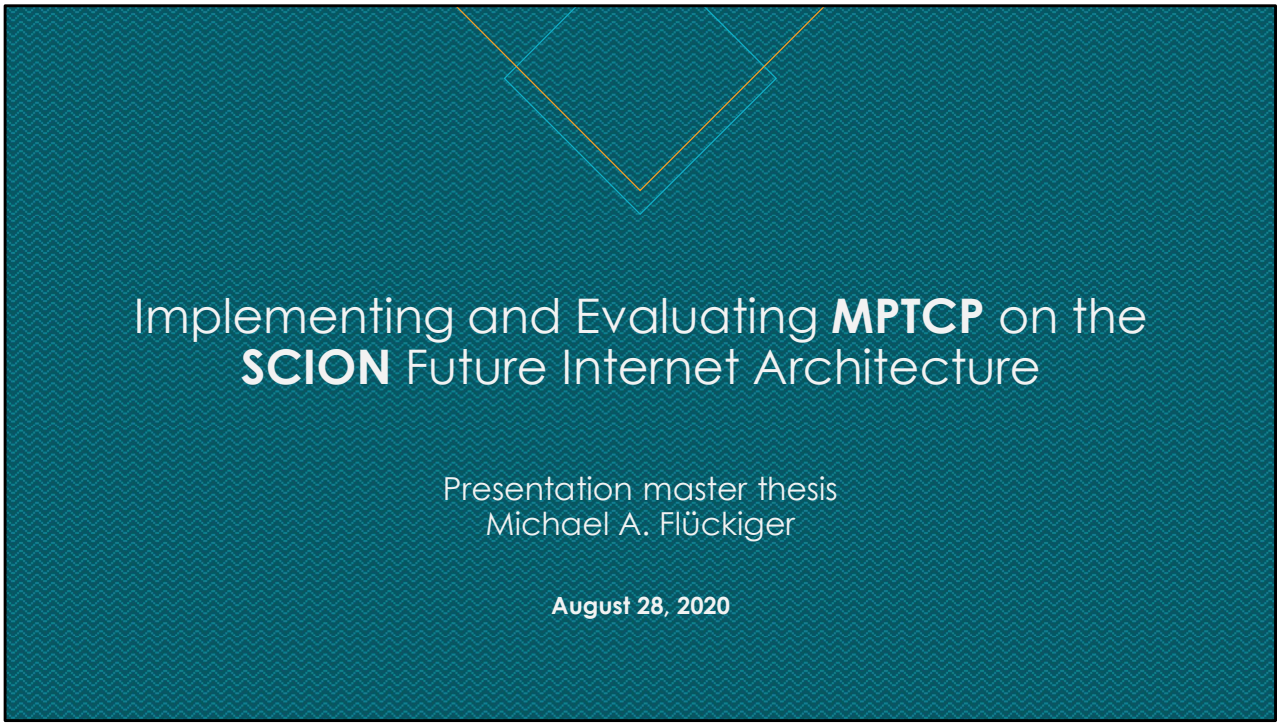


Welcome again

Happy that I today can present the work I did during my master thesis.

On the menu

- Introduction
- Functionality of Shila
- Performance evaluation
- Future work

2

Let us start with an overview what I'm going to cover in the next 30 minutes:

Start with the Introduction, where I will cover questions like:
- What are the main ingredients of my work?
- What was the goal of my work?
- And why do we want to reach that goal?

During the Introduction I will furthermore introduce you to Shila – she plays an important role in my thesis.

After the Introduction I will then explain the functionality of Shila based on an example.

After that I will present some results of the performance evaluation I did and conclude the presentation with some references to future work.

On the menu

- Introduction
- Functionality of Shila
- Performance evaluation
- Future work

3

So let's start with the Introduction...

Introduction

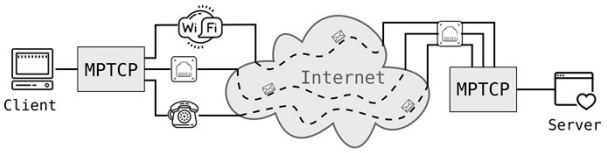
Ingredients?

TCP (Transmission Control Protocol)

- Dominant transport protocol in today's Internet
- One path per connection

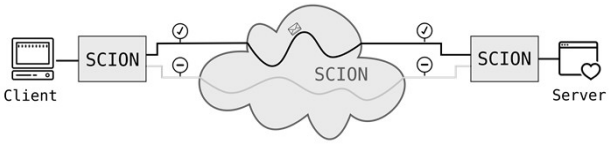
MPTCP (Multipath TCP)

- Extension to TCP
- Multiple paths (flows) per connection



SCION (Scalability, Control and Isolation on next-generation networks)

- More secure network architecture
- Implements path transparency
- Support for multiple paths



4

As the title of the work suggests, it has two main ingredients: **MPTCP** and **SCION**.

On the one hand Multipath TCP, an extension to TCP, the famous transport protocol which I'm sure you all know.

In contrast to TCP supports MPTCP not one, but multiple paths (or flows) for a single connection.
As illustrated here in this Figure, for a single connection between client and server, it creates a separate flow for every available network interface.

On the other hand SCION: The successor of today's Internet (hopefully ☺)

Im sure you all known this architecture too good, I therefore just to mention two important points here:

- SCION implements path transparency; this is in contrast to MPTCP where the sender and the receiver have no idea about the
  path their data takes through the network. In the illustration here this fact is illustrated with the dashed for MPTCP here, and the solid line for SCION.

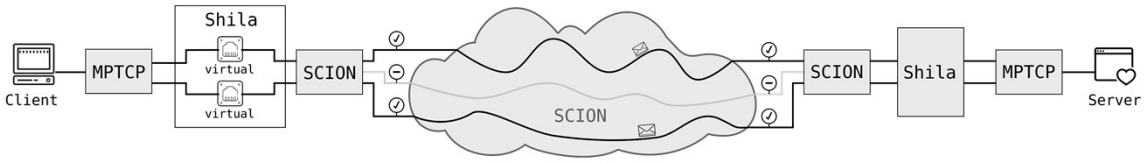
- The second point important for the presented work is the fact that SCION inherently provides support for multiple paths.

Introduction

What?

Main objective and contribution of the presented work:

Implementation (and evaluation) of a shim layer that allows the usage of MPTCP over SCION.



5

Knowning about our main ingredients, we now can talk about the goal of this work:

The main objective and cotribution of the presented work was the implementation and the evaluation of a shim
layer that allows the usage of MPTCP over SCION.

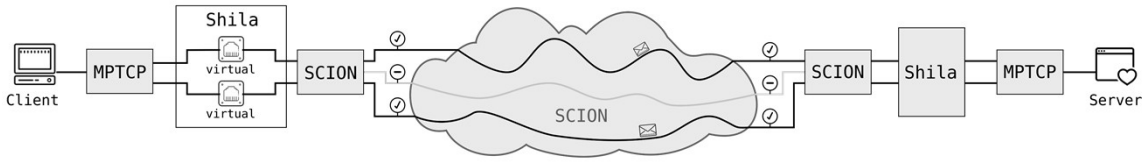
I gave the implementation the name Shila, short for shim layer.

Introduction

What?

Main objective and contribution of the presented work:

Implementation (and evaluation) of a Shim layer that allows the usage of MPTCP over SCION.



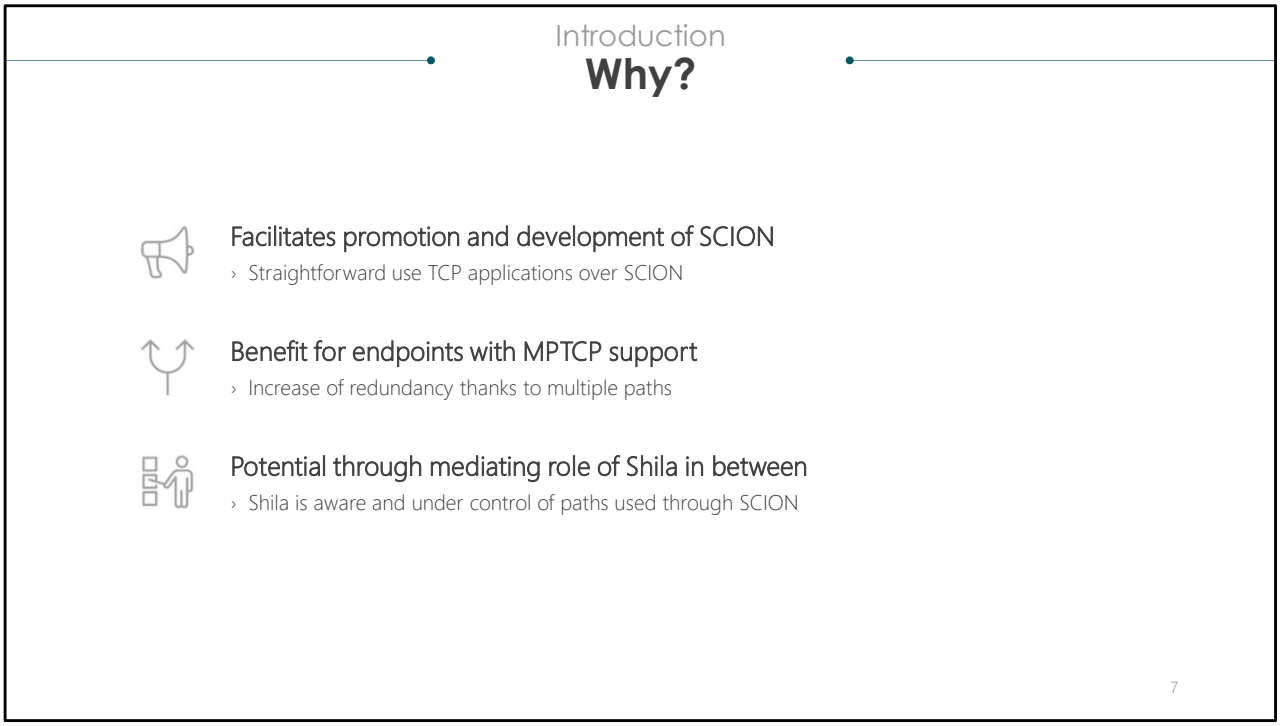
6

This image here shows why Shila deserves her name. She is placed right in between MPTCP and SCION, where she fulfills her mediating role between the two technlogies.

Strongly simplified works the shim layer as follows:

Shila creates virtual network interfaces and monitors them. A TCP client application uses one of these virtual interfaces for connection.
Shila intercepts the MPTCP connection request and forwards it over the SCION network to the Shila instance at the receiving site.
There the request is handed over back to MPTCP. From there its pushed forwarded to the TCP server application.

As I said, I will explain the functionalty in more detail soon, but before that I want quickly to talk about why one is even interested in MPTCP over SCION.



A first reason is simple but not less important.:
MPTCP facilitates the promotion and development of SCION. With Shila one can use its favorite TCP application without having
to change anything at its implementation. This lowers the hurdles to try out SCION..

Another reason is that applications benefit from MPTCP and the inherent support of SCION for multiple paths.
A connection with MPTCP over SCION benefits from redundancy through the multiple paths, even if there is just one real interface available
(at least from path redundancy, not from connection redundancy)

And as a last point I want to mention here: There is a lot of potential given through the mediating role of Shila in between. As an example:
MPTCP has a special congestion control which makes sure that the MPTCP connection does not disadvantage other
regular TCP connections in the network. This would be the case if multiple flows of a MPTCP connection share the same link.
Since SCION is implementing path transparency, Shila knows about the paths of the individual flows of a MPTCP

connection and can make sure that all this flows are distinct. Then there is no need to use a conservative congestion control,
Shila can guarantee that the flows share no links.

