

Community: Force Multiplier for Network Programmability



Lori MacVittie, 2012-31-10

#SDN Programmability on its own is not enough, a strong community is required to drive the future of network capabilities

One of the lesser mentioned benefits of an OpenFlow SDN as articulated by ONF is the ability to "customize the network".

It promotes rapid service introduction through customization, because network operators can implement the features they want in software they control, rather than having to wait for a vendor to put it in plan in their proprietary products.

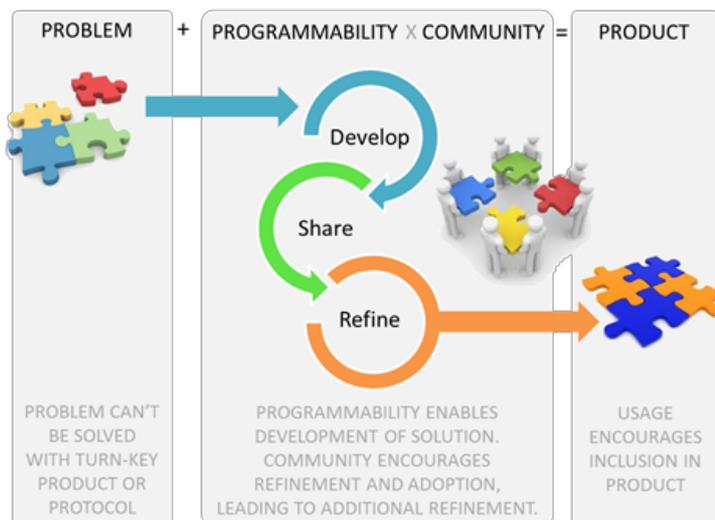
-- Key Benefits of OpenFlow-Based SDN

This ability is not peculiar to SDN or OpenFlow, but rather it's tied to the concept of a programmable, centralized control model architecture. It's an extension of the decoupling of control and data planes as doing so affords an opportunity to insert a programmable layer or framework at a single, strategic point of control in the network. It's ostensibly going to be transparent and non-disruptive to the network because any extension of functionality will be deployed in a single location in the network rather than on every network element in the data center.

This is actually a much more powerful benefit than it is often given credit for. The ability to manipulate data in-flight is the foundation for a variety of capabilities – from security to acceleration to load distribution, being able to direct flows in real-time has become for many organizations a critical capability in enabling the dynamism required to implement modern solutions including [cloud computing](#). This is very true at layers 4-7, where SDN provides the extensibility of functionality for application-layer flows, and it will be true at layers 2-3 where SDN will ostensibly provide the same for network-layer flows.

One of the keys to success in real-time flow manipulation, a.k.a network programmability, will be a robust community supporting the controller. Community is vital to such efforts because it provides organizations with broader access to experts across various domains as well as of the controller's programmatic environment. Community experts will be vital to assisting in optimization, troubleshooting, and even development of the customized solutions for a given controller.

THE PATH to PRODUCTIZATION



What ONF does not go on to say about this particular benefit is that eventually customizations end up incorporated into the controller as native functionality.

That's important, because no matter how you measure it, software-defined flow manipulation will never achieve the same level of performance as the same manipulations implemented in hardware. And while many organizations can accept a few milliseconds of latency, others cannot or will not.

Also true is that some customized functionality eventually becomes so broadly adopted that it requires a more turn-key solution; one that does not

require the installation of additional code to enable.

This was the case, for example, with session persistence – the capability of an ADC (application delivery controller) to ensure session affinity with a specific server. Such a capability is considered core to [load balancing](#) services and is required for a variety of applications, including VDI. Originally, this capability was provided via real-time flow manipulation. It was code that extended the functionality of the ADC that had to be implemented individually by every organization that needed it – which was most of them. The code providing this functionality was shared and refined over and over by the community and eventually became so demanded that it was rolled into the ADC as a native capability. This improved performance, of course, but it also offered a turn-key "checkbox" configuration for something that had previously required code to be downloaded and "installed" on the controller.

The same path will need to be available for SDN as has been afforded for ADN, to mitigate complexity of deployment as well as address potential performance implications coming from the implementation of network-functionality in software. That path will be a powerful one, if it is leveraged correctly. While organizations always maintain the ability to extend network services through programmability, if community support exists to assist in refinement and optimization and, ultimately, a path to productization the agility of network services increases ten or hundred fold over the traditional vendor-driven model.

There are four requirements to enable such a model to be successful for both customer and vendors alike:

1. **Community** that encourages sharing and refinement of "applications"
2. **Repository** of "applications" that is integrated with the controller and enables simple deployment of "applications". Such a repository may require oversight to certify or verify applications as being non-malicious or error-free.
3. A means by which applications can be rated by consumers. This is the **feedback mechanism** through which the market indicates to vendors which features and functionality are in high-demand and would be valuable implemented as native capabilities.
4. A basic level of **configuration management** control that enables roll-back of "applications" on the controller. This affords protection against introduction of applications with errors or that interact poorly when deployed in a given environment.

The programmability of the network, like programmability of the application delivery network, is a powerful capability for customers and vendors alike. Supporting a robust, active community of administrators and operators who develop, share, and refine "control-plane applications" that manipulate flows in real-time to provide additional value and functionality when it's needed is critical to the success of such a model.

Building and supporting such a community should be a top priority, and integrating it into the product development cycle should be right behind it.

- [HTML5 WebSockets Illustrates Need for Programmability in the Network](#)
 - [Midokura – The SDN with a Hive Mind](#)
 - [Reactive, Proactive, Predictive: SDN Models](#)
 - [SDN is Network Control. ADN is Application Control.](#)
 - [F5 Friday: Programmability and Infrastructure as Code](#)
 - [Integration Topologies and SDN](#)
-

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com