

AALTO UNIVERSITY

ELEC-E7910 - SPECIAL PROJECT IN  
COMMUNICATIONS ENGINEERING

---

**Performance Comparison of SCION  
with Routed IP on Virtual Machines**

---

*Author:*

Dheeraj Chandrashekar

*Advisor:*

Maria Riaz

November 5, 2020

# Acknowledgements

I thank professor Raimo Kantola for giving me this opportunity to work on SCION and guiding me throughout the project. I would also like to thank my advisor Maria Riaz, for providing all the support needed from the start of this project till finalizing this report. This work was carried out under the 5G FORCE project of the department of Communications and Networking, Aalto University.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>3</b>
<b>3</b>	<b>SCION Applications</b>	<b>6</b>
<b>4</b>	<b>SCION Network In Finland</b>	<b>9</b>
<b>5</b>	<b>Test Setup and Scenario</b>	<b>10</b>
<b>6</b>	<b>SCION Performance Testing</b>	<b>12</b>
6.1	Latency Test . . . . .	12
6.2	Bandwidth Test . . . . .	13
6.3	HTTP Load Test . . . . .	15
<b>7</b>	<b>Result Comparison and Analysis</b>	<b>18</b>
7.1	Latency test . . . . .	18
7.2	Bandwidth test . . . . .	19
7.3	HTTP Load test . . . . .	20
<b>8</b>	<b>Repeatability of Tests</b>	<b>21</b>
8.1	Test bed and Connections . . . . .	21
8.2	GitLab . . . . .	22
<b>9</b>	<b>Conclusion and Future Work</b>	<b>23</b>
<b>A</b>	<b>Appendix</b>	<b>26</b>

# 1 Introduction

SCION stands for Scalability, Control and Isolation on Next-Generation Networks, first of its kind to provide a clean-state Internet architecture for end-to-end communication [1]. SCION is posed as a network architecture that is designed to offer high availability and efficient point-to point packet delivery, even in the presence of adversaries [2]. Being an open source Internet platform project, it is currently being developed and managed by the Network Security group, ETH Zürich. There are two active deployments of the SCION network, the research version and the production version. The production version is deployed through a company called Anapaya systems [3]. The research version dubbed as “SCIONLab Network” is deployed and managed by SCIONLab team. SCIONLab is a global research network designed to test and experiment with the SCION architecture [4].

The SCIONLab infrastructure comprises a network of globally connected ASes. There are few restrictions in the research version when compared to the production version. For example, SCIONLab centralizes management of the control plane Public Key Infrastructure (PKI) where this kind of single point of failure is absent in the real deployment. Another restriction is that the SCIONLab infrastructure uses overlay links over the publicly routed Internet. Therefore, security, availability, and performance properties of SCION are not fully realized. [4] The production version is already being used by many ISPs, Banks in Switzerland and even the Swiss Government is utilizing SCION for some of their services.

Anyone can join the SCIONLab by registering as a user and can start testing the SCION network. However, in order to run the SCION AS uninterruptedly one must join the SCIONLab infrastructure. Joining the infrastructure would enable global visibility of the node and the option of global connectivity [4]. The SCION ASes deployed in Finland are part of the SCIONLab infrastructure and are visible in the world deployment map, available on the main page of the SCION website. On the deployed ASes in Finland we perform performance tests related to Latency and Bandwidth and compare them with the performance of the current Internet i.e. the routed IP. In order to keep the comparison fair between SCION and routed IP, we try to keep all the parameters same during execution of test scripts on both architectures (discussed more later).

There are currently three SCION ASes deployed in Finland which is explained in more detail in the upcoming sections. All the above-mentioned performance related tests are performed over these three SCION ASes. Since the test procedure would be the same between any two ASes and to eliminate the monotonous manual testing, we automated the performance tests using Robot framework. Using the existing libraries and few custom libraries we have written generic test scripts aiming to run between any two SCION ASes just by changing the inputs. The ability of re-using the test scripts enables easy testing with just a click of a button and saves time. All the test scripts, instructions of running them, SCION AS's topology details of Finland, important documents related to SCION, test bed, configurations of the same, documentation of each concept tested over SCIONLab, test results, analysis, open ideas, FAQs and all SCION related information concerned to Aalto are present in the dedicated GitLab page [5].

## 2 Background

Current Internet is evolving at a rapid rate, there is fast advancement of Internet applications and services. However, analyzing further, we can see that only parts of the protocol stack have changed. The application and physical layers have adapted to new needs and trends, but the core protocols have remained mostly the same for decades [2]. Internet Protocol (IP) and the Border Gateway Protocol (BGP) are the core technologies that are still running the backbone of the Internet Communication and have been in use for more than 30 years.

IP enables the forwarding of packets between end hosts by determining the next hop based on the destination address. This simple looking protocol has few drawbacks such as lack of separation between routing and forwarding, lack of transparency or control over the path the packet takes to reach a destination, maintaining forwarding tables on routers which would mean more time consuming for route table lookup [2]. Border Gateway Protocol(BGP) is a routing protocol used to achieve connectivity between independently operated network or autonomous systems (ASes). It allows ISPs to perform traffic engineering and select routes based on policies. The shortcomings of BGP are many. For example, BGP updates are slow and the outages can cause network downtime, no fault isolation – a single faulty BGP can affect routing in the entire world, scalability is not easy in BGP and BGP does not provide path control to end hosts and does not allow use of multiple AS-level paths [6]. Apart from drawbacks of IP and BGP there are many more issues the current Internet faces. Spoofing, DDoS attacks, forged certificates and prefix hijacking

are few of the attacks that pose serious threat to the current Internet.

If a new Internet architecture is designed to address all the problems mentioned above, then what would be the goals of such an architecture? Its impossible to get rid of the attackers completely in the given Internet, so having an architecture which is available in the presence of the adversaries is a desirable goal [2]. Control and transparency of the paths are also the requirements for improving the internet architecture. The Internet architecture should also meet the requirements of efficiency, scalability, extensibility and deployability. There have been a lot of research in bringing a new Internet architecture, but no one solution can solve all the problems of the Internet. This is where SCION comes into picture, SCION focuses on security and high availability for point-to-point communication [1].