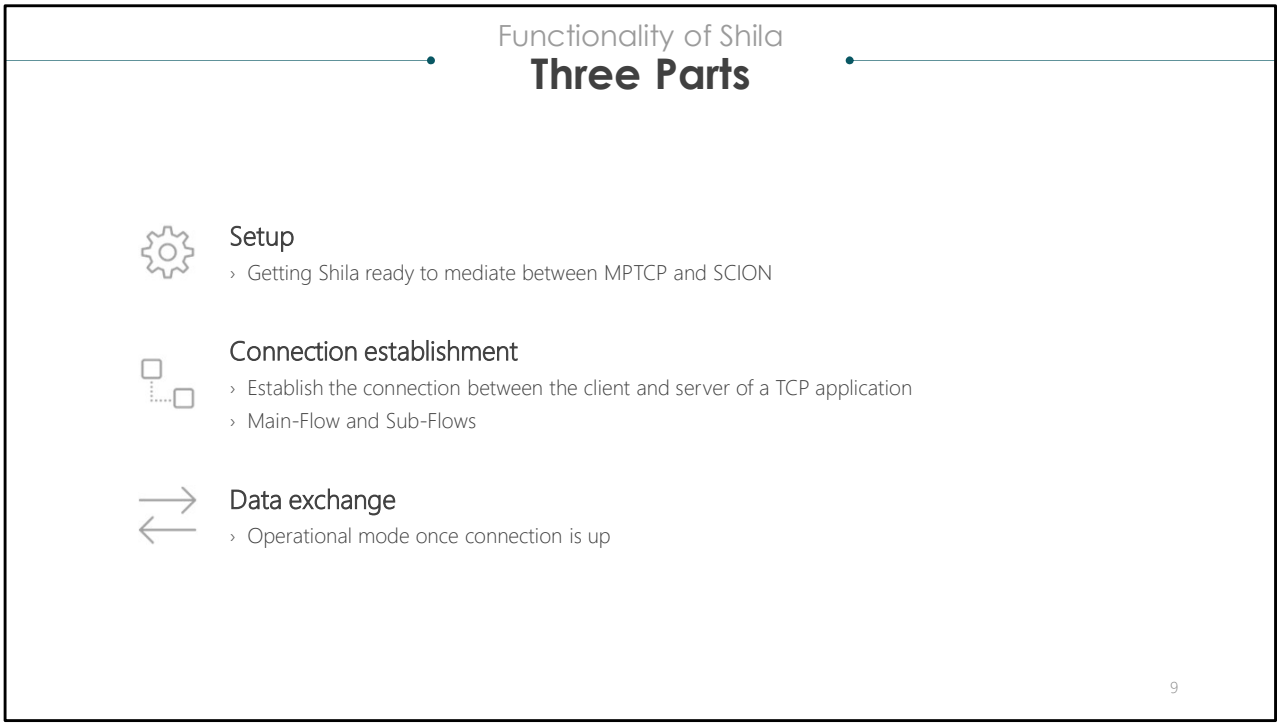
This as one example of the potential having Shila in this mediating role in between.

On the menu

- Introduction
- Functionality of Shila
- Performance evaluation
- Future work

8

So with this reasons in mind, its time to discuss the functionality of Shila.



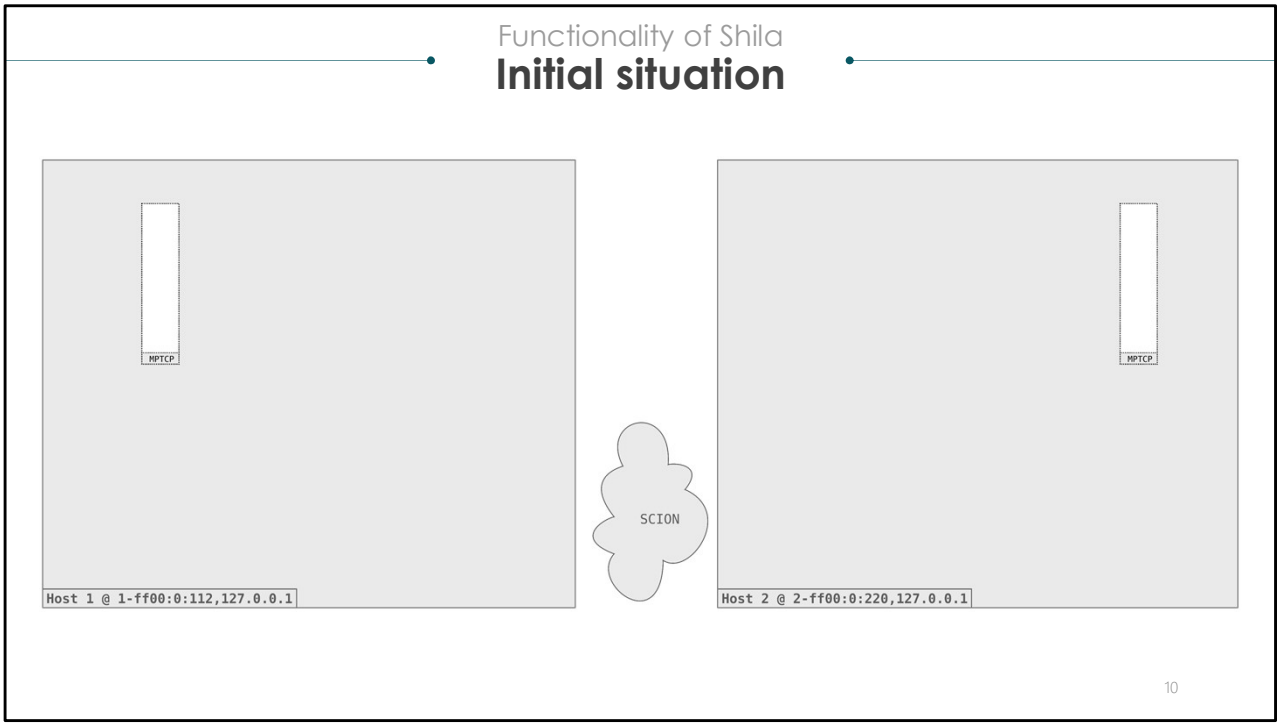
Functionality of Shila can be divided into three parts:

Setup: Shila is getting ready to mediate between MPTCP and SCION

Connection establishment:
Stage where the client of a TCP application reaches out to the server of a TCP application
It consists in the first step of a Main-Flow establishment, so the establishment of the initial connection, and then
of one or multiple Sub-Flow establishments.

Once the connection is established, Shila works in “operational mode”, rerouting the data exchanged.

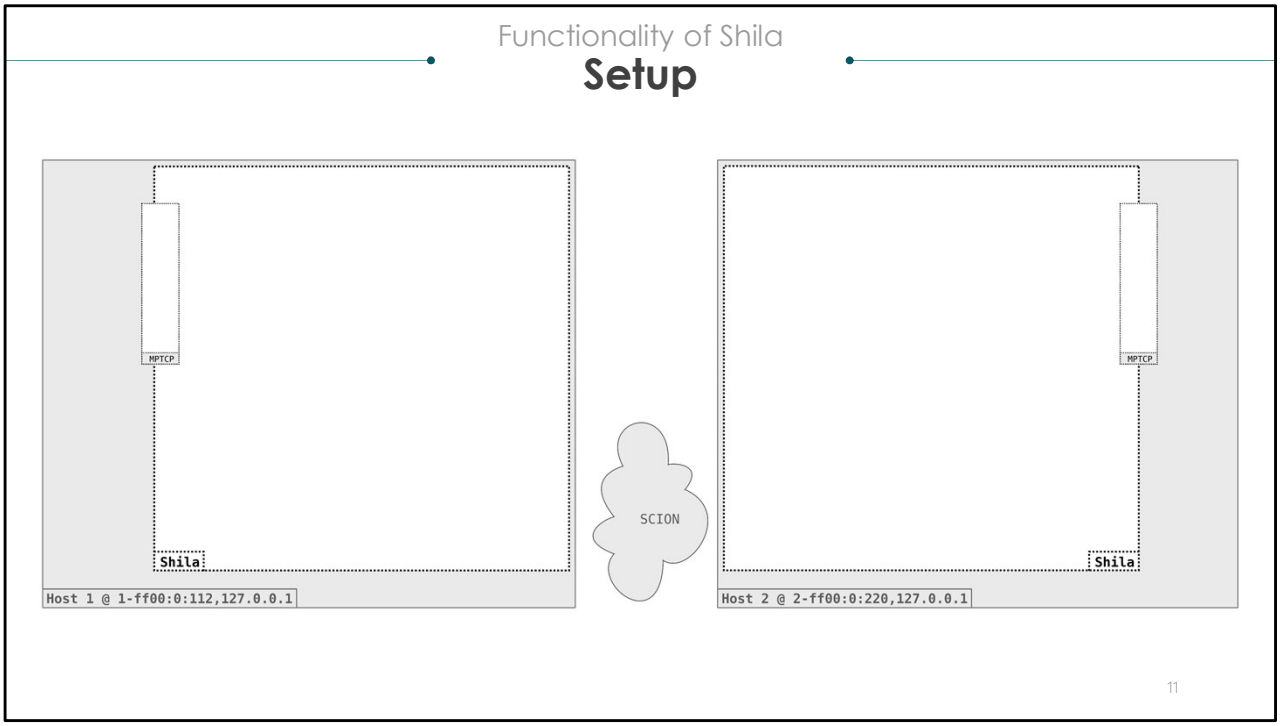
I will now illustrate this three parts based on a running example..



What we have here is the initial situation:

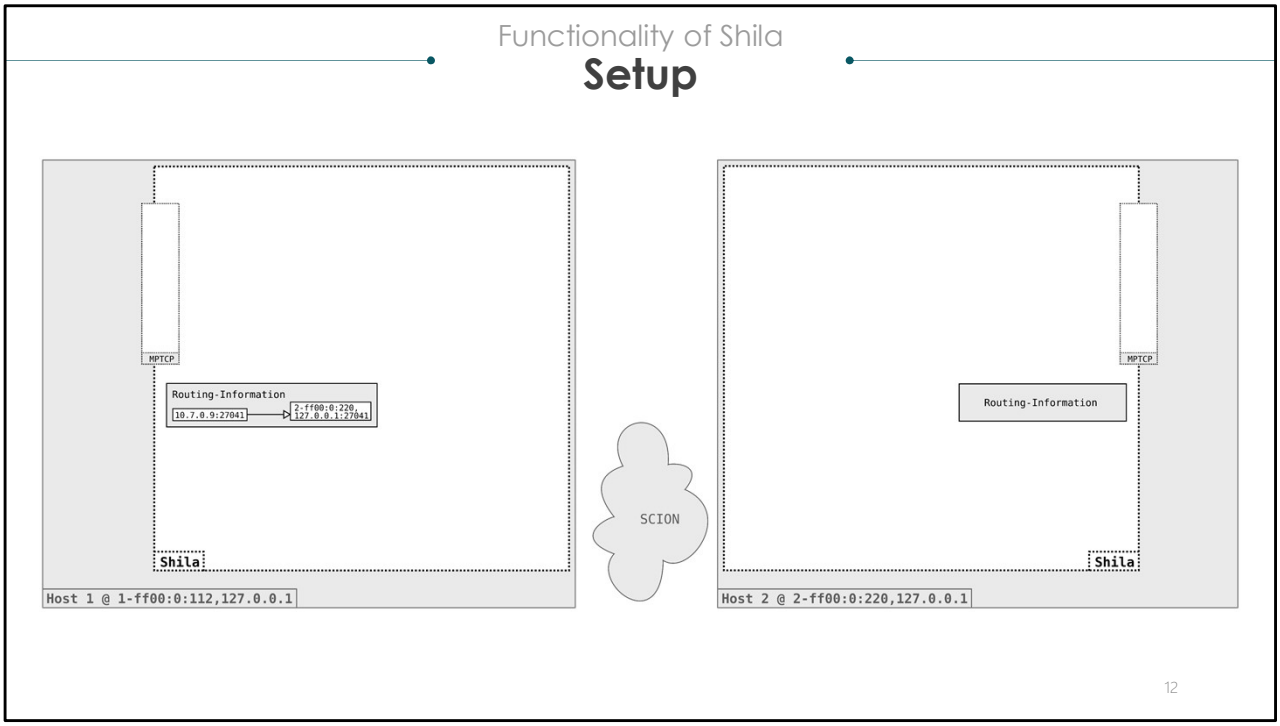
- Two Hosts, Host 1 and Host 2
- Connected to the SCION network
- Both with support for MPTCP

The example we want to cover is a TCP Client on Host 1 connecting to a TCP Server on Host 2.



