Performance and Functionality Tests of the Virtualized Metaswitch Perimeta SBC

Introduction

EANTC was commissioned by Intel under the Intel® Network Builders program to perform independent tests of the Metaswitch Perimeta SBC solution.

Perimeta is a session border controller (SBC) that is already widely deployed as a stand–alone server solution. Consistent with the current process of network function virtualization and software–defined networking, Metaswitch has been offering a virtualized Perimeta, capable of running on a range of virtualization platforms such as OpenStack and VMware on off–the–shelf server hardware.

In our series of tests we reviewed Perimeta's performance and functionality. We demonstrated that Perimeta can replace legacy systems, without sacrificing any of the functionality or performance, yet giving the operators an easy and flexible way to deploy this network function in their cloud.

Using a range of tools and analyzers, we emulated various scenarios that involved both call signaling and audio stream processing in Perimeta. In most performance tests we emulated identical functionality in a number of representative deployment scenarios, from plain Session Initiation Protocol (SIP) signaling within a company network to the advanced Voice-over-LTE (VoLTE) scenarios suitable for call handling in mobile networks.

We evaluated a range of functional features of Perimeta, its ability to assist troubleshooting and provide high–availability clustering. Finally, we evaluated how the virtualized Perimeta compares to the conventional bare-metal Perimeta appliance, or when running virtualized on OpenStack versus VMware platform.

Overview of the User Interface

In preparation for the tests we reviewed the capabilities of the user interface. Metaswitch's goal is to allow the transition to the virtualized version of Perimeta from the legacy system as straightforward as possible for their customers, and this of course includes the management and configuration.

Test Highlights

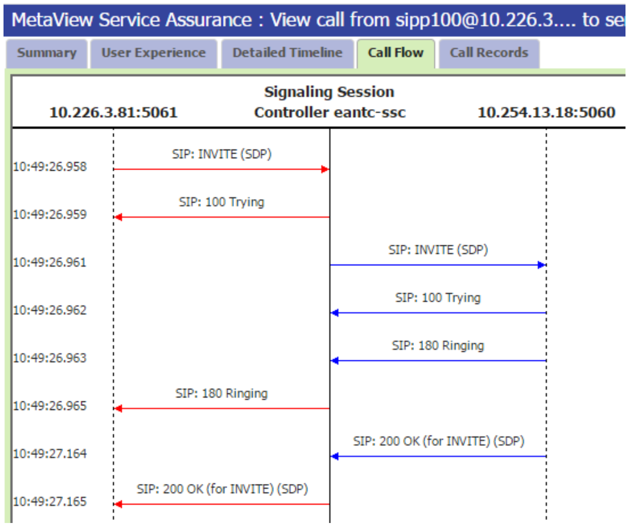
- ➔ **Consistent, identical functionality with the legacy Perimeta SBC appliances**
- ➔ **Alarms, events logging and call analysis tools with Service Assurance Server**
- ➔ **Stable performance, reliable operation in overload conditions**
- ➔ **Up to 700 signaled calls per second**
- ➔ **Up to 78 000 simultaneous media sessions**
- ➔ **High availability verified**

Primarily, Perimeta is controlled via different means, including command line interface reachable via a management network. We briefly reviewed the management interface on the OpenStack- and VMware-based deployments of the virtualized Perimeta, and compared them to the legacy non-virtualized setup. We confirmed that all three versions were practically indistinguishable in terms of configuration structure and commands. According to Metaswitch, a customer can easily transfer an existing configuration from a legacy system to a virtual one with only changes required in the IP configuration.

Perimeta provides a flexible pool licensing model, allowing the operator to easily provide licensing for all involved nodes with a single file.

The Service Assurance Server (SAS) is a tool provided by Perimeta for access to various functions useful for troubleshooting calling issues. SAS is accessible via web interface and offers the operator a flexible way to view, search and filter alarms and notifications generated by the SBC. Specific calls can be easily searched by the subscriber number, call ID, date or address. SAS collects and matches events from various sources, allowing the operator to easily correlate them. For example, call signaling can be directly superimposed on the events from the media processing.