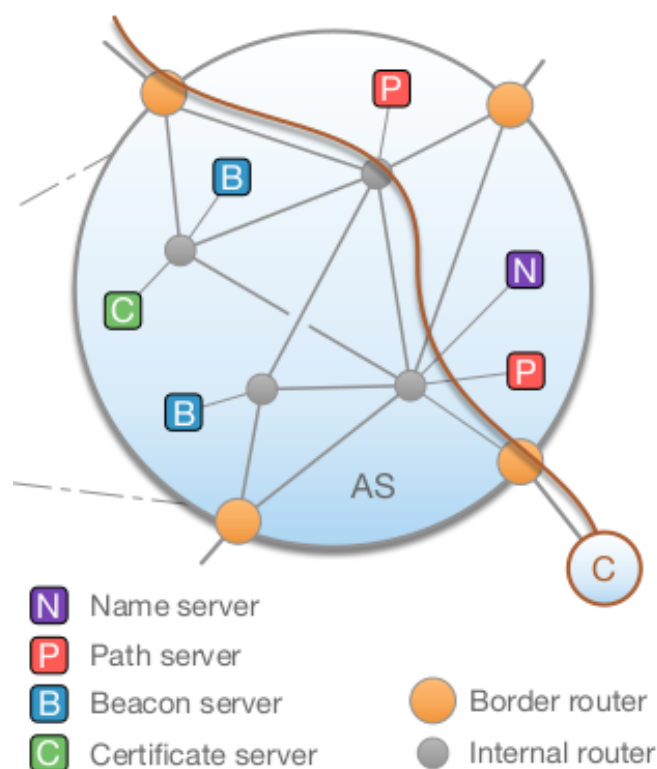


FIGURE 2.1: SCION AS  
[2]

SCION introduces the concept of an isolation domain (ISD), which is a fundamental building block for achieving the properties of high availability, transparency, scalability, and support for heterogeneous trust. An ISD constitutes a logical grouping of autonomous systems (ASes) and is administered by multiple ASes, which form the ISD core referred to as core ASes. The ISD is governed by a policy, called the trust root configuration (TRC), which is negotiated by the ISD core. The TRC defines the roots of trust that are used to validate bindings between names and public keys or addresses. [2]

Figure 2.1 shows the main components of a SCION AS. Beacon servers help in discovering path information and are responsible for generating, receiving, and propagating PCBs to construct path segments, a process referred to as bea-  
coning. Path servers help in disseminating path information, store mappings from AS identifiers to sets of announced paths segments and are organized as a hierarchical caching system like today's DNS. Certificate servers assist with validating path information, keep cached copies of TRCs retrieved from the ISD core and copies of AS certificates, manage keys and certificates for securing inter-AS communication. Name servers provide name resolution from user-understandable names to SCION addresses. In addition, border routers provide the connectivity between ASes, while internal routers forward packets inside the AS. [2]

A SCION address is a 3-tuple of the form (ISD, AS, ADDR), where ISD and AS are identifiers, and ADDR is the end-host address. The ISD identifier is globally unique, the AS identifier is locally unique within the ISD, and the end-host address is routable within the AS.



## 3 SCION Applications

SCION applications are written in Go, with some supporting code in Python. A SCION Internet connection (for instance via SCIONLab) is required to run these applications, along with a running SCION end host [7]. For the purpose of testing the SCION network in Finland, we make use of four applications namely:

- AS Visualization
- Bandwidth Tester
- BAT (cURL for SCION)
- SCION IP Gateway (SIG)

In the below section we describe each application and their uses.

AS Visualization is an application where every AS can be managed locally through a web portal that provides information of the AS, like the topology, certificates, health status etc. This web application provides a graphical level detail of the AS. One can run basic tests from the portal to check if the services are running properly. This web interface comes under the SCION's local management service and has few additional functionalities like defining the local topology, generate configurations for the local servers, send requests to join an existing ISD and obtain certificate and TRC from the core AS etc [2].

Bandwidth tester is a SCION native BW tester application that can be used between the hosts, to determine the bandwidth between them or run bandwidth

tests on the SCION network. The BW tester server can be run on one host and can be tested from the client. However, there are BW test servers throughout the SCION network, which can be used to stress test the data plane infrastructure. A server only allows a single client to perform a bandwidth test at a given point in time. Clients are served on a first-come-first-served basis with each test limited to 10 seconds. The client which connects to the server establishes two SCION UDP connections to the bwtestserver, a Control Connection (CC) and a Data Connection (DC). [8]

BAT (cURL for SCION) is a CLI cURL-like tool for testing, debugging, and generally interacting with HTTP servers over SCION/QUIC. SCION's fork extends BAT with native SCION support. BAT accepts both, SCION addresses and hostnames in URLs. Hostnames are resolved by a RAINS request, where RAINS is SCION native service equivalent to the DNS of the current Internet. The scheme defaults to HTTPS, unencrypted HTTP is not supported. The tool has a support for benchmarking as well, where we can perform HTTP load testing. [9]

SCION IP Gateway (SIG) enables legacy IP applications to communicate over SCION. It can be extended to support multiple SIG endpoints. SIG basically encapsulates the IP packets inside a SCION packet. A special feature of SIG is that applications can be run on global non routable IP address and can still be accessed by SCION clients via SIG. [10]

SCION has its native SCMP (SCION control message protocol) which is utilized for the latency related tests. The SCION Control Message Protocol (SCMP) is analogous to ICMP in the current Internet but provides additional functionalities when compared to ICMP. Network diagnostic where debugging tools such as the SCION equivalents of ping or traceroute are available and error messages can be used to signal problems with packet processing or inform end hosts about network-layer problems. The SCMP protocol is the first instance

of a secure control message protocol in a network infrastructure. [2]