Start with the setup (Shila is getting ready to mediate between MPTCP and SCION)

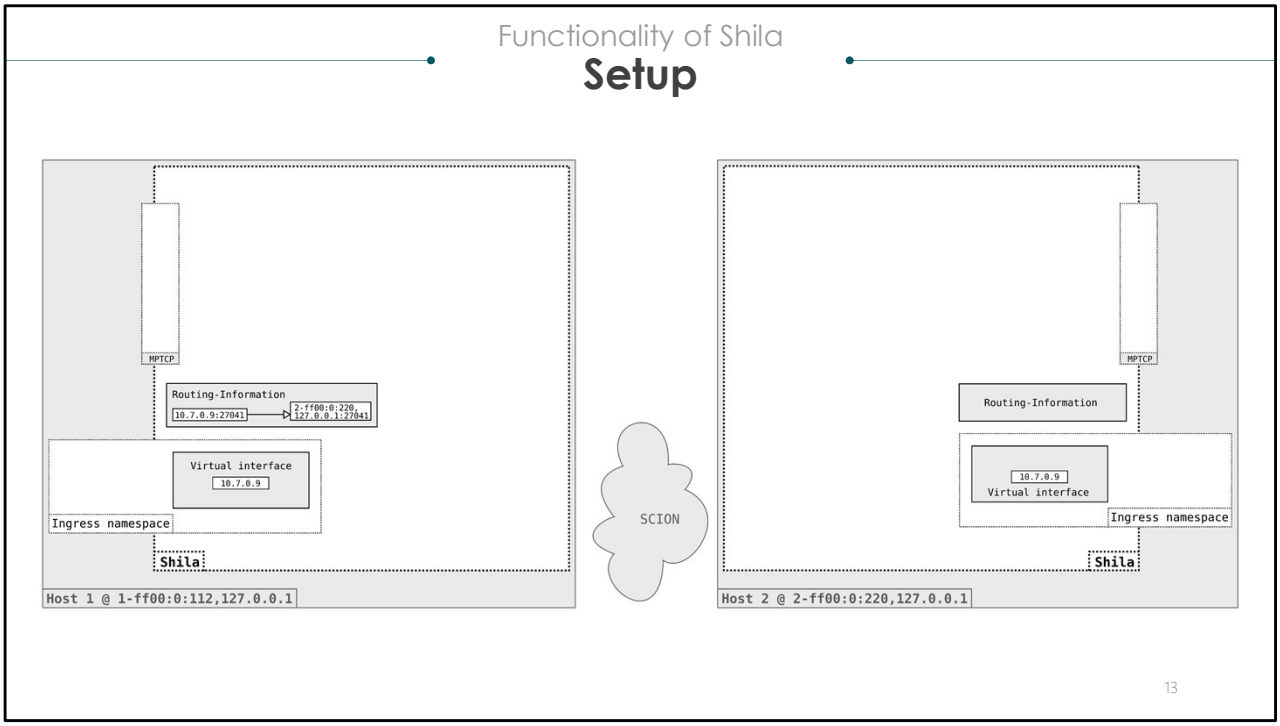
We start Shila on both Hosts. Note that the setup is the same on both, it is sufficient to concentrate
just on the left instance.



Routing Information is loaded

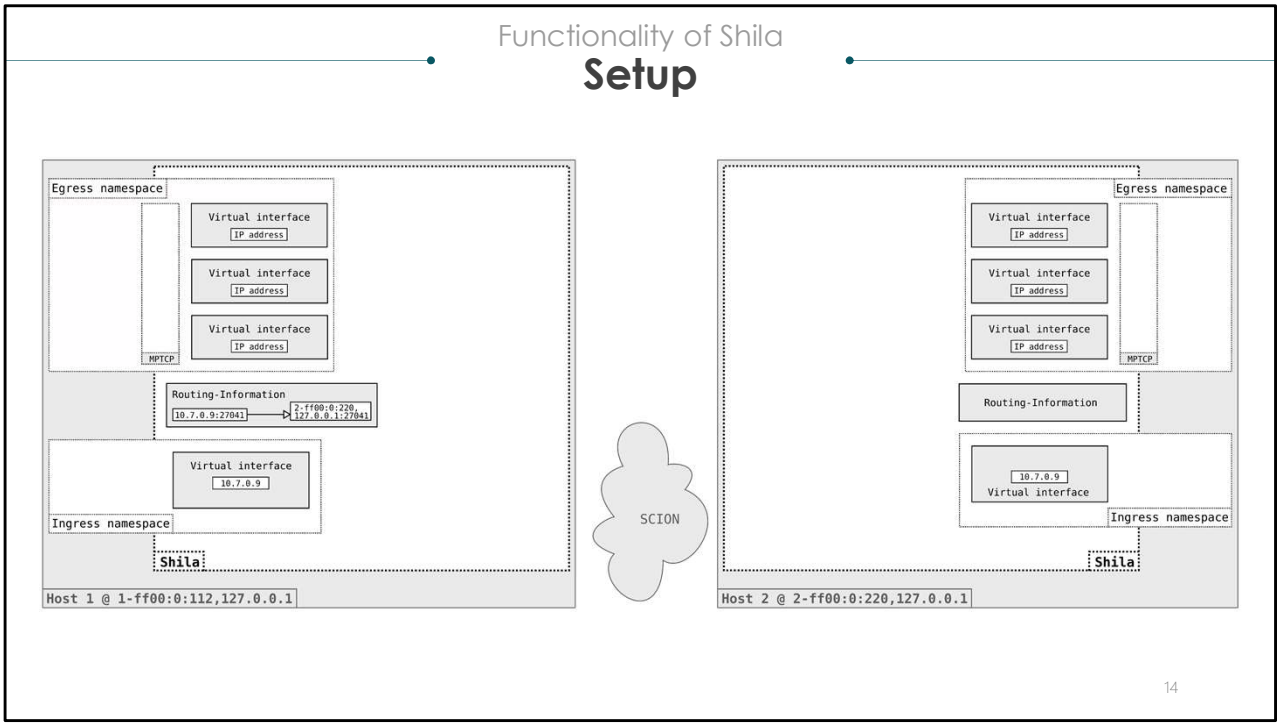
Contains the mapping from TCP destination address to SCION destination address.

We will later see how this is mapping is used and why it is necessary.



Allocation of default ingress namespace

Creation of a single virtual interface with a default IP address.

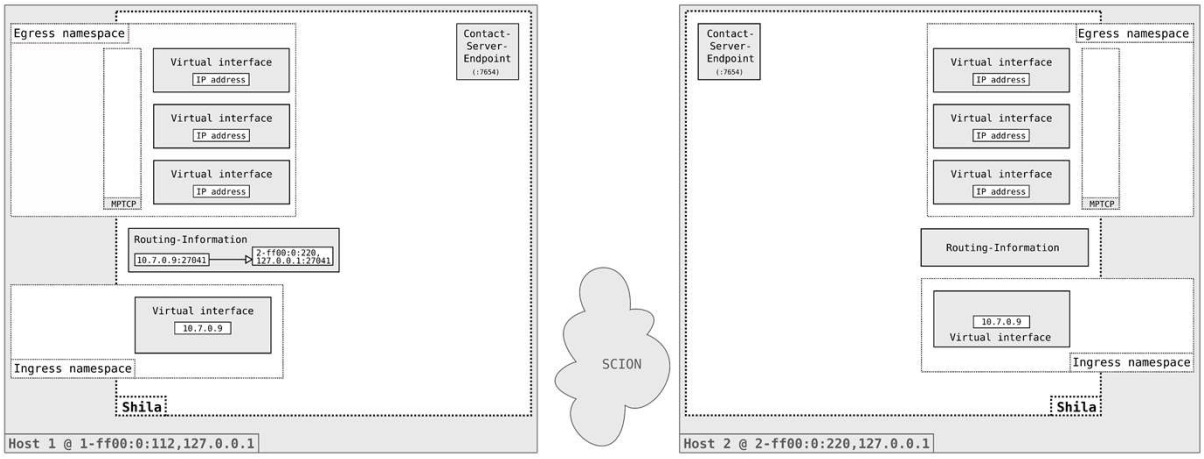


Then the egress namespace is allocated

In there, the default number of three virtual interfaces is created. This number can be chosen by the user.

The number of virtual network interfaces created corresponds later to the number of flows used for a single MPTCP connection.
So if for example just one interface is created, then we have just a simple TCP connection.
With three interfaces as we have here, we will have one Main-Flow and two Sub-Flows in the end.

Setup



15

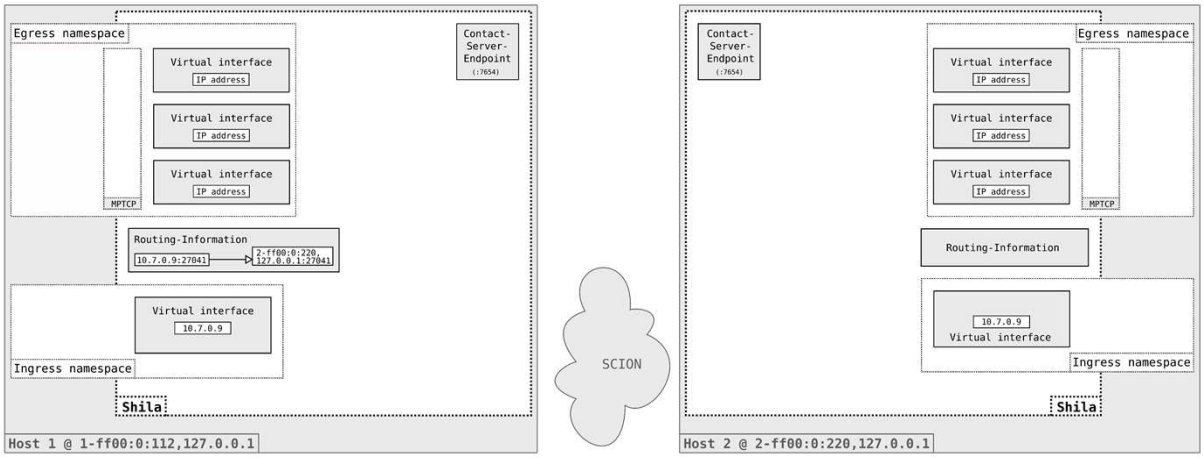
As a last step of the setup, Shila creates the Contact-Server-Endpoint listening per default on the SCION port 7654.

This endpoint, as we will see soon, serves as an initial point of contact for Shila instances running on different hosts.

This concludes the setup part, Shila is now ready to receive and mediate traffic.

Functionality of Shila

Main-Flow Establishment

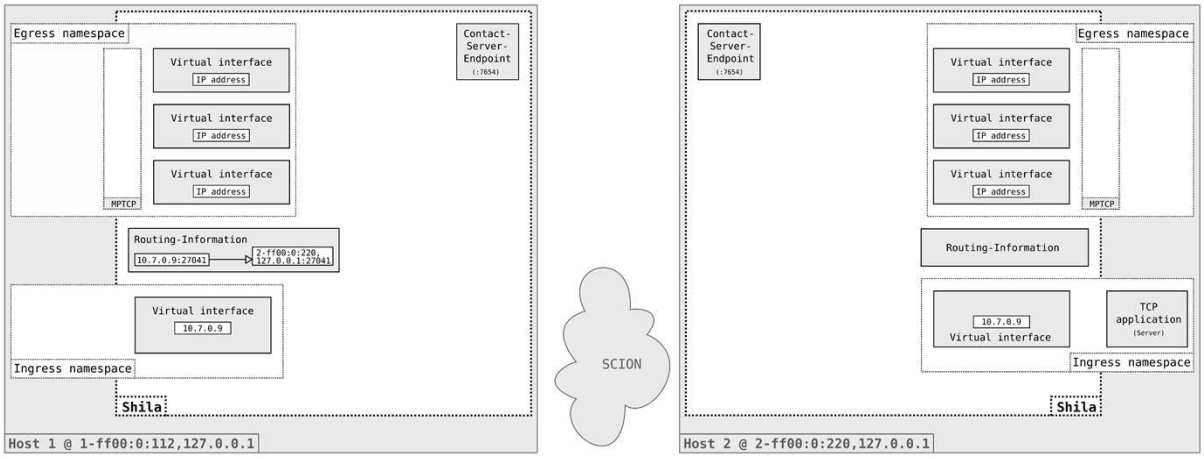


16

We will now step through the Main-Flow establishment. As I said, this is the establishment of the first of possible
multiple flows of a single MPTCP connection between two endpoints.

