
KubeCon
CloudNativeCon
North America 2019

Mizar

Futurewei Technologies

Current state and work in progress

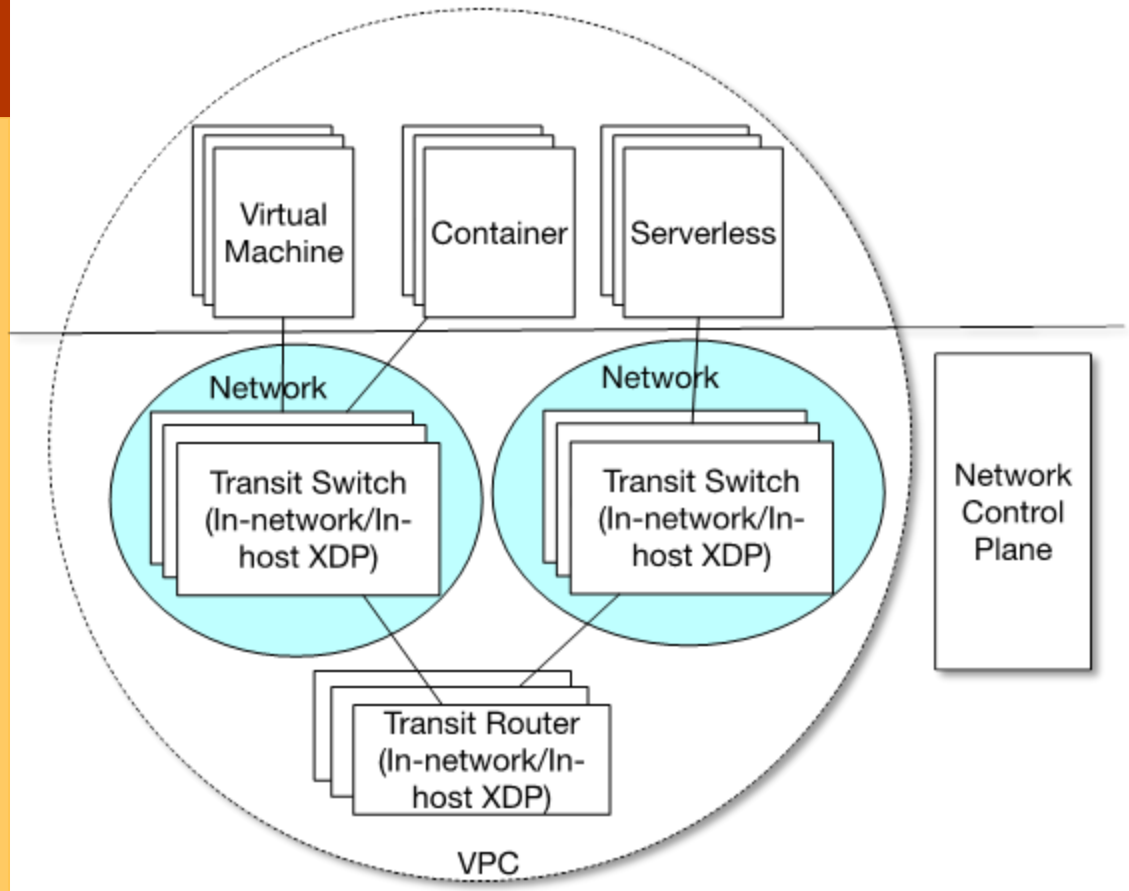
https://github.com/futurewei-cloud/mizar

The Problem We are Trying to Solve

1. Support provisioning and management of large number endpoints (10M endpoints)
2. Accelerate network resource provisioning for dynamic cloud environments
3. Achieve high network throughput and low latency
4. Create an extensible cloud-network of pluggable network functions
5. Unify the network data-plane for containers, serverless functions, virtual machines, etc

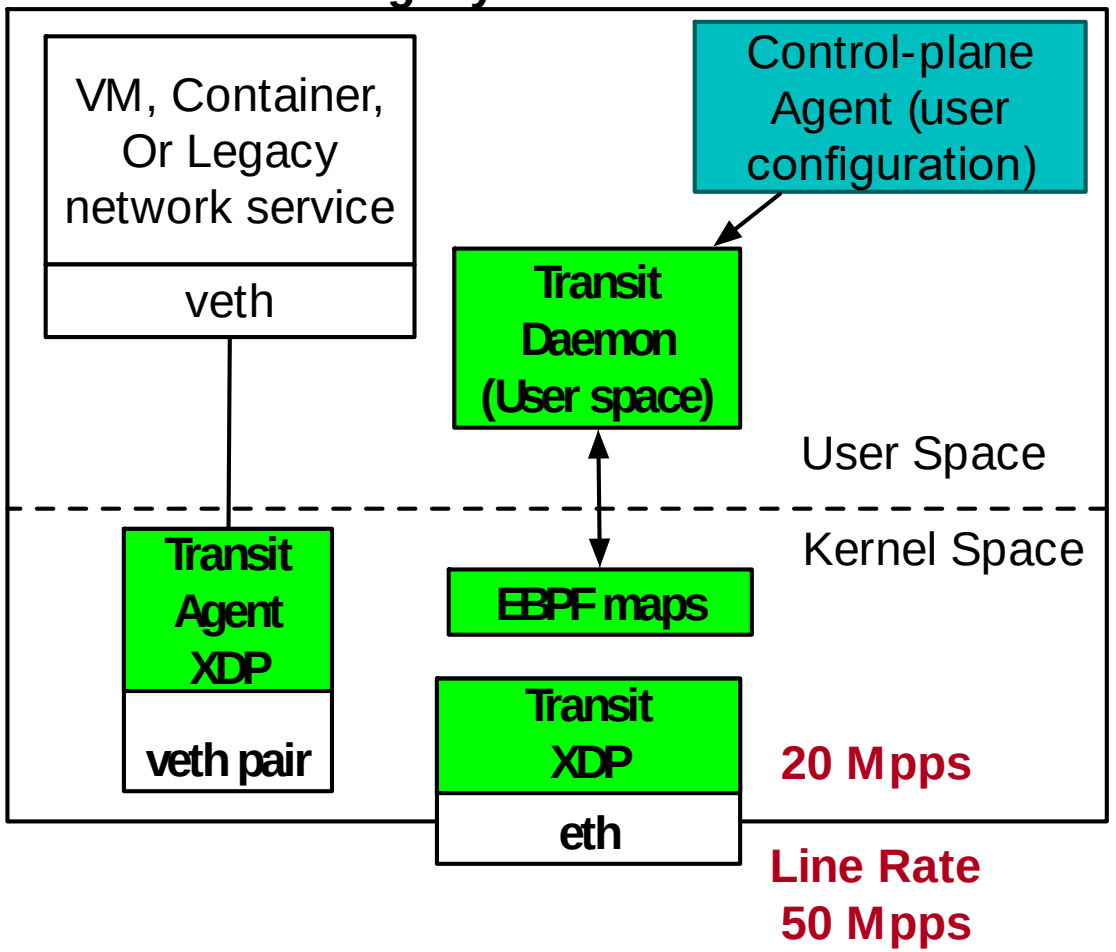
Mizar Overall Architecture!

