Supporting Dynamic Secure Interdomain Routing

Lars-Christian Schulz*, Elham Ehsani Moghadam†, Juan A. Garcia-Pardo†, David Hausheer*, Ken Calvert‡
*OVGU Magdeburg {lars-christian.schulz, hausheer}@ovgu.de
†ETH Zürich {elham.ehsanimoghadam, juan.garcia}@inf.ethz.ch
‡University of Kentucky calvert@netlab.uky.edu

***Abstract*—Path aware networking (PAN) is an approach that allows endpoints to participate in the end-to-end path selection, letting them choose paths best suited for each application. This approach offers numerous potential benefits including rapid fail-over, concurrent use of parallel paths, and QoS enabled networks, even spanning multiple domains. The dynamic interconnection of different autonomous systems (ASes) in path aware networks offers both challenges and opportunities for network service providers, which in turn provide opportunities for traffic engineering previously not possible.**

The SCION path-aware network architecture has been designed from the ground up with security in mind, and features a trust structure that can serve as a basis for more dynamic interconnection between ASes. In this paper, we describe a prototype *spot market* that lets the ASes sell time-limited excess capacity, allowing buyers to divert traffic to cheaper alternatives temporarily. We believe this market allows for new opportunities both in traffic engineering and inter-domain connectivity that have not existed before. The market benefits all parties involved, as the formerly wasted bandwidth is now used, and provides additional revenue—in varying degrees—to all the participating entities.

I. Introduction and Motivation

The global Internet evolved from a research experiment to critical infrastructure in about 40 years. Partially as a result of this evolutionary path, the business-to-business interfaces of the Internet were essentially an afterthought, developing organically. The engineers developing the standards were not originally concerned with "money flow"; today the flow of compensation between service providers is coarse-grained, slow-changing, and shrouded in mystery, despite its obvious importance. Transaction costs for autonomous systems (ASes) to agree to carry one another's traffic are decreasing, but remain high. Moreover, the technical interface between ASes, namely the Border Gateway Protocol [20], was simply not designed for today's security environment. This has well-known undesirable consequences: For one thing, the global Internet routing system based on BGP is known to be subject to a wide variety of attacks. Although countermeasures have been developed [5], [17], deployment is slow, and some have no known mitigation today [22]. Currently, the ability of the global routing system to detect and respond to routing incidents relies heavily on human observation and human reaction times. Trust relationships among ASes are complex and sometimes unclear, because in general they must cooperate in order to provide end-to-end network service, while at the same time they may compete for customers. Barriers to entry for new providers are high, stifling competition. Finally, there is essentially no incentive for an individual provider to offer end-to-end "premium" services, because there is no global mechanism for distributing any additional revenue such a service might generate to other providers, who must of necessity be involved in implementing the service.

978-1-6654-8234-9/22/$31.00 ©2022 IEEE

The high-level motivation for this work is the hypothesis that lowering transaction costs and making explicit the flow of value (and thus trust relationships) between providers can lead to a more dynamic and secure Internet. Our thesis is that enabling automatic or semi-automatic negotiation of contracts at a finer granularity, with a menu of standard and commonly-understood definitions of "service", will reduce risk and offer opportunities for providers to gain additional revenue. For example, a "spot market" for cheap transit service could enable providers to monetize their excess capacity during off-peak times, while customers could save money on certain types of traffic [25]. Although there have been some recent efforts in that direction [8], [18], they rely on the existing BGP-based system and therefore inherit the aforementioned problems.

A fundamental challenge for such a system is how the *trust* necessary for economic transactions can be established between providers. Fortunately, the SCION architecture [10], [27] is designed from the ground up to deal with this problem and was already deployed [16]. In this paper, we show how the SCION network can be combined with concepts of the *Economic Software-Defined Exchange* (ESDX) [6], [13] to enable online transactions establishing service contracts between providers.

