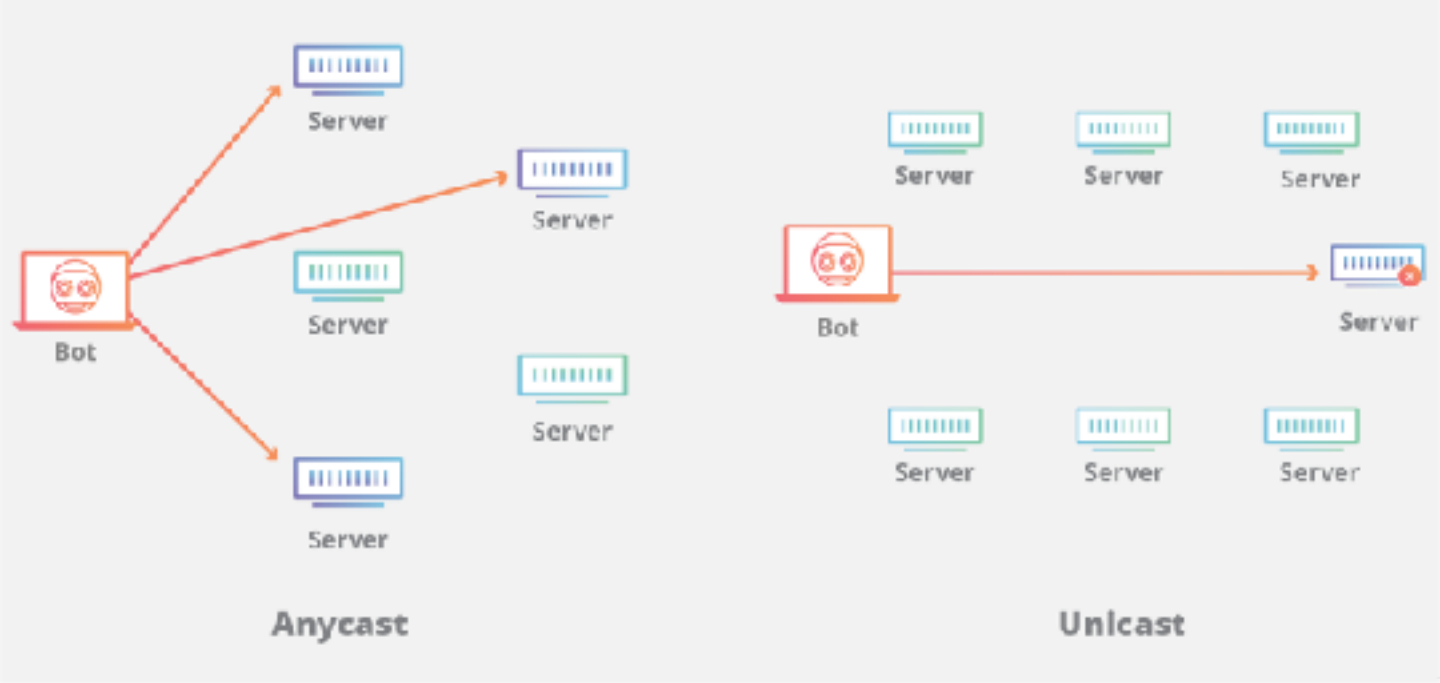
INSIDE

Software Release Notes

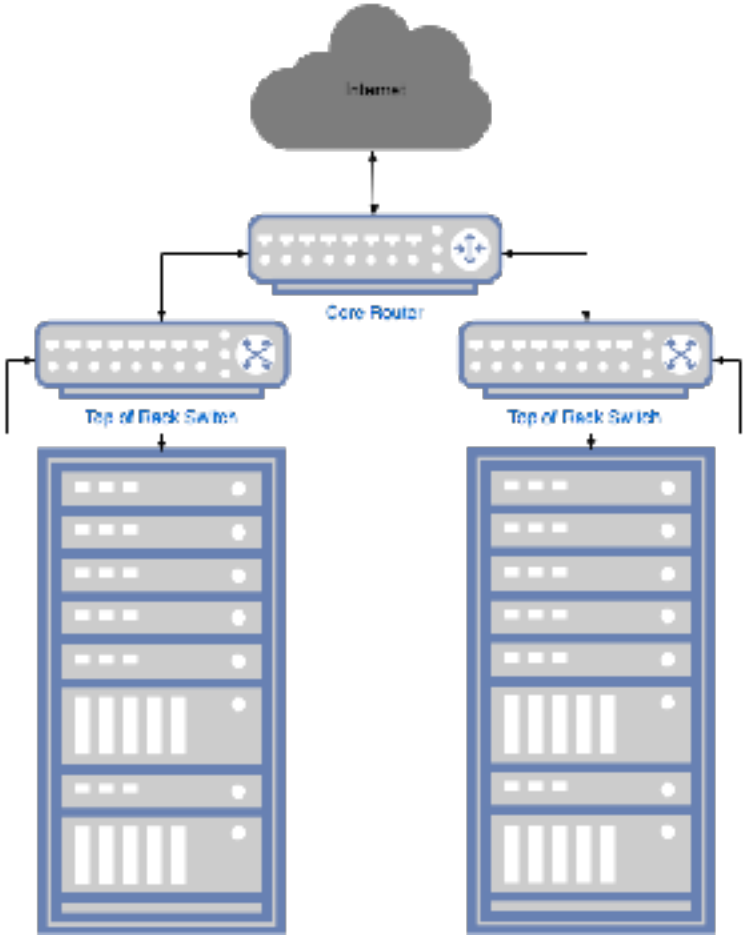
New Service Options

http://www.computerhistory.org/atchm/the-two-napkin-protocol/

Types of Routing



Okay, our little packet is inside the DC


Internet
Core Router
Top of Rack Switch
Top of Rack Switch

Problems

Problems

1. Uneven load

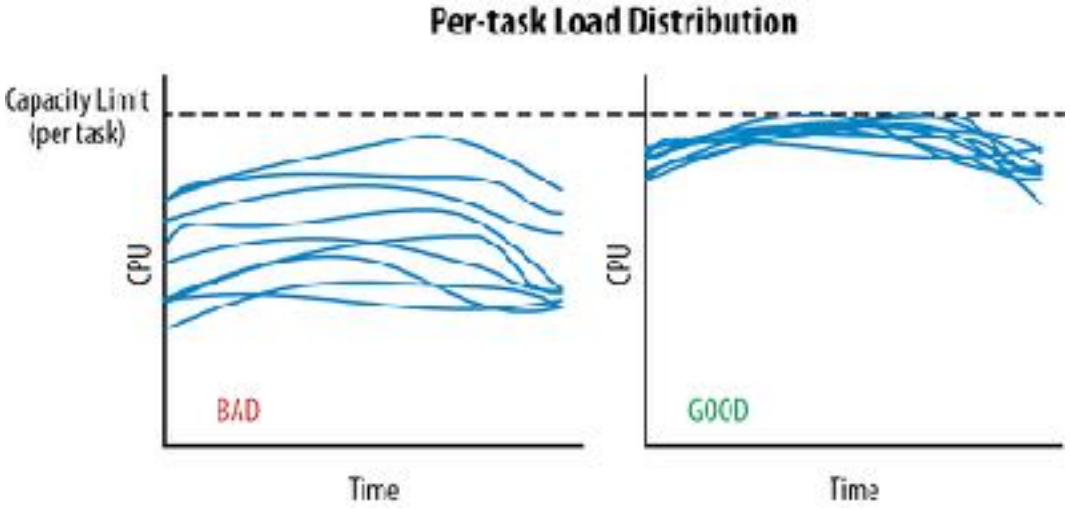
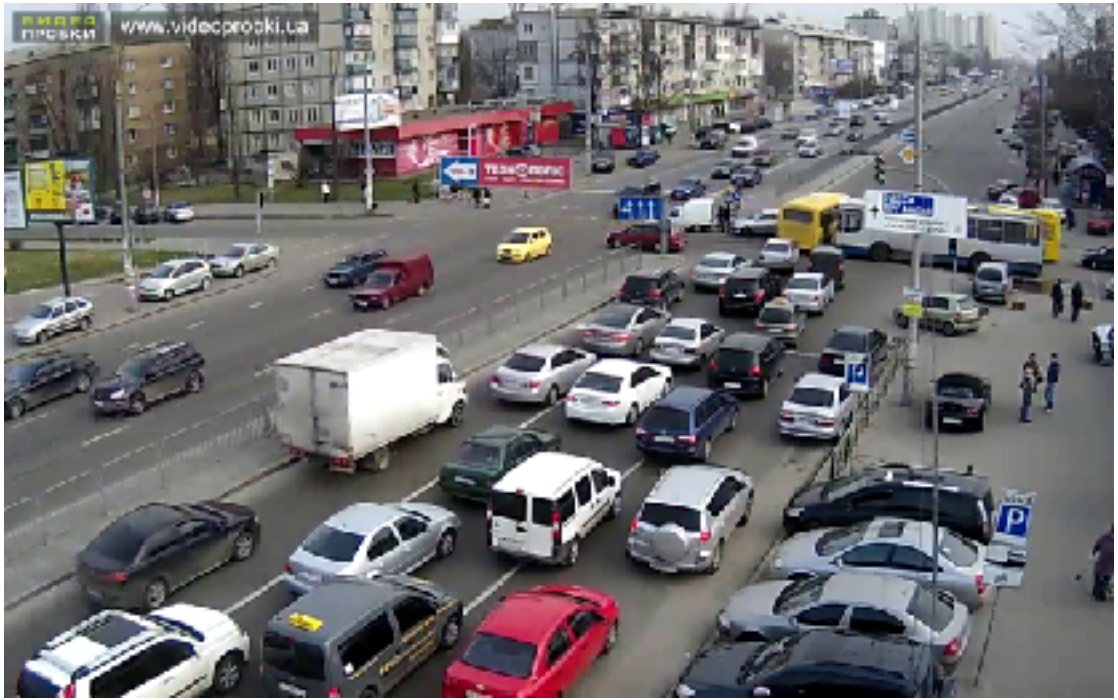


Image credit: https://landing.google.com/sre/book/chapters/load-balancing-datacenter.html

Problems

2. Different kinds of traffic



Problems

3. Per packet load balancing

Image credit: https://flic.kr/p/imuUKx

Problems

4. Heterogenous hardware

Image credit: computer animated film Madagascar

Problems

5. Locality (e.g. for cache) and transport affinity

Image credit: https://www.flickr.com/photos/10361931@N06/4259933727/

Problems

6. DDoS

Image credit: Lockheed Martin F-22 Raptor https://youtu.be/UxgiBJATe9M

Types of DDoS Attacks

	Volumetric Attack	Protocol Attack	Application Attack
What is it?	Saturating the bandwidth of the target.	Exploiting a weakness in the Layer 3 and Layer 4 protocol stack.	Exploiting a weakness in the Layer 7 protocol stack.
How does it cripple the target?	Blocks access to the end-resource.	Consume all the processing capacity of the attacked-target or intermediate critical resources.	Exhaust the server resources by monopolising processes and transactions.
Examples	NTP Amplification, DNS Amplification, UDP Flood, TCP Flood, QUIC HelloRequest amplification	Syn Flood, Ping of Death, QUIC flood	HTTP Flood, Attack on DNS Services

Problems

7. Group change

Image credit: https://flic.kr/p/9Q8yS4

Problems

8. Graceful connection draining

Image credit: German airforce

Load balancing techniques

ECMP

ID (packet) mod N,

ID - some function that produces connection ID, e.g. 5-tuple flow;
N - the number of configured backends.

- Uneven load
- Different kinds of traffic
- Per packet load balancing
- Heterogenous hardware
- Locality
- DDoS
- Group change
- Graceful connection draining

