Cilium Introduction and Improvement

Jiang Wang, System Technologies and Engineering, ByteDance

AKRAINO

ByteDance 字节跳动

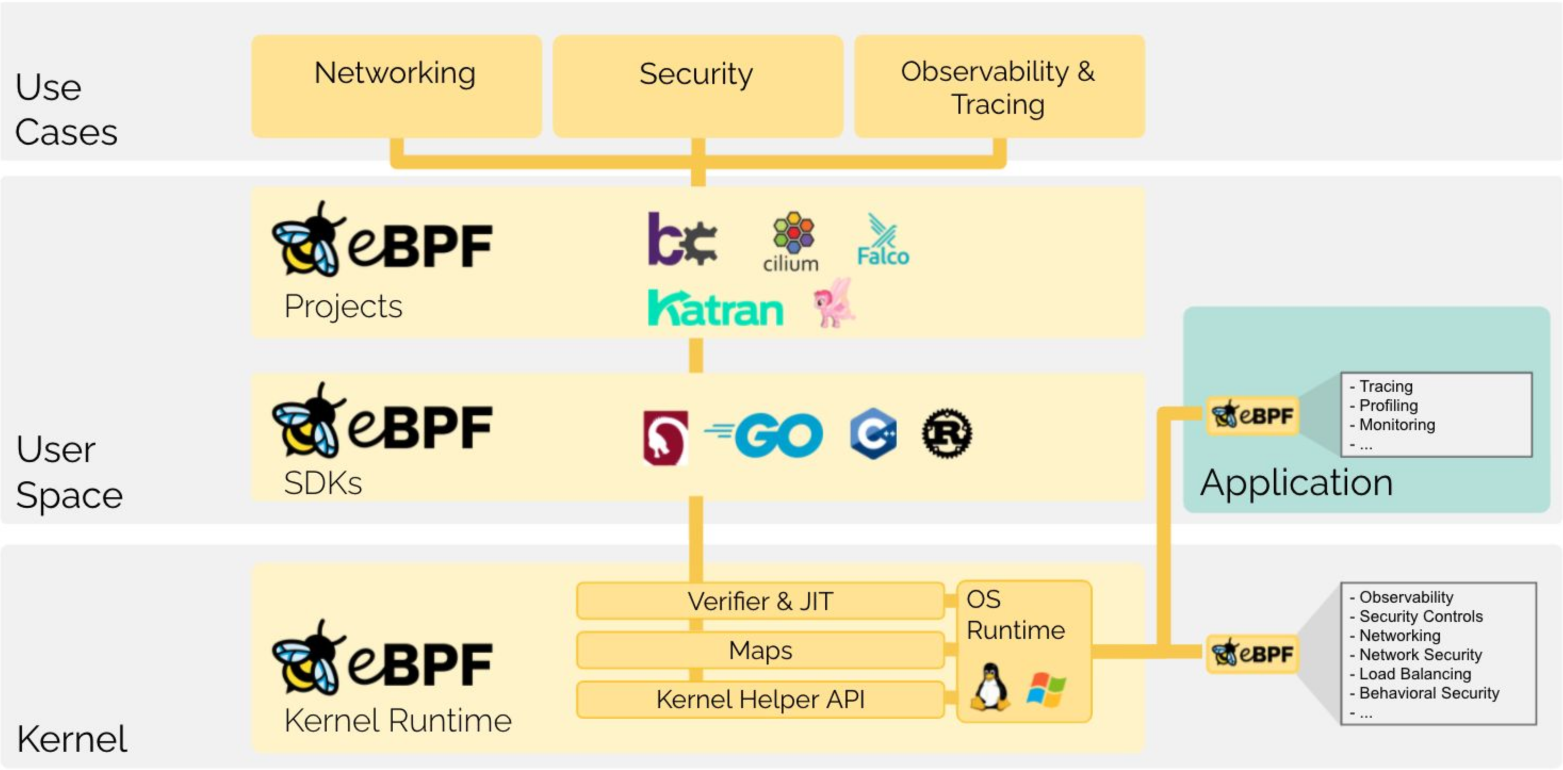
ByteDance 字节跳动

1. Introduction of eBPF
2. Introduction of Cilium
3. Cilium data path for DSR
4. Cilium DSR improvement
5. Q&A

ByteDance 字节跳动

What is eBPF

- An instruction set based or
- RISC, JIT
- A sandboxed program in kerr
- Lightweight, safe, portable