## 4 SCION Network In Finland

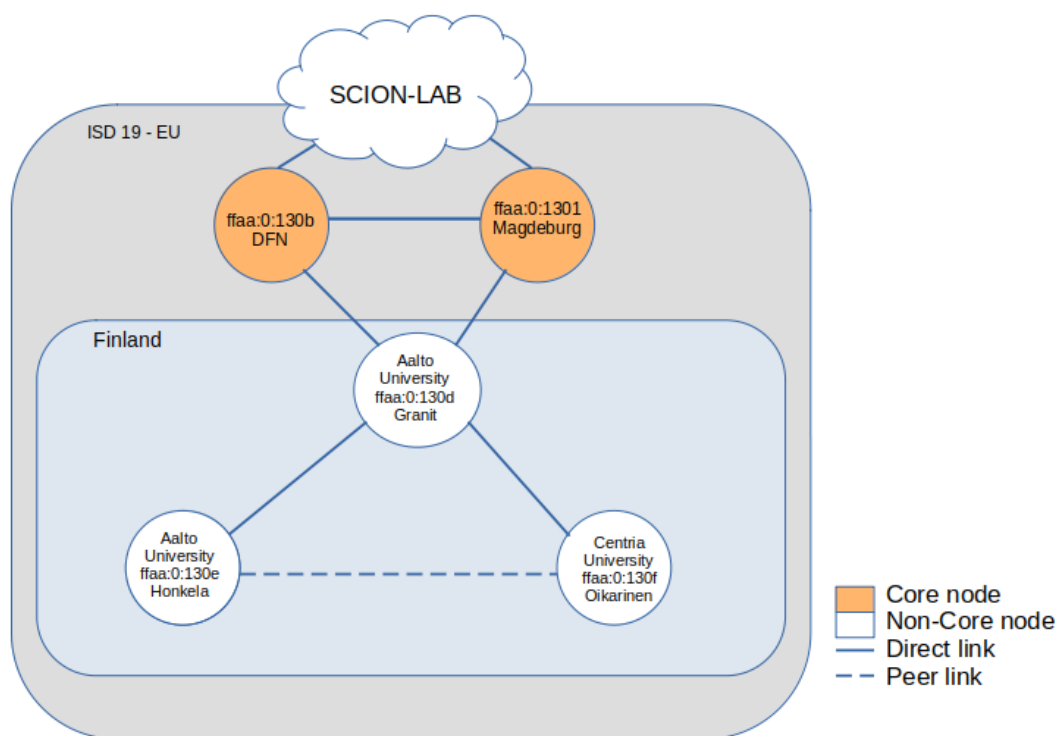


FIGURE 4.1: SCION Network Finland

There are currently three SCION ASes up and running in Finland. Two of the ASes are at Aalto University and One AS at the Centria University of Applied Sciences. All the ASes are running on virtual machines with overlay over routed IP. The connections are shown in the figure 4.1. The connections are as follows: There exists a direct link from Aalto Granit AS to other two nodes and a peer-to-peer link between Honkela and Oikarinen AS. The Granit AS is connected to two core AS i.e. DFN and Magdeburg which are in turn connected to the SCIONLab network. All the ASes shown in the figure above belong to European Isolation Domain i.e. ISD.

## 5 Test Setup and Scenario

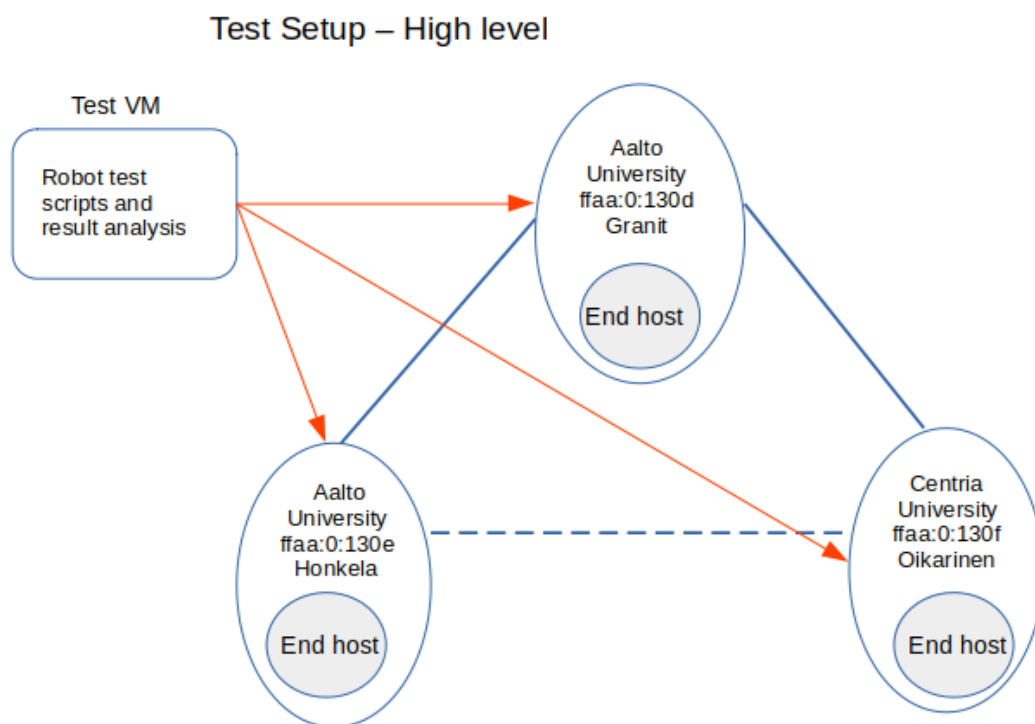


FIGURE 5.1: SCION Test Setup

Figure 5.1 shows the complete test setup and its connections as well. In addition to the three ASes mentioned in the previous section, we have three hosts each connected to the three ASes. Endhost can be configured by anyone and the support of SCIONLab is not required for the same. The instruction of bringing up a SCION endhost is available on the SCION website [5].

In order to have ease of access to all the VMs between many users we decided to configure another VM to act as entry point to the entire SCION network. This VM along with providing access to other VMs, serves as a test client for running all the robot scripts. All the VMs including the test VM is configured

with a common user with name “testuser”. If one has access to the test VM then all other VMs can be logged in via SSH with the same user.