ECMP-CH

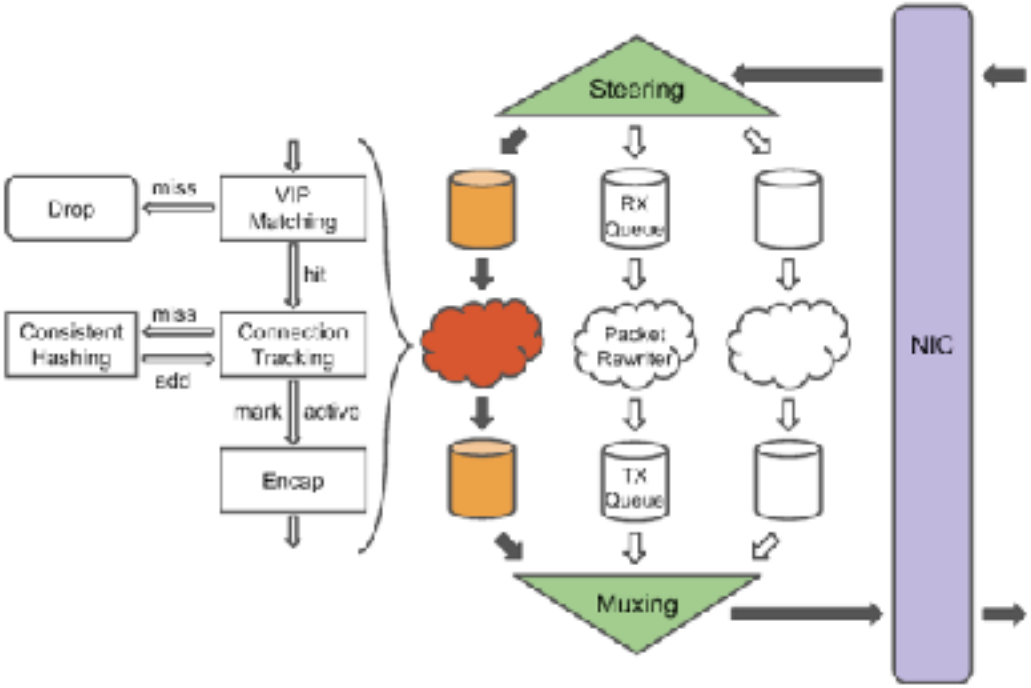
populating the ECMP table not simply with next-hops, but with a slotted table that's made up of redundant next-hops

- Uneven load
- Different kinds of traffic
- Per packet load balancing
- Heterogenous hardware
- Transport affinity
- DDoS
- Group change
- Graceful connection draining

Stateful Load Balancing

- Uneven load
- Different kinds of traffic
- Per packet load balancing
- Heterogenous hardware
- Transport affinity
- DDoS
- Group change
- Graceful connection draining

Google Maglev



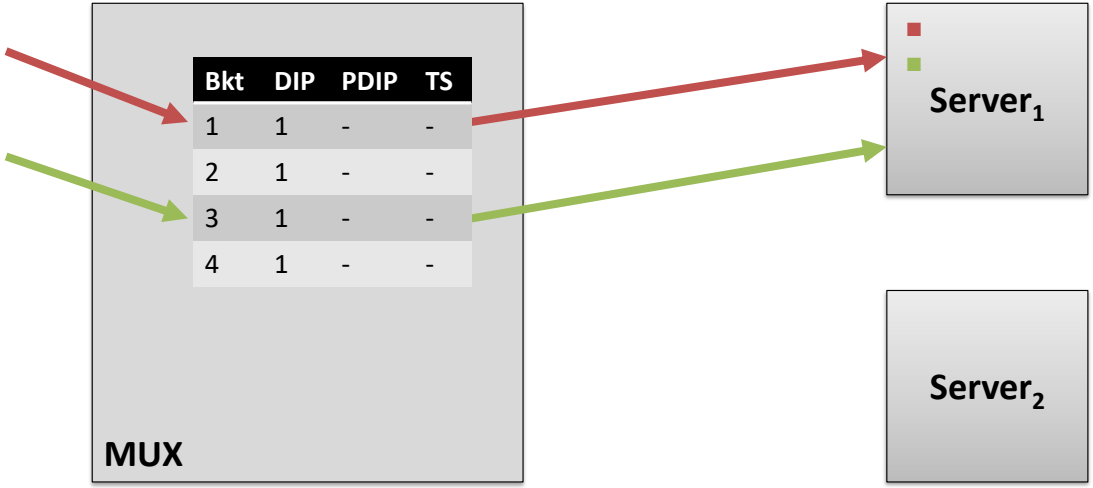
https://ai.google/research/pubs/pub44824

Daisy Chaining a.k.a Beamer

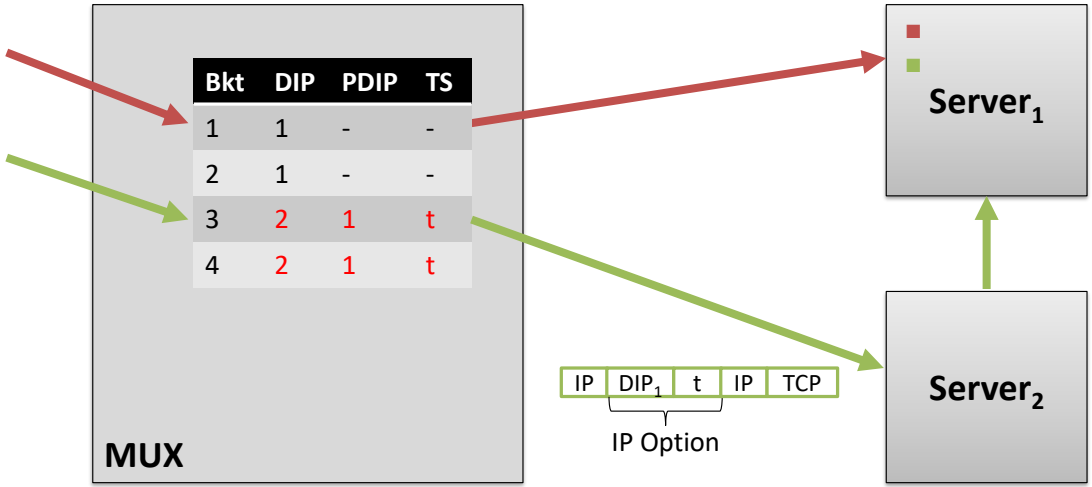
- Beamer muxes do not keep per-connection state; each packet is forwarded independently.
- When the target server changes, connections may break.
- Beamer uses state stored in servers to redirect stray packets.

https://www.usenix.org/conference/nsdi18/presentation/olteanu
https://github.com/Beamer-LB

