Internet Architecture

ECE/CS598HPN

Radhika Mittal
What is Internet Architecture?

• How an endhost identifies and specifies the destination.
• How routers understand that specification to forward packets to the destination over the Internet.

• Carried out by L3 (IP).
IP as the narrow waist

- Facilitated a lot of innovation above and below IP.
- *Hard to change IP itself.*
Security

- Clean-slate architecture.
- Establishes trust domains.
- Guarantee control plane isolation for trust domains.

SCION: Scalability, Control, and Isolation On Next-Generation Networks
Xin Zhang, Hsu-Chun Hsiao, Geoffrey Hasker, Haowen Chan, Adrian Perrig and David G. Andersen
CyLab / Carnegie Mellon University

Abstract—We present the first Internet architecture designed to provide route control, failure isolation, and explicit trust information for end-to-end communications. SCION separates ASes into groups of independent routing sub-planes, called trust domains, which then interconnect to form complete routes. Trust domains provide natural isolation of routing failures and human misconfiguration, give endpoints strong control for both inbound and outbound traffic, provide meaningful and enforceable trust, and enable scalable routing updates with high path freshness. As a result, our architecture provides strong resilience and security properties as an intrinsic consequence of good design principles, avoiding piecemeal add-on protocols as security patches. Meanwhile, SCION only assumes that a few top-tier ISPs in the trust domain are trusted for providing reliable end-to-end communications, thus achieving a small Trusted Computing Base. Both our security analysis and evaluation results show that SCION naturally prevents numerous attacks and provides a high level of resilience, scalability, control, and isolation.

I. INTRODUCTION

The Internet is the most geographically, administratively, and socially diverse distributed system ever invented. While today’s Internet architecture admits some administrative diversity, such as by separating routing inside a domain (intra-AS routing) from global inter-domain routing, it falls short in handling the key challenges of security and isolation that arise in this intensely heterogeneous setting. As a result, we see surprisingly frequent incidents in which communication is interrupted by actions or actors far from the communicating entities. In addition to classical examples such as YouTube being generally disrupted by routing announcements from Pakistan [1], other issues surrounding the lack of resource control and isolation are not solved by existing proposals such as BGP [2]: the introduction of excessive routing churn [3]; traffic flooding; and even issues of global conflicts over naming and name resolution.

This paper proposes a clean-slate Internet architecture, SCION, that provides strong guarantees for failure isolation and route control in ways that map well to existing geographic, political, and legal boundaries. We show that strong control and isolation naturally leads to security and reliability without the use of high-overhead security mechanisms, while exposing to the endpoints diverse communication path sets that can support a wide spectrum of routing policies and path preferences (path expressiveness).

We introduce the notion of a hierarchy of trust domains whose members all share a common contractual, legal, cultural, geographical, or other basis for extending limited trust among each other. Examples may be a domain of U.S. commercial and educational institutions, ISPs that participate in the same peering point who share a common, binding legal contract on their behavior, or ISPs in the same state or country who are subject to the same laws and regulations. Using this abstraction, we provide the machinery to guarantee control-plane isolation: Entities outside a trust domain cannot affect control-plane computation and communication within that trust domain. For communication that must span trust domains, we provide the property that the entities who can affect the communication are limited to a necessary and explicitly identified set of other trust domain. We leave data-plane security as future work and thus do not consider denial of service attacks. In addition, the introduction of trust domains enables sources, transit ISPs, and destinations in SCION to agree jointly on which path to use. The architecture naturally controls routing information flow, and provides for explicit trust in path selection.

Through isolation and control, SCION enables expressive trust, i.e., all the communicating endpoints can decide and control explicitly and precisely whom they need to trust for providing reliable communications. Exposing such explicit trust information for end-to-end communication can eventually benefit network availability, because the endpoints can select more "trusted" communication paths with presumably more reliable data delivery; or at least, SCION holds the parties involved in the communications accountable for their misbehavior and failures.

Contributions. We design and analyze SCION, an Internet architecture emphasizing the principles of control, isolation and explicit trust. SCION enables route control for ISPs, senders and receivers at an appropriate level of granularity, balancing efficiency, expressiveness, policy compliance, and security. The isolation properties dramatically shrink the TCB and make explicit which entities communication relies upon.

