Adding Path Awareness to the Internet Architecture

Brian Trammell
ETH Zurich

Jean-Pierre Smith
ETH Zurich

Adrian Perrig
ETH Zurich

Editor:
Yong Cui,
cuiyong@tsinghua.edu.cn

This article presents a vision for path-aware internetworking, surveys research and developments in technology that can be used to enable path awareness, explores research questions posed by a path-aware network architecture, and describes lessons learned from the development of a path-aware Internet architecture.

Today's Internet architecture is based on the ideal of "smart endpoints connected by simple pipes" laid out according to the end-to-end principle.[1] Application logic is implemented at the edge, while the core of the network provides only a best-effort packet-forwarding service based on addressing information present in the packets sent by the endpoints. Furthermore, the evolution of Internet routing protocols means that the paths used by this packet-forwarding service are determined solely by control-plane interactions: the endpoints have no visibility or control over the paths used. However, the Internet has evolved to have paths with diverse properties, and Internet endpoints are increasingly multihomed on different access networks. In other words, the pipes are no longer simple, so the endpoints must become smarter about them, receiving information from the network about the properties of the path or paths used by the traffic they send, and using that information to make choices about the paths to use.

We call this architectural property *path awareness*. This article presents a vision for path-aware internetworking, briefly surveys recent research and developments in Internet technology that can be used to enable path awareness, explores new research questions posed by a path-aware network architecture, and describes lessons learned from the development of the SCION (Scalability, Control, and Isolation on Next-Generation Networks) path-aware Internet architecture.[2]

WHY PATH AWARENESS?

The textbook view of an Internet-connected host assumes a device with a single, fixed connection, and no semantics attached to that connection beyond "This way to the Internet!" In this setting, the device is bound to trust its gateway. That gateway in turn trusts its intradomain routing protocol, which trusts information received from transit, customer, and peer networks via BGP (Border Gateway Protocol), from which a single path from source to destination emerges.

Published by the IEEE Computer Society
1089-7801/18/$33.00 USD

In the current Internet architecture, application developers and users can only assume that packets will be sent toward the intended destination. A transport layer protocol such as TCP can provide reliability over this best-effort service, and a security protocol such as TLS can authenticate the remote endpoint, but no information about the path is available. Assumptions about the path sometimes do not hold, sometimes with serious impacts on the application, as in the case of BGP hijacking attacks. (For an example, see https://dyn.com/blog/mitm-internet-hijacking.)

In addition, an increasing proportion of Internet users are now connected via mobile devices with at least two interfaces, WLAN over terrestrial Internet as well as mobile, with wildly different behavior on each path toward the Internet core. These paths can have usually dynamic properties (such as latency and the available bandwidth), as well as usually static properties (the networks they traverse, and the costs associated with the first hop). These properties are semantically interesting: application developers, users, and network administrators could usefully differentiate among paths to meet application goals. However, the interfaces that would make path information and control available to them are simply missing from the current Internet architecture.

In contrast, a fully path-aware networking architecture would make information about paths and their properties are available to endpoints. These endpoints could then use that information to select paths for given destinations, flows, or even packets. This additional transparency and control, with respect to the current Internet architecture, could be used to enable a variety of new capabilities for transport and application protocols, including

- the use of disjoint network paths for failure resilience and optimization,
- the use of multipath transport even on singly-connected endpoints,
- the exclusion of routes for goals such as resistance to surveillance, and
- the inclusion of certain routes to allow networks to offer services to customers to which they are not directly connected.

The time is ripe to pursue a path-aware architecture as viable for deployment in the Internet due to the emerging availability of several enabling technologies. Multipath TCP[3] has transitioned from a research project to a standardized and deployed transport protocol in the Internet, and shows that it is possible to deploy transport protocols that can take advantage of path information. Other emerging technologies such as the Segment Routing Header for IPv6 or software-defined networking provide interfaces to the data plane that could make path control possible.