In order to utilize all three ASes for our testing we select a scenario where the same is achieved. Performance tests between different combinations of SCION ASes and endhosts have been performed and the results have been documented in the GitLab page mentioned in the introduction. For the purpose of this report we select Honkela AS at Aalto and Oikarinen AS at Centria, refer figure 4.1. The reason for selecting these two ASes are as follows: Geographically Aalto is situated at south of Finland and Centria is situated at the Western coast of Finland, this distance would be an ideal usecase for testing the SCION network. There exists a peer-to-peer link between the two ASes. Also, there is a direct link between the ASes via the Granit AS, this would in turn mean that there are two dis-joint paths between the selected two ASes, which would provide opportunity to test other SCION features like the behaviour of default multipath communication, failures etc.

We make use of robot scripts to run tests and collect results i.e. metrics of the tests from the test VM. Scripts for the tests are attached in the appendix section. Since the final goal is to compare the test results of SCION with routed IP, we perform IP related performance tests between the same two VMs running the ASes and collect the results for comparison. Routed IP tests are also run through the help of robot scripts. In every test that is performed, the parameters from which the test is defined are kept same for both SCION and IP to be fair while comparing. Since SCION is new, the option to modify the test parameters are limited. Hence, the IP tests were aligned to match with the SCION test. The next section will explain each test in more detail.

## 6 SCION Performance Testing

### 6.1 Latency Test

In the latency test we simply send SCMP packets from one endhost connected to Honkela AS towards the endhost connected to Oikarinen AS. SCION has its own set of tools tagged under SCION utilities which provide functions like ping and traceroute. This would mean that we would be sending ping packets i.e. scmp packets between the host and calculate the round-trip time. Since SCION is a stateless path aware network the entire path from the source to destination is encrypted and present in each packet header. Hence, the size of the packet depends on the path distance between the source and destination. Longer the path, larger the packet size of the SCION packet [2].

In our tests we send different number of packets from source to destination and calculate the round-trip time. In our case the packet size of the SCMP was found to be 136 bytes from Aalto to Centria. We note this value as we will be using this value later. The results are tabulated below:

SCMP			
No of Packets	Avg RTT	Loss %	Total time
1	11.731ms	0	233.497ms
5	11.448ms	0	4.018s
10	12.053ms	0	9.019s
50	11.678ms	2	49.021s
100	11.685ms	2	1m39.224s

FIGURE 6.1: SCION SCMP

Next we perform the latency tests over the routed IP using ping utility between the same endhosts used above. To match the parameters used in the SCION latency test we configure the ping utility to send packets of size 136 bytes from the source to destination for the same number of packets. The tests were performed and the results are tabulated below:

ICMP			
No of Packet <sup>s</sup>	RTT	Loss %	Total time
1	8.296ms	0	0ms
5	8.124ms	0	4.005s
10	8.156ms	0	9.012s
50	8.174ms	0	49.065s
100	8.163ms	0	1m39.131s

FIGURE 6.2: PING using ICMP

Its clear from the results that ICMP is much faster compared to SCION. However, the additional functionalities provided by SCMP is noteworthy. The total time taken for the same number of bytes by ICMP and SCMP remain almost same.

## 6.2 Bandwidth Test

The bandwidth tester application for SCION can be installed from the scion-apps repository. The application is a simple CLI based tool which holds both server and client configurations and can be used to test the bandwidth across the network. The tool can be called in two modes: server or client mode. The tool supports specification of the test duration (up to 10 seconds), the packet size to be used (at least 4 bytes), the total number of packets that will be sent, and the target bandwidth [8]. For instance, 5,100,10,1600bps specifies that 10 packets of size 100 bytes will be sent over 5 seconds, resulting in a bandwidth of 1600bps. The question mark '?' character can be used as wildcard for any



of these parameters. Its value is then computed according to the other parameters. The parameters for the test in the client-to-server direction are specified with `-cs`, and the server-to-client direction with `-sc`. So for instance to send 1 Mbps for 10 seconds from the client to the server, and 10 Mbps from the server to the client, we can use this command: `"scion-bwtestclient -s <server-address> -cs 10,1000,1250,1Mbps -sc 10,1000,12500,10Mbps"`. In our tests we use BW and number of packets as variables and remaining defined as question mark(?). [8] According to the parameter specification mentioned, the BW tester was run in server mode on the endhost connected to Honkela AS and the endhost connected to Oikarinen was run as client and the tests were run for different BW values in both direction and the results are tabulated below:

SCION - BW Tester					
Attempted BW(bps)	Packets	Achieved Bandwidth(bps)			
		Client -> Server	Loss %	Server -> Client	Loss %
10M	1250	9.73M	2.7	10M	0
100M	12500	96.79M	3.2	100M	0
150M	18750	144.92M	3.3	149.93M	0
200M	25000	190.88M	4.5	171.86M	14
500M	62500	300M	40	115.57M	76.8
1000M	125000	281.59M	71.8	141.34M	85.8

FIGURE 6.3: SCION BW tester

Next, we perform the same bandwidth tests between the same hosts but over routed IP using IPerf3. Like the latency tests we match all the tests parameters to compare the results later. IPerf3 doesn't support running bandwidth tests on both direction in a single instance like SCION. Hence, we run two IPerf instance, one for each direction. The results are tabulated below:

IPERF3 – BW Test					
Attempted BW(bps)	Packets	Achieved Bandwidth(bps)			
		Client – Server	Loss %	Server – Client	Loss %
10M	1250	9.74M	2.6	10M	0
100M	12500	97.2M	2.8	100M	0
150M	18750	146M	2.6	150M	0
200M	25000	192M	4	200M	0
500M	62500	486M	2.8	106M	78.8
1000M	125000	944M	5.6	95M	90.5

FIGURE 6.4: IPerf BW test

From the results we can observe that SCION provides a reasonable bandwidth without significant packet loss until an attempted bandwidth of 200 Mbps after which the performance starts deteriorating. The reason is explained in Section 7.2

