Simple DNS Server with BPF

Martin Lau, Raman Shukhau

BPF “usual” use-cases

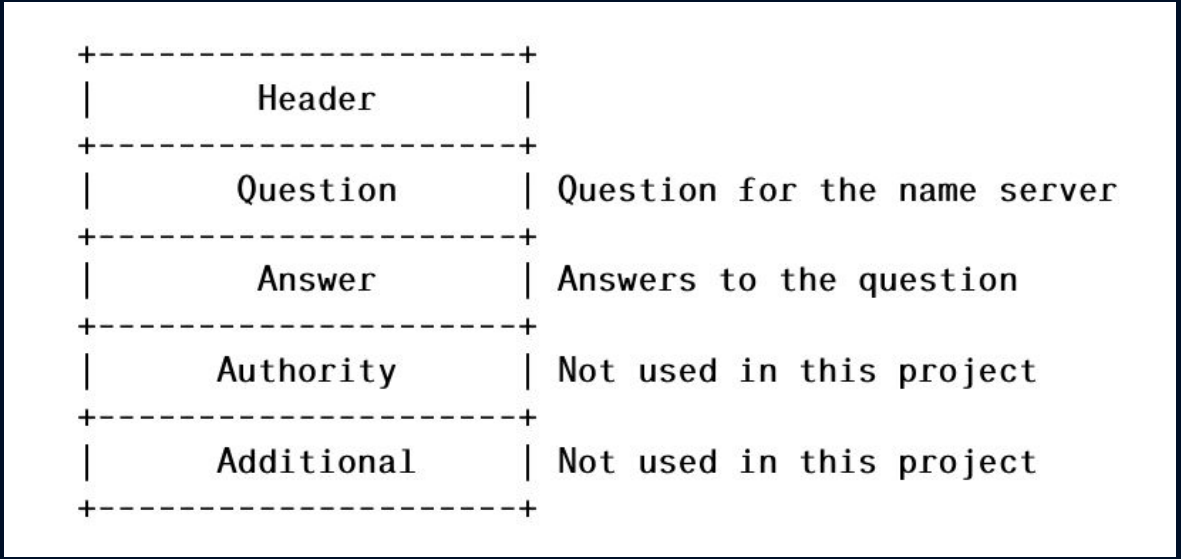
- Packet Filtering & Processing
- Custom Congestion Control
- Container Networking
- Performance Profiling & Observability
- Tracing & Debugging
- Security & Monitoring
- Custom Schedulers & I/O Accelerators
- GPU Profiling
- Kernel Debugging

BPF can also reply to network packets

What Network Protocol is a good use-case?

