

eXpress Data Path and eBPF

ECE/CS598HPN

Radhika Mittal

Performance overhead in kernel stack

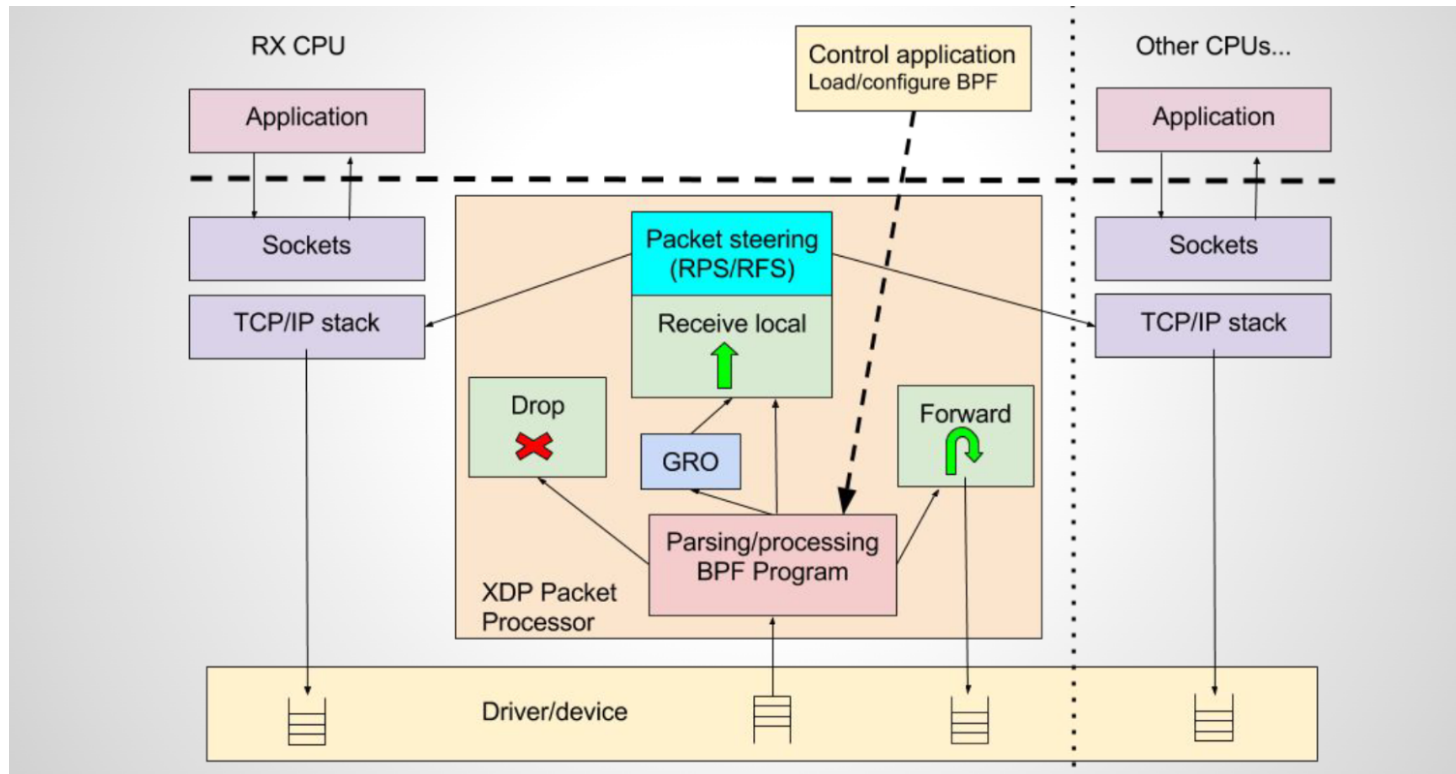
- Protocol processing
- Data copy
- Cache contention (between flows sharing same NUMA node)
- CPU scheduling overheads (locking, context switching)
- Interrupts
- Managing heavy datastructures (skbs)

Alternatives

- Kernel bypass in software (user-space):
 - Enabled by high-speed packet I/O engine (DPDK)
 - Network stacks over DPDK: mTCP, IX, TAS....
 - Software NIC offload over DPDK (later this month)
- Kernel/CPU bypass using specialized NIC
 - e.g. RDMA (next class)
 - other smartNIC based offloads (later in the course)
- Augmenting kernel datapath with programmable, high performance, packet processor: **XDP**

XDP (Express Datapath)

- Baremetal packet processing at the lowest point in the software stack.



Contents from Herbert and Starovoitov XDP presentation.

