Scaling SCIERA: A Journey Through the Deployment of a Next-Generation Network

François Wirz[*3] Marten Gartner[*9] Jelte van Bommel[*3] Elham Ehsani Moghadam[3]
Grace H. Cimaszewski[10] Anxiao He[15] Yizhe Zhang[15] Henry Birge-Lee[10] Felix Kottmann[3]
Cyrill Krähenbühl[10] Jonghoon Kwon[3] Kyveli Mavromati[1] Liang Wang[10] Daniel Bertolo[13]
Marco Canini[6] Buseung Cho[7] Ronaldo A. Ferreira[14] Simon Peter Green[11] David Hausheer[9]
Junbeom Hur[8] Xiaohua Jia[2] Heejo Lee[8] Prateek Mittal[10] Omo Oaiya[16] Chanjin Park[7]
Adrian Perrig[1,3] Jerry Sobieski[5,12] Yixin Sun[15] Cong Wang[2] Klaas Wierenga[4]

[1]Anapaya Systems [2]CityUHK [3]ETH Zurich [4]GÉANT [5]GMU [6]KAUST [7]KISTI [8]Korea University [9]OVGU Magdeburg
[10]Princeton [11]SingAREN [12]Sobieski Systems & Services [13]SWITCH [14]UFMS [15]UVA [16]WACREN

ABSTRACT

The SCION Next-Generation Network (NGN) architecture has expanded steadily since 2017, with today 20+ ISPs offering SCION connectivity. In production, IP-to-SCION-to-IP translation by SCION-IP-Gateways (SIGs) is used, such that applications are unaware of the NGN communication. To accelerate innovation and deployments, our aim is to increase the number of *native* SCION use cases, where the application is fully SCION-aware and optimizes communication across all path choices offered by the network. We set out to achieve two core objectives: (1) facilitating **simple native connectivity** for applications, and (2) enhancing the **scalability of SCION deployment** at academic sites.

With these goals in mind, we built the SCION Education, Research, and Academic (SCIERA) network infrastructure. This paper presents key lessons learned from the SCIERA deployment, which we anticipate will offer actionable insights to researchers, network operators, and system builders seeking to overcome practical challenges also for other NGN deployments. We report on establishing native SCION connectivity at research and education institutions that can reach 250,000 people across five continents, without relying on BGP. Our evaluation demonstrates that our core objectives were reached. Today, the SCIERA deployment offers tangible real-world benefits to users by providing rich global connectivity through a multitude of inter-domain paths.

CCS CONCEPTS

• **Networks → Network architectures**; Public Internet; *Routing protocols*; *Network performance evaluation.*

KEYWORDS

SCION, SCIERA network, network deployment, network architecture, path-aware networking, resilient networking, NGN

ACM Reference Format:
François Wirz, Marten Gartner, Jelte van Bommel, Elham Ehsani Moghadam, Grace H. Cimaszewski, Anxiao He, Yizhe Zhang, Henry Birge-Lee, Felix Kottmann, Cyrill Krähenbühl, Jonghoon Kwon, Kyveli Mavromati, Liang Wang, Daniel Bertolo, Marco Canini, Buseung Cho, Ronaldo A. Ferreira, Simon Peter Green, David Hausheer, Junbeom Hur, Xiaohua Jia, Heejo Lee, Prateek Mittal, Omo Oaiya, Chanjin Park, Adrian Perrig, Jerry Sobieski, Yixin Sun, Cong Wang, Klaas Wierenga. 2025. Scaling SCIERA: A Journey Through the Deployment of a Next-Generation Network. In *ACM SIGCOMM 2025 Conference (SIGCOMM '25), September 8–11, 2025, Coimbra, Portugal.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3718958.3750532

1 INTRODUCTION

Radical innovation at the network layer is notoriously difficult. Even modest changes such as IPv6 have taken over a decade to reach substantial adoption [51], despite support from major stakeholders. As a result, deploying a next-generation network (NGN) architecture that fundamentally changes the network layer is widely considered infeasible.

The main obstacle is the initial deployment: early adopters have little incentive to join a new network when few others are available to connect with. A critical mass of users is needed to make the adoption worthwhile. This dynamic is captured by Metcalfe's law, which states that the value of a network grows with the square of its user base [56]. Even when an early deployment is achieved, subsequent rollout faces resistance—especially from large organizations, wary of small markets and short-term revenue risks [11]. These challenges have prevented several promising network layer innovations from reaching broad adoption [16].

Despite these barriers, numerous research efforts have explored network layer innovations, as reviewed in Section 6. In this work, we focus on the SCION architecture, which has steadily grown since its first commercial deployment in 2017 [12]. In that year the SCION architecture was selected by the Swiss finance industry to replace the legacy Finance IP network [52, 58], meeting advanced security needs while offering an open and evolvable architecture [49]. This real-world deployment, the Secure Swiss Finance Network (SSFN), established a growing ecosystem of Network Service Providers (NSPs) [31] and spurred adoption across Europe and other regions, encouraging standards bodies and hardware vendors to support the technology.

[*] The joint first authors' order reflects their project joining time.

SIGCOMM '25, September 8–11, 2025, Coimbra, Portugal

ACM ISBN 979-8-4007-1524-2/2025/09
https://doi.org/10.1145/3718958.3750532

However, most current financial use cases rely on IP-to-SCION packet-level translation via SCION-IP-Gateways (SIGs) [12], which bypasses the application support requirement. To fuel innovation, native SCION applications and a large ecosystem of developers with native SCION connectivity are needed. To achieve this goal, we set out two core objectives: (1) facilitate simple native connectivity for applications, and (2) enhance the scalability of SCION deployment at academic sites. We specifically target academic sites as they can be more easily convinced to connect a large number of users to an NGN, providing students and researchers the opportunity to actively engage in innovation liberated from commercial considerations.

Through the construction of the SCION Education, Research, and Academic network (SCIERA), we connected research and education networks spanning five continents (Section 3). Today over 250,000 people can communicate with each other and the commercial SCION network using pure NGN inter-domain routing, i.e., without any dependence on BGP. This paper reports on deployment challenges and lessons learned in this process, with the aim of informing and inspiring further innovation at the network layer.

2 SCION BACKGROUND

In this section, we provide a brief overview of the SCION network architecture. More detailed information is available in the SCION book [12].

SCION is a path-aware inter-domain NGN architecture designed for high security, with efficient and scalable control and data planes. To achieve sovereign operation, scalability, and flexible network governance, SCION introduces *fault isolation domains* (ISDs), which define a logical grouping of ASes. Each ISD is governed by a set of *core ASes*, which define the ISD's governance policy and provide connectivity to core ASes in other ISDs. Trust within an ISD is anchored in its *Trust Root Configuration* (TRC), established by the core ASes. Each TRC is essentially a trust anchor that defines the root certificates, core ASes, and policies that govern the TRC's usage, evolution, and validity period, such as parameters for updating the TRC. The TRC is the basis of the SCION control plane *Public Key Infrastructure* (PKI), which is used, for instance, to provide each AS with an AS-specific certificate. This design addresses the monolithic global trust anchors in the current Internet by enforcing strong trust boundaries.

Routing is accomplished through path segment exploration and path segment dissemination mechanisms. The path segment exploration (referred to as "beaconing") is hierarchical: core ASes initiate *path-segment construction beacons* (PCBs) to create core path segments between core ASes, and core ASes initiate PCBs to non-core ASes within the same ISD to create path segments for intra-ISD connectivity. An AS receives PCBs from its neighboring ASes, and can decide which path segments to disclose to entities within the same AS and which path segments to publicly announce. A global path server infrastructure provides path segment registration and path segment lookup services, where path segments are associated with a given <ISD-AS> tuple. The path segments are cryptographically protected, ensuring that path segments are authentic and authorized. This design eliminates routing attacks such as prefix hijacking, one of the long-standing vulnerabilities of the current Internet architecture.

An important change in SCION is that end hosts, not intermediate routers, select the forwarding path from a set of options, ensuring predictable routing behavior. An end host wishing to reach a server in a different AS learns the ISD and AS number through a DNS TXT record, and then requests path segments from the local path server based on the <ISD, AS> tuple. The local path server contacts core path servers in the same ISD and, if needed, core path servers in the receiver's ISD, to fetch path segments. A collection of path segments typically allows for a variety of combinations, including shortcuts and utilization of peering links, to create a multitude of end-to-end paths. Considering optimizations of different criteria, a large path diversity leads to a high probability that one of the paths exhibits improved properties over the single path that is typically offered on the traditional Internet. Having such path diversity is another key departure from the current Internet and enhances the robustness against link failures. Note that SCION enables multipath communication, i.e., the simultaneous use of all these available path options.

The application embeds the end-to-end path segment in the SCION packet header and sends the packet to the local SCION border router using an IP-UDP "Layer 2.5" encapsulation to enable connectivity in (existing) local networks. In the data plane, the SCION border router discards the IP-UDP encapsulation (or other local encapsulations, if present, such as MPLS), finds the current hop field for that AS, which indicates ingress and egress interfaces (indicating the links that are used to enter and leave the AS), verifies the integrity of the hop field through an efficient symmetric cryptographic operation, moves the hop field pointer, and forwards the packet to the next SCION border router or end host.

End-host Stack. The end-host stack for a SCION network can be broadly divided into three core components: the daemon, bootstrapper, and application library. The daemon acts as the core of this stack, handling all end host interactions with the SCION control plane. It consolidates critical tasks, such as path lookup and selection, caching path information, providing information about the AS-local SCION services, and maintaining local databases for SCION's public-key infrastructure, such as storing TRCs. Meanwhile, the bootstrapper ensures that the daemon is correctly configured for the specific network environment by providing information on certificates and addresses of SCION infrastructure. Lastly, the application library serves as the bridge between applications and the SCIERA deployment, providing developers with socket APIs, as well as exposing the services of the daemon and the configuration provided by the bootstrapper to applications, forming a cohesive yet flexible stack.

3 SCIERA

In this section, we describe the deployment of SCIERA, spanning five continents and providing native SCION connectivity to 250K people. A core technical challenge was to deploy SCIERA in a "BGP-free" manner, i.e., without any dependency on BGP, such that BGP-level attacks and instabilities do not affect SCION and its security features [44] . As such, SCIERA provides native NGN

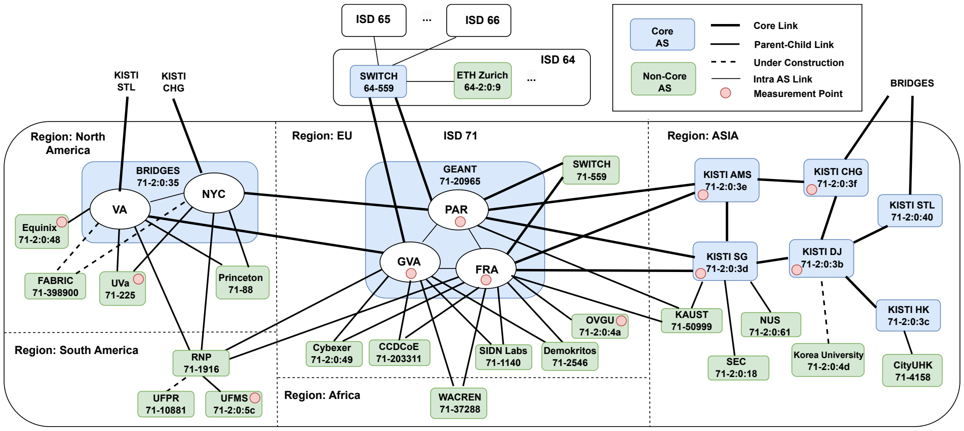


Figure 1: Topology overview of the SCIERA deployment.

properties, operating side-by-side with the traditional Internet, i.e., not as an overlay.

3.1 Deployment Strategy

Besides the "BGP-free" deployment, an important goal is to facilitate a scalable enrollment of research institutions to obtain SCIERA connectivity, where scalability refers to the effort for connecting additional institutions.

To avoid depending on BGP, we strive to connect an institution using Layer-2 links through dedicated VLANs. Since VLANs are relatively simple to set up across an IXP or a single network (but challenging across several networks!), we seek deployment at strategic Points-of-Presence (PoP) where many entities inter-connect. We follow a three-tier strategy: Tier-1 deployment at globally connected providers, which form the backbone of SCIERA; Tier-2 deployment at networks that provide national connectivity; and Tier-3 deployment at institutions. Thanks to the global footprint of two Tier-1 providers (GÉANT and KREONET), SCION connectivity is available at widely distributed PoPs (see Table 1).

GÉANT operates a vast research and education network in Europe; however, its global reach provides connectivity to almost all research and education networks across the globe. KREONET, managed by KISTI in South Korea, operates a global network that forms a geographic ring, with SCION routers strategically placed in Daejeon (South Korea), Hong Kong, Singapore, Amsterdam, Chicago, and Seattle.

National Research and Education Networks (NRENs) form the Tier-2, linking to Tier-1 providers for global connectivity, while delivering access to Tier-3 entities, such as research and education institutions, within their respective countries. NRENs can implement and operate SCION infrastructure themselves and offer carrier-grade SCION services to their customer networks, or offer customers dedicated Layer-2 interconnections (access points) to the SCION backbone.

