Anycast in the SCION Internet Architecture

Master Thesis

Dennis Eijkel

Internet Science & Technology
University of Twente

Graduation Committee:
prof.dr.ir. G.J. Heijenk, University of Twente (1st supervisor)
ir. J. Scholten, University of Twente
dr.ir J. de Ruiter, SIDN Labs

August 3, 2022

Acknowledgements

First, I would like to thank my graduation committee and supervisors Geert, Hans and Joeri, for all their guidance and support during the making of this thesis.

I would also like to thank SIDN Labs for the opportunity to do my final project and write my thesis as part of an internship with them. It was both an insightful and fun time.

Next, I want to thank Seyedali and the others from ETH Zürich for kindly providing me the SCION beaconing simulator that they built so that I could do my simulations on top of that. This meant that I had a good basis to build on and saved me a lot of time.

Finally, I would like to thank all the other friends and family that helped in one way or another with this thesis or the presentation.

Abstract

Anycast is a widely used technique to deploy globally replicated internet services. It allows operators to replicate their services over multiple geographical locations so that their end users will have a good quality of service, no matter where in the world they are located. SCION, a future internet architecture that is actively being designed and researched, does not yet support deploying such replicated services.

To push development of SCION further, we developed several designs to integrate this into SCION. We also carried out a quantitative analysis through simulations to evaluate whether replicated services in a possible future internet based on SCION would be able to provide better quality of service to the end user than anycast is able to do in the current internet.

We concluded that the simulations show SCION is able to provide a better quality of service. Due to the limited scope of the simulator however, we also concluded that we were not able to answer the question if the same holds for a real-life replicated service deployed in SCION. Furthermore, some key insights were also found in how network configuration and topology influences the quality of service of a replicated service.

Contents

Chapter 1

Introduction

The internet in its current form is a *network of networks*. Within each of these individual networks, also called autonomous systems (ASes), the operator of that network can decide what protocol they want to use to handle routing of traffic inside their network. To enable communication between these individual networks, a common protocol is needed. The Border Gateway Protocol (BGP) [1] is the protocol that handles most of that communication between individual networks in the current internet.

BGP has brought the internet to what it is today, but it has several major issues that are becoming more apparent every year, sometimes leading to outages in parts of the internet. Lack of authentication of information that is exchanged through BGP is an underlying cause of many of these issues and can lead to accidental or malicious use of IP prefixes by unauthorized ASes. Two categories of approaches are currently in development in attempts to resolve these issues. They are: extending BGP, or completely redesigning the architecture of the internet and introducing a new protocol. RPKI [2] and BGPSec [3] both are examples of BGP extensions, both solving (part of) the mentioned authentication issue. An example of an internet architecture that was designed from the ground up is SCION [4]. SCION is an attempt to rethink the way the internet should work and to provide a replacement of BGP and can therefore be referred to as a *future internet architecture*. It is important to research these future internet architectures, because not every architectural issue can be solved by amending existing protocols.

Many large scale internet services, such as DNS root servers [5] and content delivery networks, serve many clients that are spread over many different physical locations all over the world. For such services, having all servers located in the same physical location would mean that some clients, those that are physically close to the servers, experience low latencies. But most clients would then be located far away from the servers and would experience high latency. This would not be ideal, since service operators would like their service to respond quickly to all clients, no matter their physical location. This problem is often solved by using a technique called anycast [6].

Conceptually, anycast is a way of running an internet service, where the service is replicated over many different (geographical) locations. Clients from around the world can use the same address or name to access this service, and the underlying routing infrastructure will attempt to guide them to the closest replica of that service. This then enables operators give clients lower latency. It also enables them to build more redundancy and resilience to failures in their services, further improving end-user experience. Anycast is important in the current internet, it is a popular way of deploying large replicated internet services and is used to operate e.g. the DNS root servers and many content delivery networks.

In the wild, the name anycast is often used in multiple different ways. It is used either to refer to the concept of replicated internet services as described above, or to refer to the implementation of said concept on the routing level i.e. through BGP. For the remainder of this thesis, we will avoid using the word anycast in multiple ways to avoid confusion. The word anycast will only be used to refer to the specific implementation in BGP of the concept of a replicated internet service as described above. To refer to the concept of anycast, we will use the wording *global replicated internet service* or *replicated service* in short.

Given the usefulness of anycast and similar techniques in today's internet, it would also be useful to have a similar feature in SCION. It is not yet clear how a globally replicated service in an internet based on the SCION architecture would compare to anycast in the current internet. Furthermore, it is also not yet defined how to deploy such replicated services in SCION. Therefore we define the following main goals for this project: finding out how these globally replicated services could (and possibly should) work in a SCION based internet and evaluating how these techniques in SCION compare to anycast in the current internet.

1.1 Project Goal

The goal of this project is to design and evaluate one or several solutions for running globally replicated internet services in the SCION internet architecture. It is important for adoption that SCION supports or enables this, since anycast is widely used in the current internet to deploy services. Furthermore, if SCION is able to provide better quality of service, e.g. in latency or time-to-first-byte, than anycast deployments in the current internet can, that might help adoption of SCION.

From a technical point of view, these designs for replicated services in SCION do not necessarily need to work in the same way as anycast in the current internet. It only needs to provide a conceptually similar solution, solving the same problem as anycast does for the current internet. Users should be able to use a single address or name to access a replicated internet service, and with that end up connected to the *best* replica. The best replica does not always have to be the one with the lowest latency or smallest geographical distance, it could also be the replica that has the highest available bandwidth or lowest load, or a combination of any of these.

It is good to note that SCION already uses *anycast* to address some control-plane services [4], but that is not the level of anycast that this project is aiming for. These control-plane services are local to the AS that they are running in. This project aims to build a global replicated service system in SCION, evaluate it and compare it to anycast in the current internet.

1.2 Research Questions

To guide the direction of this research, we have defined two main research questions derived from the goals of this thesis as mentioned before. We also defined some sub questions that need to be answered to be able to answer their main question.

RQ1 How can globally replicated services be supported in the SCION architecture?

RQ1.1 What would be requirements for designing such solutions?

RQ1.2 How do the designs compare to the requirements?

RQ2 How do globally replicated services in SCION compare to anycast in the current internet?

RQ2.1 How can the designed solutions be compared to anycast in BGP?

To be able to answer the first research question, we will need to define requirements to guide the design process and make sure that they are in fact conceptually similar to anycast in the current internet. Furthermore, it is needed to evaluate the designs against the requirements since not every requirement is a must-have. Some of the requirements could be seen nice-to-have or they could require some form of quality of service, which would be hard to evaluate without an in-depth analysis on that requirement. These aspects have been described in the two sub research questions RQ1.1 and RQ1.2.

Also, to be able to answer the second research question, we will first need to answer the question how we can compare these designs against each other. This has been put into the sub research question RQ2.1.

1.3 Thesis Structure

The remainder of this thesis is structured as follows. First, we will delve into the background of the different concepts that are needed to understand the main body of this thesis, the backgrounds of BGP, SCION and anycast. This we will discuss in Chapter 2. Then in Chapter 3, we will answer the first research question (RQ1) of this thesis, specifying the requirements, designing methods for deploying replicated services in SCION and comparing designs against the requirements. After that, we will go into the second research question (RQ2), evaluating the designs and comparing them with anycast in the current internet in Chapter 4. Finally, we will provide conclusions and future work in Chapter 5.

Chapter 2

Background

SCION is an emerging internet architecture that aims to provide more trust and route control than the current internet powered by BGP provides. In this chapter we will introduce BGP and SCION and explain their differences. We will also go into how anycast works in BGP. The contents of this chapter are based on the Research Topics report that is the precursor to this Master Thesis report [7].

2.1 BGP

The current internet is a collection of interconnected autonomous systems, often also referred to as a network of networks. An autonomous system can be seen as a network that is operated by a single organization. Intradomain routing refers to routing of packets within an AS and the protocol that is used for that is up to the operator. Interdomain routing refers to routing between ASes and a common protocol is needed to be able to communicate between individual networks.

ASes can peer with other ASes, which means that they have a direct connection to each other. An example of how a topology would look in BGP is given in Figure 2.1. This type of topology is not necessarily unique to BGP, but is used to make it easier to show the differences in network topology and structure between BGP and SCION later on.

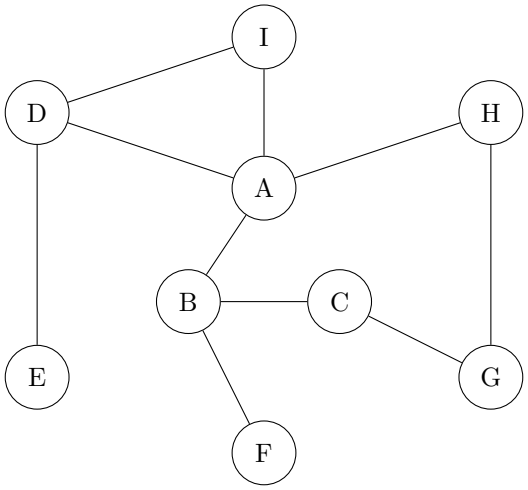


Figure 2.1: Example topology for BGP with ASes and peerings between them. Note that there is no explicit hierarchy between the different ASes, but some ASes do have more neighbors than others.

Each AS has a set of IP prefixes that they serve from their network. IP prefixes are blocks of IP addresses identified by an IP address and prefix length such as `10.1.2.0/24`. This prefix encodes the block of IP addresses ranging from `10.1.2.0` through `10.1.2.255`. The prefix length indicates how many of the bits of the IP address are fixed and thereby also how large the block is. Prefixes can also be encapsulated by other shorter prefixes, for example `10.1.2.0/24` is a subset of the prefix `10.1.0.0/16`. To make sure that each AS can reach every address that exists in the internet, ASes need to exchange information about prefixes with each other and a routing protocol is needed to enable this.

Routing

The Border Gateway Protocol [1] is currently the routing protocol that powers most of the interdomain routing and thus the internet. It is used to exchange information about where to reach different prefixes. In BGP, border routers advertise and/or withdraw routes to IP prefixes that they know of to their peers. BGP is a path-vector routing algorithm, which means that when a border router receives an advertisement for a certain IP prefix from one of their peers it will, after aggregation and filtering policies of the AS operator are applied, and after prepending its own AS number to the path that is sent with the advertisement, forward that advertisement to its other peers.

From advertisements that border routers receive from their peers, they build up a routing table, containing all of the prefixes that they have seen accompanied by the peer that can reach that prefix. A router can also receive multiple advertisements for the same prefix from different peers, it will then look at the AS paths that were sent with the advertisements and choose a path based on policies defined by the AS operator and add it to the routing table.

When border routers receive traffic that they need to forward, they look at the IP address of the packet that they need to forward and look in their routing table to see which prefixes match with that IP address. If there is only a single match in the routing table, the router can just forward the traffic to the peer indicated by that match. Whenever there are multiple matches, indicating that the IP address is part of multiple prefixes in the routing table, they will choose

the route of the longest matching prefix. So if there are entries in the routing table for prefixes `10.1.0.0/16` and `10.1.2.0/24`, and the destination IP address is `10.1.2.2`, both prefixes will match with this address but `10.1.2.0/24` is the longest matching prefix.

In essence BGP is quite simple; exchange information about reachability of prefixes and build routing tables based on that information. During the growth of the internet as it is now, it has been working quite well. But it is not entirely without problems. There are multiple security issues that can be and are exploited in the current internet.

Security

BGP does not cryptographically protect advertisements, thus routers have no way of verifying if the information in advertisements is authentic and has not been modified. Malicious actors could modify advertisements while they are in transit between peers if they are able to or just make up and send out bogus advertisements for prefixes that they do not own or paths that do not exist. Operators could use databases with peering information such as PeeringDB [8] to verify prefix and path information, but these databases must contain information about most networks to provide sufficient benefit.

This weakness could be used by malicious actors to execute a so-called *prefix* or *route hijack*. There are two things that an attacker could forge to execute a hijack, either forging the route path or the origin. When forging the route path, an attacker would advertise a shorter route to its neighbors than those neighbors currently know of and thus attract traffic destined towards the origin of the forged route. Forging the route origin means that an attacker would advertise more specific prefixes of a certain IP prefix that is owned by the victim that it wants to target. This variant of the attack abuses the fact that routers will consider the longest matching prefix to be the best route to a destination, meaning that an attacker just has to advertise a longer/more specific prefix to re-route all traffic for those addresses through their AS.

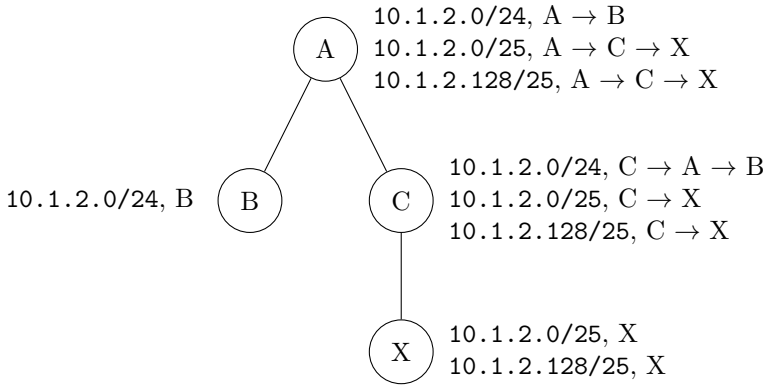


Figure 2.2: Example of a BGP hijack by attacker AS X advertising more specific prefixes of the prefix owned by AS B.

An example of a prefix hijack by advertising more specific prefixes is shown in Figure 2.2. In this example, AS B is the legitimate owner of prefix `10.1.2.0/24`. The attacker, AS X, hijacks that prefix by advertising that it is the origin for both the prefixes `10.1.2.0/25` and `10.1.2.128/25`. ASes A and C would receive both the advertisement from B and the more specific ones from X. When networks A and C receive packets destined for an IP address in the hijacked prefix, they will always choose to send the packets to AS X because the attacker

advertised more specific prefixes.