Key components of XDP

- XDP driver hook
- eBPF virtual machine
- BPF maps
- eBPF verifier

Detour: eBPF

- Allows inserting (safe) user-defined programs into the running kernel.
 - Loaded dynamically.
 - No kernel recompilation or restart required.
 - Root access still needed.
- First introduced in 2014.
- Picked up in 2016 with XDP and Cilium.

Detour: eBPF

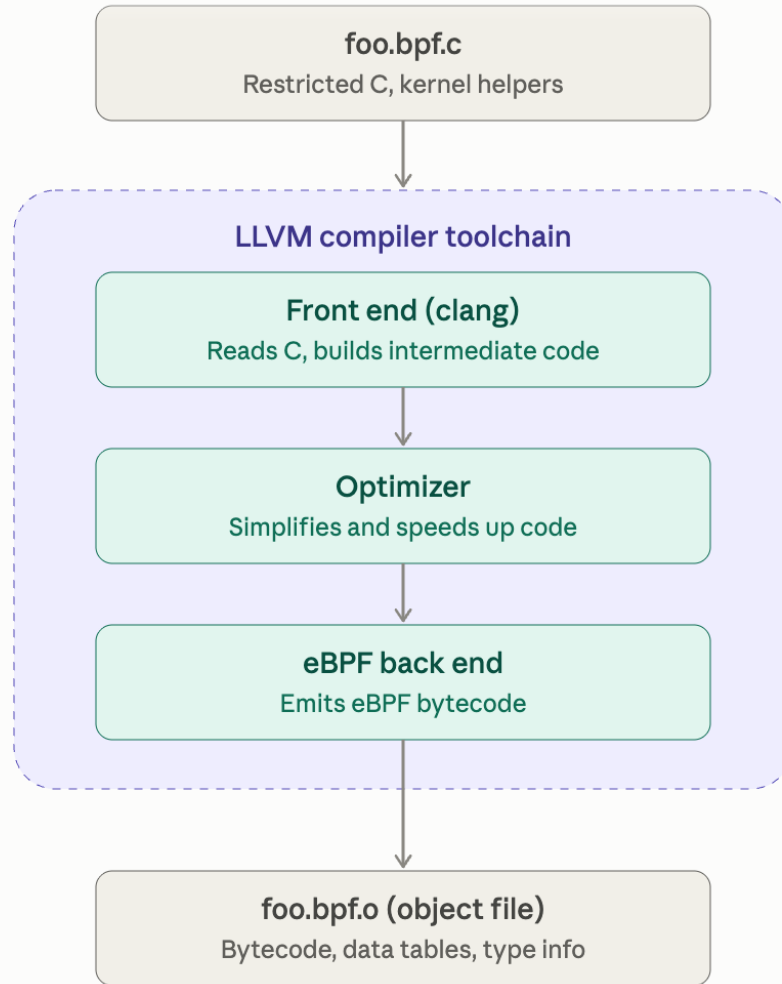


Figure drawn with help from LLM (Claude).