Alongside our performance tests, we verified that SAS was able to collect data on every call set up during the tests, at a rates of up to 700 calls per second. Each of the calls can be reviewed as a ladder diagram of SIP messages, and each message analyzed in full detail. We extensively used this functionality in many of our tests to verify the correct call progress.


MetaView Service Assurance : View call from sipp100@10.226.3.... to se
Summary
User Experience
Detailed Timeline
Call Flow
Call Records

```
                    Signaling Session
10.226.3.81:5061    Controller eantc-ssc    10.254.13.18:5060
10:49:26.958  SIP: INVITE (SDP)
10:49:26.959  SIP: 100 Trying
10:49:26.961                            SIP: INVITE (SDP)
10:49:26.962                            SIP: 100 Trying
10:49:26.963                            SIP: 180 Ringing
10:49:26.965  SIP: 180 Ringing
10:49:27.164                            SIP: 200 OK (for INVITE) (SDP)
10:49:27.165  SIP: 200 OK (for INVITE) (SDP)
```

Figure 1: Call Flow Analysis with SAS

Perimeta constantly collects a wide range of metrics including CPU and memory utilization, this data can be potentially used by the operator to monitor Perimeta's statistics in near real-time. The generated statistics logs can be copied from the machine and analyzed by external tools, for example a spreadsheet application or scripts. In our tests, we used an in-house solution Bricks to collect and graph this data automatically.

Test Methodology and Setup

All our tests were performed at Metaswitch's lab and ran over a small network incorporating the compute nodes to run the Perimeta software and several other servers for the traffic generators and analyzers. The test setup is schematically presented in the diagrams below.

From the logical perspective, the System Under Test (SUT) consists of the virtualized instances of Signaling Session Controllers (SSC) responsible for the SIP processing and multiple (in our case three) instances of Media Session Controllers (MSC), which process media streams. Each of the instances was configured with 8 vCPUs and 4 vNICs. 16 GB RAM was allocated to the SSCs and 8 GB RAM on the MSCs.

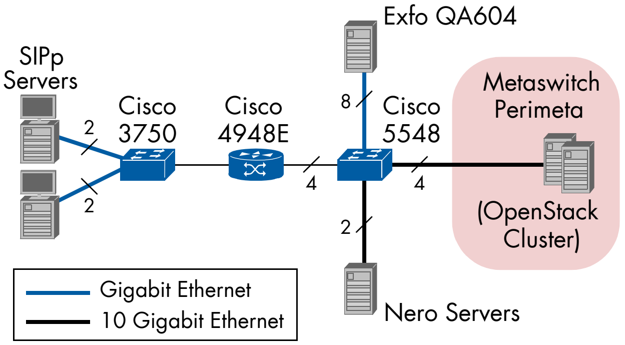


Figure 2: Physical Test Setup

All components are configured as high–availability clusters with an active and a standby instance. Our test setup used 2 OpenStack nodes. In one of the tests we replaced OpenStack installation with VMware using identical hardware and resource allocation in order to compare the performance. Both OpenStack and VMware setups used SR–IOV drivers for the increased packet forwarding performance. We used 10GE links to provide sufficient bandwidth for the RTP and attack traffic.

All active VMs were running on the same physical compute node. We also deployed the VNFs of SAS and Perimeta's licensing server to run alongside Perimeta on the same hardware to better simulate conditions of a cloud deployment by providing additional load on the compute nodes.

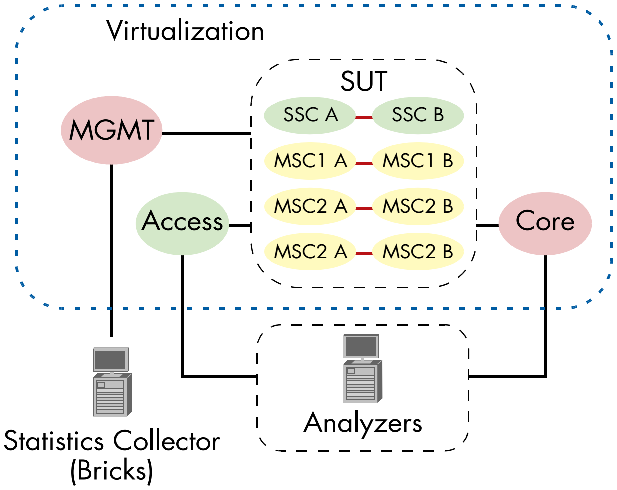


Figure 3: Logical Test Setup

The components were orchestrated with access to two virtual networks representing the access and the core networks where we attached traffic generators, as well as the management network.