OPEN QUESTIONS

Of course, several research questions remain open and must be answered before path awareness can be brought to the Internet at large. First and foremost, the Internet architecture presently has no first-order concept of a path between two endpoints, except as implied by a pair of addresses. It is therefore necessary to define a common vocabulary for describing a path and its properties. This vocabulary must be defined carefully, as its design will have impacts on the expressiveness of path information in a given path-aware internetworking architecture. Path properties may have different levels of guarantee, whether advisory or promised, which affect how they can be used by the endpoints.

Any choices impacting expressiveness also exhibit tradeoffs. For example, a system that always exposes node-level information for the topology through each network would make internal network topology universally public, which may be in conflict with the business goals of each network's operator. Moreover, while a system of path properties binding a user's identity to a path would enable universal authentication end-to-end, it would do so at the potentially significant expense of user privacy.

There are also temporal aspects to the path property vocabulary and, more generally, open questions as to the best way to disseminate path properties toward the endpoints. Any property disseminated by the path that could result in path selection changes should change far less frequently than a path selection can be made. Some properties of a path, such as instantaneous load, are too variable to be of much use by the time they could be disseminated; endpoint measurement will probably remain the best way to ascertain these.

Establishing the authenticity and trustworthiness of path property information is also a hard problem. Attempts at the autonomous system (AS) level to add authentication to the current BGP routing infrastructure have proven hard to deploy. It will also be necessary for networks to authorize endpoints to use certain paths while denying them the use of others. New architectures incorporating authentication into the control plane show promise for making these issues tractable.

The opportunities for optimization presented by a path-aware architecture also raise new questions. Transport protocol design, for example, generally is based on the assumption that packet scheduling is a binary decision: to send, or to wait? When multiple paths are available, this decision is more complicated: which packet to send on which path? Existing literature addresses this topic,[3] but more flexible abstractions and vocabularies will present more complexity. The additional transparency and control provided by a path-aware Internet will also need a richer API, as well as new interfaces for user and network administrator control over path information dissemination and path choice.

A path-aware Internet raises new questions about the dynamics and economics of Internet operations. A fundamental assumption of current Internet interdomain and intradomain routing protocols is that each network operator decides how much of which traffic flows traverse which paths through its network, and how. In a path-aware architecture, endpoints are also involved in this decision. The methods that network operators have to control traffic flows are different (e.g., dissemination or nondissemination of paths, as opposed to announcement of routes by a destination prefix), and control functions within networks and at the endpoints may need tuning to avoid conflict.

Finally, the incentives (and disincentives) to deployment must be examined and understood. Previous attempts to provide similar services through different technologies—e.g., RSVP (Resource Reservation Protocol) and ATM (asynchronous transfer mode)—saw limited uptake by application developers, due to the complexity of the interfaces they provided, the difficulty of adding new interfaces to the stack, and other factors.

To provide a venue for discussing these and other questions together with the Internet standards community, we have proposed the creation of a Path Aware Networking Research Group (PANRG; https://datatracker.ietf.org/rg/panrg/about) within the Internet Research Task Force (IRTF). In addition to providing a venue for discussion, PANRG may also publish RFCs on these questions as input to eventual Internet-scale experimentation and standardization. We briefly explore current research addressing these open questions next.

PATH-AWARE SOCKET API

In a network architecture where properties are bound to paths, the interfaces provided by the transport protocol stack must support paths and their properties as a first-order concept. Application developers, network and system administrators, and even end users may require control over and insight into the paths used by traffic they originate or consume.

We note that the necessary path abstraction is related to other current work in transport APIs, specifically connection racing for dual-stack IPv4/IPv6 hosts (also known as Happy Eyeballs[4]). The Transport Services (TAPS) working group in the Internet Engineering Task Force (IETF) is working on generalizing this concept to one of runtime selection of transport stacks over multiple interfaces. However, TAPS has not defined an API to allow applications to abstract away the differences among these stacks or to manipulate the different interfaces or implicit paths behind them.

