

**OpenFunction: An Extensible Data Plane Abstraction Protocol for Platform-independent Software-Defined Middleboxes**

**Abstract:**

We propose OpenFunction, an extensible data plane abstraction protocol for platform-independent software-defined middleboxes. The main challenge is how to abstract packet operations, flow states and event generations with elements. The key decision of OpenFunction is: actions/states/events operations should be defined in a uniform pattern and independent from each other. We implemented a working SDM system including one OpenFunction controller and OpenFunction boxes based on Netmap, DPDK and FPGA to verify OpenFunction abstraction.

**Existing System:**

A dataplane abstraction for middleboxes is needed to realize the vision of software-define middleboxes (SDM). SDMs provide network operator the ability to dynamically load/unload various network functions without changing the network hardware, similar to what OpenFlow/P4 provides for the switches. For software-define middleboxes (SDMs), a data plane abstraction should be both *platform independent* and *fully extensible*. Platform independence decouples the data plane function semantics and the underlying hardware that realizes the network function. This allows third-party SDM program’s data plane to execute at any SDM boxes (*i.e.*, SDM compliant middleboxes) with same semantics but different performance depending on hardware adequacy. Fully extensible means that any new middlebox functionality can be defined by an SDM program abstraction, which is critical to enable innovation for middleboxes. For example, this enables design of a new packet encryption algorithm for VPN, define a new flow state in the data plane for firewall, or subscribe to an event when a specific condition is trigged for IPS.

**Proposed System:**

We propose OpenFunction, an extensible and platform independent data plane abstraction protocol for software-defined middleboxes. OpenFunction architecture consists of a logically centralized *OpenFunction controller* and a number of *OpenFunction boxes* distributed around a network, where every box implements OpenFunction abstraction layer, elements to the control plane using the modular style. An SDM programmer needs not to be aware of the underlying hardware features of SDM boxes: just define the behaviour of data plane as a data flow graph of processing elements, and focus on the application logic of control plane. Under the hood, OpenFunction defined data plane can be realized by a platform dependent implementation that fully leverages the hardware features of the underlying SDM box. To support new operations, OpenFunction provides a platform-independent pseudo language for specifying customized elements beyond those predefined ones; such a platform-independent pseudo program can be compiled to a platform-dependent element by the underlying box.