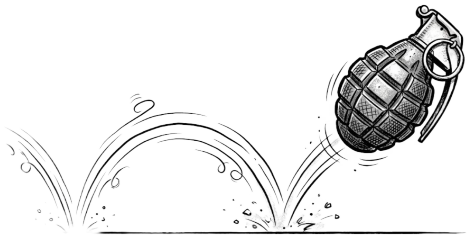
show me your PRs

a pragmatic chat about open source Network Operating Systems

nexthop ai



steve ulrich (sulrich@nexthop.ai)

NANOG 97 01-june, 2026

assertion

- we (the networking industry) need to start realizing the strategic benefits of open source operating systems
- network operators should be positioning themselves to take advantage of open source network operating systems - if not now, then as a lifecycle hedge

disclaimer

let's be clear - i'm not saying open source NOS are drop-in replacements for what you're currently deploying - i am suggesting you should be laying the groundwork to use open NOS as levers or hedges

so what? … why?

our peers in the compute/systems space have been able to realize significant gains leveraging open source operating systems…

- rapid adoption and deployment of multiple CPU architectures
  - x86, ARM, etc. - linux runs on everything
- linux is the de facto platform for innovation and development
- support lifecycle is largely decoupled from vendor whims

networking has largely remained proprietary & in the clutches of vertically integrated network operating systems

is networking actually special, or have we just been told it is?

what's an open source Network Operating System?

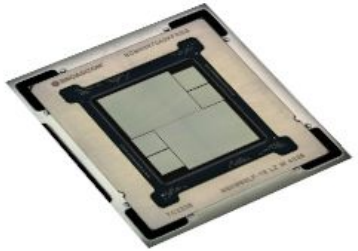
what do we mean when we say … "open source NOS"?

a switch/routing platform with an open base OS and a dedicated packet forwarding ASIC that exhibits the following characteristics:

- has a community driven governance and contribution model
- built upon a standard linux kernel base
- supports a breadth of hardware with a common abstraction interface to control the forwarding ASIC(s)
- supports open source routing protocols

SONiC is the natural center of gravity for this discussion but other projects meet this definition to varying degrees

yeah but: networking's special … (ASICs)



networking ASICs used to be special

this has historically been a load-bearing assertion for vendors

ASICs used to have tight NOS couplings - increasingly less so

lately - all the cutting edge ASICs come out with a SAI* (typically for SONiC) first

now, look deep within your soul … *are your features "special"?*

(more on this later)

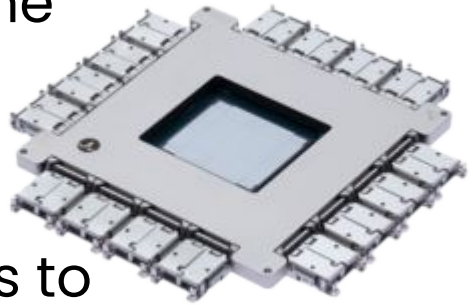
the latest state of things…

- DC feature sets and protocols are very well covered by SoC ASICs
  - increasingly adding support for novel forwarding behaviors
- campus switching applications
  - variants of DC architectures with additional first-hop auth (802.1X, etc.)
  - campus switching ASICs are usually chopped and mod'd versions of DC ASICs

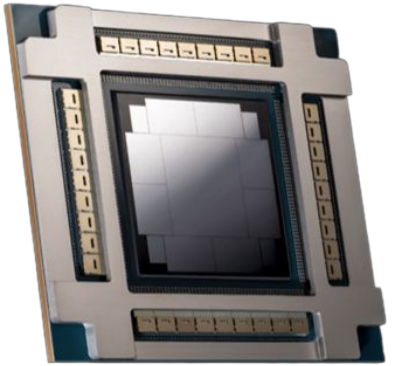
SAI - Switch Abstraction Interface - think of it as a driver for your ASIC

yeah but … networking's special … (optics)

vendors used to assert that the latest optics came on their vertically integrated NOS first



volume, ASIC alignment, etc. are driving suppliers to focus on ensuring that these technologies are increasingly available on open source NOS first



co-packaged optics, LPO, LRO, etc.

now - look for these on open NOS just as fast

yeah but … networking's special (features)

beware special features …

- you've been avoiding vendor feature lock-in already … right?
- n.b., there's probably some discussion re: scale that's warranted

if you truly need something special …

SAI potentially gives you access to "trick" ASIC features

- this often requires development and/or a partner with access to the SDK
- engagement with the ASIC supplier may be required

edge cases exist

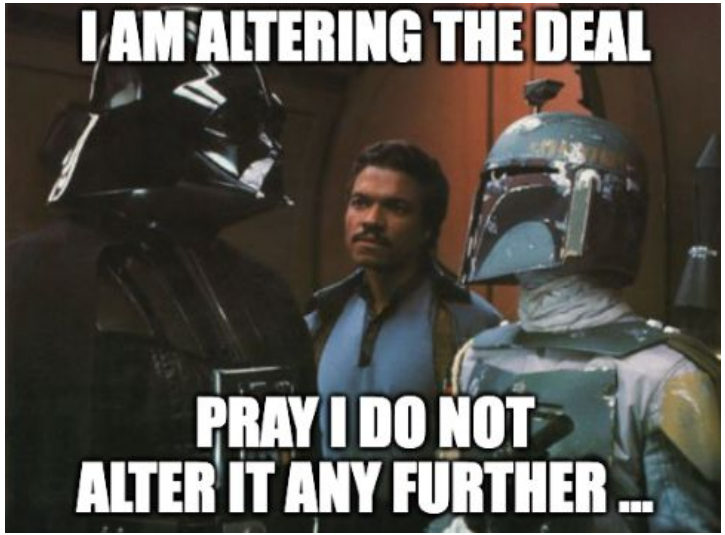
- exotic encryption mechanisms
- funky tunnel encapsulations (or combos of both of these …?)
- edge cases narrower than vendors imply

the right answer is usually "avoid these if possible"

yeah but … X is only for hyperscalers

hyperscaler problems are your problems too … just on different timescales.