### 6.3 HTTP Load Test

SCION has a native web application which is a clone of *astaxie/bat*, a *cURL*-like tool implemented using Go for sending HTTP requests to web servers and retrieve information in a human-readable format. *Bat* can be used for testing, debugging, and generally interacting with HTTP servers. The scheme defaults to HTTPS – HTTP is not supported. The method defaults to GET in case there is no data to be sent and to POST otherwise [9]. The application mentioned works only via CLI, there is no support for a GUI yet. Also, the CLI tool not only works for sending single requests but rather it supports benchmarking i.e. load testing the server. In this test we will be using this benchmark mode with two important parameters: number of benchmark requests to send and number of parallel clients in benchmark. We will be requesting data in the json format from the server with varying clients and connections. Currently there is a simple web server written in Go for SCION, even though this server is not a full fledged web server we make use of the same for this test. The results are tabulated in Figure 6.5

SCION – Bat			
No of requests	No of Clients	Requests per Sec	Total time
1000	100	1233.7	0.81s
1500	200	1499.96	1s
2000	500	626.71	3.2s
2500	1000	382.4	6.53s
3000	2000	352.7	8.5s

FIGURE 6.5: SCION BAT test

Next, for testing the same over routed IP we setup Apache server and use its own benchmark tool for load testing. Like followed in previous tests, we match the same number of clients and connections for the same request of json data and collect the metrics. SCION http load testing application CLI accepts parameters in a specific format, in order to keep the fairness quotient, we tested many http load testing tools available for the IP network but apache benchmark tool was the only tool which gave us an option to correctly configure the test parameters in such a way that it would match with the SCION Bat test. The test results are tabulated below:

IP – Apache benchmark tool			
No of requests	No of Clients	Requests per Sec	Total time
1000	100	1715.72	0.583
1500	200	1815.45	0.826
2000	500	1848.72	1.082
2500	1000	1795.58	1.392
3000	2000	-	-

FIGURE 6.6: Apache benchmamrk test

Since the SCION webserver was a simple go written file and the Apache web server was a full-scale web server, the test results even though the parameters being same could differ. Also, the server supports only four routes namely: /hello - replies with a greeting, /json - replies with some JSON data, /image -replies with an image and /form - extracts POSTed form data and puts it in

the reply [9]. All these routes can be load tested, however we test only the JSON data route for the purpose of this report.

## 7 Result Comparison and Analysis

### 7.1 Latency test

In this section we compare the results of latency tests explained in previous section and analyze the same. Comparing figure 6.1 and figure 6.2 we can see that the average round-trip-time(rtt) is comparatively higher for SCION. However, the total time taken to send all the packets is similar for both ICMP and SCMP. ICMP is light weight and faster than SCION. But, SCION's SCMP provides additional functionalities which are missing in the ICMP. Every SCMP packet is authenticated unlike ICMP. The path from source to destination is visible in SCION i.e. this in turn highlights the concept of transparency of the path feature of SCION. Summarizing, SCMP has all the features of ICMP along with additional functionalities and provide more or less the same performance as ICMP.

Figure 7.1 shows the output of the path traversed from source to destination in the SCION network. The output clearly shows the hops taken by the packet to reach the destination.

```
dheeraj@scion4:~$ scmp tr -remote 19-ffaa:0:130f,[195.148.67.13]
Using path:
Hops: [19-ffaa:0:130e 1>3 19-ffaa:0:130d 4>1 19-ffaa:0:130f] MTU: 1472, NextHop: 195.148.125.33:30042
0 19-ffaa:0:130e,[195.148.125.33] IfID=1 1.736ms 850µs 1.037ms
1 19-ffaa:0:130d,[195.148.124.136] IfID=3 2.168ms 1.736ms 2.01ms
2 19-ffaa:0:130d,[195.148.124.136] IfID=4 2.026ms 1.867ms 1.678ms
3 19-ffaa:0:130f,[195.148.67.12] IfID=1 10.218ms 10.688ms 10.999ms
4 19-ffaa:0:130f,[195.148.67.13] 11.499ms 11.29ms 10.925ms
dheeraj@scion4:~$
```

FIGURE 7.1: SCION traceroute

Figure 7.2 shows the output of the path traversed from source to destination in the IP network. The command prints only a few intermediate hops and the rest is not available.

```
dheeraj@scion4:~$ traceroute 195.148.67.13 -m 15
traceroute to 195.148.67.13 (195.148.67.13), 15 hops max, 60 byte packets
 1 gw-125.research.comnet.aalto.fi (195.148.125.1) 0.448 ms 0.399 ms 0.394 ms
 2 funet-rtr.research.netlab.hut.fi (195.148.124.2) 0.606 ms 0.599 ms 0.573 ms
 3 turku1-et-0-1-0-1.ip.funet.fi (86.50.255.201) 8.469 ms 8.463 ms 8.351 ms
 4 turku2-et-0-1-0-1.ip.funet.fi (86.50.255.197) 8.599 ms 8.595 ms 8.671 ms
 5 rauma1-et-0-0-0-1.ip.funet.fi (86.50.255.211) 8.192 ms 8.173 ms 8.163 ms
 6 pori1-et-0-0-0-1.ip.funet.fi (86.50.255.209) 8.118 ms 7.910 ms 7.956 ms
 7 pori2-et-0-0-1-1.ip.funet.fi (86.50.255.207) 7.987 ms 7.923 ms 7.914 ms
 8 vaasa2-et-0-0-0-1.ip.funet.fi (86.50.255.203) 7.921 ms 7.982 ms 7.940 ms
 9 vaasa1-et-0-1-1-1.ip.funet.fi (86.50.255.174) 8.561 ms 8.582 ms 8.350 ms
10 kokkola1-et-0-0-0-1.ip.funet.fi (86.50.255.173) 8.035 ms 7.998 ms 7.969 ms
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
dheeraj@scion4:~$
```

FIGURE 7.2: IP traceroute

## 7.2 Bandwidth test

In this section we compare the bandwidth results collected from both BW tester and IPerf for SCION and routed IP respectively. Comparing figure 6.3 and 6.4 SCION BW tester performs well until 150Mbps and the performance degrades after this threshold. Whereas the IPerf performs well until 200Mbps and degrades after this. IPerf does not have any kind of intelligence to check if the BW allotted to a client is based on the number of other clients that are concurrently connected to the same server. Higher rates of BW can be achieved when only one client is accessing the sever. However, this would not be situation in real world. Therefore, SCION utilizes the SIBRA extension(explained below) to provide guaranteed BW between two hosts. The BW supported shall be agreed on not just by the end hosts but also by the nodes i.e. ASes connecting the source and destination. Currently 150Mbps is allotted to the hosts when BW test is performed and this can be verified from the AS visualization tool which allows max of 150Mbps of BW test.