With this hijacking attack an attacker could attract some, a lot or sometimes even all traffic in the internet that was destined for the victim's network to its own network. The amount of traffic that is attracted depends on the AS topology and the policies of AS operators. With the hijacked traffic the attackers could eavesdrop, execute a denial of service attack by dropping the traffic, or they could provide the expected service but with (unnoticeable) modifications to extract sensitive data.

Effort has been made to design extensions to BGP that solve (some of) these issues, e.g. RPKI [2] and BGPSec [3]. These extensions solve two different verification aspects and they will only work well enough when most operators use them. RPKI allows for origin validation, to verify if an AS that advertises that a certain IP prefix originates from their network that it actually owns that prefix or is allowed to use it. BPGSec allows for path validation, to verify if paths advertised by peers actually exist and are not forged. Widespread adoption is crucial for these extensions to be able to provide their benefits.

2.2 SCION

SCION [9, 4] is an internet architecture that aims to replace BGP and was built from the ground up. Its aim is to provide more control, transparency, scalability and availability than the current internet can with BGP.

The concept of autonomous systems still exists in SCION, but SCION adds a new layer of hierarchy on top of that: *isolation domains* (ISDs). Isolation is a core concept in SCION, other entities outside an ISD should not be able to affect routing within that ISD. Administration of these ISDs is done by one or multiple so-called *core* ASes. An ISD would then contain core ASes that peer with each other and with core ASes in other ISDs, and *regular* ASes that mainly have peerings with other ASes in the same ISD and are directly or indirectly connected to the core of the ISD. An ISD contains a trust root configuration (TRC), which contains the cryptographic trust root that is used for validation of path information and is maintained by the ISD's core ASes. Figure 2.3 contains an example topology for SCION, indicating ISDs, core ASes and regular ASes with peerings between them.

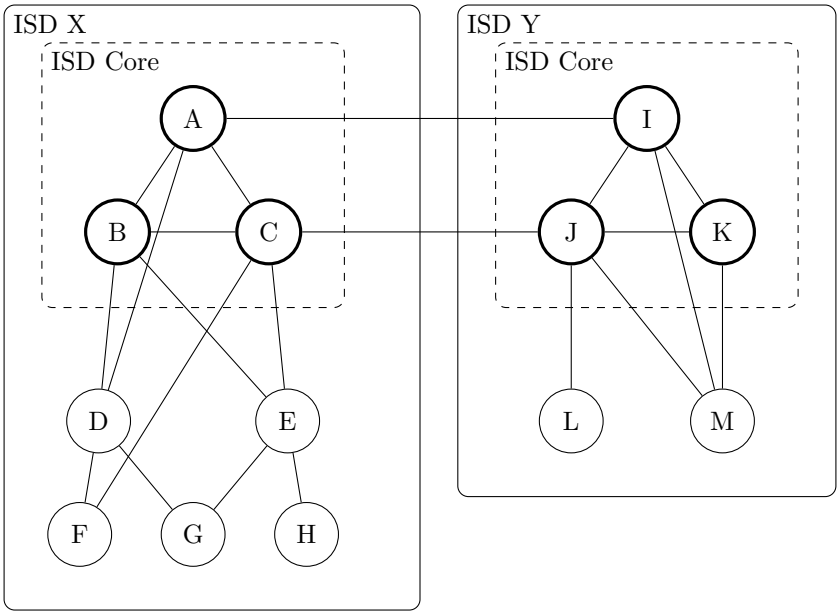


Figure 2.3: Example topology in SCION with ASes, ISDs and ISD cores. Showing the hierarchy that SCION has with its concept of ISDs.

An address in SCION is a triple (3-tuple) of the form (ISD, AS, address). An example of a SCION address from the topology from Figure 2.3 would be (X, E, `10.1.2.3`). The first two parts of the triple are identifiers for the ISD and the AS within that ISD. The address in the third part of the triple denotes the address that can be used to reach the destination host inside of the destination AS, which is denoted by the combination of the ISD and AS identifiers. The address in the third part therefore only needs to be used by the destination AS and this means that each AS can decide for themselves which addressing format (IPv4, IPv6 etc.) they want to use inside of their network.

Path finding

Discovering paths in SCION is done through the *beaconing process*. Beaconing servers, one of the core components of a SCION AS, periodically send out path-segment construction beacons (PCBs) to discover paths. This is done on either an inter-ISD level (between core ASes) or intra-ISD level, thus beacons only have to travel as far as required by their scope, thereby splitting up the beaconing process, and increasing scalability of the beaconing process in the global network.

Beacons contain cryptographically signed information about the path that the beacon has traversed starting from the AS that initiated the beacon. When a beacon server receives a PCB, it appends its own connection information to it and then forwards the PCB to its other neighbors. There is quite a bit of information that is appended to the PCB at each hop, but we will focus on the part that is most important to understand SCION routing and that is the hop field.

Hop fields contain the ingress and egress interface identifiers over which the beacon entered and left the AS on its traversal. A message authentication code is also included in the hop field to allow authenticating the information. Interface identifiers are unique in the scope of the AS that they belong to, when an AS has multiple links to the same neighbor, they would have to be represented by different interfaces.

Discovered path segments are also registered at the path servers, another core component of a SCION AS, so that these segments can later be used for routing packets over the network. Certificate servers, another core component, are used to aid in verification of information received in PCBs.

Routing

End-to-end paths in SCION can consist of up to three segments: an *up*-, *core*- and *down*-segment. The complete path will travel from the source AS *up* to one of the cores of its ISD, then from the core AS in the source ISD to a core AS in the destination ISD, finally from the core AS in the destination ISD *down* to the destination AS. When the source and destination of the path are in the same ISD, it is not necessary to have a core-segment. End-to-end paths and their segments are also indicated in Figure 2.4, showing two example paths with the same source and destination and their segments.

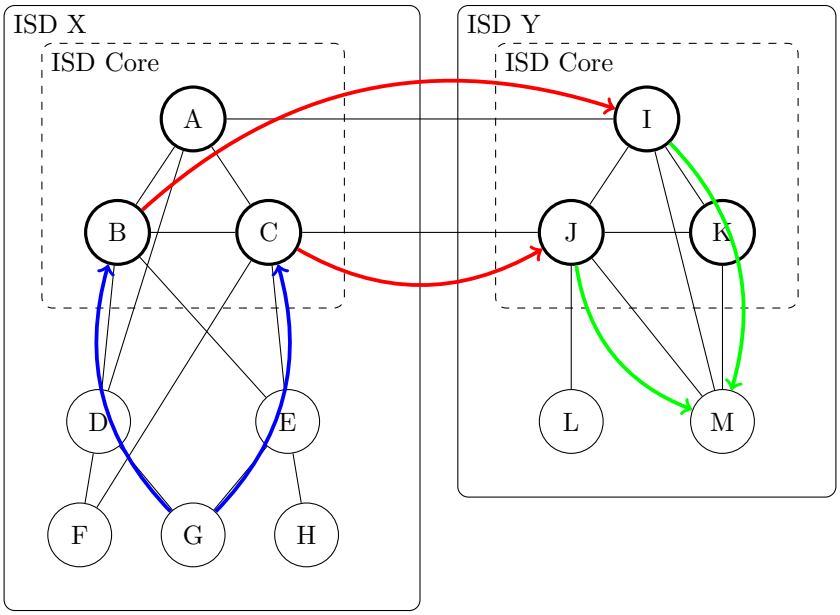


Figure 2.4: Example SCION topology also indicating example paths from G to M and their path segments. The blue, red and green arrows indicate that those segments are up-, core- and down-segments respectively.

When a host wants to send packets to another host on the network, the sender will first have to obtain a SCION address for that destination. This address could also be retrieved through a name resolution system first. Once the source host has an address, it can then query the path server of the AS that it is in to obtain path segments to that host. The host can then construct a graph out of all of the received path segments and select a path from the graph that it would like to send its traffic over. Next, the host takes the hop fields from the path segments that make up the chosen path, and inserts those into the header of a SCION packet. Finally, it appends the payload and sends the packet out on the network.

Figure 2.5 contains simplified examples of the contents of SCION packets, to illustrate how paths are encoded in hop fields and sent over the network. As mentioned before, path segments

contain a sequence of hop fields which contain cryptographically verified instructions of what each SCION router on the path segment needs to do with the packet to make it travel through the network over the intended path.

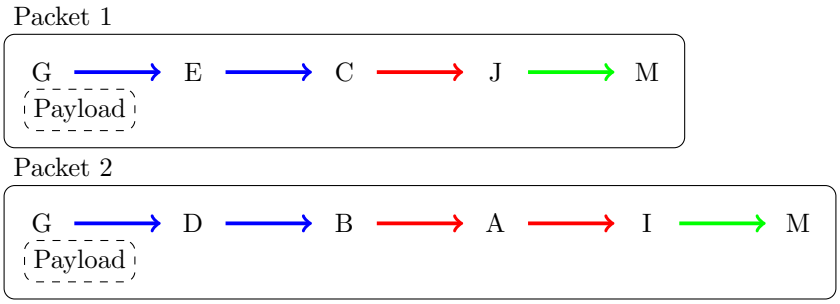


Figure 2.5: Simplified example of the contents of SCION packets, showing the path encoded in hop fields and the payload. The paths are the same paths as depicted in Figure 2.4.

Path lookup

Path lookup starts with the source host sending a query to its local path server. The path server local to the source host will upon receiving a path request fetch one or several up segments that were stored in its database by the beacon server. It will then ask for core- or down-segments at the path server of the core AS at the end of the up-segment, depending on whether the destination address is in the same ISD as the source, and return the segment(s) to the local path server. The core path server of the source ISD will in turn ask the core of the destination ISD for down segments if the destination is in a different ISD and also return those segment(s) to the local path server. All of the fetched segments are then returned by the local path server to the source host.

After path lookup is complete and the source host has received the path segments, the source can construct a graph out of the received path segments and use that graph to select the shortest path, or any other path that it wants. Since the source host has constructed a graph which will contain many available paths to the destination, there could also be disjoint paths to the destination in that graph. Using disjoint paths can increase availability by using both paths simultaneously or allow for quick failover in case the main path fails.

Name resolution

To do name resolution, SCION proposes a DNSSEC-like protocol called RAINS in the SCION book section 6 [4]. With RAINS, each ISD will have to maintain some extra information as part of the ISD configuration on who is the authority for a certain top-level domain (TLD). In theory, different ISDs could have different authorities for each TLD, leading to different views of the internet based on which ISD an end host is connected to. However, the SCION authors argue that this is an unavoidable consequence of having isolation (of ISDs) as part of the architecture. They also argue that the transparency that SCION has on this point makes it so that operators that deviate from the consensus on who the authority is for each TLD can be called out on that, and that that will make sure that there will not be too many deviations from this consensus.

2.3 Comparison between BGP and SCION

Since SCION is a clean-slate design, there are many differences between SCION and BGP. This section will elaborate on some of those differences.

SCION has a clear hierarchy and isolation in the isolation domains. Each ISD has a configuration which is managed by the group of core ASes in that ISD. These core ASes thus have a managerial task within the ISD which regular ASes do not have. BGP in theory does not have such a clear hierarchy. However, it is common in BGP for ASes to be *customers* to *higher tier* ISPs, where the only connectivity that the customer ASes have to the internet is through that ISP. So where BGP has hierarchy to some extent, it is not as clear as the ISDs in SCION.

The isolation principle in SCION plays a big part in the security benefits it has over BGP. In SCION, an address is made up by the tuple (ISD, AS, address) where the first two parts identify the ISD and AS inside that ISD. The information in the beaconing process is signed by keys that are part of a public key infrastructure (PKI), with a root key that is managed by the core ASes of the ISD. The fact that cryptographic authentication and verification are an integral part of SCION makes it resilient to attempts by malicious actors to interfere with the beaconing process. Plain BGP does not have this level of security, thus allowing for route hijacking. Extensions such as RPKI [2] and BGPSec [3] could alleviate most of these problems, but require every AS operator to run them.

The SCION authors refer to SCION as being path aware networking and having a packet-carried forwarding state. SCION also has some resemblance to source routing, and link-state routing. In source routing protocols, senders have to specify the path that the packet travels through the network in the packet header. In SCION this is partially the case, the sender specifies the path that a SCION packet travels over. However, the client does not need knowledge of the entire topology of the internet, which would be the case in pure source routing. Link-state routing is a class of protocols where routers disseminate information about the connections that they have with other routers. Each router can then build a network topology graph and routing table based on that information. During the beaconing phase of SCION, the beaconing servers share this type of information with neighboring networks.

In contrast, BGP is a path-vector routing algorithm as mentioned before. Border routers speaking BGP regularly exchange information about their routing tables with each other: which IP prefixes they can reach and what the path is that packets destined to an address in that prefix would (most likely) take. When a border router receives a packet to forward, it will perform a lookup in its routing table and determine where the packet should go next.

In SCION, end hosts construct end-to-end paths through a graph that they construct from the path segments that they receive from their path lookup query. Therefore end hosts can construct disjoint end-to-end paths. Using these disjoint paths allows for improved resilience against path failures. There exist protocols in the current internet, such as multipath TCP [10] and QUIC [11], that can do some limited form of multipath communication. But those are generally limited to the first hop, after which there is no way to guarantee that the end-to-end paths are disjoint. End hosts in a BGP-based internet can not dictate the path that individual packets must take, whereas end hosts in SCION can. Multipath TCP and QUIC therefore can not provide multipath routing to the extent that SCION can, which can construct completely disjoint paths if the topology allows that.