Natural Partitioning domains of Cloud Network	Transit Switches	Transit Router
• Virtual Private Cloud VPC • Networks within a VPC • Endpoints within a network	• In-network hash tables • Holds the configuration of endpoints within a network • Determines an endpoint host • Implements all Network functions within a network	• In-network hash tables • Holds the configuration of networks within a VPC • Determines a transit switch of an endpoint • Implements all functions within a VPC

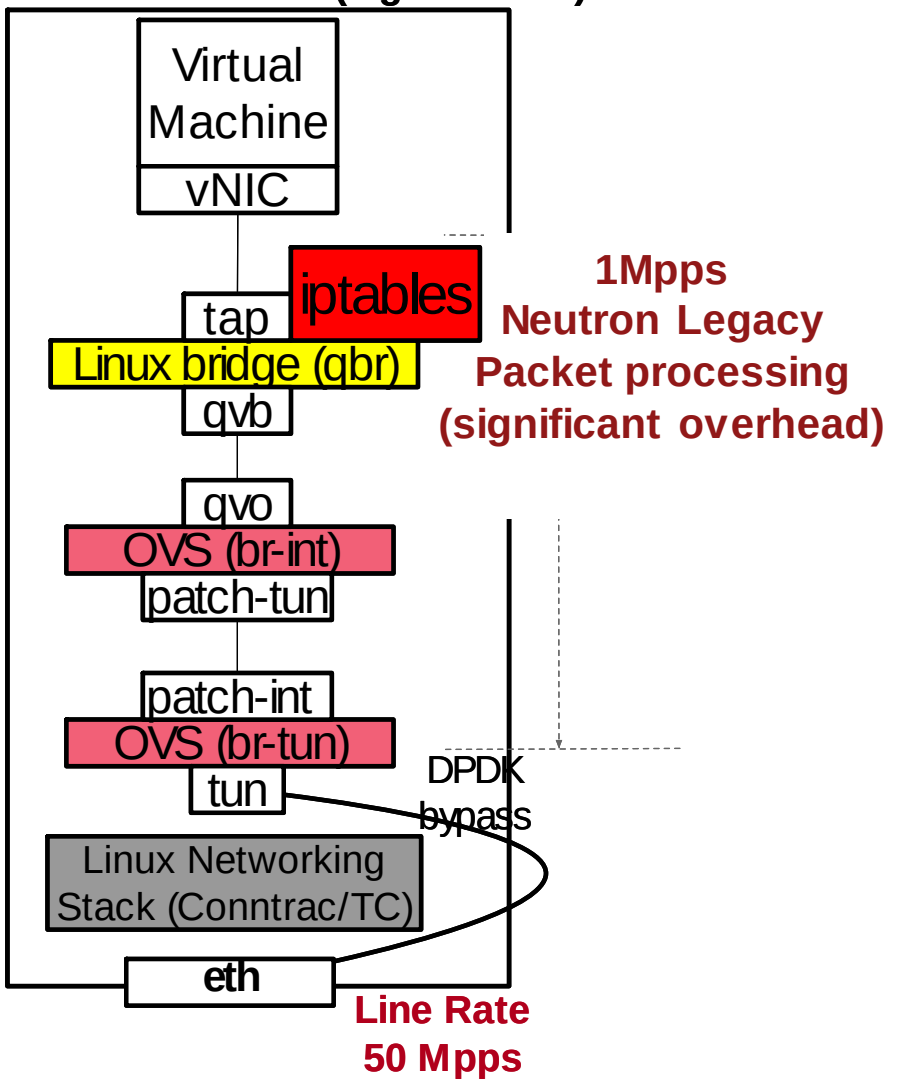


Inside a Mizar host

Mizar Simplified Node design for VMs, Containers, and legacy network services



OVS Based Solutions (e.g. Neutron)



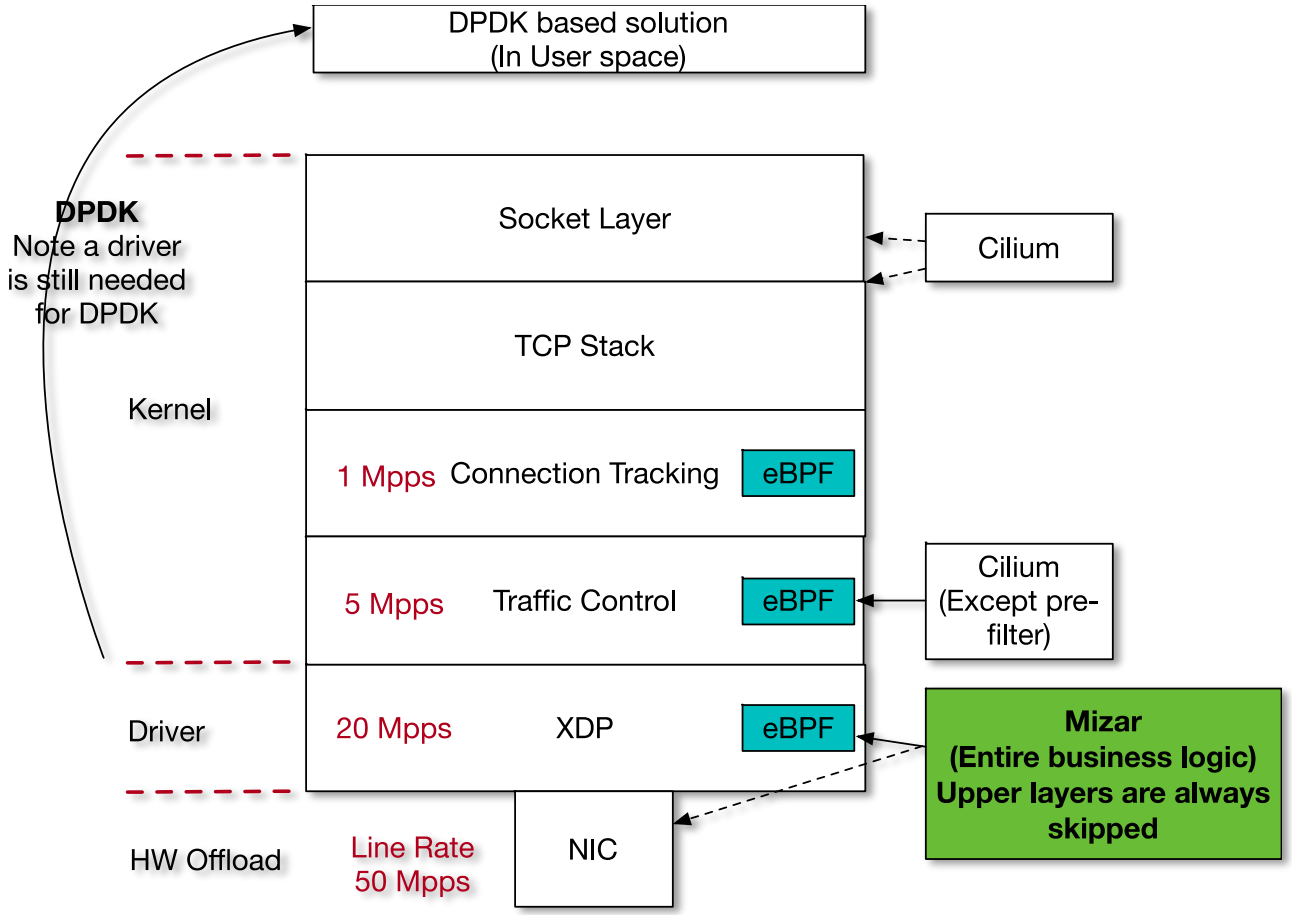
Background XDP: Simplified and Extensible Packet Processing Near Line Rate

Packet processing is entirely in-kernel.

Makes the best use of kernel packet processing constructs without being locked-in to a specific processor architecture.