II. Overview

The SCION architecture offers several advantages, including path transparency, support for multipath transmission, and a solid security framework. However, it was designed to support the kind of inter-domain routing relationships that exist today. Our project is exploring ways to enable more dynamic and flexible inter-domain routing. To that end, we present a mechanism that allows ASes to "offer" transit services to peers for a limited time and for a limited price. The approach leverages SCION's secure routing system to ensure that (i) advertisements cannot be spoofed; (ii) only legitimate (i.e., paid) traffic can make use of the specific service; ASes continue to enjoy the other benefits provided by SCION, such as path-awareness and the ability to use multiple paths to the same destination.

A. SCION architecture

In SCION, ASes are grouped into isolation domains (ISDs), each administered by a few distinguished *core ASes*; ISDs may also contain some non-core ASes. Each ISD independently defines its roots of trust, and the routing process within an ISD is isolated from external influences. Each end-to-end path consists of up to three path segments: up-path (from a non-core to a core AS in the same ISD), core path (between core ASes in different ISDs), and down-path (from a core to a non-core AS ISD) segments.[1] Inter-domain routing in SCION proceeds through a process called beaconing, in which core ASes originate path-segment construction beacons (PCBs), which are forwarded as a policy-constrained flood to other ASes within the ISD and among core ASes, to explore intra-ISD paths and core paths, respectively. PCBs accumulate cryptographically protected AS-level path information as they traverse the network, including protected forwarding information, encoded in the form of hop fields (HFs). End-hosts obtain policy-compliant path segments from *path servers* within their ISD, and embed the corresponding sequence of segments in each packet header to create end-to-end forwarding paths for data packets.

B. Motivating Example

To illustrate how our system might be used, Figure 1 presents a simple example involving four ASes (which are assumed to be in the same isolation domain, and thus share a common trusted root configuration and have each others' public keys). Circles in the figure represent ASes, while hexagons represent IXPs. AS S normally routes traffic destined for D via AS 1. The traffic along this route normally runs just under the threshold of S's service level agreement (SLA) with AS 1 that would increase its costs significantly. (Assume each AS's connection to IXPs X and Y is amply provisioned and will not be a bottleneck.) One of S's subscribers has a periodic job

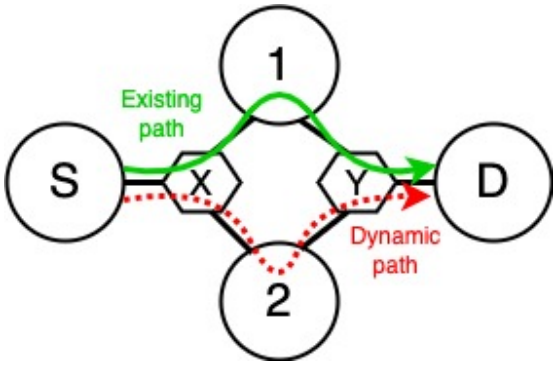

Fig. 1. AS interconnection for motivating example.

that requires moving a substantial amount of data to AS D, which would put AS S's traffic over the threshold if sent via AS 1. Meanwhile, AS 2 has unused capacity that it wishes to monetize. So it advertises transit from its port on IXP X to its port on IXP Y. S accepts the offer, and begins to forward the data along the path S-X-2-Y-D, while still sending its regular traffic along the normal path S-X-1-Y-D. It composes the SCION header using the hop fields in the beacon it receives from AS 2, plus the information given in the contract obtained from buying the offer in the market.

[1]Paths between hosts in the same ISD omit the core path.

C. Key System Components

A system to support dynamic transit agreements like the one above requires several components:

- A protocol for making offers and completing transactions.
- A trusted entity for running the *marketplace*, i.e., connecting ASes making offers and those wanting to purchase transit service. In this work, we propose the IXP as a natural (presumably neutral) intermediary.
- A common unit of service in which offers and contracts are denominated. This is needed in order to assemble end-to-end services efficiently, and needs to be developed based on common operational needs in deployment. Our prototype uses 10-minute segments of 1 Mbps capacity, plus additional attributes that indicate the level of service (e.g., loss rate).
- A dispute resolution system. This consists of both a technical piece, which provides credible and objective evidence of actual performance with minimal overhead, and a forum for adjudication of complaints—e.g., a legal forum. We focus here on the former (see Section IV), and observe that one or more entities operated by a consortia of ASes would be good candidates to implement the adjudication as well as the technical parts.

