

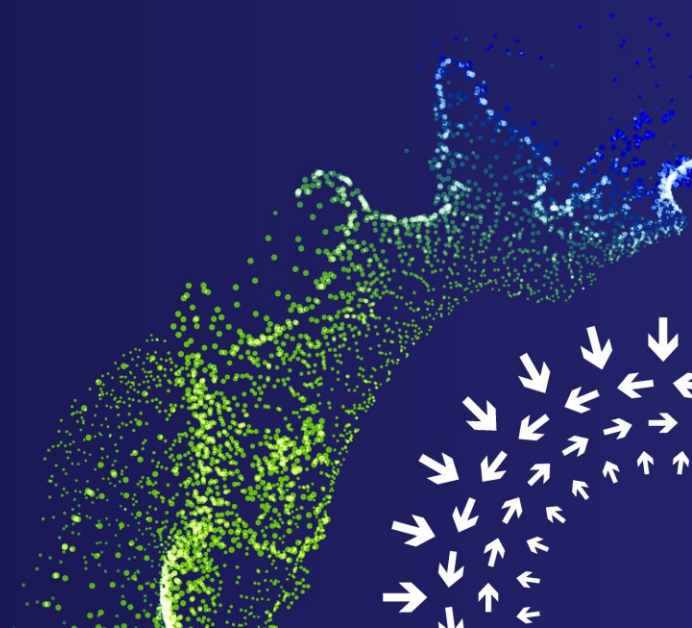
SRv6 for AI Backend Network

Changrong Wu, Microsoft

Abhishek Dosi, Microsoft



October 13-16, 2025
San Jose, California

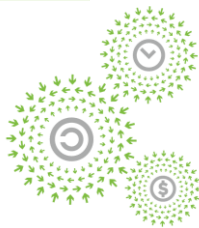




SRv6 for AI Backend Network

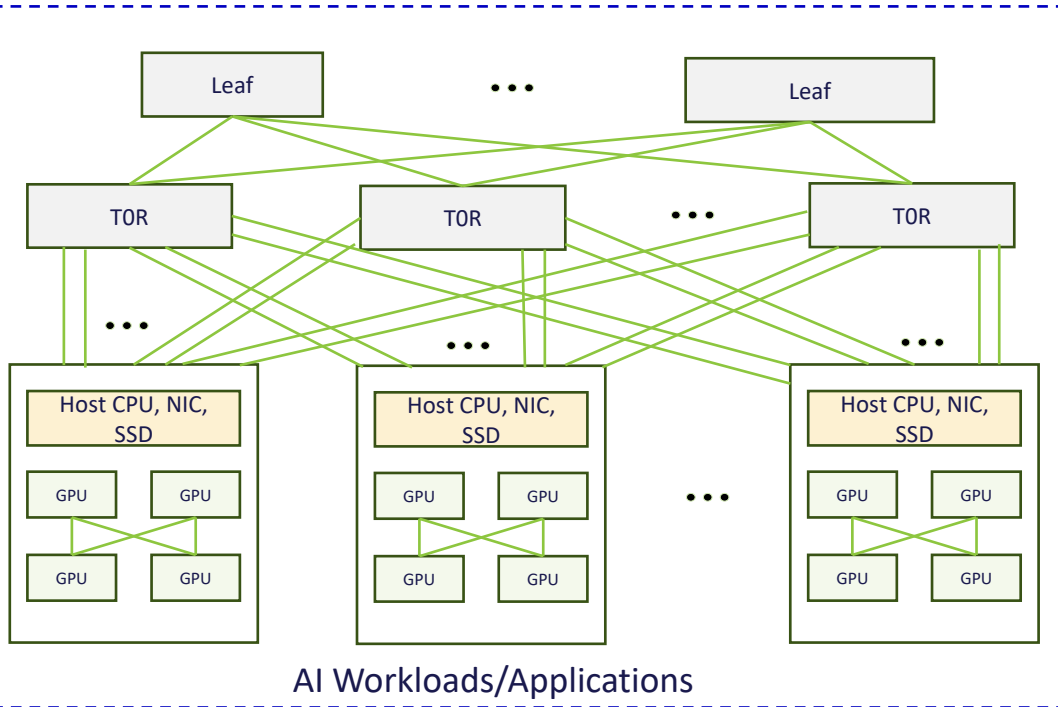
Changrong Wu, Microsoft

Abhishek Dosi, Microsoft



OPEN
PLATINUM®

Artificial Intelligence in the Cloud



New Traffic Pattern:

