SONiC Extensions

Adding Application Logic to Open NOS
What is SONiC+?

An application running in SONiC NOS, solving specific use case

Combination of application needs and dataplane capabilities

A ‘plus’ application should be containerized for modularity

Integrate with SONiC infrastructure i.e. SAI, Redis DB
- No impact on SONiC - Using SAI flex interface + REDIS DB
- All enhancements over SONiC are automated & auto generated
SONiC+ Telemetry

SONiC+ Flow base telemetry
SONiC+ Load Balancer

Making use of the SAI Flex APIs

- Stateful Load balancing @ switch line rate
- Can run on existing ToR switch excluding the extra load balancer servers hop
- Can run in addition to all SONiC functionality
SONiC+ – Components

- SONiC+ Application extension
- SONiC+ Orch Agent extension
- SONiC+ syncD extension
- SONiC+ SAI extension
SONiC+ Application Extension

FRR
TeamD
User App

SWSS API

APP DB
SONiC+ SAI Extension

Ability to add changes to data plane according to customer needs

https://github.com/opencomputeproject/SAI/tree/master/flexsai/p4
SONiC+ Tunnel

Making use of the SAI Flex APIs

- Support tunnel mapping to overlay networks
- Support VRF peering to increase routing scale
- Use case:
  - Connect bare metal servers to overlay networks

SONiC

- Tunnel encapsulation cache
- Tunnel decapsulation
- Underlay routing

VM

Bare Metal Host

Tunnel encapsulation
SONiC+ Tunnel

VNET peering in Legacy network

- VNET-virtual network
- VNET peering - Peering between virtual networks

Implementation:
- VNET -> VRF
- VNET1 peering with VNET2 -> copy route from VNET1 to VNET2 and vice versa

SONiC-BMToR

1K VMs and 100 VNETs will require up to 10M routes !!!
SONiC+ Tunnel

VNET peering in programmable network

- Two match action tables

- Port to VNET
  - Key: Port
  - Action Set metadata
    - metadata = VNET ID

- VNET routing
  - Key: metadata, prefix
    - metadata vector of VNET peers
  - Action: next hop

- VNET1 peering with VNET2 - > turn on VNET1 VNET ID in VNET routing metadata of all routes originated by VNET2
  - A single route per VM
  - Single update per VM route
SONiC+ Tunnel

Current

Orch Agent

syncD

Mellanox SDK

Auto generated API

SAI pipeline

Parser → port → BMTor PipeLine → Bridge → router → tunnel → deparser
SONiC+ - syncD extension

- Auto generate SAI API
- Dynamic link new SAI API
SONiC+ - Orch Agent extension

- Auto gen SAI Redis API
- Create APP DB abject ID in order to be able to correlate ASIC DB entry to APP DB entry
Application resource monitoring

- Use Abject id in order to map between ASIC resources to App resources
- ACRM generic code
- APP Abject id to ASIC may be vendor specific