In the latter case, the Tier-3 network must install (commodity off-the-shelf) servers running the SCION control service and border router(s), and set up VLANs with the upstream ASes. Networks use either the open-source SCION software or a performance-optimized closed-source version from Anapaya Systems[1] [2], which are interoperable.

The global footprint of these networks enables the deployment of SCIERA, as described in the next section.

3.2 The SCIERA Deployment

Figure 1 depicts the SCIERA topology[2]. All of its ASes are located in the SCION isolation domain (ISD) 71, except for two ASes in the Swiss ISD 64 connected via SWITCH. Core ASes, depicted in blue, are Tier-1 autonomous systems that provide both inter-ISD and intra-ISD paths, serving as critical nodes in the SCION network architecture. Each solid line represents a Layer 2 link that provides native SCION connectivity. Next, we give an overview by region.
North America. The North American segment of SCIERA is underpinned by a SCION core AS deployed on two nodes in BRIDGES [21]. This AS ties up the transatlantic links facilitated by GÉANT with the transpacific links through KREONET and the inter-American link to RNP, Brazil's NREN. BRIDGES, GÉANT, KREONET, and RNP are all connected to Internet2, which plays a critical role as

Table 1: SCIERA PoPs and collaborating networks.

Location	Peering NRENs	Partner Networks
Amsterdam, NL	GÉANT/KREONET	Netherlight
Ashburn, US	BRIDGES	Internet2/MARIA
Chicago, US	KREONET	Internet2/StarLight
Daejeon, KR	KREONET	KISTI
Frankfurt, DE	GÉANT	
Geneva, CH	GÉANT	CERN/SWITCH
Hong Kong, HK	KREONET	CSTNet/HARNET
Jacksonville, US	RNP	Internet2/AtlanticWave
Jeddah, SA	GÉANT/KREONET	KAUST
Lisbon, PT	GÉANT/RNP	RedCLARA
London, GB	GÉANT/WACREN	AfricaConnect
Madrid, ES	GÉANT/RNP	RedCLARA
McLean, US	BRIDGES	Internet2/WIX
Paris, FR	GÉANT	SWITCH
Seattle, US	KREONET	Internet2/PacificWave
Singapore, SG	GÉANT/KREONET	SingAREN

[1] Anapaya currently provides the software for the majority of the commercial SCION deployments and also provides managed services for ISPs.
[2] Refer to https://sciera.readthedocs.io/en/latest/ for the most recent topology.

an NREN by offering both national and international interconnection to universities and research facilities in North America using multipoint VLANs [25]. To this end, major institutions such as the University of Virginia and Princeton University, as well as the FABRIC [3] infrastructure, are connected to BRIDGES via Internet2.

Europe. The European region is administered via the GÉANT core AS, which consists of three nodes and connects various institutions in Europe, including SIDN Labs, NCSR Demokritos, OVGU Magdeburg, CybExer, and the CCDCoE NATO campus. The GÉANT core AS connects to the commercial SCION network [1] via SWITCH, enabling interconnection with the rest of the SCION production network, including ETH Zurich and more than 60 other entities. GÉANT also connects to the Asian region through two links to Amsterdam and Singapore.

Asia. Unlike other regions, Asia is structured with multiple Tier-1 core ASes. This deployment choice enhances routing control over KREONET's ring infrastructure, which spans the Northern Hemisphere, thereby offering more diverse and compelling multipath routing options. Each KREONET PoP functions as a Tier-1 core AS providing global connectivity, while also connecting leaf ASes, such as the National University of Singapore (NUS), Singapore-ETH Centre (SEC), King Abdullah University of Science and Technology (KAUST), Korea University, and CityU HK, to the network. To enhance connectivity between these PoPs, multipath connections are established, leveraging KREONET's own links and those provided by collaborating NRENs. For instance, KREONET, CAE-1 Consortium link, and KAUST I & II all offer connectivity between the Singapore and Amsterdam PoPs, leading to four distinct paths and emphasizing the potential of SCION's multipath routing. Complementing the PoPs at commercial IXPs, open exchanges of national research networks, such as the SingAREN Open Exchange in Singapore, facilitate last-mile connectivity of leaf ASes.

South America. In the South American region, RNP (the Brazilian NREN) is among the first NRENs facilitating native SCION connections to connect institutions. To this end, RNP established SCION-dedicated VLANs to both GÉANT and Internet2 to enable connections to the respective core ASes in Europe and North America. RNP's backbone [45] spans the entire country and connects all major research universities in Brazil. Each RNP PoP has at least two links to other PoPs, providing multiple disjoint paths to connect universities to RNP's SCION border router. UFMS is the first university connected to SCIERA, but at least three others (UFPR, UFMG, and UFES) will soon join.

Africa. WACREN is the first network in Africa to connect to SCION. WACREN is similar to a Tier-1 entity, as its network covers much of Western Africa. The aim is to connect universities within their network. WACREN's connection is facilitated via two VLANs between GÉANT and WACREN@London.

3.3 SCIERA ISD Evolution

Currently, SCIERA operates primarily within its dedicated ISD 71, with partial integration of two ASes from ISD 64 (the Swiss ISD). This arrangement arose out of historical reasons, specifically, the ASes in ISD 64 were early adopters of SCION prior to the formation of SCIERA. The current structure of ISD 71 evolved organically with core ASes added incrementally as needs emerged and opportunities arose. This organic structure is sufficient at present: managing a single ISD reduces operational complexity, and the existing level of trust among participants renders the governance manageable.

However, looking ahead, transitioning to more narrowly scoped ISDs, such as regionally scoped ISDs, offers clear benefits. For instance, establishing dedicated domains such as SCIERA-NA (North America) or SCIERA-EU (Europe) would enhance fault isolation by containing failures within specific geographic regions. Furthermore, regional ISDs would enable more efficient and autonomous governance, allowing each continent's academic and research networks to manage their own TRC policies and related processes. This structure would distribute the coordination and operational overhead, making tasks such as complying with regulations, onboarding new ASes, and designating core ASes more scalable and resilient as SCIERA participation continues to grow globally.

4 LESSONS LEARNED

Deploying SCIERA was not a single well-planned rollout, but rather a journey in which we encountered unexpected challenges and made decisions that in hindsight were not ideal. In this section, we distill these experiences by sharing the lessons we learned. The lessons are presented roughly in the order we encountered them, beginning with our earliest priorities (user onboarding and developer experience), and progressing toward scaling operations, managing deployment diversity, and sustaining the adoption. Some lessons reflect technical solutions, while others capture strategic or social realities.

4.1 Facilitating Fast Bootstrapping

One of the earliest and most important lessons we learned was the importance of first impressions. Today's users expect the Internet to "just work," and SCIERA should be no exception. If a new user, IT administrator or student, cannot connect to SCIERA within minutes, they may not try again. Hence we prioritize instant, out-of-the-box success for end users as a core deployment strategy, leveraging the impact of immediate results to drive adoption.

To support broad adoption, we established three key principles for connecting a SCION end host to SCIERA:

- **P1. Minimize User Interaction**: In a world of bring-your-own-device environments and automatic connections to familiar Wi-Fi networks, the process of joining, using, and moving between SCIERA ASes must be fully automated.
- **P2. Maximize Network Reachability**: SCIERA must be accessible to as many users as possible, even in complex, segmented network environments. Modern networks often include multiple segments, such as VLANs or VXLANs, where the SCION router may be located in a DMZ, while end users are on a Wi-Fi VLAN. The SCION end-host stack must be capable of traversing these segments.
- **P3. Maximize Portability**: The end-host stack should run on a wide range of operating systems and hardware architectures without requiring significant modifications. From smartphones to enterprise servers, SCIERA must support diverse devices to enable broad adoption.

4.1.1 Bootstrapping. A so-called "bootstrapping" process is essential for seamless onboarding (P1), as key SCION parameters, such as the AS topology with the local IP addresses of SCION border routers and control service(s), must be configured before connectivity can be established. However, implementing fully automatic bootstrapping introduces several challenges around security, privacy, and scalability. Malicious actors can pose as bootstrapping servers, hijacking traffic much like rogue DHCP servers in IPv4 [29]. Broadcast-based mechanisms, where new clients must broadcast their presence to all devices on the local network, can lead to unintentional information leaks, for instance, when clients expose identifiable data such as MAC addresses. Additionally, in large LANs multiple clients joining simultaneously or periodically polling can create a significant load on bootstrapping servers. SCIERA's bootstrapping piggybacks on existing bootstrapping methods already present in networks, avoiding the introduction of new security, privacy, or scalability issues.

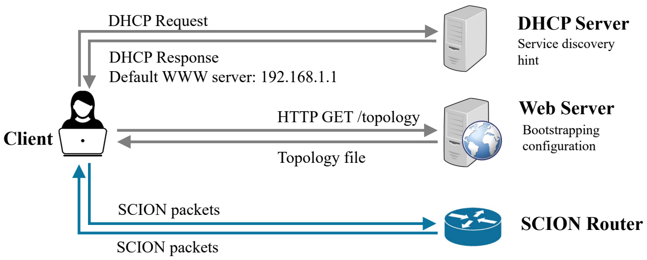


Figure 2: Overview of the automated end-host bootstrapping.

4.1.2 Bootstrapper Design. Figure 2 shows the three main components of the end-host bootstrapping system.

When a client connects to an AS within SCIERA that supports automated bootstrapping, it must first obtain a "bootstrapping hint." This hint allows the client to discover the bootstrapping server without manual configuration. We rely on fields in protocols that typically run by default (i.e., without any additional end-user opt-in) on all IP networks, such as DHCP, IPv6's Neighbor Discovery Protocol (NDP), or DNS (a full list is discussed in Appendix A). Because of limited space in these fields and the need to support a variety of mechanisms, the bootstrapping hint usually contains just the bootstrapping server's IP address. By using IP as a Layer 2.5 underlay, a single bootstrapping server can serve clients across different network segments.

The bootstrapping server is an HTTP server that provides the essential SCION configuration. Clients retrieve the local AS topology data, which includes, for instance, the network addresses of SCION border routers and control service(s), from the "/topology" endpoint. The server also distributes the TRCs, required to authenticate SCION entities and paths. The initial TRC must be obtained securely, either via TLS (potentially introducing a reliance on the Web PKI if not self-signed), or by validating the TRC out-of-band. Future TRCs are validated through TRC chaining. To ensure authenticity and integrity, the local AS signs the topology data with its AS certificate. After bootstrapping, the client has all the necessary information to fetch paths and make use of SCIERA.

4.1.3 Evolving the Bootstrapping Process. Our approach to SCION bootstrapping evolved over the years. We eventually realized that bootstrapping does not need to be a system-level prerequisite. Instead, applications can bootstrap SCION themselves. By embedding bootstrapping functionality directly into the SCION application libraries, there is no need for any pre-installed components. As a result, users can run SCION-enabled applications without even knowing what SCION is: it will "just work." We explore the concept of self-contained, application-driven bootstrapping further in the following section, where developer experience is key to adoption.

4.2 Supporting Application Development with Libraries

In our pursuit of broad adoption, we realized that supporting application developers meant more than just some documentation and a library. Their time is limited, their attention span is a precious resource, and they have little patience for clunky APIs. If we wanted SCION in real-world applications, we had to focus on the user experience for developers integrating SCION support.

4.2.1 Effortless In-App Bootstrapping. To minimize the entry bar for developers, we understood that they should neither have to manage, nor be aware of the bootstrapping process. Instead, this functionality is abstracted away and handled entirely within the SCION application library provided to developers. Concretely, the SCION application libraries can function in three distinct modes:

- **Daemon-dependent:** The application library interacts with a SCION daemon running in a separate process to access SCION functionality, such as path lookups. The daemon is aware of SCION infrastructure due to the bootstrapper running as a separate, pre-installed, (background) system component.
- **Bootstrapper-dependent:** For platforms where running a separate daemon is not feasible[3] (P3), the application library can operate independently, incorporating essential SCION functions directly within the application process. This mode sacrifices some of the benefits of a centralized daemon (such as shared caching and consolidated control plane interactions). The separate bootstrapper component is still used for bootstrapping information.
- **Standalone:** This mode further removes the dependency on a separate bootstrapper, thus removing the requirement for any installed components. The application library fetches the bootstrapping hints and directly communicates with SCIERA's network infrastructure. This mode is less efficient in environments with frequent network migrations, as each application must detect the change and re-bootstrap individually, unlike a shared bootstrapper.

There is no need to explicitly choose a mode of operation. Once it is established that there is no daemon or bootstrapping information present, the application library can fall back to the integrated bootstrapper in standalone mode (P1). This design ensures that developers do not need to implement or concern themselves with the bootstrapping procedure or any inter-process communication between the application and the SCION end-host stack.

4.2.2 Drop-In Socket Replacement. SCION uses an IP-UDP "Layer 2.5" underlay to ensure broad compatibility across networks and

[3] For instance, mobile platforms and low-power IoT devices often have constraints that make running long-term background services, such as a daemon, undesirable. It may also be impossible to share the service due to the strict isolation between apps.

platforms. This implementation detail is fully abstracted from application developers, who instead work with a socket API provided by the SCION application library. This socket transparently handles all Layer 2.5 encapsulation and serves as a drop-in replacement for standard IP-UDP sockets, enabling quick and seamless integration of SCION functionality into existing applications (P1). The socket API mirrors traditional socket programming interfaces while enabling path-aware features such as optimizing routing for latency, bandwidth, or geographical constraints. To ensure compatibility with a variety of codebases, SCION application libraries are available in common programming languages, such as C, C++, Go [40], Java [61], Python, and Rust [38].