- Small number of large flows
- Periodic bursts of data sent synchronously

→ Dedicated Backend Network for AI

Continental-scale GPU Cluster Emerging



The bar for Hyperscale Datacenter Network is rising

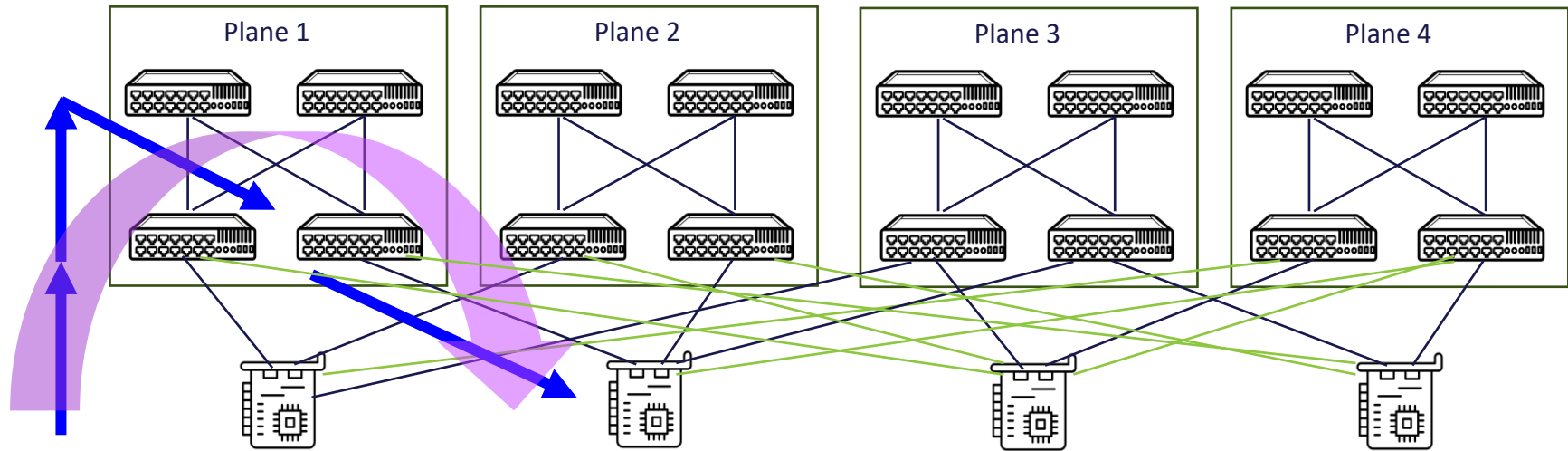
- Power Supply, Physical Space, etc. limit the scale of a single DC site.
- The demand of GPU capacity from a single job is set to grow beyond the capacity of a single DC.

