Independent Scalability and Functionality Test: Sandvine Virtualized Traffic Steering Engine (TSE) and Virtualized Policy Traffic Switch (PTS)

Introduction

Network Functions Virtualization (NFV) has kicked off a revolution of service provider networks and services. During the initial wave of virtualization, application use cases such as traffic classification were often virtualized with a per-box, appliance-type approach. Meanwhile, service provider focus is shifting towards manageable, resilient, and versatile solutions that scale to Millions of subscribers and multiple data center sites.

Traffic classification engines use Deep Packet Inspection (DPI) and other statistical analysis methods to classify traffic. DPI is a network traffic-heavy application; naturally all customer traffic needs to pass through the DPI function. NFV provides the opportunity to scale arbitrarily by simply adding Virtual Network Function (VNF) instances as needed. One major question is how to distribute and load balance customer traffic across multiple DPI instances. Traditionally, external load balancers have conducted this job; in a virtualized world, it is only logical to virtualize the load balancers as well. In the virtualized data center, traffic streams transit through application-layer services optimized by Service Function Chaining (SFC). In Sandvine's case, the load balancer and the packet inspection functions are stitched together in an SFC.

There are multiple approaches to chaining virtual network functions. Sandvine uses the Network Service Header (IETF Draft: draft-ietf-sfc-nsh) on the data plane to steer traffic via a chain of services.

In this test campaign, we put multiple use cases of Sandvine's SFC implementation under scrutiny. These use cases were designed to verify the functionality and scalability of the solution under test, combined.

Sandvine has recently introduced its virtualized Traffic Steering Engine (TSE) as a means to load balance and chain data plane traffic. The TSE was part of the tested solution. To demonstrate a realistic deployment scenario, multiple instances of Sandvine's Policy Traffic Switch (PTS) Virtual Series were chained. TSE was configured to forward traffic towards PTS instances via a dedicated layer 2 network, referred to by Sandvine as the "service network".

Test Highlights

- **Verified scale-out of PTS Virtual Series instances in a TSE-enabled environment**
- **Successfully created two service function chains between PTS Virtual Series instances and verified the traffic distribution mechanism between both chains**
- **Measured 361 milliseconds maximum PTS failover time and hitless recovery**
- **PTS Virtual Series detected and mitigated DNS amplification DDoS[a] attacks without affecting safe traffic**
- **PTS Virtual Series successfully detected and blocked 1.2 Mpps[a] of SYN flood DDoS[b] attacks without affecting legitimate traffic**

a) Million packets/second
b) Distributed Denial of Service

Most of the scenarios we examined were focused purely on service function chaining. Additionally, we verified two Sandvine Network Protection features on the PTS. The addition of security features illustrated some of the functions that service providers can activate on PTS Virtual Series instances.

We conducted this test at EANTC's lab in Berlin, Germany, in September and October 2016, using pre-production software supplied by Sandvine. The tests were commissioned by Intel as part of the Intel Network Builders program.

Test Bed Configuration

The Sandvine TSE and PTS software was hosted on two HPE ProLiant DL380 Gen9 servers. Each of the two servers was equipped with four 10GbE network interfaces for data plane traffic. Two 40GbE ports on each server interconnected the cluster network.

We used Ixia's IxLoad Virtual Edition (VE) and BreakingPoint Virtual Edition to generate stateful test traffic, emulate two attack scenarios and measure failover and recovery times. We hosted all the virtual tester components on two HPE ProLiant DL380 Gen9 Servers, namely the IxLoad Virtual Chassis and Virtual Load Modules, BreakingPoint Virtual Controller and Virtual Blades. Each server had six 10GbE ports. 16 Virtual Load Modules shared four 10GbE ports on each server equally using SR-IOV technology (four SR-IOV virtual functions on each 10GbE port). In this configuration, we conducted a back-to-back physical port reference test with IxLoad VE resulting in 80 Gbit/s Ethernet physical layer throughput carrying HTTP traffic. For this reference test, the links were saturated bidirectionally by using HTTP GET and PUT commands. Two BreakingPoint Virtual Blades utilized the remaining two 10GbE on both servers (a single SR-IOV virtual function on each).