For more details about the SCION architecture, please refer to the SCION book [10].

III. Transaction Protocol

In order to standardize and streamline the process of selling and buying bandwidth from the spot market, we propose an initial protocol that allows ASes to interact by means of remote procedure calls (RPCs) to reach agreement on quantity, price and the parameters for the SCION configuration that will then implement the data plane connection. The protocol has two components: the marketplace proper, which consists of frequently used calls (implemented using gRPC [23]) involving selling and buying bandwidth; and the configuration plane, dealing with establishing the public keys of the ASes. The latter is currently implemented manually, ensuring that the certificate introduced for an AS includes the correct public key for that AS (according to the ASes' ISD). The marketplace RPC server is managed by a trusted broker, which can be located at the IXP where all ASes connect, although it need not be. (Further details of messages are given in Section V-A.)

A. Marketplace Plane

The prototype does not currently describe any RPCs related to actual clearing, i.e., money escrow/transfer. We believe this can be decoupled from the marketplace itself to permit flexibility; the main requirement is that the identifiers used for clearing can be used in the transactions below.

In the marketplace the ASes have three possible actions:

- Sell bandwidth: an AS creates a bandwidth offer.
- List offers: any AS can list the available bandwidth offers.
- Buy bandwidth: an AS buys (a part of) an available offer.

1) Sell Bandwidth: An AS can sell bandwidth from one of its interfaces to another (transit), or to its internal network (access). The AS can only sell *standard bandwidth units*, defined as 1Mbps for 600 seconds. These units can be stacked as if they were containers on a ship (instead they occupy a slot in time), and each 600 seconds there is a stack of N units. For instance, an AS may want to sell 1Gbps during twenty minutes, then 500Mbps for ten minutes, then 1Gpbs for ten more minutes. It would specify the *bandwidth profile* to be $\{1000, 1000, 500, 1000\}$. The aggregation of units is important, because this way sellers let buyers atomically specify which portions of the offer they want to buy in a single purchase operation. The offer also includes the *time window* during which the capacity will be available.

Once the broker receives the offer to sell, it checks the authenticity and stores it as an *original offer*. It then proceeds to create an exact replica of the offer, but signed with the broker's own private key instead of the seller's. The broker then publishes this offer as an *available offer* and returns that offer's ID to the seller. Swapping the signature allows the broker to republish an offer when another AS buys only a portion of the available offer; this way no further interaction with the seller is necessary. Since the original offer (signed by the seller) is stored, it can be proven that the broker created the available offer from the original one.

2) List Offers: This simply returns all available offers at the broker as a sequence of *offer specification* messages plus their IDs. We define an *available offer* as one that contains a non-zero *bandwidth profile*, is signed by the broker, is derived either from an *original offer* or from another available offer and a contract, and has not passed its expiration time.

3) Buy Offer: An AS can buy the whole or a part of an *available offer*. After listing the available offers and identifying one that meets its needs, an AS may request to purchase all or part of that offer by specifying its identifier. Once the broker receives the *purchase order* message, it checks the authenticity and correctness of the message. If the offer is no longer available, the buyer is notified with an error. Otherwise, the parameters of the purchase request are checked against the offer, and the buyer is notified if there is a mismatch; this should seldom happen (e.g., requested bandwidth not available would imply a mistake on the buyer's side). If the request is correct, the broker atomically creates a new *available offer* and a *contract*. The new available offer is created only if a non-zero bandwidth profile remains after subtracting the amounts purchased, and the contract is returned to the buyer. The seller is notified that a new contract has been created.

This design is simple, and has an increasing potential for contention and overhead to re-issue available offers as the number of concurrent buyers increases. (We evaluate this overhead and provide the results in the evaluation section.) An alternative could involve the broker collecting multiple purchase orders and fulfilling them as a group; it could even collect purchase orders from several buyers and after certain time perform the actual purchases. This approach allows for a more efficient partitioning of the offer, and may serve better the interests of the seller, but has worse trust implications for the buyers, as the temporal ordering of the purchase orders is opaque to everybody but the broker. We also provide an example of a mediation in the evaluation section.