Skip unnecessary stages of network stack whenever possible and transit packet processing it to smart NICs.

Very small programs < 4KB

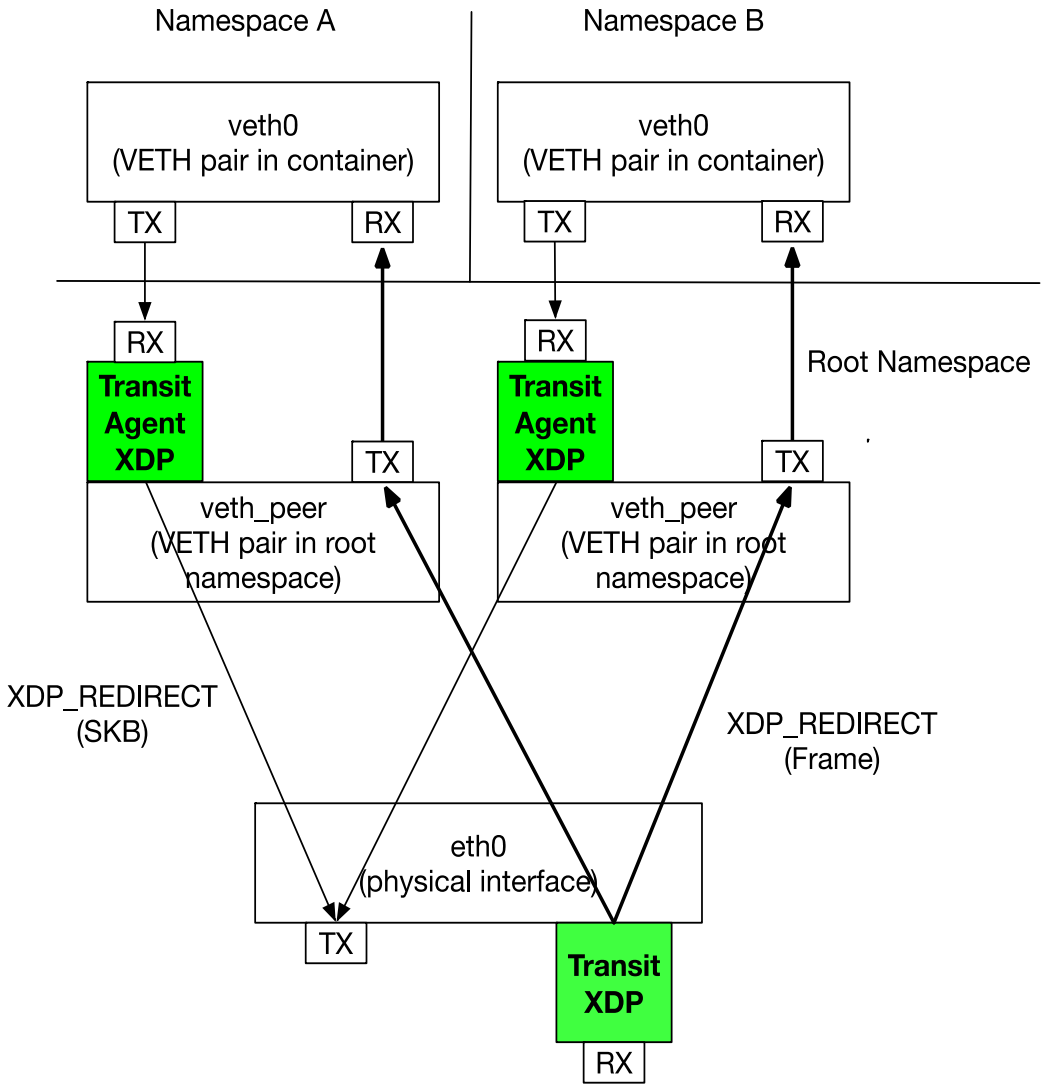


In-host packet flow: Bypass network stack

Packets traverses only the container stack

On egress packets are redirected (SKB) to the main interface after tunneling.

On ingress packets are redirected directly to the container veth peer in the root namespace.

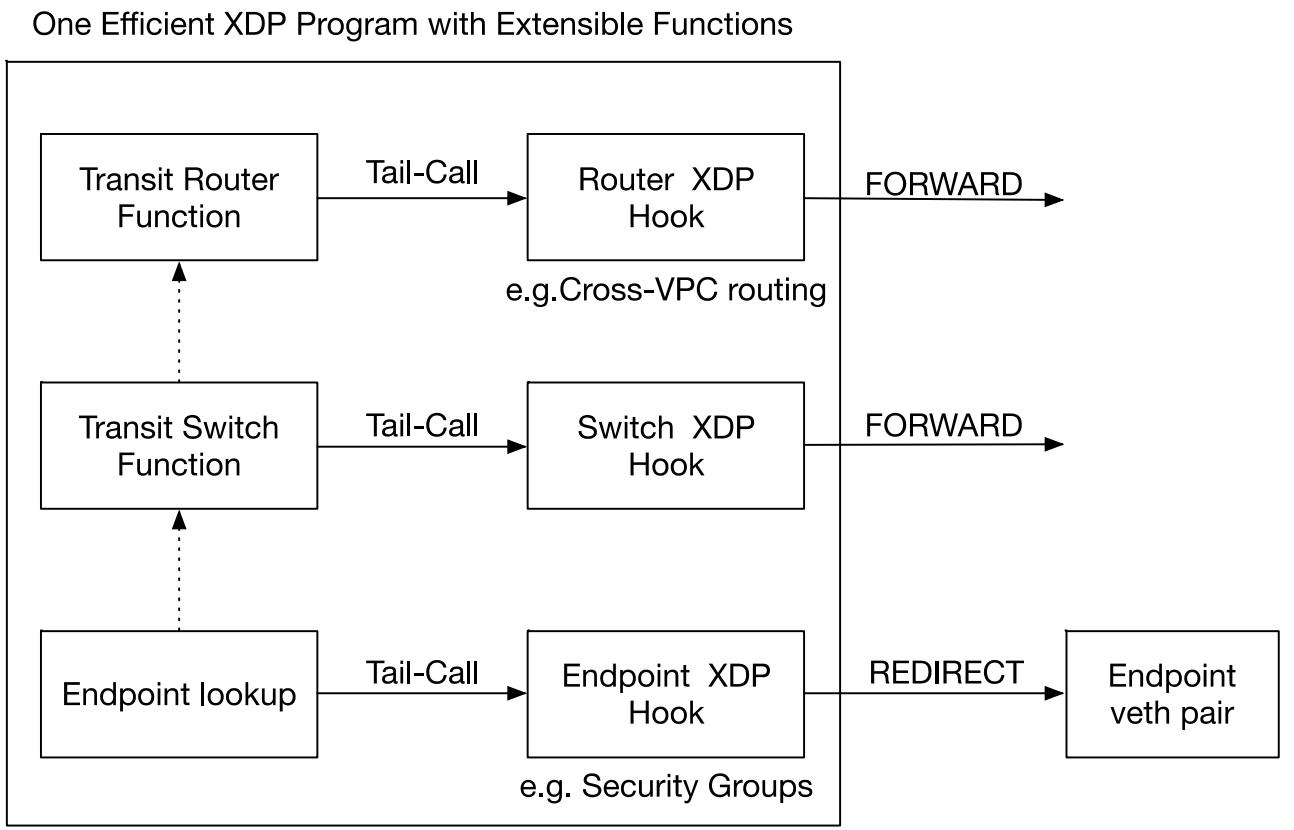


Extensible Packet Processing inside the main XDP program!

