Simple Unified Control for Packet and Circuit Networks

Saurav Das, Guru Parulkar and Nick McKeown
350 Serra Mall, Packard Bldg, Stanford University

Abstract: OpenFlow is proposed as an architectural platform and a unified control plane for packet and circuit networks, with the main goal of simplifying network control and management while fostering innovative change in them.

Introduction

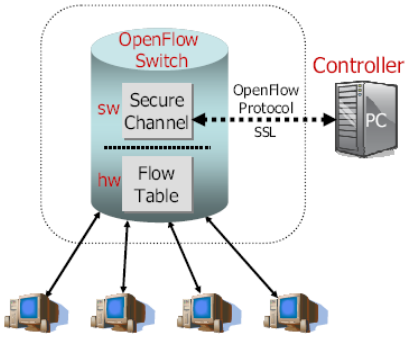
There is a well recognized view in the academic and commercial networking communities that networking infrastructures have essentially ossified. IP networks today remain best-effort in performance and are largely unpredictable in their behavior, much as they were 40 years ago. Furthermore, they were never designed to support what they are quickly attempting to morph into – a *unified* communications infrastructure.

At the same time, researchers have had many great ideas that have not been deployed, as it is extremely hard to test new ideas at scale - in part a result of the massive installed base of equipment and protocols, and the fact that in current networks, routers, packet switches and transport network elements are closed and run proprietary OS's. Moreover the rigid nature of the platform itself adapts poorly when disturbed from established norms. This singular fact discourages network operators from adapting/experimenting with the installed base, which in essence handcuffs the owners/operators of the network to the limitations of the platform. Furthermore the platform takes the power of controlling the network *away* from the stakeholders in the network and establishes it instead in the vendor equipment that supports the platform.

OpenFlow is a recently proposed network architectural platform that seeks to redress network ossification [1]. Its premise is simple – can we create a way for networking researchers to test out their ideas, at scale, that include the control and management of networks with both packet and circuit switches. We believe that OpenFlow allows us to do this.

OpenFlow

The basic idea is simple: *OpenFlow is based on a packet switch, with an internal flow table and a standardized interface to add/edit/remove flow table entries* (Fig. 1). It exploits the fact that most modern switching equipment contains tables that run at line-rate and map incoming traffic to outgoing ports. While each vendor's equipment is different, OpenFlow exploits a common set of functions implemented in all switches. Furthermore, the open standardized interface (the OpenFlow protocol) takes the control of the switches out-of-the-box and places it in the Controller, which is under the network operator's control.



In Port	VLAN ID	Ethernet			IP			TCP	
		SA	DA	Type	SA	DA	Proto	Src	Dst

Fig. 1: OpenFlow Switch & Flow-Table Entry Header

OpenFlow is completely backward compatible. The datapath of an OpenFlow switch consists of a flow-table, with entries that specify in the header what exactly constitutes a flow, plus associated actions to be performed on incoming packets that match such a flow table entry. The OpenFlow switch can behave as an IP router, Ethernet switch or a standalone application layer firewall. Alternatively, *OpenFlow allows switching at different network layers to be combined*. A flow can be defined as a combination of any of the 10 fields that make up a header [2]. While each packet is switched individually, all packets in a flow are switched the same way, making the *flow the fundamental unit of manipulation within the switch.* For example, a flow could be defined as a combination of Ethernet and IP src/dest addresses, or a flow could be defined as a combination of an incoming port and the TCP/UDP src/dest ports, so that traffic for a certain application could be routed differently from the rest of the traffic. In short, OpenFlow allows granularity and layering to be at the discretion of the owner/operator and suited to the user/application, and not be limited by the platform or be enforced by the protocols.

Layer 1 Switching

In this paper, we extend the flow abstraction to include Layer 1 switching technologies such as time-slot, lambda and fiber switching. We exploit the fact that commercial transport network equipment also maintain flow tables in the form of cross-connect tables with entries that are suited to the switching type of the switch. A super-set of such an entry is shown in Fig. 2. Multiple flow tables can be maintained in switches that can switch at different granularities – for example, switches that have both packet and circuit interfaces can incorporate both flow tables shown in Fig. 1 and 2, one for the packet side and the other for the circuit side along with a way to specify the adaptation between them.