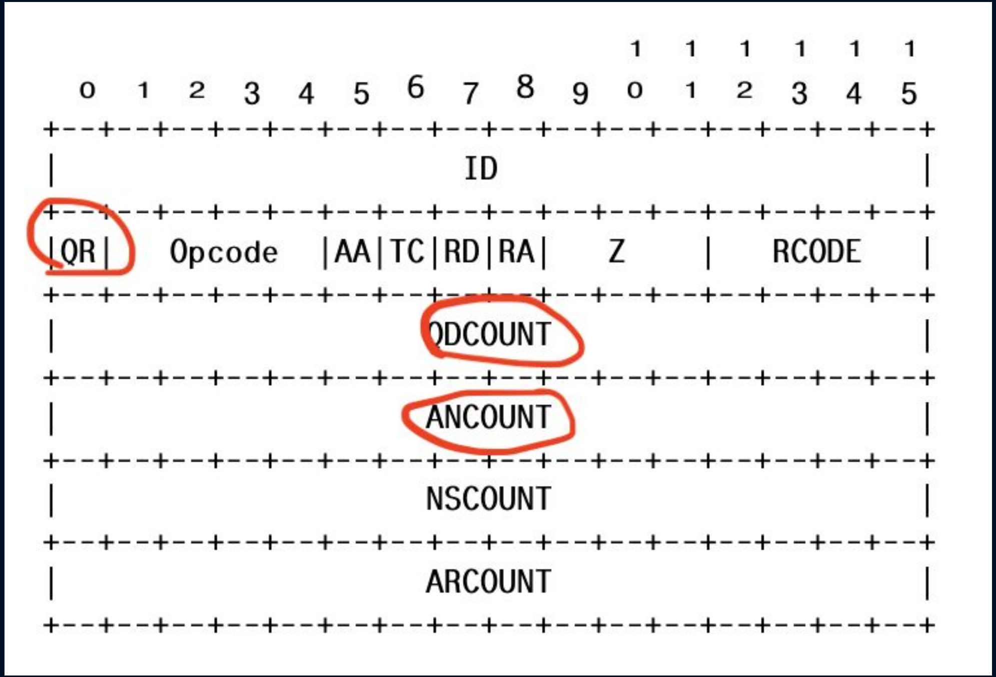
DNS

DNS Structure Request



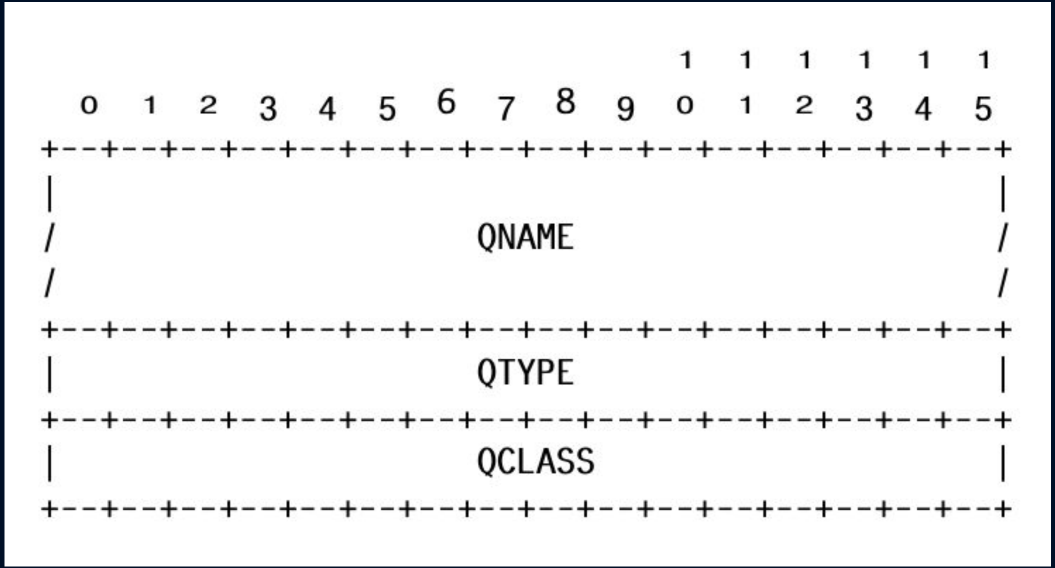
```
+---------------------+
|        Header       |
+---------------------+
|       Question      | Question for the name server
+---------------------+
|        Answer       | Answers to the question
+---------------------+
|      Authority      | Not used in this project
+---------------------+
|      Additional     | Not used in this project
+---------------------+
```

DNS Structure Request: Header

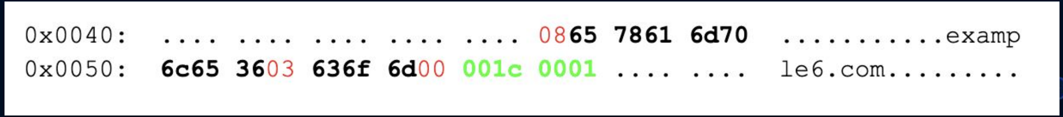


```
                                1  1  1  1  1  1
  0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                      ID                       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|QR|   Opcode  |AA|TC|RD|RA|   Z    |   RCODE   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    QDCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    ANCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    NSCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                    ARCOUNT                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

DNS Structure Request: Question