Clearly the broker will need to use a mechanism that maximizes utility for all parties, since ASes will keep using the broker only if they find it useful, thus incentivizing the search for better mechanisms.

B. Configuring SCION

After selling or buying a part of an offer, the SCION infrastructure needs to know that the topology has changed. The event happens synchronously for the buyer—immediately after getting a new contract they can reconfigure their infrastructure, paying attention to the starting times—and asynchronously for the seller. In this latter case, the broker notifies the seller AS with the contract message using an already open channel created when the seller submitted their offer.

Because a new inter-AS link is being created dynamically, two events must happen for each of the two ASes involved:

1) The AS adds the details of the link to its topology.
2) The AS reloads the configuration of the affected components.

Both items are feasible according to SCION and the reference implementation [21]. Adding a new link to the topology is usually a manual process, but there is nothing preventing an automated configuration if the contract contains enough information regarding the remote part of the link, and if the local part can be pre-configured or derived from the existing configuration. The topology is only modified by adding a new interface to a particular border router, which contains all interfaces configured from the marketplace; other border routers of the AS are left unmodified. We label this router the *"ESDX border router"*, to indicate its role in the marketplace. If there is no such router in the topology, one is added or designated; thus, the AS always has one or zero ESDX border routers in the topology. Once the configuration process has finished, the AS reloads only the ESDX border router, since no other component could have been affected by the transaction. Finally, when the contract expires, the interface is removed from the ESDX border router, and this router is reloaded—or removed if empty. Both interface adding and removing operations are atomic, as there could be a number of contracts that start or expire at the same time.

In the case of the seller, the SCION beaconing process can start just moments after the ESDX border router has the new interface, following the seller's internal policy as usual. This can be done with enough time in advance (seconds) to allow the propagation of the novel PCBs to reach the client with enough anticipation for the client to start using the link exactly when the contract starts.

Another configuration possibility for the seller implies configuring the ESDX border router early, and propagating

beacons before the contract starts. The problem with this approach is that the definition of the hop fields in the SCION forwarding protocol do not include a starting time field, so the buyer AS could use the dataplane on the link earlier than it should.

Even without modifying the SCION hop field in the forwarding protocol, and from the buyer AS's perspective, the PCBs that the seller propagates to the buyers using the new link have a high novelty score, thus generally preferred and selected in the path exploration process. In general these beacons reach all downstream ASes very early in the process, unless prevented by the buyer's policy.

IV. Verifying Contract Fulfillment

Dispute resolution over the correct fulfillment of contracts requires two key pieces: (1) a set of trusted monitors observing the packet stream and (2) a forum for adjudication of complaints that accepts evidence captured by the monitors.

The monitoring component has to be able to collect evidence of the behavior or misbehavior of buyers and sellers. Concerning buyers, the monitors have to record the inbound and outbound traffic volume from and to the buyer AS over the SCION link negotiated by the contract. In case the seller accuses the purchaser of overusing the provided link, the bandwidth trace can be compared to the bandwidth profile in the contract. For bandwidth sellers, the monitors make sure that the forwarding bandwidth between the IXP interface of the seller and the upstream interfaces named in the contract is actually delivered.

We can achieve both goals by placing bandwidth monitors on the IXP-link between the seller and buyer ASes and between the seller AS and all upstream providers that are offered as reachable paths. We refer to the monitor between seller and buyer as downstream monitor and the monitors between seller and upstream providers as upstream monitors. Monitors are implemented as transparent "bump in the wire" systems that passively examine traffic exchanged between the border routers.

Since monitors must be trusted by both seller and buyer, we propose that the downstream monitor be under the administrative control of the already trusted IXP. Upstream monitors should be provided by a trusted 3rd party, for example another IXP that connects the seller AS to its upstream provider or another organization that provides a neutral trusted service similar to CAs in the current Internet.