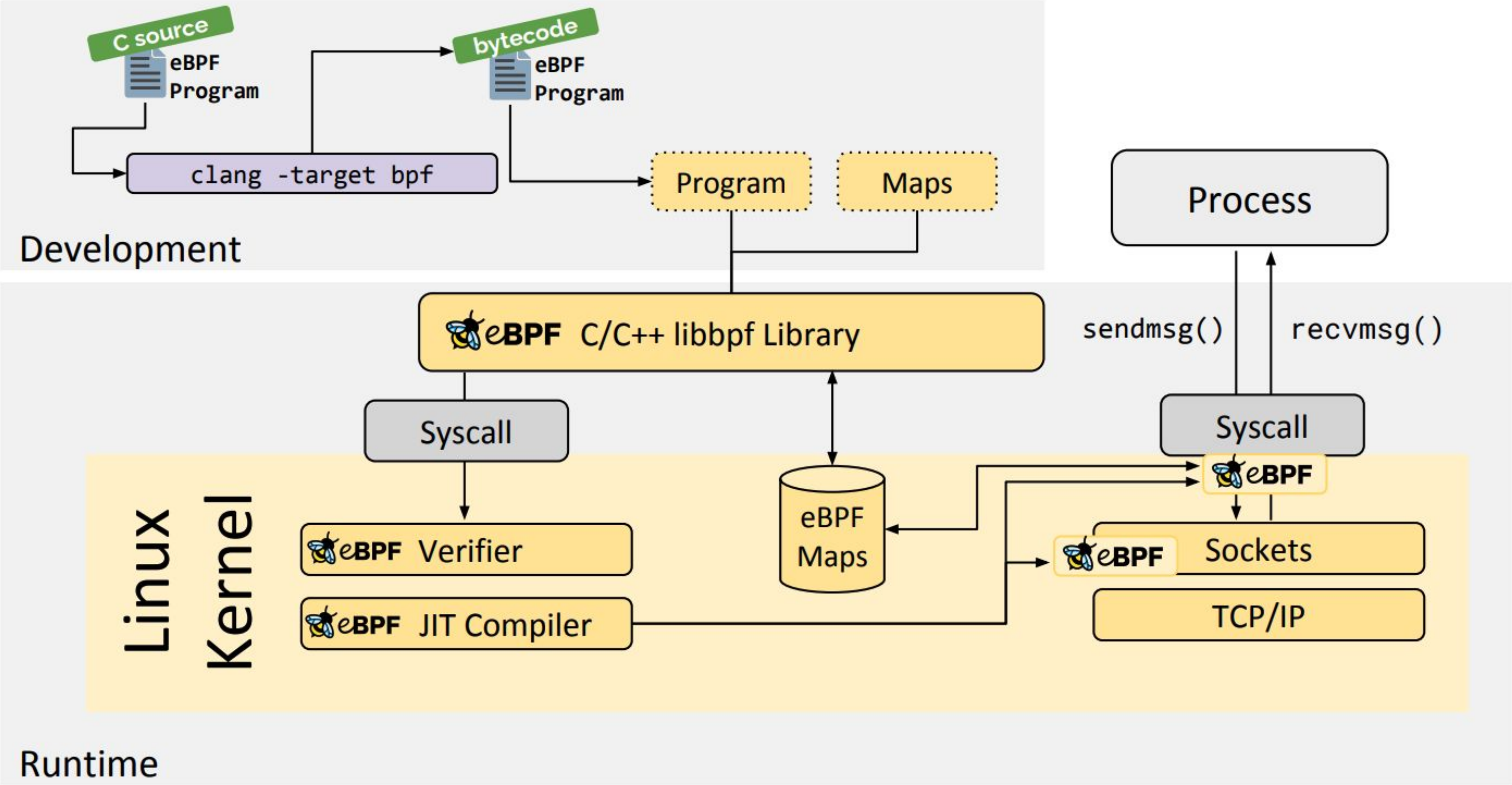
https://ebpf.io/

Why eBPF

- Hook pre-defined kernel functions
- A programmable interface, much more powerful than procfs/sysfs/syscall
- Safer, lighter and finer-granularity than kernel modules
- Compile-Once Run-Everywhere (?)

eBPF Architecture

- eBPF instruction set
- eBPF in-kernel verifier and JIT
- eBPF syscalls via bpf()
- Kernel components: programs, maps, helper functions, iterators, BTF, XDP etc.
- LLVM support
- Libbpf, bpftools


C source
eBPF Program
bytecode
eBPF Program
clang -target bpf
Program
Maps
Process
Development
eBPF C/C++ libbpf Library
sendmsg()
recvmsg()
Syscall
Syscall
eBPF
Linux Kernel
eBPF Verifier
eBPF Maps
eBPF Sockets
eBPF JIT Compiler
TCP/IP
Runtime