2.4 Anycast

The basic concept behind anycast, as described in RFC 1546 [6], is that there can be an internet service replicated over multiple different (geographical) locations. All of these replicas would

serve the same content, so it does not matter which of the replicas a client connects to. The user should be able to access that anycast service through a single name or address, after which the routing layer or some separate infrastructure makes sure that the client will be connected to any one of the replicas that the anycast service has.

BGP does not restrict network operators to advertise a specific prefix from only a single location. It allows operators to advertise the same prefix from multiple different locations, even from different ASes. This fact is (ab)used to implement anycast using BGP. Routers in the network will then see multiple peers advertise that they can reach a certain prefix and will make a decision on which of those to add to their routing table based on their policies.

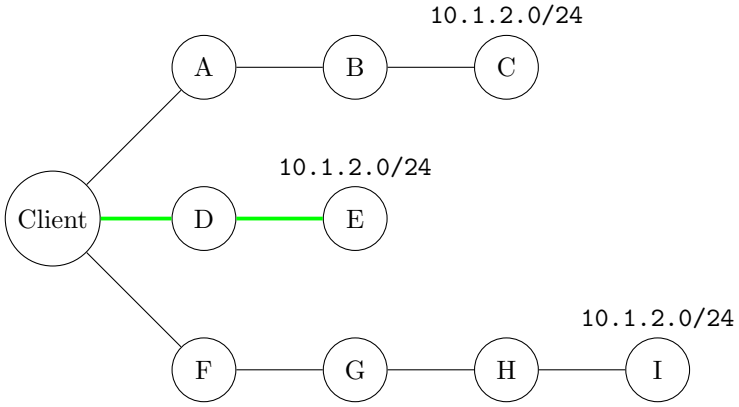


Figure 2.6: An example network topology with an anycast service replicated in ASes C, E, and I. The client ends up connected to the replica at AS E because that can be reached in the shortest number of hops.

Figure 2.6 shows an example topology with an anycast service that has been deployed in ASes C, E and I. All three ASes advertise the same IP prefix (`10.1.2.0/24`) that gets disseminated through the network. When looking from the point of view of the client AS, its border router will receive all three advertisements. Since the three advertisements are for the same prefix, the router will decide based on which has the lowest hop count, if there are no other local policies that have higher priority, which of those advertisements to take into its routing table. When the client sends packets to an address inside of this prefix, they will end up at AS E because that is the shortest path.

As mentioned in the introduction, anycast is currently often used to deploy large scale replicated internet services at multiple different geographical locations around the world. The reasoning behind this is that anycast can provide more performance and is more robust than a single replica deployment. Performance can refer to the performance that clients of the service experience, in latency or bandwidth, while they do not have to decide which replica to connect to. But performance can also refer to the total capacity of the anycast service. Better robustness has to do with the fact that there is no single point of failure. If a certain replica would fail or lose connection to the internet, the underlying routing infrastructure would notice and redirect clients to a different replica.

The term anycast is sometimes used to refer to different techniques implementing the concept of a replicated service or to refer to the concept itself. As mentioned before in the introduction, in the remainder of this thesis, the term anycast will only be used for the specific implementation in BGP as introduced above.

Chapter 3

Design

In this chapter we will explore possible designs for globally replicated services in the SCION architecture. First, in Section 3.1 we will give requirements that sketch an ideal solution. Then, in Sections 3.2 through 3.6 we will explore the possible designs and discuss them, the different considerations and pros/cons that each solution has. Finally, in Section 3.7 we will provide a qualitative analysis of the designs against the requirements.

3.1 Requirements

To be able to design solutions, requirements must first be formulated. These can be used to qualitatively assess different solutions. This set of requirements form an ideal system, it is not necessary for a solution to fulfill all of them.

Requirement 1: Service groups should be globally reachable

The service groups should be globally reachable. This is not meant to restrict service operators of the choice to allocate specific replicas to serve only traffic coming from certain networks. But merely that any design should allow the deployment of a globally reachable replicated service.

Requirement 2: Clients should be (re)directed to the best replica

The design should direct clients to the best replica based on one or more metrics. This metric will often probably be latency, since that is important to allow for good quality of service from the users side. But this requirement does not restrict to having latency as the only metric to base replica selection on.

Requirement 3: Individual replicas should not be overloaded with legitimate traffic

One could argue that preventing overloading of individual endpoints should not be a requirement for replicated services. Events where individual replicas are overloaded with legitimate traffic should be rare, otherwise it would not be a problem of the design but merely a capacity problem for that specific endpoint. However, a counter argument is that even though such events should be rare, it can still be a requirement for a design to be able to mitigate such events.

Requirement 4: The service should be resilient against attacks or illegitimate traffic

The previous requirement could be considered the same as this with the only difference being the type of traffic, legitimate or illegitimate. But for example resisting or combating DDoS attacks is a different problem than preventing legitimate users from overloading a replica, they are different problems requiring different solutions. Therefore it is also useful to analyze them separately and therefore there are two requirements for this.

Requirement 5: The service should be able to recover from failing replicas

There should be some kind of failover mechanism so that a service is able to recover from a failing replica.

Requirement 6: The design adds minimal overhead on clients, routers and control-plane services

The design should be as lightweight as possible to prevent overhead.

Requirement 7: Clients should not have their connection interrupted

Designs should not interrupt active connections. If a client would for example have a long lasting TCP connection with a replica in of the replicated service, it would be good if the client could finish that connection without it being interrupted by for example the underlying routing infrastructure deciding to send the clients traffic to a different replica in the middle of a connection as can be the case in BGP based anycast. In the case that the connection would be interrupted by a change in replica selection the quality of service will probably suffer because the state that was at the originally selected replica is probably not available at the newly selected replica and will have to be transferred or otherwise recovered from.

Requirement 8: The design should not break the core properties of SCION

With this requirement it is meant that the isolation, path transparency and security properties that SCION has should not be broken by any of the designs. If a specific design e.g. breaks with the isolation principle, then that could mean that the replicated service is vulnerable to interference by non-trusted entities whereas regular unicast hosts in SCION are not.

3.2 Multiple Advertisements

This approach was inspired by the BGP-based anycast that is often deployed in the current internet, where the same address block is advertised from multiple different points in the network. It leverages the ability (and purpose) of the path servers to find paths to the requested end host, returning multiple possible paths which the client can use.

The main idea of the solution is that the same AS is *advertised* from multiple (or many) different points. This means that during the beaconing process, many different paths to that AS will be found and registered with path servers.

The main difference between this solution and BGP based anycast lies in the fact that SCION only exchanges information about ASes and interfaces during its beaconing phase whereas in BGP, IP prefixes are advertised to neighbors. This means that BGP allows anycast on the scope of IP address blocks instead of on the AS level as would be in SCION.

Addressing

In the multiple advertisements solution, the same AS number is advertised from different points in the network, thus making the AS replicated and therefore also the services that reside inside of it. A SCION address is a triple of (ISD, AS, address) and does not allow for multiple ISD or AS identifiers in a single address. Therefore to have a single address for all of the different replicas that make up the service, all of the replicas must be put in the same AS that resides in a single ISD. A way to work around this limitation would be to extend the addressing format of SCION, either by allowing multiple ISD identifiers in the same address or a wildcard instead of the ISD identifier.

Putting a wildcard in the address in the place of the ISD identifier would make that the address does not have the hijacking protection through isolation that regular SCION addresses have, thus possibly allowing for hijacking of routes. This means that traffic for that wildcard address can route to any ISD that hosts that AS number in their network, the *rightful* owner of the AS number has no control over which ISDs the traffic intended for their network would end up.

Putting multiple ISD identifiers in a single address would mean that we would get practically the same system as the naming solution described in Section 3.3, where instead of through the naming system, alternate replicas are given in a single address.

The conclusion is that both of these workarounds are not favorable.

ISD considerations

Considering the issues that exist around the addressing described before, replicated AS would be part of a (single) regular ISD that might also have ASes that are not replicated. But it is also possible to have dedicated ISD(s) for replicated services. These could come in multiple different forms.

Operators of big replicated services might want to run their own ISD. These ISDs would then only have core ASes or only a limited number of non-core ASes. The core ASes would then have many peerings with other ISD cores at different geographical locations. Replicated service operators are probably not interested in providing transit for traffic through their ISD, thus they would not propagate beacons that would lead to paths that travel through their ISD being created.

Another scenario could be that there are third parties that operate an *anycast ISD* and provide transit service to customers that want to operate a replicated service. The anycast ISD operator would operate the ISD core ASes and peer those with many other cores. Customers can then peer at multiple locations with (some of) the anycast core(s).

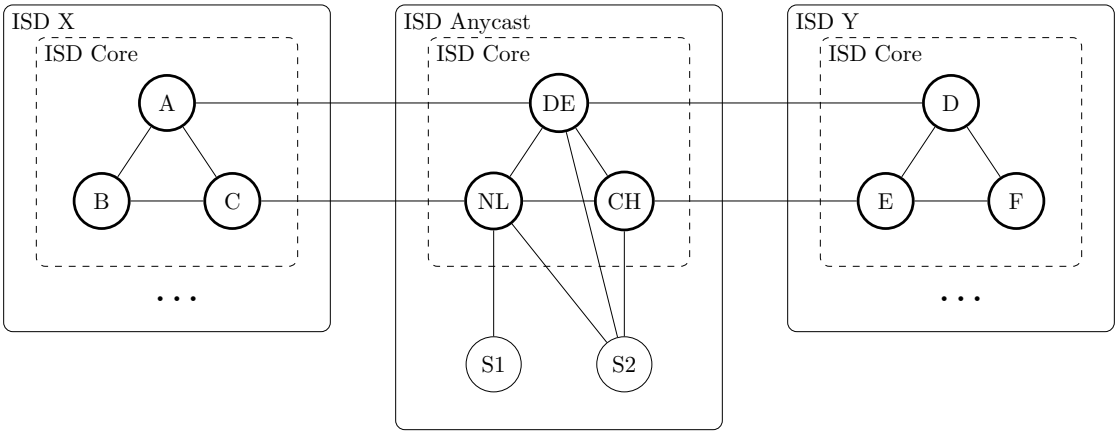


Figure 3.1: Example topology in SCION with a dedicated anycast ISD, the names of the core ASes serve just as an example. The anycast ISD contains two individual replicated services that connect to the cores at certain locations.

Figure 3.1 shows an example of such an anycast ISD. In this example, the ISD is operated by a single entity that manages the ISD core. The core contains multiple ASes, one for each geographical location that the operator is available at. This does not necessarily need to be this way, the core could also be only a single AS with the same outbound connections, but this is just done to make the example easier to interpret. Many services, in this example only two, can then peer with the core at one or more of the locations to deploy their replicated service.

Connection Stability

In the most basic implementation of this solution, the client does not know which paths correspond to which replica. This is not a problem initially, but when the path that the client uses becomes unavailable, they do not know which alternative paths there are to the same replica not allowing for (seamless) continuation of a stateful connection. Also in the case that the client would want to use multiple disjunct paths to reach the destination, this would not be possible.

To alleviate this problem, information about replicated services could be distributed during the beaconing process. The beacon server could e.g. indicate with a flag in the beacon that a certain AS contains replicated service(s), the path server (or the path itself) can then indicate to the client that their connection might be disrupted when they change paths.

Another option is that during the beaconing process, more information about anycast services can be added in the beacons such as an identifier of which endpoint of the anycast service is available at said link. This would allow for the client to know which path(s) correspond to which anycast endpoint(s).

Scalability

Depending on the exact implementation of this design, when a client submits a request to the path server requesting paths to a replicated AS, there could be more path segments that would have to be returned to let the end host make a decision. This depends on how many path segments path servers will return and is something that can be tuned.

The number of up-segments does not necessarily change, but the number of core- and down-segments could increase proportionally to the number of locations that the anycast AS has as well

as depending on the specific implementation of the path servers. This could lead to scalability concerns that might be worth assessing during an evaluation of this solution.

DDoS Protection

This solution does not have inherent DDoS protection for the replicas in the same way as is currently possible with BGP based anycast. In BGP, clients have no influence on which replica their traffic will end up, they even have no information on which replicas are available. In SCION, after path lookup, an end host can construct a graph containing paths to some of the replicas that make up the replicated service.

The size of this graph depends on how many path segments the path servers return upon request; if path servers only return a few out of the total available segments, then the end host will have a limited view. Due to the information that is available to the client in those path segments, the client is able to distinguish between the different replicas at the end of each segment. This could enable an attacker to direct its attack to a specific replica.

This is inherent to the fact that SCION is transparent on paths that are available in the network and this problem will therefore be present in many of the other anycast solutions described below. Luckily, the SCION project also describes COLIBRI, which is a protocol to allow end hosts to reserve bandwidth on a path to attempt to mitigate the effects of a DDoS attack.

3.3 Naming System

This solution uses the naming system to suggest alternative replicas or do the actual selection. This design is inspired by the use of DNS to deploy replicated services [12, 13, 14] as is sometimes done in the current internet. The naming system used to accomplish this could be any, for example DNS or RAINS, the naming system that is designed as part of the SCION architecture.

In this solution, the replicated service is composed of replicas that each have distinct unicast addresses. The authoritative name server for the domain name of the service could then return all or a subset of these unicast addresses for the different replicas. The client can then query the path server for path segments to all (or some) of these unicast addresses.

This solution can therefore also consider information from the service operator, besides the information from the client and the path server. And since every replica can have their own SCION unicast address, the replicas of the service can easily be located in different ISDs and ASes.

Number of Addresses in Response

The basis of the naming solution is that the authoritative name server for the domain name of the replicated service responds with several or many different unicast addresses for each replica that is available. How many and which addresses should be returned is something that can be varied, the name server can choose to simply return a list of all addresses in the response or a shortlist based on the information that it has on the client or the replicas.

The authoritative name server can choose to not include addresses based on information that it has of the replicas. Since the authoritative name server can be operated by the operator of the replicated service in question, the name servers can be made aware of this information. This could be information about load, excluding or throttling inclusion of replicas that are experiencing high amounts of load.