SIBRA, the Scalable Internet Bandwidth Reservation Architecture, enables global bandwidth resource allocation. End hosts can use resource allocations to obtain end-to-end bandwidth guarantees to defend against DDoS attacks. SIBRA also provides scalable inter-domain resource allocations and botnet-size independence. Botnet-size independence enables two end hosts to set up communication regardless of the size of distributed botnets. SIBRA can be implemented with per-flow stateless fastpath operations on transit routers for reservation renewal, flow monitoring, and policing, which results in highly efficient data-plane operation on core routers. SIBRA also enables dynamic inter-domain leased lines (DILLs), which offer new business opportunities for ISPs. [2]

### 7.3 HTTP Load test

In this section we compare the results of the HTTP load testing performed for both IP and SCION. From the figure 6.5 and 6.6 we can see that Apache outperforms the SCION's Bat server. Apache server used is a full scale powerful server whose processing capabilities are higher than running a simple web-server through a script. Hence, these results cannot be taken seriously because of the limitation on the server end.

## 8 Repeatability of Tests

### 8.1 Test bed and Connections

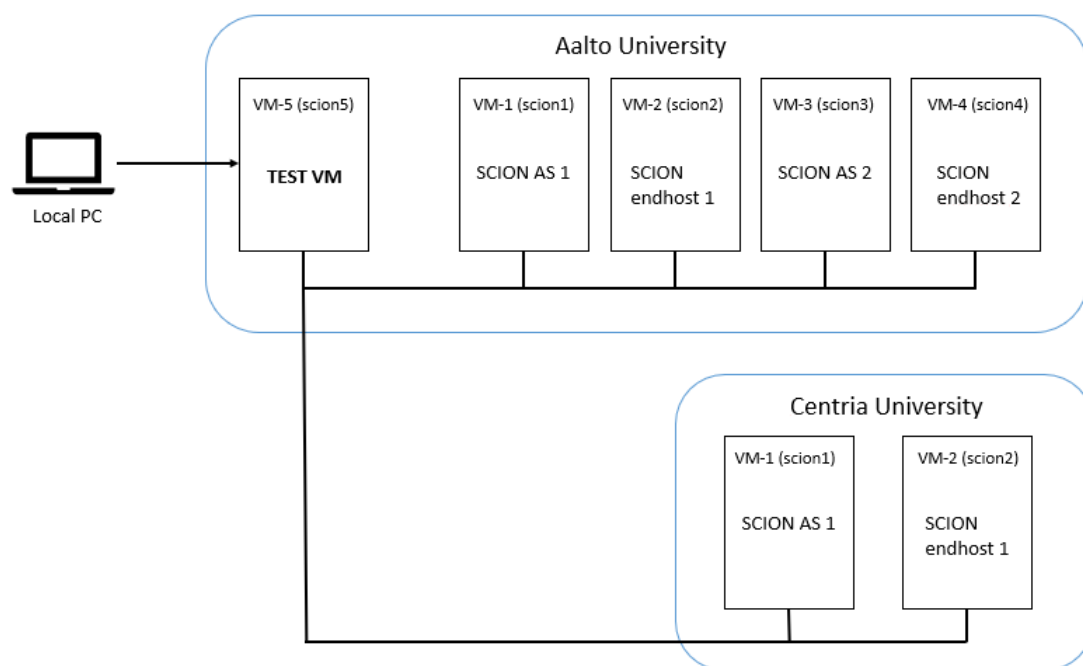


FIGURE 8.1: SCION Test bed

Figure 8.1 shows the overview and connections of all the VMs that are part of SCION research network in Finland. The robot test scripts are available on the GitLab page and also a copy is present in the test VM. To execute any tests from the test VM, access to the test VM is required. Once there is an SSH connection to the test VM, then anyone can run the test cases using robotframework's GUI "ride". Login to the test VM via the GUI is necessary to use ride.

The test scripts are written to work between any two nodes . There are two versions of each test, one meant for SCION and the other for routed IP. Each



script accepts two inputs which are the names of the two VMs, changing them to the desired name is all that is needed to run the tests. More instructions of using the robotframework and debugging are available on the GitLab page [5].

## 8.2 GitLab

The repository created in GitLab "SCION-Research-COMNET-Aalto" [5] serves as the source of all SCION related information at Aalto. The repository has multiple sub repositories inside them each covering a topic related to SCION research.

Tests performed between different nodes that were not included as part of this report are available in the Gitlab. Along with these there have been tests to verify SCION applications like AS visualization and SCION features like multi-path communication. The results of these tests and documentation of the same is available in the repository. It also contains instructions of setting up services on SCION like SIG and provide debugging information, there is a FAQ section where we have collected some issues that we faced during the tests and have provided solutions to the same. Also, the repository has a section dedicated for all the documents that are collected or written as part of SCION.

## 9 Conclusion and Future Work

SCION introduces quite interesting new features compared to the current Internet. In terms of security and reliability these features are better performing. Failure isolation and bandwidth guarantee features do not just serve as enhancements but they also provide security against DDoS, botnet and prefix hijacking etc. Performance is of utmost importance to both end user and the ISP provider. In this report we try to execute few performance related tests over the SCION network and analyse the results. We go one step further by executing the same performance tests over the current Internet i.e. routed IP and compare the results with the SCION. In order to keep the comparison fair we try to match all the test parameters of SCION and IP.

The comparison results look promising as SCION performed quite well with respect to the routed IP. In tests where the performance dropped compared to the IP was due to either design or application limitations of SCION. Another point to note is that the whole SCION network was setup and run over virtual machines, we believe this would impact the performance.

The same set of tests can be performed with hosts outside EU domain or an endhost situated outside Finland. The robot script would come in handy running any kinds of performance tests across the SCIONlab network. It would be interesting to observe the results of the tests performed with an endhost outside EU domain.