Beamer at work

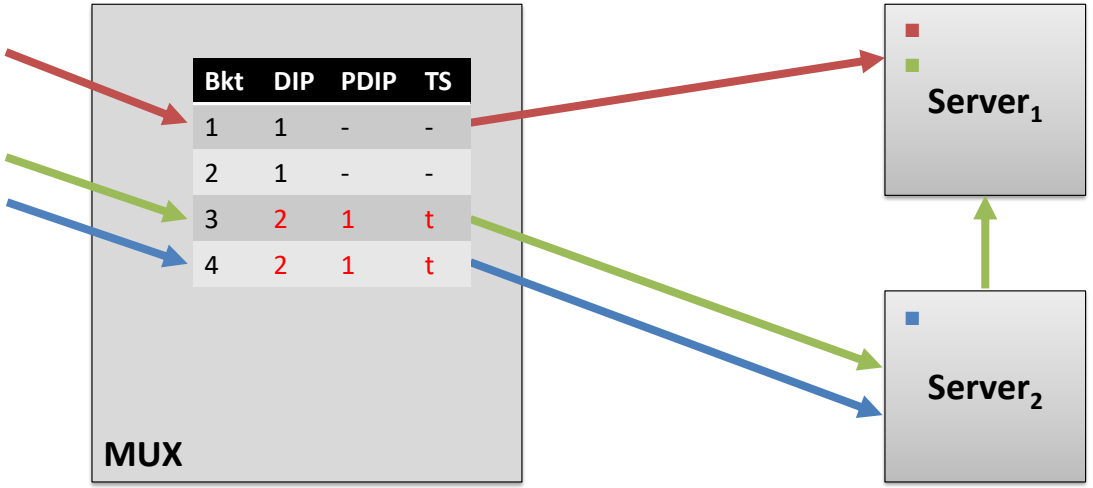


Beamer at work



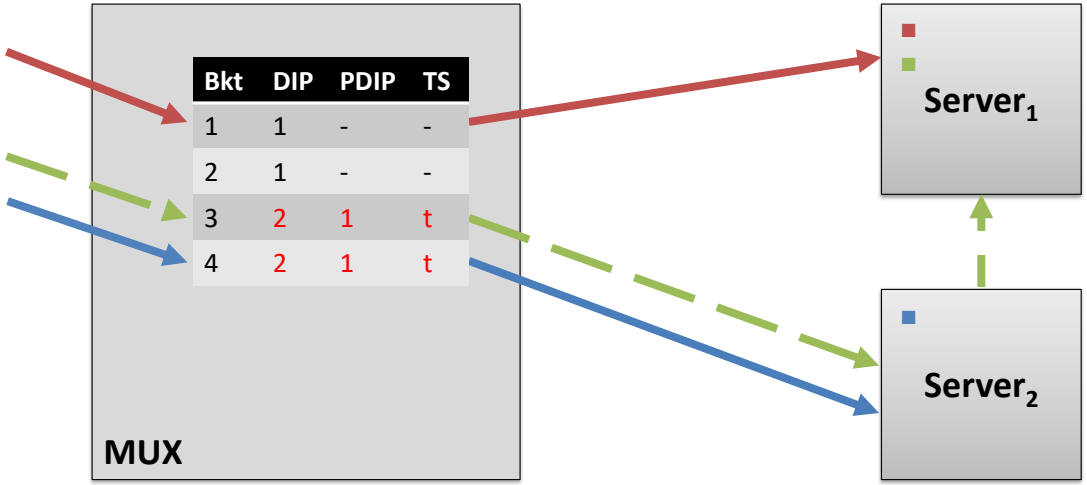
- Packets contain previous server and time of reassignment

Beamer at work



- New connections are handled locally

Beamer at work



- Daisy chained connections die off in time

Daisy Chaining a.k.a Beamer

Uneven load

Different kinds of traffic

Per packet load balancing

Heterogenous hardware

Transport affinity

DDoS

Group change

Graceful connection draining

Performance

Spoilers: could be even better

https://www.usenix.org/conference/nsdi18/presentation/olteanu

https://github.com/Beamer-LB

Also FPGA Packet Processing

- Early experiments with FPGAs
- Smart NICs
- P4 language

Fun (?) Facts

https://www.fastly.com/blog/anatomy-an-iot-botnet-attack

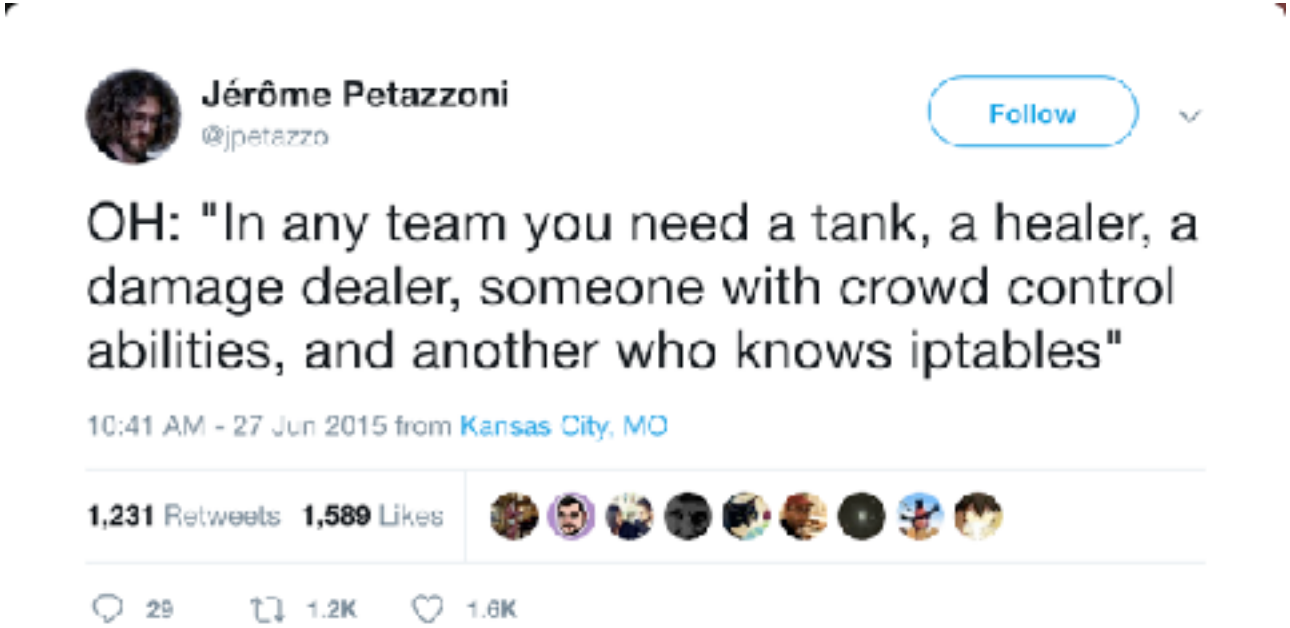
An average IoT device gets infected with malware and launches an attack within 6 minutes of being exposed to the internet.