Detour: eBPF

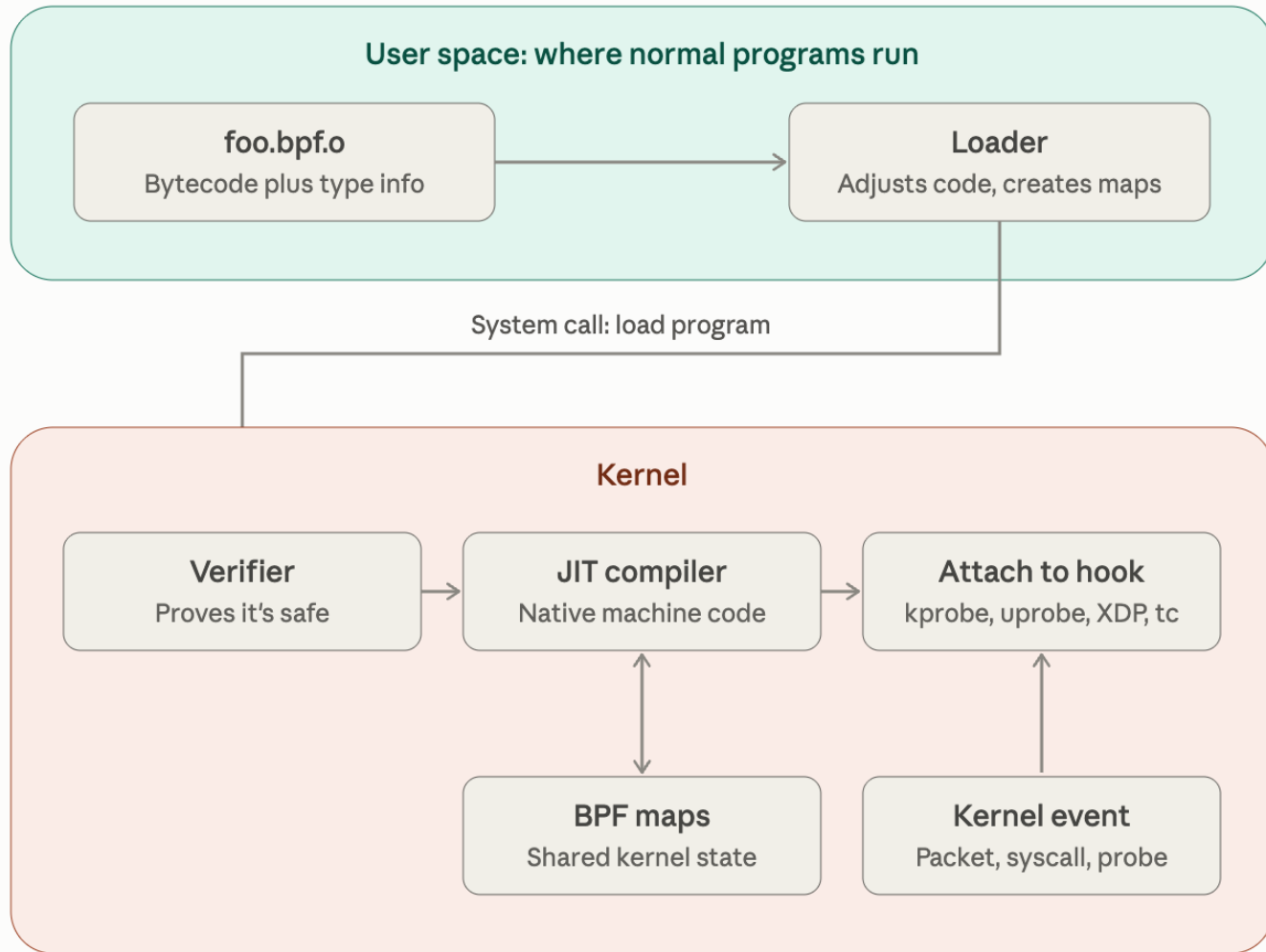
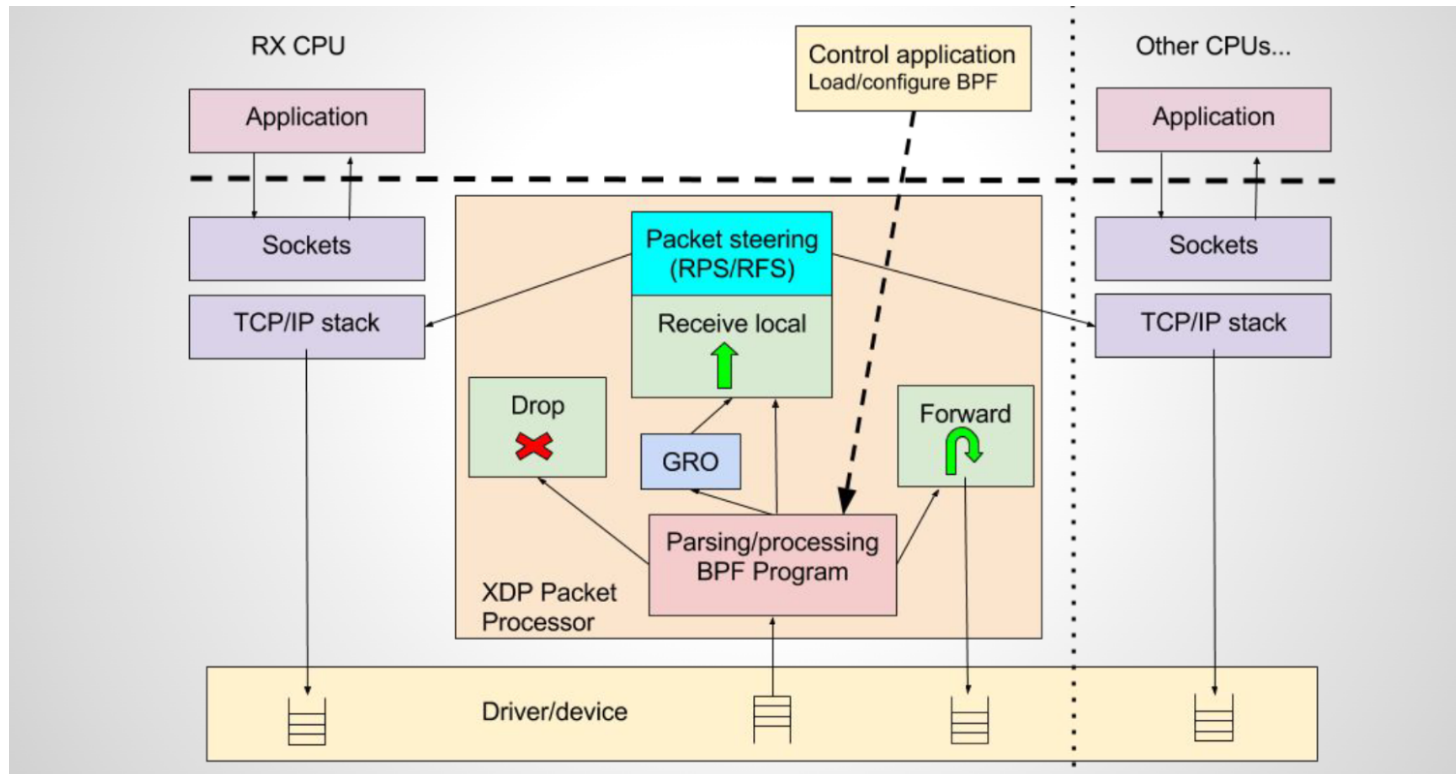