In/Out Port	In/ Out Lambda	VCG	Starting Time-Slot	Signal Type

Fig. 2: OpenFlow Circuit Switch Flow-Table Entry

When we use the abstraction of a flow in a packet switched network, we effectively *blur the distinction between packets and circuits and regard them both simply as flows in a flow switched network.* This in turn readily presents us with an opportunity to establish a simple, automated unified control plane for controlling and managing flows, across the domain of the Controller, through OpenFlow switches that switch at different granularities – packet, time-slot, lambda and fiber (Fig. 3). Conceptually it is a simple extension from mapping a flow to an output port/queue to mapping a flow to an outgoing circuit.

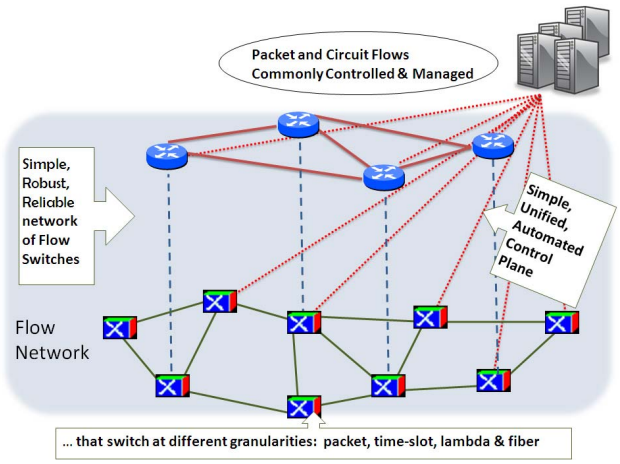


Fig. 3: OpenFlow Network

OpenFlow advocates a separation of the data and control plane for the flow network. By extracting the intelligence from the switches and housing it in the controller, OpenFlow simplifies the data plane switches compared to existing switching platforms, which have become bloated and complex, with millions of lines of code, multiple layers of protocols, and massive, power hungry hardware. Simplifying the data plane brings down cost and complexity while improving reliability & availability. Furthermore bringing packet and circuit switches under a unified, automated control platform allows the data plane to accommodate and benefit from different switching technologies. At the same time the control plane is simple, robust and efficient, without layers and layers of protocols, and yet allows for greater control, higher determinism, and simpler management. Service providers can benefit from not having to duplicate functionality and resources in multiple layers. More significantly they need not bear the burden and complexity of owning and managing two separate networks or hire separate operators experienced in different management tools such as SNMP for IP networks and TL-1/CORBA for transport networks.

OpenFlow allows innovations in control and management of networks. Service providers can experiment with many alternative control paradigms – eg. routing, network recovery, traffic engineering, congestion control – without being subject to the limitations of the platform (unlike the case today). Multiple such paradigms can even co-exist under the network operator's discretion.

The research community can benefit from being able to try out new ideas at scale by experimenting in their 'slice' of the network. Such a slice can be enabled by flow level virtualization of the switches and other resources in the network. Such partitioning/virtualization can be enforced in the Controller, much like an operating system in a computer partitions between user space and kernel space. If successful, new ideas can be tested exhaustively in one slice of the network before being deployed in the production slice, leading to a transfer of technology from research to industry (which barely exists today).

Summary

OpenFlow was presented as a simple, unified control plane and architecture for packet and circuit switches. We believe that both IP and Transport networks will benefit and evolve from such a construction while becoming indistinguishable in the way that they are managed.

References

[1] N. McKeown, et. al., "OpenFlow: Enabling Innovation in Campus Networks", SIGCOMM CCR, Vol. 38, Issue 2, March 2008

[2] OpenFlow Switch Specification, http://www.openflowswitch.org/