OSF – Open Service Framework

An integrated high-speed load balancing and flow steering framework

Anand Gorti
IBM
3039 Cornwallis Rd
Research Triangle Park, NC 27709
+1-919-254-8820

Vijoy Pandey
Blade Network Technologies
2350 Mission College Blvd, #600
Santa Clara, CA 95054
+1-408-931-3803

Abstract

This paper describes Open Service Framework, a framework for enabling policy based flow steering and service load balancing capabilities in Layer 2/3/4 Ethernet switches. These software enhancements enable an Ethernet switch to steer packets based on user specified rules that consist of fields from the packet header and the ingress port. Additionally, the switch can load balance traffic across multiple services – hosted directly on physical servers, or as virtual machines (virtual appliances) on bare metal servers – at line speed, detect server or application failures and either remap the affected flows on the remaining servers or forward them to a backup server for high availability. In a virtualized environment, the switch also keeps track of the steering and load balancing policies across virtual appliance live migrations and disaster recovery migrations. These mechanisms have been implemented on a commercial 10GbE switch and are being deployed initially for blade chassis based multi-vendor security solutions for protecting enterprise and hosted Data Centers.

1. Introduction

Data Centers are evolving into virtualized environments with higher server densities and increased resource utilization. The bandwidth requirements are also escalating with new e-commerce, multi-player gaming and video streaming services being offered. In addition, storage and data networks are converging onto a common Ethernet fabric thus driving the demand for 10GbE switches with low latency and guaranteed bandwidth requirements. The Data Center design is trending towards combining some of the chassis, edge, aggregation and core networking layer functionalities. At the same time security continues to be a challenge with more sophisticated attacks on the critical infrastructure and end devices.

The current practice of adding servers, single function appliances or connection based load-balancers, does not scale in terms of throughput (upwards of 40Gbps) and manageability. The Data Center requirements are driving the need for integrating more and more functions like load balancers, firewall and intrusion detection systems into the Ethernet switches. However, administrators also need the flexibility to identify and control traffic flows for service differentiation creating a need for virtualized switching capability right up to the Virtual Machine (VM) where the applications are running. This increased integration along with the need for flexible service differentiation motivated us to develop

- a scalable stateless load balancer based on existing layer-2 trunk and layer-3 ECMP mechanisms
- a policy based packet forwarder that uses existing Access Control List mechanisms
- enhanced health check mechanisms to detect and recover from link, server and application failures
- a framework that works transparently across services running on bare metal servers as well as services running as virtual appliances

Initially, we will focus ourselves on the bare metal server incarnation of this software for readability.

Applicability to the virtual appliance domain is explored in Section 3, along with the modifications needed to achieve this. Additionally, throughout this paper we use a security services deployment as the illustrative example, though the mechanisms described are more generally applicable.

2. Functional overview

The basic idea of the Flow Control Ethernet switch is to allow the administrator to define policies that control the packet forwarding path in support of OSF. A sample policy consists of:

- a rule as defined by the fields in the packet header shown in Table 1
- an associated action of
  - pass or drop
  - redirect to port/trunk X

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

Table 1: Packet header fields matched by a rule

Typical 10GbE Data Center switches support thousands of rules depending on the TCAM capacity. A functional view of the flow control switch and its external interface are shown below in Figure 1.

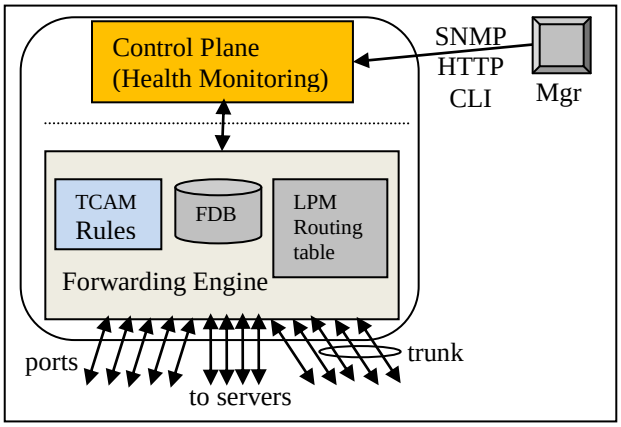

Figure 1: Flow Control switch functional view

Once the policies are configured in the switch, incoming packets are matched against the rules and the associated action is taken by the forwarding engine. By setting the action to *redirect*, the packet on a specific outgoing port or trunk, the forwarding engine can achieve traffic steering and load balancing of flows. Optionally, the packet headers may be modified before forwarding the packet. If there is no rule match then the default action of *pass* causes forwarding of the packet based on layer 2/3 switching.