```
                                    1  1  1  1  1  1
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                                               |
    /                     QNAME                     /
    /                                               /
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                     QTYPE                     |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
    |                     QCLASS                    |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```



```
0x0040:  .... .... .... .... .... 0865 7861 6d70  ...........examp
0x0050:  6c65 3603 636f 6d00 001c 0001 .... ....  le6.com.........
```

In this example host will be encoded as:

- 0x08 + **example6** +
- 0x03 + **com** +
- 0x00 +
- 0x001c (type) +
- 0x0001 (class)

What program types we can use?

- Cgroup Types - Not Supported
  - routing decision is made at this point
  - change to SKB structure limited or not allowed
- TC/TCX - Supported
  - Can modify SKB and use bpf_redirect
- XDP - Supported
  - Can change XDP struct and do XDP_TX

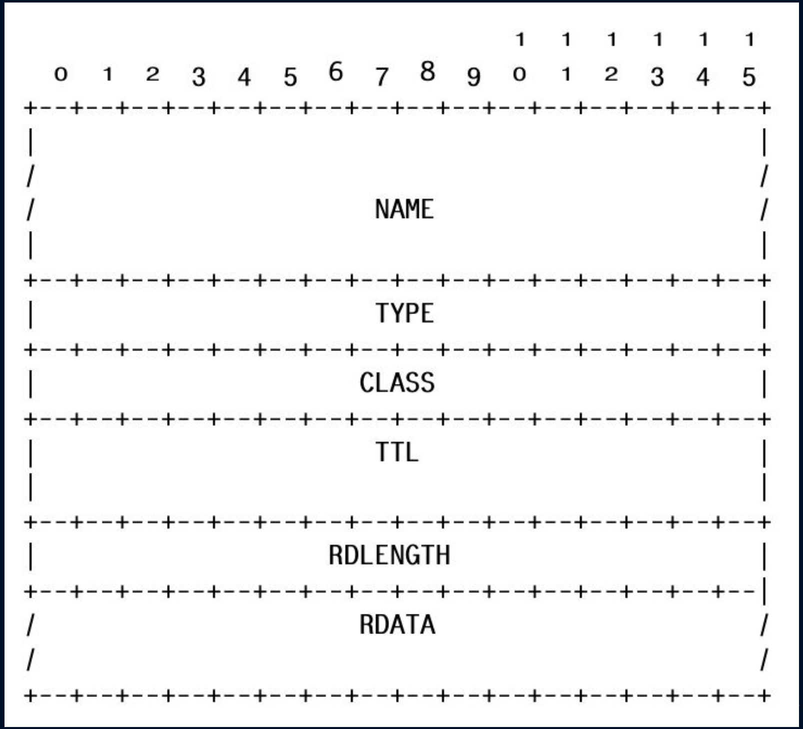
What do we need to do?

- Swap Source and Destination
- Create Answer section
- Adjust packet size
- Calculate new Checksum
- Flip packet “direction”

Preparing DNS Response: Src/Dst

- Swap MAC address in ETH header
- Swap Src/Dst IP in IPv4/IPv6 header
- Swap Ports UDP