Test Tools

For our performance tests, we utilized a number of tools and analyzers to generate the load on the SUT and evaluate the results.

Most of our test cases dealt with the SIP signaling, which is processed in Perimeta's SSC (Signaling Session Controller) component. In most of these test cases we used the open-source tool SIPp to generate and verify a predefined flow of messages to perform subscriber registration or set up calls. The tool emulated both subscriber and the network side and verified whether each call progressed as intended or encountered issues.

Using an external script, we performed multiple test runs for each test case, gradually increasing load until we detected the first call failures or degradation in media quality, indicating that the SUT had reached its performance limits. The results specified in the sections below correspond to the maximum setting where the test runs performed successfully.

In the test cases involving audio streams, we utilized Metaswitch's in–house tool Nero, an application that controls multiple SIPp instances to perform signaling and then utilizes signaling information to transmit and analyze audio streams. Nero determines the quality of the call based on packet loss and delay measured in the test.

In the test cases involving TLS– or IPsec-encrypted SIP signaling, we used a different analyzer better suited for this task, Exfo QA604. This analyzer is capable of both SIP signaling and audio quality measurements.

For the tests related to security, we utilized Metaswitch's in-house attack framework DOOM, which utilizes several other tools such as Scapy and Mausezahn to emulate large-scale distributed attack traffic.

Test Scenarios

In order to realistically evaluate the performance we emulated two call scenarios presented in the diagrams below – a simple SIP call with seven messages and a more complex VoLTE call with 20 messages. We tested both scenarios also in their secure form, involving TLS and IPsec (in case of IMS-AKA).