An alternative approach to providing a SCION application library is to add SCION support to the "Happy Eyeballs" [47, 57] library, which is, for instance, present on iOS and Android. This library handles the decision on whether to connect to a server through IPv4 or IPv6. Adding SCION as a third option to this library would immediately enable all applications using it to communicate through SCION, if available and supported by the destination.

4.3 Scaling with Goodwill and Recycled Parts

Another humbling realization hit hard and early: most of the people who were enthusiastic about deploying SCION could not free much budget to do so, even less for new, experimental infrastructure. NSPs, campus networks, and research institutions were excited about the architecture, but could not justify purchasing extra infrastructure, nor overhauling their network to support it.

4.3.1 Recycling Infrastructure. To lower the barrier to entry, the SCIERA deployment balances the use of existing infrastructure with the implementation of new protocols by leveraging IP routing for local (intra-AS) communication. This approach allows clients across various network segments to access the SCIERA infrastructure (P2), without requiring a complete overhaul of network systems, limiting deployment barriers and operational disruptions. Essentially, IP is repurposed as a bridging layer, functioning as a "Layer 2.5" in the OSI model, to transport SCION packets across IP-routed network segments within an AS, with SCION serving as the Layer 3 protocol.

To retain the security and efficiency benefits of SCION, it is important to operate the SCION infrastructure as a native service and not as an (inter-AS) overlay relying on BGP. Routing SCION traffic via overlay links can reintroduce the security issues that plague today's Internet (e.g., BGP attacks) as discussed by Ribeiro et al. [44]. By limiting IP to intra-AS communication (e.g., between end hosts and bootstrapping servers) and handling inter-AS communication fully by SCION, the approach preserves security while maintaining advantages such as multi-path communication.

To further enhance SCIERA's resilience, NSPs are encouraged to reserve dedicated bandwidth for SCION traffic on links also carrying IP traffic. This isolation helps ensure that SCION can be rigorously evaluated without undue variables related to IP-level attacks or heavy usage. Additionally, if the SCION connection between border routers relies on BGP, then BGP-level attacks and instabilities would also sever the SCION inter-AS link. In order to avoid this, SCION connections between border routers are preferably established using Layer 2 technologies (e.g., MPLS, VLAN) that do not rely on BGP and which are readily available in most network infrastructures.

4.3.2 Low-cost deployment. We stopped imagining our deployment as if all participants were well-funded telecommunication providers and prioritized lean deployments in terms of operational expenditures (OpEx) and capital expenditures (CapEx). OpEx in the short-term is elevated because network administrators need to learn SCION's configuration and management, but its security benefits promise lower OpEx on an intermediate- and long-term time scale because of increased stability, and the resilience to configuration errors and attacks as experienced by entities in the SSFN.

To reduce CapEx, SCION relies on software-based switching on commodity servers or virtualized execution environments on existing routers (offered for instance by Extreme Networks or Juniper), and re-uses the AS-internal switching fabric for communication [31]. In practice, deploying a SCION AS requires only a single server running a control service and a border router connected to upstream SCIERA providers. This "lean start and expand as you grow" model has proven effective, with over 20 commercial NSPs now offering SCION services (Appendix D). Many SCIERA participants run reliably on hardware costing well under $7,000.[4] This low-cost infrastructure-reuse model has enabled expansion without waiting for new budgets.

4.4 Ensuring Maintenance, Alerting, and Observability

Early deployments were done the hard way: manually edited configurations, scattered documentation, and lots of hand-holding. As adoption grew, some operators preferred to manage their own AS, without granting us access. At the same time, it became clear that most sites deploying SCION had minimal staff, minimal time, and zero appetite for manual fiddling. The lesson was obvious: if it was not automated, it would not scale. We began formalizing reusable procedures and examples based on earlier deployments. As we encountered a range of technical and logistical challenges, we built a repertoire of scripts, guides, and best practices to streamline new deployments.

This effort culminated in the development of the SCION Orchestrator [41], a toolchain that cut SCION AS setup and management from days to a few hours. It features a graphical user interface that automates core management tasks—such as adding certificates or adding new links—enabling sites and operators without SCION experience to deploy and operate their AS *on their own.* At least until something broke. Diagnosing issues still demanded in-depth knowledge of SCION: knowing where logs were stored, which errors to look for, and how to interpret them. To further simplify operations, the orchestrator includes an aggregated service status dashboard with easy access to relevant logs, making it easier for new operators to troubleshoot their deployments or request help.

SCION itself does not provide a built-in alerting mechanism. To proactively (and automatically) detect issues in SCIERA, monitoring is essential. While we have full visibility into the SCIERA ASes managed by us, external ASes pose a challenge due to limited access. To address this challenge, we implemented continuous connectivity

[4] A qualified example setup: HPE Proliant DL380 Gen10 (Intel Xeon 8 cores, 32 GB Memory, Intel E810 100Gbps NIC)

monitoring from our infrastructure to all connected ASes. This approach reduces the need for independent operators to set up and maintain their own monitoring systems, or connect directly to ours. When an issue arises, our system alerts the affected parties via email. Operators can then check the orchestrator's status page and logs to identify and resolve the issue themselves, or with our assistance.

4.5 Embracing Heterogeneity

When deploying and scaling SCIERA, there was a strong temptation to optimize for a single controlled environment: specifying one operating system, one hardware vendor, and one firewall, all predictable. Real-world deployments quickly humbled us. This meant facing a wide range of heterogeneous conditions: different firewall configurations, firewall implementations, VLAN setups, NAT quirks, and incompatible middleboxes. There exists no one-size-fits-all deployment; optimizing for a single environment was futile.

Embracing deployment diversity strengthens deployment robustness by reducing the blast radius of any single failure or vulnerability [42]. SCIERA's use of both commercial and open-source SCION implementations across various operational models, ranging from Anapaya-managed ASes to self-operated ASes, boosted the diversity further. This diversity not only accounted for the varying adoption readiness of the different stakeholders, but also exposed issues and corner cases that might have been missed in a more uniform environment.

For example, SCION introduces a custom Control Plane PKI with AS-level certificates that follow strict formats and include mandatory fields. These certificates are needed for essential network operations, such as beaconing. Due to their (intentionally) short validity period—typically just a few days—SCION requires fully automated certificate issuance and renewal. This design requires that specific ASes within an ISD operate as Certificate Authorities (CAs).

Prior to the deployment of SCIERA, the SCION production network primarily relied on Anapaya's certificate management system, which issued certificates through a proprietary, closed-source CA. However, SCIERA included participants that use the open-source SCION stack, which lacked such a CA. This lack of tooling created a significant interoperability challenge: There was no open solution for certificate management that supported both Anapaya's CORE implementation and the open-source SCION control plane. We addressed this by developing a new open-source SCION CA using the smallstep framework [50]. This CA facilitates automated certificate renewals in SCIERA and ensures compatibility with both implementations. The SCION Orchestrator, described in Section 4.4, takes care of initiating the renewal process for certificates using the SCIERA ISD CAs.

Another challenge originated from the lack of a native SCION end-host solution in Anapaya deployments at the time of setup. Here, the open-source implementation played a crucial role by enabling support for native SCION applications within Anapaya-managed ASes.

4.6 Finding The Champions

One of the most valuable lessons we learned, which saved us time, energy, and many late-night email threads, was to focus on the enthusiasts. Every deployment has its champions and its skeptics, we stopped trying to convince the latter.

We sought out the people already excited by the new architecture. It is a challenge to convince commercial NSPs to upgrade their networks with SCION functionality and set up the host bootstrapping service, because the majority of customers would not pay for a service for which only a few specialized applications are available (first requiring the developer adoption described in Section 4.2). Thus, a more promising approach is to start with research and education networks that have an innovation perspective and are less exposed to competitive pressures. Therefore, we began working closely with NRENs and Regional Research and Education Networks (RRENs), which were not only more open to innovation, but also offered SCIERA a broader reach across several institutions. While initially demanding to set up, multipoint-VLANs at NRENs or RRENs significantly streamlined subsequent deployments at institutions. We further improved the deployment process by deriving a set of different (flexible) deployment options (Appendix B).

Once the operational readiness in SCIERA had been established, the deployment started to grow organically, and it became easier to find new champions. Nevertheless, it was still important that we made a case for SCION and SCIERA to these early adopters. In Section 4.7, we describe our experiences when convincing these early adopters.

4.7 Avoiding Feature Creep

Early on, we realized that scaling SCIERA was not just a technical challenge, it had to be appealing too. We then fell into a trap: we tried to make SCION everything to everyone. We had built a protocol that could offer secure routing, path control, geofencing, high-throughput file transfer, privacy, scalability, multipath load balancing, green routing, and DDoS resilience. Years later, we learned we had to stop trying to be the Swiss Army Knife of networking and focus instead on a few core competencies. That meant sharpening messaging and focusing on key benefits, rather than diluting the message by catering to every technically possible niche use case. Our early temptation to make SCION everything to everyone only resulted in confusion.

We shifted from pitching all of SCION's features to showcasing how it solves real, visible problems that people already had. For organizations, business continuity became an important aspect, which depends on *high availability* and *security*. SCION delivers both: its native multipath architecture enables rapid failover, while its secure control and data planes protect against routing attacks and allow avoiding (potentially) untrusted ASes.

For general users, who may lack awareness or interest in the networking stack they use, we focused on *low-latency gaming* and *sustainability*. SCION enables selecting the path with the lowest latency and switching paths instantly if performance worsens. This can improve connectivity for competitive gaming by providing an alternative to IPv4/IPv6 (as fallback options). Low-latency connectivity can be crucial for winning online games [5, 33]. In terms of sustainability, SCION allows users to choose "green" paths based on energy or carbon metrics [54], incentivizing ISPs to reduce emissions [46].

4.7.1 SCIERA SCIENCE-DMZ. For SCIERA, the Science-DMZ use case has been useful for driving interest. Research partnerships often require the sharing of large and sensitive data sets, such as clinical trials or simulation results based on confidential input, which demands both high performance and strong security. However, institutions frequently face a trade-off between the two: traditional security appliances often become performance bottlenecks, leading to packet drops and retransmissions due to limited buffer capacity. As a solution, specialized network architectures, so-called Science-DMZs, have been proposed [15]. Science-DMZs isolate designated traffic from regular (campus) Internet traffic, enabling secure and efficient transfers, while still providing carefully managed access to internal resources.

In SCIERA, the combination of SCION, LightningFilter [12], and Hercules [18] forms a SCION-based Science-DMZ [53] that enables secure, high-speed data transfers with path control and geofencing, using cost-effective hardware. LightningFilter, an open-source SCION firewall, delivers line-rate throughput with 100 Gbps, at a significantly lower cost than commercial firewalls, while Hercules enables high-speed file transfers, making use of SCION's multipath capability. In a SCIERA Science-DMZ, the border router isolates SCION traffic, while a dual-homed transfer node bridges SCION and local networks for managed supercomputing access. To support this use case, the KREONET network SCIONabled a 20 Gbps ring around the Northern hemisphere, with routers in Daejeon (South Korea), Seattle, Chicago, Amsterdam, Singapore, and Hong Kong. Additional SCION routers were set up at GÉANT, peering with KREONET and SWITCH.

4.8 Leveraging User-Space Networking

Our original vision for SCION's end-host stack was deep OS integration: a new address family, AF-SCION, built directly into the kernel. This integration would have enabled native socket support and efficient packet handling. But we quickly recognized the steep cost: writing and maintaining kernel code across a variety of systems, relying on OS vendor approval, and facing long update cycles even for minor changes. Kernel integration and maintenance required resources we did not have, and would severely slow the iteration speed—so we pivoted.

We decided for the initial stages that SCION's end-host stack would live entirely in user space. This approach made the stack immediately portable to a wide range of systems, including those with closed-source kernels, and enabled rapid iteration. Still, we did not give up the dream entirely: hoping to one day enter the kernel world, we anticipated a transitional tool: the *dispatcher*. This background process listened on a single fixed UDP port (with IP-UDP as the "Layer 2.5") and demultiplexed incoming network-facing SCION traffic via Unix domain sockets to the correct application. It was a faithful recreation of what a kernel socket might do, just in user space.

For a time, this approach worked well. However, limitations surfaced in high-performance scenarios, such as the Science-DMZ use case with Hercules file transfers (Section 4.7.1). The dispatcher introduced overhead and a bottleneck, since its processing capacity was shared across all SCION applications. Instead of revisiting our end-host stack, we reached for a band-aid: Hercules would use XDP (eXpress Data Path) [24] to bypass the dispatcher and kernel to handle SCION packets itself. We ran into similar issues with LightningFilter, which depends on high-speed packet processing. Because all traffic passed through a single UDP port, we could not leverage Receive Side Scaling (RSS) [23] to distribute traffic across multiple cores. Performance hit a wall—again.

The dispatcher component became even more of an obstacle when we started supporting mobile platforms. Mobile OSes make it difficult to run background daemons that are shared between multiple applications. Running a dispatcher was impractical. We considered building a tunnel adapter, like a VPN, to mimic dispatcher behavior—but at that point, we asked the hard question: why are we still doing this?