Figure drawn with help from LLM (Claude).

XDP (Express Datapath)

- Baremetal packet processing at the lowest point in the software stack.



Contents from Herbert and Starovoitov XDP presentation.

Benefits of XDP over DPDK?

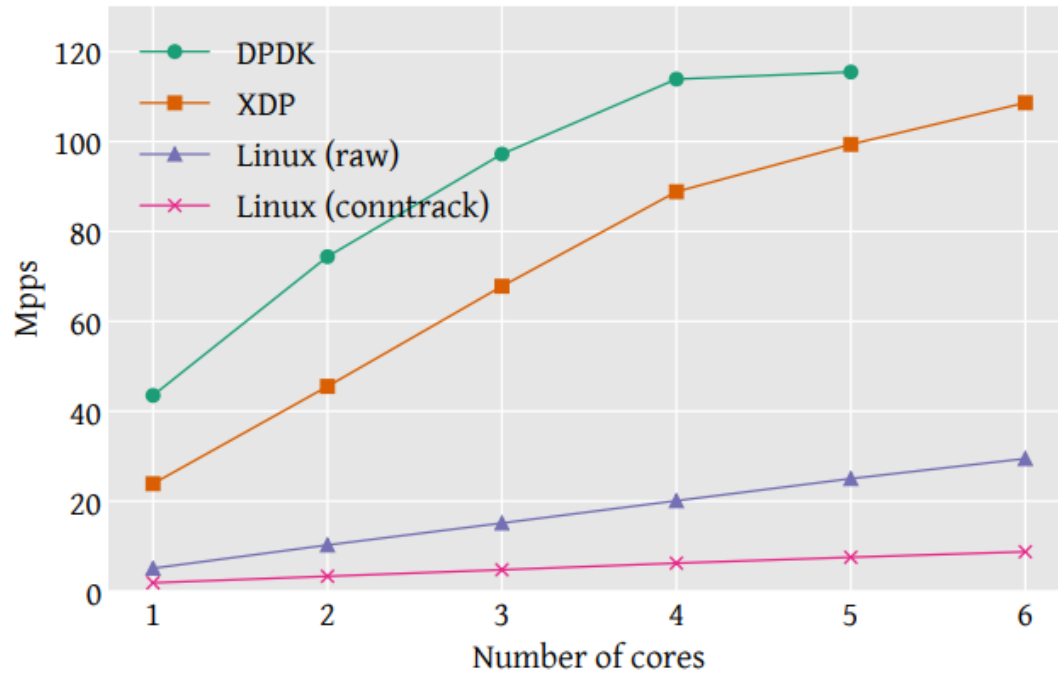
- Retains kernel security boundary.
- No special hardware requirements.
 - Only needs basic support like multiqueue NIC, TSO, etc.
 - Easier to integrate with existing NICs and drivers.
- Allows selective utilization of kernel stack (routing, TCP, etc).
- No expensive packet re-injection from user-space to kernel.
- Transparent to applications running on host.
- Dynamic runtime re-programming.
- No need to dedicate CPU cores to packet processing.

Limitations of XDP over DPDK?

- Performance!

Limitations of XDP over DPDK?