- the core issues are the same
  - multiple NOS drive operational waste
  - vendor lifecycle whims strand hardware
  - management plane fragmentation
- networking vendor consolidation accelerates some of these issues
  - when Vendor X acquires Vendor Y will they keep that box around?
  - will the NOS be supported and patched as previously committed, or will there be product family deduplication?
  - when in the product lifecycle did you purchase, and will this see you through your intended lifecycle?



security … misaligned incentives

vendor incentives are to get you onto new hardware - not to patch the old stuff

operators on hardware that's physically capable but "out of support" face a choice between security compliance and CAPEX

open NOS shift the equation

- you can patch it yourself
- a support vendor can patch it
- the community can patch it

contrast with linux kernel LTS support models

- the ecosystem has figured this out

prolonged security compliance is a structural problem baked into the proprietary NOS model

sovereign infrastructure and supply chain

- geopolitical reality: governments and critical infra operators are increasingly worried about technology supply chain dependencies
- a NOS you can't audit, can't fork, and can't maintain without vendor permission is increasingly a liability
- relevant beyond government
  - any operator with regulatory requirements or data sovereignty obligations is likely going to need to start thinking about this
- SONiC as a practical answer
  - auditable, forkable, independently supportable
  - broad vendor ASIC support
  - open hardware designs

capitalist realities

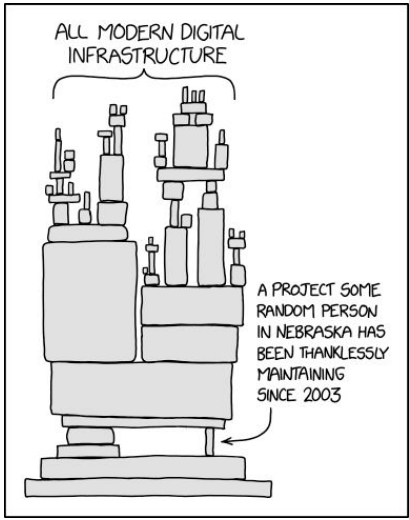
durable open source isn’t free

(one could contend it probably shouldn't be …)

- 💰💰 drives features & development priorities
- “maintenance taxes” must be paid

how do we avoid the xkcd dependency problem applied to network infrastructure?

- there needs to be reliable, distributed and verifiable investment across the community
- operators have a role to play
  - applying community pressure to open designs
  - ensuring vendors are being consistent in their support of open designs
  - working with/nudging vendors to build a sustainable ecosystem



gratuitous xkcd: Dependency reference

so … what do you push for?

before you buy the kit …

- target platforms that can run an open NOS in the future
- let your `${VENDOR}` know you require an open NOS offramp
- know the status of the SAI/driver support for the platforms of interest
- is it possible to get a community image for the platforms you're looking to deploy?
- will your vendors commit to providing this?

can your vendor *show you their PRs*?

vendors that are serious about open NOS show their work

- are your vendors making verifiable & public contributions?
- are they on the technical steering committees?
  - do they show up on the contribution reports?
  - can you readily identify their developers?
- are their contributions actually being upstreamed?
- are bootloaders available for their platforms?
- can the vendor provide details on how to build your own images with their drivers & SAI implementation?
  - nb: this doesn't mean you need to do this - but can you?

practicalities

- legacy platform → open NOS transitions; mostly unrealistic
  - there may not even be a pathway to SAI support for legacy h/w
  - new CAPEX is the best way to get `${VENDOR}` attention
- hardware interface novelty remains a real barrier
  - SAI / SDK coverage requires careful attention
  - secure boot & measured boot introduce additional twists
    - SONiC has mechanisms for supporting this
- current development leverage is concentrated in a handful of large operators - they drive the community roadmap
  - broad community engagement is required

where can you get your hands dirty?

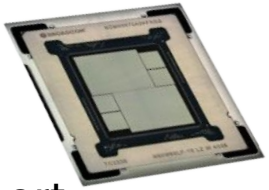
where open NOS makes sense today

- greenfield DC
  - modern L3 - leaf/spine designs have strong Si & feature support
- AI/GPU cluster fabrics
  - builds on the above, adds PFC and new hashing capabilities
- campus refresh cycles
  - solid Si alignment for many roles
  - volume and use case are interesting to many in the community
  - campus EVPN features emerging

where it doesn't (yet)

- Internet edge
  - capable hardware exists
  - control-plane and data-plane features are uneven
- legacy SP core infrastructure

this is a Q3D ASIC glamour shot



- Q3D has sufficient FIB capacity for the Internet DFZ
- Q3D hardware can support high-scale ACLs
- Q3D has SAI support
- open NOS do not (yet) support high-scale ACLs

what does good look like?

a durable open NOS ecosystem needs:

- ecosystem & governance model with commitment
- real vendor code contributions and support
- platform architectures that are open, published, & supported
- broad operator community pressure and commitment to adoption

the linux model can serve as a reference point

this is a nudge / reminder / call to action …

if we're not looking to create options - choices narrow

questions / discussion