The name server can also choose to return a shortlist of possible endpoints in the name query response based on information about the client. This could be for example information about

the approximate geographical location of the client, but the quality of this shortlist then also depends on the quality of the reference data (e.g. geolocation databases) that the name server uses. In the case of DNS in the current internet, the authoritative name server only knows the IP address of the resolver and that resolver might not be located close to the client in case the client is using a public resolver. The client might end up connecting to a replica that is geographically located far away from him while there may be replicas located close by.

In the current internet, this can be worked around if the resolver includes the clients IP prefix in the DNS extension field described in [15]. In the context of implementation of this in SCION, this will also have to be considered. Another way to work around this is that the name server could choose to return all available addresses if it knows that the query is coming from a public resolver, leaving selection fully up to the client.

In essence, the number of addresses the name server should return does not need to be fixed, and can be varied between different operators that use this design. This solution does not force name servers to return a fixed number of addresses, however there are some scalability concerns when it would return hundreds of addresses as elaborated below.

Path Lookup Racing

Assuming that clients request path segments to all replicas they have received in the answer from the name server, it must wait until it has received answers for all requested paths before it can build the full graph and evaluate for every replica which one of those would suit best. But waiting for all segments to return might take a while if some of the involved path servers are located far away from the client.

However, the fact that some of the path lookups take longer than others can be used as an indication that the respective replicas are probably located far away as well, and that those probably will not end up being selected anyway. Thus the source client could apply some kind of path lookup racing. It could start a path lookup for all the addresses that it has received from the name server and wait until it has received the results for some fraction of these lookups. After receiving enough results or when some timeout has been reached, it could do the end-to-end path calculation based on the results it has gathered thus far.

This is only applicable when the local path server of the source's AS has not cached these far away path segments. When all the segments are cached, path lookup can be done quickly anyway, thus the client can compute the best paths from the full graph. Most of the clients will then not have to do this path lookup racing. When only some of the segments are cached, those segments might only be segments that lead to suboptimal replicas, leading to a suboptimal selection for that specific client. But then again, only the first client or first few will be affected by this, because this gap in the cache will be quickly resolved.

Scalability

If the authoritative name server for the domain name of the replicated service returns many unicast addresses, the end host would have to request paths to many replicas at once. This leads to increased load on the path servers for looking up those paths and at the end host for evaluating all those paths. Scalability must therefore be assessed as part of an evaluation of this solution.

DDoS Protection

This solution would have the same problems with DDoS protection that the multiple advertisements solution that is described in Section 3.2 has. An attacker can find out the individual

replicas and their addresses by making a request to the authoritative name server. If the name server does not return all addresses, it could do queries from different vantage points to try to find more addresses. Then an attacker could use those addresses to launch a targeted attack against a single or multiple replicas.

This also holds for the current internet when using a replica selection mechanism based on DNS. But such an implementation in the current internet would only return a handful of addresses at maximum, since the client will only choose one at random, the selection must mainly be done by the name server. In the design that is suggested in this section for SCION, the name server has more freedom in returning any number of addresses, thus it would be easier for an attacker to enumerate all those addresses.

3.4 Aliases

This section will describe the aliasing design. The basis for this approach is that during the beaconing process or the path lookup process, the client or routing infrastructure can be made aware of alternative ISD-AS pairs or addresses. Essentially, the beaconing server or path server can provide alternate identifiers for what is being requested or exchanged.

This approach can be executed on two different levels, but they both attempt to do similar things and therefore both will be discussed in this section.

Beaconing Server

In this variant of the approach, ISD-AS pairs are the identifiers that are aliased. During the beaconing process, the beaconing servers of the ASes that make up the replicated service will not only share the regular information that is shared in the beaconing process, but will also share alternative ISD-AS pairs that a client could use to substitute in addresses that belong to the AS that is advertising them.

For example, the AS identified by (A, X) in its beacons announces that both (B, Y) and (C, Z) are aliases for its ISD-AS pair. Then when the path server sees path lookup requests coming in for an address in (A, X) it can also simultaneously do path lookup for the same internal address in the other pairs, (B, Y) and (C, Z), that were given in the beacons. All the ASes identified by the alternative ISD-AS pairs will also advertise the others as being aliases, thus they all confirm each other.

Path Server

In this variant of the approach, either full addresses or only ISD-AS pairs could be aliased. This is a choice that is open because at this point in the process, path lookup, the full address that is requested by the client is known. Instead of during the beaconing process, path servers could signal during path lookup that a certain address has aliases.

For example, the path server of the AS where the client is located receives a request for path lookup from the client. It will then fetch an up segment from its local database, and ask the path server at the core of its ISD for the remainder of the path to the requested destination. That core path server will then fetch a suitable core segment from its database and request the path server at the core of the destination ISD for the final down segments. That path server at the destination ISD could then indicate that the requested address has aliases, and then the path server near the client can do additional path lookups to those alternates.

Authentication

However the alias addresses are being provided, they need to be authenticated somehow, but the existing authentication mechanisms that exist in SCION for the beaconing and path lookup processes can be used for this. The AS that *owns* the initial address of the replicated service that is used by the client to access it authenticates the alternates. And because both the beaconing process and the path lookup process authenticate the exchanged information, an attacker will not be able to tamper with these aliases.

DDoS Protection

Similar to the other solutions, the aliasing solution does not have inherent DDoS protection as is the case with BGP based anycast. The client is able to make a final decision on where its traffic will go and due to the transparency of SCION it is able to easily gather information about all the possible targets.

3.5 Application Layer

This section covers some solutions that would use application layer protocols to direct clients to a replica, either using existing protocols or by creating a new protocol. Operating a globally replicated service where replica selection is done in some custom application layer protocol is something that is also done in the current internet [16, 14]. This *design* is more a summary of the considerations around these application layer designs and what SCION could offer them over BGP.

Since these are all solutions that solely use the application layer, they are also possible in the current BGP powered internet. They can also, without any modifications, be deployed in a SCION internet. When comparing replicated services in SCION to the current internet when using these protocols, the results are likely similar if one would just compare SCION to the current internet in a unicast scenario. Therefore, these solutions do not seem interesting to consider when comparing replicated services in a SCION based internet to a BGP based internet.

HTTP Redirect

This solution would mean that clients are redirected to a replica through HTTP redirection. This assumes that the service that is replicated is also using HTTP, or must use HTTP to set up the initial connection. When a client makes the first HTTP request to the service, the initial replica that it is connecting to will evaluate if the client should be connected to a different replica, based on the information that it has, and if so, redirect the client through an HTTP redirect.

If the replica that is initially connected to is not replicated by any other means, which would be the case if this approach would be the only way to set up replicated services in a SCION internet, then initial requests for all clients would have to be handled by the same replica. This will most probably lead to large delays during the first request to the service.

Furthermore, in the most basic implementation, the client has no influence on which replica they will end up connected to. The replica that is making the decision does not have information about latency from the client to each other replica and would have to make its decision purely based on the inferred geographical location of the client.

Manifest Based Redirection

In this solution, when the client is initially requesting a piece of content through the application layer protocol that is used, the replicated service will first respond with a list or manifest of replicas where this content is located. The client can then probe each of these replicas, to measure for example latency, and determine for themselves which replica they want to fetch the content from.

This solution would be useful to services that from a quality of service point can afford initial delays when fetching the manifest and making the selection decision, but that would still like to optimize latency or bandwidth from the point of view of the client and also information about load of individual replicas.

In a SCION-based internet, this solution would be similar to the naming solution in Section 3.3. Where the name server returns a list of possible replicas to connect to, and the client makes the decision, with the help of the path server, on which replica to choose. The main difference between the two would be that the application layer based solution has more flexibility than the naming solution because one would not be limited by the naming system protocol.

3.6 Shared Multicast Trees

This solution is inspired by shared multicast tree systems such as Core-based Trees (CBT) [17]. In shared tree multicast structures, multiple sources would use the same routing tree, that is centered around a single root router, to disseminate information to end hosts that want to receive that. To receive content, end hosts request to join a group by sending control messages through the Internet Group Management Protocol (IGMP) to their gateway, which then forwards these towards the tree core. By doing this also on the other side of the tree towards the sources will eventually lead to a tree that is rooted at the core router.

Applying this structure to SCION, each core AS, or maybe a single core AS per ISD, would operate such a multicast routing core as being another core service. These cores operate as the root of the shared trees. Replicated service replicas will, through a protocol that will have to be designed, advertise themselves to these cores. The protocol will also need to ensure that replicated service operators can find these cores in each ISD or AS.

The cores keep track of all the replicas that advertise themselves to them and keep track of which of these replicas is `best` for each of the individual services. Cores would also have to share this information with cores in other ISDs. Clients can then send their traffic to the core that is part of their ISD, indicating the replicated service that the traffic must be forwarded to, and the core can then forward the traffic to the replica that they designated as best suitable. Response traffic will be sent by the replica back to the core that forwarded the traffic and the core will send it back to the client. Therefore the core will also need to keep track of the open connections.

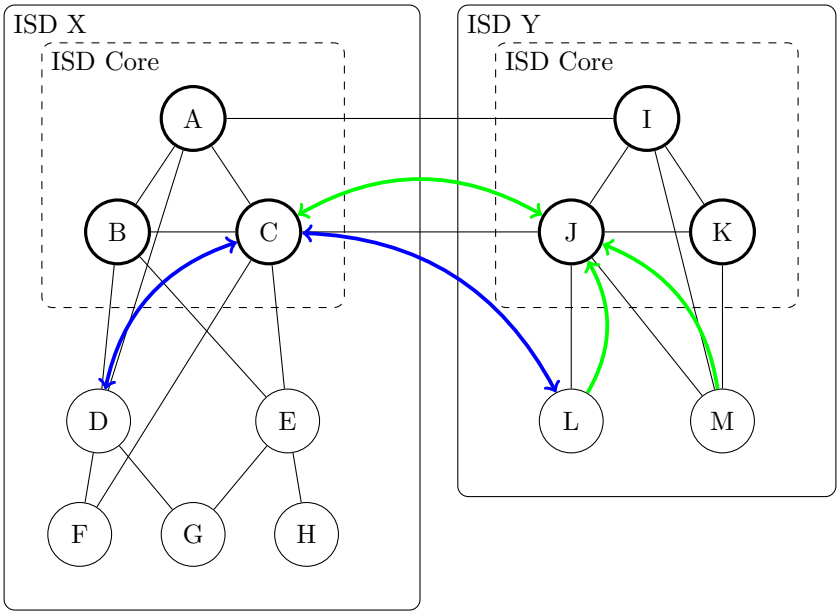


Figure 3.2: Example diagram indicating communication in the Shared Multicast Tree design. Green lines indicating advertisements from replicated services to the routing service in the core, and the blue line indicating communication of a client to the anycast service.

Figure 3.2 shows an example of how this design would work in practice. At first, the replicated service would advertise itself to a multicast routing core in its own ISD, indicated with the green lines from AS L and M to AS J. The multicast routing cores, in this example in AS C and J, would then share information about the replicas that they know. When a client wants to connect to a replicated service, it would send traffic to the multicast routing core in its own ISD, labeled with the service the client wants to reach. The multicast routing core would then select a suitable replica and proxy the traffic to that replica.

This solution also forces every end-to-end path to go through the multicast core, and if that is located in the ISD core it could mean that direct peerings between leaf ASes in different ISDs can not be used as part of these paths. Furthermore, it requires a new protocol to be designed, which can be based on the multicast group membership protocols that currently exist. But this protocol would also need to be designed and validated with security and DDoS resilience in mind. Finally, this solution also builds single points of failure for accessing any replicated service in each ISD; if the core fails, every replicated service will suffer downtime because of that.

In essence, this solution would mean that a new routing system is designed on top of SCION which cannot leverage most of the benefits of SCION.

3.7 Qualitative Analysis

This section contains a qualitative analysis of the different designs and how they compare against the requirements as specified in Section 3.1. Table 3.1 gives a summary of this comparison and more elaboration is given below grouped per requirement.

Not every requirement can be fully analyzed in this comparison, some require additional evaluation. This will then be stated in the relevant elaboration.

	Multiple Advertisements	**Naming**	**Aliases**	**App. Layer**	**Multicast Trees**
Req. 1 Global reachability	yes				
Req. 2 Optimal selection	to be evaluated				
Req. 3 Legitimate overloading	no	yes	no	yes	maybe
Req. 4 Illegitimate overloading	no				
Req. 5 Failover	yes	yes	yes	depends	maybe
Req. 6 Overhead	to be evaluated				
Req. 7 Connection interruption	depends	yes	maybe	yes	maybe
Req. 8 SCION core properties	yes	yes	maybe	yes	yes

Table 3.1: Comparison of replicated service designs for SCION against requirements

Requirement 1: Service groups should be globally reachable

All of the solutions provide globally reachable replicated services. They were designed with this in mind.

Requirement 2: Clients should be (re)directed to the best replica

The goal of each design is to direct clients to the best replicas. But an answer can not be given only based on the design itself, its performance will have to be evaluated to give an answer.

Requirement 3: Individual replicas should not be overloaded with legitimate traffic

The multiple advertisements and aliasing designs do not incorporate information about load on the individual replicas in the target selection. Therefore they are probably not able to prevent overloading of individual replicas during traffic spikes.

The naming and application layer approaches can incorporate information about load into the decision making, therefore they are able to prevent overloading of individual replicas.

Requirement 4: The service should be resilient against attacks or illegitimate traffic

The multiple advertisements and alias designs do not have built-in resilience to DDoS attacks as BGP based anycast has. It is possible for an attacker to specify the replica where they would

want their traffic to be sent to due to the way that SCION is designed. An attacker is also able to figure out which replicas exist in the network due to the fact that path servers will provide the client with some information about this in these designs. This is unfortunately an inherent property of SCION because it is transparent about routes that exist to other end hosts in the network and allows clients to specify which path their traffic should take. DDoS resilience therefore can not be provided by any of the mentioned designs that are built on the network layer and additional protocols or systems would have to be designed to bring DDoS resilience to SCION, such as COLIBRI.