Apart from the tests mentioned in this report, we have planned to carry out different tests over the same nodes. For example, performance testing over

SIG, running some service(website) over SCION and measure its performance etc. All the tests mentioned here run over the nodes in Finland. Running performance tests between nodes of different domains, country and location can be very interesting and important to evaluate SCION.

SCION can be best suited for specialised networks where security is of utmost importance, where attacks such as DDOS or spoofing need to be detected and blocked actively. Enterprises can run their services over SCION which can provide better security and control compared to the current Internet. A typical example would be of the Financial and Banking services having Business-to-Business or Business-to-Consumer use cases [3]. Reliability, failure isolation are features that can be of benefit to many other use cases, which in turn points to more research on SCION.

# Bibliography

- [1] E. Z. Network Security Group. [Online]. Available: <https://www.scion-architecture.net/>
- [2] R. M. R. L. C. Adrian Perrig, Pawel Szalachowski, *SCION: A Secure Internet Architecture*, 2017.
- [3] A. Perrig, *SCION Overview*.
- [4] E. Z. Network Security Group. [Online]. Available: <https://docs.scionlab.org/>
- [5] S. R. COMNET-Aalto. [Online]. Available: <https://version.aalto.fi/gitlab/scion-research-comnet-aalto>
- [6] A. P. R. M. R. DAVID BARRERA, LAURENT CHUAT and P. SZALACHOWSKI, "The scion internet architecture," vol. 60, 2017.
- [7] E. Z. Network Security Group. [Online]. Available: <https://docs.scionlab.org/content/apps/>
- [8] ——. [Online]. Available: <https://docs.scionlab.org/content/apps/bwtester.html>
- [9] ——. [Online]. Available: <https://docs.scionlab.org/content/apps/bat.html>
- [10] ——. [Online]. Available: [https://docs.scionlab.org/content/apps/remote\\_sig.html](https://docs.scionlab.org/content/apps/remote_sig.html)

# A Appendix

## Latency robot scripts

\*\*\* Settings \*\*\*

Documentation      This test calculates the latency between the given  
two entities using both SCMP and ICMP.

...                      Additionally it logs the output to a an output file

Suite Setup          Latency Test Setup

Resource             resources/latency\_resource.robot

\*\*\* Variables \*\*\*

\${VM\_1}                aalto\_scion\_endhost2

\${VM\_2}                centria\_scion\_endhost1

@{Number\_Of\_Packets}    5      10      15      20

\*\*\* Test Cases \*\*\*

Send SCMP And ICMP Requests For Given Number Of Packets

Remove File          \${OUTPUT\_DIR}/latency\_log.txt

FOR      \${item}      IN      @{Number\_Of\_Packets}

    \${result}=      Run      \${RUN\_FILE} \${SSH\_USER} \${\${VM\_1} ['eth0\_ip']}

    'scmp echo -remote \${\${VM\_2} ['scion\_ip']} -c \${item}'

    \${lines}=      Get Lines Containing String      \${result}      packet loss

Append To Log File      latency\_log      SCMP: \${lines}

    \${result2}=      Run      \${RUN\_FILE} \${SSH\_USER} \${\${VM\_1} ['eth0\_ip']}

```

'ping ${${VM_2} ['eth0_ip']} -c ${item} -s ${ICMP_PACKET_SIZE}'
${lines}=    Get Lines Containing String    ${result2}    packet loss
Append To Log File    latency_log    ICMP: ${lines}

END

```

### BW IPerf robot scripts

\*\*\* Settings \*\*\*

```

Resource          resources/bandwidth_resource.robot
Suite Setup       IPERF BW Test Setup
Suite Teardown    IPERF Bandwidth Test Teardown

```

Documentation

... Bandwidth test of IP routed paths using IPERF3

\*\*\* Variables \*\*\*

```

${BW_SERVER}      aalto_scion_endhost2
${BW_CLIENT}      centria_scion_endhost1
${TIME_IN_SEC}    5
${PACKET_SIZE}    1400
@{Attempted_Bandwidth}    1M    10M    50M    100M    200M    500M    1G

```

\*\*\* Test Cases \*\*\*

Send Packets From Client To Server With Given Bandwidth

```

Remove File    ${OUTPUT_DIR}/iperf_log.txt

FOR    ${item}    IN    @{Attempted_Bandwidth}
    ${result}=    Run    ${RUN_FILE}    ${SSH_USER}    ${${BW_CLIENT} ['eth0_ip']}
    'iperf3 -c ${${BW_SERVER} ['eth0_ip']} -p 9090 -f m -u -b ${item}
    -t ${TIME_IN_SEC}'
Append To Log File    iperf_log    IPERF test for ${item}bps

```

```

    ${lines}=    Get Lines Containing String    ${result}    sender
    Append To Log File    iperf_log    ${lines}
    ${lines}=    Get Lines Containing String    ${result}    receiver
    Append To Log File    iperf_log    ${lines}

END

Log To Console    Log file iperf_log.txt with result of the run is
generated in the output directory

```

### BW SCION robot scripts

#### \*\*\* Settings \*\*\*

```

Documentation    Bandwidth test of SCION using SCION's native BW tester
tool for different values
Suite Setup      SCION BW Test Setup
Suite Teardown  SCION Bandwidth Test Teardown
Resource        resources/bandwidth_resource.robot

```

#### \*\*\* Variables \*\*\*

```

${BW_SERVER}          aalto_scion_endhost2
${BW_CLIENT}          centria_scion_endhost1
${PACKET_SIZE}        ?
${NUM_OF_PACKETS}     ?
${TIME_IN_SEC}        10
@{Attempted_Bandwidth}  1Mbps    10Mbps    50Mbps    100Mbps    200Mbps
500Mbps    1Gbps    10Gbps

```

#### \*\*\* Test Cases \*\*\*

Send Packets To And From Client And Server With Given Bandwidth

```

Remove File    ${OUTPUT_DIR}/bandwidth_log.txt
FOR    ${item}    IN    @{Attempted_Bandwidth}