→ The AI backend network needs to connect geo-distributed GPU clusters at scale

Challenges

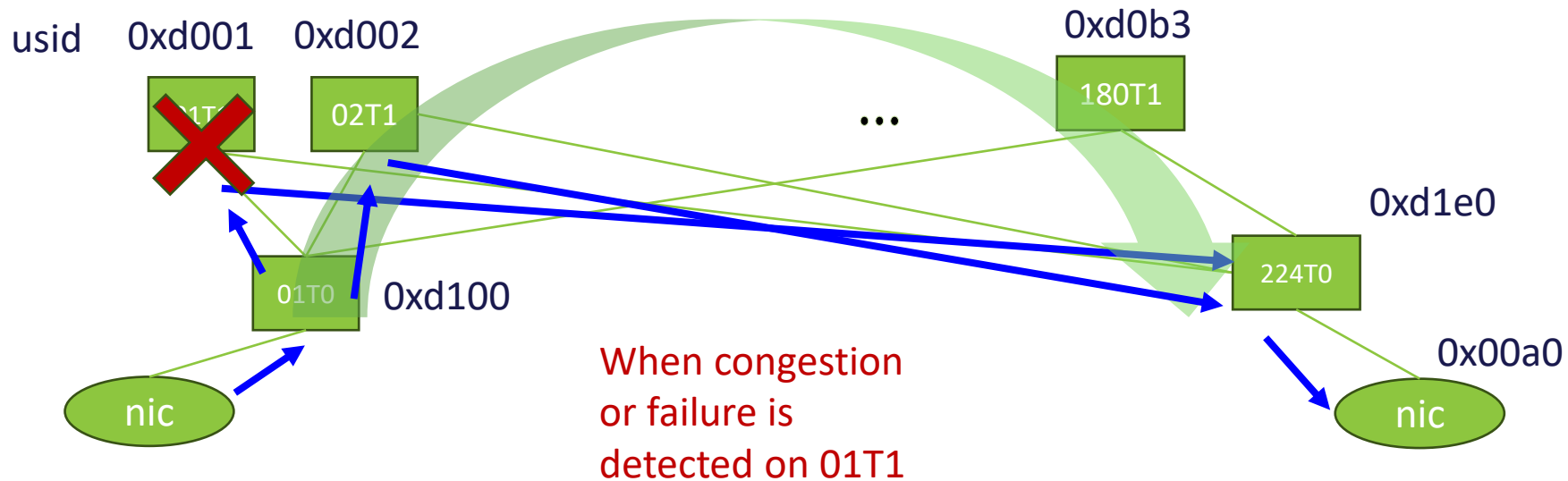
- The solution must be cost-effective and scalable
 - No Proprietary Technology
- Traditional passive hash-based load balancing mechanisms suffered from low entropy problem.
 - Need more active traffic engineering
- Failures is inevitable at this scale
 - Fast failover is necessary
- Multi-path transport is desired for efficient bandwidth utilization
 - Demand for fine-grained path control

SRv6 in AI Backend Network



- Provides fine-grained network control based on source routing
- Enables path enumeration for traffic management
- Integration with AI workloads flow scheduling provides optimal network performance
- Allow source to quickly reroute upon path failures or congestion

SRv6 with uSID



DstIPv6: *fcbb:bbbb:d100:d001:d1e0:00a0::*

NO SRH

1st hop 2nd hop 3rd hop 4th Hop

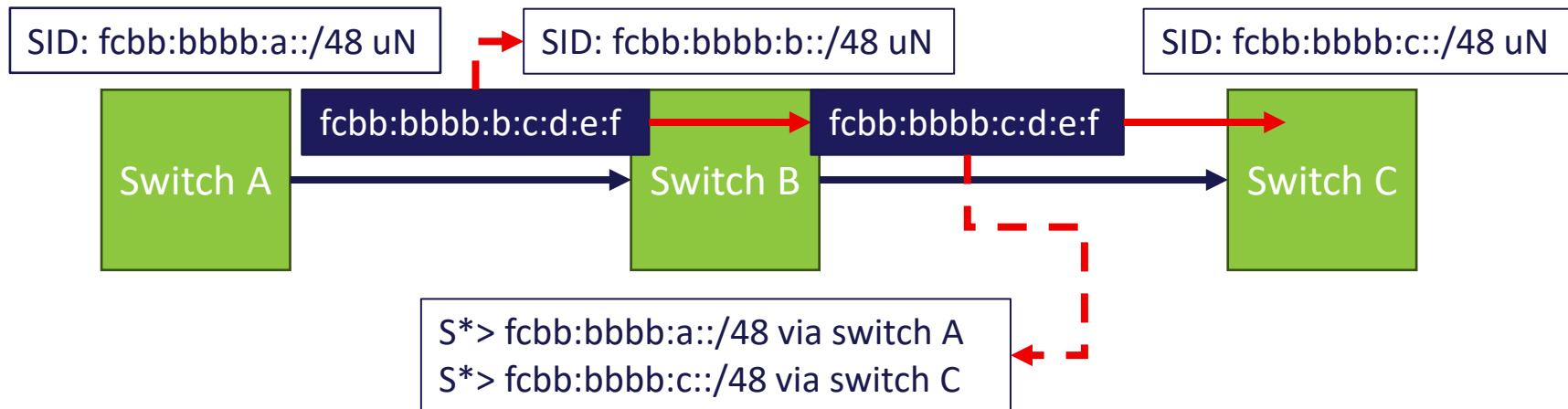


fcbb:bbbb:d100:d002:d1e0:00a0::