AKRAINO

eBPF Example for Tracing

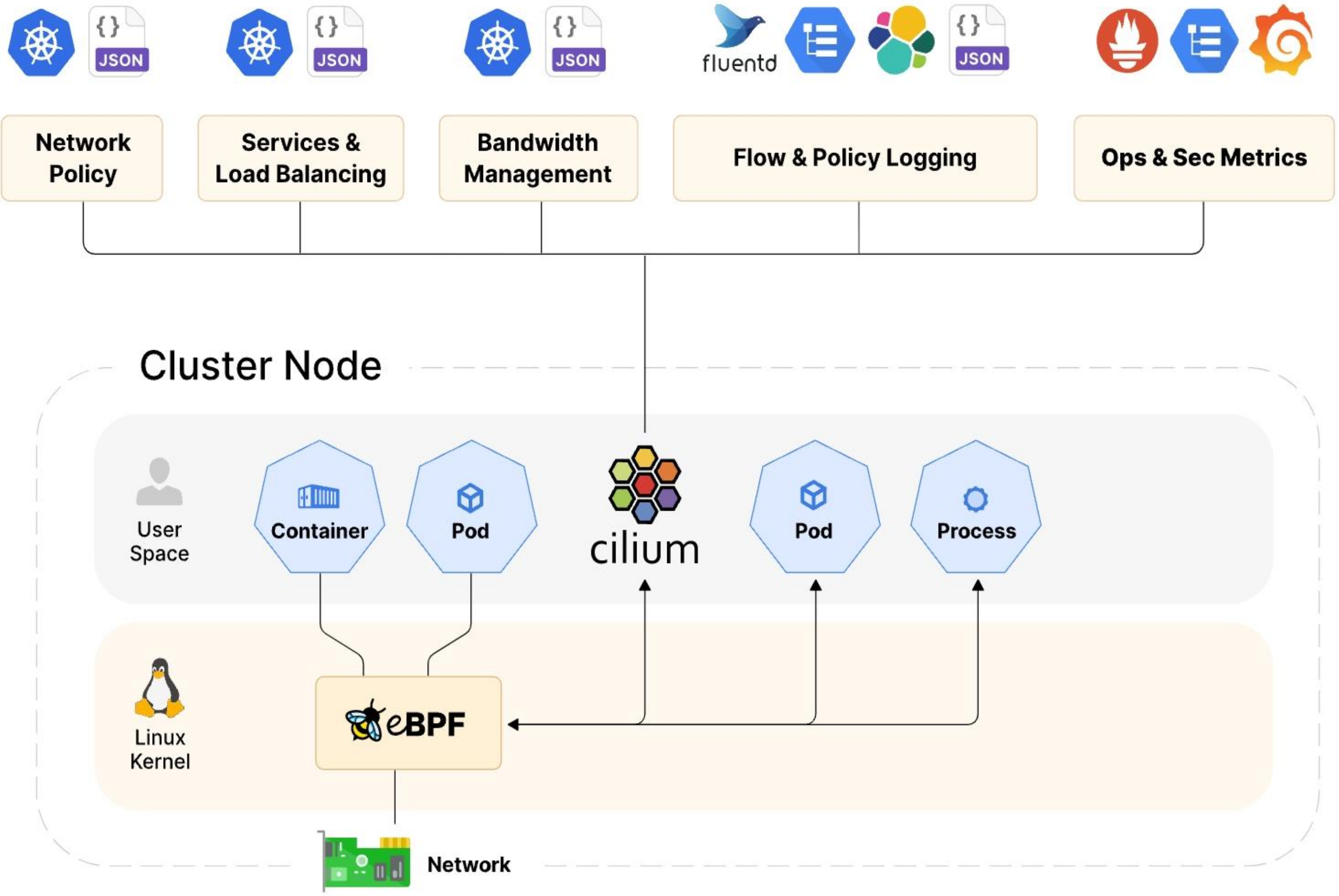
```
# ./tcpconnect -t
TIME(s)  PID    COMM         IP SADDR            DADDR            DPORT
31.871   2482   local_agent  4  10.103.219.236   10.251.148.38    7001
31.874   2482   local_agent  4  10.103.219.236   10.101.3.132     7001
31.878   2482   local_agent  4  10.103.219.236   10.171.133.98    7101
90.917   2482   local_agent  4  10.103.219.236   10.251.148.38    7001
90.928   2482   local_agent  4  10.103.219.236   10.102.64.230    7001
90.938   2482   local_agent  4  10.103.219.236   10.115.167.169   7101
```

https://github.com/iovisor/bcc/blob/master/tools/tcpconnect_example.txt

Cilium

- open source software providing, securing and observing network connectivity between container workloads
- High scale, low overhead

