Using VPP as Envoy’s Network Stack

Florin Coras, Cisco
Vector Packet Processing (VPP)

forwarding model as a graph

batch processing
Vector Packet Processing (VPP)

Input | L2 features | L3 features | L4 termination

Diagram showing connections between input and different layers.
Network Stack

- DPDK and native device drivers
- Switching, Routing
- IPsec, ACL, NAT, MPLS, SR, tunneling ...
Host Stack

-Transports: TCP, UDP, QUIC, TLS
-Session scheduling, on demand tx pacing, builtin app support
-Shared memory infra for io/ctrl events and data exchanges with external apps
-POSIX-like apis through VCL
Why?

- User space performance
- Tighter Envoy integration with network stack
- Extensibility
Why?

- User space performance
- Tighter Envoy integration with network stack
- Extensibility
- Kubernetes integration via Calico-VPP (see talk)
Envoy VCL Interface

- Prerequisite: Socket and IoHandle refactor
  - Avoid direct use of socket fds
  - Pluggable IoHandle factories (SocketInterfaces)
  - Delegate FileEvent creation to IoHandle
  - Custom BIOs for TLS
Envoy VCL Interface

- Prerequisite: Socket and IoHandle refactor
  - Avoid direct use of socket fds
  - Pluggable IoHandle factories (SocketInterfaces)
  - Delegate FileEvent creation to IoHandle
  - Custom BIOs for TLS
- VCL interface
  - Retrieves async VPP session events via VCL worker eventfd FileEvent
  - Maintains per worker epoll fd that tracks all IoHandles
  - Custom VCL IoHandle FileEvents are created and dispatched independent of libevent
Prerequisite: Socket and IoHandle refactor
- Avoid direct use of socket fds
- Pluggable IoHandle factories (SocketInterfaces)
- Delegate FileEvent creation to IoHandle
- Custom BIOs for TLS

VCL interface
- Retrieves async VPP session events via VCL worker eventfd FileEvent
- Maintains per worker epoll fd that tracks all IoHandles
- Custom VCL IoHandle FileEvents are created and dispatched independent of libevent

Next steps VCL IoHandle Optimizations
- Optimize socket read operations
- Investigate minimizing data copying
Performance

Server 1 (Intel Xeon Gold 6146)
- 8 workers
- 64B or 2kB payload

Server 2 (Intel Xeon Gold 6146)
- 40 threads
- 120 connections

- nginx - tap - vpp - 40GE - wrk
Thank You

- Early Envoy-VPP integration code
  - Github: https://github.com/florincoras/envoy-vpp
  - Slack: https://envoyproxy.slack.com/

- VPP Project:
  - Parent LFN project: https://fd.io/
  - Mailing lists: https://lists.fd.io/g/main
  - Host Stack documentation: https://wiki.fd.io/view/VPP/HostStack