The data path of our prototype monitors is implemented as an eBPF XDP program with a control plane in Go [1]. There is no explicit communication between the marketplace and the monitors, since they can learn of new contracts by observing the interface IDs in the SCION packet header. Monitors log a byte count for every unique egress/ingress interface pair between buyer and seller. Buyers and seller can request the log files pertaining a certain set of interface pairs by presenting the corresponding contract to the monitors. In case of a dispute, the log files can be used as evidence in the appropriate forum.

V. Prototype and Evaluation

A simple initial prototype of the marketplace and SCION configuration service has been built using Django and gRPC [1]. The prototype implements the marketplace RPCs described in Section III, as well as some scripts to configure ASes' certificates, which must be run manually. Marketplace messages sent by ASes are authenticated by the broker using the ASes' certificates. The broker's certificate is available to any AS via the same configuration mechanism used to configure its own certificates; thus, ASes can validate the broker's signature on re-issued offers and contracts. The purchase process follows what is described in Section III; only one purchase order is satisfied at a time; any additional concurrent purchase orders (e.g., from other ASes) are notified their offers were not accepted, and the broker re-issues the sale offer (provided the bandwidth profile is non-empty). Thus it is up to the ASes that did not succeed to re-list the available offers and try again; as we shall see, this leads to significant overhead when many ASes are interested in the same sale offer.

The SCION configuration process assumes a running and valid topology for the reference implementation, and although it could be triggered by most task managers for Django (`Celery`, `cron`, etc.), we have opted to simply trigger it manually for our tests.

A. Messages

The messages are specified using gRPC, which easily allows inter-process communication in different languages. For our prototype, we have implemented the spot market using Django-gRPC, and spot-market clients using python and Go.

1) Sell Bandwidth: When an AS wants to sell part of its bandwidth, it creates an *offer_specification* message that contains the following fields:

`ia_id`: the ISD-AS identifier of the seller.

`not_before`, `not_after`: starting time and ending time of the bandwidth offer. $\texttt{not\_after} - \texttt{not\_before} = n \times 600$ (seconds), with $n \in \mathbb{N}, n \geq 1$.

`reachable_paths`: a `\n` separated list of paths that the seller offers (expressed as policies according to [21]).

`qos_class`: the type of service assurance. This allows to specify the level of commitment to detect the fulfillment of the contracts made on this offer. We currently define classes "0" (no QoS) and "1" (monitored bandwidth).

`monitor_provider`: if the `qos_class` is "1", this field specifies the provider doing the monitoring.

`price_per_unit`: the price in $ per standard bandwidth unit.

`bw_profile`: the bandwidth profile. To be valid it must contain $\frac{\texttt{not\_after} - \texttt{not\_before}}{600}$ slots.

`br_address_template`: how the seller's border router address looks like. E.g., `10.1.1.4:50000-51000` means that the seller's border router address in the contract will be `10.1.1.4:PORT`, with $50000 \leq \texttt{PORT} \leq 51000$.

`br_mtu`: the MTU the seller specifies for this link.

`br_link_to`: with possible values `CORE`, `PARENT`, or `PEER`. It specifies the type of link it is being offered.

`signature`: the signature of the seller on all previous fields.

The seller AS receives the ID of the registered offer after submitting the offer specification.

2) Buy Bandwidth: The *purchase_order* message contains the following fields:

`offer_id`: the ID of the offer from which to buy (obtained from the broker's list of available offers).

`offer`: the *offer_specification* message.

`buyer_ia_id`: the ISD-AS identifier of the buyer.

`bw_profile`: a bandwidth profile describing the portion that the buyer intends to buy.

`starting_on`: the time at which the above bandwidth profile should start. Note that this time is always $\geq$ `not_before` in the *available offer*. It allows sparing many `0`... in the initial part of the `bw_profile` of this *purchase_order*.

`signature`: the signature of the buyer on all previous fields, except `offer_id`.

The *contract* message that is obtained from the broker after a successful purchase contains the following fields:

`contract_id`: the ID of this contract.