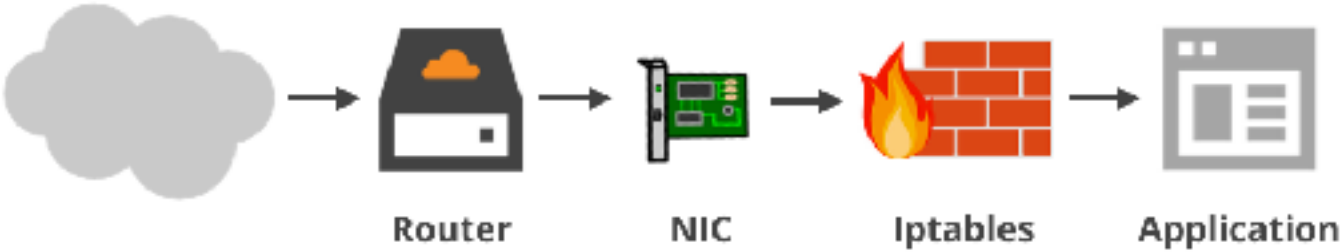
Over the span of a day an average of over 400 login attempts per device; 66 percent of them on average are successful.

Over the span of a day, IoT devices are probed for vulnerabilities 800 times per hour.

DDoS Mitigation

Disclaimer

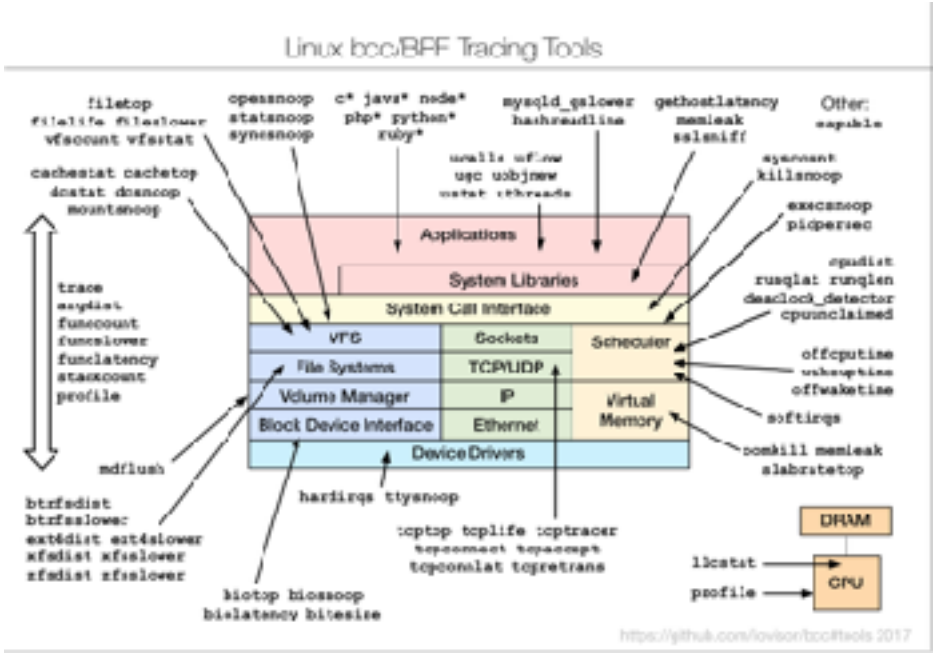

Jérôme Petazzoni
@jpetazzo
Follow
OH: "In any team you need a tank, a healer, a damage dealer, someone with crowd control abilities, and another who knows iptables"
10:41 AM - 27 Jun 2015 from Kansas City, MO
1,231 Retweets 1,589 Likes
29
1.2K
1.6K

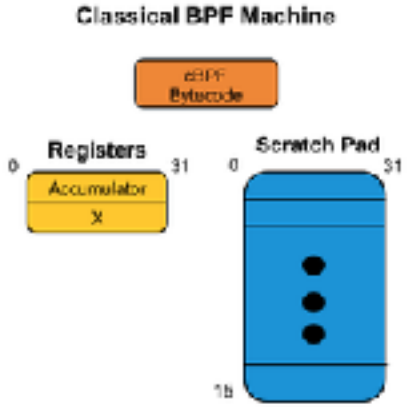

Router
NIC
Iptables
Application

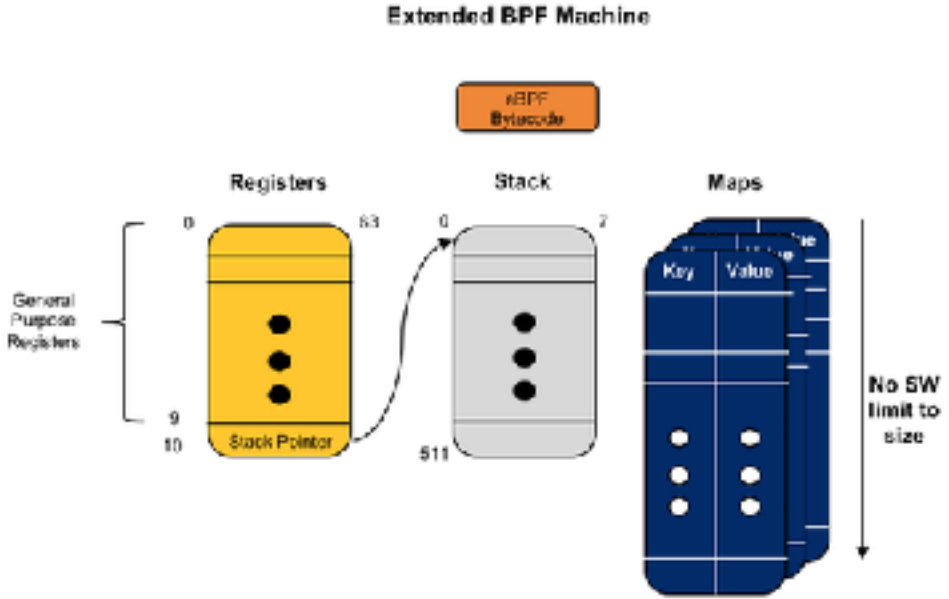
Disclaimer

BPF and eBPF

- Low overhead sandboxed user-defined bytecode running in kernel
- Written in a subset of C, compiled by clang llvm
- It can never crash, hang or interfere with the kernel negatively
- If you run Linux 3.15 or newer, you already have it
- Great intro from Brendan Gregg: http://www.brendangregg.com/ebpf.html





