

POSTED ON MAR 10, 2015 TO [DATA CENTER ENGINEERING, NETWORKING & TRAFFIC](#)

## Facebook Open Switching System (“FBOSS”) and Wedge in the open



By Adam Simpkins



Last summer, we shared that the networking team at Facebook was developing its own data center switches and switch software. The switches and software are now in production across multiple data centers. Today, we are happy to announce the initial release of our Facebook open switching system (code-named “FBOSS”) project on GitHub and our proposed contribution of the specification for our top-of-rack switch (code-named “[Wedge](#)”) to the OCP networking project.



---

*Wedge, the TOR switch*

## Anatomy of a Network Switch

Modern network switches perform almost all packet forwarding in hardware. If forwarding were handled by software, even the most powerful CPUs would achieve only a fraction of the speed expected of current switches. Therefore, switches contain [ASICs](#), specialized hardware that is designed specifically for routing and forwarding packets at very high speeds, and with limited interaction with software running on the main CPU.

Even though forwarding ASICs can handle most packets without software interaction, they still need to be programmed and told how to process packets for different destinations. This is where the FBOSS software comes in.

The initial [FBOSS](#) release consists primarily of the FBOSS agent, a daemon that programs and controls the ASIC. This process runs on each switch and manages the hardware forwarding ASIC. It receives information via configuration files and thrift APIs and then programs the correct forwarding and routing entries into the chip. It also processes packets from the ASIC that are destined to the switch itself, such as control plane protocol traffic, and other packets that cannot be processed solely in hardware.

Our Wedge top-of-rack switch follows this basic design and uses a single Broadcom Trident II ASIC for high-speed forwarding. We have proposed the specification as a contribution to the Open Compute Project and will be working with the OCP networking group/ecosystem to review Wedge and to integrate Wedge with other software projects, such as ONIE and Open Network Linux. Wedge will be available through Accton and its OEMs and channel partners.

The design of Wedge itself is distinctive from other top-of-rack switches in that we've embraced the idea of making the switch feel as much like a server as possible for ease of management and software development. Further, Wedge is the basic building block for our new chassis switch (code-named "6-pack"), so once you're familiar with the Wedge specification, you'll be pretty close to understanding 6-pack, as it is simply a combination of Wedge boards networked together with Ethernet on the backplane.