The flow control switch also implements various high availability (HA) mechanisms with enhanced health-checking in addition to the normal link failure detection. Periodically, the switch control plane sends out the configured health-check packets to the servers for detecting failure. When a failure is detected, the switch:

- remaps the affected flows by spreading them across the operating ports within a trunk when there is no backup server defined, or,
- replaces the failed server with a configured backup server or bypass link.

3. Flow Control Switch Application Scenarios

The flow control switch may be used for various applications that require policy based traffic steering and stateless load balancing. It is ideally suited for appliance consolidation in a network with flexible administrative flow controls. A few sample application scenarios are described below.

3.1. Policy based packet steering

In this scenario, all traffic (irrespective of the VLAN tag which could be used to identify customers for firewall policies rather than port membership) from the external router needs to get inspected by the firewall appliances before being sent into the internal VLANs. In current implementations, this is achieved by using separate load balancers and layer-2 switches (these could be combined in the same switch chassis via different line cards). However, this approach poses scalability, reliability and manageability issues.

The flow control switch combines the load balancing and switching functions without any performance impact. Table 2 shows the policies required to handle the packet routing for the above use case.

Ingress Port	VLAN ID	Eth	IP	TCP	Action

EXT	*	*	*	*	Redirect to TRK1
INT	*	*	*	*	Redirect to TRK1

Table 2: Packet steering sample rules

- Traffic coming from the external router matches the first rule, so the action redirects it towards the internal trunk (TRK1) that connects the firewalls. The trunk ports are configurable to use various hash based load balancing techniques using layer 2-4 header fields (e.g. Src IP XOR Dest IP)
- The firewall modifies packet headers so that traffic flows via default layer-2 forwarding.
- Traffic flowing from the internal VLANs towards the external network matches rule for INT ports and gets redirected to the firewall via TRK1.

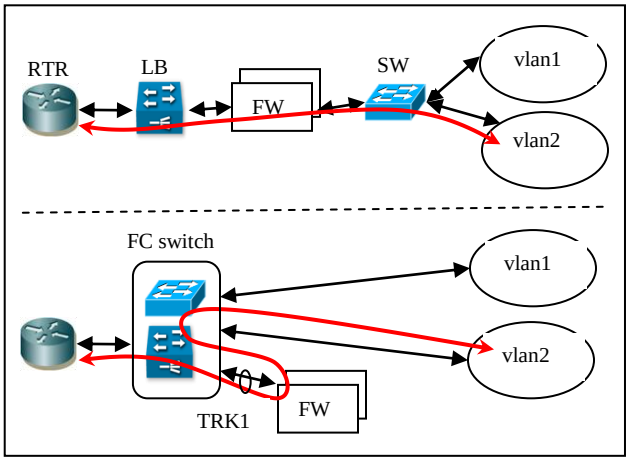

Figure 2: Packet steering example

Another use case with DMZ servers and internal network protected by the firewalls is illustrated in Figure 3. Some designs protect the DMZ servers by just using router filter rules but for critical applications a logical three homed firewall is recommended.

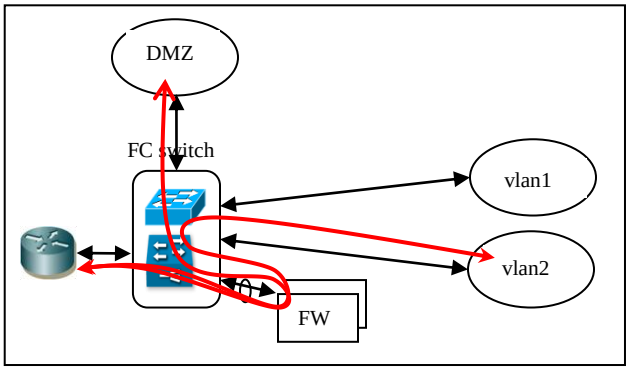

Figure 3: Protecting DMZ servers in a Data Center

With evolving sophisticated attacks emanating from the network, it may be necessary to include multiple best-of-breed inline security measures. One such example with an Intrusion Prevention System (IPS) in-line with the firewalls is shown in Figure 4. Figure 4