The environment simulated an edge-based setup with 2,000 host IP addresses. 80 % of the emulated hosts used IPv4 addresses while the remaining 20 % were equipped with IPv6 addresses. The test traffic was bi-directional with half of the emulated hosts residing on either side of the system under test. The default traffic load was 16 Gbit/s at layer 2 (8 Mbit/s per emulated network host). The average packet size in the traffic mix was 1,000 Bytes resulting in a packet rate of ~2 Mpps. We also took into account the overhead introduced by the Sandvine solution on the service network; 22 Bytes per packet.

Table 2: Hardware and Software Configuration on page 6 lists the details of the different hardware and software elements in the test bed.

TSE: Chaining PTS Instances

Service providers have deployed traffic classification using deep packet inspection for years. As networks have evolved, traffic classification has become a cornerstone to many design frameworks. For example; the mobile services 3GPP framework incorporates DPI in its functional definitions, referring to it as Policy and Charging Enforcement Function (PCEF). IETFRFC 7665 utilizes a traffic classifier in a virtualization domain to implement application-aware service chaining.

Chaining multiple virtual network functions may be used to enable forwarding of all matching traffic streams via two or more services in a specific order. For example, a virtual firewall may be chained to a virtual router. The chain comprises a security service which network operators can provide to their subscribers.

Sandvine's TSE uses SFC classification rules to define traffic forwarding policies. TSE creates a hash value based on a combination of the Source IP (or the first 64 bits of it in the case of IPv6) and the ingress port. The SFC forwarding policies will contain a list of PTS chains that matching traffic needs to be forwarded to. We asked Sandvine to re-mark the type of service (ToS) bits in the test traffic differently on each PTS instance. We captured packets before and after passing through the test bed, providing an indication of which PTS(s) the traffic had traversed through. Two TSE and two PTS instances took part in this test as depicted in Figure 1: Sandvine Test Setup on page 3.

In this test, we verified the service chaining policies listed in the table below:

Chain	Traffic Forwarding Graph
SFC1	TSE-1 — PTS-1 — TSE-1
SFC2	TSE-2 — PTS-1 — PTS-2 — TSE-2

Table 1: Configured Service Chains

We executed this test in three stages: First we sent part of the test traffic towards TSE-1 and verified that traffic passed through PTS-1 only as per service function chain 1 (SFC1). We then sent part of the traffic towards TSE-2 and verified that traffic passed through PTS-1 and PTS-2 (two ToS bits were remarked) as per SFC2. We increased both traffic streams to near 5 Gbit/s each. Combined, the two streams saturated the10GbE link on PTS-1. We verified that streams were forwarded via the correct service chain accordingly.

The main goal of this test case was to examine how the TSE pairs with a PTS cluster and forward traffic to the appropriate PTS instances.