By then, networking had changed. QUIC and mTCP had revolutionized user space networking [26, 28], and userland networking stacks were no longer unorthodox ideas. Therefore, we embraced a 'fully in user space', dispatcherless future, where each application would open its own UDP socket, over which it would directly send SCION packets. Unfortunately, by this point, the dispatcher was already deeply embedded in SCIERA's deployment. Still, we began the (long) migration toward becoming dispatcherless, which involved requiring clunky backward compatibility mechanisms and extensive coordination. In real-world deployments nowadays sequential steps in user space facilitate progress.

4.9 Ethics

It turns out that getting a new Internet Architecture deployed involves more than just a strong protocol and automation, but also requires careful attention to local policies and regulations.

As SCIERA's connectivity expanded, we found ourselves facing a surprising network dilemma: what if our research network became so attractive that commercial traffic started flowing through it? While it is flattering to build something useful enough to be misused, allowing SCIERA to act as transit for commercial providers would have violated the policies of typical academic networks—potentially landing someone in a conference room justifying operations to lawyers. To prevent this, we instituted a strict SCION path policy to ensure that traffic from/to any commercial providers can only terminate/originate within (but not transit) SCIERA.

We found that careful attention to legal compliance, including aspects such as local data privacy laws, helped ensure not only smoother deployments, but also peace of mind for stakeholders. SCION's architecture helped here, offering transparency and control in ways that could be easily documented and justified.

In addition to legal obligations, two important ethical considerations surfaced during SCIERA's deployment. First, stakeholders were concerned about the loss of network source-destination traffic insight on middleboxes at the IP layer. This concern is alleviated in practice by Anapaya's commercial SCION border router implementation, providing NetFlow exports with the required flow data.

A second concern relates to legacy firewalls not being able to inspect SCION traffic beyond the outer IP-UDP encapsulation. To assuage this concern and provide benefits beyond what is possible with legacy IP traffic, we developed the LightningFilter software in DPDK, enabling SCION traffic filtering and authentication at line rate.

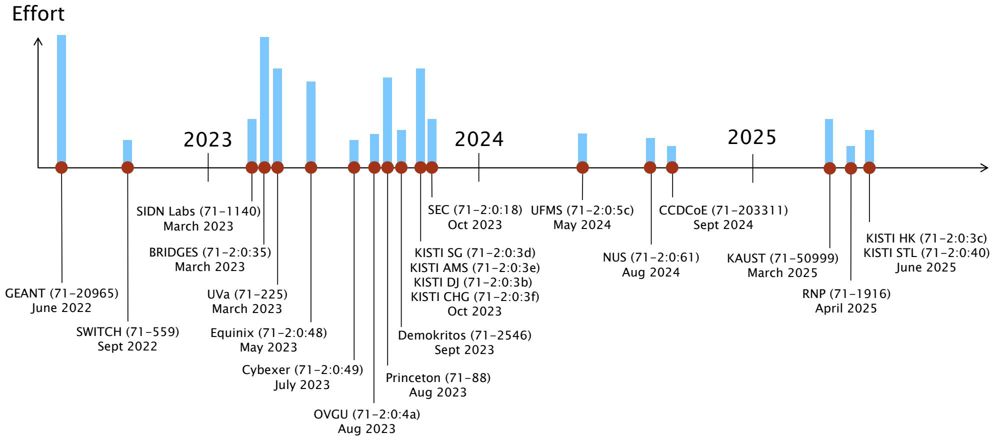


Figure 3: SCIERA deployment and estimated effort over time.

5 EVALUATION

The preceding lessons shaped SCIERA's design and deployment. While each was motivated by a specific challenge or constraint, their cumulative effect was a network that performs robustly, scales globally, and delivers measurable benefits. Not all lessons lend themselves to clean metrics. For example, the value of "embracing deployment diversity" (Section 4.5) and "finding the champions" (Section 4.6) cannot be reflected in a graph, but is reflected in the fact that today over 250,000 people have access to the network, and a growing ecosystem of NSPs exists (Appendix D). Similarly, the lesson of "ensuring compliance with legal and ethical expectations" (Section 4.9) cannot be plotted—but so far, our caution has fended off trouble. In this section, we assess how well the lessons translated into operational reality.

5.1 End-host Bootstrapping Performance

The performance of end-host bootstrapping, outlined in Section 4.1, is evaluated by executing the bootstrapper on all major desktop OSes for the various hinting mechanisms. The evaluation focuses on two critical steps: hint retrieval from the network and fetching the SCION configuration from the discovered SCION bootstrapping server. Figure 4 shows the timing distribution for 30 runs per hinting mechanism. Given the efficiency of the bootstrapping process, the time to connectivity in SCIERA can be considered minimal from a user perspective, as such brief durations (i.e., median ≤ 150 ms) are imperceptible in terms of user experience. This serves as clear evidence of success for "facilitating fast bootstrapping" (Section 4.1): we delivered an onboarding experience that works across platforms and is fast enough that users cannot notice it is happening at all.

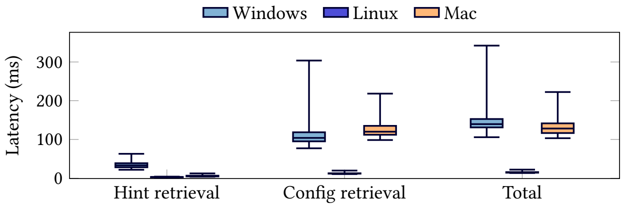


Figure 4: Network hint retrieval, configuration retrieval, and overall latency for bootstrapping on each of the supported platforms.

5.2 Application Enablement Effort

We conduct a case study on application enablement for the SCIERA infrastructure through the lens of 3 specific examples.

Bat tool. The pre-existing open source `bat` [30] command line tool, which is a pure Go re-implementation of a cURL-like web client application. Using the existing open source SCION Go libraries [48], we successfully integrated SCION support into the `bat` tool with *fewer than 20 lines of code.* The integration primarily involved adding SCION-specific command-line interface (CLI) flags and updating the default transport to use a SCION connection for both sending and receiving data. Specifically, we introduced CLI flags to `bat` that allow for interactive SCION path selection, the specification of a SCION path policy, and the selection of a path optimization strategy. We provide a SCION-aware transport by instantiating the default transport in `bat` with the SCION-enabled `http.Transport` from the `shttp` package, which in turn uses the `PAN` library to support the aforementioned options exposed via the CLI flags. A summary of the differences is listed in Appendix E.

Caddy plugin. By using the modular plugin architecture of the popular web server Caddy, we implemented a SCION reverse proxy. An overview of the SCION specific additions to enable SCION support is reported in Appendix F.

Java netcat. The JPAN [61] Java library fully implements the `DatagramSocket` class, which means that it can act as a drop-in replacement to provide SCION support in instances where no stream socket behavior is required. We provide a listing in Appendix G that shows the changes necessary to adapt a third-party netcat Java implementation with SCION support.

The minimal code changes required to enable SCION in all of these applications highlight a success in creating a developer experience that is frictionless and low-effort, as outlined in Section 4.2.

5.3 Deployment Effort

In Figure 3, we show a timeline of the SCIERA deployment over time. On the y-axis, we provide a relative estimate of the work hours that were required to deploy each AS. These estimates are based on a subjective assessment of efforts, cross-checked with the volume of email exchanges and the approximate time between the first interaction and successful SCIERA integration. The figure shows that initial SCION network setups demanded significant effort, particularly for hardware procurement, installation, and coordination. However, subsequent deployments of the same type and connection were simplified. This increased deployment efficiency was the result of both the team's accumulated experience, our automation efforts (Section 4.4), and the growing familiarity of the NSPs with the SCION deployment procedures. We provide a more detailed analysis of the deployment timeline in Appendix C.

5.4 Connectivity Analysis

To collect key metrics in SCIERA, we deploy a measurement tool called *scion-go-multiping* [20] across 11 ASes (5 in Europe, 2 in Asia, 3 in North America and 1 in South America). This tool performs ping measurements to other SCIERA participants ASes every second, both over SCION and the IP Internet, and aggregates path statistics every 60 seconds. The nodes running the tool are marked in Figure 1.

Path statistics are collected by performing a full path probe every minute, where we record all paths known via a `scion showpaths` query. A full path probe is also performed if in the previous measurement interval at least two pings failed.

Ping measurements are conducted by sending ICMP echo pings over the IP Internet, which follows the path defined by BGP, as well as SCION Control Message Protocol (SCMP) pings in parallel over three SCION paths: the shortest, the fastest, and the most disjoint. The *shortest* path is defined as the path with the smallest number of AS hops and lowest path identifier. The *fastest* path is the path for which the lowest round-trip time (RTT) was measured in the last full path probe. The *most disjoint* path is the path with the lowest number of interface IDs shared with the *shortest* and the *fastest* path among all paths measured during the last full path probe. We combine the AS-unique interface identifiers with SCION's ISD-AS numbers to generate globally unique interface IDs to calculate disjointness. For each SCION ping, we store the minimum RTT observed among the three paths during each time interval, the respective path, and the success ratio of the three pings in a database. Note that we also send ping messages to ASes where the tool is not deployed.

Over a 20-day period, we collected ~265M ping measurements and 3M path statistics. We provide our collected dataset as well as setup instructions in a public repository [19, 20].

We analyze the minimum RTTs observed across the three SCION paths over the SCIERA topology, comparing them with RTTs over IP for the same sites. In our collected dataset, we observed gaps in the ICMP ping measurements from certain source nodes caused by the stalling of our measurement tool. These stalls typically occurred after the first 15-30 minutes of each hour, after which no ICMP messages were sent or received until the tool was automatically restarted at the beginning of the next hour. To ensure a fair comparison, we exclude both SCION and IP measurements for the affected time intervals where a majority of ICMP pings were missing, but we keep the data for those intervals where only a few ICMP pings failed. As a result, we considered 89M pings over SCION and 82M pings over IP in our analysis. Figure 5 shows the distribution of RTTs to other SCIERA participants, as measured from the 11 ASes where the measurement tool was deployed. Our results show a similar trend for SCION and IP in the first 50% of the pings and a small latency reduction of 6.9% for the median of the pings, reducing it from 160.9 ms over IP to 149.8 ms over SCION. The latency improvement is more pronounced for the slowest pings: the 90th percentile RTT is reduced 23.7% from 376 ms to 287 ms.

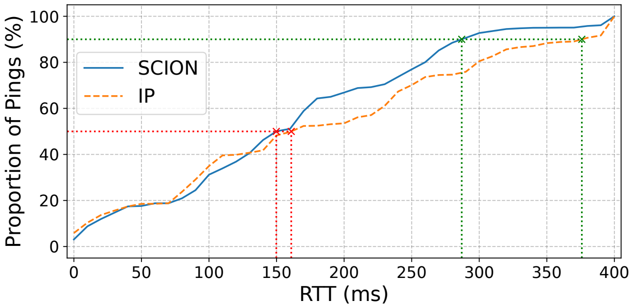


Figure 5: CDF of ping latency for SCION and IP.

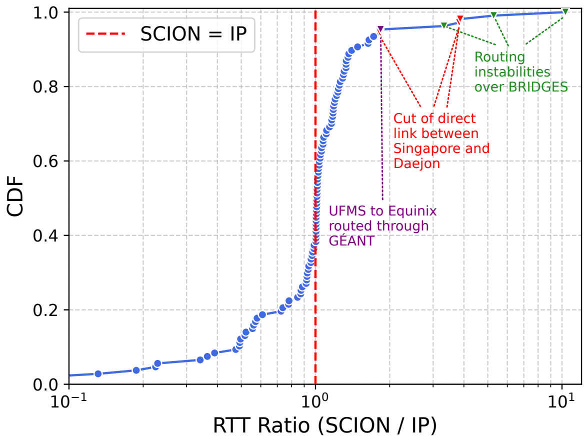


Figure 6: CDF of the Round-trip Time Ratio of SCION compared to IP.

To visualize the RTT relation, as shown in Figure 6, we first calculate the average SCION and IP RTTs for each AS pair over the entire measurement period. We then compute the ratio by dividing the SCION RTT by the IP RTT. We observe that 80% of the pairs see less than 25% RTT inflation compared to IP, and around 38% even have a lower latency over SCION. Apart from this, we observe a set of outliers, where SCION provides significantly higher relative RTT compared to the rest of the measurements. First, we identified that a direct link between two core ASes in KREONET was unavailable for a while, causing SCION to route traffic via a backup path around the globe. While this demonstrates SCION's ability to react to link failures, it significantly increased the RTT pings for this AS pair. Second, we measured elevated RTTs between UVa, Princeton and Equinix. These ASes typically connect via BRIDGES, which introduces longer physical paths compared to more direct IP paths. Moreover, BRIDGES experienced instabilities during the

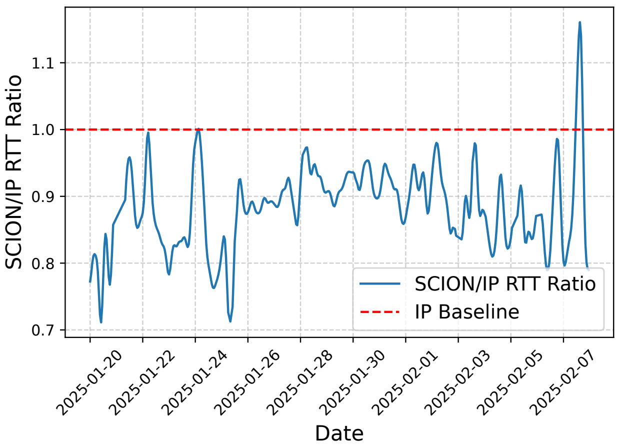


Figure 7: Round-trip Time Ratio of SCION compared to IP over Time.

measurement period, further contributing to increased latencies via alternative SCION paths. Since the corresponding IP paths exhibit relatively low RTTs, these deviations appear more pronounced as outliers. The third set of outliers involves pings between UFMS and Equinix. During the measurement period, this route required an additional hop through GÉANT before reaching the BRIDGES network and ultimately Equinix, further increasing latency relative to the IP baseline.

