Device/Control Plane Models

Controller

Network Device

Open standard control plane

Network Device

Open Standards

Open Source NOS

Network Device

Disaggregated
<table>
<thead>
<tr>
<th></th>
<th>SDN</th>
<th>White Box</th>
<th>Disaggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower cost hardware</td>
<td>Potentially</td>
<td>X</td>
<td>Potentially</td>
</tr>
<tr>
<td>Centralizing control and management</td>
<td>X</td>
<td>Potentially</td>
<td>Potentially</td>
</tr>
<tr>
<td>Application integration</td>
<td>X</td>
<td>Orthogonal</td>
<td>X</td>
</tr>
<tr>
<td>Business alignment</td>
<td>Potentially</td>
<td>Orthogonal</td>
<td>X</td>
</tr>
</tbody>
</table>
Disaggregated Model

Routing Stack

- BGP
- IS-IS
- Configured Routes
- RIB
- Platform Abstraction (PAL)
- Hardware Abstraction (HAL)
- Forwarding ASIC
- Fan/LED/etc.

Kernel

ONIE
Platform Abstraction Layer

- Often one of the hardest components to source
- Must connect your hardware platform with your chosen stack and O/S

- Provided by
  - Hardware vendor
  - Software vendor
  - Consulting companies will also write these
## ASIC Hardware Abstraction

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| SAI       | Supports (pretty much) all chip vendors  
|           | Pluggable architecture |
| OpenNSL   | Broadcom only |
| P4        | Barefoot Networks  
|           | Programming language rather than an API |
| asicd     | Snaproute’s interface to a wide variety of asics |
| swtichd   | Cumulus’s interface to a wide variety of asics |
| fd.io     | Based on DPDK  
|           | Largely focused on accelerated NICs, rather than network switching hardware |
BGP
Performance & Scale fixes
AddPath Support
Remote-AS internal/external Support
BGP Hostname support
Update Groups
RFC 5549 (unnumbered) Support
Next hop tracking
32-bit route-tags

RIB (Zebra)
MPLS Support IPv4/v6 for static LSPs
32-bit route-tags
Next hop Tracking
RFC 5549 (unnumbered) Support

OSPF (v2/v3)
OpenBSD Support restored
32-but route-tags
RFC 5549 (unnumbered) Support

LDP
RFC 5036 (LDP Specification)
RFC 4447 (Pseudowire Setup and Maintenance using LDP)
RFC 4762 – (Virtual Private LAN Service (VPLS) using LDP)
RFC 6720 - The Generalized TTL Security Mechanism (GTSM) for LDP
RFC 7552 - Updates to LDP for IPv6

Others
JSON Support
VRF Lite
Snapcraft Packaging
FR ROUTING

NEXT VERSION 3.0

BGP
BGP Shutdown Message
Large Communities (RFC8092)
eVPN (partial) (RFC 7432)
IDR Tunnel (draft-ietf-idr-tunnel-encaps-03#section-3.2.1)
IPv6 VPN (misc fixes)
IPv4/IPv6 VPN Graceful Restart

PIM
Unnumbered interfaces
MSDP (RFC4611)
Sparse Mode (RFC4601)

NHRP
NHRP (RFC2332)
( Linux only, for NBMA-GRE tunnels; no ATM; not supported on BSD)

Label Manager

LDP
Unnumbered interfaces
Capabilities (RFC5561)
Typed wildcard FEC (RFC5918, RFC6667)
Advertisement completion (RFC5919)
Controlling State Advertisements (RFC7473)

IS-IS
SPF Backoff

OSPFv3
Authentication/Confidentiality (RFC4552)

CLI
Parser rewritten in Bison
Lexer rewritten in Flex
Definition grammar overhauled
FR Routing - What’s different?

- Methodical vetting of submissions
- Extensive automated testing of contributions
- Git Pull Requests
- Github centered development
- Elected Maintainers & Steering Committee
- Common Assets held in trust by Linux Foundation
FR Routing – How to get it

• Binary package
  • Snap package available now
    • Snap is a new universal package format – see snapcraft.io
    • FRR 2.0 in stable channel and FRR 3.0 in beta channel
  • Debian / Ubuntu / RedHat packages coming soon
  • Other packages will follow

• Source
  • Github (https://github.com/FRRouting/frr)
    • Branch stable/2.0 → Released Version 2.0
    • Branch stable/3.0 → Version 3.0 (upcoming release)
    • Branch master → Latest development (“unstable”)
Some Architectures
configuration and management tools

- Ansible
- Puppet
- Chef
- Telemetry

third party & custom

- OCP
- Business Application

network applications

- SNMP
- BGP
- DHCP
- IPv6
- ... More apps

SONiC Base

- Database
- Platform
- FR Routing
- LLDP
- SWSS
- Utility
- RedisDB
- SYNC
Network Applications

Switch Services (SWSS)

Object Library

Orchestration

AppDB

SAIDB

synchd

SAI

ASIC

- persist app objects
- persist SAI objects
- redisdb w/ object library
- synchs SAI objects from software to hardware
- translation between apps and SAI objects
- resolution of dependency and conflict

SONiC
Challenges
“One Neck to Choke”

• Also known as…
  • A “single point of failure”
  • “my vendor makes all of my architectural decisions”

• Do you really have this today?
  • Be honest! 😊
Market Challenges

• This is an immature market
  • Vendors and projects are in flux
  • Projects are often based on small communities

• Skill set often == unicorns
  • There are no certifications, paths, etc.
  • You have to be an engineer/architect
    • Rather than “just” a CLI/vendor jockey
  • You have to be “full stack”
    • Integrate business architecture with network architecture
    • Understand applications, how they use the network, etc.
    • Know the bits and pieces of a router, what they all mean, etc.
    • Keep up with ten different sources, rather than one
Hardware Challenges

• Silicon support
  • Route count
  • Queue depth/buffering
  • Label imposition depth
  • Many others—this is an area where you must be careful

• Project/Vendor overlap
  • Most ASICs are supported by most every option covered here
  • System/support chipsets are a different story
    • Fans, LEDs, CPUs, other components
  • Be very careful to ask about this when building a system
Other Challenges

• *We ain’t got no features*
  • But part of the point is to stop throwing features and nerd knobs at every imaginable problem

• No tech support *unless you buy it*

• You must be an educated consumer
  • Participate in open standards
  • Pay attention to provider venues, papers, etc.
  • Much is under NDA

• If you’re using open source, you should be a part of the community