`offer`: a full copy of the *offer_specification* message.

`purchase_order`: a full copy of the *purchase_order* message.

`contract_timestamp`: the time when the broker signed the contract. Should be the same moment when the contract was created, just after the purchase order validation.

`br_address`: the address that the seller's border router will make available for this link. It is covered by the `br_address_template` in the offer, and it has the form `IP:PORT` or `[IP]:PORT`.

`contract_signature` the signature of the broker on all previous fields, except the `contract_id`.

B. Evaluation

The evaluation involves a number of ASes whose keys and certificates have been configured statically, each of which has a predefined role as seller or buyer. All the offers present in the marketplace are suitable for any of the buyers' preferences, and as long as there is any available bandwidth left in an offer's profile, buyers will try to buy it. This scenario intends to mimic a simplified typical IXP setup with the addition that the offers are useful to any client.

For this simple first-come, first-served prototype the time needed to purchase all the offered capacity in a single offer grows quadratically with the number of concurrent ASes purchasing the offer. This is because each time a contract is issued, the broker has to inform every AS that did *not* succeed of their failure, and each of those ASes must issue another offer to buy. However, the total time for 100 ASes to purchase a part of the same original offer is still less than 20 seconds, which makes even this simple prototype suitable for, e.g., medium size IXPs (*same unique offer* in Figure 2).

More realistic is the *normally dist. offer* case, where the clients purchases are distributed normally among the existing

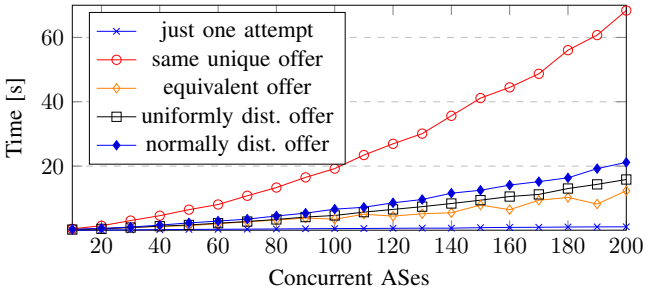

Fig. 2. Time to buy bandwidth. The *just one attempt* case depicts one AS purchasing the offer and $N-1$ getting a failure notification.

offers. In this situation, less than 7 seconds are required to resolve all the purchases. The uniform distribution of the offers case (*uniformly dist. offer*) is also shown for comparison.

Finally, and as an example of a simple broker mediation, we show the time needed to complete the purchase the clients specify, or an equivalent one (*equivalent offer* case). The equivalent offer is defined in this broker to be an offer that is available, derived from the requested offer (all fields except `bw_profile` identical), and has sufficient bandwidth.

With these results, we can assert that the prototype performs sufficiently well to allow the clients buy bandwidth only some minutes before they actually need it, thus preventing a very long term prediction of the bandwidth requirements that would not be realistic.

VI. Related Work

Many path aware architectures have been proposed, going back to Nimrod [7], Pathlets [12], Platypus [19], NEBULA [3], NIRA [26], and SCION [4], [27]. Significant progress has been made to enable the deployment of these architectures in order to realize their benefits, including rapid fail-over, concurrent use of parallel paths, and QoS enabled networks. SCION, in particular, is the first inter-domain multi-path PAN architecture used in practice [16]. Our thesis is that the PAN architecture combined with the trust structure of SCION can serve as a basis for more dynamic interconnection between ASes, and that economic incentives will play an important role in such an architecture, as proposed in the ChoiceNet project [24].

We are not the first to consider how to enable more dynamic peering among ASes. Route Bazaar [8] and Dynam-IX [18] are prominent previous efforts. Route Bazaar automates the initiation, establishment, and verification of end-to-end connectivity agreements by using a trusted decentralized public ledger. These agreements are for a fixed volume of traffic, sampled at routers to generate a forwarding proof for path conformance verification. All the advertisements, agreements, and forwarding and payment proofs are recorded on the public ledger. One challenge for such an approach is that, to provide resiliency without centralized trusted parties, decentralized public ledgers require that each added transaction be confirmed, which can add significant uncertainty and delay before a contract could be used. We argue that the foundational trust structure of SCION (i.e., ISDs) removes the need for a decentralized public ledger, so that immediately after establishing a contract the

new connectivity can be announced through SCION beaconing and used soon thereafter.