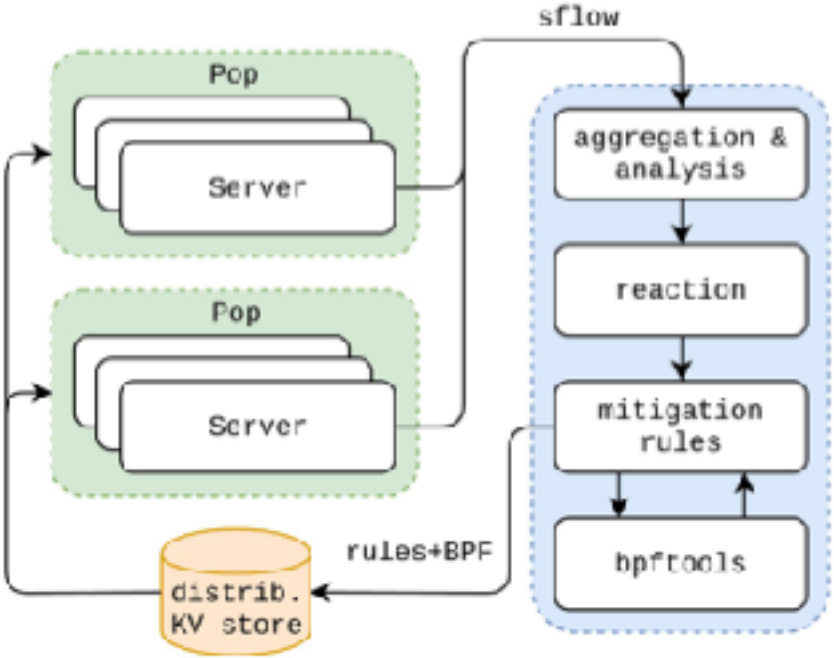


https://www.netronome.com/blog/bpf-ebpf-xdp-and-bpfilter-what-are-these-things-and-what-do-they-mean-enterprise/

End of disclaimer

DDoS Mitigation Pipeline

Gatebot


sflow
Pop
Server
Pop
Server
aggregation & analysis
reaction
mitigation rules
bpftools
rules+BPF
distrib. KV store

iptables

- Initially it was the only tool to filter traffic
- Leveraged modules ipsets, hashlimit, connlimit
- With the xt_bpf module it was possible to specify complex filtering rules
- But we soon started experiencing IRQ storms during big attacks
- All CPUs were busy dropping packets, userspace applications were starving of CPU

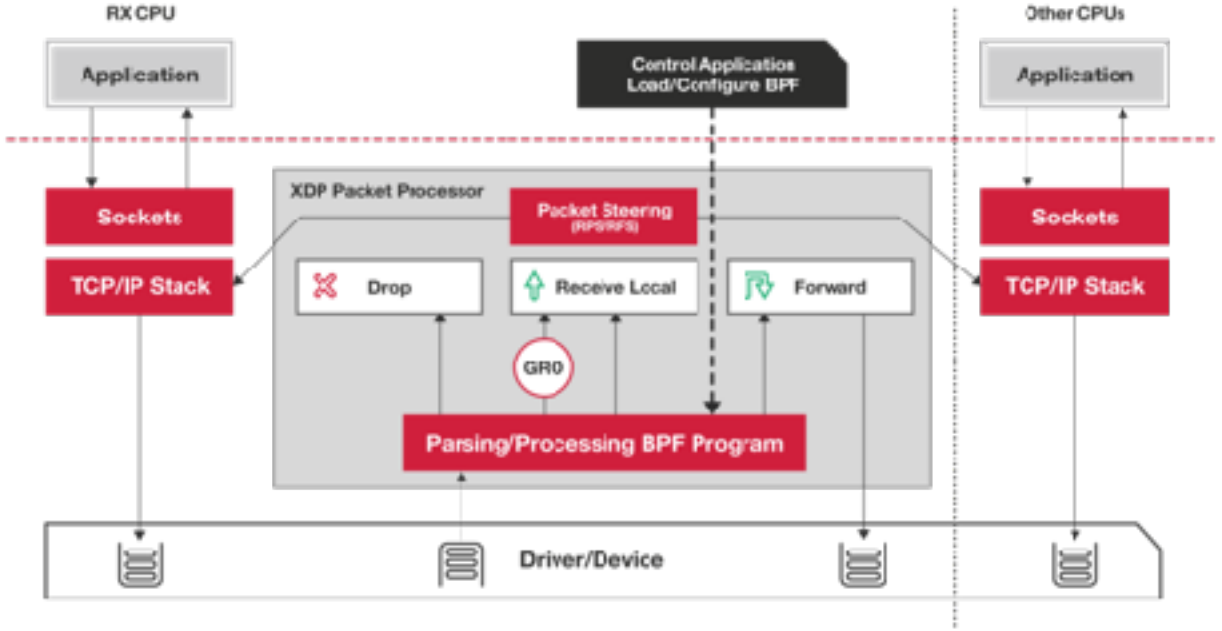
Userspace Offload a.k.a. Kernel Bypass

- Network traffic is offloaded to userspace before it hits the Linux network stack
- Allows to run BPF in userspace
- An order of magnitude faster than iptables (5M pps)

- Requires one or more CPUs to busy poll the NIC event queue
- Reinjecting packets in the network stack is expensive
- Hardware dependant

XDP to the rescue!