In the naming system design it is still possible to direct traffic over a specific path, thus to attack a specific replica, due to the way that SCION works. The only difference between the assessment mentioned above for the multiple advertisement and alias designs is that some control over the information about which replicas exist in the network is left over to the operator of the replicated service. They hold and provide this information in the name servers and could choose to not share all of the information in the answers to queries that they receive. However, this is only a minimal difference, since an attacker can still direct traffic to any point it wants to.

An application layer design has a similar analysis as the naming system approach. Attackers can probably quickly figure out the list of all individual anycast targets and direct their attack to any specific replica that they would like to attack.

The multicast design might have some resilience to DDoS attacks, because an attacker can not know which replica will end up handling their traffic. However, there is a single point of failure in this design in the multicast cores that have to exist in each ISD. This also means that these multicast cores might end up being attacked instead of the replicas, since when the multicast core cannot be reached, no replicated service can be reached. But since this design will require an entirely new routing architecture to be designed on top of SCION, it is unclear if that routing architecture itself can be made resilient to DDoS attacks.

Requirement 5: The service should be able to recover from failing replicas

In many designs it is possible to quickly retract replicas from the set of replicas making up the service, but not all. The multiple advertisement and alias designs work on the network layer and therefore the paths to the failing replica can simply be retracted to stop any traffic going to that replica.

In the naming system approach, the name server can simply stop answering with the failing address. However, due to the fact that DNS answers might be cached in resolvers during an amount of time specified by the Time To Live (TTL) field, these domain names would require low TTL values to be able to make clear quickly to clients that a certain replica is not accessible any more. The naming design does not have an active retraction mechanism, thus is dependent on the TTL values of the query answers as to how long information that the service would like to retract is still considered valid by the resolver caches.

For the application layer design, the operator can determine where the client would be redirected to, but depending on the type of solution, this might only happen on first connection. If this only happens on the first connection, then it is not possible for the service to actively retract a certain replica from the set of available ones, the initiative for switching replicas would be on the client.

The multicast tree architecture could incorporate some mechanism to retract replicas from the set of available replicas. Since in that design replicas have to actively register themselves with the multicast cores, they could also actively be retracted if this is incorporated in the protocol.

Requirement 6: The design adds minimal overhead on clients, routers and control-plane services

This is similar to the second requirement. Only based on the information given in this chapter and merely through a qualitative analysis, an answer to this question can not be given and would need additional quantitative evaluation. However, this is not part of the scope of the evaluation of this thesis, and therefore will still be an open question.

Requirement 7: Clients should not have their connection interrupted

Clients in SCION have control over the path that their traffic travels between them and replicas. Thus changes in routing topology will not cause the replica selection to change, like they would in BGP. Only in the case of path failure or revocation a different path will have to be selected, and those events are considered as part of this requirement. The design should be able to handle failover to a different replica, without interrupting active stateful connections, if that is necessary.

In the basic multiple advertisement solution without any additional modifications to SCION as mentioned in the connection stability subsection of Section 3.2, clients may not always know if they end up connected to a different replica if they let their traffic follow a different path. It was mentioned that it can be considered to add additional information to the beaconing process to indicate which replica is available at the end of which path, which would alleviate this problem.

The naming and application layer designs allow clients to clearly identify the different replicas and can thus make sure they select a path to the same replica in case of a path failure.

The multicast tree and alias approaches are marked as maybe, this is because some additional mechanism would have to be incorporated in the design to allow clients to stay connected to the same replica in case of path failures.

Requirement 8: The design should not break the core properties of SCION

The multiple advertisements design does not break the core properties of SCION since it does not require major modifications to the SCION architecture to work. The only change would be to optionally include additional information about each link in PCBs such as estimated latency.

The alias design on first sight also does not break core properties of SCION. Even though it adds additional information to the beacons indicating alternate ISD-AS pairs for the clients to use, since that information is placed in the PCBs it is secured. Malicious actors are not able to insert additional aliases. However, this design does not necessarily *neatly* fit into SCION. The route selection code in clients would have to be adjusted, requesting additional paths to the alternate destinations. All-in-all it is a heavy-handed approach and therefore is marked as maybe.

The naming, application layer and multicast tree designs do not involve changing anything about the SCION architecture because they do not operate on the network level, therefore they do not break the core properties of SCION.

3.8 Conclusions

This chapter laid out the designs that were made to tackle the question of deploying replicated services in a SCION internet. A comparison between those designs and the requirements that were written down for a design was also given at the end of this chapter. This comparison showed that there are still some open questions as to some requirements that can only be solved with a more elaborate evaluation, with regards to if the replica selection is optimal and what the overhead on the network of those designs would be.

Chapter 4

Quantitative Evaluation

In this chapter we will elaborate on the quantitative evaluation part of the project. The design comparison in Section 3.7 contained a qualitative comparison between the different designs and also with the requirements as given in Section 3.1. As stated in the design comparison, some aspects need additional quantitative evaluation to give a good answer on how well the designs meet that aspect. In this chapter, we attempt to fill in some of those unanswered questions.

The main goal of the project, as stated in Section 1.1 and the second research question, is to find out how a replicated internet service in a SCION internet would compare to anycast in the current internet. We will therefore focus our evaluation on the differences between a SCION internet and the current (BGP-based) internet.

4.1 Methodology

The evaluation will be carried out by simulation: simulating different setups on different topologies, measuring performance, and then comparing those results. The simulator that is used for this project is based on the ns-3 simulator [18]. The ns-3 simulator has been extended with primitives and several simulation setups for simulating the beaconing process in SCION. This SCION beaconing simulator has been kindly provided to us by by Tabaeiaghdaei et al. For the purpose of this evaluation, the simulator has been further extended with path server mechanics and some mechanics for simulating replicated services. More detail on the simulator is given in Section 4.1.3. Section 4.1.2 will go in more detail on the simulator design and its assumptions.

4.1.1 Simulation Aspects

There are several different aspects that we are going to look at in the simulations and we will compare their effects on the performance of anycast deployments in both BGP and SCION. Below we will elaborate more on these aspects. Different simulation scenatios will be built combining one or more of the mentioned aspects. The results in Section 4.3 will mention the scenario that was used for each of the simulation results and why that scenario was chosen.

Replica Selection Method

The first aspect is the method of replica selection. This is obviously the most important aspect since this is the major defining aspect between BGP and SCION. Replica selection in BGP is mainly influenced by hop count between the two end hosts and in SCION it is mainly influenced

Topology	Number of nodes	Number of edges	Density
us	7028	8317	0.00034
eu	1383	1439	0.00151
ru	272	314	0.00852

Table 4.1: General information on the realistic topologies that were used in the simulations.

by latency, as has been elaborated on in the designs in Chapter 3. We therefore simplify the replica selection methods for BGP and SCION to selecting based on only hop count and only latency between end hosts respectively for the two architectures.

Section 4.1.2 below contains some more consideration on the limitations of this simplification.

Network Topology

Another aspect that we will look at is network topology. One major difference between the way that BGP works and SCION works is that in SCION, there is a clear hierarchy in the topology due to the introduction of ISDs (Isolation Domains). In practice in the current internet based on BGP, there exist many hierarchical relations between networks in that there are customer-provider relationships between ASes. In theory there does not need to be such a hierarchy in BGP.

The way we take this into account in the evaluation is by feeding the simulator with different sets of topologies. These sets of topologies fall into two main categories: real-world derived (or realistic) topologies and randomly generated topologies.

Realistic Topologies

The realistic topologies are derived from data provided by CAIDA in their internet topology research [19]. Using the complete topology of the global internet is not feasible since the simulator will require too much resources to be able to process that in a reasonable time. Therefore the global topology has been split into three parts based on the geographical location of each network. Table 4.1 contains some general information on the topologies that were used for the simulations.

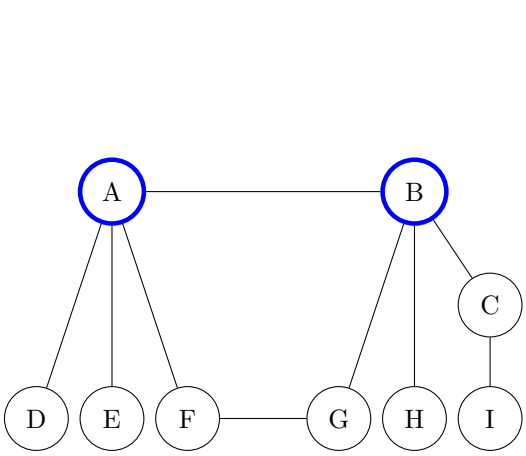
However, since these topologies are derived from real-world internet measurements, they represent the topology of an internet based on BGP and that does not necessarily have to look the same in a SCION-powered internet. Therefore it is also necessary to evaluate what would happen to replica selection performance if the realistic topology was split into ISDs. Simulations will be carried out on both the base and split sets of topologies and those results will be compared against each other to see what the effect of the splitting is.

Splitting the topology is done with the following method. First, we determine which ASes in the topology are most central by looking at the number of peers that each AS has. Then we take the highest ranked ASes from that analysis and assign each to a different ISD. Each of these highest ranked ASes will also be a core in their ISD. Then we determine the assignment of all the other ASes to an ISD by looking at the distance between them and the highest ranked ASes. After each AS has been assigned to an ISD, we determine if there are other ASes in each ISD that are high ranked and also mark them as cores. These secondary cores also need to have at least one peer outside of their assigned ISD. Finally, we remove every link that crosses an ISD border and is not between cores.

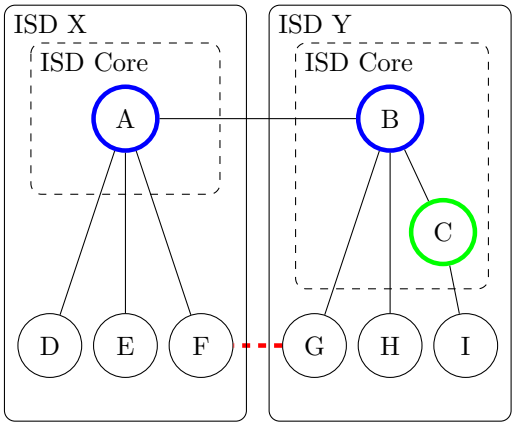
Sometimes this process leads to an unconnected graph if a core AS did not have any direct connection to cores in other ISDs. If this is the case, we look at the two hop connections that

exist in the original topology between every pair of core ASes and fix the topology by adding those connections to the split topology. With this, the split topology will be a connected graph with multiple ISDs and only contains inter-ISD links between core ASes.

This process is illustrated in Figure 4.1, which contains a small example topology that is being split up into two ISDs. First, the most central ISDs in the topology are determined and marked, those are networks A and B. Then, for each other AS we will determine which marked AS is the closest in number of hops and add the AS to that ISD, this is done in Figure 4.1b. Figure 4.1b also shows the other ASes that are marked as cores due to their centrality in the ISD (in green) and the links that will be deleted because they are between non-core ASes in different ISDs.



(a) Determining the most central ASes



(b) Assign ISDs and remove non-core inter-ISD links

Figure 4.1: Small example topologies to show how the ISD splitting method works.

Randomly Generated Topologies

The set of realistic topologies is a subset of the infinite set of all possible topologies. There will be certain patterns or graph characteristics that appear more often in these realistic topologies than they would in a set of random topologies. Furthermore, the topology of an internet based on SCION will probably not look the same as the topology of the current internet. Therefore we will also carry out simulations on randomly generated topologies, complementing the evaluation on realistic topologies. Doing this we will be able to give a better answer to the question of which type of internet is better suited for replicated services: a SCION internet or a BGP internet.

The reason for simulating with random generated topologies is to figure out if there are certain patterns or characteristics of topologies that influence performance that do not appear in the set of realistic topologies. Topologies that are derived from real life networks such as the internet are always going to be biased, having certain patterns or characteristics appearing more often than they would in the (super)set of all possible topologies.

These random topologies are generated by several different methods using the NetworkX Python library [20]. The focus will lie on finding topologies where performance differs significantly from the average and finding out if there are certain patterns or characteristics that cause this.

In the results of Section 4.3 each randomly generated set of topologies will carry a name indicating the method that was used to generate them. Each random topology set is prefixed with *random-*, followed by the generation algorithm (*geometric-* or *erdos-*), and can be further

suffixed to indicate some specific parameters were used in the method.

The `random_geometric_graph(n, radius, dim=2)` algorithm generates a random geometric graph. This algorithm will randomly place n nodes on a *dim*-by-*dim*-sized plane and create links between nodes if the distance between them is at most the *radius* parameter. These graphs were generated with $n = 100$ nodes, a radius of $radius = 0.2$, and default plane dimension.

The `erdos_renyi_graph(n, p)` algorithm generates a random Erdős-Rényi graph. It will contain n nodes and p specifies the probability that a link will be created between each pair of nodes. These graphs were also generated with $n = 100$ nodes and a base probability of $p = 0.2$.

The *-middle* suffix indicates how the weight or latency on each link is determined. When this specifier is present in the name of the topology set, it means that the latency on each link is determined by the placement of the nodes. Each node will be placed on the earth randomly (in the case of geometric graphs this is already done by the algorithm), and the latency of the link is computed by the propagation delay that a network cable spanning these points would have. When this specifier is absent, it means that each link will have a random latency assigned to it. This is done by picking two random locations on the earth and also computing the propagation delay of a network cable.

The *-sparse* and *-vary* suffixes is used to specify graph sets with different densities. These suffixes are only used for the Erdős-Rényi graphs. Sparse graphs have their probability parameter set to $p = 0.05$ and graphs with varying probability have theirs set to a random value between $0.03 <= p <= 0.2$.