Dynam-IX enables ISPs to exploit the rich interconnection opportunities at IXPs to implement traffic engineering policies quickly. It also utilizes a distributed tamper-proof ledger to build trust. In Dynam-IX, peers interact directly with other ASes to offer and query interconnection opportunities rather than with a third party. This may offer advantages in terms of reducing trust in third parties. On the other hand, if incentives can be properly aligned, a third party may enable greater flexibility (e.g., by splitting an offer), negotiation of service levels, more efficient matching of offers and requests, auction-based markets, and dispute handling. IXPs, which are already trusted by peers to handle AS interconnections, and have been proposed as a neutral and trusted source of measurements [2], are promising candidates to take on this responsibility.

Previous studies investigated the important role of IXPs in the current Internet architecture [9], [15]. Enabling SDN at IXPs (SDXs) [14] even further increases the potential of IXPs to improve the inter-domain routing. A type of SDX called an Economic Software-Defined Exchange Point (ESDX) [13] has been proposed to serve as a trusted intermediary, enforcing interdomain policy at interconnection points.

VII. Summary and Future Work

We have proposed to combine market concepts [24] with the path aware and secure-by-design architecture of SCION [27] to enable dynamic, low cost, and granular establishment of service contracts between providers. Earlier work [25] has shown the potential for a "spot market" based on such mechanisms to benefit all participants, by enabling providers with temporary (e.g., diurnal) excess capacity to monetize it, while also giving other providers additional traffic engineering options (e.g., to shift predictable elastic loads).

In future work, we intend to investigate extending the service beyond a single AS hop, more sophisticated "match-making" services, study the viability of contract composition (purchases could depend on other contracts), QoS orchestration (creation of QoS connections and contracts between ASes, e.g., via COLIBRI [11]), and other ways to align incentives among the parties, e.g., in the measurement and validation system. Finally, we consider deploying a pilot program in some IXPs and institutions, in particular those that interconnect educational networks such as GEANT or certain NRENs.

References

[1] ESDX for SCION. https://github.com/juagargi/esdx-scion, 2022.

[2] Yousef Alowayed, Marco Canini, Pedro Marcos, Marco Chiesa, and Marinho Barcellos. Picking a partner: A fair blockchain based scoring protocol for autonomous systems. In *Proceedings of the Applied Networking Research Workshop*, pages 33–39, 2018.

[3] Tom Anderson, Ken Birman, Robert Broberg, Matthew Caesar, Douglas Comer, Chase Cotton, Michael J Freedman, Andreas Haeberlen, Zachary G Ives, Arvind Krishnamurthy, et al. The nebula future internet architecture. In *The Future Internet Assembly*, pages 16–26. Springer, 2013.

[4] David Barrera, Laurent Chuat, Adrian Perrig, Raphael M Reischuk, and Pawel Szalachowski. The scion internet architecture. *Communications of the ACM*, 60(6):56–65, 2017.

[5] Randy Bush and Rob Austein. The resource public key infrastructure (RPKI) to router protocol, RFC 6810. Technical report, 2013.

[6] Kenneth L Calvert, James Griffioen, Anna Nagurney, and Tilman Wolf. A vision for a spot market for interdomain connectivity. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1860–1867. IEEE, 2019.

[7] Isidro Castineyra, Noel Chiappa, and Martha Steenstrup. The nimrod routing architecture RFC 1992. Technical report, 1996.

[8] Ignacio Castro, Aurojit Panda, Barath Raghavan, Scott Shenker, and Sergey Gorinsky. Route bazaar: Automatic interdomain contract negotiation. In *15th Workshop on Hot Topics in Operating Systems (HotOS XV)*, 2015.