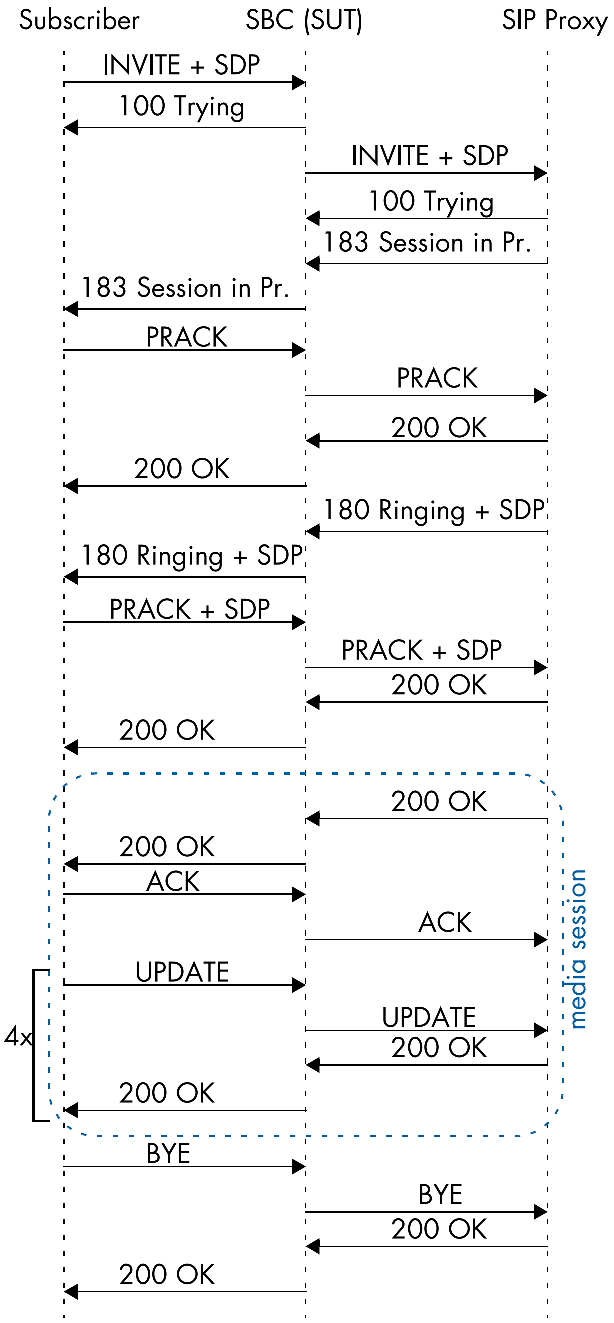


Figure 4: VoLTE Call Flow

In some of the test cases that required calls to be established and run indefinitely, we removed the call tear down phase, effectively reducing the sequence of SIP messages to 5 and 18 respectively.

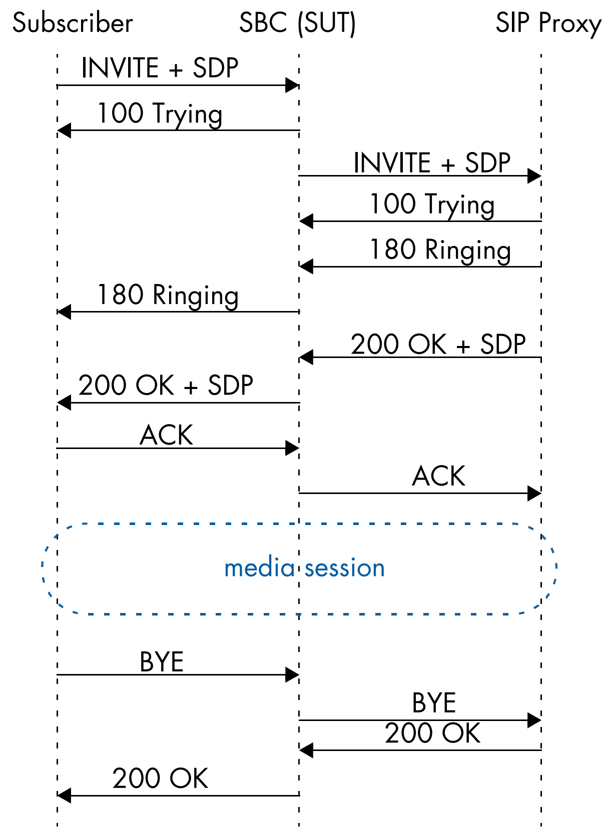


Figure 5: SIP Call Flow

Signaling Performance Tests

Call Signaling Performance

The main metric for SSC performance is the rate at which it can successfully signal calls. When we take a look at a low level, the signaling of a single call involves a number of SIP messages exchanged via SIP protocol, which depends both on the call scenario and on the underlying transport. Depending on the call flow scenario used in the test, we expected different performance to be achieved.

We used SIPp software to emulate the unencrypted SIP traffic and an external Exfo QA604 analyzer in the test cases that involved TLS and IPsec transport. In each case, the analyzers emulated one SIP proxy on each of access (toward subscribers) and core (toward network) sides, interacting with the Perimeta SSC. On the subscriber side, we defined a pool of 1 million subscribers, through which the tester would constantly cycle, establishing short 30–second calls to generate a steady signaling load on the SUT.

All calls in this test case used media bypass, i.e. only the signaling component of Perimeta – the SSC, was involved. The signaled media streams would be exchanged between the subscribers and the network directly without processing in Perimeta's MSC component and were not emulated in the test.

Scenario	Max.Call Rate [calls/s]
SIP	700
SIP+TLS	600
VoLTE	240
VoLTE+IMS–AKA	228

Table 1: Call Signaling Performance

During the test, we continuously monitored the CPU load on Perimeta’s VMs. Each of the instances occupies 8 vCPUs (hyper threads). Some of the cores were exclusively used by the Data Plane Development Kit (DPDK) drivers and loaded at 100 % by design regardless of traffic. Other vCPUs were allocated to the Perimeta application; we verified that these cores were fully utilized during the test run, confirming our expectation that the application uses all available compute resources for maximum performance.