XDP Packet Processing Overview



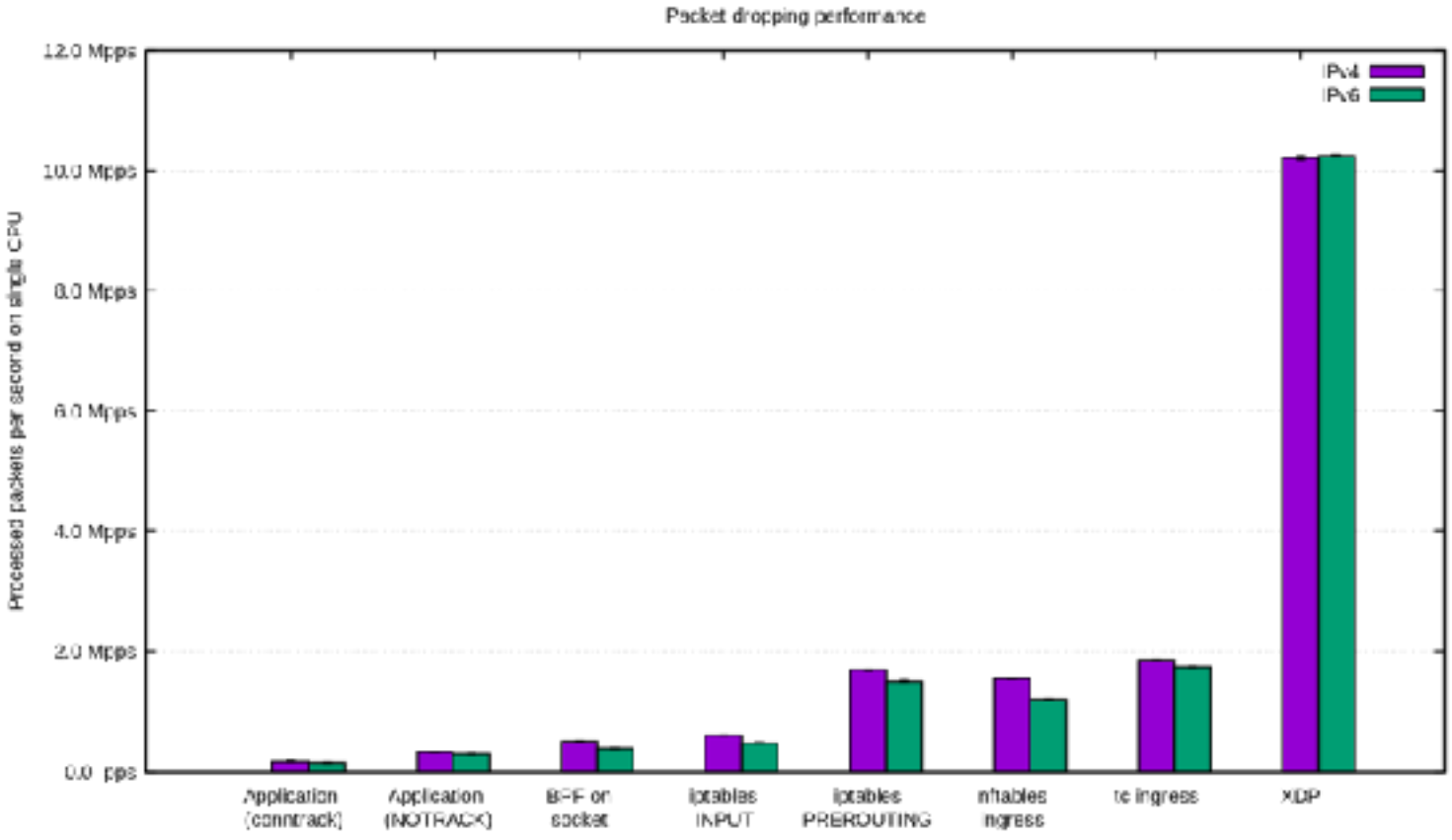
+AF_XDP since 4.19

https://www.iovisor.org/technology/xdp

```c
SEC("xdp1")
  int xdp_prog(struct xdp_md * ctx)
  {
    void * data     = (void *)(long)ctx->data;
    void * data_end = (void *)(long)ctx->data_end;
    int ret;

    ret = rule_1(data, data_end);
    if (ret != XDP_PASS) // multiple if statements
    {
      return ret;
    }

    ret = rule_2(data, data_end); // one for each rule
    if (ret != XDP_PASS)
    {
      return ret;
    }

    ...
    return XDP_PASS; // if none of them match - the packet is accepted
  }
```

```
1 static inline int rule_1 (void * data, void * data_end)
2 {
3   if (!condition_1)
4   {
5     return XDP_PASS;
6   }
7   if (!condition_2) // multiple if conditions
8   {
9     return XDP_PASS;
10  }
11
12  ...
13
14  update_rule_counters(1); // do additional work
15  sample_packet(data, data_end);
16
17  return XDP_DROP;
18 }
```

```c
// eBPF map shared with userpace
struct bpf_map metrics_map __section("maps") = {
    .type        = BPF_MAP_TYPE_PERCPU_ARRAY,
    .key_size    = sizeof(unsigned int),
    .value_size = sizeof(rule_metrics_t),
    .max_entries    = STATS_MAP_SIZE,
};


static inline void update_rule_counters(int rule_id)
{
  long * value = bpf_map_lookup_elem(&metrics_map, &rule_id);

  if (value)
  {
    *value += 1;
  }
}
```



https://blog.cloudflare.com/how-to-drop-10-million-packets/

A perfect match: XDP both for load balancing and DDoS mitigation <3

XDP L4LB with daisy chaining using encapsulation

- Uneven load ✓
- Different kinds of traffic ✓
- Per packet load balancing ✓
- Heterogenous hardware ✓
- Transport affinity ✓
- DDoS ✓
- Group change ✓
- Graceful connection draining ✓
- Performance ✓

And They Lived Happily Ever After

But… DPDK?