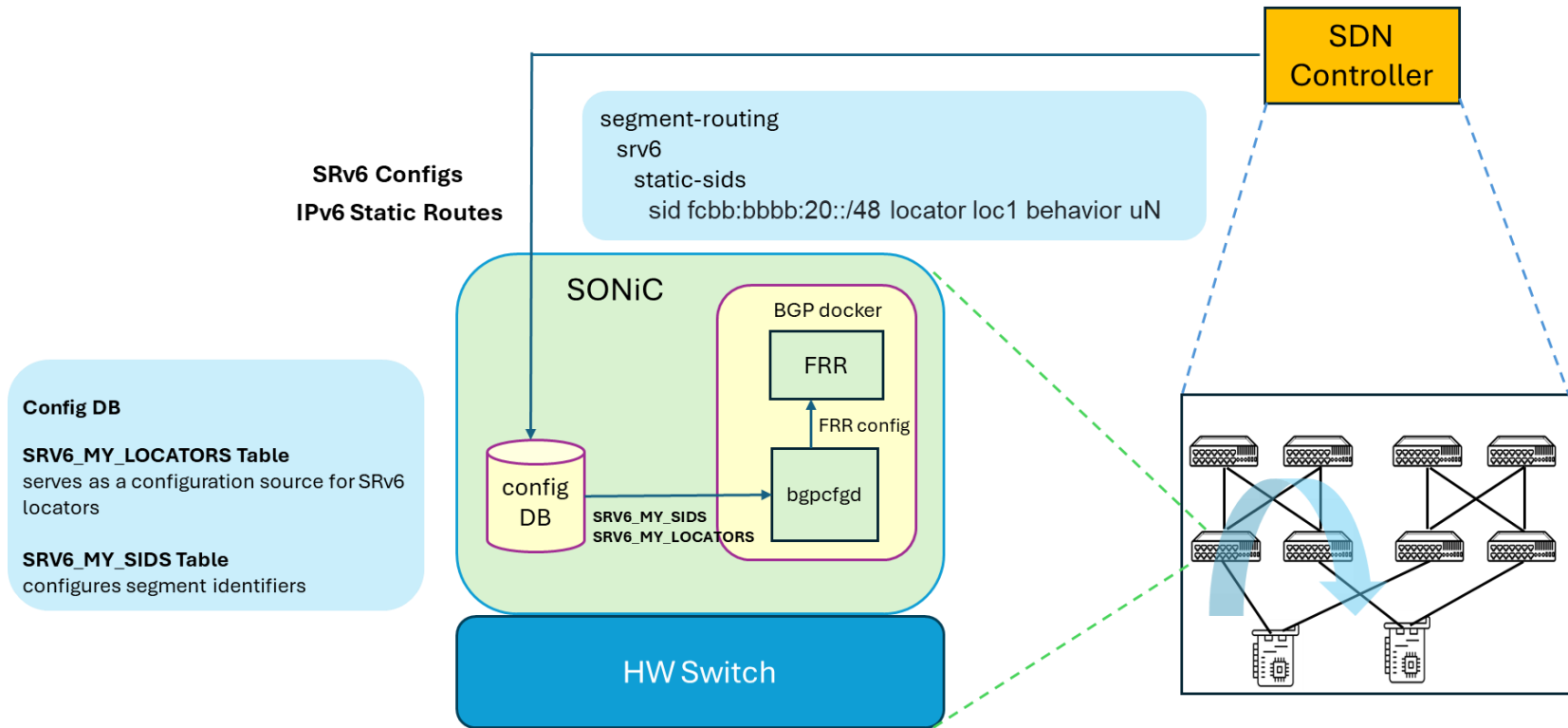
NIC switches to new path by change the uSIDs encoded in the packet header

Simplify SRv6 with Static Config

- Segment Identifiers(SIDs) are configured on switches statically.
- The switch has a static route configured for each of its neighbor's SIDs.
- Hosts get the list of SRv6 paths (encoded as a list of SIDs) that it should use from the central controller.