6-pack chassis switch

## FBOSS — a set of applications, not an operating system

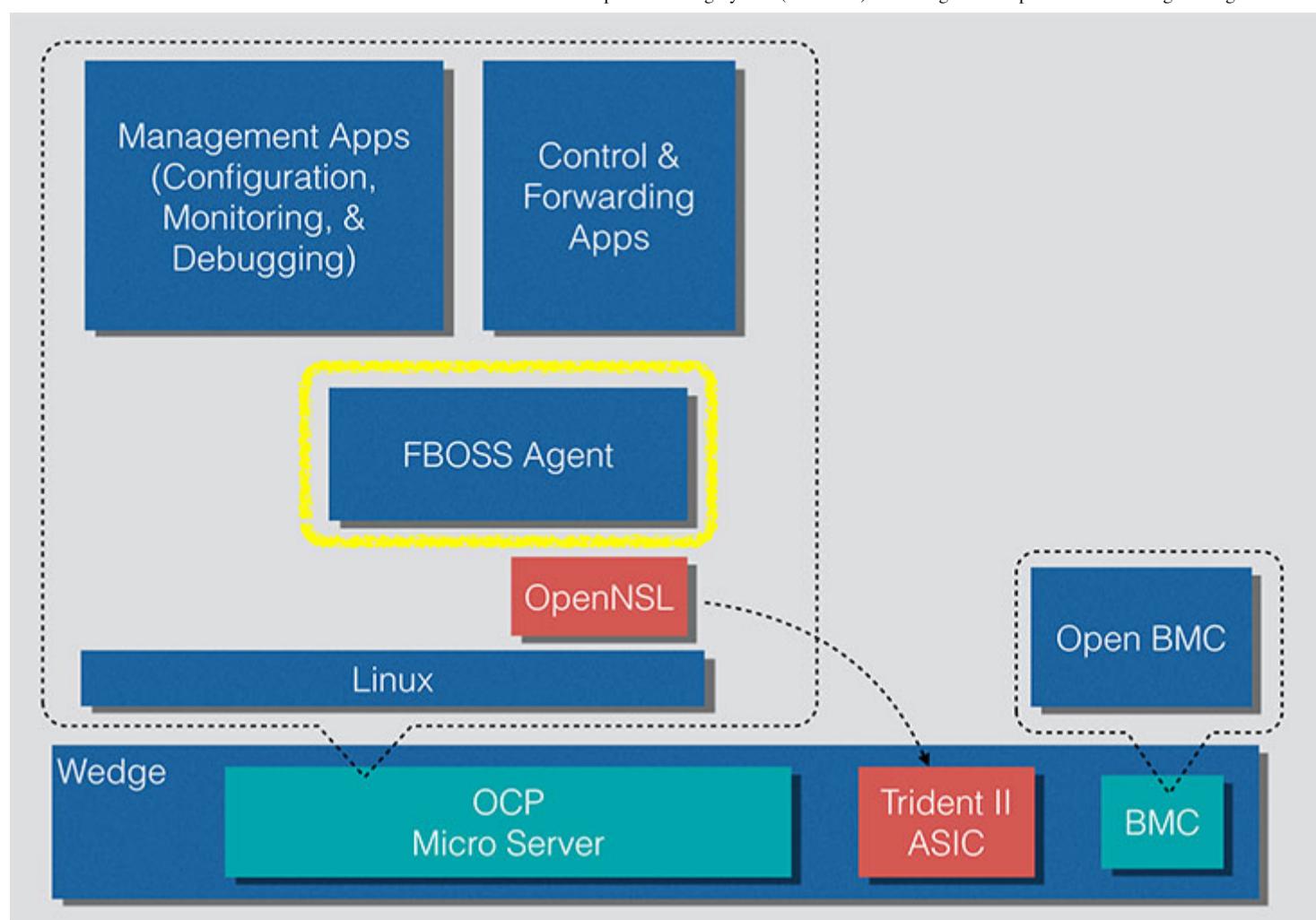
The actual design of our switch and switch software is different from most existing network switches. If you have dealt with existing network switches before, you probably know that most run proprietary operating systems. Traditional network vendors sell hardware and software bundled together, and switches from any given vendor will run only that vendor's operating system. Facebook and the OCP networking project have been explicitly driving change in that monolithic model by separating or "disaggregating" the network hardware and software.

Despite having "OS" in its name, FBOSS is not a full operating system. Rather, it is a set of applications that can be run on a standard Linux OS. Remember, we wanted to make switches feel like servers. For example, we deploy a set of applications supporting big-data-style computations onto specific tiers of servers, while we deploy packages like proxygen onto servers in our web tiers. We do the same thing for our network switches — each is just like another server that needs an FBOSS set of packages/applications.

At a high level, we can group the applications in FBOSS into several large categories:

- Low-level apps such as the FBOSS agent that deal directly with the forwarding ASIC. We also have OpenBMC, low-level software that provides management functions for power, environmental, and other system-level modules. The FBOSS agent and OpenBMC are the applications that we are contributing to open source now.
- Automation apps for configuration, monitoring, and troubleshooting. FBOSS is designed to be controlled primarily via software, rather than by humans. Instead of providing a command line interface to configure it, FBOSS has thrift APIs for interacting with it (on top of which command line interfaces can be built). Configuration changes are performed using higher-level automation tools that can consistently update many hundreds or thousands of switches at a time.
- Control apps that use the FBOSS agent to implement particular forwarding/routing protocols or support centralized decision-making. These apps, such as the one that implements BGP, can be developed independently of the actual hardware as they build upon the lower-level apps like FBOSS agent.