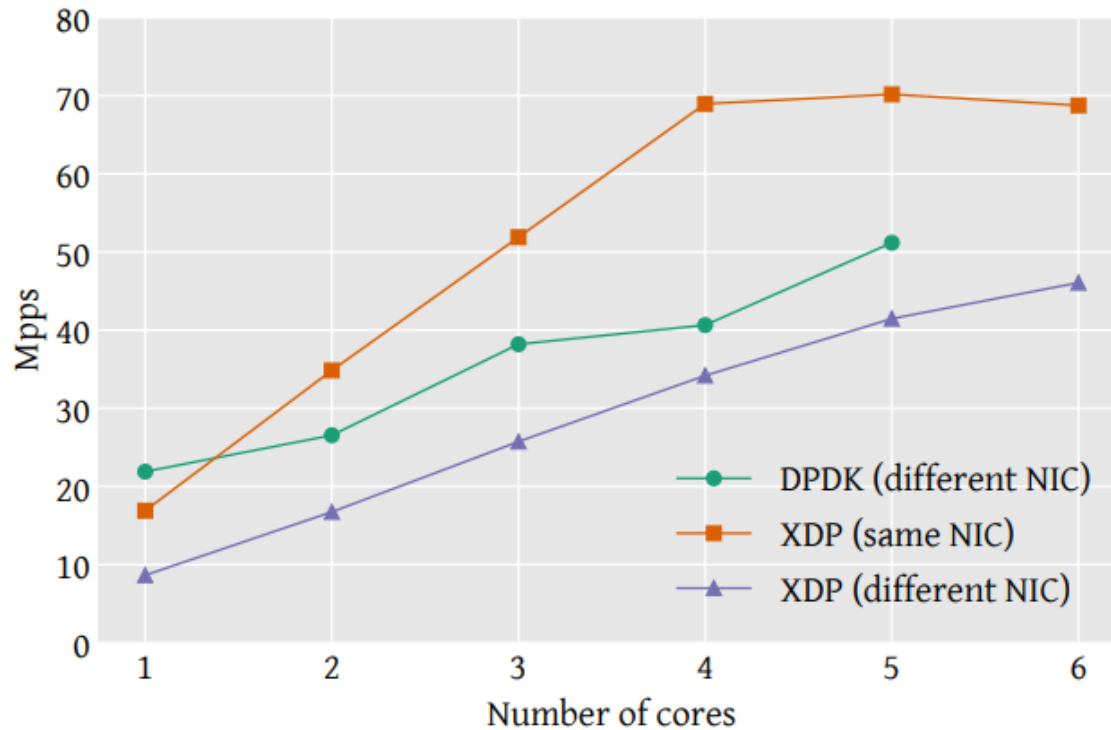
- Performance!



Packet drop throughput

Limitations of XDP over DPDK?

- Performance!



Packet forwarding throughput

Limitations of XDP over DPDK?

- Performance!