SCION offers strong security properties and demonstrates that the resulting routes widely mirror those in place under BGP today. We anticipate that the proposed architecture offers a useful design point for a next-generation Internet.

This research was supported by CyLab at Carnegie Mellon under grants DA-2018-02-1-0198, W911NF-08-1-0277 and W911NF-07-0287 from the Army Research Office, and by NSF under awards CNS-1040801, CNS-050224, AN1031343, and CNS-0220817. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARQ, CMU, CyLab, or the U.S. Government or any of its agencies.
Security

• Architecture to limit DoS attacks.

• Receivers grant sending capabilities to senders.

• Routers check for this capability to determine if the packet is wanted.

TVA: A DoS-limiting Network Architecture
Xiaowei Yang, Member, David Wetherall, Member, Thomas Anderson, Member

Abstract—We motivate the capability approach to network denial-of-service (DoS) attacks, and evaluate the TVA architecture which builds on capabilities. With our approach, rather than send packets to any destination at any time, senders must first obtain “permission to send” from the receiver, which provides the permission in the form of capabilities to those senders whose traffic it agrees to accept. The senders then include these capabilities in packets. This enables verification points distributed around the network to check that traffic has been authorized by the receiver and the path is between, and hence to clearly discard unauthoized traffic. To evaluate this approach, and to understand the detailed operation of capabilities, we developed a network architecture called TVA. TVA addresses a wide range of possible attacks against communication between pairs of hosts, including spoofed packet floods, network and host buffer overflows, and router state exhaustion. We use simulations to show the effectiveness of TVA at limiting DoS floods, and an Implementation on a Click router to evaluate the computational costs of TVA. We also discuss how to incrementally deploy TVA into practice.

I. INTRODUCTION

The Internet owes much of its historic success and growth to its openness to new applications. A key design feature of the Internet is that any application can send anything to anyone at any time, without needing to obtain advance permission from network administrators. New applications can be designed, implemented, and come into widespread use much more quickly, if they do not need to wait for key features to be added to the underlying network. Quietly, however, the Internet has become much less open to new applications over the past few years. Pervasively, this has happened as a rational response of network and system administrators: in order to cope with the consequences of the Internet’s openness. The Internet architecture is vulnerable to denial-of-service (DoS) attacks, where any collection of hosts with enough bandwidth (e.g., using machines taken over by a virus attack) can disrupt legitimate communication between any pair of other parties, simply by flooding one end or the other with unwanted traffic. These attacks are widespread, increasing, and have proven resistant to all attempts to stop them [26].

Operationally, to deal with persistent and repeated DoS and virus attacks, network and system administrators have begun to deploy automated response systems to look for anomalous behavior that might be an attack. When alarms are triggered, often by legitimate traffic, the operational response is typically to “stop everything and ask questions later.” Unfortunately, any new application is likely to appear to be anomalous! Our experience with this comes from operating and using the Planetlab testbed, which is designed to make it easy to develop, geo-spatially distributed, Internet applications [27]. On several occasions, we have observed innocuous, low-rate traffic from a single application trigger alarms that completely disconnected entire universities from the Internet. Since alarm rules are by nature secret, the only way to guarantee that a new application does not trigger an alarm (and the resulting disproportionate response) is to make its traffic look identical to some existing application. In other words, the only safe thing to do is to precisely mimic an old protocol.

The openness of the Internet is likely to erode if there is no effective solution to eliminate large scale DoS attacks. Attackers are winning the arms race with anomaly detection by making their traffic look increasingly like normal traffic. The CodeRed and follow-on viruses have demonstrated repeatedly that it is possible to recruit millions of machines to the task of sending normal HTTP requests to a single destination [24], [25]. This problem is fundamental to the Internet architecture: no matter how over-provisioned you are, if everyone in the world sends you a single packet, legitimate traffic will not get through. We argue for taking a step back, to ask how, at an architectural level, we can address the DoS problem in its entirety while still allowing new applications to be deployed. Our goal, in essence, is to let any two nodes exchange whatever traffic they like (subject to bandwidth constraints of intermediate links), such that no set of third parties can disrupt that traffic exchange.