https://cilium.io/



AKRAINO

ByteDance 字节跳动

Cilium at ByteDance

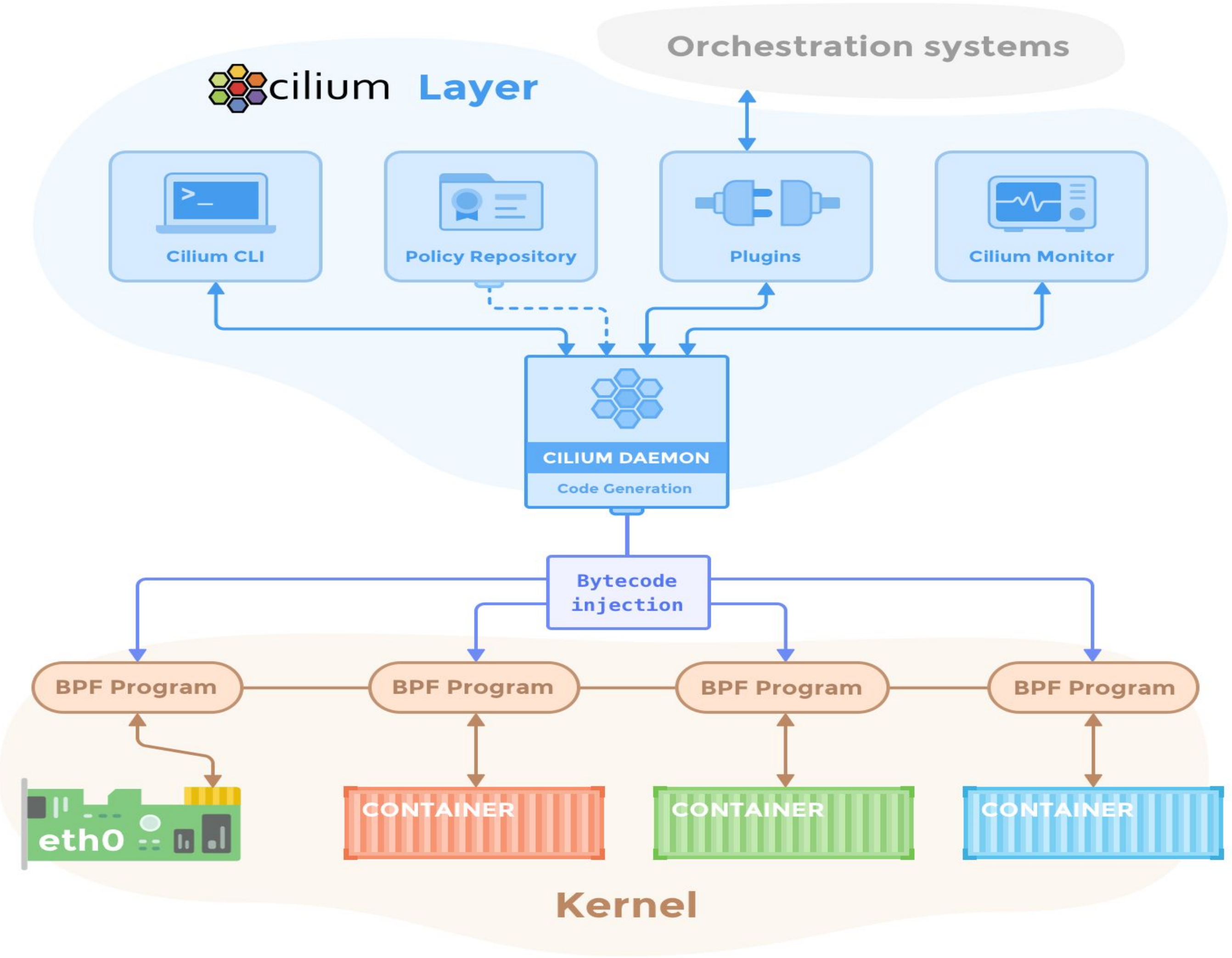
- Use Cilium as CNI for Kubernetes to replace kube-proxy
- Better performance, less cpu usage
- Deployed on a few edge networks.

Hubble

- Hubble is a fully distributed networking and security observability platform
- built on top of Cilium and eBPF to enable deep visibility
- Can answer the following questions:
  - Service dependencies & communication map
  - Network monitoring & alerting
  - Application monitoring
  - Security observability

Cilium Components

- Agent (Daemon)
- Client (CLI)
- Operator
- CNI Plugin



ByteDance 字节跳动

Cilium Terminologies

- Labels
  - E.g io.cilium.mykey=myvalue
  - Label Source: Kubernetes or container
- Endpoint
  - Unique IP
  - By default, assign both IPv4 and IPv6
- Identification
  - Uint32 value