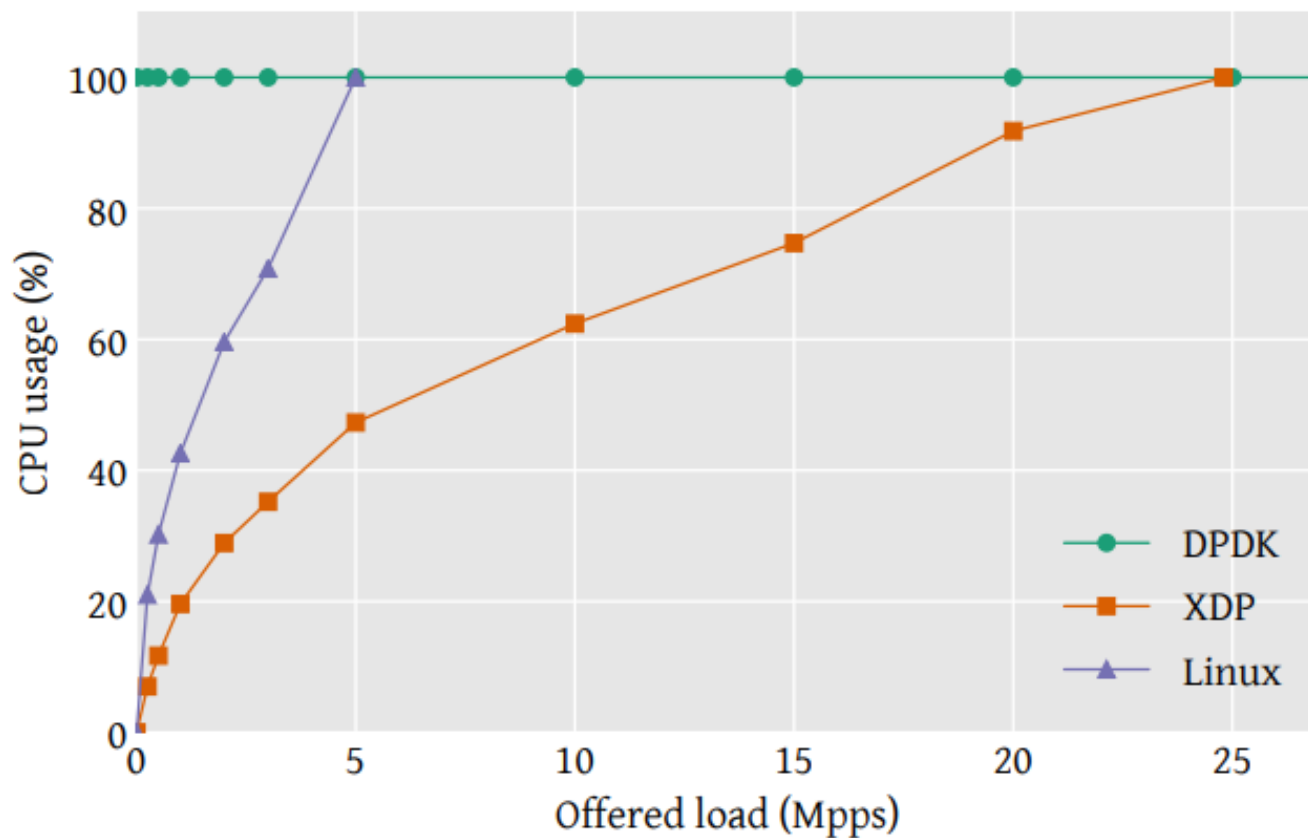
	Average		Maximum		< 10 μ s	
	100 pps	1 Mpps	100 pps	1 Mpps	100 pps	1 Mpps
XDP	82 μ s	7 μ s	272 μ s	202 μ s	0%	98.1%
DPDK	2 μ s	3 μ s	161 μ s	189 μ s	99.5%	99.0%

Latency

Limitations of XDP over DPDK?

- Performance!
- Reasons?
 - Interrupt vs polling
 - Overhead of generic device driver

CPU utilization



Relationship to Kernel-bypass (DPDK)

- If kernel network stack is a freeway,
 - kernel-bypass is a proposal for high-speed train infrastructure.
 - XDP is a proposal for adding carpool lanes on the freeway.

Your thoughts?

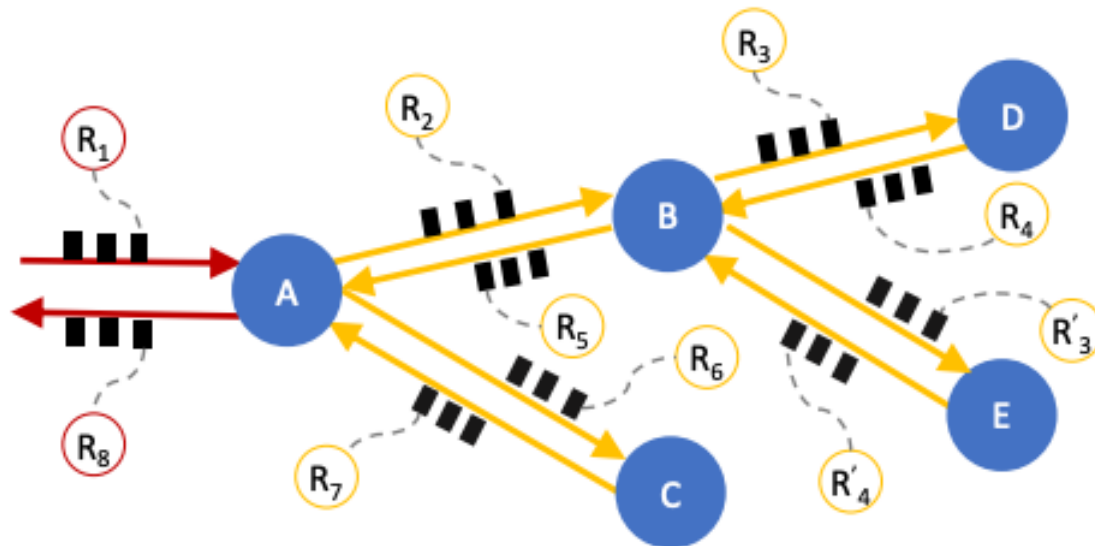
Useases: service meshes

Many modern applications deployed as modular *microservices* that communicate through RPCs.