Path Server Configuration

Another aspect that is expected to influence performance is the configuration of the path servers. In SCION, when a (source) host wants to send traffic to another (destination) host in the network, the source would have to find a path to the destination by performing a path lookup at its local path server. When the lookup process is finished, the path server will return path segments to the source for each leg (up, core and down) of the end-to-end path.

If the client would receive every possible segment between itself and the destination, the client would be able to select the most optimal path. But this could also mean that a lot of traffic would have to be sent if there are many path segments, which could lead to scalability issues. Therefore, in SCION, path servers return only k segments to a query, so k up-, core- and down segments each. As mentioned in the SCION book in Section 2.1.2 [4], k should be a small integer, but this might be configurable for each path server.

How many and which path segments are returned by the path servers is expected to have an influence on the performance of the replicated services. Therefore additional simulations will be done by simulating the SCION selection method and setting k in the path servers to different values between 1 and 10.

4.1.2 Simulation Limitations

This evaluation will not be able to give a complete answer to the second research question, how SCION compares to BGP with respect to replicated services, because some aspects that play a role in the performance of real life deployments are left out and several other aspects are simplified. We will discuss these limitations and simplifications below.

Replica Selection

For SCION, the simulation is simplified to selecting a replica based on latency. Clients are able to see, for the path segments that they receive from the path server, the expected latency

for each link. Therefore they are able to select a suitable replica based on that. The Multiple Advertisements and Naming designs both have in common that they connect clients to the replica to which they have the lowest latency. The path servers in the network are expected to have some influence on the quality of the selection depending on how many and which path segments they will return to the client.

As for BGP, the simulation is simplified to selecting a replica based on hop count, local policies will not be taken into account. In reality, routers in between the client and server make a decision for each packet on what link they will send it out on, based on their routing table. Apart from policies defined by the operators of the routers, the only objective metric that routers have is the hop count that is derived from the AS path in the advertisements. Therefore, if there are multiple paths in the network that have the same number of hops, we use the average latency of the paths with the lowest hop count as the average latency on the path between client and replica. This essentially divides the traffic between all least-hops end-to-end paths, the same as BGP would behave in theory.

Local Network Policies

Aspects regarding policies set by AS operators are not included in the evaluation on purpose. Examples of this are: SCION policies on which beacons to (not) import, BGP policies used to build routing tables, path prepending in BGP and BGP route aggregation. Especially in BGP, it is not possible to see most of the policies that the different AS operators use and in which order. The policies are used to build routing tables, but they are not visible to the outside world. It would be hard to get a realistic view over these policies since they are not public. Therefore local policies are left out of the simulations.

Dynamic Parameter and Events

There are some dynamic parameters that will affect performance in real life scenarios. These are things such as load, congestion and changes in the network topology. The simulations that we carry out are static in that sense, they do not incorporate these factors. This was done to keep the simulations simpler to save time in the project.

Simulator Base

In this thesis, we use one simulator to do both the BGP and SCION simulations, namely the extended SCION beaconing simulator. Different simulators will have different assumptions in how they are built, the results might not be valid or would be harder to validate if we would use different simulators for BGP and SCION. Therefore we will use the extended SCION beaconing simulator for both protocols.

If a topology without ISDs is put into this simulator, it will be able to find all end-to-end paths in the topology and the hop counts associated with them. Therefore it is possible to emulate BGP by using the SCION simulator by supplying a topology without ISDs and doing path selection based on hop count. By emulating BGP with the SCION simulator, we eliminate the risk mentioned above, having different simulators that make different assumptions possibly leading to invalid results.

Summary of Limitations

Concluding, these limitations mean that this quantitative evaluation is not able to give a complete answer to what realistically would be the performance of SCION and BGP. However, it is able

to sketch the theoretical maximum performance of both of these systems and with that we will be able to give at least some key insights.

4.1.3 Simulator Structure

We will now describe the structure of the extended SCION beaconing simulator and the measurements that it does. The simulator is made up of two parts or stages, the SCION beaconing simulator and the replica selection simulator extension.

First, the beaconing process of SCION is simulated using the SCION beaconing simulator. The beaconing process in SCION is used to find path segments from core ASes down to non-core ASes (and opposite), and between core ASes in other ISDs. This information is then put into the path server of each individual AS. The implementation of the beaconing simulator is according to the specification in the SCION book [4]. Input to this part of the simulation is a topology and some parameters: beaconing interval (5 seconds), beacon time to live (3 minutes), and total simulation runtime (60 seconds). The beaconing simulator will then output all of the path segments that were present in each path server at the end of the simulation.

Then, the results of the beaconing process are put into the replica selection simulator. The other parameters to this simulator are the method for replica selection (BGP or SCION) and the list of locations of the replicas, or rather the list of AS numbers that make up the set up replicas. This number is set to $r = 10$ for the realistic topologies and $r = 5$ for the randomly generated topologies. The number of replicas for the randomly generated topologies is lower because there are less nodes ($n = 100$) than in the realistic topologies.

This replica selection simulator will then simulate a client connecting to the replicated service from each AS in the topology that does not contain a replica. It will then measure performance by taking the one-way latency between the client and the selected replica. These measurements are the output of the replica selection simulator.

4.1.4 Summary

Now we will summarize the methodology and list which aspects are going to be simulated in which combinations.

The main aspect of the simulation is the replica selection method, BGP or SCION, which was simplified to selecting the closest replica based on the number of hops or the closest based on latency. Then we described two sets of topologies, real-world derived (or realistic) topologies and randomly generated topologies.

With the realistic topologies, we are going to simulate the baseline performance of both BGP and SCION. Then we also simulate the effect that splitting the topology into ISDs would have on performance. Finally, we are going to look at what effect the path server configuration has on the performance in the realistic topologies.

For the randomly generated topologies, we are also going to simulate the baseline performance of both BGP and SCION. With the different generation methods we can analyze the performance in relation to different properties or patterns of the topologies.

4.2 Validation

It is important to verify whether the simulator behaves how it should. To validate the results that are generated by the simulator extension, we have written a test suite for it. This test suite contains unit tests for each component or code function of the simulator and it also contains integration tests that validate the output of the entire simulator.

For the integration tests, we wrote tests to check the basic functionality of each function or component and also for any edge cases we could identify. Furthermore, we created some topologies to serve as the input: a complete graph, a (star-shaped) tree, a spiderweb shaped graph and a graph with all vertices forming a line. These graphs are relatively small, between 5 and 30 vertices, and we computed the expected result by hand and checked those against the results of the simulator. We also generated three random topologies that we ran through the simulator and verified the results for afterwards and then included those in the set of integration tests.

4.3 Results

In this section we will elaborate on the results of the simulations, as well as on the details and parameters of the simulations. First, we will elaborate on the measurements that we take and what we expect from the simulations. Secondly, we will present the simulations on the realistic topologies, the baseline simulations and the influence of the path server configuration. Then, we will present the results on the randomly generated topologies, diving further into different properties of topologies that have an influence on performance. We will then finish with a discussion about the results and what they mean for the designs from Chapter 3 and the research questions.

4.3.1 Performance Measures

The main performance metric that we are going to use in all of the results below is the one-way end-to-end latency between the client and the replica that was selection by the replica selection method. Doing these measurements from every AS in the topology enables us to show how good on average the experience of the end-user is going to be with a replicated service in either BGP or SCION.

Since we measure performance in latency and the SCION replica selection method is also using latency as its selection criterium, we expect that SCION is always going to select the best replica. BGP, since it is using hop count as its selection criterium, is sometimes going to select the best replica, sometimes not, how often is to be seen. The question of how well does SCION perform can thus also be rephrased to: how good of an estimator is hop count for latency.

4.3.2 Realistic Topologies

In this subsection we will elaborate on the results of the simulations that were done on realistic topologies. First, we will clarify on the parameters that were used in these simulations. Then, we will present the results of the baseline simulations. And finally, present the results of the simulations on the path server configuration.

Simulation Parameters

The beaconing simulator is using the same parameters as described in Section 4.1.3. The number of replicas that are placed in the topology is $replicas = 10$ for these simulations on realistic topologies. Furthermore, we are doing multiple runs ($runs = 100$) for each scenario, using a different set of replicas in every run. However, we always use the same set of replicas across the different scenarios to ensure that the environment is exactly the same for each different scenario.

Baseline

For each of the topologies presented, results of three different simulation scenarios are presented. The *BGP* setup is a combination of the hop count replica selection method simulated on top of the base topology. Then, the *SCION unsplit* setup uses the same base topology, but with the SCION replica selection method, taking the lowest latency path. Path servers are not yet needed in this setup and therefore not simulated. Finally, the *SCION split* uses the split topology, generated with the method described in Section 4.1. In this scenario, path servers are configured to return every matching path segment upon querying.

The SCION unsplit scenario functions as a control setup in these simulations. We expect the performance of any other SCION based scenario to be worse than this. The other SCION based scenarios will have aspects such as path servers and the splitting of topologies in ISDs taken into account. And due to the fact that the topology splitting only removes some edges in the topology, we do not expect better performance on average on the split topology. Also with regards to the path server, when returning every matching path segment upon querying, a path server is not expected to have any positive nor negative influence on performance. When a path server is configured to return less than all segments for queries it is not expected that performance would improve, due to the fact that the client has less choice over the replica that they select.

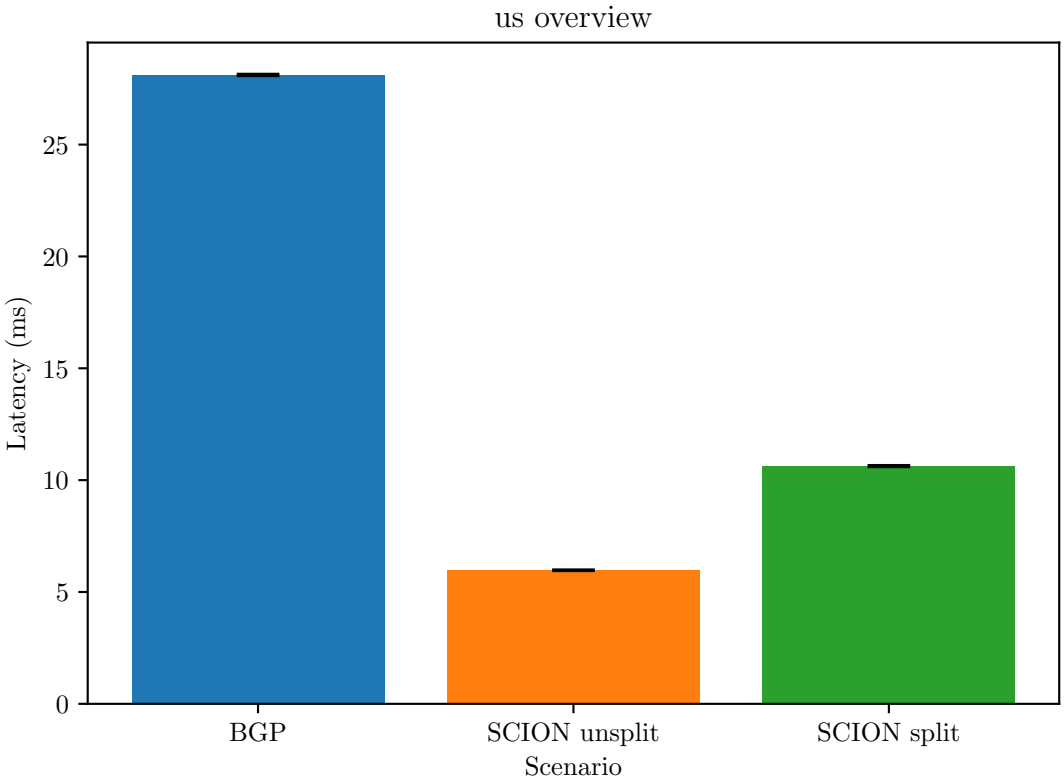


Figure 4.2: Overview of the mean and confidence intervals for the different simulation setups on the us topology, showing the mean performance of each setup and the corresponding 95% confidence interval. The small black bars on top of the colored bars are the confidence intervals.

Figure 4.2 shows a summary of performance for the different simulation setups on a single topology. In this case the *us* topology. As mentioned before, performance for these simulations is the latency between every client and the replica that the selection method used in the setup chooses for them. All of the measurements on all of the different runs are aggregated into a single average which is presented in this graph, showing the mean performance and the 95% confidence interval.

What can be seen from this is in general how the different setups affect performance of replica selection. And this example shows that the SCION unsplit seup performs best and the BGP setup performs worst. The SCION split setup performs slightly worse than the unsplit variant. Table 4.2 shows the same results, only then for all topology sets and only the mean performance of each setup.

We will now dive a bit deeper into the measurements on the us topology, and show for each individual client, if they experience better, same or worse performance in the different setups. This is presented in Table 4.3 for the us topology.

This comparison for example shows that no client experiences better performance in the BGP setup compared to the SCION unsplit setup, and also that almost all clients experience worse performance, however big or small that may be. Only a fraction of the clients experience the exact same performance. This can be explained by the fact that the us topology is more dense

Topology	BGP (mean)	SCION unsplit (mean)	SCION split (mean)
us	28.1 (95% CI [28.1, 28.2])	6.0 (95% CI [5.9, 6.0])	10.6 (95% CI [10.6, 10.7])
eu	24.9 (95% CI [24.8, 25.0])	5.5 (95% CI [5.5, 5.6])	7.1 (95% CI [7.1, 7.2])
ru	3.2 (95% CI [3.1, 3.3])	1.1 (95% CI [1.0, 1.1])	1.4 (95% CI [1.4, 1.5])

Table 4.2: Summary of the results on realistic topology sets, showing mean performance for different simulation setup and topology combinations.