Functionality of Shila

Main-Flow Establishment

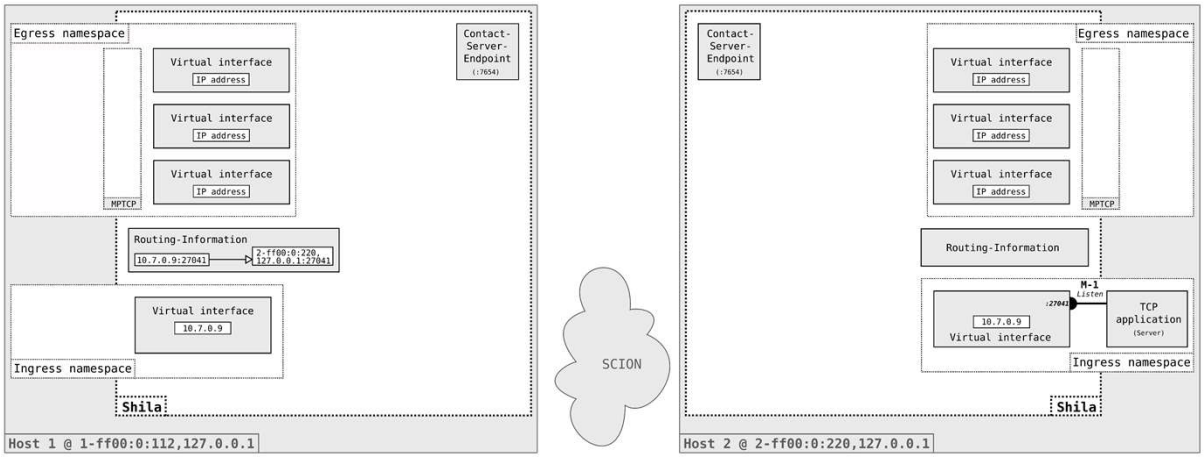


17

Start of the TCP Server application in the ingress namespace of the Shila instance on Host 2.

Functionality of Shila

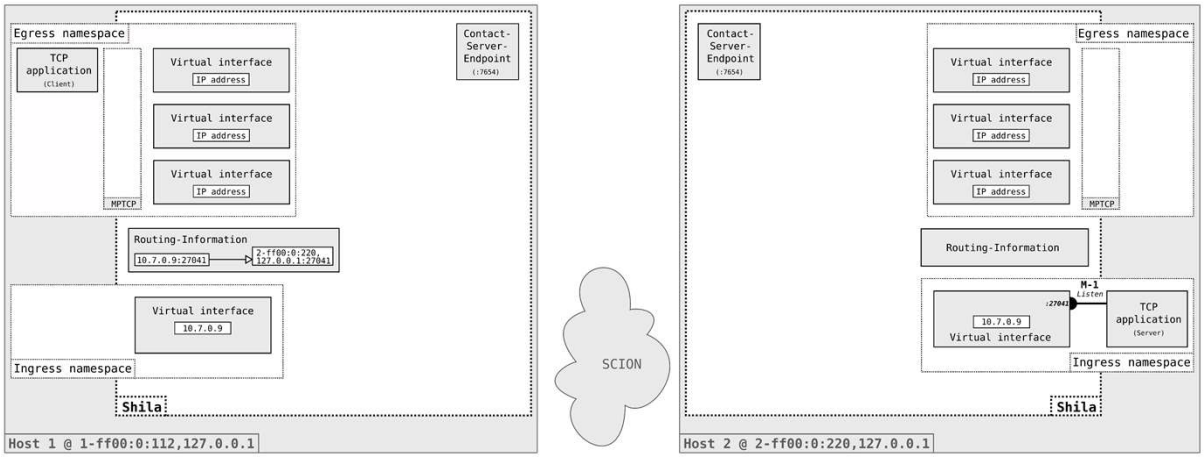
Main-Flow Establishment



Binds to the virtual interface on, let's say, port 27041. This step is not recognized by Shila.

Functionality of Shila

Main-Flow Establishment

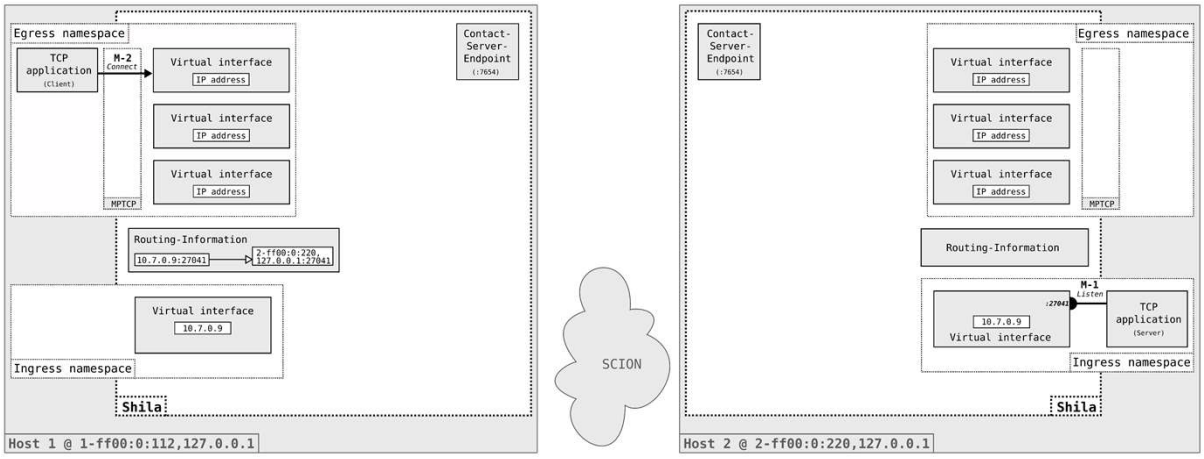


19

The TCP Client instance is started in the egress namespace of the Shila instance on Host 1.

Functionality of Shila

Main-Flow Establishment

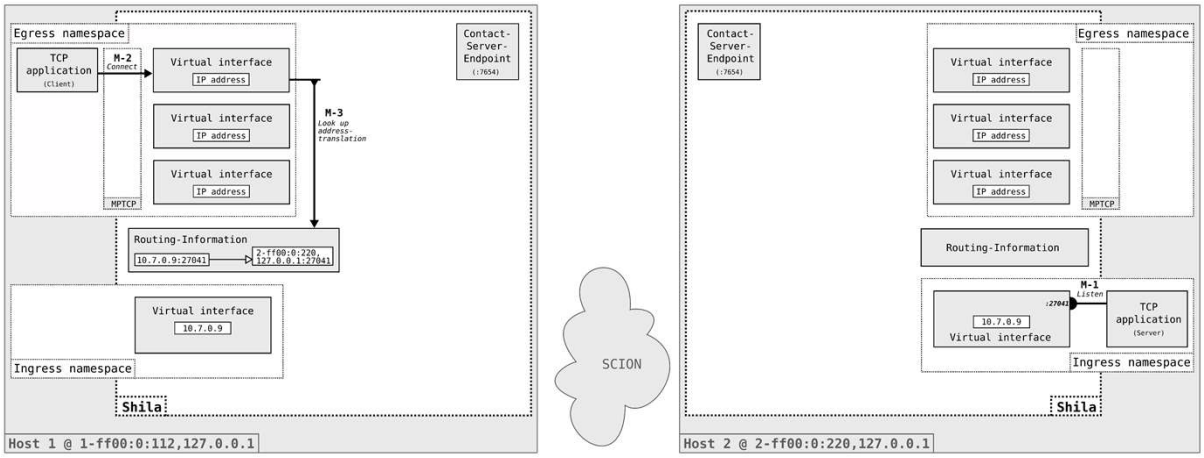


It binds a socket to one of the virtual interfaces.

As soon as connected a very first datagram containing the SYN is sent out by the application and intercepted by Shila.

Functionality of Shila

Main-Flow Establishment



21

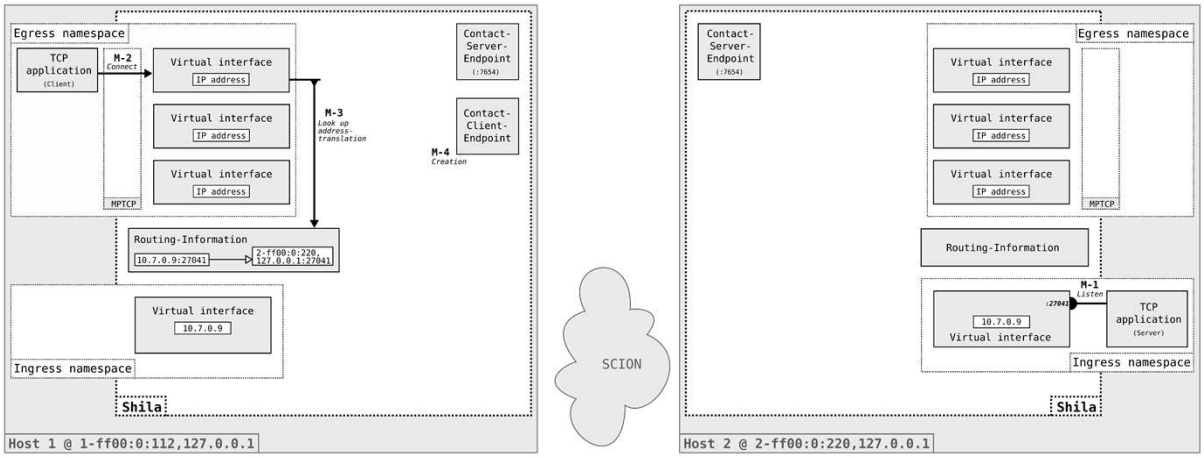
Shila parses this datagram and extracts its destination address, namely 10.7.0.9. (This is the IP address assigned to the virtual interface in ingress namespace of Host 2.)

Using the destination TCP address as a key, it extracts the destination SCION address from the Routing-Information.

For our example, this is the SCION address of Host 2.