[9] Nikolaos Chatzis, Georgios Smaragdakis, Anja Feldmann, and Walter Willinger. There is more to ixps than meets the eye. *ACM SIGCOMM CCR*, 43(5):19–28, 2013.

[10] Laurent Chuat, Markus Legner, David A. Basin, David Hausheer, Samuel Hitz, Peter Müller, and Adrian Perrig. *The Complete Guide to SCION - From Design Principles to Formal Verification*. Information Security and Cryptography. Springer, 2022.

[11] Giacomo Giuliari, Dominik Roos, Marc Wyss, Juan Angel García-Pardo, Markus Legner, and Adrian Perrig. Colibri: a cooperative lightweight inter-domain bandwidth-reservation infrastructure. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 104–118, 2021.

[12] P Brighten Godfrey, Igor Ganichev, Scott Shenker, and Ion Stoica. Pathlet routing. *ACM SIGCOMM CCR*, 39(4):111–122, 2009.

[13] James Griffioen, Tilman Wolf, and Kenneth L Calvert. A coin-operated software-defined exchange. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE, 2016.

[14] Arpit Gupta, Laurent Vanbever, Muhammad Shahbaz, Sean P Donovan, Brandon Schlinker, Nick Feamster, Jennifer Rexford, Scott Shenker, Russ Clark, and Ethan Katz-Bassett. Sdx: A software defined internet exchange. *ACM SIGCOMM CCR*, 44(4):551–562, 2014.

[15] Vasileios Kotronis, Rowan Klöti, Matthias Rost, Panagiotis Georgopoulos, Bernhard Ager, Stefan Schmid, and Xenofontas Dimitropoulos. Stitching inter-domain paths over ixps. In *Proceedings of the Symposium on SDN Research*, pages 1–12, 2016.

[16] Cyrill Krähenbühl, Seyedali Tabaeiaghdaei, Christelle Gloor, Jonghoon Kwon, Adrian Perrig, David Hausheer, and Dominik Roos. Deployment and scalability of an inter-domain multi-path routing infrastructure. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 126–140, 2021.

[17] Matt Lepinski and Kotikalapudi Sriram. BGPsec protocol specification, RFC 8205. Technical report, 2017.

[18] Pedro Marcos, Marco Chiesa, Lucas Müller, Pradeeban Kathiravelu, Christoph Dietzel, Marco Canini, and Marinho Barcellos. Dynam-ix: a dynamic interconnection exchange. In *ACM CONEXT*, pages 228–240, 2018.

[19] Barath Raghavan and Alex C Snoeren. A system for authenticated policy-compliant routing. In *ACM SIGCOMM*, pages 167–178, 2004.

[20] Yakov Rekhter, Tony Li, and Susan Hares. A border gateway protocol 4 (BGP-4), RFC 4271. Technical report, 2006.

[21] SCION. The SCION reference implementation. https://github.com/scionproto/scion, 2022.

[22] Haya Shulman. How (not) to deploy cryptography on the internet. In *ACM Conference on Data and Application Security and Privacy*, 2022.

[23] The gRPC Authors and The Linux Foundation. gRPC: A high performance, open source universal RPC framework. https://grpc.io/, 2021.

[24] T. Wolf, J. Griffioen, K. Calvert, R. Dutta, G. Rouskas, I Baldin, and A. Nagurney. Choicenet: Toward an economy plane for the internet. *ACM SIGCOMM CCR*, 44(3):87–96, 2014.

[25] Hong Xu and Baochun Li. Spot transit: Cheaper internet transit for elastic traffic. *IEEE Transactions on Services Computing*, 8(5):768–781, 2014.

[26] Xiaowei Yang, David Clark, and Arthur W Berger. Nira: a new inter-domain routing architecture. *IEEE/ACM Transactions on Networking*, 15(4):775–788, 2007.

[27] Xin Zhang, Hsu-Chun Hsiao, Geoffrey Hasker, Haowen Chan, Adrian Perrig, and David G Andersen. Scion: Scalability, control, and isolation on next-generation networks. In *2011 IEEE Symposium on Security and Privacy*, pages 212–227. IEEE, 2011.