Our approach is based on the notion of capabilities, which are short-term authorizations that senders obtain from receivers and stamp on their packets. This allows senders to control the traffic that they receive. Our attraction to capabilities is that they cut to the heart of the DoS problem by allowing unwanted traffic to be removed in the network, but do so in an open manner by providing destinations with the control over which traffic is filtered. However, while capabilities may be an appealing approach, they leave many questions unanswered, such as how capabilities are granted without being vulnerable to attack.

To answer these questions and help evaluate the capability approach, we have designed and prototyped the Traffic Vali-
dication Architecture (‘TVA’). TVA is a DoS-limiting network architecture that details the operation of capabilities and combines mechanisms that ensure a broad set of possible denial-of-service attacks, including those that flood the setup channel, that exhaust router state, that consume network bandwidth, and so forth. The TVA system that we present in this paper is a revision of our earlier work [35] that pays greater attention to protecting the capability request channel.

We have designed TVA to be practical in three key respects. First, we bound both the computation and state needed to process capabilities. Second, we have designed our system to be incrementally deployable in the current Internet. This can be done by placing inline packet processing boxes at trust boundaries and points of congestion, and upgrading collections of hosts to take advantage of them. No changes to Internet

1The same TVA is inspired by the Tennessee Valley Authority, which operates a large-scale network of dams to control flood damage, saving more than $200 million annually.
Accountability

• Make Internet addressing more accountable.

• Use self-certifying host addresses.
Information-centric Networking

- Name *bits* (data or content) instead of locations.

- Self-certified or signed in some manner.
Replace IP addresses with location-independent, flat, globally unique IDs.

Replace IP addresses (and ports) with service names.
Pathlet Routing

P. Brighten Godfrey¹, Igor Ganeriwala², Scott Shenker¹, and Ion Stoica¹
¹University of Illinois at Urbana-Champaign ²UC Berkeley ¹ISCBI
pgb@illinois.edu, (igor.shenker,ion.stoica)@cs.berkeley.edu

ABSTRACT
We present a new routing protocol, pathlet routing, in which networks advertise fragments of paths, called pathlets, that source concentrate into end-to-end source routes. Intuitively, the pathlet is a highly flexible building block, capturing policy constraints as well as enabling an exponentially large number of path choices. In particular, we show that pathlet routing can emulate the policies of BGP routing, and several recent multipath proposals.

This similarity lets us address two major challenges for Internet routing: scalability and source-controlled routing. When a router’s routing policy has only "local" constraints, it can be represented using a small number of pathlets, leading to very small forwarding tables and many choices of routes for senders. Crucially, pathlet routing does not impose a global requirement on what style of policy is used, but rather allows multiple styles to coexist. The protocol thus supports complex routing policies without enabling or incentivizing the adoption of policies that yield small forwarding plane state and a high degree of path choice.

Categories and Subject Descriptors
C.2 [Network Architecture and Design]: Packet-switching networks
C.2.2 [Network Protocols]: Routing Protocols
C.2.8 [Internetworking]: Routers

General Terms
Design, Experimentation, Performance, Reliability

1. INTRODUCTION

Challenges for Interdomain routing. Interdomain routing faces several fundamental challenges. One is scalability: routing protocols in the Internet need to scale to the enormous number of hosts and networks. Second, path selection is complicated by the fact that many hosts and networks have complex routing policies.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

1 The first and fourth authors were supported in part by a Case Collaborative Research Initiative grant.

that scales linearly in the number of IP prefix advertised in the Internet. This is particularly a concern in the data plane where the router stores the routing table, or forwarding information base (FIB). Because it has to operate at high speed and is often seen DRAM rather than commodity DRAM, FIB memory is arguably more constrained and expensive than other resources in a router [22]. Moreover, the number of IP prefixes is increasing at an increasing rate [15], leading to the need for expensive hardware and upgrades. The Internet Architecture Board Workshop on Routing and Addressing recently identified FIB growth as one of the key concerns for future scalability of the routing system [22].

A second challenge for interdomain routing is to provide multipath routing, in which a packet’s source (an end host or edge router) selects its path from among multiple options. For network users, multipath routing is a solution to two important deficiencies of BGP: poor reliability [1, 14, 17] and suboptimal path quality, in terms of metrics such as latency, throughput, or path rate [1, 27]. Sources can observe end-to-end failure and path quality and their effect on the particular application in use. If multiple paths are exposed, the end-hosts could use these observations by switching traffic much more quickly and in a more informed way than BGP’s control plane, which takes minutes or tens of minutes to converge [19, 21]. For example, for network users, multipath routing represents a new service that can be offered. In fact, route control policies today are so complex that selecting paths based on availability, performance, and cost for multi-armed edge network users [3]; exposing more flexibility in route selection could be an opportunity. Greater choice in routes may bring other benefits as well, such as enabling competition and encouraging "coalitions" between different parties to be resolved within the protocol [8].