These tests confirmed Perimeta’s performance in various deployment scenarios. The figures achieved for the call signaling rate were then used as the baseline performance in the other test cases.

However, we did not stop here: Additionally we evaluated the performance of the SBC in an overload situation. Using two of the call scenarios, the unencrypted SIP and VoLTE calls, we increased the attempted call rate to twice the maximum rates we had previously measured.

Metaswitch explained that Perimeta constantly monitors its compute usage. When it detects that the message queue length exceeds a certain threshold, it will begin to dynamically prioritize messages for active calls over messages related to new call attempts. This way, existing calls are allowed to be signaled correctly while new calls may be rejected. In overload conditions, Perimeta continued to serve calls within its performance limits and handle at least 95 % of the call rate achievable under normal conditions. In the SIP scenario, we achieved 668 calls/s, or 95.4 % from the baseline 700, and in the VoLTE scenario 237 from the baseline 240, or 98.8 %.

Subscriber Registration Performance

In our next series of tests we examined the performance of the SBC when it comes to registering the subscribers. We tested five registration scenarios that may occur in different subscriber environments.

The subscribers need to perform a registration against SBC before calls from these subscribers are allowed. The process verifies that the subscriber is the same as configured in the softswitch. The registration is also needed to know where a specific subscriber is located, in case of incoming call.

By default, clients located in the enterprise networks only issue re–registrations once per hour. However, the subscribers in the open Internet are often located behind NAT. For these subscribers to be reachable by the SBC in case of an incoming call, the NAT mapping established from the client side has to be regularly renewed. This is achieved by the client by re–sending the registration every 30 seconds.

These re–registrations would place a significant additional load on the network. Perimeta optimizes this process by caching the subscriber registrations and only forwarding them to the adjacent softswitch once every 30 minutes to prevent the registration from timing out in the core. With this method, we expected Perimeta to handle a reasonably high number of subscribers and a high rate of re–registrations. In our tests we achieved 14000 re-registrations per second by simulating 420000 subscribers for the unencrypted SIP, and 4400 re-registrations per second (132000 subscribers) with TLS encryption.

We tested both these scenarios in their plain and TLS–encrypted form, as well as a more complex registration process used in VoLTE/IMS–AKA.

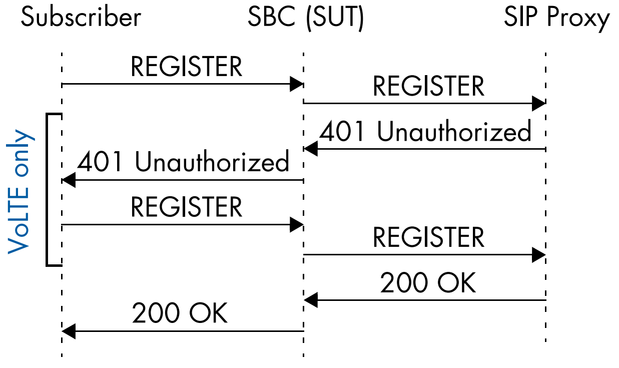


Figure 6: Registration Message Flow

With this test we confirmed that Perimeta is able to register a reasonably large subscriber base within short time. For example, in case of a power or network failure, many SIP clients may attempt to register within a short time. With the supported registration rate of 2600 subscribers per second, our Perimeta instance would be able to register its supported maximum of 700 thousand subscribers within just 5 minutes.

Scenario	Reg. Rate [msg/s]
Plain	2600
TLS–Encrypted	720
Fast registration (Plain)	14000
Fast registration (TLS)	4400
Authenticated VoLTE IMS–AKA	810

Table 3: Registration Performance

SIP Session Capacity

In the next step we tested the capability of Perimeta to support a large number of parallel calls. Since our call flow does not include media or further SIP transactions during the call, a large number of calls primarily has an effect on the memory consumption necessary to hold a large number of active sessions.