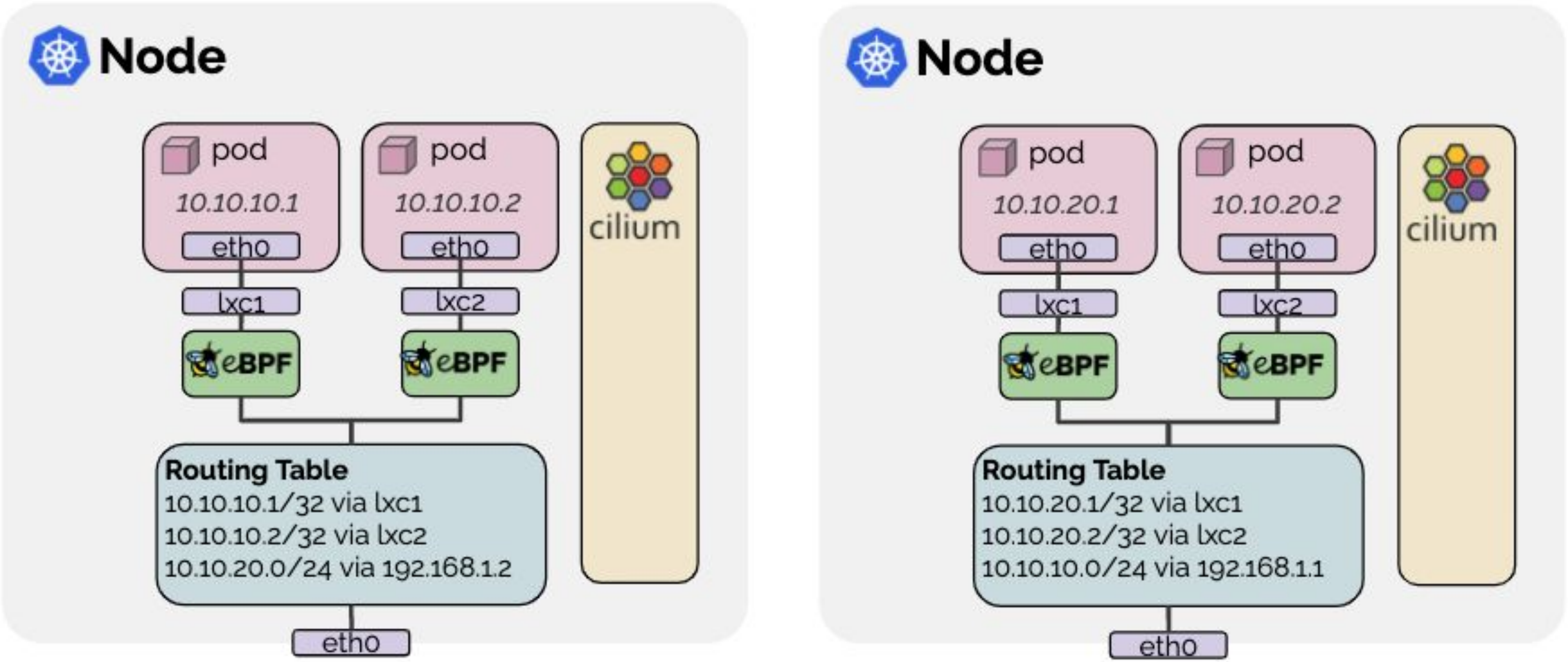
```
0x00000001 - 0x000000FF (1           to 2^8  - 1)        => reserved identities
0x00000100 - 0x0000FFFF (2^8         to 2^16 - 1)        => cluster-local identities
0x00010000 - 0x00FFFFFF (2^16        to 2^24 - 1)        => identities for remote clusters
0x01000000 - 0x0100FFFF (2^24        to 2^24 + 2^16 - 1) => identities for CIDRs (node-local)
0x01010000 - 0xFFFFFFFF (2^24 + 2^16 to 2^32 - 1)        => reserved for future use
```

ByteDance 字节跳动

Networking mode 1: Encapsulation/tunneling

- Encapsulation (or tunneling mode)
- Vxlan or Geneve
- Requirement:
  - the underlying network and firewalls must allow encapsulated packets
- Advantage:
  - Simplicity
  - Addressing space
  - Auto-configuration
  - Identity context in the pkt
- Disadvantage:
  - MTU Overhead

Networking mode 2: native routing



Node

pod
10.10.10.1
eth0
lxc1
eBPF

pod
10.10.10.2
eth0
lxc2
eBPF

cilium

Routing Table
10.10.10.1/32 via lxc1
10.10.10.2/32 via lxc2
10.10.20.0/24 via 192.168.1.2