Setup 1	Setup 2	Setup 1 better than setup 2	Setup 1 same as Setup 2	Setup 1 worse than Setup 2
BGP	SCION unsplit	0.0%	3.4%	96.6%
BGP	SCION split	4.7%	3.4%	91.9%
SCION unsplit	SCION split	46.9%	49.4%	3.7%

Table 4.3: Summary of comparison of individual measurements over different setups on the us topology.

than the other realistic topologies. The hop count selection method will have more chance to select an unfavorable path and therefore performance is in most cases worse.

When looking at the other topologies in Tables 4.4 and 4.5, this clear difference does not appear. These topologies are more sparse than the us topology and therefore more often have similar performance. Also when looking back at Table 4.2, it can be seen that the ratio of performance difference between the SCION unsplit and BGP setups is smaller for the eu and ru topologies.

Furthermore, if we look at the comparison of performance between the SCION unsplit and SCION split setups, it is in all topologies sometimes the case that the SCION split setup performs better. This is not as expected but can be explained by the fact that when splitting the topology, we had to sometimes correct a topology where one or multiple ISDs ended disconnected from the rest of the network. This might then accidentally lead to better performing paths being added to the network that did not exist in the original topology.

Setup 1	Setup 2	Setup 1 better than setup 2	Setup 1 same as Setup 2	Setup 1 worse than Setup 2
BGP	SCION unsplit	0.0%	11.5%	88.5%
BGP	SCION split	3.5%	10.3%	86.3%
SCION unsplit	SCION split	30.7%	63.6%	5.7%

Table 4.4: Summary of comparison of individual measurements over different setups on the eu topology.

Setup 1	Setup 2	Setup 1 better than setup 2	Setup 1 same as Setup 2	Setup 1 worse than Setup 2
BGP	SCION unsplit	0.0%	43.6%	56.4%
BGP	SCION split	3.8%	42.1%	54.1%
SCION unsplit	SCION split	10.8%	88.0%	1.3%

Table 4.5: Summary of comparison of individual measurements over different setups on the ru topology.

Path Server Configuration

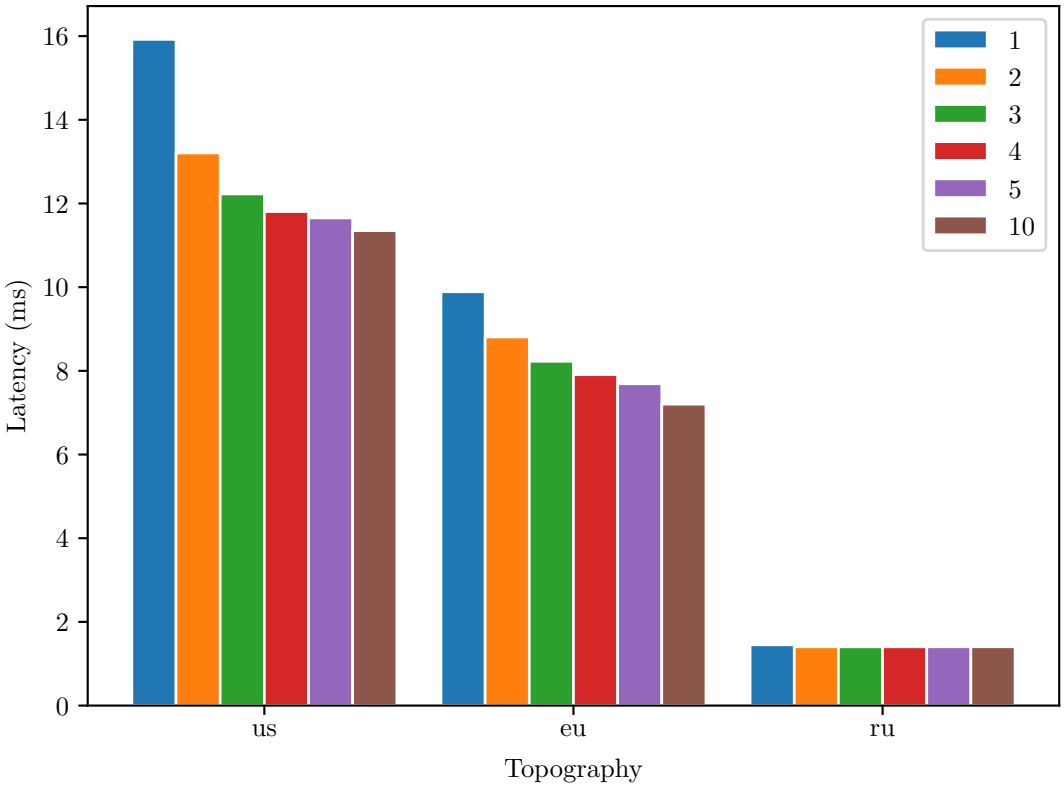


Figure 4.3: Overview of the influence of path server configuration on performance. Each set of same-colored bars indicates the maximum number of segments returned by each involved path server per topology, indicated on the x-axis.

Figure 4.3 shows what influence the path server has over the performance of replica selection depending on how many segments (k) it is configured to return. The figure contains a set of colored bars for each topology, where the colors indicate the configuration value. As can be seen in the figure, the difference in performance heavily depends on the topology that is used. In the ru topology, the path server configuration has next to no effect on performance, but for the other

topologies it does.

4.3.3 Randomly Generated Topologies

This section contains the results of the simulations on the random topologies. The main goal of this part of the evaluation is to figure out how topologies influence replica selection. We will start by presenting some basic results from each of the different sets of topologies and then dive in further to see why they give different results.

Simulation Parameters

The beaconing simulator again is using the baseline parameters as described in Section 4.1.3. The number of replicas that are placed in the topology is only $replicas = 5$ for the random topology simulations, due to the fact that there are only $n = 100$ nodes in each topology. The different topology sets that are present in the results are described in Section 4.1. Each topology set contains 300 randomly generated topologies. Since we have many topologies in each set, we are only doing a single run on each topology. We are not splitting the topologies into ISDs for these simulations, only running replica selection on the unsplit versions.

Basic Results

Similar to the simulations on realistic topologies, first the results of the *control* simulations are presented. These control simulations are built from two different simulation setups: the BGP setup and the SCION setup. Both of these setups are run on the same topology, and with that the only difference between these setups is the replica selection method: hop count (BGP) and latency (SCION).

Similar to the realistic topologies, it is not expected that the SCION setup will perform worse on any measurement in any topology than the BGP setup. This is due to the fact that replica selection in the SCION setup is done by selecting the replica with the lowest latency instead of hop count. And when the performance metric is also latency, the SCION setup is expected to always select the best performing route in the network. However, it is interesting to see how the BGP setup performs compared to this, mainly to see how much worse the theoretical upper limit of performance of both replica selection methods is.

Table 4.6 shows the overview with some general statistics about the replica selection methods on different types of random generated topologies. The mean is computed by first taking the average latency of all measurements for a single topology, then taking the average over the averages of all topologies. As can be seen in this table, the SCION setup never performs worse than the BGP, which is as expected. However, the ratio of difference in performance is different between the different graph generation methods.

Figure 4.4 shows a *summary* graph of the performance of the different simulation setups next to each other for the different runs. The measurements of each run are aggregated and the average is presented in the graph. Each x-coordinate corresponds to a single topology. For each of the topologies, the performance is evaluated for both simulation setups, BGP and SCION. The mean and confidence interval for those runs are presented in the graph. Data points for different setups but that are run on the same topology are thus on the same x-coordinate, so on top of each other. So this summary graph attempts to present a summary of the performance of the different simulation setups on the same topology set.

From this summary graph, it can be seen that in none of the runs, the BGP setup on average performs better than the SCION. There is a small amount of overlap in the confidence intervals

Graph generation method	**BGP (mean)**	**SCION (mean)**	**Ratio BGP vs. SCION (means)**
random-geometric	45.7	23.8	2.0
random-geometric-middle	16.0	13.0	1.2
random-erdos	16.5	3.2	5.3
random-erdos-middle	11.3	3.4	3.4
random-erdos-sparse	51.9	34.2	1.5
random-erdos-sparse-middle	35.5	23.0	1.5
random-erdos-vary	33.2	16.4	3.0
random-erdos-vary-middle	22.2	11.5	2.5

Table 4.6: Overview of the results of the control simulation setups on random topologies.

in some of the runs. This is as expected, the SCION setup will always show the best possible performance, the BGP setup uses hop count as a selection method and is therefore not perfect.

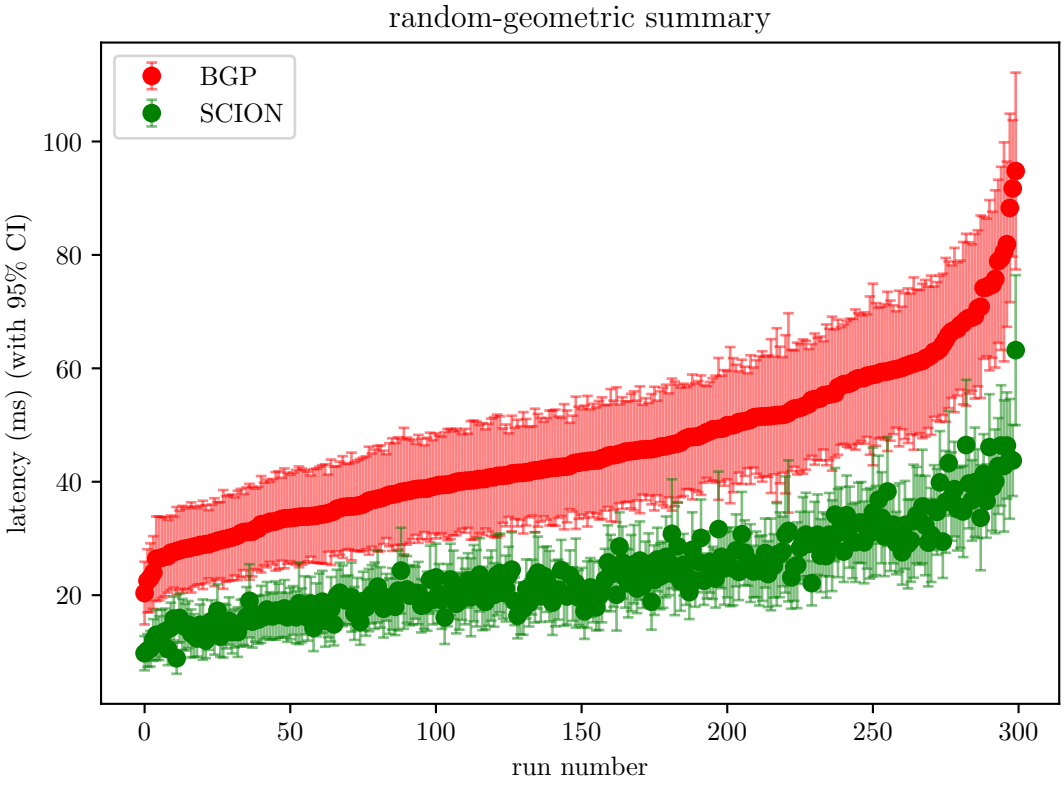
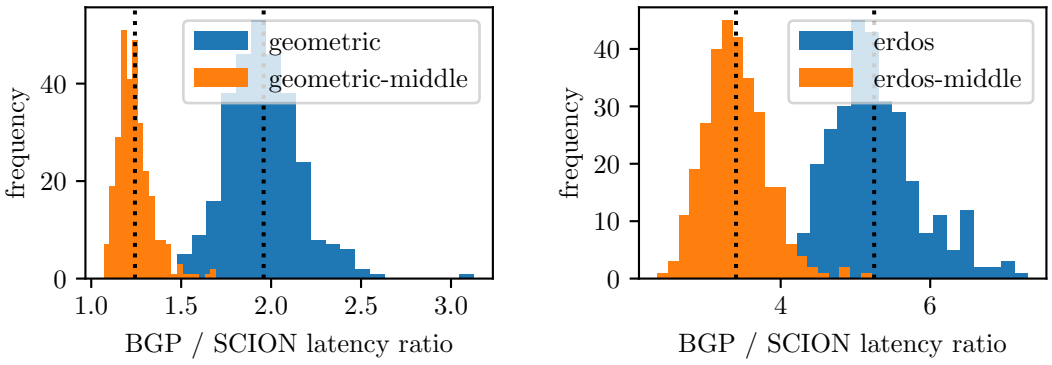


Figure 4.4: Overview of the performance difference between simulation setups. Each point and corresponding confidence interval bar shows data for a single topology. There are two data points for each topology, one BGP and one SCION. Data points for the same topology are put on the same x-coordinate. The data points were then sorted based on their BGP mean latency to make the graph more readable.

Differences Between Graph Generation Methods

As mentioned before, Table 4.6 shows that under some graph generation methods, the BGP setup performs much worse compared to the SCION setup. When looking into this in more detail, two reasons can be found that can explain this difference. These are: the graph generation method and the density of the generated graphs.

First, when comparing performance of the different graph generation methods, the main parameter that seems to influence the relative performance of the two setups is the way that geographical position of peering locations is determined. This difference can be seen clearly by comparing the spreads of the ratio between the performance of the two setups and is presented in Figure 4.5. In these histograms, the ratio between the performance of the BGP and SCION setups is shown. Each data point is a ratio between corresponding measurements in both setups.



(a) geometric

(b) erdos

Figure 4.5: Histograms for several graph generation methods of ratio of performance difference. The data points in these histograms are the performance of the BGP setup divided by the performance of the SCION setup for each measurement.

