Building an Open, Transparent NOS and Ecosystem Using Linux as the Key

Open Networking Summit 2018

Shrijeet Mukherjee

Chief Architect, Cumulus Networks
NOS
What is it really?
## Agenda for this session...

<table>
<thead>
<tr>
<th>What we <strong>WILL</strong> be talking about</th>
<th>What we <strong>WILL NOT</strong> be talking about</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>How Networking is ...</strong></td>
<td><strong>Networking as a vertical</strong></td>
</tr>
<tr>
<td>- Key to the data center</td>
<td>- It is NOT about best solution for a narrow problem</td>
</tr>
<tr>
<td>- An architectural choice</td>
<td>- Networking solutions that only belong to the physical network</td>
</tr>
<tr>
<td>- Inside the hosts/servers as well</td>
<td></td>
</tr>
<tr>
<td>- A data center wide interoperability problem</td>
<td></td>
</tr>
<tr>
<td><strong>Infrastructure use in the data center is evolving daily ...</strong></td>
<td></td>
</tr>
<tr>
<td>- VM, containers, micro-service meshes, etc.</td>
<td></td>
</tr>
<tr>
<td>- Networking needs to keep up</td>
<td></td>
</tr>
</tbody>
</table>
A Network OS .. What are the considerations?

Key Factors:
- Is there an ecosystem for the utility apps?
- Are custom apps portable, build environments, etc.
- A Platform isn’t just a forwarding ASIC, sensors, fans, thermals, reset all need to work and all need a model!

Network OS

Utility Apps
- ARP, DHCP, FRR

Custom Built Apps
- OpenR, Redist Neighbor

Most Portable

Least Portable

HW Platform

There is a best of breed OS ecosystem that provides All this
A general NOS lives in two worlds

**LEGACY**
- Complex Networks
- Interop critical feature
- Alphabet soup of protocols
- Historical and human operations
- Combination of VMware, bare metal, enterprise clustered apps, etc

**FUTURE DESIGN**
- Green field and purpose built
- Easy to scale
- Efficient
- Automated
- Feature velocity
- Homogenous app environment (typically your own)

Where Are You on the Scale?
Typical legacy NOS model

Most commercial NOS’s = Proprietary UI & Internal Daemons

What this means...

- No external validation model exists
- Any vendor change impacts deployment
- Custom-built to the H/W and SDK used

Portability achieved through standardization
Mega scale, custom networks = Simplified Networks with narrow use case

What this means...

- This is a step in the right direction
  Simplified network model
  Works for re-imagined networks
- Modern interaction model
  Uses JSON/REST exclusively
- Networking model is still proprietary
  May end up porting ARP
Can we do better?

**Best of all worlds**

**What this means...**

- Built upon a well understood and transparent network model
  - Enables building fully open protocols
  - Enables distributed development
  - Unthought of applications crop up
- Simulation becomes a no-brainer
  - Well worn standardized interoperability tools
Which model works for you? And why?

- **Custom CLI**
  - Proprietary protocol implementations

- **Open CLI & API**
  - Proprietary protocol implementations

- **Open CLI & API**
  - Open/Community protocol implementations

Which model works for you? And why?

- **User Space**
  - Hardware
  - Ports
  - SDK
  - VRF
  - FIB
  - NH
  - FDB
So what does Linux kernel networking mean?

Critical Difference...

The KERNEL is THE Data Plane Model for HW Programming

1. Kernel is a reference abstraction model that the HW will follow
2. Kernel defines how the HW will be programmed
3. Apps don’t have to care or know about new HW...the abstraction handles all that
A Data Center

How does the NOS fit in?
The Data Center Cookie Cutter (for Networking)

Attributes
- Elastic
- Mobile
- Automated
- Self-Healing

Infrastructure
- VMware/KVM hosts
- Container hosts

APP/Frameworks
- VMware apps
- Container apps
- Clustered apps

VM/Container
Virtual Network
Physical Servers
Physical TOR Switch
Physical Spine Switch
The Data Center Cookie Cutter (all networking)

Attributes
- Elastic
- Mobile
- Automated
- Self-Healing

Infrastructure
- VMware/KVM hosts
- Container hosts

Network
- Host networking is present
- Virtualized network functions

The Power of LINUX NETWORKING

APP/Frameworks
- VMware apps
- Container apps
- Clustered apps

Diagram showing physical TOR and spine switches connecting to virtual and container hosts.
The Data Center Cookie Cutter (all networking)

Establish a consistent ARP in your DC

Configuration & Management
Use the same tools for all elements

SIMPLIFY THE NETWORK
Apps can deploy anywhere
  Why not have DHCP on your TOR?
Networking is configured by templates
  Manage all networking similarly
Self-Healing, self debugging
  Tools have full visibility

VM/Container
Virtual Network
Physical Servers
Physical TOR Switch
Physical Spine Switch
Imagine Your Network being ...