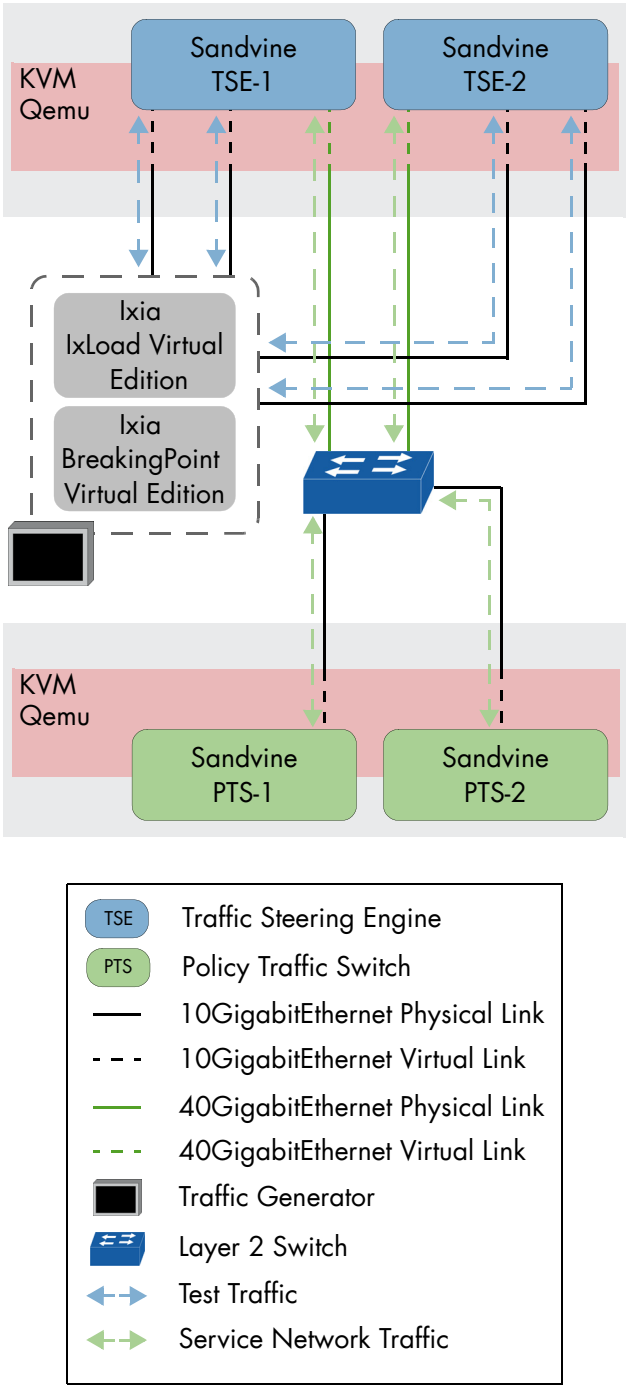


Figure 1: Sandvine Test Setup

TSE: GTP-U Tunneled Traffic Distribution

In mobile networks, user traffic between the service and the packet gateways is encapsulated inside GPRS Tunneling Protocol (GTP) tunnels. It is common that service providers implement value added services (VAS) on the SGi interface – usually referred to as SGi-LAN services. It is also possible that service providers need to apply subscriber specific polices on the GTP tunnels before they are terminated on the packet gateway to utilize signalling, as an example.

In this test, we emulated part of the mobile core and generated ten GTP-U tunnels with unique IPv4 user equipment (UE) addresses inside the tunnel. After registration, each UE generated multiple data transactions with an emulated HTTP server on the PDN side. The TSE was expected to monitor both directions of the traffic and correlate both traffic streams to the emulated UE IPs regardless of the traffic direction.

To verify the traffic distribution, we asked Sandvine's team to configure a unique ToS re-marking policy on each PTS instance. Initially, we checked the traffic behavior when the default traffic forwarding policy was configured; based on the source IP. Since the traffic appeared with a single external source IP, it traversed solely via a single PTS port. We then asked Sandvine to change the forwarding policy to use the internal source IP addresses. We verified that the TSEs forwarded traffic consistently from, and to, each UE towards a specific PTS instance.Traffic was equally distributed on the ports of both PTS instances. The test topology is depicted in Figure 1.

For the purpose of generating quantifiable results, we limited the scope of the test to verifying the traffic distribution at the TSE regardless of how the PTS processed the traversing traffic (HTTP in this scenario).

TSE: Traffic Distribution in a PTS Scale Out Scenario

NFV implies a major change of network service scalability concepts. Scaling out virtual instances, for example, can add on-demand capacity under certain load conditions. In this test we triggered the scalability operation manually. The distribution of traffic on the scaled out setup, as Sandvine's team explained, is handled via the policy configuration file on the TSE. We modified the original test bed to evaluate the scale out function. Two TSE instances were configured to send traffic towards a single PTS instance. The second PTS instance was initially deactivated. We generated

test traffic at a rate that saturated the links connecting the operating PTS to the cluster network. The throughput was limited by the single PTS network port capacity of 10 Gbit/s. Next, we launched the second PTS instance and expected the traffic rate to increase. We measured 16 Gbit/s of total throughput after launching the second PTS.

We observed that when TSE forwarding service configuration was updated, TSE's forwarding service required a manual restart to activate the new policy configuration. During the creation of this report Sandvine's team informed us that this issue has been addressed in the following software release.