Preparing DNS Response: Answer



```
                                1  1  1  1  1  1
  0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                               |
/                                               /
/                      NAME                     /
|                                               |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                      TYPE                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                     CLASS                     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                      TTL                      |
|                                               |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                   RDLENGTH                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--|
/                     RDATA                     /
/                                               /
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Little difference between A (IPv4) & AAAA (IPv6) answers:

Type = 1 (A IPv4) **or** 28 (AAAA IPv6)
RDLength = 4 **or** 16
RDATA = IPv4 **or** IPv6 address

Preparing DNS Response: Packet Size

- sizeof(DNS Header + DNS Name + Domain Info + **DNS Answer(s))**
- Adjust SKB/XDP packet size
- Set new UDP/IP packet length

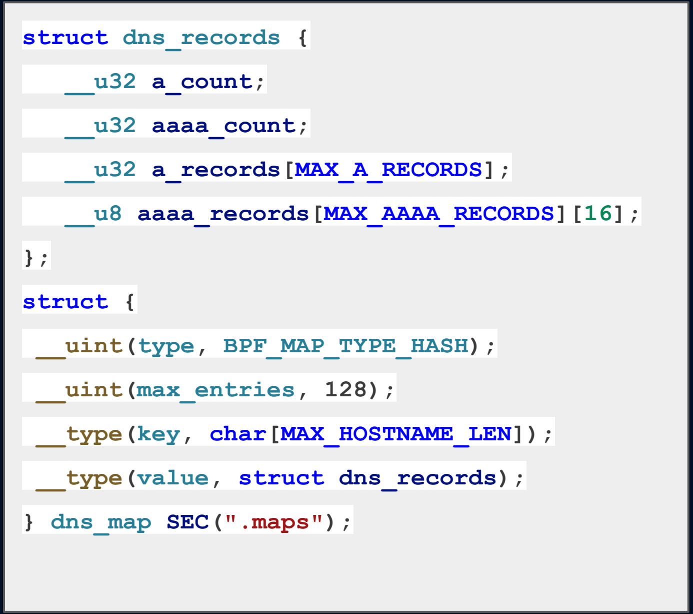
Preparing DNS Response: Checksum

CSUM chain of the following items:

- UDP header
- DNS Header + DNS Name + DNS Answer(s)
- ip6h->saddr
- ip6h->daddr
- UDP packet len
- IPPROTO_UDP

csum_fold at the end to get u16 value

How can we store DNS Records?



```
struct dns_records {
    __u32 a_count;
    __u32 aaaa_count;
    __u32 a_records[MAX_A_RECORDS];
    __u8 aaaa_records[MAX_AAAA_RECORDS][16];
};
struct {
  __uint(type, BPF_MAP_TYPE_HASH);
  __uint(max_entries, 128);
  __type(key, char[MAX_HOSTNAME_LEN]);
  __type(value, struct dns_records);
} dns_map SEC(".maps");
```

… or can use BPF LPM_TRIE Map, but need to “invert” hostname in BPF for lookup.

How can we use DNS BPF?

- Resolving external queries
  - DNS DDoS Protection (XDP)

- Resolving local queries (TC)
  - Alternative to local DNS cache server
  - Test mocking

- Alternative for /etc/hosts (TC)
  - More flexible support with similar effort

Use Case: Resolving external queries

⊕ No burden with service maintenance

⊖ Only simple use-cases supported

⊕ better performance with XDP DNS vs. DNS Server

BPF XDP

Queries per second:	286K
AVG Latency (ms):	0.205
P0 Latency (ms):	0.047
P100 Latency (ms):	4.195

Unbound DNS Server

Queries per second:	117K
AVG Latency (ms):	0.772
P0 Latency (ms):	0.062
P100 Latency (ms):	6.785

Use Case: Resolving local queries

- No messing with overriding /etc/resolv.conf
- Same “no burden with maintenance” point
- Same better performance in comparison to local DNS server

BPF TC

Queries per second:	211K
AVG Latency (ms):	0.008
P0 Latency (ms):	0.004
P100 Latency (ms):	0.891

Unbound DNS Server

Queries per second:	179K
AVG Latency (ms):	0.531
P0 Latency (ms):	0.009
P100 Latency (ms):	1.424

Use Case: Alternative for /etc/hosts

- ⊕ Easy to modify list of hosts with BPF Map
- ⊕ Observability with stats/counters/logs
- ⊕ Multiple IP addresses per Entry supported
- ⊖ /etc/hosts is still expected to be faster, as no network involved

Live Demo
error
no connection

Q&A

Thank you