In Figure 7, we present the RTT ratio of SCION and IP for all AS pairs over time, observing episodes where SCION provides around 15-20% lower RTTs. Interestingly, we measure two spikes. On January 21st, the SCION RTT shows a significant increase due to maintenance work affecting several links, resulting in longer paths being selected. In the following days, we performed various maintenance tasks and changes to the network, relating to the strong fluctuation of RTT ratio compared to IP. After January 25th, several new links between EU and US became available, which contributes to the path diversity and stabilizes the RTT ratio, despite slightly increasing the overall average RTT. After February 6, several nodes were upgraded, and some links received maintenance, resulting in fluctuating SCION RTTs and showing a significant spike of RTT ratio again.

The ability to conduct these measurements across SCIERA, demonstrates that the deployment achieved broad connectivity using lean infrastructure as highlighted in Section 4.3. Furthermore, observed RTTs for SCION traffic indicate that even with a minimalist deployment, partially running on spare parts, remarkably usable interconnectivity was achieved.

5.5 Multipath Quality

To demonstrate SCIERA's multipath communication capability, we first evaluate the *path diversity*. Figure 8 shows the highest number of active paths observed at any specific time during the measurement period between 9 ASes, where `active` means a path that is both known to the control plane and is usable on the data plane.

For each source-destination AS pair, there are at least 2 distinct paths available, with some ASes having over 100 different paths for end hosts to choose from. For example, while there is only a single BGP path from the Daejeon core (`71-2:0:3b`) to the Singapore core (`71-2:0:3d`), SCIERA also provides paths around the globe via Chicago and Amsterdam. This multipath capability ensures high resilience in real-world network operations: e.g., during a submarine cable cut between Korea and Singapore in August 2024, communication seamlessly continued without any disruption.

In extreme cases, such as between UVa (`71-225`) and UFMS (`71-2:0:5c`), over 100 path options exist, although the last mile segments at both ends consist of only two physical links. This case demonstrates that many (feasible) routing options are available across the path segments through Europe and via North America. Unlike conventional network configurations that rely on a single primary link and maintain redundant links solely as backups for failures, SCION's multipath capability enables simultaneous use of all available link options.

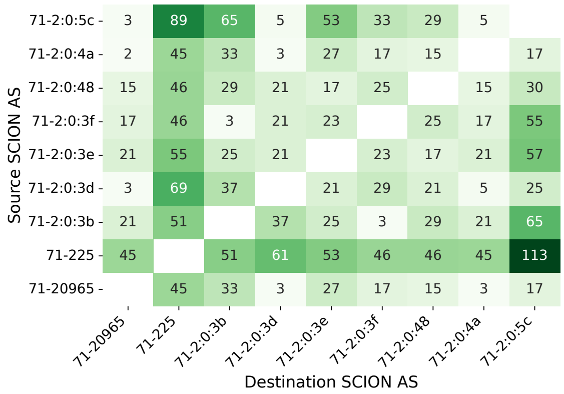


Figure 8: Maximum number of active paths between AS pairs.

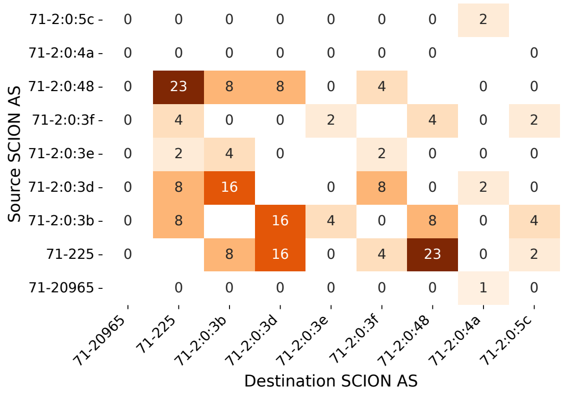
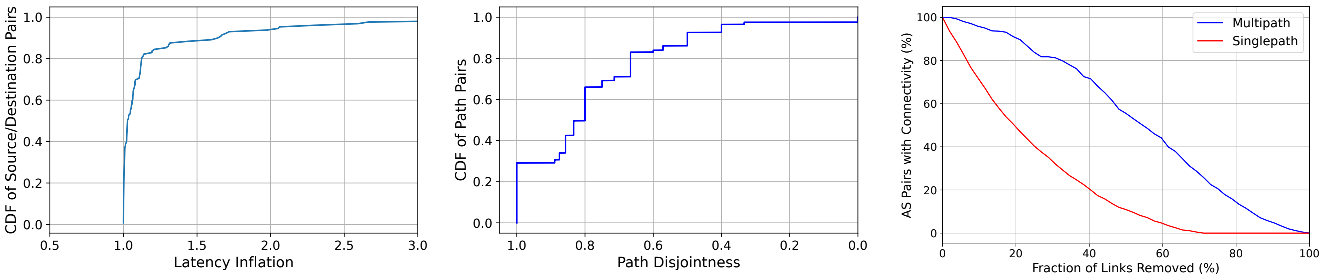


Figure 9: Median deviation from the highest number of active paths in SCIERA.

Figure 9 shows the median deviation from the highest number of active paths measured, providing insight into how consistently the path count between an AS pair reaches this upper bound. For most AS pairs, the median deviation is 0, indicating that the maximum number of active paths shown in Figure 8 was usable most of the time, offering a variety of path options for end hosts in SCIERA.



(a) CDF of path latency inflation (d_a/d_s). **(b) CDF of path disjointness for all AS pairs.** **(c) Impact of link failures on AS connectivity.**

Figure 10: Benchmark results characterizing the path diversity and resilience benefits

Nevertheless, we observe two AS pairs that show a high median deviation from the maximum. The cable cut between Korea and Singapore affected the connectivity between Daejeon core (`71-2:0:3b`) and Singapore core (`71-2:0:3d`), resulting in a median difference of 16 compared to the maximum of 37 and 61 paths shown in Figure 8. We observe another significant deviation for the AS pair UVa (`71-225`) and Equinix (`71-2:0:48`). Here, issues with the BRIDGES core AS (`71-2:0:35`) serving as an intermediate hop between those ASes lead to a notable median difference compared to the maximum number of paths.

Path Diversity Benefits. Figure 10 highlights the path diversity and resilience benefits based on several metrics. Specifically, we evaluate *path latency inflation*, i.e. the fraction $d_\mathrm{a}/d_\mathrm{s}$, where d_s is the delay of the lowest RTT path and d_a is the delay of the second lowest RTT path between two ASes, calculated over all SCION ping entries performed in the measurement period. Figure 10a illustrates the CDF of the latency inflation for all AS pairs. We note that 40% of AS pairs show a path latency inflation close to 1.0, proving the existence of at least one similar-RTT path. Notably, 80% of AS pairs have less than 20% ($d_\mathrm{a}/d_\mathrm{s} \leq 1.2$) latency inflation for the second best path. Those measurements demonstrate that SCIERA not only provides a large number of available paths between AS pairs, but also that there exist alternatives for the fastest paths with similar RTTs in most cases.

Moreover, *path disjointness* characterizes to what degree paths are different from each other, considering the links included in these paths. Path disjointness not only offers benefits in terms of resilience against individual link failures, it also provides promising bandwidth aggregation opportunities that can be leveraged by end-host applications.

We calculate path disjointness for each pair of paths by dividing the number of distinct interfaces by the total number of interfaces for both paths. In Figure 10b we show the path disjointness of all path combinations for every AS pair. We observe that 30% of all path combinations of AS pairs are fully disjoint. Furthermore, 80% of the path pairs still reach a path disjointness of 70%, i.e., having only 30% of links in common.

Finally, we perform a simulation based on the SCIERA topology, to evaluate how link failures affect SCIERA. We compare its multipath capability with a single-path routing alternative that provides only the shortest path between two ASes. In 100 simulation runs, we randomly remove between 0% and 100% of the links (one link per step) and calculate how many AS pairs still have connectivity. This simulation does not relate to the actual link failures we observed while operating SCIERA, but demonstrates the resilience in case larger parts of its topology become unavailable. Figure 10c shows how the connectivity among AS pairs is preserved despite link failures. For example, 90% of all pairs still have connectivity when 20% of the links are failing in the multipath case, whereas this number drops to 50% when using only a single path.

In Section 4.7, we described how we shifted from trying to support every possible use case to focusing on a few specific, compelling scenarios. Particularly those that highlight SCION's strengths in path awareness and multipath communication. This evaluation confirms that path diversity is a practical reality in the SCIERA deployment, offering tangible benefits to its users.

5.6 Operator Survey

We carry out a survey among network operators to assess the capital and operational demands of participating in SCIERA. The survey includes 20 questions focused on three key areas: general deployment experience, capital expenditure (CAPEX), and operational expenditure (OPEX). While participants for the survey were recruited through direct outreach to representatives of each SCIERA participant, yielding eight responses, survey participation was voluntary and anonymous. Among the respondents, 50% have over a decade of experience in networking and security. Half of the respondents are network engineers with hands-on operational expertise, while the other half are network researchers from research institutions and universities.

Our survey reveals that interconnecting with the SCIERA network is both readily feasible and efficient. Notably, 37.5% of respondents completed the native SCION setup—including L2 circuit provisioning and control and data plane configuration—within one month. While another 50% required up to six months, and the remaining cases took longer, the primary cause of delay was the negotiation and provisioning of L2 circuits across multiple networks.

Hardware procurement and software deployment were conducted independently, with only final connection tests needed once L2 provisioning was complete. As a result, the overall deployment timeline was largely dictated by the time required for L2 circuit provisioning. Regarding SCION software deployment, 62.5% of participants successfully implemented it without vendor support.

To assess the financial investment required to join SCIERA, we surveyed network operators about costs related to hardware acquisition, software licensing, and personnel training or hiring. The results indicate that CAPEX is minimal compared to other commercial network solutions. Specifically, 75% of organizations spent less than 20,000 USD on hardware, benefiting from SCION's compatibility with off-the-shelf machines, which reduces dependence on specialized equipment. Software licensing costs varied based on the number of core, router, and gateway instances. Networks using only L2 circuits and open-source SCION versions incurred no licensing costs (62.5%). Additionally, 75% of respondents reported that SCION deployment required no additional hiring or training. When applicable, associated personnel costs were estimated at around 20,000 USD for the project duration.

From an operational cost perspective, 75% of respondents rated SCIERA's expenses as comparable to or lower than their existing infrastructure, while the remaining 25% found them slightly higher. The primary cost drivers were hardware maintenance (62.5%) and staff workload (50%), followed by monitoring and troubleshooting (25%) and power consumption (12.5%). Maintenance costs, including initial infrastructure setup, are expected to decline over time. Likewise, the staff workload associated with deploying and managing a new Internet architecture represents an initial investment that is likely to decrease in the near future. Encouragingly, 87.5% of operators reported that SCIERA-related tasks accounted for less than 10% of their overall operational workload. Most operational challenges—such as cable cuts, hardware failures, and misconfiguration—were typical network maintenance tasks. Issues specific to SCIERA, such as certificate expirations or software updates, were infrequent. As a result, 62.5% of respondents required vendor support fewer than three times per year—a positive outcome given that SCIERA is based on a new Internet architecture.

The operator survey demonstrated that deploying and operating SCIERA is both practical and efficient (Section 4.3 and Section 4.4) compared to traditional network infrastructure. We hope these findings provide valuable insights and set realistic expectations for network operators and researchers, encouraging broader adoption and participation in SCIERA.

6 RELATED WORK

Over the past two decades, several NGN architectures have been proposed [14, 27, 27, 39, 43, 55, 59]. In this process, our community has witnessed the difficulty of deploying these new architectures in commercial networks. Named Data Networking (NDN) [27] stands out as one of the few NGN architectures to have achieved long-term development and deployment, and while SCIERA differs greatly in scope and architecture, we recognize NDN's longevity and impact as an important milestone in the evolution of NGNs. In this section, we focus our discussion specifically on approaches that facilitate the deployment of NGN infrastructures. Although not targeting an NGN architecture, the PlanetLab project represents an important milestone for the creation of a global infrastructure that facilitates experimentation [13].

For research testbeds, the GENI [4] and Fed4FIRE+ [17] infrastructures enable rapid distributed NGN deployment. This supports the initial deployment to collect operational experience and demonstrate the operation of the NGN.