TSE: Traffic Failover and Recovery in a PTS High Availability Scenario

Redundancy and network continuity are very important topics to any service provider. In our test bed, the traffic forwarding layer at the TSE was expected to react to failures in any PTS network function and to minimize service interruption times. Sandvine utilizes a "keep-alive" protocol to monitor the health of the PTS instances over the cluster network (seen in Figure 1).

To measure interruption time we emulated UDP traffic from 2,000 network hosts at a total rate of 100,000 Packets/s. Each packet was 64 bytes in size.

We began the test by starting test traffic generation and monitoring the traffic's stability for 180 seconds. After the stable state had been verified, we manually unplugged the cable connecting PTS-2 to the cluster network. We noticed that TSE's directed all traffic destined to PTS-2 from the TSE ingress to the egress interface (without forwarding it to any PTS instance). This process is often referred to as interface shunting. At this stage, we observed a maximum interruption time of 361 milliseconds (measured on three test runs). Then after 6 minutes the TSEs began forwarding all the traffic that was previously destined to PTS-2 towards PTS-1. We did not observe any network interruptions when the TSEs changed their forwarding policy state from shunting to forwarding state. Sandvine explained that the 6 minutes between both states is set in a configurable policy parameter to avoid network issues should the links flap.

Six minutes after we had reconnected the cable to the PTS-2 physical port, the TSEs started forwarding traffic again towards PTS-2. We observed no traffic loss during the recovery process. Similar to the failover scenario, the six minutes waiting period may be changed according to the use case, as per Sandvine.

PTS: Domain Name System (DNS) Amplification Attack Mitigation

DNS interruptions can cause havoc to any network infrastructure. DNS amplification is a distributed denial of service (DDoS) attack in which the attacker uses small, spoofed DNS requests to direct large DNS responses by legitimate DNS servers towards the victim's network. This can flood the attacked network with unwanted traffic and bring down complete networks and services. Such attacks are frequent and harm both the compromised DNS servers and the victim's infrastructure.

Sandvine implements DNS amplification detection and mitigation algorithms in their Sandvine Network Protection features on the PTS. The detection, as Sandvine explained, works by looking at malicious responses that have a high rate and that cannot be linked to hosts' DNS requests. The algorithm also takes destination ports into account, Sandvine's engineers added. The TSE instances' role was to load balance the traffic effectively amongst the PTS instances, an important function in an attack scenario.

This test was conducted in two steps. In the first step, we generated emulated attacks and measured how much of the DNS attack traffic was detected. The second step involved sending attack traffic alongside regular legitimate traffic expecting that mitigation rules would not affect normal traffic. The DNS attack rate was 12.5 percent of 16 Gbit/s total traffic rate. The attack was generated from 1,000 emulated host IPs towards ten emulated servers. All attacking hosts had IPv4 addresses. Regular legitimate traffic was generated by 2,000 users in the same 80:20 distribution ratio of IPv4 to IPv6 addresses. All the attacker hosts spoofed the IPs of legitimate hosts.