We performed the test using the same set of four call flow scenarios, excluding the terminating sequences. The calls were established at 500 calls/s in the SIP and 200 calls/s in VoLTE scenarios.

Perimeta automatically places a hard limit on the number of concurrent calls in order to avoid resource exhaustion. Depending on their type, each call consumes a certain number of resource units. Depending on the complexity of the call, transport and whether the media is being present and processed, Perimeta estimates the number of concurrent calls it can process and calculates a limit.

Scenario	Concurrent Calls	RAM usage [%]
SIP	85000	56
SIP+TLS	85000	78
VoLTE	85000	52
VoLTE+IMS–AKA	50000	37

Table 4: SIP Session Capacity

In our tests, we selected a value close to that limit demonstrated that all sessions could be successfully established and maintained for the entire test duration.

Media Performance Tests

Our previous series of tests only dealt with SIP signaling performance and only the SSC component of Perimeta was involved in simulating scenarios where media streams would flow directly between subscribers.

Our next series of tests covered the situations where the media streams have to go through the MSC component of the SBC in order to be forwarded to a different network and possibly transcoded. The main metric of the tests was the number of simultaneous calls with media.

Media Session Capacity

In order to test the media session performance of Perimeta, we used the in–house tool Nero in combination with SIPp to establish a large number of calls with media sessions at a rate of 250 calls per second and maintain them over a long period. For the media streams we used G.729A codec with 20ms packetization interval.

To locate the performance limit, we periodically added a batch of new calls to reach the next load level and measured packet loss in the existing sessions for five minutes. We determined the SUT's capacity as the highest number of parallel calls where the packet loss in the RTP flows did not exceed 0.001 %. With plain RTP traffic we achieved 78000 parallel media sessions. This number of calls corresponds to approximately 7.8 million packets per second that the SBC had to forward.

We performed a similar test using Secure RTP (sRTP) traffic. Since Perimeta can act as a SIP gateway between networks, its ability to transparently encrypt and decrypt media streams can be used to transport phone calls over untrusted networks. In our test setup we emulated subscribers on the access side using sRTP media, which was then decrypted by Perimeta and transported to the core side using plain RTP. In order to achieve necessary performance, we used a combination of Exfo analyzer and Nero software. An Exfo QA604 chassis and two Nero servers provided a continuous background load of 15000 calls, and a third Nero server provided an increasing call load to locate the performance limit at 24000 standing calls.

In our third measurement scenario for the media capacity test, we used G729AB codec for the emulated media streams that allows use of silence suppression, effectively reducing the packet rate and allowing us to establish even more concurrent calls. In our final test we ran 35000 parallel calls with sRTP media through our Perimeta setup with three MSC units.

Performance on VMware platform

In the conclusion of our performance tests, we demonstrated that Perimeta is able to run on a different virtualization platform without performance degradation.

For this test, we orchestrated an identical setup consisting of one SSC and three MSC units in a high–availability cluster on a VMware platform using identical hardware.

We repeated the media capacity test by running up to 70 000 parallel calls with RTP media in a setup with three MSC units, thus subjecting it to a similar load we could achieve on the OpenStack–based installation.

The following table summarizes the results of the three scenarios:

Scenario	Concurrent Calls
Plain RTP	78000
sRTP	24000
sRTP w/ silence suppression	35000
Plain RTP (VMware)	70000

Table 5: Media Session Capacity

In all media session capacity test scenarios we monitored media session statistics on each MSC instance. We confirmed that Perimeta SBC evenly distributed media sessions across multiple MSC instances. This feature allows the operators to easily adjust the media processing capacity of Perimeta SBC to their demand by adding additional MSC instances.

Advanced Functionality

High Availability

All components of Perimeta are operated as high–availability clusters in order to protect them against hardware, software or network failures. Both SSC and MSC functions are running in active/standby pairs, constantly synchronizing their state and verifying availability via a proprietary protocol.