This means that FBOSS is not tied to a specific Linux distribution, and it is easier for developers to modify and hack on it. It is also possible to use only certain pieces of FBOSS, and build new and different applications on top of it. Our goal is to help make networking hardware that is open, and to foster a wide variety of open source software that can run on top of it.



## FBOSS Agent

FBOSS currently provides a very lean feature set. It has been designed with Facebook's data center network in mind, so we have implemented features important for this role.

While we have a very sophisticated [fabric topology](#), our network is built using a fairly small, simple feature set. We use L3 unicast routing for all traffic, with ECMP for balancing traffic across links, using BGP to configure the routes throughout the network. We do not support multicast traffic within our data center, nor do we use features such as L2 overlays, spanning tree, or even trunking.

This limited feature set helps keep the implementation of FBOSS relatively simple and has allowed us to build it using a very small team of developers.

Specifically, the FBOSS agent includes support for:

- Programming various tables within the Broadcom ASIC, such as L2, L3, and VLAN tables.
- Handling low-level control packets for host and neighbor learning (ARP, IPv6 NDP, DHCPv4/v6 relay, LLDP).
- Packet parsing/construction of ICMP/UDP packets.

Our implementations work at scale within the Facebook network — chances are if you're using Facebook right now, your requests are likely being handled by machines connected via FBOSS. However, we know the implementation for nearly all the protocols is partial, as we currently implement only those portions of the RFC that are needed within the Facebook environment.

## OpenNSL

Up until now, building open source switching software has been difficult, because there are only a handful of companies that build switching ASICs. The specifications and SDKs for these ASICs have generally been kept secret, available only under nondisclosure agreements. Even if you wanted to open-source your own switching code, you couldn't release all of it because the lowest layer of it that uses these SDKs to program the ASIC would have to remain closed.

Aided in part by the efforts of the OCP, several ASIC vendors are now beginning to open up some of their APIs and SDKs.

The initial FBOSS agent release is targeted for the Broadcom StrataXGS series of Ethernet switch ASICs (specifically the Trident and Trident II chips). Broadcom has now released its [OpenNSL APIs](#), and the FBOSS agent can leverage those APIs to program the ASICs. Because OpenNSL has been released, we can open-source the FBOSS agent, allowing others to see how we program the Broadcom ASIC.

## Looking Forward

Like most software projects, FBOSS is still very much a work in progress. While it is already running in our production data centers, it is still a fairly young project, and there are a lot of features and improvement still planned on our roadmap.

Since FBOSS has been designed specifically with the Facebook network in mind, it likely will require some additional development and modification to support using it in other environments. It isn't yet at the state where nondevelopers can easily download FBOSS, run it on their own hardware, and easily deploy it in their network. For now, we think the FBOSS code base will be of interest primarily to other developers.

Our initial release of FBOSS also does not include any of the control/routing protocol code yet. Much of our routing code is very specific to Facebook's data center design and would not be of much use in other networks. However, it should be possible to modify existing open source routing daemons to call the FBOSS agent's thrift APIs for adding and removing routes.

We are committed to continuing FBOSS development in the open and working with the community to build a rich set of applications that can run on OCP networking hardware. It's an exciting time for open networking development, and we are looking forward to seeing what we can achieve together with the community.

Many people worked on this project. Credit should go to: Tian Fang, Jasmeet Bagga, Alex Eckert, Andrei Alexandrescu, Kevin Lahey, Petr Lapukhov, Guohui Wang, Omar Baldonado, Manikandan Somasundaram, Dave Gadling, Cooper Lees, Yuval Bachar, Jimmy Williams, David Swafford, Bryan Wann, Ori Bernstein.