Functionality of Shila

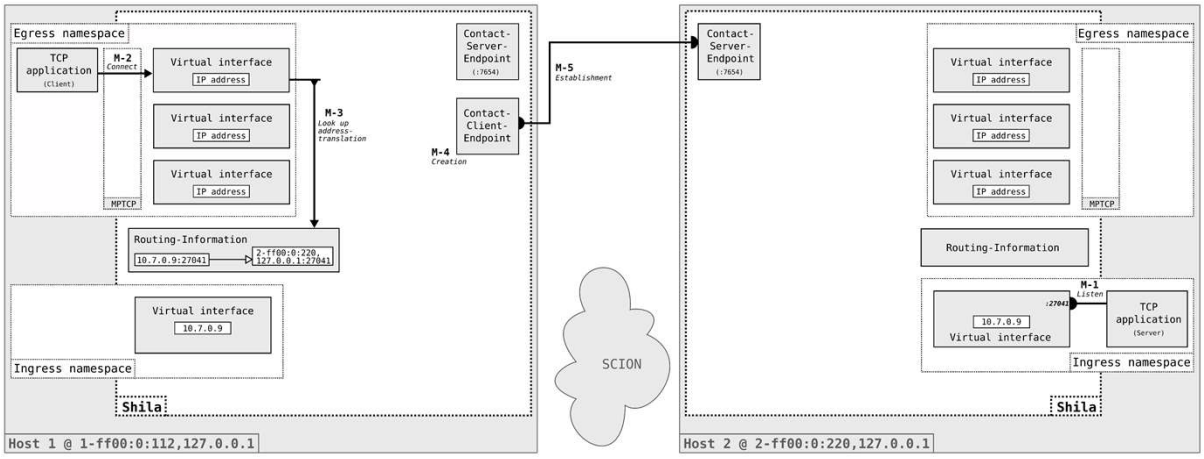
Main-Flow Establishment



With this information at hand, it creates a so-called Contact-Client-Endpoint…

Functionality of Shila

Main-Flow Establishment

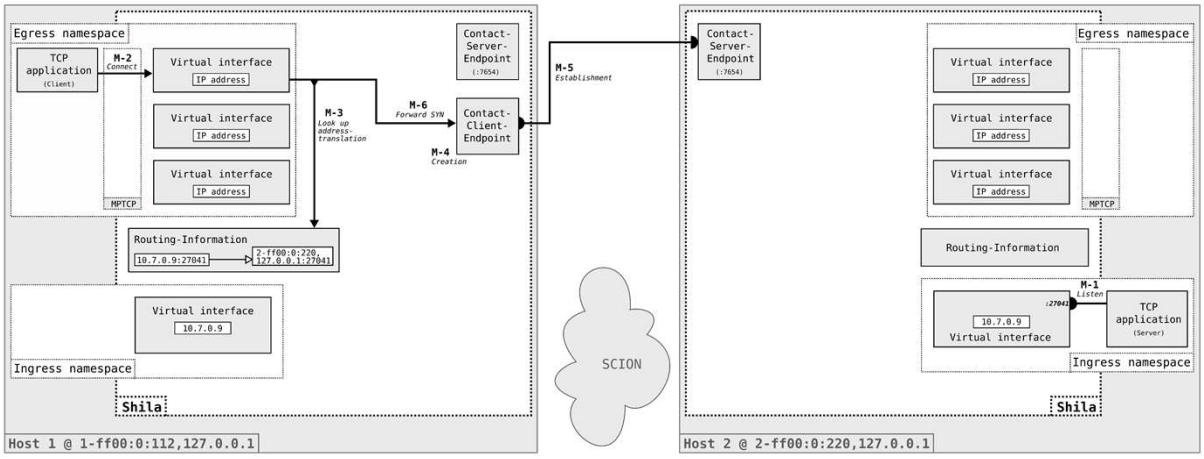


This Contact-Client-Endpoint then establishes a backbone connection to the Contact-Server-Endpoint listening in the Shila instance on Host 2.

Note: The SCION destination address is known from the Routing-Information.
For the port, a default one (7654) is used.

Functionality of Shila

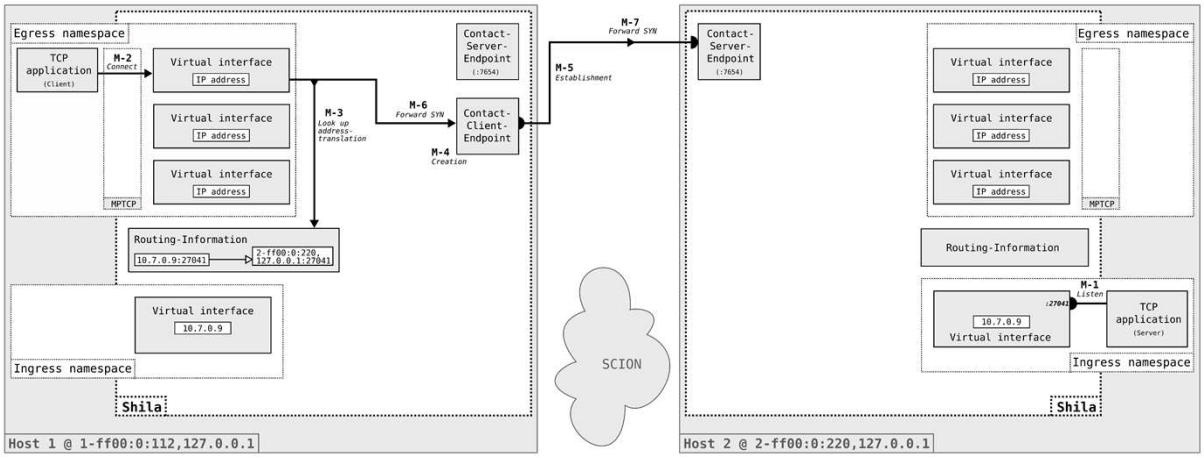
Main-Flow Establishment



Once the so-called Contact-Backbone-Connect is established, the datagram with the SYN is forwarded…

Functionality of Shila

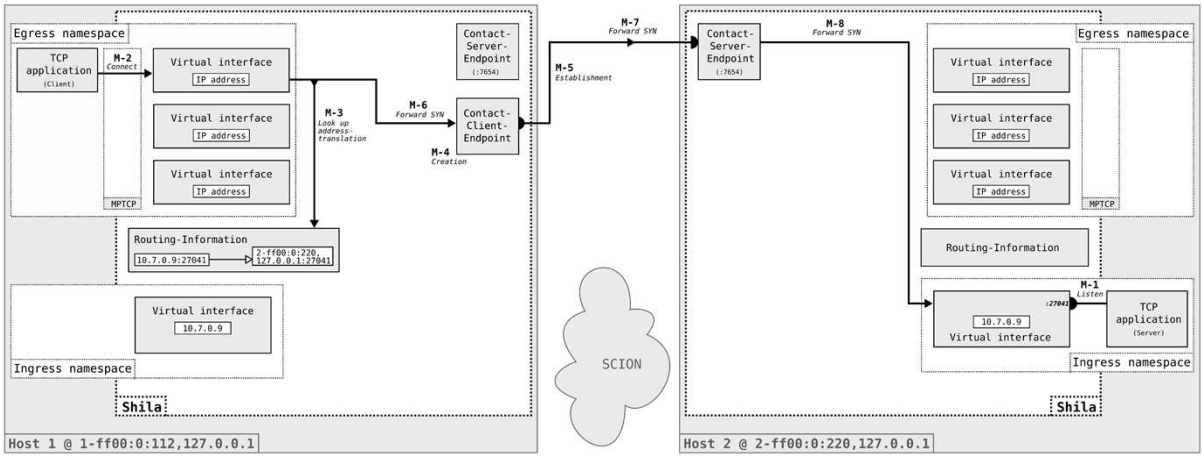
Main-Flow Establishment



…

Functionality of Shila

Main-Flow Establishment

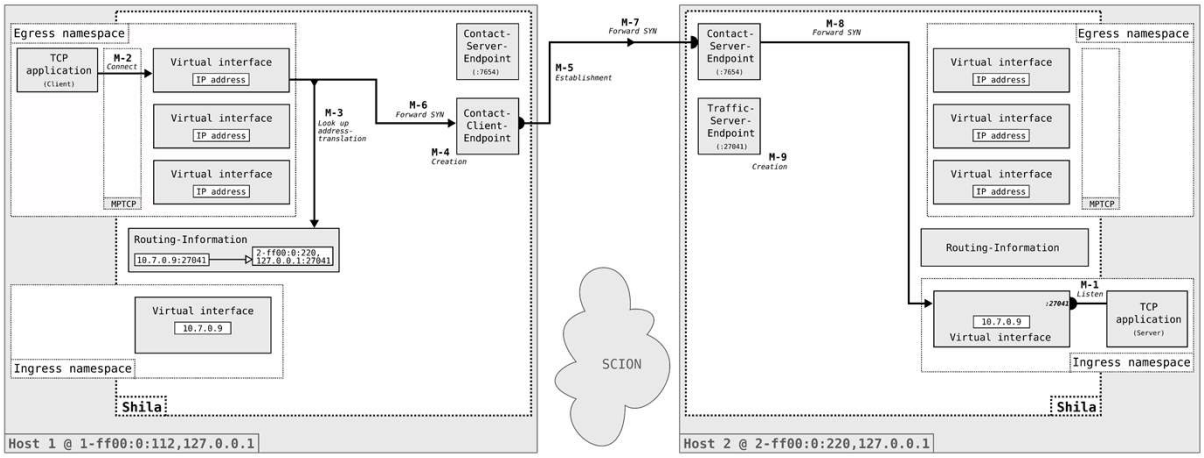


… to the ingress virtual interface on Host 2.

From there it is forwarded via the kernel networking to the listening TCP application.

Functionality of Shila

Main-Flow Establishment



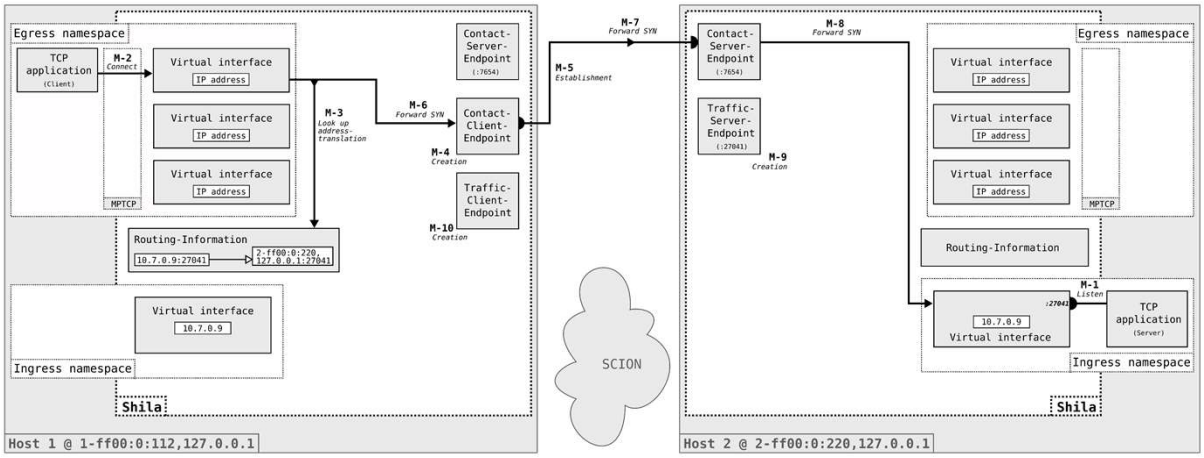
Upon reception of the SYN datagram, Shila on Host 2 creates a so-called Traffic-Server-Endpoint listening on SCION port 27041.

The necessary information, that is, what port to listen on, was exchanged as part of the creation of the Contact-Backbone-Connection.

There is no need for Shila to parse the IP datagram on the receiving side.