The Post Sockets API[5] is an initial attempt at exposing the more dynamic nature of current networking environments to application programmers in a path-aware manner. It is a message-oriented, asynchronous interface in which *Carriers* (analogous to sockets) are bound to an *Association* between two endpoints. Post Sockets abstracts away the selection of transport- and network-layer protocols and interfaces behind this Carrier, as does TAPS. The Association keeps long-term state associated with an endpoint pair (for example, cryptographic identity and resumption parameters, address resolution, and measured path information) at each endpoint. Each

Association is made up of one or more *Paths* between the endpoints, and explicitly associates state with these Paths. The arrangement of abstractions in Post Sockets is shown in Figure 1.

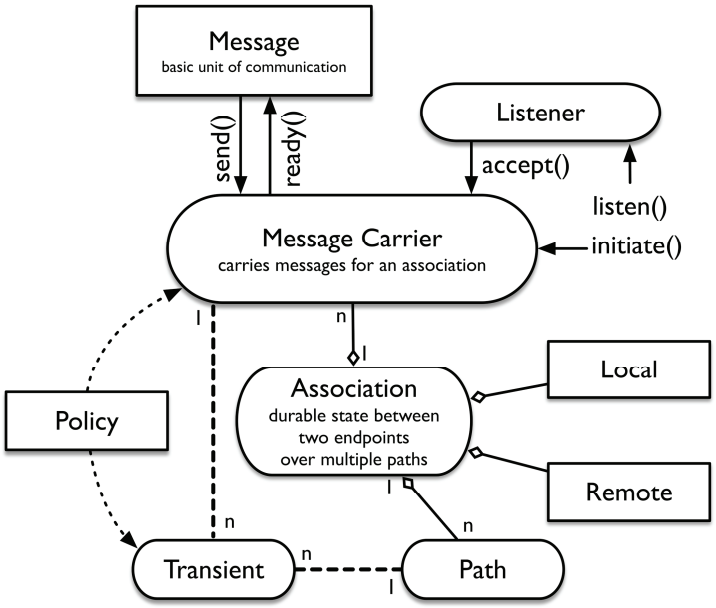

Figure 1. Relationships among abstractions in Post Sockets.

In the current Internet architecture, a Path is represented simply as an address pair, and Path information can be derived from measurement or from current path property dissemination techniques as described in the following section. In a path-aware architecture, this path information would be enriched with topological information and additional metrics describing the path, which could provided directly by the network layer.

Message orientation and the exposure of path information in the API allow the implementation of relatively sophisticated behaviors in the transport layer, taking advantage of the properties of multiple paths when available. Post Sockets' *Message* abstraction exposes priority and deadline information, which can be used in a multipath transport-scheduling algorithm such as that proposed by Chuat et al.[6] Work on Post Sockets is currently ongoing within the IETF (see https://datatracker.ietf.org/doc/draft-trammell-taps-post-sockets).

PATH PROPERTY DISSEMINATION AT INTERNET SCALE

As the Internet has grown, so have the requirements of applications using it. Overlay, content delivery, and peer-to-peer networks all operate in a space where endpoint selection is flexible but critical to the performance of the system. Furthermore, multipath protocols such as Multipath TCP have been shown to be critically sensitive to the initial choice of subflow.[7] To address this, efforts such as iPlane[8] and the Application-Layer Traffic Optimization (ALTO) Protocol[9] have undertaken the task of defining systems for recording and querying path properties.

The increase of scale introduced by a fully path-aware network creates new logistic problems not faced in the current Internet. A common vocabulary of path properties must be defined; these properties must be measured throughout the network and disseminated to end hosts. Both the IP Performance Measurement and ALTO working groups have taken steps toward standardizing the basic metrics associated with a path and the presentation of these metrics, but this work is only a foundation.

How best to generate these metrics at Internet scale still remains unanswered. iPlane uses active measurement from vantage points within the network, but this yields only partial and inferred results. Leveraging network-reported link-state and traffic-engineering information,[10] while more efficient, requires every network along a path to support the same measurement system. Once measured, properties or their deduced metrics must be disseminated to end hosts. Dissemination of path properties typically is done through a client-server architecture. This assumes a central authority with replicated databases in iPlane, or local authorities with partial views of the network in ALTO. In such server-oriented approaches, there is an inherent tradeoff between synchronization overhead and completeness, consistency, and freshness.