eth0

Node

pod
10.10.20.1
eth0
lxc1
eBPF

pod
10.10.20.2
eth0
lxc2
eBPF

cilium

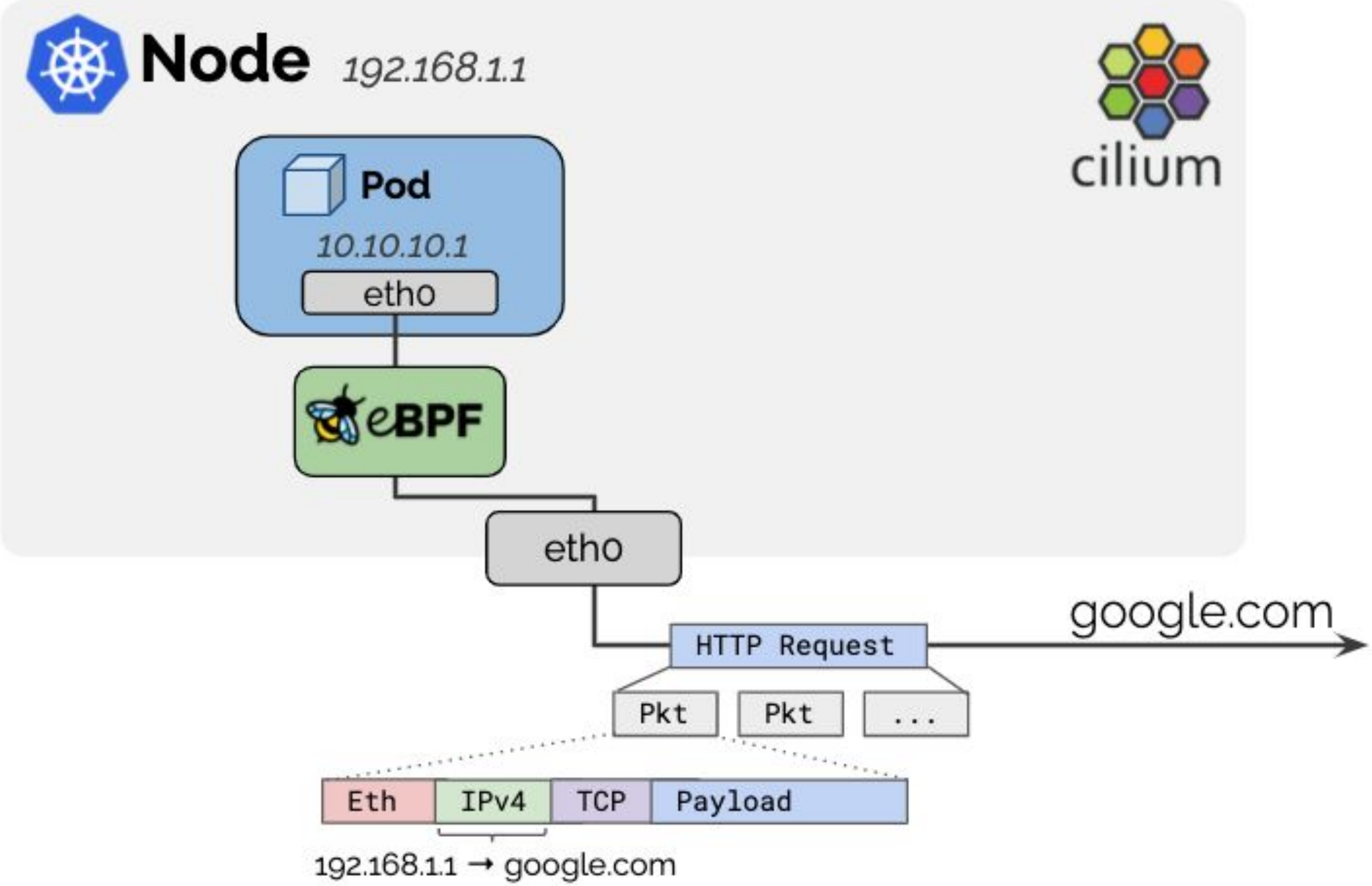
Routing Table
10.10.20.1/32 via lxc1
10.10.20.2/32 via lxc2
10.10.10.0/24 via 192.168.1.1