Functionality of Shila

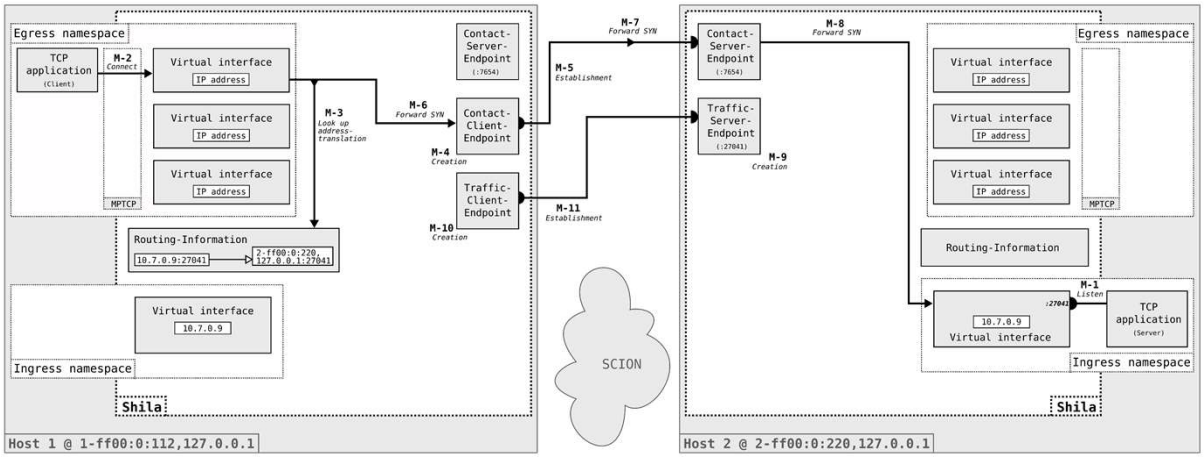
Main-Flow Establishment



Shila on Host 1 in the meantime creates the Traffic-Client-Endpoint…

Functionality of Shila

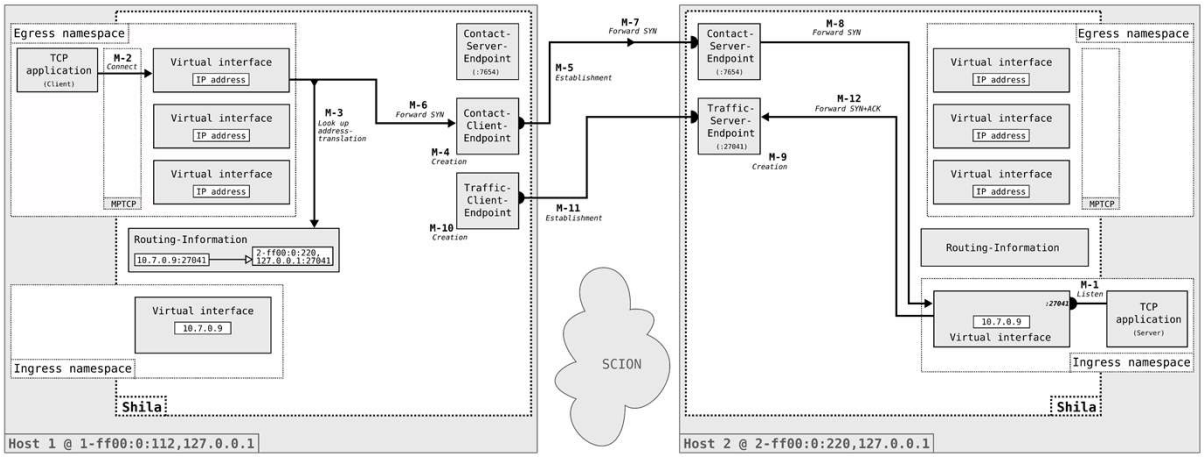
Main-Flow Establishment



And the so-called Traffic-Backbone-Connection is established.

Functionality of Shila

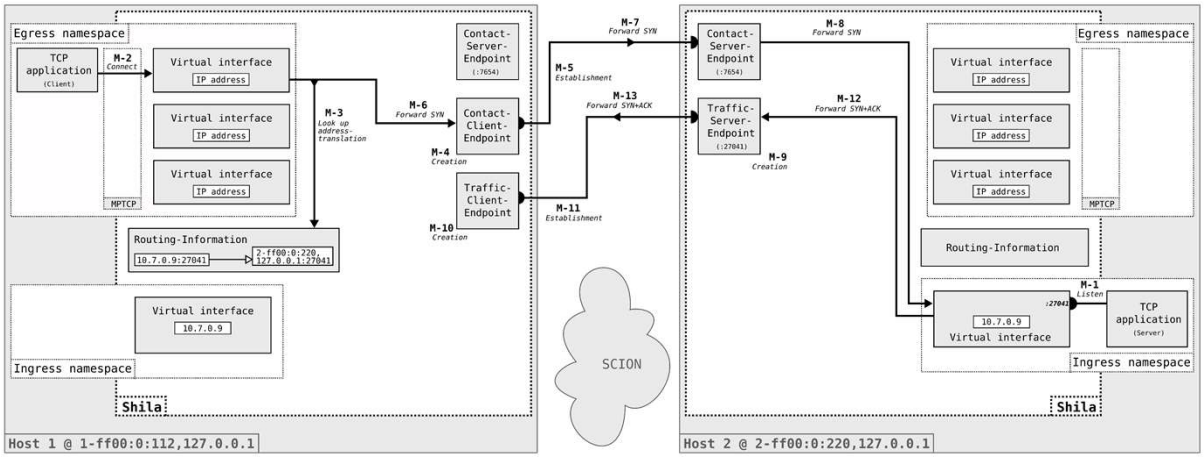
Main-Flow Establishment



In the meantime Shila intercepts the SYN+ACK of the TCP server application on Host 2 and routes it via the newly created Traffic-Backbone-Connection..

Functionality of Shila

Main-Flow Establishment

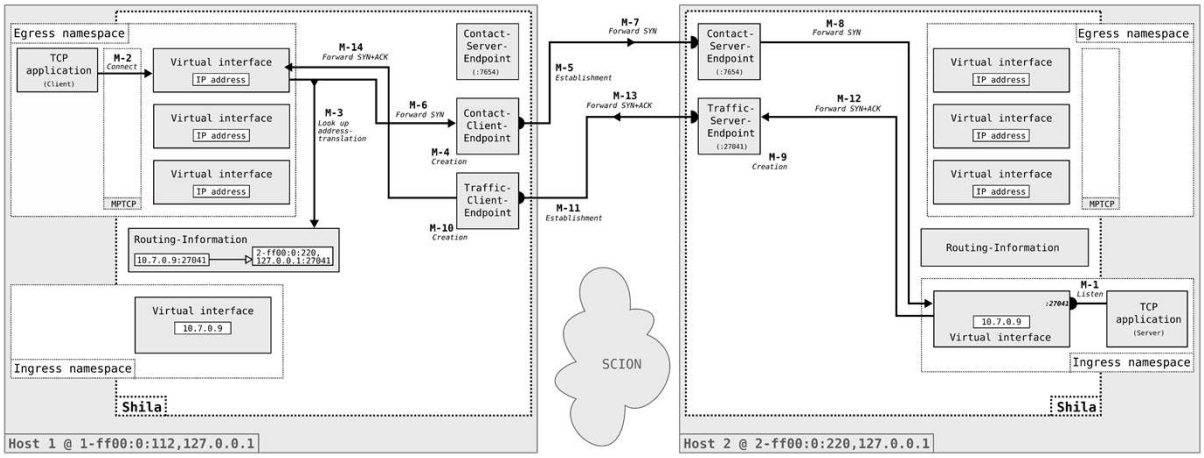


31

…

Functionality of Shila

Main-Flow Establishment

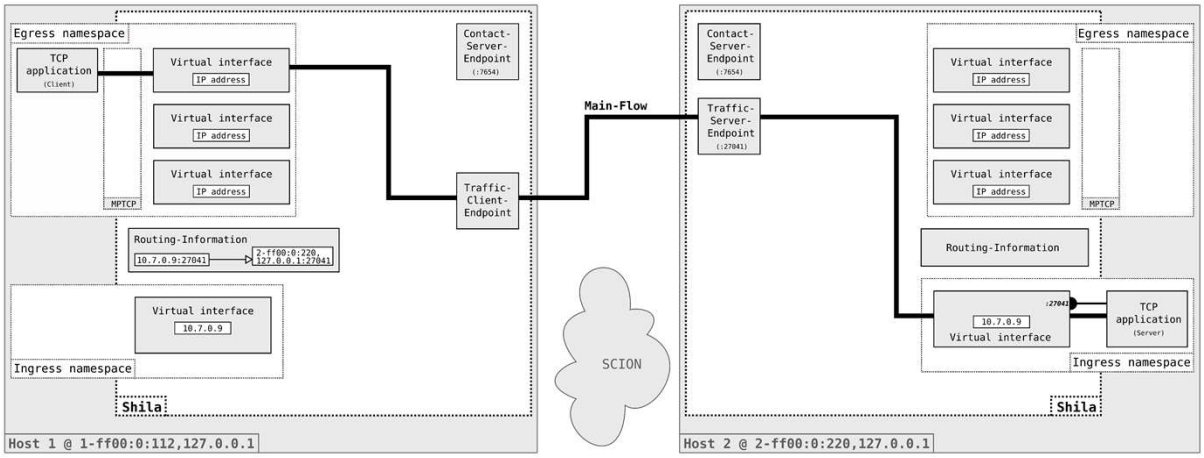


..back to Shila on Host 1. There the SYN+ACK is forwarded through the respective virtual interface to the TCP client.

This concludes the establishment of the Main-Flow…

Functionality of Shila

Main-Flow Establishment

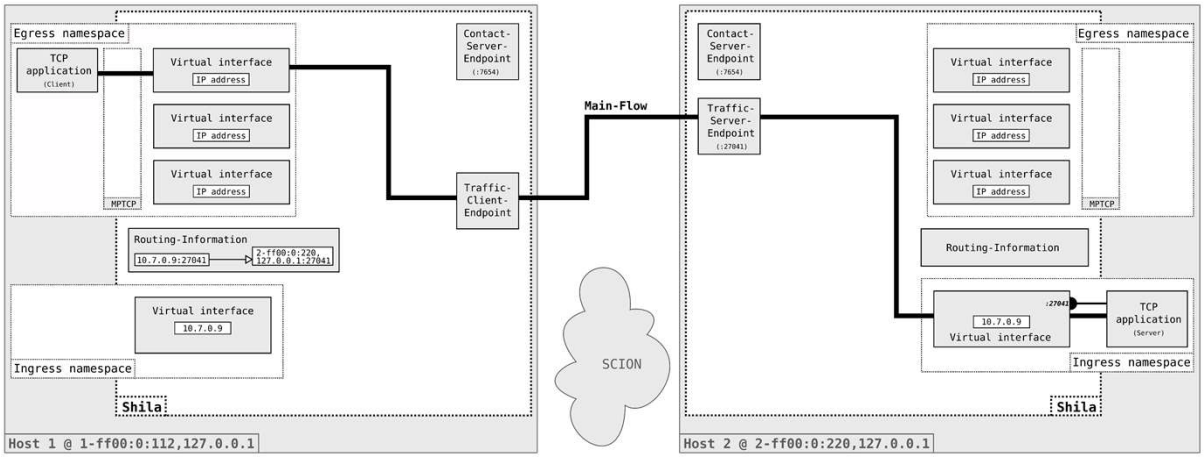


33

..The Contact-Backbone-Connection and the associated endpoint (Contact-Client-Endpoint) is removed, and the data can be exchanged via the Main-Flow.

Functionality of Shila

Sub-Flow Establishment



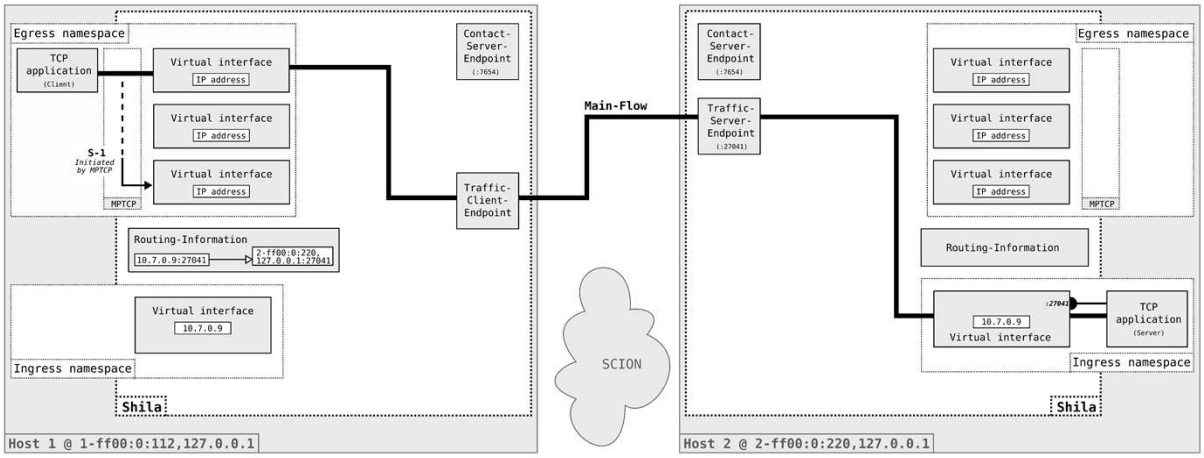
34

As soon as the Main-Flow is established, MPTCP starts to initiate further flows using the other available virtual interfaces in the egress namespace..

We call this flows Sub-Flow and I will now go through the steps of a Sub-Flow establishment. However, the most steps are similar to the establishment of a Main-Flow,
so I will therefore just mention the differences..