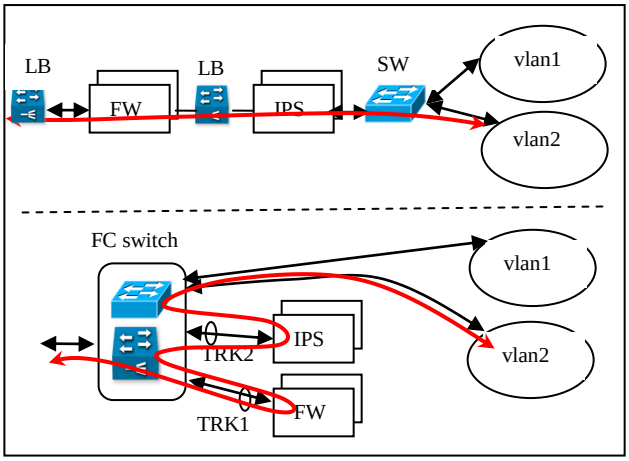

Figure 4: Packet steering with multiple inline services

The policies required to achieve packet steering through a sequence of applications is shown in Table 3.

Ingress Port	vlan	Eth	IP	TCP	Action
EXT	*	*	*	*	Redirect to TRK1
TRK1	*	SMAC RTR	*	*	Redirect to TRK2
TRK2	*	SMAC FW	*	*	Default layer-2
INT	*	*	*	*	Redirect to TRK2
TRK2	*	*	*	*	Redirect to TRK1

Table 3: Policies for multiple inline services

Note that the rules in the policy table are ordered so the rule with the best match decides the forwarding action.

3.2. Scalable load balancing and high-availability

This section highlights the Flow Control switch capabilities for achieving scalable high-performance load balancing and high availability that are critical to meet modern Data Center requirements. Traditional "state" based load balancers maintain per micro-flow session tables. For throughput in excess of 40Gbps, the load balancer needs to handle a high connection rate

(million/second) and large total number of connections (tens of millions). Even though these load balancers are capable of intelligently distributing the load based on various usage based criteria, it very expensive, difficult to scale and provide fault tolerance with current network processing technology. To overcome these limitations, our approach uses a "stateless" statistical load balancing alternative that already exists in current layer 2/3 Ethernet switches.

Typically Layer 2/3 managed switches support some form of EtherChannel or IEEE 802.3ad Link Aggregation with up to a maximum of 8 links in a trunk group. Traffic is load balanced across all the active ports using a hash algorithm. The hash function typically uses one or more packet header fields like Source MAC, Destination MAC, Source IP, Destination IP, Source TCP/UDP port and Destination TCP/UDP port. Should a link fail, the switch automatically redistributes traffic across the remaining links achieving fault tolerance.

The Flow Control switch incorporates several enhancements to the traditional EtherChannel as follows:

- Additional layer 2-7 health-check mechanisms that include link, IPS, ARP, Ping, TCP, HTTP, SMTP, SSL, stateful health checks, user-level scripting etc.
- load balancing to more than the allowed EtherChannel limit of the switch
- N+1 server redundancy model to detect and replace a failed server with a hot-standby
- configurable option to either remap all the flows in a server load balancing group (for better distribution) or re-distribute only the affected flows (for persistency) when a server fails
- the ability to define dependency actions between a set of monitored ports and a set of control ports which is useful for upstream re-routing or bypassing critical failure [3] as shown in Figure 5.

The solution is also very scalable at the blade chassis level as well as the rack level. A set of Flow Control switches can be aggregated into a rack-level (or larger) virtual switch, thereby creating a rack-level OSF with multiple levels of redundancy built into the solution.