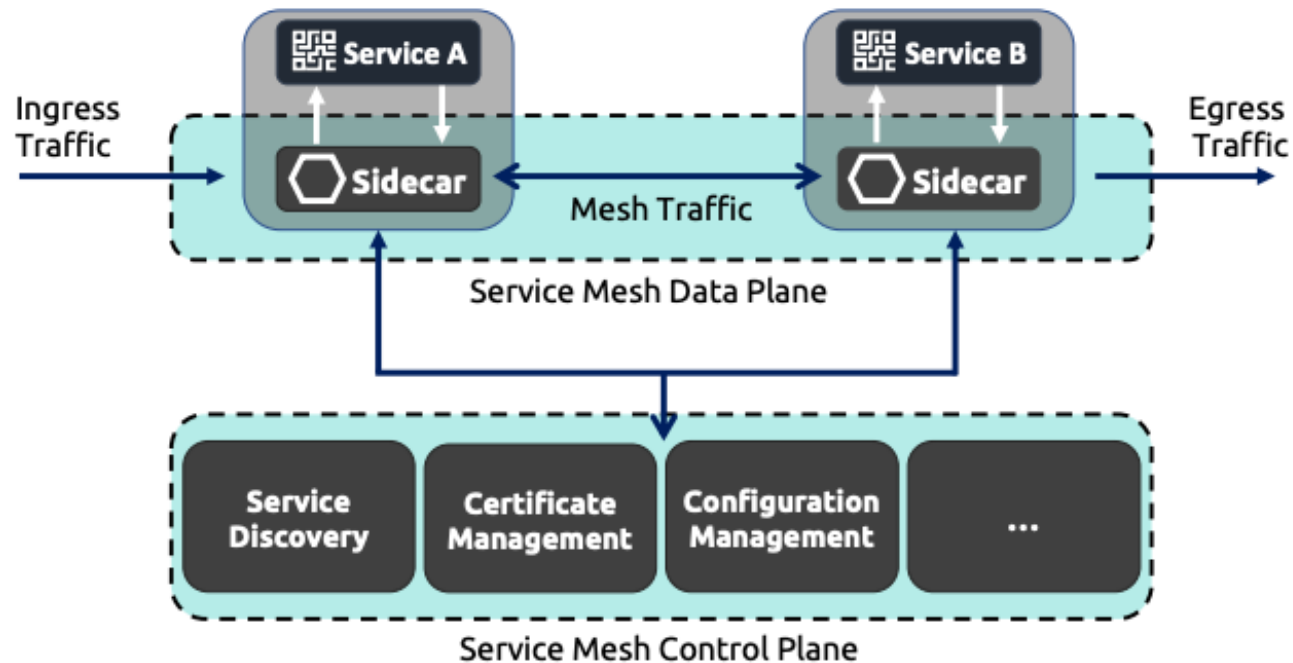
Each microservice can have multiple replicas (auto-scaled).

Each replica deployed as a pod in a container.

Managed using systems like Kubernetes and **service meshes** (Istio/Envoy).