But providing multiple paths while respecting network users’ policies is nontrivial. BGP provides no multipath service; it selects a single path for each destination, which it installs in its FIB and advertises to its neighbors. Several multipath proposals have been proposed, but these have drawbacks such as not supporting all of BGP’s routing policies [32, 35], imposing at least a limited number of additional paths [28], making it difficult to know which paths will be followed as BGP converges to the same FIB [21, 23], which would exacerbate the scalability challenge.

Our contributions. This paper addresses the challenges of scalability and multipath routing with a novel protocol called pathlet routing. In pathlet routing, each network advertises pathlets—fragments of paths represented as sequences of virtual nodes (vnodes) along which the network service, and driven by the price to a small fraction of its pre-competition starting point. For the consumer, especially the residential broadband consumer, there is likely to be a very small number of competitive local ISPs offering service [8, 12, 15]. With cable competing only with DSL, the market is a duopoly at best (the facilities level) and often a monopoly in practice. In some areas, it cannot choose their backbone ISPs separately from their local ISPs; local providers can then control the selection of backbone providers and capture the market power of consumers. This will reduce the competition pressures on the backbone providers and lead to a vertically integrated ISP market. The recent merger of SBC and AT&T [5] to (use the old names) [5] and Verizon and MCI [63] only add to this concern. In the worst case, SBC sends all of its traffic to the AT&T backbone, and Verizon sends its traffic to the MCI. We base the emergence of market power in the backbone market.

Conversely, when users can control the sequence of providers their packets take, the power of user choice fosters competition. In a competitive market, ISPs that are more efficient attract more users, which creates a positive loop for them further advance their networks and to improve efficiency. In the long term, competition disciplines the market, drives innovation, and lowers the costs to provide services [12, 46].

Moreover, recent studies on overlay networks and multi-pathing show that letting the user choose routes also brings technical benefits. The default routing path chosen by BGP [48] may not be the best in terms of performance, reliability, or cost. End users on an overlay network can often find better routes themselves by BGP [37, 51]. For instance, Densor found that for almost 80% of the default paths, an alternative route offers a lower loss rate [51]. Similarly, recent studies also show that multipathing can improve path quality and reduce monetary cost by intelligently choosing their upstream providers [21, 22].

The prevalence of these alternative paths suggests that giving the users the ability to choose routes can lead to appreciable performance, reliability, or user satisfaction. Only users know whether a path works for their applications or not. A user may choose a path that has a high throughput, a low latency, or a low cost for playing online games, even if the path may cost more. In contrast, a user may prefer a low cost path for peer-to-peer file downloads. Furthermore, letting the user choose routes also brings reliability. A user can multipath in case of failures to improve the reliability for mission-critical applications, such as 911 calls, or quickly switch to an alternative route if the default routing path fails.
Lots of proposals
How do we enable innovation in Internet Architecture?
Trotsky: Enabling a Permanent Revolution in Internet Architecture

James McCauley, Yotam Harchol, Aurojit Panda, Barath Raghavan, Scott Shenker

SIGCOMM’19

Some slide contents borrowed from McCauley’s SIGCOMM’19 talk.
How do we enable innovation in Internet Architecture?

• Remove the narrow waist!

• How?

• Two steps
Step 1: Fix Layering

• We are missing a layer!

• Internet is not a composition of L2 networks.

• It is a composition of *domains*. 
Step 1: Fix Layering

- We are missing a layer!
- Internet is not a composition of L2 networks.
- It is a composition of domains.
Step 1: Fix Layering

• Decouple how data is delivered:
  • within a domain (L3)
  • across domains (L3.5)

• Decouple how two domains deliver data internally.
How do we enable innovation in Internet Architecture?

• Remove the narrow waist!

• How?

• Two steps:
  • Layer 3.5: decouple intra-domain and inter-domain data delivery.
Step 2: Embrace multiple architectures

• Support multiple L3.5 protocols.
  • Up to the domain to choose which ones it wants to support.