Implements essential logical networking function within the same XDP program that provides multi-tenant cloud networking solutions through **new** transit switch and transit routers concepts

Smart Control Plane **will** allow Mizar to Autonomously adapt to various traffic demands in immense scale cloud environments. Thus, allowing Mizar to serve various cloud workloads in a multi-tenant environment optimally.

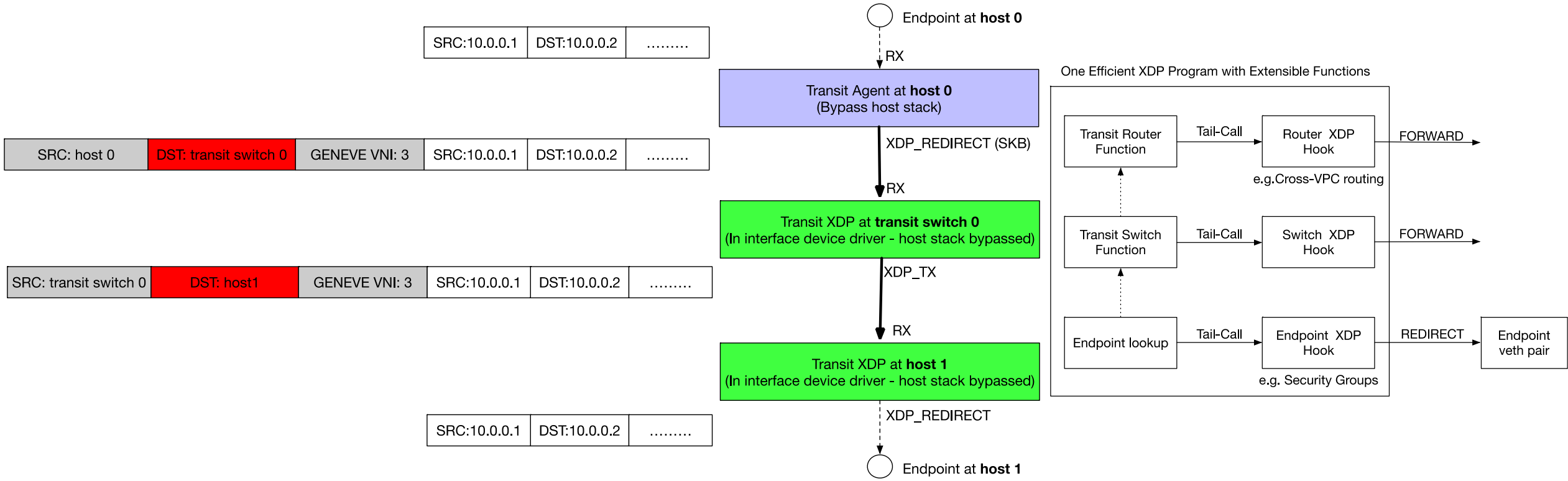
Extensible support of native networking features through custom chains of optimized XDP programs hooks and Geneve protocol options. **Future** possible Features including, Security, Load-balancing, Connectivity, Traffic Shaping Control.