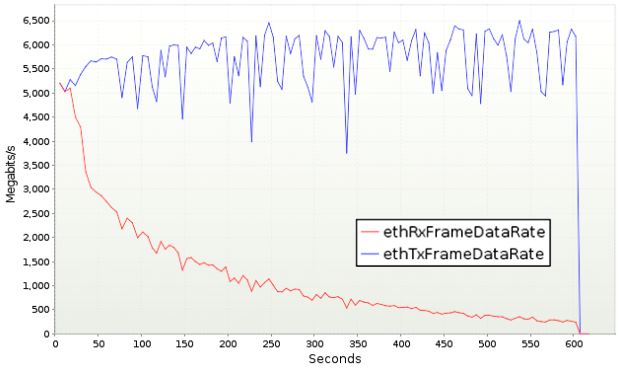


Figure 2: Sandvine PTS DNS Amplification Attack Mitigation Rate

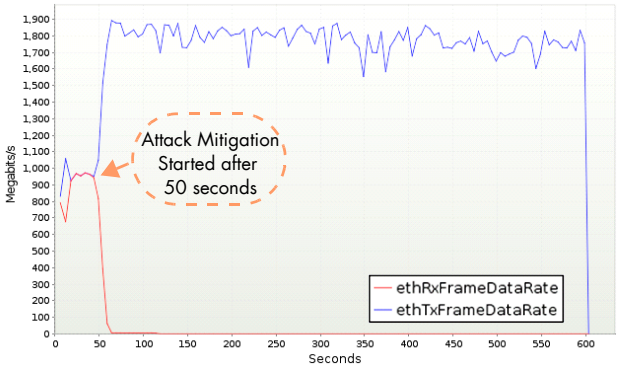
In this test we verified that Sandvine's PTS successfully detected and blocked up to 90 percent of attack traffic. Normal traffic was unaffected by the mitigation policies. Figure 2: Sandvine PTS DNS Amplification Attack Mitigation Rate on page 5 shows the attack traffic rates transmitted and received by the test equipment. The attack mitigation rate was derived from the curve in the received data rate (ethRxFrameDataRate) compared to the generated attack traffic (ethTxFrameDataRate) over a period of ten minutes.

PTS: TCP SYN Flood Attack Mitigation

The TCP protocol can be misused by DDoS attackers to paralyze network services. Stateful protocols, like TCP, depend on session maintenance. SYN is the first transaction in the session establishment process. Attackers can simply generate a burst of - what could look like - legitimate SYN messages and overwhelm the targeted servers.

We emulated 1,000 attacker hosts, half of which were spoofed normal users' IPs while the other half were unique (not shared with any active users). This was to ensure that the detection policy worked in both scenarios combined. We started the test by generating normal traffic and verifying that the SYN flood protection policy had no affect on legitimate traffic. In the second step, we combined normal and attack traffic expecting that only attack traffic would be filtered by PTS. Attack traffic tallied to 1.2 Mpps of SYN messages consuming a total bandwidth of 1.5 Gbit/s split equally on two TSE instances. Legitimate traffic throughput was sent at a rate of 16 Gbit/s towards the same TSE instances. Both TSE instances forwarded traffic towards two PTS instances where the security policies were configured. The result of the test was positive. We observed no interruption on normal traffic while attack traffic was completely eradicated 50 seconds after it had started.

Figure 3 depicts the total attack mitigation rate and a snapshot of the mitigation policy counters on both TSE instances.




```
pts-1> show network-security network-protection mitigation flow-flood

Detector   Type          ActiveRules Detected(bps) Detected(pps) Mitigated(bps)
---------- ------------ ----------- ------------- ------------- -------------
flow-flood [flow-flood]       10,540       613.54M          1.20M       613.54M

Detector   Mitigated(pps) Blocked(bps) Blocked(pps)
---------- -------------- ------------ ------------
flow-flood          1.20M      613.54M        1.20M

pts-2> show network-security network-protection mitigation flow-flood

Detector   Type          ActiveRules Detected(bps) Detected(pps) Mitigated(bps)
---------- ------------ ----------- ------------- ------------- -------------
flow-flood [flow-flood]        9,460       556.68M          1.09M       556.68M

Detector   Mitigated(pps) Blocked(bps) Blocked(pps)
---------- -------------- ------------ ------------
flow-flood          1.09M      556.68M        1.09M
```

Figure 3: Sandvine PTS TCP SYN Flood Attack Mitigation Rate and SUT Counters

Conclusion

Our tests verified four traffic forwarding and distribution use cases of Sandvine's TSE. We chained and load-balanced traffic on two PTS Virtual Series instances. We also verified traffic hashing and load distribution using the inner IP information in normal and GTP tunneled environments. We focused the scalability tests on verifying how the TSE handles a PTS Virtual Series scale out scenario. We also verified network failure and recovery scenarios on the PTS to assess the platform's high availability.

On the PTS Virtual Series side, we tested and verified two security applications, namely the mitigation of TCP SYN flooding and DNS amplification attacks.

With a performance of up to 16 Gbit/s, 2 Mpps and 2,000 subscribers, Sandvine's TSE proved to scale already quite well for some applications. TSE and PTS Virtual Series software worked together neatly in failover scenarios, and PTS's denial of service protection features were state of the art. This report presents the hardware and software configuration all results were achieved with.