eth0

AKRAINO

ByteDance 字节跳动

Masquerading



AKRAINO

ByteDance 字节跳动

SNAT mode

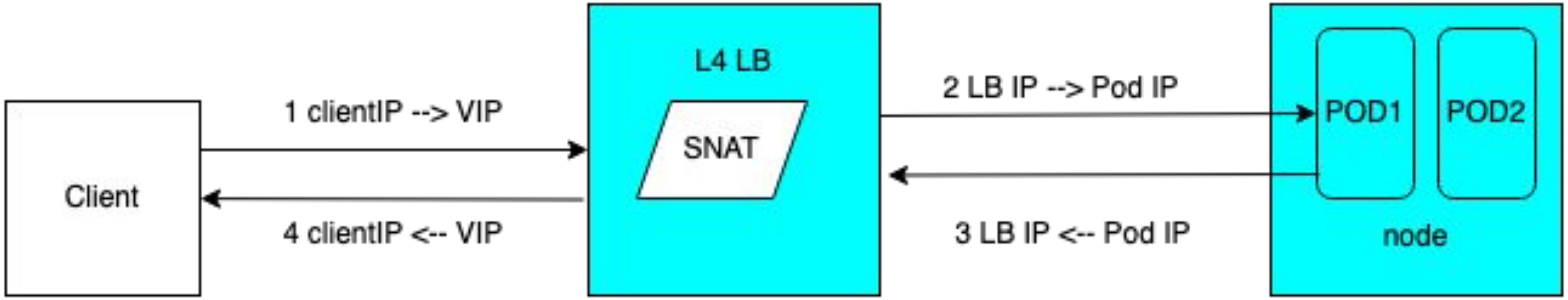


Figure1: SNAT packet flow

Direct Server Return (DSR) mode

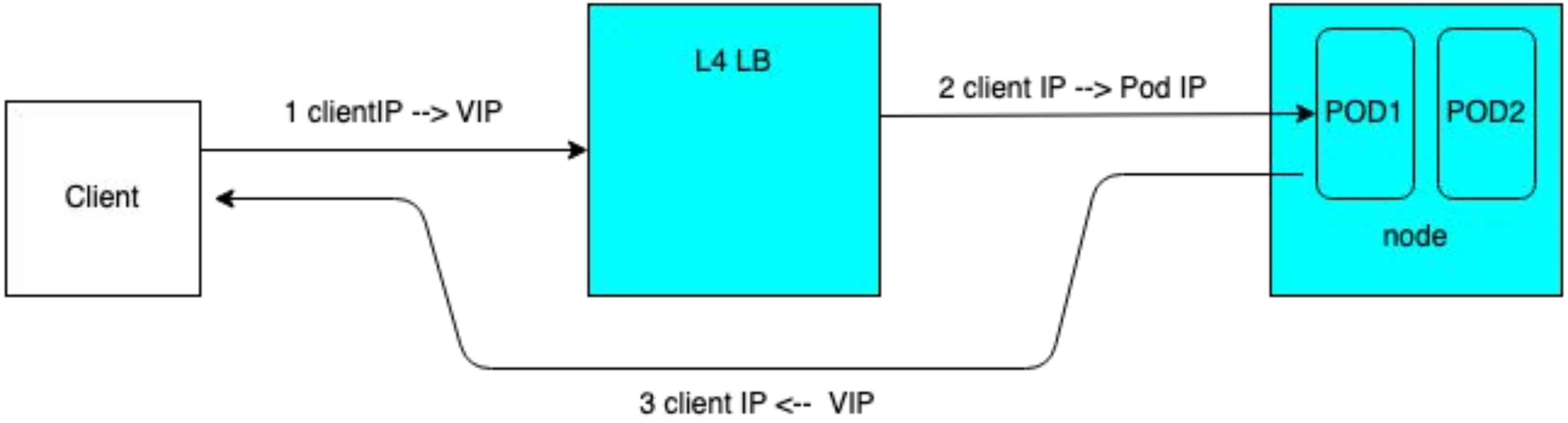
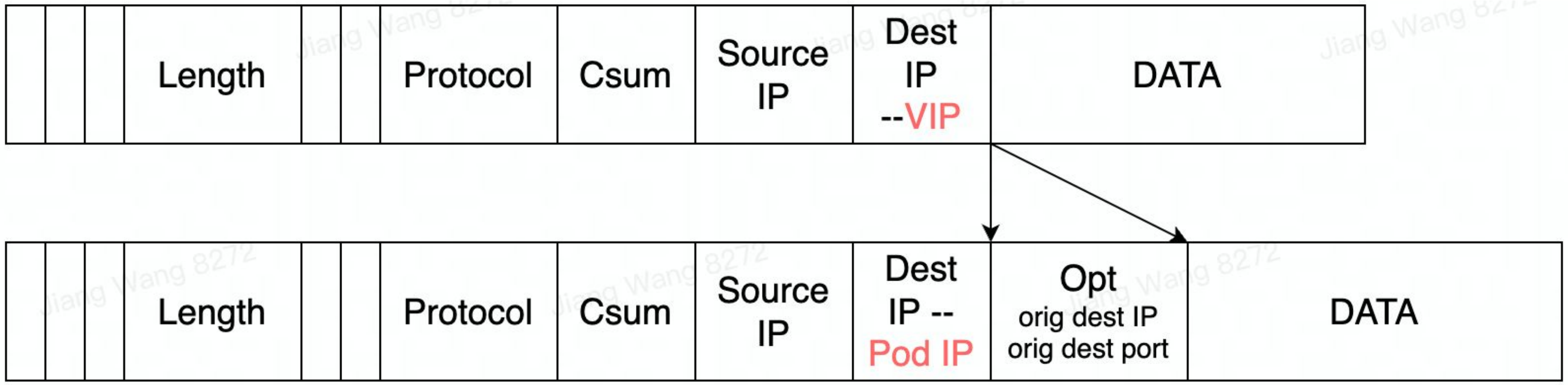


Figure1: DSR packet flow

- Benefit: less processing on LB.
- Where to store VIP?

Current Cilium DSR Support for IPv4

- Use IPv4 options



Length | Protocol | Csum | Source IP | Dest IP --VIP | DATA

Length | Protocol | Csum | Source IP | Dest IP -- Pod IP | Opt orig dest IP orig dest port | DATA

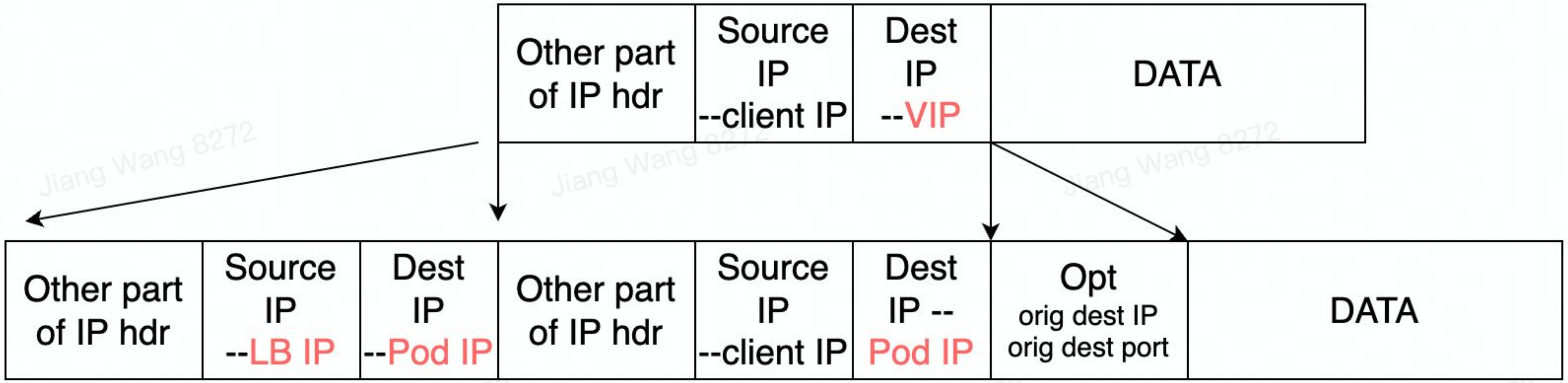
Figure 2: IP packet format in DSR with IP option