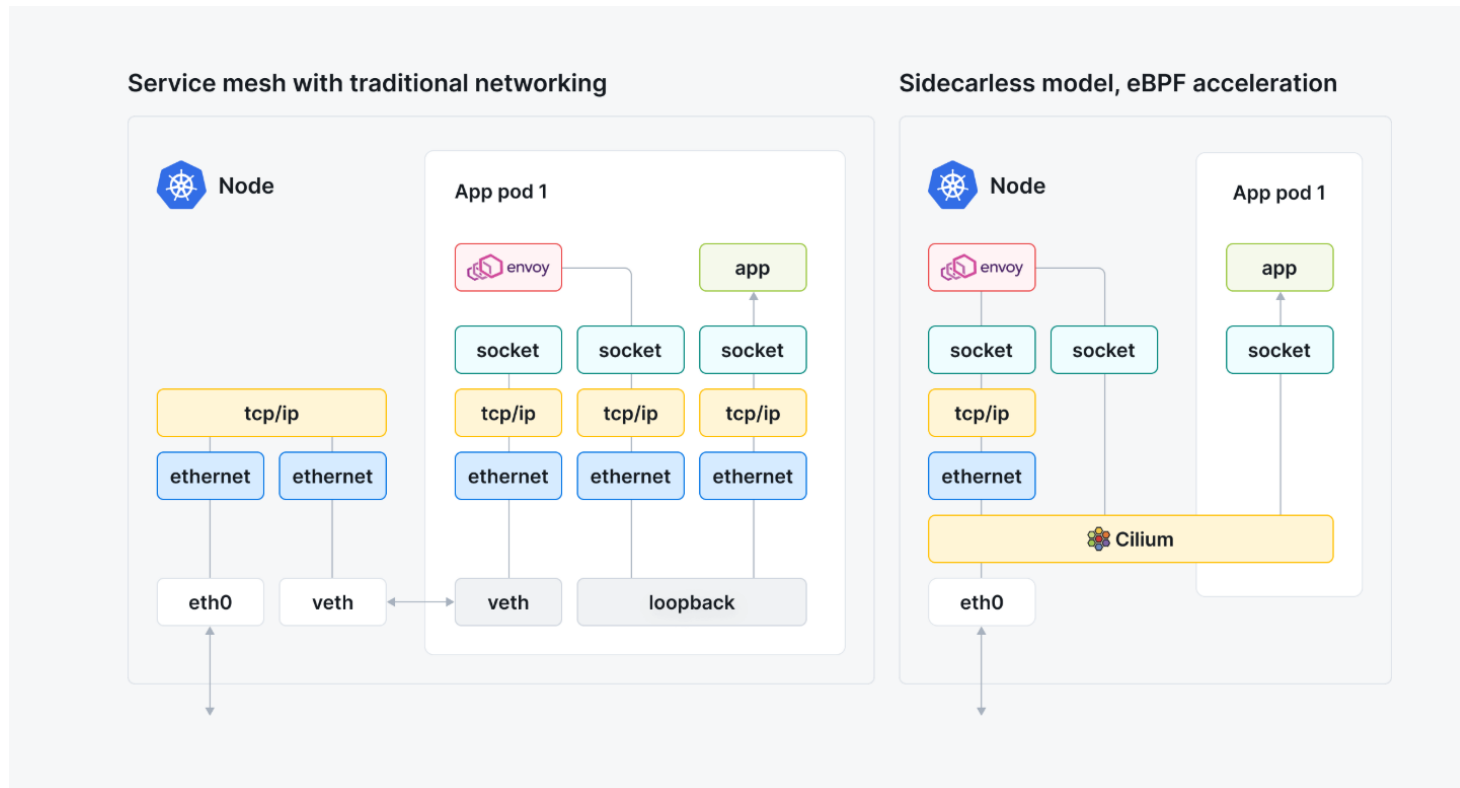
Useases: service meshes



Useases: service meshes

Cilium uses XDP/eBPF to accelerate sidecar proxies.

Some of the sidecar features (e.g. network policies and load balancing) also offloaded to XDP.



Usecases of eBPF: observability

- Observability and monitoring
 - Multiple platforms for cloud-native microservices.
 - e.g. OpenTelemetry, New Relic, etc.
 - Advantage: infrastructure for app-layer monitoring without requiring application modification.

Usecases of eBPF: Meta

- NetEdit platform incorporates several eBPF usecases to tune various aspects of the kernel stack (SIGCOMM'24)

	Tuning Feature	Description	Hookpoint	Granularity
1	Init CWND	tune initial window	Sockops	Transport
2	RWND	tune receiver window (flow control)	TC	
3	Connection timeout	Tune initial connection time-out	Sockops	
4	Pacing Tuner	Tune flow pacing rate	Sockops	
5	MSS Tuner	Toggle Jumbo MTU setting	Sockops	
6	BPF DCTCP	New Congestion Control	Struct_ops Sockops	
7	D2TCP	New Congestion Control	Struct_ops Sockops	
8	BET	New Congestion Control	Struct_ops Sockops Sockopt	
9	ILA	V6 virtual networking	TC	Connection
10	Nat64	Nat64 gateway	TC	

Table 3: TuningFeatures developed on NetEdit.

- Large scale deployment of DCTP congestion controller (NSDI'24)
 - selectively enable use of DCTCP for some flows.

Other usecases

- Rate limiting (Cloudflare)
- DDOS mitigation (Cloudflare)
- L4 Load balancing (Cloudflare, Meta, ...)
- Research papers:
 - Accelerating in-memory key-value stores (memcached): Ghigoff et. al., NSDI'21.
 - Accelerating storage functions: Zhong et. al., NSDI'21
 - Accelerating distributed protocols such as Paxos: Zhou et. al., NSDI'23
 - Accelerating distributed transaction processing: Zhou et. al., NSDI'24
 - Request tracing: Ashok et. al, SIGCOMM'24, Shen et. al, SIGCOMM'23
- Active area of research in recent years.