SRv6 with static uSID in SONiC



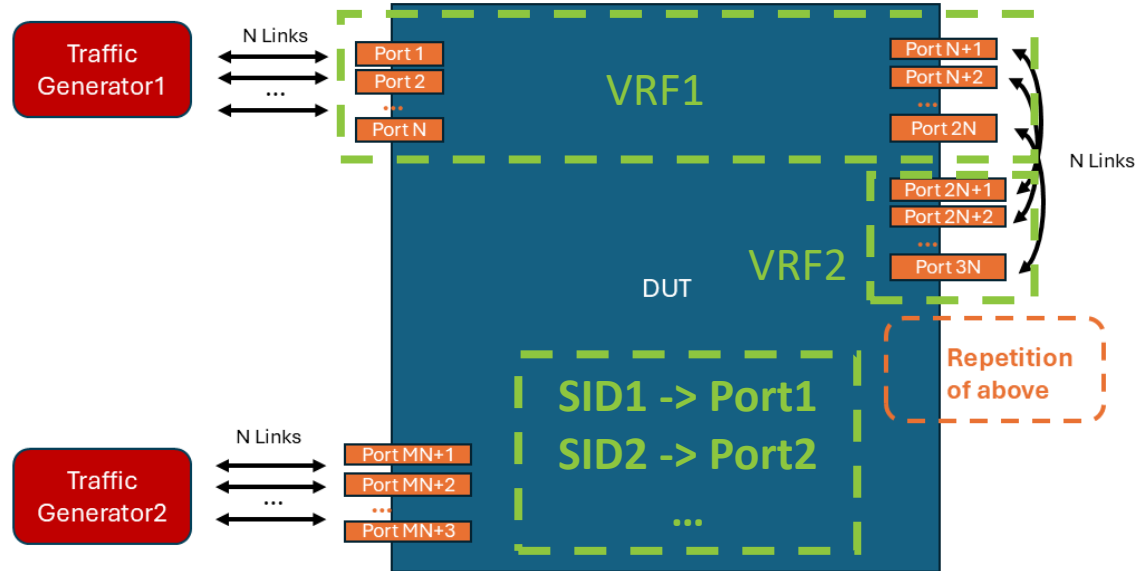
OCP SAI Extensions to Support SRv6 uSID

1. Allow uN and uA for SAI_MY_SID_ENTRY_ATTR_ENDPOINT_BEHAVIOR_FLAVOR
→ Enable USD/USP/PSP control for uSID
2. Allow uA for SAI_MY_SID_ENTRY_ATTR_NEXT_HOP_ID
→ Enable uA forwarding
3. Extend allow list of SAI_MY_SID_ENTRY_ATTR_TUNNEL_ID
→ Enable decap for uDT4/uDT6/uDT46
→ Enable decap for uN & uA when using USD
4. Allow uDT4/uDT6/uDT46 for SAI_MY_SID_ENTRY_ATTR_VRF
→ Enable egress VRF control for uDT4/uDT6/uDT46

SRv6 Stress Testing in SONiC

Build snake topology to maximize stress on the ASIC

- Use VRFs to avoid routing domain collision
- Use VLANs instead of raw interfaces to avoid MAC address collision
- Use multi-SIDs to allow packets selecting interfaces as egress ports



SRv6 Ecosystem in SONiC

- Mature Support
 - uN and uDT4/6/46 functions
 - Static SID allocation and provisioning
 - Static steering of traffic with SRv6 SID list
 - BGP-EVPN based L3VPN Services.
 - Evolving with FRR routing stack
- Rich community
 - Contributors: Microsoft, Cisco, Alibaba, Broadcom, Nvidia
 - Use cases: Telecom, Enterprise, Cloud Network

Future of SRv6 in SONiC

Why we only use uN for now?

- Easy for software/hardware support
- Intuitive usage

Next: the addition of uA

- End-to-end routing support, no static route needed
- Enable direct link-level path control (more fine-grained)

Future: define your own SRv6 function and bring it into SONiC!

Call to Action

Inviting contributions to SONiC community to introduce more technology into Open NOS

- SONiC/SAI
- Hardware platforms
- Testing and tooling
- Download, test, deploy!

Project Wiki with latest specification : <https://sonicfoundation.dev/>

Becoming a contributor: <https://github.com/sonic-net/SONiC/wiki/Becoming-a-contributor>

Mailing list: <https://lists.sonicfoundation.dev/g/sonic-dev>

SONiC Community meeting: <https://sonic-net.github.io/SONiC/Calendar.html>

Thank You!