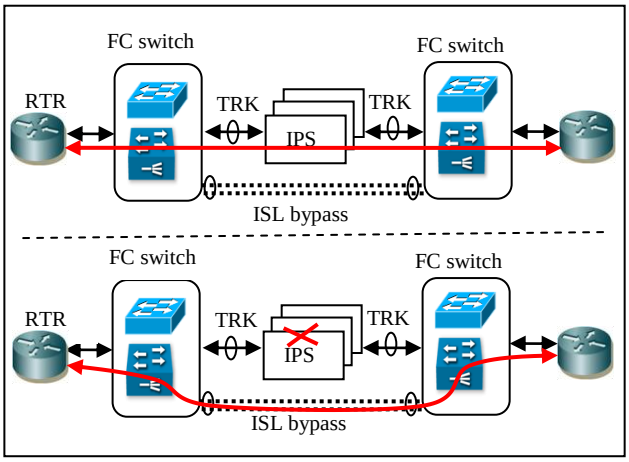

Figure 5: Bypass a critical failure

3.3. Virtualized Environments

.

With the proliferation of server virtualization technologies in the Data Center, there has been an increase in the number of pre-created virtual appliances that can perform various tasks on bare metal servers, ranging from security to databases. The OSF software, when combined with various virtualization-aware switching technologies, will create a powerful framework for load balancing and flow steering through a set of virtual appliance-based services.

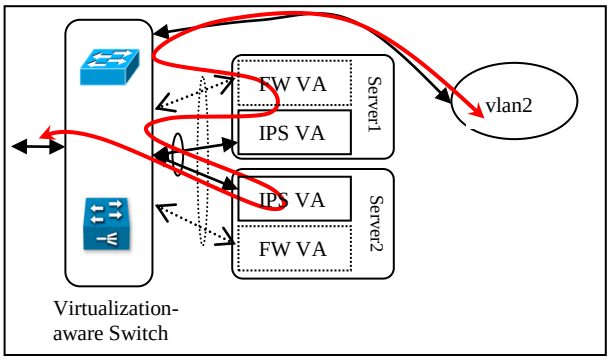

Figure 6: Sequencing and load balancing Virtual Appliances with a Virtualization-aware Switch

Figure 6 shows a set of firewall and IPS virtual appliances (VAs) distributed across a set of bare metal servers for high-availability. To be able to load balance and steer traffic flows across these sets of virtual services, the switch has to be server virtualization aware [5] and be able to bridge across virtual ports as defined by the virtual Ethernet interfaces of the VAs. Since the switch is virtual port aware, it can transparently steer traffic between the FW VA on

server 1 and the IPS VA on server 2. This sequence is also honored if and when a VA is migrated to another server due to disaster recovery policies.

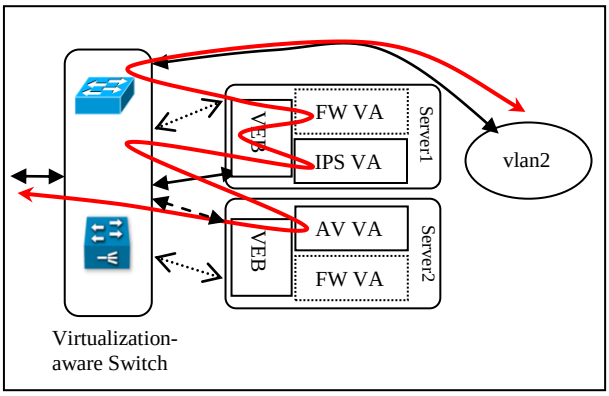

Figure 7: Sequencing and load balancing Virtual Appliances with a Virtualization-aware Switch and a Server-resident Virtual Ethernet Bridge

Figure 7 shows a set of firewall, IPS and Anti-Virus (AV) VAs deployed on two servers. An efficient way of handling flow steering between VAs on the same server is by installing a Virtual Ethernet Bridge (VEB) on the server that bridges between the virtual appliances on that server. VEBs can be software-based [7] or can be part of the physical network interface on the server. The OSF Manager will install rules at configuration time in a way such that VEBs are utilized for VA-to-VA steering in the same server, while the external virtualization aware switch is utilized for VA-to-VA steering across servers. The same heuristic is also employed to define load balancing across the set of VA-defined virtual ports.

4. Acknowledgements and Related Work

The OSF software was initially co-developed by IBM and Blade Network Technologies over early-2007 to mid-2008. In its first incarnations, the software enabled an IBM BladeCenter H/HT (High-speed) chassis equipped with a Blade Network Technologies embedded 1:10Gb Ethernet Switch Module to perform as a high performance integrated security appliance. Architects who contributed significantly to this effort were John Lloyd (IBM), Tim Chao (BNT), Cynthia Gabriel (BNT) and Bill Shao (BNT). OSF as a software load is currently shipping on Blade Network Technologies switches, and is now being actively enhanced for the Virtual Appliance use cases.