From the networking point of view, each component in the pair has its own set of IP addresses, which includes a management IP and multiple service interfaces. At the same time, each pair maintains virtual IP addresses visible to the external entities. In case of a failure, the standby component can immediately detect it and take over the function and any existing traffic flows.

Perimeta does not place any restriction on the location of each component. Multiple components can be located on the same hardware platform, or spread over multiple servers within the cloud. The usual OpenStack or VMware orchestration takes care of the mapping of the virtual networks or SR–IOV interfaces to the SBC instances.

In the case of failover, each Perimeta component, including the individual MSC instances can perform a switchover to its standby instance separately. For example, a failure of one MSC instance will lead to the failover of this instance only and will not produce unnecessary failover impact on the other instances still intact.

In our test, we analyzed the impact of a failover on the standing calls and their quality. Using both Nero servers and Exfo analyzer, we established 75000 parallel calls with media. In our standard test setup, these calls were evenly spread over 3 MSC instances.

When all calls were active, we forced a failover of one of the MSC instances by issuing a command via the management interface of Perimeta. Although it was a manually triggered failover, internally it is implemented as a shutdown of an instance and triggers exactly same detection mechanisms as in case of hardware or software failure.

From the packet loss and delay measurements that were constantly performed by the Exfo analyzer during the test, we determined that the failover only caused a brief degradation of quality that lasted less than 1 second. At most 13 lost packets were reported for any call.

After the failover was completed, we did not observe any further impact on existing or new calls and no measurable change in the audio quality.

Header Manipulation

One of the important functions offered by Perimeta is the ability to manipulate SIP headers in accordance with rules defined by the operators. This functionality can be important for deployments suffering from interoperability issues. For example, a certain type of IP phones used in a company may turn out to be incompatible with other clients, or the SIP servers, because they expect a different fields or formatting in the SIP headers.

Perimeta is capable of manipulating selected values and can provide an interoperability layer between these devices and servers. The rules can be applied for incoming and outgoing messages, and on each of the service interfaces (i.e. access–facing or network–facing) separately.

We verified the header manipulation function of Perimeta by defining a set of rules and simulating calls through Perimeta. We defined a ruleset on both access and core–facing interfaces that modified two headers and added two others for each forwarded message. We used the logs produced by the Service Assurance Server (SAS) for each call to confirm that the intended manipulation was applied correctly.

We also measured the impact of the function on the call signaling performance. We used SIPp to emulate SIP calls at a high rate. With the header manipulation enabled, we achieved 650 calls per second without failures, a figure only slightly lower than the 700 calls per second in the baseline test.

Protection against Attacks

As with any service, a SIP server may become a target for various attacks. They can arise both from malicious intent, or due software malfunction or misconfiguration. The service may be also subjected to a spike of calling activity.

In our final series of tests, we verified the stability and performance of Perimeta under heavy attack traffic and its ability to protect further SIP infrastructure.

Softswitch shielding

One of the basic protection functions is the graceful limiting of the call rate to a defined value. Instead of attempting to process as many calls as possible and running a risk of overloading the Perimeta itself or other servers, e.g. adjacent SIP proxy, this function can reduce the call rate to a safe limit, while properly rejecting excess calls with an appropriate SIP error message.

We verified this functionality by running a call rate test with the offered load of 700 calls per second – the value we achieved in the baseline test, while a rate limit of 400 calls per second was configured on Perimeta.

Perimeta successfully applied the defined limit and we verified the 400 calls per second were signaled without issues. The remaining 300 calls per second did not fail but were rejected with a 486 Busy Here message.

Protection Against DDoS

Finally, we subjected Perimeta to various DDoS attacks. The SBC had to sustain the baseline call signaling rate, while receiving high volume attack traffic.

We used SIPp in combination with Metaswitch's in–house attack simulation framework DOOM that combines a number of widely available tools to generate various attack flows.

Two of the attacks were directed at SIP service itself. We generated a high volume of INVITE and REGISTER messages. Since SIP used UDP by default, such attacks can be emulated using a large number of spoofed source IP addresses.