Agile

Extensible

Portable

An Easy Assessment

Are You Running the Linux Stack?

1. Do your networking apps “just work?”
2. Can you use any app on the Host AND on the Network unchanged?
   
   *Did you need to port ARP?*
3. Is your data plane modeled by the Linux kernel as is? Or does it do special things that cannot be validated without hardware?
# Why Linux in a NOS

<table>
<thead>
<tr>
<th>Linux Stack is Everywhere!</th>
<th>Unify the DC... Speak Same Language</th>
<th>Total Insight Across DC</th>
<th>Supply Chain Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 500 Supercomputers run Linux*</td>
<td>president</td>
<td>Apps: 40% of top 500 Supercomputers run Linux*</td>
<td>70K Applications</td>
</tr>
<tr>
<td>Apps: 67% of all web servers</td>
<td>Host: 70% of all web servers</td>
<td>Network: 70% of all web servers</td>
<td>Freedom: 70K Applications</td>
</tr>
</tbody>
</table>

*Data as of 2022
Value of Linux Community

50k

Linux Community

5K

Developers from 500 different companies contributed to Linux Kernel in past year

Treasure Trove of An Ecosystem

VALUE

• Fast security patches
• Not reliant on vendor updates
• Easy new feature introductions
Linux as a NOS
A little more detailed look
The Vibrant **Kernel Ecosystem**

- **Ethtool**
  - Interface naming
  - 25G, 100G standards and management

- **Bridging Model**
  - Active-Active (MLAG)
  - IP Multicast, IGMP
  - MLAG primitives
  - VXLAN scale

- **VRF**
  - New device model

- **Route Path Selection**
  - IPv4/IPv6 uniformity
  - Nexthop management

- **EVPN**
  - ARP suppression
  - Learning optimizations

- **MPLS**
  - LWT
  - MPLS datapath
But you need a greater ecosystem

*Driven by the kernel*

<table>
<thead>
<tr>
<th>Ecosystem</th>
<th>USER INTERFACE</th>
<th>PROTOCOLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONIE</td>
<td>Ifupdown2</td>
<td>EVPN</td>
</tr>
<tr>
<td>Created the platform separation</td>
<td>Handle interface scale</td>
<td>In Hypervisor</td>
</tr>
<tr>
<td>FRR</td>
<td></td>
<td>MPLS</td>
</tr>
<tr>
<td>Routing performance VRF Control Plane</td>
<td></td>
<td>Complete IP tunnel story</td>
</tr>
</tbody>
</table>
Don’t vendors need to innovate?

**Application interface:** Portable and maintainable apps can use this layer

- Non-portable apps/infra
- Highlighting narrow use cases
- Hair pulling

**Low level device interface:** Empowers system software developers, but only useful in the context of the full system around it.

- Sustainability
- Improved experiences
- Switchdev is a project to join
Linux

Pros and Cons
Do You Want Features? Or Architectural Innovation?

INNOVATION IS ARCHITECTURAL

Tenancy and isolation
Use Linux VLANS, VXLAN and VRF consistently in the network and the hosts

Convergence
Transparent and well understood layering
Linux tools like kexec allow for robust, well understood and broadly applicable implementations

Simulation is a necessity
Use Linux, kernel becomes the native data plane
Existing tools like GNS3 are box simulators

Try running a topology with VXLAN, VLAN, VRF’s in them
Are there downsides?

1. The kernel community has some rituals
   But they are very welcoming
   Procedures are documented, try it

2. Living in an open ecosystem has commercial complexity
   No private intellectual property
   You have to enable your competitor
   Not all investments have clear benefits

3. All technology will get commoditized
   Or they will disappear
DEMO

Unifying the entire fabric w/ Linux
The topology

**L2 topology used**

- STP needed to keep system sane
- Host is using a Linux Bridge
- Connected to TOR using multiple links
- TOR to spine also has multiple links

---

VM/Container
Virtual Network
Physical Servers
Physical TOR Switch
Physical Spine Switch
Use Linux for your data center networking

- Create a loop by
  - It has the best ecosystem (you may have people already)
  - Automation tools and ecosystem is built-in
  - Debuggability and transparency are built-in
  - Feature complete for the Enterprise use case

Unified Linux scanner can identify the broken network element **ANYWHERE** in the system
Join the Linux Networking Revolution

Linux 101 eBook

Linux Networking Guides

Linux Resource Center

QUESTIONS?

www.cumulusnetworks.com/lp/linux-networking-resources/