The use of health monitoring of services to feed into a load balancing construct is commonly used by network vendors. The construct utilized by the OSF platform was first developed and deployed by Alteon WebSystems [4] in their load balancers.

While OSF was being developed, there was a parallel effort, OpenFlow [1], for the definition of flow based switching as a platform for developing and testing new protocols. While both parallel efforts defined a flow switching framework with similar inner-workings, their goals are orthogonal. OpenFlow only concentrated on traffic sequencing itself, while OSF concentrated on load balancing application groups in addition to traffic sequencing. OpenFlow has defined a special OpenFlow protocol to configure traffic sequencing while OSF is configured and managed from the network switch itself, exposing all the well-known mechanisms for management such as SNMP, NETCONF and CLI. Additionally, while OpenFlow concentrates on traffic sequencing between physical ports, OSF sequences and load balances across physical as well as virtual ports since applications could be running on bare metal or as a virtual appliance.

5. Future Directions

While this paper concentrates on the use of a VEB within the server for traffic sequencing and load balancing across virtual appliances, other technologies such as Virtual Ethernet Port Aggregators (VEPA) and Port Extenders [8, 9] can also perform this function, and these are currently being explored by the authors to provide the best possible deployment choice to customers.

The current solution also has a limitation on how effectively the traffic can be spread across the load balancing groups. This limitation stems from the way hashing functions are implemented in TCAM on commercially available merchant switching silicon, as fewer hash buckets (h) map the traffic to the load balancing group of size (g). In other words, $h \approx g$. Next generation silicon will address this issue by decoupling the hashing buckets from the size of the load balancing group, whereby $h >> g$, providing greater granularity for even traffic spreading. A workaround that is being proposed in the meantime is to implement a multi-tier hashing scheme, whereby traffic is first spread to a first-tier pseudo load balancing group, which in turn each comprises of multiple application servers or virtual appliances.

6. Conclusion

This paper outlines the Open Service Framework – a framework built upon Layer 2/3/4 Ethernet Switches to enable policy based flow steering and service load balancing capabilities across sets of services either hosted directly on servers, or hosted as virtual appliances on a virtualized bare metal server.

We describe the implementation of this framework in a physical server world using security services as the illustrative example. The framework is then extended to a virtualized server environment where the switch is capable of switching between virtual ports defined by each virtual appliance.

7. References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, *"OpenFlow: Enabling Innovation in Campus Networks"*, Mar 14, 2008. http://www.openflowswitch.org/documents/openflow-wp-latest.pdf

[2] S. Lambert, K. Wheeler, C. Freeland, *"SPANIDS: A Scalable Network Intrusion Detection Loadbalancer"*, Proceedings of Computing Frontiers, ACM CS Press New York, N.Y., 2005, pp. 315-322. http://www.cse.nd.edu/~spanids/papers/spanids_design.pdf

[3] D. Katz, D. Ward, *"Bidirectional Forwarding Detection"* draft-ietf-bfd-base-09.txt, Feb 5, 2009

[4] Alteon WebSystems, Wikipedia entry, http://en.wikipedia.org/wiki/Alteon_WebSystems

[5] Blade Network Technologies, *"VMready Server Virtualization-Aware Switch"*, http://www.bladenetwork.net/smartconnect.html

[6] M. Ko, R. Recio, *"Virtual Ethernet Bridging"*, http://www.ieee802.org/1/files/public/docs2008/new-dcb-ko-**VEB**-0708.pdf

[7] Cisco Systems, *"Cisco Nexus 1000v Series Switches"*, http://www.cisco.com/en/US/products/ps9902/

[8] Paul Congdon, Chuck Hudson, "*Modularization of Edge Virtual Bridging – proposal to move Bridging proposal to move forward"*, *http://www.ieee802.org/1/files/public/docs2009/new-evb-congdon-vepa-modular-0709-v01.pdf*

[9] Joe Pelissier, *"Network Interface Virtualization Review"*, http://www.ieee802.org/1/files/public/docs2009/new-dcb-pelissier-NIV-Review-0109.pdf