Functionality of Shila

Sub-Flow Establishment

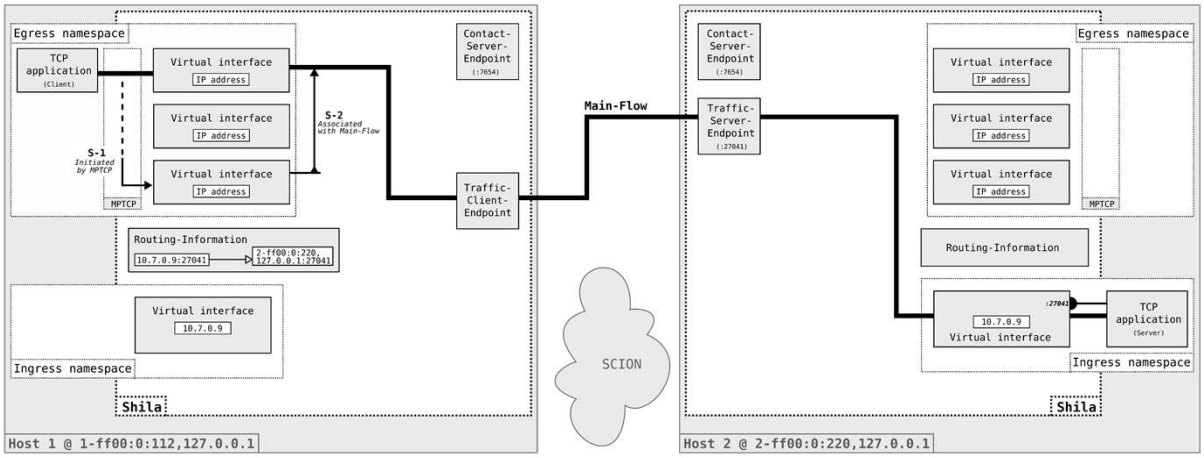


35

As I already said, a Sub-Flow is initiated by MPTCP itself and not by the application.

Functionality of Shila

Sub-Flow Establishment



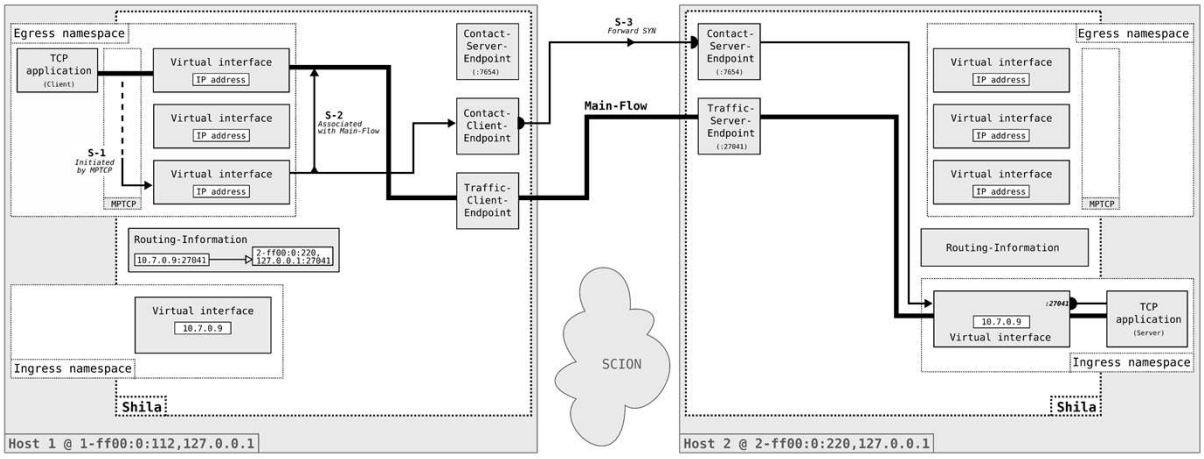
The SYN for the Sub-Flow is intercepted by Shila.

To get the destination SCION address she does not need to consult the Routing-Information again.

In MPTCP, every Sub-Flow is associated with a Main-Flow. Using this association Shila figures out the destination SCION address.

Functionality of Shila

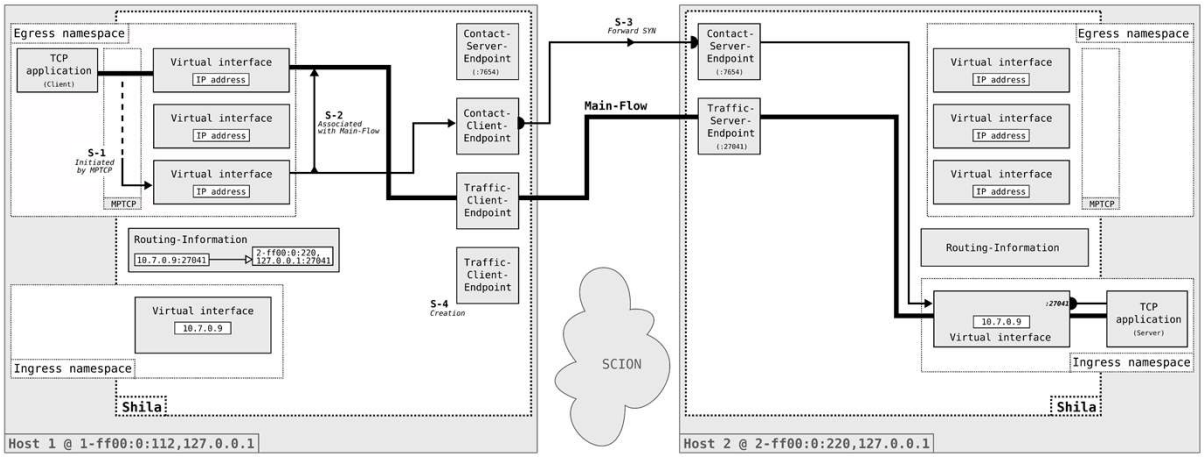
Sub-Flow Establishment



With the destination SCION address at hand Shila can contact its opposite instance and forward the SYN to the TCP Server application.

Functionality of Shila

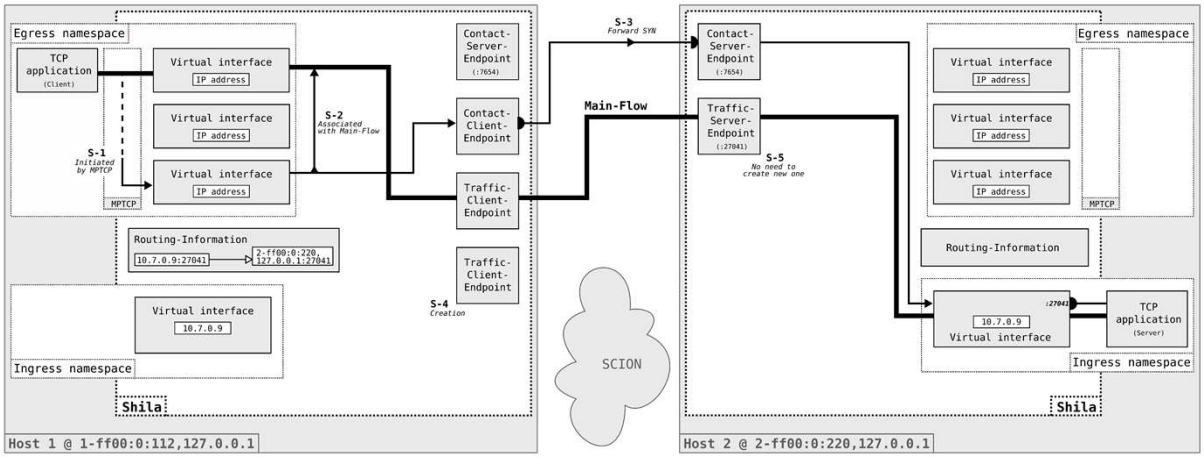
Sub-Flow Establishment



It creates a new Traffic-Client-Endpoint..

Functionality of Shila

Sub-Flow Establishment

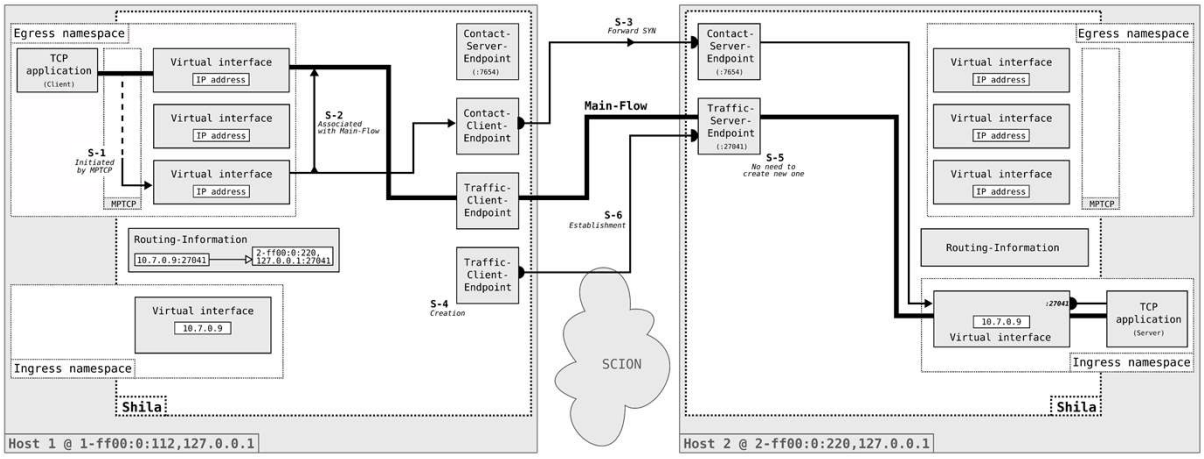


39

and..

Functionality of Shila

Sub-Flow Establishment



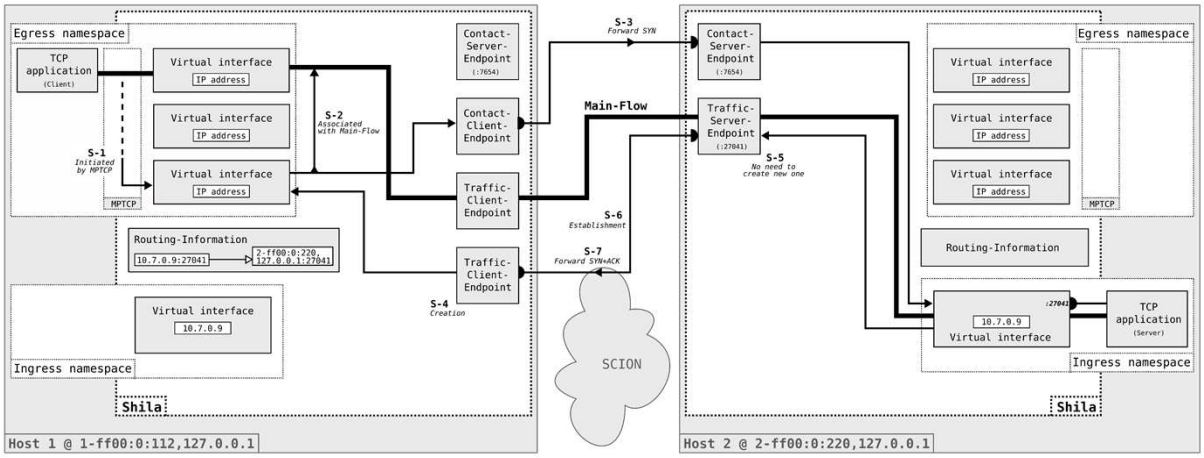
40

Creates a new Traffic-Backbone-Connection to the already existing Traffic-Server-Endpoint.

Note that there is no need to create a new Traffic-Server-Endpoint on Shila Host 2.