Problem with Current DSR Support for IPv4

- Problem: will go to slow path on switches
  - Network switch has a fast path and slow path
  - IP packets with option will go via slow path on many switches
  - Switches CPU usage arrived 100% for some cores. Bottleneck!
- Could we do better?

Proposal: Use IP-in-IP for DSR on IPv4

- Use IP-in-IP



Other part of IP hdr | Source IP --client IP | Dest IP --VIP | DATA

Other part of IP hdr | Source IP --LB IP | Dest IP --Pod IP | Other part of IP hdr | Source IP --client IP | Dest IP -- Pod IP | Opt orig dest IP orig dest port | DATA

Figure 3: IP packet format in DSR with IP-in-IP

- Will go to fast path on switches
- Drawbacks: smaller MTU for UDP and TCP Syn packets

Cilium eBPF DSR Data Path Changes

- On the LB:

```
nodeport_lb4
    |- tail_nodeport_ipv4_dsr
        |- dsr_set_ipip4
```

- On real node:

```
handle_to_container
 |-tail_ipv4_to_endpoint
    |-ipv4_policy
      |- handle_dsr_v4
        |- snat_v4_create_dsr
```

Cilium eBPF DSR Data Path

- When sending a reply to the client
  - bpf_lxc finds out that the "dsr" bit was set
  - does a lookup in the NAT table to find the mapping
  - rewrites the source addr and port to the svc addr and port.
- No changes here

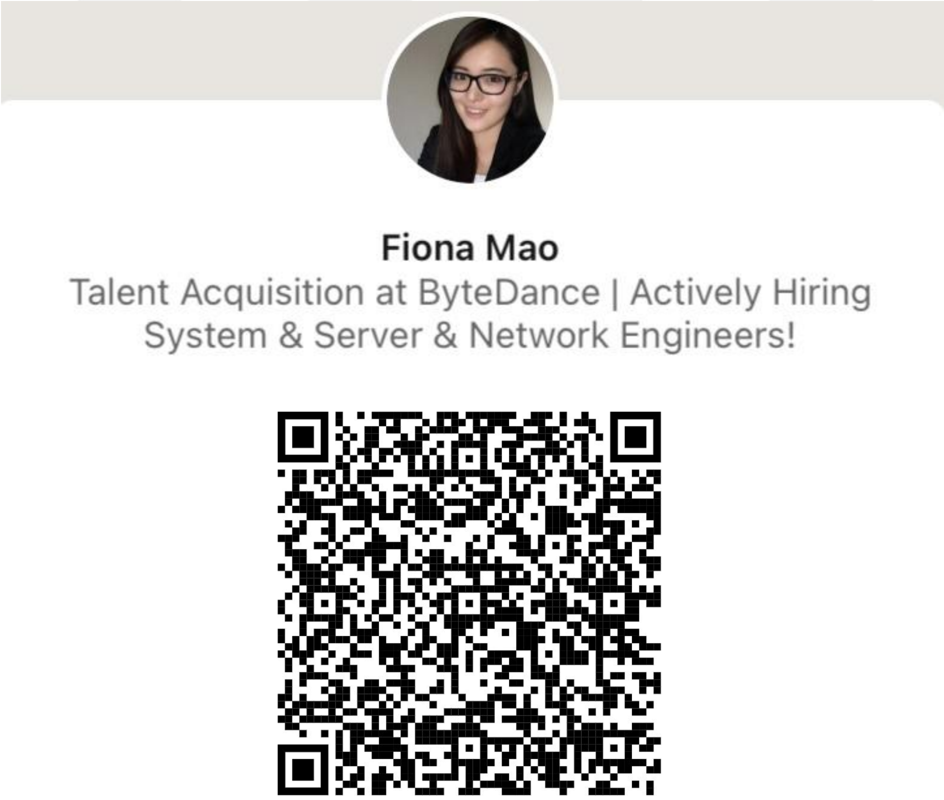
Future Work

- IPv6 support? We may not need that.

PR link:

https://github.com/cilium/cilium/pull/18449

We Are Hiring!



Fiona Mao

Talent Acquisition at ByteDance | Actively Hiring System & Server & Network Engineers!

Interested in our job openings (Linux Kernel, Compiler, Networking, Virtualization)?

Please scan the QR code for JDs or contact our recruiter Fiona Mao via email: fiona.mao@bytedance.com

AKRAINO

ByteDance 字节跳动

ByteDance 字节跳动

Thank you!

AKRAINO

ByteDance 字节跳动