To further compound the issue, factors such as the useable lifetime and granularity of metrics, and thus their required update frequency for various application profiles, also need to be determined. The degree to which the requirements of these application profiles can be satisfied must then be balanced with the network overhead associated with satisfying them, to maximize the utility of any chosen dissemination system.

Finally, it is essential that the resulting path-aware network be both stable and secure. The integrity and authenticity of the measurements must be ensured to prevent adversaries from influencing end-host forwarding decisions, yet ISPs need to be able to effectively engineer the traffic received from hosts to respond to unforeseen situations. Additionally, centralized approaches raise the concern of providing yet another vantage point for mass surveillance of user traffic patterns, thereby risking further violation of user privacy.

SCION, A PATH-AWARE INTERNET ARCHITECTURE

The SCION internetworking architecture[2] is a security-oriented and path-aware network architecture that has been developed since 2009. Its goal is to provide high availability for communication in the face of adversarial actions. While intradomain routing and forwarding continue to use existing intradomain control-plane protocols and IP forwarding, interdomain communication follows a different approach.

Forwarding in SCION between ASs is based on packet-carried forwarding state. This state consists of an endpoint-selected path expressed as AS-level path information in each packet's header. The authenticity of the data used to construct this state is rooted in the *isolation domains* (ISDs), of which each AS is a member. Each ISD is managed by a small set of large-scale ISPs, each called a *core AS*, which manage the control plane and establish a trust foundation for the ISD.

Core ASs are responsible for issuing *path-segment construction beacons* (PCBs), which determine network paths. Through inter-ISD PCBs, paths among core ASs are established, and intra-ISD PCBs determine paths from core ASs to smaller leaf ASs. Figure 2a depicts intra-ISD PCB dissemination. An end-to-end SCION path is composed of an intra-ISD path from the source to a core AS, plus an inter-ISD path to a core AS near the destination, and again an intra-ISD path from the final core AS to the destination AS. Thanks to this structure, SCION minimizes the effort for path construction, yet mimics the economic aspects and paths chosen in today's Internet. Path segments are uploaded to a path server infrastructure, enabling destination control of ingress traffic.