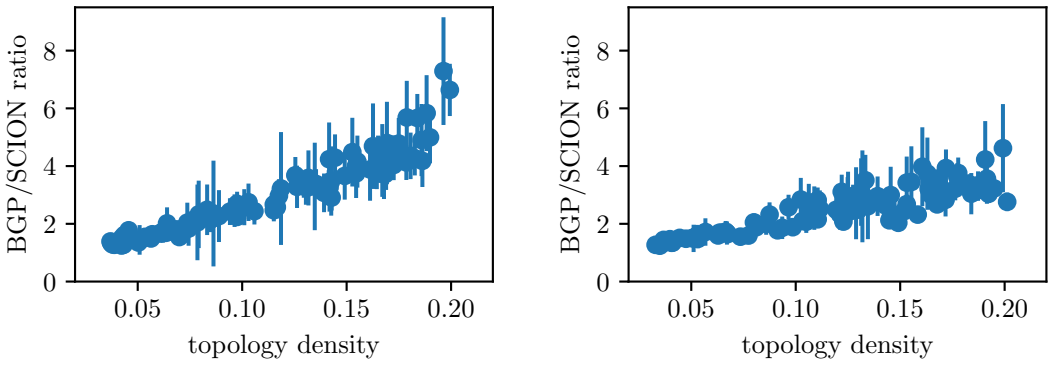
As can be seen in the histograms, and also in more summarized form in Table 4.6, the performance ratio is consistently lower in the topology sets that are suffixed with *-middle*. The difference between the *base* topology and the *middle* topology sets is the way which the geographical locations of the links in the topologies are determined.

Section 4.1 describes the different random graph generation methods. Repeating in short what was stated there, the random topology sets with *-middle* as suffix compute the latency of each link by looking at the location of the networks at both ends. The topology sets without the suffix give each link a random latency.

From the results in Figure 4.5 and Table 4.6 it seems that hop count is a better estimator of path latency in graphs where the latency of those links is derived from the nodes it it connecting. This probably also counts for any topology where the location of the links is not completely random.

Topology Density

Another simulation parameter that seems to heavily influence the ratio of performance between the BGP and SCION simulation setups is the (edge) density of the topology. In dense topologies, the ratio between the setups is lower than in sparse topologies. This is shown in Figure 4.6, where scatter plots of the density versus the mean performance ratio is shown for two toplogy sets. Both these graph generation methods generate random graphs with uniform random varying densities in the interval $(0.03, 0.2)$.



(a) erdos-vary

(b) erdos-vary-middle

Figure 4.6: Scatter plots of topology (edge) density versus setup performance ratio of BGP and SCION setups. Including 95% confidence intervals.

Figure 4.6 shows some correlation between topology density and the ratio of performance of both setups. The aggregation of measurements in these graphs is done by taking the ratio of the latency on each client first, then aggregating that into a mean and confidence interval for each topology.

An explanation for this is that in dense topologies, there are more paths between two nodes than in sparse topologies. More paths means more options to choose from for the replica selection method. This means there is more chance that a certain path is attractive due to low hop count but in fact has higher latency than another path with more hops. The BGP setup will always select the path with lowest hop count and the SCION setup will always choose the path with lowest latency. This leads to the phenomenon shown in Figure 4.6, correlation between topology density and ratio of performance of the setups.

4.4 Discussion

In this section we will elaborate more on the results presented above and discuss what their effect is on the answers to the research questions. This section is split into multiple topics that we will discuss.

Relation to Designs

The simulations setups that we built were not exact implementations of any of the designs from Chapter 3. Therefore we must discuss how these results relate to the designs.

The setups are most relevant to the Multiple Advertisements design, which is close to how anycast works in BGP. Advertise the same block of addresses, or in the case of SCION the same AS, from multiple different physical locations and let the routing protocol handle the replica selection. So the results that we gathered are going to mostly be relevant to the Multiple Advertisements design.

The Alias design is relatively close to this, with some additional overhead on the pathfinding phase. Thus the results are also relevant to that design.

In the Naming design, instead of a single address, the client will get multiple addresses from its name resolution query, and will then to path lookup to all those. If the client would

get the addresses of all replicas in the answer, from then on replica selection would happen in practically the same way as the Multiple Advertisements design, selecting the replica with the lowest latency. However, this will most likely not be the case, with only a small number of addresses in the answer. Thus the results are somewhat relevant to the Naming design, but significantly less than the Multiple Advertisements and Alias designs.

In the Applicaion Layer and Shared Multicast Trees designs, the SCION is not involved in the replica selection at all. And since the results are measuring performance of BGP and SCION as routing protocols, the results are not relevant to those designs.

Replica Selection Method

When comparing the replica selection methods against each other, selection based on latency (SCION) generally performs better than selection based on hop count (BGP). We can see this in all figures and tables in this chapter. The magnitude of the performance difference between the two selection methods differs depending on the topology that has been used as input of the simulation. Most notably Figure 4.4 and Table 4.6 comparing the results between the two selection methods show that the performance difference between the selection methods is not always the same. But a conclusion from these findings is that selection based on latency is thus clearly better than selection based on hop count.

Another way of looking at these results is to note that we select the best replica based on hop count and latency and measure performance of both selection methods based on latency. Therefore, the latency based selection method (SCION) is always going to result in the optimal selection whereas the hop count based selection method might not select the optimal replica. Thus, a better way of describing these findings is that selection based on hop count is significantly worse than selection based on latency.

Topologies

Looking at the results that from the random topology simulations, we can see that it depends on the topology how well BGP performs compared to SCION. Figure 4.6 shows that it partially depends on the density of the topology how big this performance difference is. On average, the denser a topology is, the better SCION performs relative to BGP. Also, when looking at the overview of realistic topologies in Table 4.2, the ru topology, which is the most sparse of the set, shows the smallest difference in performance. Thus, this means that topology density plays a big role in the performance difference between BGP and SCION.

However, this also means that it is hard to predict if in any real-life deployment, SCION would perform better than BGP. It is hard to predict what the topology of a real-life SCION network will look like. ISDs could be formed for each country and ASes would have to join the ISD where they do (most of) their business. It could be that ISDs are going to be centered around large transit providers or internet exchanges. And a hybrid of these could also be an option. All of these scenarios will yield different topologies, and therefore influence replica selection performance. Therefore we must also conclude that even though SCION shows good results in the simulations, that this does not necessarily translate to real-life scenarios.

Path Server Configuration

When looking at the influence the path server configuration has on the performance of SCION in Figure 4.3 it can be seen that this heavily depends on the topology that is used. For example, the ru topology shows no significant difference between the different configuration values, but the others do.

An explanation for this could be that the number of replicas in relation to the number of nodes is different for each of the topologies. The ru topology has almost 300 nodes with 10 of them being marked as replicas, the other topologies have 7 and 14 thousand nodes also with 10 marked as replicas.

Thus, the path server configuration is of limited influence on the performance and heavily depends on the density of the topology that is used.

Chapter 5

Conclusions & Future Work

In this thesis, we investigated how anycast as a concept can be deployed in an internet based on the SCION architecture and how that compares to anycast in the current internet based on BGP. We will first present the general conclusion of the research executed for this thesis and then also suggest directions for future work.

5.1 General Conclusions

As part of answering the first research question, how anycast as a concept can be brought to the SCION architecture, we wrote requirements for an anycast solution. Then we designed several solutions for making anycast deployments in SCION and verified those designs against the requirements. The requirements, designs and the comparison can all be found in Chapter 3. Among those designs there are several good candidates that can be used within SCION. The Multiple Advertisements and Naming designs are both suitable for use in SCION Thus, concluding the first research question; it is possible to deploy an internet service in a SCION based internet in a similar way as services are deployed through anycast in a BGP powered internet.

To answer the second research question, comparing the performance of the SCION designs to anycast in a BGP powered internet, we simulated some of the designs. These simulations however were limited in several ways. We purposefully left out simulating the effects of local policies set by network operators on the performance of both the SCION and BGP systems, as these policies are private and we could not make reasonable estimations in the timeframe of this thesis. We also left out dynamic network properties such as effects of load on the network and failing network links. To give a good answer to the research question, the focus of the simulations was to figure out the impact on relative performance (between BGP and SCION simulations) for several different properties. We also looked at topologies derived from the internet topology mapping project at CAIDA (the realistic topologies) as well as random generated topologies generated by multiple different algorithms.

When comparing performance, SCION is outperforming BGP on all real world derived (realistic) topologies and all random generated topologies. But since it cannot be assumed that an exact copy of the topology of the current internet will be present in a SCION internet, we also evaluated the same topologies after they have been artificially split up into ISDs. The performance of SCION on those split topologies is worse than on the original topologies, but still better than the performance of BGP.

The edge density of a topology seems to have a great influence on the relative performance difference between the SCION and BGP simulations. When the density of a topology is low the performance of both protocols is closer to each other than when the density of the topology is high. An explanation for this is that with lower density, there are fewer distinct paths between node pairs and thus less room for SCION to perform better than BGP. Or in other words: BGP has less chance to make a bad routing decision in low density topologies. Due to the fact that SCION is able to not only look at the hop count of end-to-end paths, which is the only discriminator in the case of the BGP simulations, but also look at latency, SCION is more often able to find the path with the lowest latency than BGP is.

We also looked at the effect of the configuration of the SCION path server on the performance of the anycast designs. The main configuration parameter for path servers is the number of path segments k that they return when responding to a path lookup request from a host in the network. When including this parameter in the SCION simulations we do see reduced performance compared to an unlimited number of returned segments but this still mostly performs better than the BGP simulations. Only in the case of extreme settings, topologies with very high density and a low value for k does BGP perform similar to SCION. But with a topology density similar to that of the realistic topology set and $k = 3$ as is suggested by the SCION authors, SCION still performs much better than BGP.

Taking all this into account, we conclude that SCION performs better than BGP in the replica selection scenarios that we simulated. However, due to the limitations of the simulations, we can not extend those conclusions to any real-life replicated service deployments. There are too many aspects that were not taken into account in the simulations to be able to do that. Even though we were not able to fully simulate all of the different parameters and properties of both protocols, we did find clear indications that SCION is able to perform better than BGP.

5.2 Future Work

In this section we will list and explain some possible directions for future work.

First, the effect of local network policies on the anycast performance. This was purposefully left out of the simulations executed in this thesis because it would not be possible to do this given the limited resources. However, it is reasonable to assume that this will have a significant effect on performance and therefore is worth looking into.

Another factor that was left out of the simulations in this thesis is the effect of traffic load and congestion. For this it is also reasonable to expect that this has a significant impact on the performance. On the one hand one could look into the (negative) performance impact of load and/or congestion in the network. On the other hand it could also be interesting to look at if some kind of load or congestion indication could be included in the SCION beacons to make a certain network link less attractive to use for clients.

Something that in future work could also be incorporated in the simulations is realistic latencies between ASes in the realistic topologies. The latencies used in the set of realistic topologies used in this thesis is based on the estimated geographical location of the ASes. If possible, the quality of the realistic topologies could be improved by using more accurate latency information.

It could also be interesting to include dynamic events in the simulations. For example to investigate how well each design is able to cope with failing network links or replicas. But also making other previously static simulation properties dynamic, such as network load changing over time or network operators changing their local policies.

Finally, it might also be interesting to investigate a real life anycast deployment in a SCION

network. Deploying global anycast services in for example a SCION testbed is a logical next step in bringing the anycast concept to the SCION architecture.

Bibliography

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," Tech. Rep. RFC4271, RFC Editor, Jan. 2006.

[2] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1," Tech. Rep. RFC8210, RFC Editor, Sept. 2017.

[3] M. Lepinski and K. Sriram, "BGPsec Protocol Specification," Tech. Rep. RFC8205, RFC Editor, Sept. 2017.

[4] A. Perrig, P. Szalachowski, R. M. Reischuk, and L. Chuat, *SCION: A Secure Internet Architecture*. Information Security and Cryptography, Cham: Springer International Publishing, 2017.

[5] "DNS Root Servers." https://root-servers.org/.

[6] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service," Tech. Rep. RFC1546, RFC Editor, Nov. 1993.

[7] Eijkel, Dennis, "Anycast and the SCION Internet Architecture," research Topics, University of Twente, Oct. 2020.

[8] "PeeringDB." https://www.peeringdb.com/.

[9] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen, "SCION: Scalability, Control, and Isolation on Next-Generation Networks," in *2011 IEEE Symposium on Security and Privacy*, (Oakland, CA, USA), pp. 212–227, IEEE, May 2011.

[10] C. Paasch and O. Bonaventure, "Multipath TCP," *Communications of the ACM*, vol. 57, pp. 51–57, Apr. 2014.

[11] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, (Los Angeles CA USA), pp. 183–196, ACM, Aug. 2017.

[12] A. Flavel, P. Mani, D. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev, "FastRoute: A scalable load-aware anycast routing architecture for modern CDNs," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, (Oakland, CA), pp. 381–394, USENIX Association, May 2015.

[13] F. Chen, R. K. Sitaraman, and M. Torres, "End-User Mapping: Next Generation Request Routing for Content Delivery," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM '15*, (London, United Kingdom), pp. 167–181, ACM Press, 2015.

[14] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "DONAR: Decentralized server selection for cloud services," in *Proceedings of the ACM SIGCOMM 2010 Conference on SIGCOMM - SIGCOMM '10*, (New Delhi, India), p. 231, ACM Press, 2010.

[15] C. Contavalli, W. van der Gaast, D. Lawrence, and W. Kumari, "Client Subnet in DNS Queries," Tech. Rep. RFC7871, RFC Editor, May 2016.

[16] E. Zegura, M. Ammar, Zongming Fei, and S. Bhattacharjee, "Application-layer anycasting: A server selection architecture and use in a replicated Web service," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 455–466, Aug./2000.

[17] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT)," *ACM SIGCOMM Computer Communication Review*, vol. 23, pp. 85–95, Oct. 1993.

[18] "Ns-3 — a discrete-event network simulator for Internet systems." https://www.nsnam.org/.

[19] "CAIDA AS Relationship Dataset." https://publicdata.caida.org/datasets/as-relationships-geo/.

[20] "NetworkX." https://networkx.org/.