SCIONLab [32] has been the first SCION research network deployed. In contrast to SCIERA, SCIONLab is running as an overlay network over today's BGP-based Internet. Considering BGP-free deployment and the control-plane scalability of the SCION network, Krähenbühl et al. present approaches for infrastructure deployment [31]. In contrast, this paper focuses on the native SCION use on end systems, and enhancing the deployment scalability of new ASes.

The Extensible Internet (EI) [7] is an extension to the Trotsky [36] architecture, proposing to leverage existing hyperscalers' cloud infrastructure to set up distributed service nodes. NGN client nodes and hosts connect to the EI service nodes to obtain access to the NGN which runs on an Internet overlay. Such an infrastructure would be beneficial to start the deployment of a new NGN. The vision of the EI was refined in subsequent work, proposing an InterEdge [6]. The InterEdge is deployed at service providers near the edge of the network, i.e., access ISPs, cloud providers, CDNs, and IXPs. The InterEdge better differentiates between interconnection and enabling enhancements, and sharpens the economic clarity to provide benefits to early adopters.

7 CONCLUSION

Although the deployment of an NGN architecture is a revolution in terms of protocols, we show in this paper that deployment through an evolution of current systems is possible—while maintaining the beneficial properties of the NGN. Through SCIERA we expand the SCION network with research and education institutions spanning 5 continents and providing native SCION connectivity to over 250K people at these institutions. Thanks to the lessons learned in this deployment, the scalability of deploying SCION has increased. Moreover, the end-host bootstrapping and libraries we built, facilitate the creation of native SCION applications, which we anticipate to set in motion an innovation cycle to further expand the deployment and use of SCION. We hope that by sharing our experiences, we can inform and inspire others who are building or deploying NGN technologies.

8 ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their valuable feedback, our shepherd Marwan Fayed for his guidance during the revision, and the anonymous participants of our survey for their contributions to this project. This work was partially supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00440780, Development of Automated SBOM, VEX Verification Technologies for Securing Software Supply Chains), CAPES Finance Code 01, CNPq Procs. 308101/2022-7, 420934/2023-5, and FAPESP Procs. 2023/00812-7, 2020/05183-0, 2023/00811-0.

REFERENCES

[1] Anapaya. ISD and AS Assignments. https://docs.anapaya.net/en/latest/resources/isd-as-assignments/, 2024.

[2] Anapaya. Anapaya CORE. https://www.anapaya.net/core, 2025.

[3] I. Baldin, A. Nikolich, J. Griffioen, I. I. S. Monga, K.-C. Wang, T. Lehman, and P. Ruth. FABRIC: A national-scale programmable experimental network infrastructure. *IEEE Internet Computing*, 23(6):38–47, 2020.

[4] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar. GENI: A federated testbed for innovative network experiments. *Computer Networks*, 61:5–23, 2014.

[5] D. Bhattacherjee, W. Aqeel, S. A. Jyothi, I. N. Bozkurt, W. Sentosa, M. Tirmazi, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla. cISP: A Speed-of-Light Internet Service Provider. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2022.

[6] M. Blumenthal, R. Govindan, E. Katz-Bassett, A. Krishnamurthy, J. McCauley, N. Merrill, T. Narechania, A. Panda, and S. Shenker. Can we save the public internet? *ACM SIGCOMM Computer Communication Review (CCR)*, 53(3), Oct. 2023.

[7] L. Brown, E. Marx, D. Bali, E. Amaro, D. Sur, E. Kissel, I. Monga, E. Katz-Bassett, A. Krishnamurthy, J. McCauley, T. Narechania, A. Panda, and S. Shenker. An architecture for edge networking services. In *Proceedings of ACM Sigcomm*, Aug. 2024.

[8] M. Carney, C. E. Perkins, B. Volz, T. Lemon, and J. Bound. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315, July 2003.

[9] S. Cheshire and M. Krochmal. DNS-Based Service Discovery. RFC 6763, Feb. 2013.

[10] S. Cheshire and M. Krochmal. Multicast DNS. RFC 6762, Feb. 2013.

[11] C. M. Christensen. *The Innovators Dilemma: When New Technologies Cause Great Firms to Fail*. Harvard Business Review Press, Boston, MA, USA, 2016.

[12] L. Chuat, M. Legner, D. Basin, D. Hausheer, S. Hitz, P. Müller, and A. Perrig. *The Complete Guide to SCION. From Design Principles to Formal Verification*. Springer International Publishing AG, 2022.

[13] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. PlanetLab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33, July 2003.

[14] D. Clark, K. Sollins, J. Wroclawski, D. Katabi, J. Kulik, X. Yang, R. Braden, T. Faber, A. Falk, V. Pingali, M. Handley, and N. Chiappa. NewArch: Future generation Internet architecture. Technical report, Air Force Research Labs, 2004.

[15] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski. The Science DMZ: A network design pattern for data-intensive science. In *SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–10, 2013.

[16] S. Dawkins. Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not Taken). RFC 9049, IETF, June 2021.

[17] European Commission. Fed4FIRE+: Federation for FIRE plus. https://www.fed4fire.eu. Accessed: 2025-06-26.

[18] M. Gartner, J.-P. Smith, M. Frei, F. Wirz, C. Neukom, D. Hausheer, and A. Perrig. Hercules: High-speed bulk-transfer over SCION. In *Proceedings of IFIP Networking*, June 2023.

[19] M. Gartner and F. Wirz. SCION Go Multiping Dataset obtained in the SCIERA Network. https://doi.org/10.5281/zenodo.15672257.

[20] M. Gartner and F. Wirz. SCION Go Multiping Tool. https://github.com/FR4NK-W/scion-go-multiping/tree/sigcomm-results.

[21] George Mason University. BRIDGES: A Programmable Cyber-System Funded By NSF. https://cnl.gmu.edu/bridges/, Dec. 2022. Accessed: 2025-01-12.

[22] A. Gulbrandsen and D. L. Esibov. A DNS RR for specifying the location of services (DNS SRV). RFC 2782, Feb. 2000.

[23] T. Herbert and W. de Bruijn. Scaling in the linux networking stack. https://www.kernel.org/doc/Documentation/networking/scaling.txt. Accessed: 2025-06-26.

[24] T. Høiland-Jørgensen, J. D. Brouer, D. Borkmann, J. Fastabend, T. Herbert, D. Ahern, and D. Miller. The express data path: fast programmable packet processing in the operating system kernel. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '18, page 54–66, New York, NY, USA, 2018. Association for Computing Machinery.

[25] Internet2. Layer 2 Service. https://internet2.edu/services/layer-2-service/, 2024. Accessed: 2025-06-26.

[26] J. Iyengar and M. Thomson. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, IETF, May 2021.

[27] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of the International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2009.

[28] E. Jeong, S. Woo, M. Jamshed, H. Jeong, S. Ihm, D. Han, and K. Park. mTCP: a highly scalable user-level TCP stack for multicore systems. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Apr. 2014.

[29] A. Jony and M. N. Islam. An effective technique to automatically detect and neutralize rogue dhcp server. In *2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pages 244–248. IEEE, 2023.

[30] X. Jun. bat. https://github.com/astaxie/bat, Dec. 2019. Accessed: 2024-09-11.

[31] C. Krähenbühl, S. Tabaeiaghdaei, C. Gloor, J. Kwon, A. Perrig, D. Hausheer, and D. Roos. Deployment and scalability of an inter-domain multi-path routing infrastructure. In *Proceedings of the International Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, 2021.

[32] J. Kwon, J. A. García-Pardo, M. Legner, F. Wirz, M. Frei, D. Hausheer, and A. Perrig. SCIONLAB: A Next-Generation Internet Testbed. In *IEEE International Conference on Network Protocols (ICNP)*, 2020.

[33] K. Lee, D. Chu, E. Cuervo, J. Kopf, Y. Degtyarev, S. Grizan, A. Wolman, and J. Flinn. Outatime: Using Speculation to Enable Low-Latency Continuous Interaction for Mobile Cloud Gaming. In *Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 151–165, 2015.

[34] J. B. Littlefield. Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4). RFC 3925, Oct. 2004.

[35] S. Madanapalli, J. P. Jeong, S. D. Park, and L. Beloeil. IPv6 Router Advertisement Options for DNS Configuration. RFC 6106, Nov. 2010.

[36] J. McCauley, Y. Harchol, B. Raghavan, A. Panda, and S. Shenker. Enabling a permanent revolution in internet architecture. In *ACM Conference on Data Communication (SIGCOMM)*, Aug. 2019.

[37] M. H. Mealling and D. R. Daniel. The Naming Authority Pointer (NAPTR) DNS Resource Record. RFC 2915, Sept. 2000.

[38] MystenLabs. SCION in Rust. https://github.com/MystenLabs/scion-rs/, 2024. Accessed: 2024-09-16.

[39] D. Naylor, M. K. Mukerjee, P. Agyapong, R. Grandl, R. Kang, M. Machado, S. Brown, C. Doucette, H.-C. Hsiao, D. Han, T. H.-J. Kim, H. Lim, C. Ovon, D. Zhou, S. B. Lee, Y.-H. Lin, C. Stuart, D. Barrett, A. Akella, D. Andersen, J. Byers, L. Dabbish, M. Kaminsky, S. Kiesler, J. Peha, A. Perrig, S. Seshan, M. Sirbu, and P. Steenkiste. XIA: Architecting a More Trustworthy and Evolvable Internet. *ACM SIGCOMM Computer Communication Review (CCR)*, 44(3):50–57, 2014.

[40] NetSec ETH Zurich. PAN API in scion-apps. https://github.com/netsec-ethz/scion-apps/, 2024. Accessed: 2024-09-16.

[41] NetSys Lab OVGU Magdeburg. scion-orchestrator: Run a cross-platform SCION Host. https://github.com/netsys-lab/scion-orchestrator/, 2024. Accessed: 2025-06-24.

[42] O. Ogundipe and T. Aweto. The shaky foundation of global technology: A case study of the 2024 crowdstrike outage. *International Journal of Multidisciplinary Research and Growth Evaluation*, 09 2024.

[43] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani. MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet. *ACM SIGMOBILE Mobile Computing and Communications Review*, July 2012.

[44] G. Ribeiro, L. Pedrosa, S. Signorello, and F. M. Ramos. Internet architecture evolution: Found in translation. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*, pages 300–307, 2024.

[45] RNP. Ipê network. https://www.rnp.br/en/ipe-network, Dec. 2022. Accessed: 2024-09-25.

[46] S. Scherrer, S. Tabaeiaghdaei, and A. Perrig. Quality competition among internet service providers. In *Proceedings of the International Symposium on Computer Performance, Modeling, Measurements and Evaluation (PERFORMANCE)*, 2023.

[47] D. Schinazi and T. Pauly. Happy Eyeballs Version 2: Better Connectivity Using Concurrency. RFC 8305, IETF, Dec. 2017.

[48] SCION open-source contributors. SCION open-source implementation. https://github.com/scionproto/scion/, 2025. Accessed: 2024-09-16.

[49] SIX. Secure Swiss Finance Network (SSFN) For secure, flexible and resilient data communication. https://www.six-group.com/en/products-services/banking-services/ssfn.html, 2021.

[50] Smallstep. Smallstep API. https://github.com/smallstep, 2025. Accessed: 2025-06-24.

[51] I. Society. Enabling technologies. https://pulse.internetsociety.org/technologies#ipv6, May 2024. Accessed: 2025-06-26.

[52] Swiss National Bank. SNB and SIX launch the communication network Secure Swiss Finance Network . https://www.snb.ch/en/publications/communication/press-releases/2021/pre_20210715, 2021. Accessed: 2025-06-26.

[53] SWITCH. SCION-basierte Science DMZ. https://static.www.switch.ch/sites/default/files/2023-10/switch-report-scion-basierte-science-dmz-de.pdf, Oct. 2023. Accessed: 2024-07-09.

[54] S. Tabaeiaghdaei, S. Scherrer, J. Kwon, and A. Perrig. Carbon-aware global routing in path-aware networks. In *Proceedings of ACM International Conference on Future Energy Systems (e-Energy)*, 2023.

[55] Y. Wang, F. Esposito, I. Matta, and J. Day. RINA: An architecture for policy-based dynamic service management. Technical Report BUCS-TR-2013-014, Boston University, November 2013.

[56] Wikipedia. Metcalfe's law. https://en.wikipedia.org/wiki/Metcalfe%27s_law, Jan. 2024. Accessed: 2025-06-26.

[57] D. Wing and A. Yourtchenko. Happy Eyeballs: Success with Dual-Stack Hosts. RFC 6555, IETF, Apr. 2012.

[58] A. R. Wirth. Glarner Kantonalbank betreibt als erste Bank ihre Bancomaten über das Secure Swiss Finance Network. https://glkb.ch/newsdetails/-/article/mm-21032024-bancomaten-auf-ssfn-umgestellt, 2024. Accessed: 2025-06-26.

[59] T. Wolf, J. Griffioen, K. L. Calvert, R. Dutta, G. N. Rouskas, I. Baldine, and A. Nagurney. ChoiceNet: toward an economy plane for the Internet. *ACM SIGCOMM Computer Communication Review*, 44(3):58–65, July 2014.

[60] T. Young. GÉANT Testbeds Service - Flexible, Powerful, Tools for Advanced Research. https://wiki.geant.org/display/PLMTES/GEANT+Testbed+Service, Nov. 2014. Accessed: 2025-06-26.