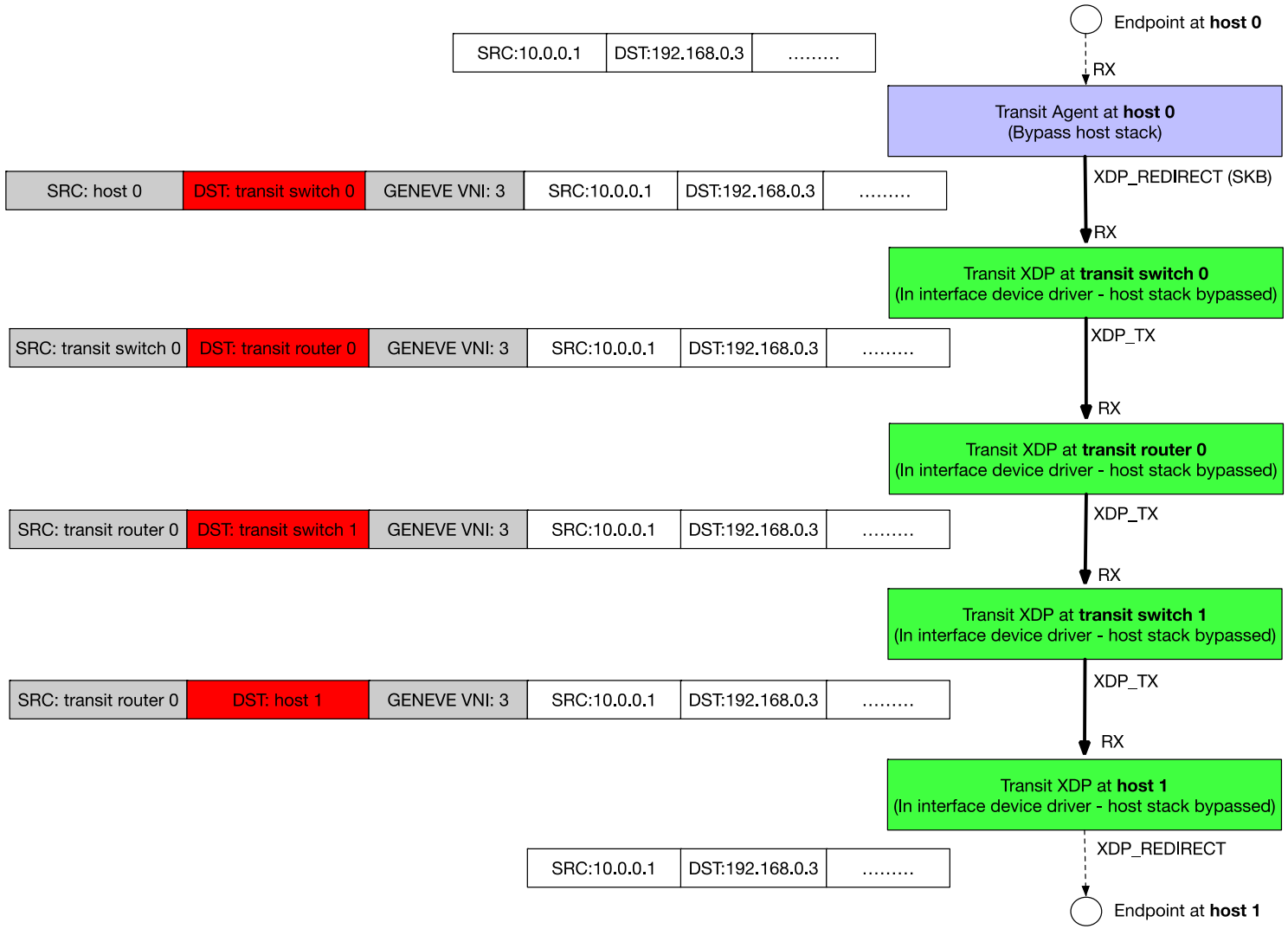
Example packet within a network

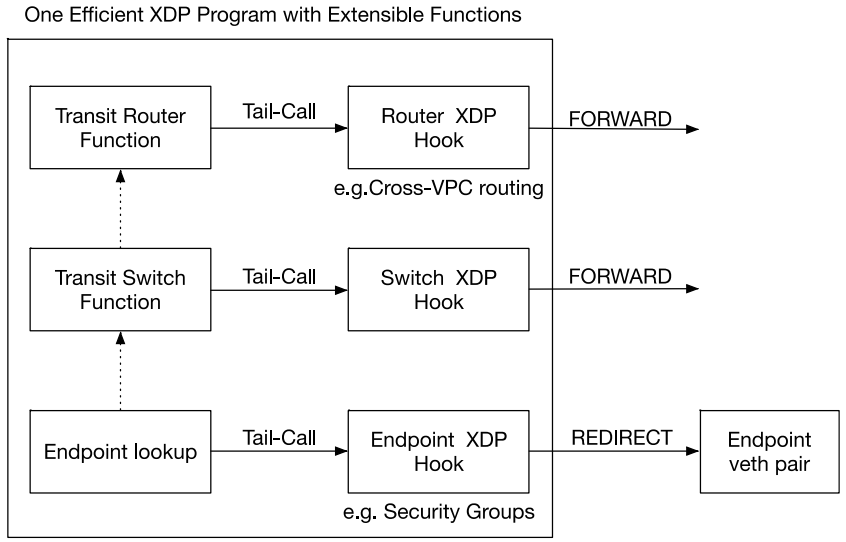
Three steps to provision an endpoint

- Add the endpoint to N transit switch table
- Provision the endpoint on the host
- Configure the host transit agent to tunnel the endpoint traffic to the transit switch