• Trotsky Processors (TPs) deployed at domain edge (in software) responsible for implementing supported L3.5 protocols.
How do we enable innovation in Internet Architecture?

• Remove the narrow waist!

• How?

• Two steps:
  • Layer 3.5: decouple intra-domain and inter-domain data delivery.
  • Embrace multiple L3.5 protocols instead of upgrading to a single one.
Inside a domain

Domain B
IP+NDN

Both NDN and IP packets

Domain A
IP+NDN

Domain C
Legacy IP-Only
Inside a domain

Domain B
IP+NDN

Domain A
IP+NDN

Domain C
Legacy IP-Only

Global layer router which implements IP and NDN
Inside a domain

Domain B
IP+NDN

Domain A
IP+NDN

Domain C
Legacy IP-Only

Global layer router which implements IP and NDN

Pipe layer router implements ... ?
Inside a domain

Domain B
IP+NDN

Global layer router which implements IP and NDN

Domain A
IP+NDN

Pipe layer router implements another IP at the pipe layer

Domain C
Legacy IP-Only
Host initialization

1. Host arrives
Host initialization

1. Host arrives

What protocols does this domain speak?!
Host initialization

1. Host arrives

2. Bootstrap to learn which pipe and global protocols supported
Host initialization

1. Host arrives
2. Bootstrap to learn which pipe and global protocols supported
3. Configure pipe layer (e.g. DHCP for pipe layer IP address)
Host initialization

1. Host arrives
2. Bootstrap to learn which pipe and global protocols supported
3. Configure pipe layer (e.g., DHCP for pipe layer IP address)
4. Configure global layer (e.g., DHCP for global layer IP address)
Host initialization

1. Host arrives
2. Bootstrap to learn which pipe and global protocols supported
3. Configure pipe layer (e.g., DHCP for pipe layer IP address)
4. Configure global layer (e.g., DHCP for global layer IP address)
5. Initialization complete
Web Download

Downloading web page with *http:* URL

- App (HTTP)
- Transport (TCP)
- Global (IP)
- Pipe (IP)
- Link (Eth)

To Neighboring Domain
Web Download

Downloading resources with *ndnchunks*: URL

<table>
<thead>
<tr>
<th>Transport (NDN chunks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global (NDN)</td>
</tr>
<tr>
<td>Pipe (IP)</td>
</tr>
<tr>
<td>Link (Eth)</td>
</tr>
</tbody>
</table>

To Neighboring Domain
Partial Deployment of L3.5 Designs

Domain A --- Domain B ---- Domain C
Key Contribution of Trotsky

• A framework that allows incremental side-by-side deployment of new architectures.

• And is itself incrementally deployable.
Summary

• Goal: enable extensibility in Internet architecture.

• Problem: the universal narrow waist.

• Solution: remove it!
  • Decouple intra-domain and inter-domain data planes.
  • Embrace co-existence of multiple inter-domain protocols.

• Result:
  • An incrementally deployable design.
  • ..which can incrementally deploy new architectures.
Discussion

• Is the universal narrow waist truly removed?

• What are the limitations of Trotsky design?
Your opinions

• Pros
  • Backwards-compatible and incrementally deployable.
  • Framework providing only a minimal set of functionality.
  • No need to change all routers.
  • Opens up avenue for future research.
Your opinions

• Cons
  • Overhead of mapping L3.5 to/from underlying layers.
  • Overhead of implementing an L3.5 protocol (in software).
  • Pairwise translators needed at domain edge.
  • Requires some form of cooperation between ASes.
  • Can a network middlebox provide the same functionality at Trotsky?
  • How crucial is the decoupling between L3 and L3.5?
  • To what extent can it provide security?
  • Initial deployment is challenging.
  • “Simplicity is a feature not a bug” – do we really need more complex Internet architectures?
Your opinions

• Ideas
  • Is Trotsky against end-to-end argument?
  • Design DDoS resilient network architecture.
  • Implementation and evaluation of L3.5 protocols.
  • Why not implement Trotsky Processors in programmable switches?
  • What are the limitations of proposed L3 protocols?
  • Explore what incentivizes domains to support an L3.5 protocol.
  • Experiment testbed that allows multiple architecture to co-exist.