[61] T. Zaeschke. JPAN - Java API for path aware networking with SCION. https://github.com/scionproto-contrib/jpan/, 2024. Accessed: 2024-09-16.

APPENDIX

Appendices are supporting material that has not been peer-reviewed.

A BOOTSTRAPPING DISCOVERY METHODS

Table 2: Preferred hinting mechanisms in relation to existing technologies in the target network. "Y" stands for available mechanisms, "M" for mechanisms available in combination with other mechanisms, and "N" for not applicable or unavailable mechanisms.

	Static IPs only	dyn. DHCP leases	dyn. DHCPv6 lease	IPv6 RAs	local DNS search domain
DHCP VIVO	N	Y	N	N	N
DHCPv6 VSIO	N	N	Y	N	N
IPv6 NDP	N (Y if IPv6)	N	M	Y	Y
DNS -SRV	N	M	M	Y	Y
DNS-SD	N	M	M	Y	Y
mDNS	Y	M	M	Y	Y
DNS-NAPTR	N	M	M	Y	Y

In this section, we provide an overview of possible bootstrapping mechanisms that can be used to bootstrap hosts (Section 4.1). The choice of the bootstrapping mechanism is primarily guided by the existing deployment of zero-conf mechanisms in a network domain, the administrative access required to deploy each of the methods, and the familiarity of the operators with each method.

A.1 DHCP Based

DHCP-based solutions require the presence of a DHCP server in the broadcast domain of the clients using one of these bootstrapping methods. Larger networks often deploy multiple DHCP servers, often even from different vendors, making centralized configuration management more challenging.

DHCP Vendor-Identifying Vendor Option (VIVO). This option allows both the IP address and port of the bootstrapping server to be encoded in a DHCP option, utilizing the specifically allocated Private Enterprise Number (PEN) to identify the option structure for the bootstrapping hint [34]. DHCPv6 introduces the Vendor-Specific Information Option (VSIO) [8], relying on the same PEN based option matching and code/length/value field encoding as for the IPv4 VIVO.

DHCP Server Option. DHCP provides a default server option[5] which we use to provide the bootstrapping server IP address in cases where custom DHCP options cannot be configured.

IPv6 NDP. With IPv6 and the Neighbor Discovery Protocol [35], DNS resolvers and DNS search lists can be advertised in-band as part of the router advertisements (RA), enabling the network administrators to provide network segment specific DNS configurations for the DNS based discovery mechanisms described in the next subsection.

A.2 DNS Based

The DNS-based bootstrapping mechanisms rely on the fact that public DNS records providing the hints are set under the local DNS search domain or that clients obtain their DNS responses from resolvers located in the network they are bootstrapping into.

DNS Service (SRV) Records. For the DNS-SRV [22] based lookup, a service record for the name '_sciondiscovery._tcp' is set under the search domain pointing to the bootstrapping server.

DNS Naming Authority Pointer (NAPTR) Record. The DNS NAPTR Record [37] for 'x-sciondiscovery:TCP' is set at the search domain with a value specifying either the "S" flag with the service record for the bootstrapping server, or the "A" flag with the hostname of the bootstrapping server directly resolving to an IPv4 or IPv6 address.

DNS Service Discovery. For DNS service discovery (DNS-SD) [9] a pointer record (PTR) for '_sciondiscovery._tcp' is set under the search domain, which points to a service record entry for the bootstrapping service, which in turn contains the hostname(s) of the bootstrapping server(s).

Multicast DNS. For multicast DNS [10] to function, the clients need to be in the same broadcast domain as the bootstrapping server. The bootstrapping server responds to multicast DNS requests from the clients for the service '_sciondiscovery._tcp' in the corresponding DNS search domain.

B DEPLOYMENT SUPPORT

We support various deployment models that meet the needs of each participating network and provide technical support through multiple channels.

B.1 Enabling Diverse Operational Models

The SCIERA networks aim not only to allow students and researchers to experiment with the SCION architecture, gain experience, and benefit from the improved security and performance of SCION, but also to offer new opportunities to network operators. Network operators participating in the SCION research and education networks can choose various deployment models based on their desired network operation direction and vision. We broadly categorize these into three models:

Internet AS Model. This model replicates the existing Internet AS context. It involves operating a centralized control service with a cohesive routing policy for all entities within the network as

[5] The 'www-server' option, option field ID 72, also known as "Default WWW server" option, allows to define a default WWW server on the network to be used to retrieve web resources.

a single domain. The difference from the existing BGP-based operation is that multipath construction and utilization for a single destination are achieved using multiple border routers deployed within. Therefore, it is generally recommended that both transit and customer ASes establish at least two physical links with neighboring or provider networks to ensure opportunities for multipath communication.

Multi-AS Model. A network operator seeking more sophisticated intra-domain routing can run multiple virtual SCION ASes within the network. KREONET, a Korean NREN, is distinct from other NRENs with its ring connection circumnavigating the Northern Hemisphere. This global ring connection, which links Amsterdam, Chicago, Seattle, Daejeon, Hong Kong, Singapore, and Amsterdam, could allow communication through links connected to the east and west from any region. However, the current BGP-based Internet routes the majority of traffic through Pacific and Atlantic links. KREONET has implemented intuitive and immediate routing control within KREONET by establishing a dedicated AS at each PoP for each region, enabling internal packet routing via both east and west-bound forwarding paths simultaneously.

Edge (Non-AS) Model. ASes with limited resources or those that want to experience NGN without the complexities of NGN management can opt for the Edge model, leveraging the Anapaya Edge server that acts as a SCION border router, as well as a SCION-IP Gateway (SIG) that can transparently translate SCION into IP packets and vice-versa. This model allows the participating AS to become a logical extension of the provider AS, using the provider's NGN infrastructure (e.g., control services and routers) to communicate with other networks. While the participating AS's ability to establish independent routing policies is limited, end hosts can be NGN-enabled with minimal deployment effort by simply running an Anapaya EDGE appliance.

B.2 Technical Support

Another important factor in encouraging network participation is the availability of broad technical support. Running a network requires a delicate operation and a deep understanding of the underlying network technologies. In most cases, even for technology enthusiasts, fully understanding design principles and technical details of a new network architecture demands time and effort. When network providers wish to experience NGN—beyond the readiness of hardware and software required—the availability of channels to share expertise is an intangible factor that greatly influences the deployment. Through the SCIERA initiative and in collaboration with the SCION association, various efforts have lowered the bar to access SCION expertise. The bridged SCION Matrix[6] and Slack[7] channels allows both newcomers and existing participants to share their current issues and experiences. Through weekly developer and operator meetings that anyone can join, new ideas, future directions, and operational issues related to SCION and its applications are discussed.[8] If requested, experienced SCION engineers provide technical support and consulting, alleviating any technical complexities the participants might encounter.

C DEPLOYMENT OVER TIME

Our first efforts in deploying SCIERA date back to February 2018, where we presented our proposal for a SCION Pilot Service within GÉANT at the Services and Technology Forum (STF), a meeting of National Research and Education Network (NREN) representatives in Europe. Following this meeting, we got the opportunity to deploy SCION on the GÉANT Testbed Service (GTS) [60], an experimental network facility spanning the GÉANT infrastructure. GTS enabled us to deploy SCIONLab [32] instances on VMs in several GÉANT PoPs and interconnect them via virtual circuits with external domains including SWITCH, SIDN Labs, and DFN, as well as other testbeds including Grid5000 in France and VirtualWall at Interuniversity Microelectronics Centre (imec). However, since GTS was focused on rather short-term experimental research, it was quite challenging for an operational deployment, as the facility, e.g., did not offer any support for deployment changes and required a manual configuration of links to external domains. In June 2020, we were able to migrate our GTS deployment into a more stable setup with three bare metal servers in different PoPs (AMS, PAR, HAM), which offered much higher performance compared to the initial VM-based setup.

However, as GTS was going to be decommissioned by September 2021, we started discussions with representatives from GÉANT, SWITCH, and DFN to find a more permanent deployment possibility for SCION within GÉANT. These efforts finally resulted in a Memorandum of Understanding between GÉANT and ETHZ as well as a service agreement between GÉANT and SWITCH, that enabled the deployment of SCION on dedicated servers at three GÉANT PoPs (GVA, PAR, FRA), along with the opportunity to connect these instances with other SCION ASes over GÉANT Plus links (L2 VLANs). This setup marked the beginning of the SCIERA deployment, as it allowed the roll-out of the SCION production connectivity in GÉANT and the connection to the existing commercial SCION network.

The SCION setup in GÉANT required a major effort. Most of the effort, however, was required for the hardware and software purchase, shipping, and installation of SCION servers, which took several months, whereas the setup and configuration of the servers with Anapaya's SCION software was later completed within only a few days. The first production SCION border router (GVA) in GÉANT was finally up and running in June 2022, while the remaining two (PAR, FRA) followed in August and October 2022, respectively. Since GÉANT became a SCION Core AS within a newly established SCION Education ISD (71), we needed to set up and configure our own CA based on open source software, which required a few weeks—partly to acquire expertise on PKI infrastructure management.

In comparison, connecting SWITCH to ISD 71 was rather straightforward, as they were already very experienced with SCION and connected to ISD 64 before. Only minimal additional effort was required to establish a core connection to ISD 64 via SWITCH as well. Also, connecting SIDN Labs was quite straightforward, as their AS was already connected to SCIONLab before, and establishing two additional VLANs did not take much effort.

However, setting up SCION in BRIDGES [21] took again more time, similar to GÉANT. Much of the effort was again used for

[6] https://matrix.to/#/#scion:matrix.scion.org
[7] https://scionproto.slack.com
[8] https://sciera.readthedocs.io/en/latest/scion-research/meetings/index.html

hardware procurement, but also establishing and troubleshooting point-to-point VLANs back to GÉANT took around 1.5 months, requiring numerous email exchanges. Also, when the BRIDGES infrastructure needed to be migrated, much effort was required to migrate all established VLANs. Deploying SCION core ASes in BRIDGES, however, was rather easy using the SCIONLab open source stack.

UVa was the first site to be connected via BRIDGES. To this end, most effort was required to establish a range of VLANs between the two sites, as many parties needed to collaborate to approve and implement them. Fortunately, once VLANs are in place, they are usually running quite reliably (unless they need to be migrated). Also, based on the established VLANs, it is only a small step to run SCION connections over it. However, since UVa was among our first customer ASes, we experienced some early deployment troubles, including configuration and time synchronization issues as well as path expiration problems that required much effort to investigate.

We then also connected Equinix in Ashburn to BRIDGES, via a cross-connect within the data center. Establishing a VLAN between the two sites actually took more effort than initially expected, with numerous troubleshooting sessions to investigate why there was no signal over this connection.

On the other hand, connecting Cybexer in Estonia to GÉANT was again very fast, as Cybexer only needed to request two GÉANT Plus links via EENet, which terminate directly on our SCION border routers within GÉANT.

Connecting Princeton again required more effort. Similar to UVa, establishing a range of point-to-point VLANs that, in this case, required the involvement of 4 parties (BRIDGES, Internet2, NJEdge, and Princeton) was very cumbersome.

In contrast, connecting Demokritos (similar to Cybexer) was straightforward as it only required requesting a GÉANT Plus link via GRNet.

Deploying SCION productively over KISTI's Kreonet required much effort, mainly due to its large footprint across three continents. Since these nodes were previously used for SCIONLab, most effort was spent to reinstall the machines with the SCION stack from Anapaya, which proved a bit difficult as management access to the servers was limited. Also, establishing VLANs from GÉANT to KISTI took again some effort, as these VLANs needed to be requested by SWITCH, initiated from GÉANT side, and coordinated with both SingAREN and KISTI. However, once these VLANs were in place, configuring the SCION instances was straightforward, despite the complex setup with several distinct core ASes.

Establishing connectivity with the SEC in Singapore presented unique challenges, since it was not possible in their case to establish a native VLAN to Kreonet in Singapore, but only a VXLAN over SingAREN.

Later on, when UFMS approached us to connect to SCIERA via RNP, we already knew how to trigger the VLAN setup from the GÉANT side, and also GÉANT/SWITCH were already experienced for this setup, so the connection setup was done very quickly. To improve the connectivity, the goal was to connect UFMS also with BRIDGES and Kreonet in the US. However, we refrained from establishing further point-to-point VLANs over Internet2, as it would have required the involvement of Kreonet and BRIDGES as well. Fortunately, Internet2 offers the possibility to establish multipoint-VLANs. Princeton kindly helped us to request these multipoint-VLANs and establish a SCIERA organization within AL2S, which enables connecting additional sites in the future via shared VLANs, greatly reducing the effort for connecting additional entities.

Building up on the collected experience, the SCION connections of NUS in Singapore and CCDCoE over EENet were straightforward on our side. CCDCoE was even able to reuse the existing VLANs established by Cybexer over EENet, so no new links needed to be requested.

Finally, while the SCION deployment at KAUST took again a bit more time due to a long-lasting hardware delivery, the most recent SCION deployments in 2025 at RNP as well as KISTI HK and STL took considerably less effort than previous comparable setups.