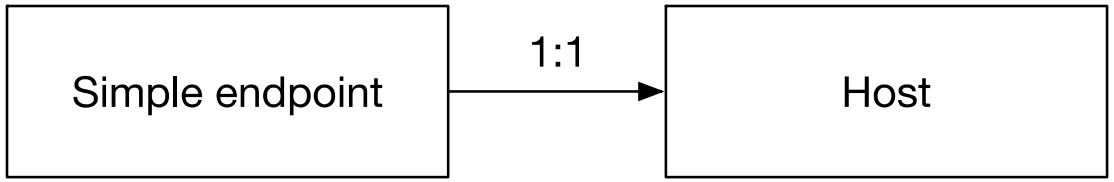


Example packet cross networks

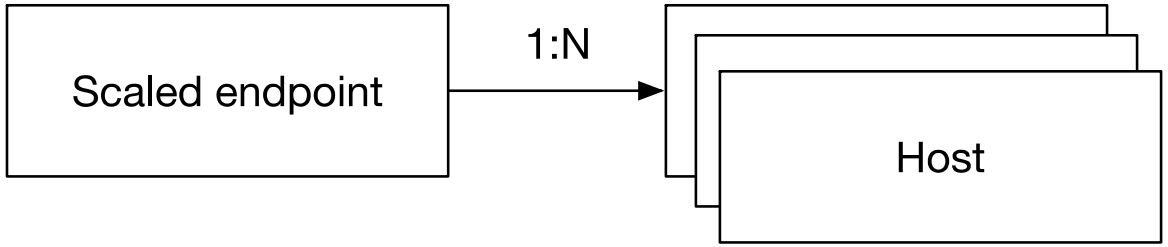




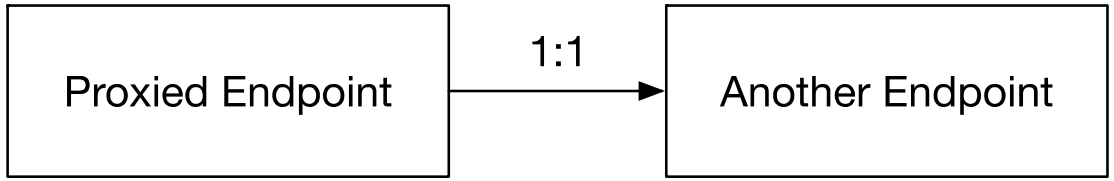
New endpoint types



e.g. container, VM

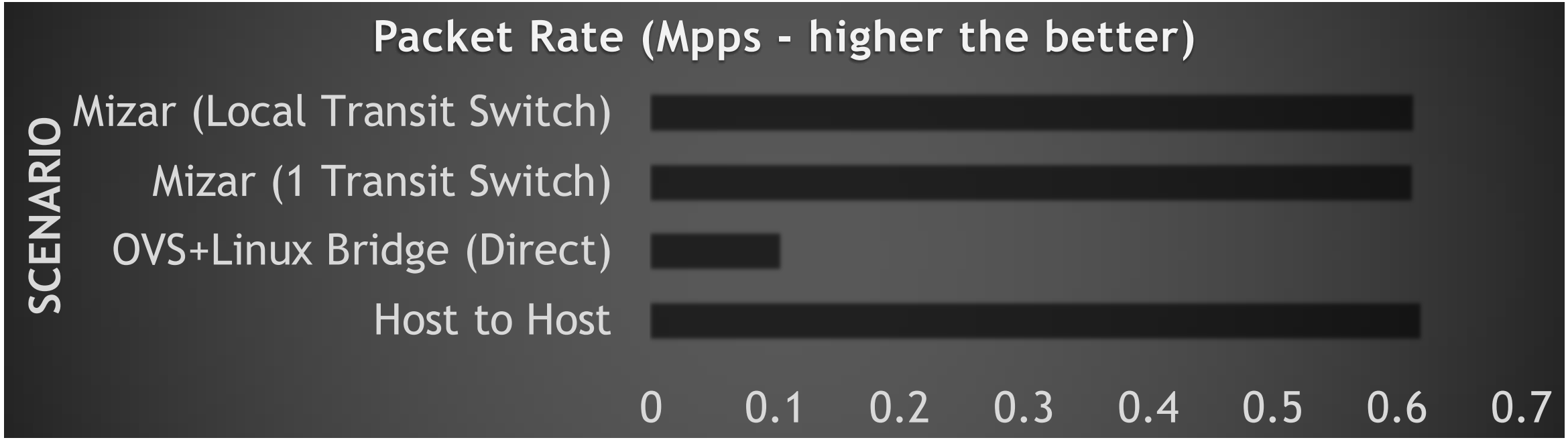


e.g. autoscaling network function: load-balancer, NAT



e.g. autoscaling network function: load-balancer, NAT

Packet Rate (non TCP) – Scaling Network Services



HIT

Near line rate packet per second

Endpoint Update Time with multiple Transit Switches

HIT

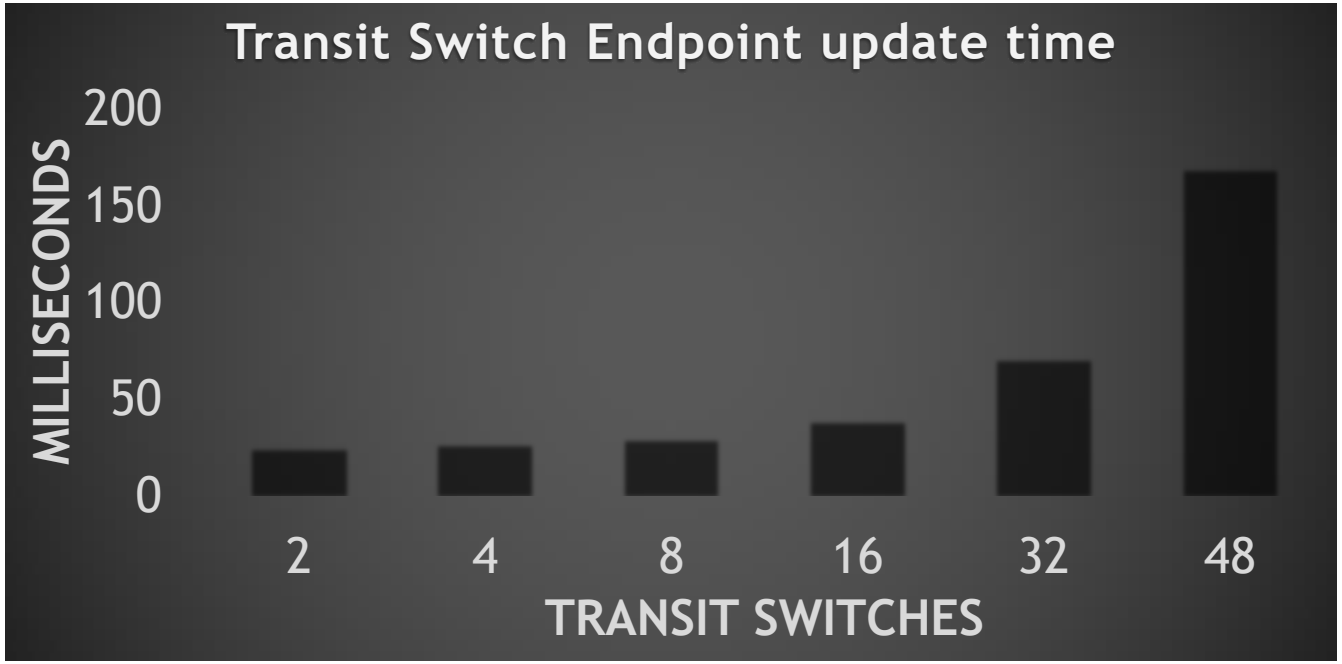
- Constant time with parallel updates (20ms) until the Test Controller starts to Hit its re

Scale

- With a scalable Control-plane (on multiple machines), we foresee maintenance of constant time scaling.

IMPROVEMENT

- Simplifications in data-plane as we introduce the scaled endpoint. One core required.



Endpoint E2E provisioning time multiple Transit Switches

HIT

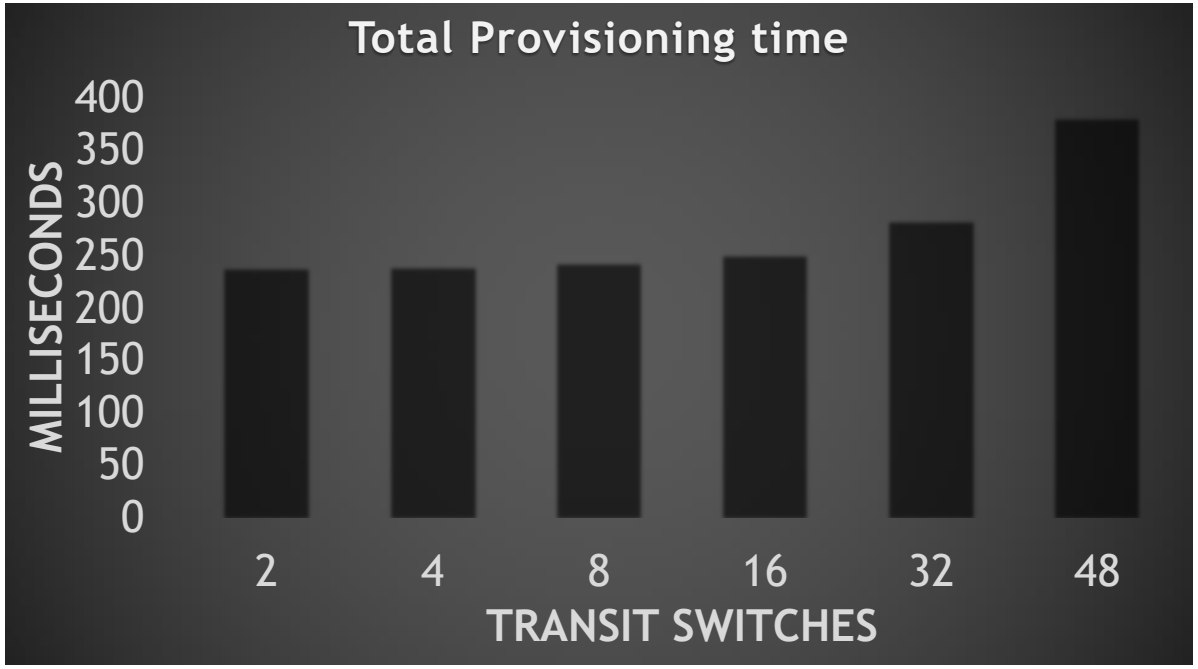
- Scale remains constant (until hitting test controller machine limits)

Overhead

- Primarily overhead on the host from creating the virtual interfaces by executing shell command (~250 ms).

IMPROVEMENT

- Expected to improve with production ready control-plane as it makes use of netlink.



Round Trip Time Effect on End-user

HIT

- Mizar direct path is faster than OVS+Linux Bridge. Though, Still has minimal impact on PPS and TCP BW.

HIT

- Even with an increased latency due to the extra hop, the packet per second processed by endpoints remains close to line rate.

Benefit

- Primarily benefit of fast-path is latency sensitive applications.

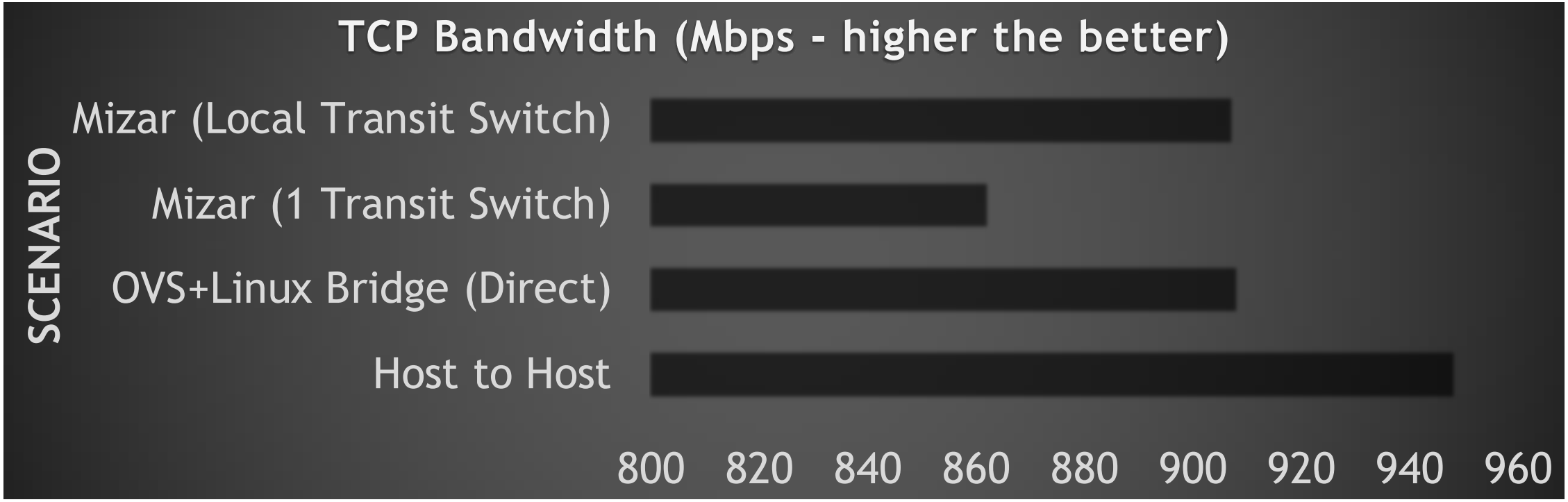
TCP Bandwidth (On a slow NIC 1Gbps)