TAGS: [INFRA](#) [OPEN COMPUTE](#) [OPEN SOURCE](#)

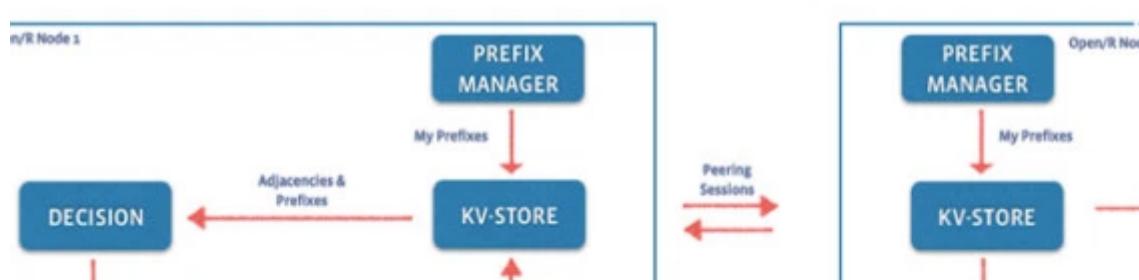
[Like](#) [Share](#) One person likes this. Be the first of your friends.

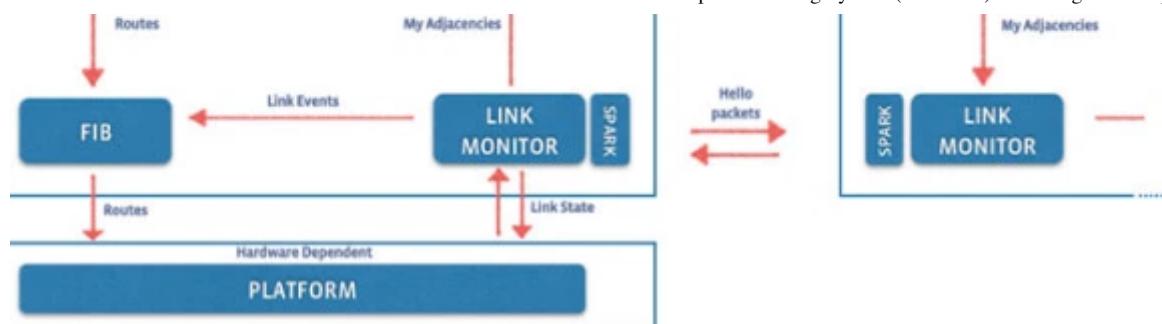
## Related Posts



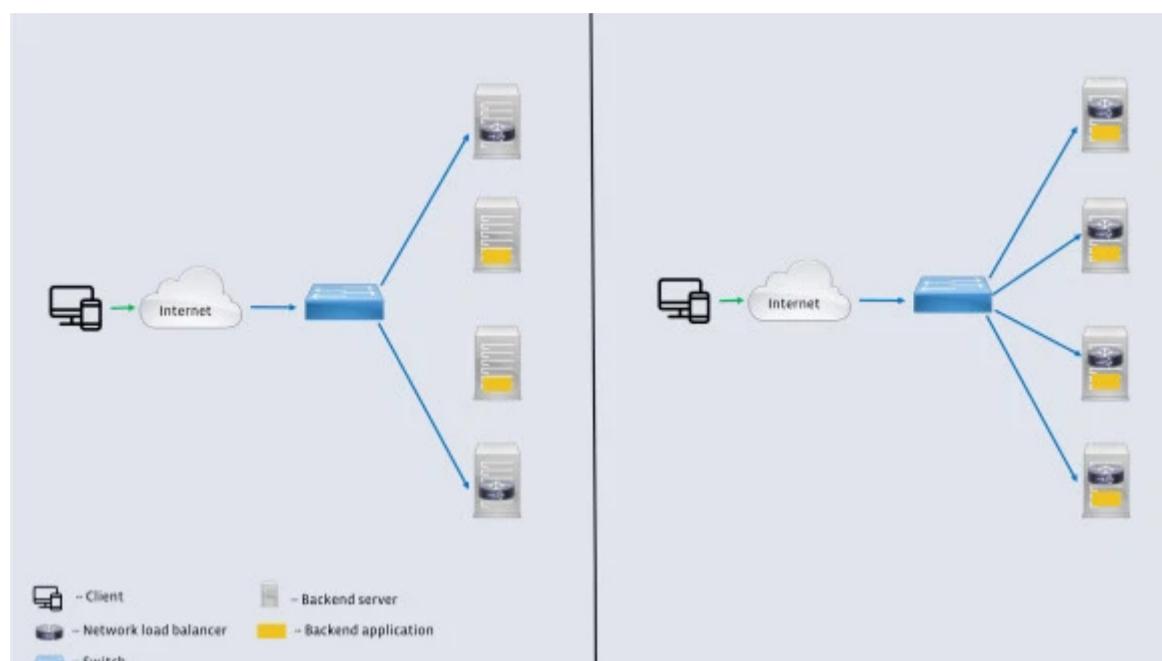
[Mar 14, 2019](#)