```

```

    ${result}=      Run
...      ${RUN_FILE} ${SSH_USER} ${${BW_CLIENT} ['eth0_ip']}
'scion-bwtestclient -s ${${BW_SERVER} ['scion_ip']}:30200 -cs
${TIME_IN_SEC},${PACKET_SIZE},${NUM_OF_PACKETS},${item}'
${lines}=      Get Lines Containing String      ${result}
bandwidth
Append To Log File      bandwidth_log      ${lines}
${lines}=      Get Lines Containing String      ${result}
Loss rate
Append To Log File      bandwidth_log      ${lines}
END
Log To Console      Log file bandwidth_log.txt with result of
the run is generated in the output directory

```

### cURL for IP

#### \*\*\* Settings \*\*\*

```

Documentation      Https tests
Library            OperatingSystem
Library            Process
Library            String

```

#### \*\*\* Variables \*\*\*

```

${Http_SERVER_IP}      195.148.124.136
${Http_CLIENT_IP}      195.148.125.34
@{NUMBER_OF_REQUESTS}  1000    5000    10000    15000
20000
@{NUMBER_OF_PARA_CLIENTS}  100    500    1000    1500    2000
${HTTP_REQUEST}      dog.jpg      # can take value json
or image(dog.jpg)

```



```
#{SSH_USER}                dheeraj
```

```
*** Keywords ***
```

```
Send Https Requests
```

```
[Arguments]    #{number_of_requests}
FOR    #{item}    IN    @{NUMBER_OF_PARA_CLIENTS}
    #{result}=    run process
    ...    #{CURDIR}/run_file.sh    #{SSH_USER}    #{Http_CLIENT_IP}
    ab -n #{number_of_requests} -c #{item}
    https://#{Http_SERVER_IP}/#{HTTP_REQUEST}
    append to file    #{CURDIR}/http_log.txt    Https Test for
    #{number_of_requests} Number of requests with #{item}
    number of parallel clients${\n}
    append to file    #{CURDIR}/http_log.txt    #{result.stdout}${\n}
    sleep    5s
END
```

```
*** Test Cases ***
```

```
Testcase Setup
```

```
# Below two lines checks if the Bwtester app is installed on the
AS node
#{result}=    run    #{CURDIR}/run_file.sh    #{SSH_USER}
#{Http_SERVER_IP} 'sudo systemctl status apache2'
should contain any    #{result}    active (running)
Apache server is not running
```

```
Send Https Requests From Client To Server
```

```
remove file    #{CURDIR}/http_log.txt
FOR    #{num}    IN    @{NUMBER_OF_REQUESTS}
```

```

        Send Https Requests    ${num}
    END
    log to console    Log file http_log.txt with result of
    the run is generated in the current directory

```

## BAT for SCION

### \*\*\* Settings \*\*\*

```

Documentation    Https tests
Library          OperatingSystem
Library          Process
Library          String
Suite Teardown  Terminate The Bat Test Server Process

```

### \*\*\* Variables \*\*\*

```

${Bat_SERVER_IP}          195.148.124.136
${Bat_SERVER_SCION_IP}    19-ffaa:0:130d, [195.148.124.136]
${Bat_CLIENT_IP}          195.148.125.34
${Bat_CLIENT_SCION_IP}    19-ffaa:0:130e, [195.148.125.34]
@{NUMBER_OF_REQUESTS}    1000    5000    10000    15000    20000
@{NUMBER_OF_PARA_CLIENTS} 100    500    1000    1500    2000
${HTTP_REQUEST}          image    # can take value json or image
${SSH_USER}              dheeraj

```

### \*\*\* Keywords \*\*\*

#### Terminate The Bat Test Server Process

```

    ${pid}=    run    ${CURDIR}/run_file.sh ${SSH_USER} ${Bat_SERVER_IP}
    'ps -C go -o pid='
    run    ${CURDIR}/run_file.sh ${SSH_USER} ${Bat_SERVER_IP}
    'kill -9 ${pid}'

```

```

# restarting scion services as running http server creates
some problems
run    ${CURDIR}/run_file.sh ${SSH_USER} ${Bat_SERVER_IP}
'sudo systemctl restart scionlab.target'
run    ${CURDIR}/run_file.sh ${SSH_USER} ${Bat_SERVER_IP}
'sudo systemctl restart scionlab.target'
run    ${CURDIR}/run_file.sh ${SSH_USER} ${Bat_SERVER_IP}
'sudo systemctl restart scionlab.target'

```

### Send Https Requests

```

[Arguments]    ${number_of_requests}
FOR    ${item}    IN    @{NUMBER_OF_PARA_CLIENTS}
    ${result}=    run process
    ...    ${CURDIR}/run_file.sh    ${SSH_USER}
    ${Bat_CLIENT_IP}    scion-bat
    -b -b.N ${number_of_requests} -b.C ${item}
    https://${Bat_SERVER_SCION_IP}:8080/${HTTP_REQUEST}
    append to file    ${CURDIR}/bat_log.txt
    Bat Test for ${number_of_requests} Number of
    requests with ${item} number of parallel clients${\n}
    append to file    ${CURDIR}/bat_log.txt
    ${result.stdout}${\n}
END

```

\*\*\* Test Cases \*\*\*

#### Testcase Setup

```

# Below two lines checks if the Bwtester app
is installed on the AS node
${result}=    run    ${CURDIR}/run_file.sh

```

```

${SSH_USER} ${Bat_SERVER_IP} 'sudo apt list scion-apps-bat'
should contain any    ${result}    scion-apps-bat
scion-apps-bat is Not Installed
${result}=    run    ${CURDIR}/run_file.sh
${SSH_USER} ${Bat_SERVER_IP}
'ls /home/${SSH_USER}/scion-apps/_examples/shttp/server'
should contain any    ${result}    main.go
scion-apps-repository is Not Installed
# Below steps starts the Batester app in
server mode on the AS node
${result}=    start process
...    ${CURDIR}/run_file.sh    ${SSH_USER}
${Bat_SERVER_IP}
cd /home/${SSH_USER}/scion-apps/_examples/shttp/server;
go run main.go -p 8080    shell=True
log    ${result}
Sleep    10s

```

Send Https Requests From Client To Server

```

remove file    ${CURDIR}/bat_log.txt
FOR    ${num}    IN    @{NUMBER_OF_REQUESTS}
    Send Https Requests    ${num}
END
log to console    Log file bat_log.txt with result of
the run is generated in the current directory

```