For the INVITE flood, we set up a single SIPp server to emulate regular SIP traffic, while two other SIPp servers generated INVITE messages at a high rate. In case of INVITEs, Perimeta is able to separate legitimate traffic from the attack traffic, as it must come from trusted IP addresses – either registered clients, or adjacent SIP proxies. With 6.5 Gbit/s of attack traffic, Perimeta successfully signaled 700 calls per second

The REGISTER flood however is a more complicated issue, as legitimate registration requests can potentially come from new, untrusted addresses. In this case, Perimeta uses a different technique, by detecting the attack and maintaining a dynamic blacklist. Once it detects that specific addresses send the registration messages at a high rate, these are added to the blacklist.

However, some time is required to correctly recognize the REGISTER flood attack. In our test we experienced a brief overload of the SBC due to attack traffic at the very beginning, which led to approx. 20 % of failed calls. Once the blacklist was established 30 seconds later, we observed no further call failures.

With both INVITE and REGISTER flood attacks we showed that Perimeta is very capable of protecting the infrastructure against SIP–based attacks.

The final two tests involved ICMP Echo and TCP SYN floods. In all test cases we confirmed the ability of Perimeta to maintain the baseline call signaling performance of 700 calls per second under ongoing attack:

Attack Type	Attack Rate [Mbit/s]	Call Rate [calls/s]
INVITE	6550	700
REGISTER	470	700
ICMP Echo	246	700
TCP SYN	115	700

Table 6: DDoS Attacks

Test Bed Specifications

The following table summarizes the specifications of the hardware and software we used in our test bed to run the Perimeta's instances as well as the test tools.

Perimeta Platform
Hardware: Dell PowerEdge R630 CPU: 2x 12-core Intel Xeon Haswell E5-2690v3, 2.6GHz (hyperthreading enabled) NIC: 2x 10GE Intel 82599ES, 2x 1GE Intel I350 RAM: 12x 16GB Hynix DDR4-2133 SDRAM Software: Hypervisor: QEMU 2.1.2, VMware ESXi 6.0.0 (VMware test only) OpenStack Kilo Metaswitch Perimeta V4.0.00_SU3_P01 SSC: 8 vCPUs, 16 GB RAM, 4 vNICs MSC: 8 vCPUs, 8 GB RAM, 4 vNICs
SIP Traffic Generators
Hardware: Dell PowerEdge R610 Fedora 12 SIPp v3.4.1-PCAP-RTPSTREAM Nero V1.1.3 Exfo QA604 v9.8.3
DDoS Attack Simulation
Hardware: Dell PowerEdge R610 CentOS 6.7 Scapy v.2009-07 Mausezahn v.0.40
Supporting Network Infrastructure
Cisco 3750 Cisco 4948E Cisco Nexus 5548

Table 7: Hardware and Software used in the Tests

Conclusion

With this series of tests we verified and measured the capabilities of Metaswitch's virtualized Perimeta solution from a number of perspectives. We measured signaling and media forwarding performance in a variety of call scenarios and could confirm the figures suggested by Metaswitch.

We saw Perimeta achieving up to 700 established calls per second and up to 78000 simultaneous calls with audio streams, evenly load-balanced across multiple media session controller instances. In heavy overload conditions, Perimeta was still able to successfully process calls at least 95% of the baseline rate.

In the subscriber registration scenarios, we confirmed that Perimeta can handle up to 2600 registering clients per second and is also able to protect adjacent SIP proxies by handling high-rate re-registrations load.

In our functional tests we reviewed and verified Perimeta's advanced functionalities giving the operator the possibility to easily troubleshoot and workaround signaling issues that may occur in the network.

With our security tests, we confirmed Perimeta's ability to withstand various application- and network layer attacks without negative effects on the performance.

In summary, Metaswitch successfully demonstrated the readiness of their Perimeta solution for the transition from the legacy systems to the cloud.

About EANTC

The European Advanced Networking Test Center (EANTC) offers vendor–neutral network test services for manufacturers, service providers and enterprise customers. Primary business areas include interoperability, conformance and performance testing for IP, MPLS, Mobile Backhaul, VoIP, Carrier Ethernet, Triple Play, and IP applications.

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