HIT

- Comparable throughput to OVS+Bridge (even though we don't use XDP driver mode). *This is applicable for NICS < 4Gbps*

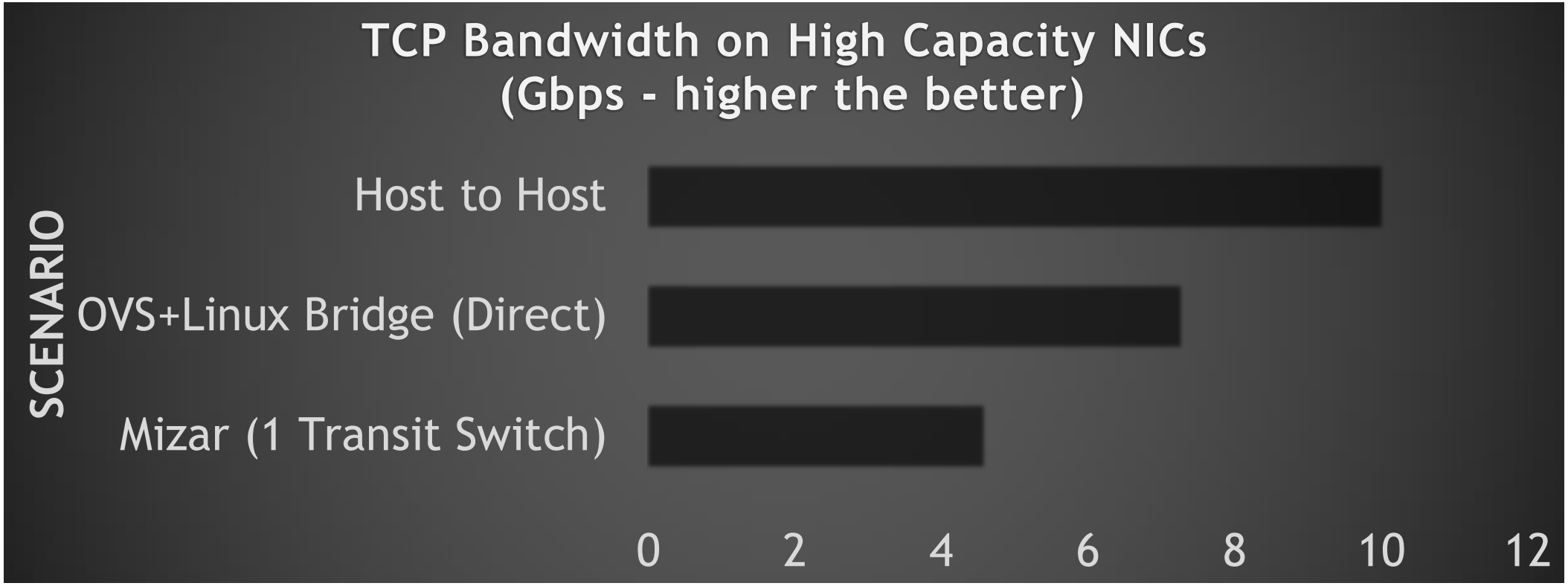
Hops:

- The transit switch hop accounts only for 5% less TCP throughput, which shall be negligible for very high bandwidth NICs. This is despite that RTT of the extra hop accounts for 45% more latency.

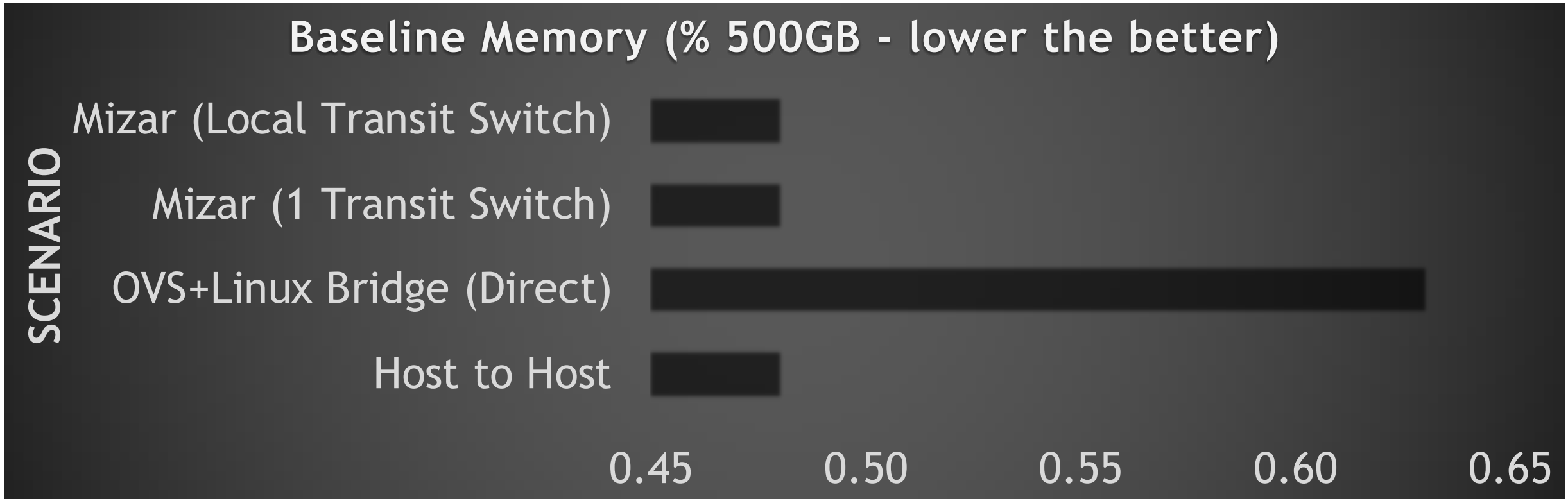


TCP Bandwidth (On a faster NIC 10 Gbps)

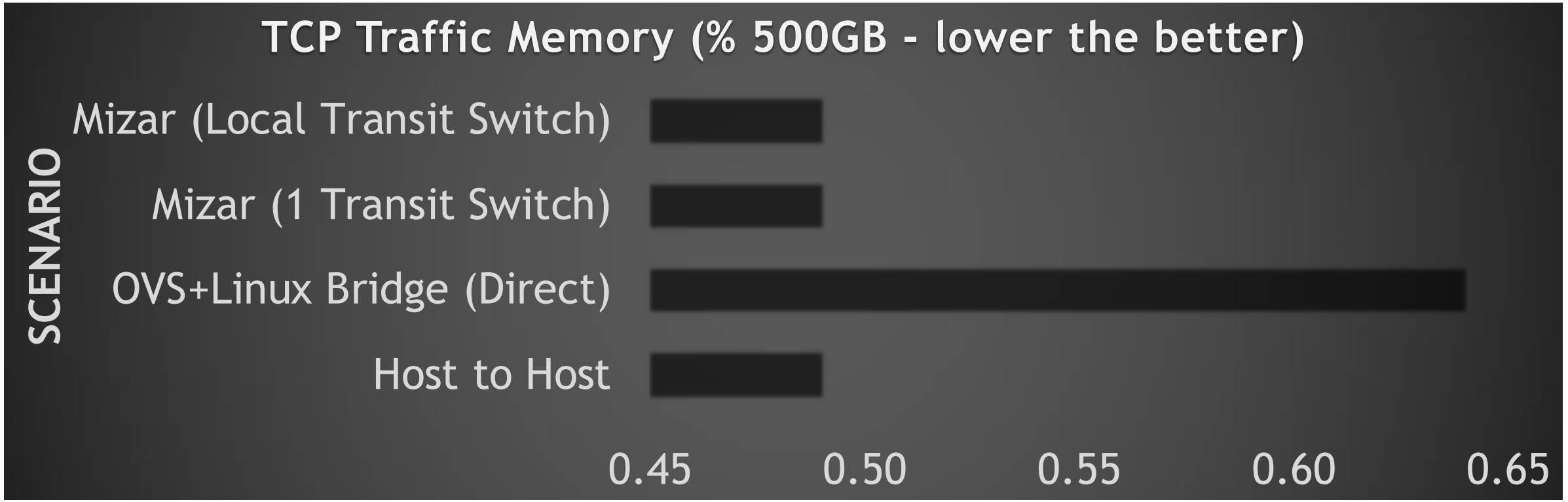
MISS:	IMPROVEMENT:	IMPROVEMENT:	IMPROVEMENT:
• The TCP bandwidth caps at around 4Gbps.	• Change to Driver mode (require support in NIC)	• Change on-host wiring architecture and reduce reliance on Transit Agent	• Improved device driver for veth