Sandvine plans a followup test where EANTC will be permitted to evaluate even higher scalability numbers in a managed, multi-host service chain environment.

Sandvine TSE Platform	**Host Configuration** One HPE ProLiant DL380 Gen9 with • Compute: Two Intel Xeon Processors E5-2699 v3 • Network: – Two dual-port 10GbE Intel Ethernet Converged Network Adapters X520-SR2 – One dual-port 40GbE Intel Ethernet Converged Network Adapter XL710-DA2 • Storage: Two HP EG1200JEHMC 1.2 TB SAS HDD (RAID 1) • Memory: 512 GB DDR4 • Hypervisor: qemu-kvm-1.5.3-105.el7_2.7.x86_64 **Sandvine TSE Instance Configuration** • Flavor: TSE-BG1 • CPU Allocation: Ten vCPUs • Data Ports: Two 10GbE (Passthrough) • Service Ports: One 40GbE (SR-IOV) • Software Version: – svtse-1.00-0036.pts_tse_dev_integration – svpfm-7.40-0036.pts_tse_dev_integration
Sandvine PTS Platform	**Host Configuration** One HPE ProLiant DL380 Gen9 with • Compute: Two Intel Xeon Processors E5-2699 v3 • Network: – Two dual-port 10GbE Intel Ethernet Converged Network Adapters X520-SR2 – One dual-port 40GbE Intel Ethernet Converged Network Adapter XL710-DA2 • Storage: Two HP EG1200JEHMC 1.2 TB SAS HDD (RAID 1) • Memory: 512 GB DDR4 • Hypervisor: qemu-kvm-1.5.3-105.el7_2.7.x86_64 **Sandvine PTS Instance Configuration** • Flavor: VPL-1MD • CPU Allocation: Ten vCPUs • Data Ports: Two 10GbE (Passthrough) • Service Ports: One 40GbE (SR-IOV) • Software Version: – svptsd-7.40-0118.pts_tse_dev_integration – svptsm-7.40-0118.pts_tse_dev_integration – svpts-7.40-0118.pts_tse_dev_integration
Traffic Generator	**Host Configuration** Two HPE ProLiant DL380 Gen9 servers, each with • Compute: Two Intel Xeon Processors E5-2699 v3 • Network: Three dual-port 10GbE Intel Ethernet Converged Network Adapters X520-SR2 • Storage: Two HP EG1200JEHMC 1.2 TB SAS HDD (RAID 1) • Memory: 512 GB DDR4 • Hypervisor: VMware ESXi 6.0 Update 2 **Ixia BreakingPoint System** • Software Version: 8.10.0_EA **Ixia IxLoad VE** • Software Version: 8.10.0_EA

Table 2: Hardware and Software Configuration

Acronyms and Abbreviations

3GPP	3rd Generation Partnership Project
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DNS	Domain Name System
DPI	Deep Packet Inspection
GTP	GPRS Tunneling Protocol
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
KVM	Kernel Virtual Machine
Mpps	Million Packets per Second
NFV	Network Functions Virtualization
NIC	Network Interface Controller
NSH	Network Service Header
PCEF	Policy and Charging Enforcement Function
PDN	Packet Data Network
PTS	Policy Traffic Switch
Qemu	Quick Emulator
SFC	Service Function Chaining
SR-IOV	Single Root Input/Output Virtualization
TCP	Transport Control Protocol
ToS	Type of Service
TSE	Traffic Steering Engine
UE	User Equipment
VAS	Value-added Service
VNF	Virtual Network Function

About EANTC

EANTC (European Advanced Networking Test Center) is internationally recognized as one of the world's leading independent test centers for telecommunication technologies. Based in Berlin, Germany, the company offers vendor-neutral consultancy and realistic, reproducible high-quality testing services since 1991. Customers include leading network equipment manufacturers, tier-1 service providers, large enterprises and governments worldwide. EANTC's proof of concept, acceptance tests and network audits cover established and next-generation fixed and mobile network technologies. http://www.eantc.com

EANTC AG, Salzufer 14, 10587 Berlin, Germany
info@eantc.com, http://www.eantc.com/

v4 20170111