Functionality of Shila

Sub-Flow Establishment

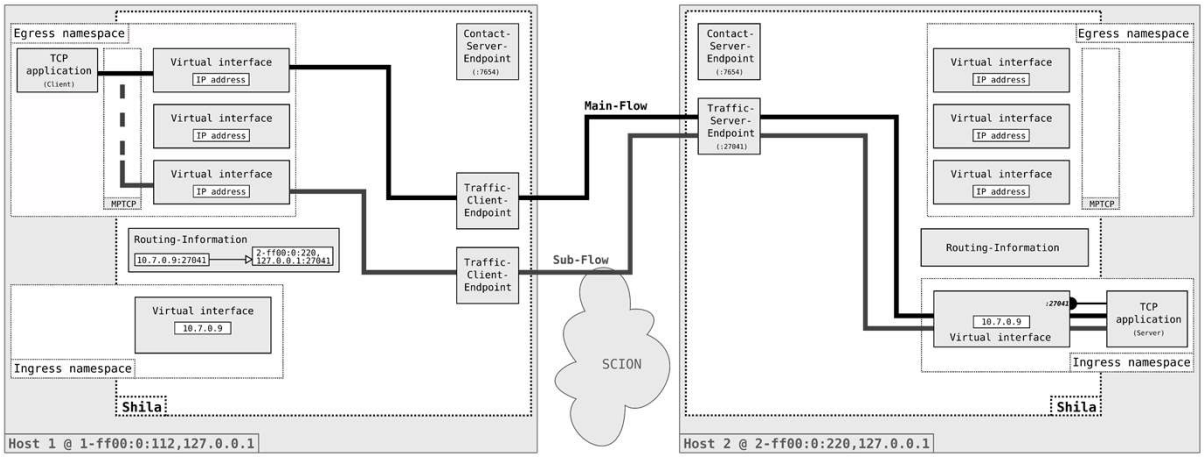


41

Once the Traffic-Backbone-Connection is established, the returning SYN-ACK can be forwarded through, towards the client TCP application.

Functionality of Shila

Data Exchange



This concludes the establishment of the Sub-Flow.

The MPTCP connection illustrated here now consists of two flows. The mechanisms of MTPCP decided on which flow to use during the data exchange.
Although theoretically possible, Shila does not affect the selection of the flow to be used.

(Shila could for example intentionally drop packets on a certain flow..)

Of course would MPTCP initiate a third flow over the yet unused virtual interface.

Shila maintains the flows as long as there is data flowing through. The Backbone-Connections and its corresponding Endpoints are removed if
there is no more data flowing.

On the menu

- Introduction
- Functionality of Shila
- Performance evaluation
- Future work

43

So with an idea how Shila works, lets move on to the performance evaluation.

Performance evaluation

Questions of interest

How does the performance behave in relation to the number of paths used for a connection?

How well does Shila perform compared to QUIC over SCION?

44

Here, I would like to present the answers to two questions that I examined in the course of this performance analysis:

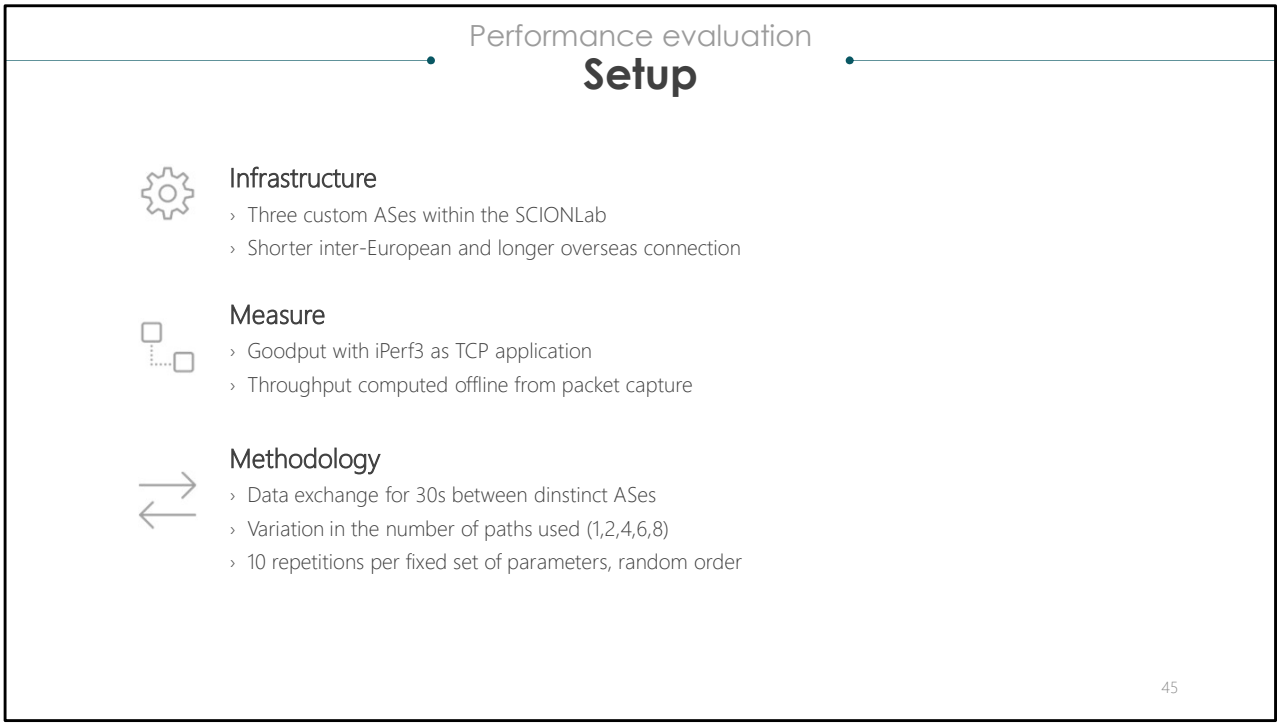
firstly:

How does the performance behave in relation to the

number of paths used for a connection?

and secondly:

How well does Shila perform compared to QUIC over SCION?



To answer this questions I conducted measurements with the following setup:

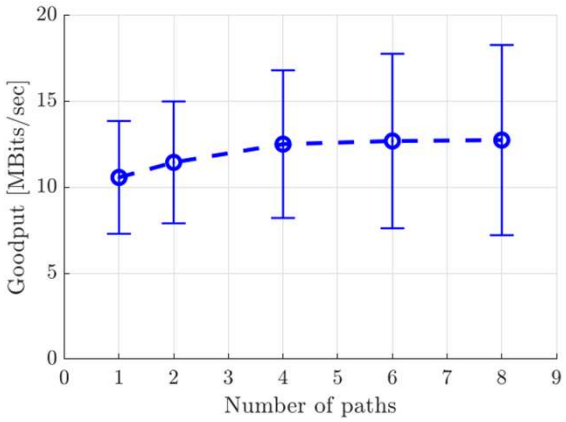
As an infrastructure I used the SCIONLab with three custom ASes. I choose the attachment points for these three ASes such that I can represent shorter inter-European as well as longer overseas connections.

As a measure I recorded values for the goodput and the throughput. The measurement of goodput was done with iPerf3 as a TCP application, and the throughput I computed offline from the packet capturing.

I recorded the metrics for a 30s data exchange between two distinct custom ASes. Between the experiments I varied both the ASes and the number of paths used. Every experiment I repeated 10 times and I did them in a random order.

Performance evaluation

Result



How does the performance behave in relation to the number of paths used for a connection?

An increase in the number of paths leads to an increase of the average goodput.

46

Let's answer the first question:

As we can see in this plot here: Increasing the number of paths leads to an increase in the average goodput.

This is what one would expect, nevertheless this results serves as some sort sanity check

What I want to mention here is the relatively large deviation:
This comes from the different performances of the connections. Shorter inter-European connections achieved higher values than the longer oversea connections. However, the shape looked for all the same.

Performance evaluation

Result

Paths	Shila		QUIC over SCION	
	Goodput	Throughput	Goodput	Throughput
1	10.75 ± 3.28	17.45 ± 5.91	33.31 ± 3.28	36.82 ± 3.62
8	12.74 ± 5.53	19.37 ± 8.36	-	-

MBits/s

≈ 2.6 ×

How well does Shila perform compared to QUIC over SCION?

Shila gets outperformed by QUIC over SCION with respect to goodput as well as overhead.

47

To the second question, the comparison to QUIC over SCION.

Unfortunately got Shila outperformed for both measures, Goodput as well as Overhead.

QUIC over SCION achieves a goodput that is about 2.6 times higher than the one achieved with Shila using eight paths. One reason, for this is that the data that is sent with MPTCP over SCION makes a detour through the kernel back to the userland application Shila on both endpoints. This increases the round-time-time what
Itself has a negative impact on the goodput.

The verdict is also clear on the overhead: while QUIC over SCION works with an overhead of about 10%, MPTCP over SCION has an overhead of about 30%.

Of course such results do not make friends but Shila is willing to improve and develop in the future, which brings me to the last part of my presentation..

On the menu

- Introduction
- Functionality of Shila
- Performance evaluation
- Future work

48

The future work.

Future work

What is next?

Conduction of revision cycles

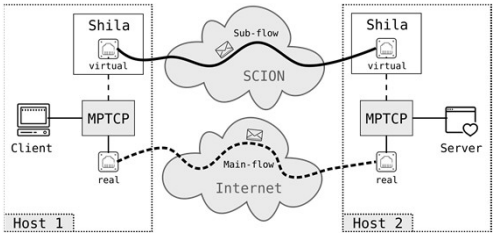
- Improve implementation of Shila
- Further testing

Addition of flexibility

- Provide mapping betwenn TCP and SCION destination address upon connection establishment
- Remove need for namespaces

Side-by-side approach

- Main-flow over conventional Internet
- Sub-flow(s) over SCION



49

For further work on my thesis one should consider the following points:

Conduction of revision cycles: The implementation of Shila has still in some sort prototype character. Further testing and revision cycles should
increase not just the quality but also the performance of the implementation.

Addition of flexibility: In current state, user have to supply the mapping from TCP to SCION destination addresses upon start-up of Shila. This is not really practicable, and the goal should be
to find and implement an approach which is more flexible. (One possible approach for example, which I could think of would be to provide the necessary option via IP socket options.)

As a last point I would like to mention a slightly different direction in which the journey of Shila could go. Instead of using her capabilities only for data exchange via MPTCP over SCION,
a side-by-side approach is also conceivable. The Main-Flow of an MPTCP connection goes over the conventional Internet. Whereas the Sub-Flows are escorted by Shila through the SCION network.

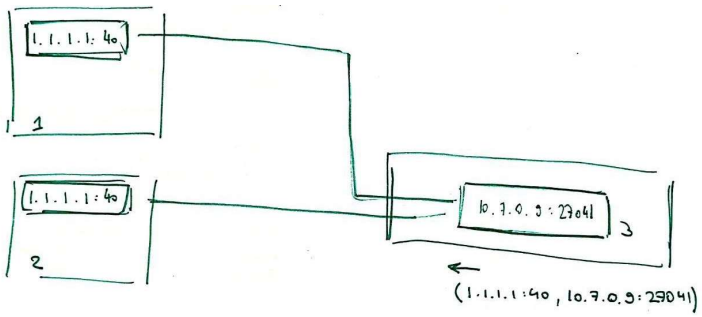
This approach does not interfere with a users known working environment but facilitates the development of SCION. Users could benefit from the better internet architecture without even
realizing that they are using it.

With this outlook I reached the end of my presentation. Thank you.

Thank you.

Notes

Random Egress IP's

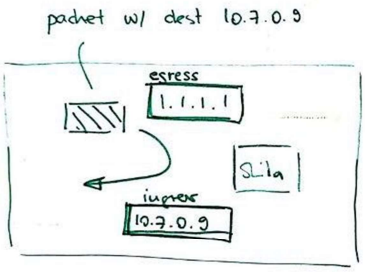


↳ need to be able to map TCP to Backbone-Connection.

↳ Sibra (host) could (does) not accept request from different SCION with already observed src TCP address.

Notes

Why namespaces?



→ is possible, but needs unique ip for every hosts ingress virtual interface.

→ ip address part of SCION not necessarily unique between different ASes

→ not scalable

→ don't wanted to mess w/ local routing.

↳ if inside namespace choice doesn't matter.

Shila as a girls' name means "blind". Shila is a varian form of Sheila (Irish, Gaelic): version of Sile.

STARTS/ENDS WITH Sh-, -la