Memory Idle case



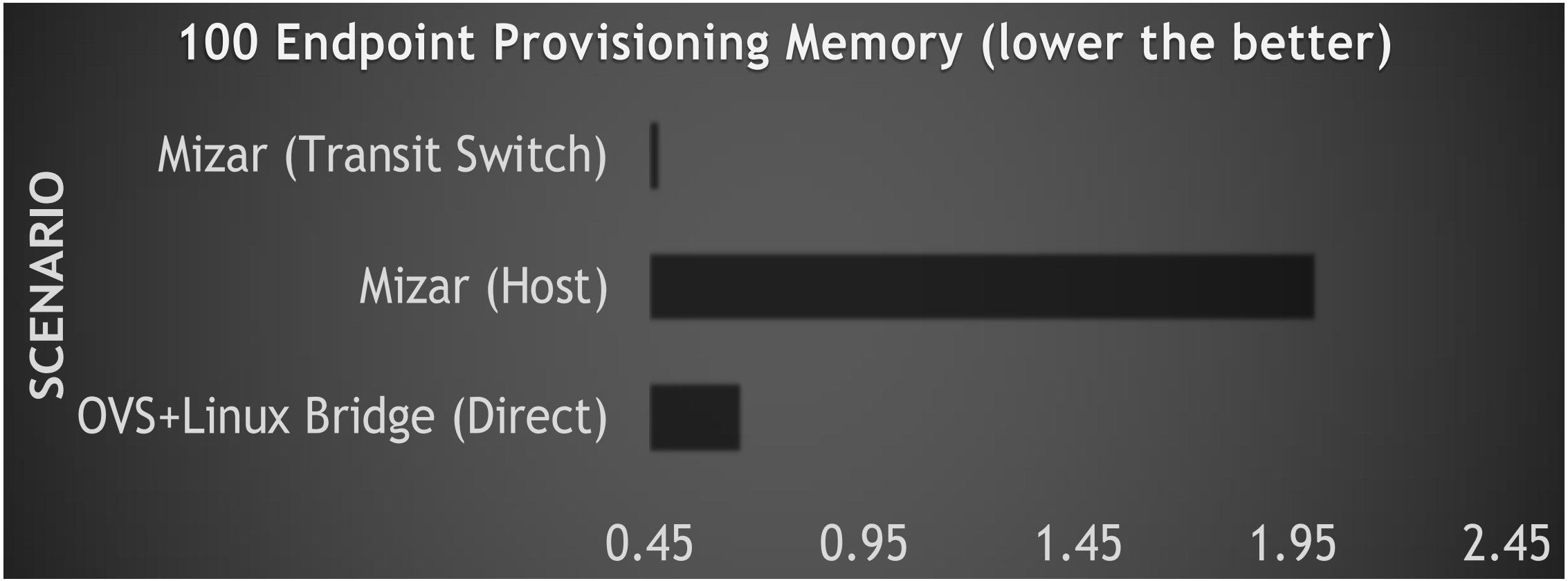
Memory During TCP Performance Tests



HIT

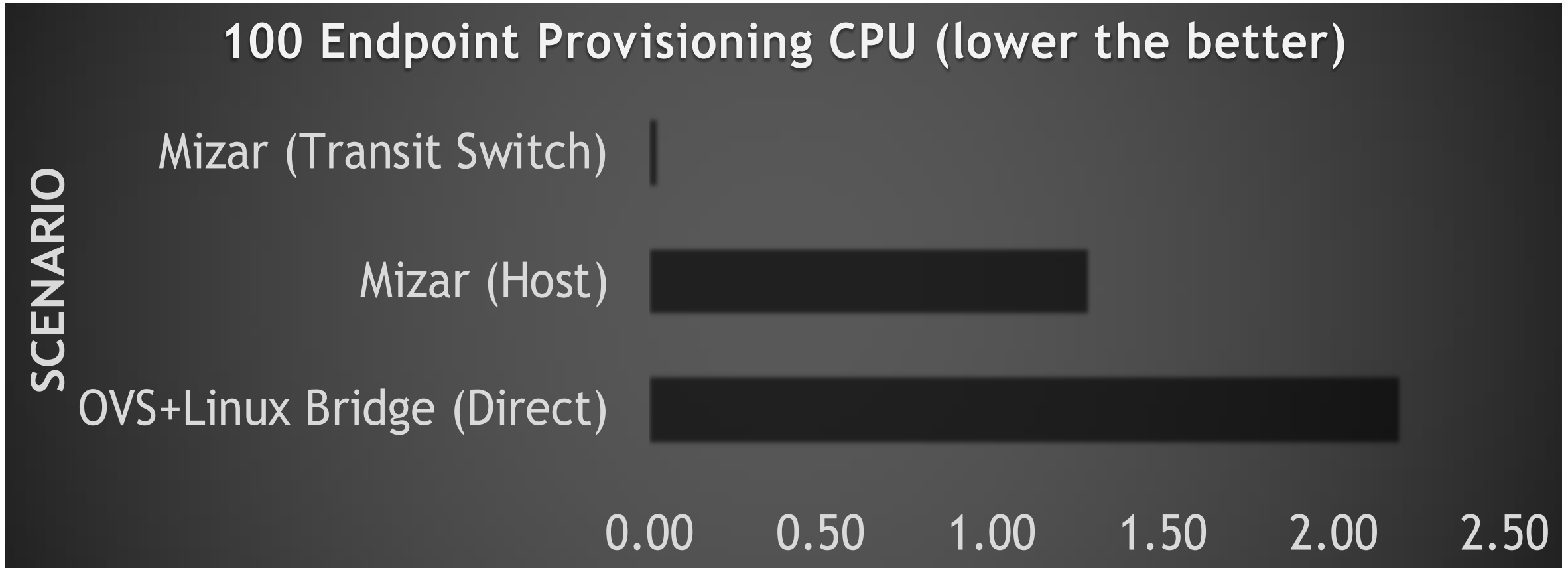
- Negligible Memory overhead very close to an idle host without networking constructs event with Traffic processing

Memory Idle case (100 Endpoints per host)



HIT	MISS	IMPROVEMENT
• Memory overhead on Transit Switch remain at baseline level	• On Host memory increases as we provision more endpoints	• Share one transit agent across multiple endpoints

CPU During TCP Performance Tests



HIT

- Significantly less CPU overhead during provisioning on both transit switch and host