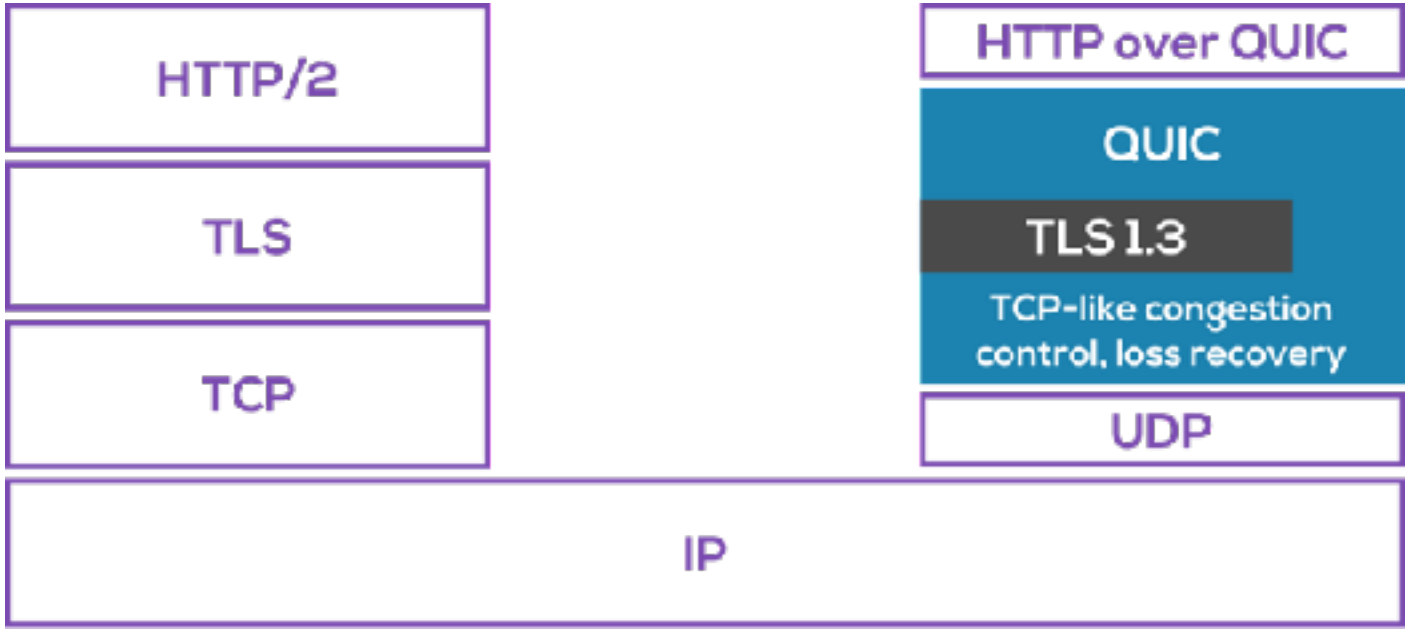
Advantages of XDP over DPDK

- Allows option of busy polling or interrupt driven networking
- No need to allocate huge pages
- Dedicated CPUs are not required, user has many options on how to structure the work between CPUs
- No need to inject packets into the kernel from a third party user space application
- No special hardware requirements
- No need to define a new security model for accessing networking hardware
- No third party code/licensing required

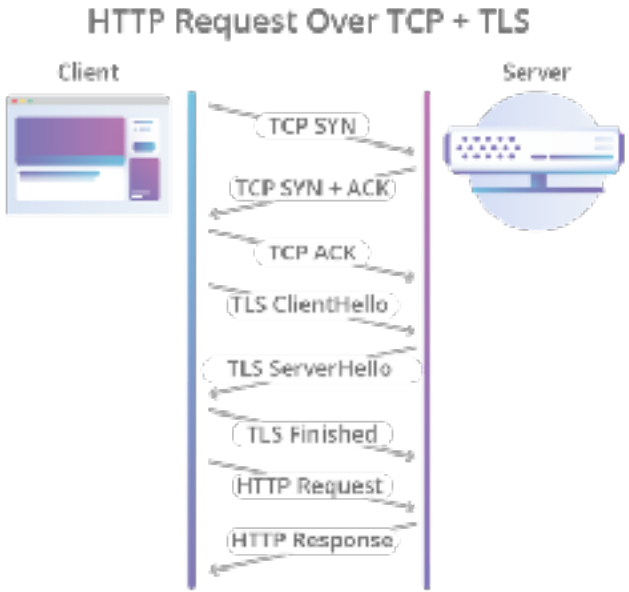
https://github.com/iovisor/bpf-docs/blob/master/Express_Data_Path.pdf

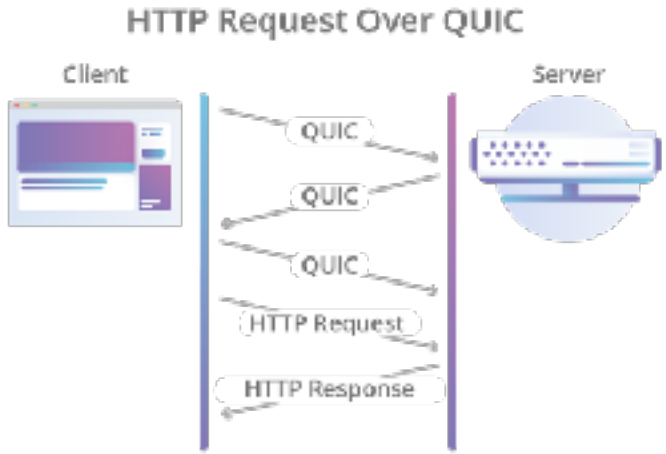
Bonus part

(Quick UDP Internet Connections)

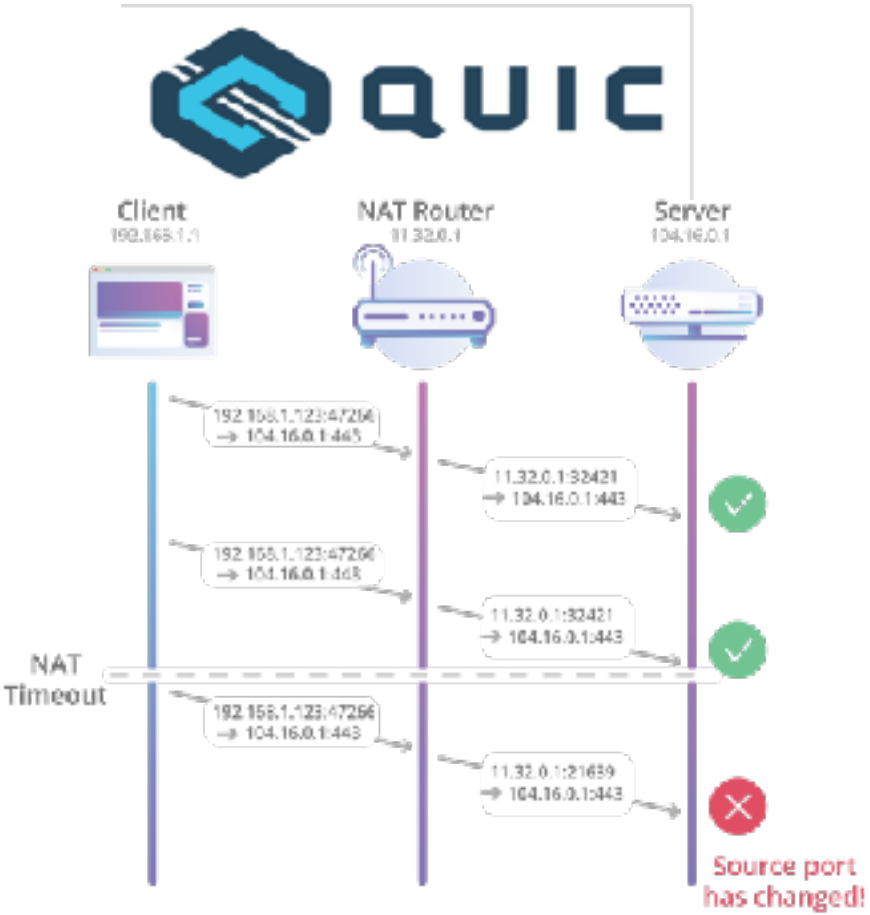


QUIC





https://blog.cloudflare.com/the-road-to-quic/



https://blog.cloudflare.com/the-road-to-quic/

L4LB to the rescue!

Thank you!

@advance_lunge