[Reinventing Facebook's data center network](#)





Nov 15, 2017

[Open/R: Open routing for modern networks](#)

May 22, 2018

[Open-sourcing Katran, a scalable network load balancer](#)

## Related Positions

[Data Center Systems Thermal Engineer](#)[MENLO PARK, US](#)[Workforce Planning Program Manager, Mission Control](#)[MENLO PARK, US](#)[Architect, Data Center Design](#)[DUBLIN, IRELAND](#)[Enterprise Support Tech](#)[DUBLIN, IRELAND](#)[Data Center Facilities Engineering, Electrical Engineer](#)[WASHINGTON, US](#)[See All Jobs](#)

## Join Our Engineering Community

### Available Positions

[Data Center Systems Thermal Engineer](#)[MENLO PARK, US](#)[Workforce Planning Program Manager, Mission](#)[Control](#)[MENLO PARK, US](#)

### Stay Connected



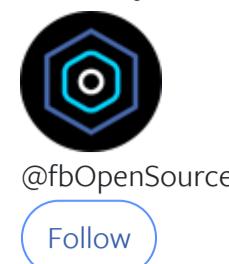
Facebook Engineering

Like

### Open Source

Facebook believes in building community through open source technology. Explore our latest projects in Artificial Intelligence, Data Infrastructure, Development Tools, Front End, Languages, Platforms, Security, Virtual Reality, and more.

[Architect, Data Center Design](#)  
[DUBLIN, IRELAND](#)  
[Enterprise Support Tech](#)  
[DUBLIN, IRELAND](#)  
[Data Center Facilities Engineering, Electrical Engineer](#)  
[WASHINGTON, US](#)



@fbOpenSource

[Follow](#)

ANDROID



iOS



WEB



BACKEND

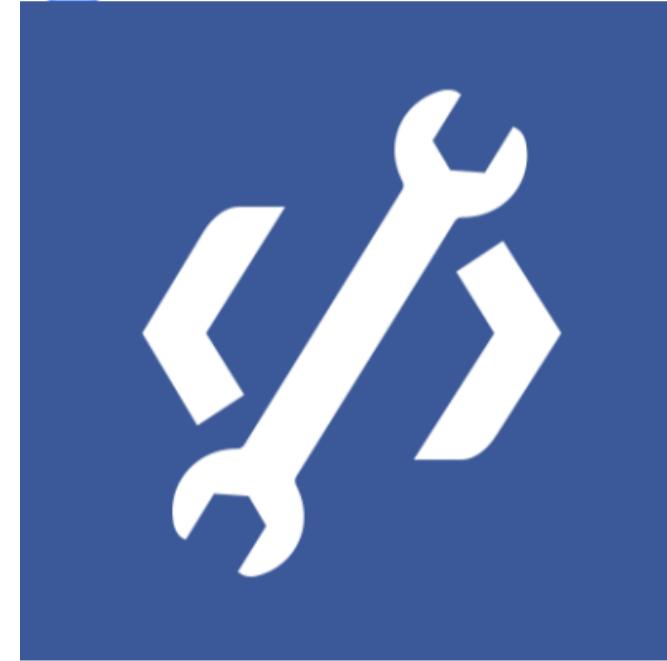


HARDWARE

[See All Jobs](#)

# facebook research

Facebook Research

[Like](#)[Learn More](#)

Facebook Developers

[Like](#)

RSS

[Subscribe](#)