Several universities, research institutes, and NRENs have already expressed interest in joining SCIERA additionally. Among them are, e.g., UIUC, University of Amsterdam, SURF, CERN, TUM, TUB, and many more. We expect that our collected experience and management tools will allow us to greatly accelerate the number of ASes that will be connected to SCIERA in the future.

D SCION NSPS

The SCION ecosystem has steadily expanded over recent years, with over 20 NSPs that offer SCION connectivity. In alphabetical order: Anapaya, Axpo Systems, BICS, BSO Network Solutions, British Telecom (BT), Celeste, COLT, Cyberlink, Everyware, GÉANT, Iristel / Karrier One, KREONET, Litecom, LG U+, Megaport, Odido, Proximus Luxembourg, RNP, Sunrise, Swisscom, SWITCH, Varity BV, and VTX Services.

SCION connectivity is also available via peering groups or L2 connections at IXPs, such as BBIX, LINX, NYIIT, and SwissIX.[9] As a data center operator, Digital Realty (DLR) is offering SCION connectivity at over 450 of their data centers via their ServiceFabric Connect offering.

Several cloud providers offer SCION connectivity through third-party connectivity providers and offerings in their respective marketplaces, such as at AWS, Azure, and GCP. Several smaller cloud operators are starting to offer native SCION connectivity directly through their infrastructure, such as at Cherry Servers, and cloudscale.ch.

The SCION association provides a list of SCION service providers and NSPs providing SCION services.[10] Finally, Anapaya publishes a list of ISDs and SCION ASes, with currently over 200 registered ASes.[11]

[9] SwissIX maintains an updated list of participant NSPs offering SCION peering at their facilities. https://www.swissix.ch/services/scion-peering-mesh/scion-peering-participants/

[10] https://www.scion.org/business/#providers

[11] https://docs.anapaya.net/en/latest/resources/isd-as-assignments/

E SCIONABLING DIFFS (BAT)

This section shows the changes made to the bat[12] application to allow the application to use SCION[13], as described in Section 5.2.

```
@@ -32,6 +32,9 @@
        "strconv"
        "strings"
+
+       "github.com/netsec-ethz/scion-apps/pkg/pan"
+       "github.com/netsec-ethz/scion-apps/pkg/shttp"
 )

 const (
@@ -43,6 +46,9 @@
 )

 var (
+       interactive      bool
+       sequence         string
+       preference       string
        ver              bool
        form             bool
@@ -64,6 +70,11 @@
 )

 func init() {
+       flag.BoolVar(&interactive, "interactive", false, "Prompt user for interactive path selection")
+       flag.StringVar(&sequence, "sequence", "", "Sequence of space separated hop predicates to specify path")
+       flag.StringVar(&preference, "preference", "", "Preference sorting order for paths. "+
+               "Comma-separated list of available sorting options: "+
+               strings.Join(pan.AvailablePreferencePolicies, "|"))
        flag.BoolVar(&ver, "v", false, "Print Version Number")
        flag.BoolVar(&ver, "version", false, "Print Version Number")
@@ -111,6 +122,11 @@
        flag.Usage = usage
        flag.Parse()
+       policy, err := pan.PolicyFromCommandline(sequence, preference, interactive)
+       if err != nil {
+               log.Fatal(err)
+       }
+       defaultSetting.Transport, _ = shttp.NewTransport(nil, policy)
        args := flag.Args()
        if len(args) > 0 {
@@ -153,7 +169,7 @@
        if !strings.HasPrefix(*URL, "http://") && !strings.HasPrefix(*URL, "https://") {
                *URL = "http://" + *URL
        }
-       u, err := url.Parse(*URL)
+       u, err := url.Parse(shttp.MangleSCIONAddrURL(*URL))
        if err != nil {
                log.Fatal(err)
        }
@@ -177,7 +193,7 @@
        // Proxy Support
        if proxy != "" {
-               purl, err := url.Parse(proxy)
+               purl, err := url.Parse(shttp.MangleSCIONAddrURL(proxy))
                if err != nil {
                        log.Fatal("Proxy Url parse err", err)
                }
```

[12] https://github.com/astaxie/bat

[13] https://github.com/netsec-ethz/scion-apps/tree/master/bat

F SCIONABLING DIFFS (CADDY PLUGIN)

This section shows the changes to make a caddy[14] plugin acting as a reverse proxy serving content via SCION, as described in Section 5.2.

```
headers.go
@@ -6,4 +6,5 @@ import (
         "github.com/caddyserver/caddy/v2"
         "github.com/caddyserver/caddy/v2/modules/caddyhttp"
+        "github.com/scionproto/scion/pkg/snet"
 )

@@ -17,4 +18,5 @@ type Middleware struct{}
 func (Middleware) CaddyModule() caddy.ModuleInfo {
         return caddy.ModuleInfo{
+                ID:   "http.handlers.scion",
                 New: func() caddy.Module { return new(Middleware) },
         }
@@ -23,4 +25,10 @@ func (Middleware) CaddyModule() caddy.ModuleInfo {
 // ServeHTTP implements caddyhttp.MiddlewareHandler.
 func (Middleware) ServeHTTP(w http.ResponseWriter, r *http.Request, next caddyhttp.Handler) error {
+        if _, err := snet.ParseUDPAddr(r.RemoteAddr); err == nil {
+                r.Header.Add("X-SCION", "on")
+                r.Header.Add("X-SCION-Remote-Addr", r.RemoteAddr)
+        } else {
+                r.Header.Add("X-SCION", "off")
+        }
         return next.ServeHTTP(w, r)
 }
```

```
network.go
@@ -26,4 +26,10 @@ import (

         "github.com/caddyserver/caddy/v2"
+        "github.com/scionproto/scion/pkg/addr"
+        "github.com/scionproto/scion/pkg/snet"
+        snetmetrics "github.com/scionproto/scion/pkg/snet/metrics"
+        "github.com/scionproto/scion/pkg/sock/reliable"
+        "github.com/scionproto/scion/pkg/sock/reliable/reconnect"
+        "github.com/scionproto/scion/private/app/env"
         "go.uber.org/zap"
 )
@@ -39,9 +45,15 @@ var (
         globalNetwork = Network{
                 connPool: NewUsagePool[string, *conn](),
+                nets:       map[addr.IA]*snet.SCIONNetwork{},
         }

         envFile = func() string {
+                if file := os.Getenv("SCION_ENV_FILE"); file != "" {
+                        return file
+                }
+                return "/etc/scion/environment.json"
         }()

+        metrics = snetmetrics.NewSCIONPacketConnMetrics()
 )
```

[14] https://github.com/scionassociation/scion-caddy/

```
@@ -51,4 +63,5 @@ type Network struct {

        netsMtx sync.Mutex
+       nets    map[addr.IA]*snet.SCIONNetwork

        logger  atomic.Pointer[zap.Logger]
@@ -74,4 +87,8 @@ func (n *Network) ListenBlocked(
        cfg net.ListenConfig,
 ) (any, error) {
+       if network != "scion" {
+               return nil, fmt.Errorf("network not supported: %s", network)
+       }
+       conn, err := n.listen(ctx, address, cfg)
        if err != nil {
                return nil, err
@@ -88,4 +105,8 @@ func (n *Network) Listen(
        cfg net.ListenConfig,
 ) (any, error) {
+       if network != "scion+quic" {
+               return nil, fmt.Errorf("network not supported: %s", network)
+       }
+       conn, err := n.listen(ctx, address, cfg)
        if err != nil {
                return nil, err
@@ -99,4 +120,43 @@ func (n *Network) Listen(
 // configuration changes.
 func (n *Network) listen(ctx context.Context, address string, cfg net.ListenConfig) (*conn, error) {
+       listen, err := snet.ParseUDPAddr(address)
+       if err != nil {
+               return nil, fmt.Errorf("parsing listening address: %w", err)
+       }
+       if listen.Host.Port == 0 {
+               return nil, fmt.Errorf("wildcard port not supported: %s", address)
+       }
+       env, err := n.loadEnv()
+       if err != nil {
+               return nil, fmt.Errorf("loading environment configuration data: %w", err)
+       }
+       if _, ok := env.ASes[listen.IA]; !ok {
+               return nil, fmt.Errorf(
+                       "listening address (%s) not covered by configured ASes %q",
+                       address,
+                       env.ASes,
+               )
+       }
+
+       network := func() *snet.SCIONNetwork {
+               n.netsMtx.Lock()
+               defer n.netsMtx.Unlock()
+               if net, ok := n.nets[listen.IA]; ok {
+                       return net
+               }
+               net := &snet.SCIONNetwork{
+                       LocalIA: listen.IA,
+                       Dispatcher: &snet.DefaultPacketDispatcherService{
+                               Dispatcher: reconnect.NewDispatcherService(
+                                       reliable.NewDispatcher(env.General.DispatcherSocket),
+                               ),
+                               SCMPHandler:            ignoreSCMP{},
+                               SCIONPacketConnMetrics: metrics,
+                       },
+               }
+               n.nets[listen.IA] = net
+               return net
+       }()
+
        c, loaded, err := n.connPool.LoadOrNew(address, func() (caddy.Destructor, error) {
                ctx, cancel := context.WithTimeout(ctx, 2*time.Second)
@@ -121,5 +181,18 @@ func (n *Network) listen(ctx context.Context, address string, cfg net.ListenConf
 }

+func (n *Network) loadEnv() (env.SCION, error) {
+       raw, err := os.ReadFile(envFile)
+       if err != nil {
+               return env.SCION{}, err
+       }
+       var e env.SCION
+       if err := json.Unmarshal(raw, &e); err != nil {
+               return env.SCION{}, err
+       }
+       return e, nil
+}
+
 type conn struct {
+       net.PacketConn
        addr    string
        network *Network
@@ -162,2 +235,11 @@ func (l *blockedListener) Addr() net.Addr {
        return l.conn.LocalAddr()
 }
+
+// ignoreSCMP is a SCMP handler that ignores all SCMP messages. This is required
+// because SCMP error messages should not close the accept loop.
+type ignoreSCMP struct{}
+
+func (ignoreSCMP) Handle(pkt *snet.Packet) error {
+       // Always reattempt reads from the socket.
+       return nil
+}

scion.go
@@ -36,4 +36,8 @@ func init() {
        globalNetwork.logger.Store(zap.NewNop())

+       caddy.RegisterModule(SCION{Network: &globalNetwork})
+       caddy.RegisterNetwork("scion", globalNetwork.ListenBlocked)
+       caddy.RegisterNetwork("scion+quic", globalNetwork.Listen)
+       caddyhttp.RegisterNetworkHTTP3("scion", "scion+quic")
 }

@@ -41,2 +45,19 @@ func init() {
 // logger for the global network. In the future, additional configuration can be
 // parsed with this component.
+type SCION struct {
+       Network *Network
+}
+
+func (SCION) CaddyModule() caddy.ModuleInfo {
+       return caddy.ModuleInfo{
+               ID:  "scion",
+               New: func() caddy.Module {
+                       return new(SCION)
+               },
+       }
+}
+
+func (s *SCION) Provision(ctx caddy.Context) error {
+       s.Network.SetLogger(ctx.Logger(s))
+       return nil
+}
```

G SCIONABLING DIFFS (JAVA NETCAT EXAMPLE)

This section shows the minimal changes required to a netcat sample application[15] to add SCION support using the SCION java library JPAN[16].

```
NetcatUDPClient.java
@@ -6,4 +6,6 @@ import java.net.InetAddress;
 import java.util.Scanner;

+import org.scion.jpan.*;
+
 /**
  * Binds to a specified port number on the local host and waits for a connection request from a client. Once a
@@ -27,5 +29,5 @@ public class NetcatUDPClient {
     @SuppressWarnings("InfiniteLoopStatement")
     private static void start(String host, int port) throws Exception {
-        clientSocket = new DatagramSocket();
+        clientSocket = new ScionDatagramSocket();
         ipAddress = InetAddress.getByName(host);
         if (System.in.available() > 0) {
@@ -80,4 +82,5 @@ public class NetcatUDPClient {
      */
     public static void main(String[] args) throws Exception {
+        System.setProperty(org.slf4j.simple.SimpleLogger.DEFAULT_LOG_LEVEL_KEY, "ERROR");
         if (args[0] != null && args[1] != null) {
             start(args[0], Integer.parseInt(args[1]));
```

```
NetcatUDPServer.java
@@ -5,4 +5,6 @@ import java.net.DatagramSocket;
 import java.util.Scanner;

+import org.scion.jpan.*;
+
 /**
  * Establishes a connection to the server on the given host name (or IP address) and port number and operates in one of
@@ -28,5 +30,5 @@ public class NetcatUDPServer {
     @SuppressWarnings("InfiniteLoopStatement")
     private static void start(int port) throws Exception {
-        serverSocket = new DatagramSocket(port);
+        serverSocket = new ScionDatagramSocket(port);
         if (System.in.available() > 0) {
             downloadMode = true;
@@ -79,4 +81,5 @@ public class NetcatUDPServer {
      */
     public static void main(String[] args) throws Exception {
+        System.setProperty(org.slf4j.simple.SimpleLogger.DEFAULT_LOG_LEVEL_KEY, "ERROR");
         if (args[0] != null) {
             start(Integer.parseInt(args[0]));
```

[15] https://github.com/tlmader/netcat
[16] https://github.com/scionproto-contrib/jpan