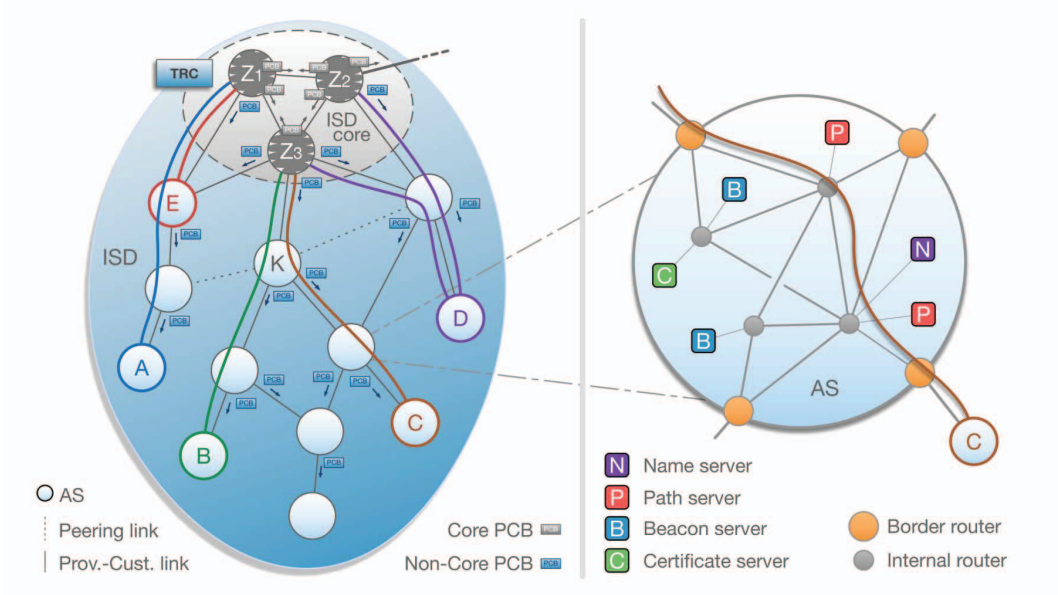

Figure 2. The SCION (Scalability, Control, and Isolation on Next-Generation Networks) ISD (isolation domain). (a) An ISD with three core autonomous systems (ASs—Z1, Z2, and Z3) and several noncore ASs (A, B, etc.). The core ASs define the Trust Root Configuration (TRC), a collection of roots of trust relied upon in that ISD. Core ASs also issue intra-ISD PCBs (path-segment construction beacons) to establish paths from a noncore AS to a core AS. Several paths are depicted with different colors. (b) The structure of an AS containing four SCION border routers and several SCION servers (name, beacon, path, and certificate servers).

As shown in Figure 2b, intradomain routing remains unchanged in SCION. When a SCION packet transits an AS, it is sent directly from border router to border router, based on information carried in the endpoint-selected path, using an intra-AS path determined by existing intradomain routing protocols.

Resilience to link failure in SCION is provided through rapid revocation of paths containing failed links, quick reestablishment of paths over working links, and multipath connectivity: a given connection between two SCION endpoints is backed by multiple paths at any given time. SCION's path control allows the endpoints to use these paths simultaneously, which opens up the possibility to experiment with a much richer set of transport layer semantics.[6]

SCION is an open source project and can be evaluated on an operational testbed. Part of the testbed is based on SCION routers installed in the operational network of several ISPs in Switzerland, which mimics a full deployment within the area covered by these ISPs. Through the SCIONLab system (https://coord.scionproto.net), installation is completely automated within a VirtualBox VM (virtual machine) running on Linux, OS X, or Windows. SCIONLab sets up an AS within the VM and connects it to the SCION testbed, with the advantage that the user has full control over the local AS that participates in the actual network's control plane.

SCION provides specific instantiations of answers to some of the questions presented above, and for the open questions, it provides a foundation for investigation.

The core issues of the dynamics and economics of Internet operations and of the control of traffic flows are handled in several stages. ISPs control the propagation of PCBs, which establish the paths that can be used by end hosts, as cryptographic techniques prevent the formation of unauthorized paths. Moreover, ISPs can attach authorization information to PCBs. Destinations control ingress traffic by determining which paths to themselves are announced to path servers. An extreme case is hidden paths, which are not publicly announced and approximate the functionality of a leased line. The source combines path segments to form an end-to-end path, which also provides control and transparency. The combination of these mechanisms defines a common vocabulary for describing a path and its properties, and define the temporal aspects. Interfaces for

user and network administrator control, though, can benefit from further research based on the results from the current deployment.

Transport protocol design can make use of the SCION architecture and testbed as a foundation for path awareness and, for instance, can make use of SCION's inherent multipath ability. Questions such as which packet to send on which path also need to be further investigated. To take advantage of the properties of path-aware networking, developers need a richer API, such as that provided by Post Sockets.

CONCLUSION

The Internet architecture has been successful in large part because of its adherence to the end-to-end principle. It is, however, based on an assumption that the properties of the network paths carrying packets are semantically uninteresting to the respective applications. We argue that this assumption may have outlived its usefulness in today's increasingly mobile and heterogeneously connected networking environments, and that involving endpoints in path selection based on the properties of those paths is a purer interpretation of the end-to-end principle. PANRG is being chartered to examine the research questions this rethinking raises, and we invite you to join us in this conversation on our mailing list: panrg@irtf.org.

REFERENCES

1. J.H. Saltzer, D.P. Reed, and D.D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, 1984, pp. 277–288.
2. D. Barrera et al., "The SCION internet architecture," *Communications of the ACM*, vol. 60, no. 6, 2017, pp. 56–65.
3. C. Paasch and O. Bonaventure, "Multipath TCP," *Communications of the ACM*, vol. 57, no. 4, 2014, pp. 51–57.
4. *Happy Eyeballs: Success with Dual-Stack Hosts*, IETF RFC RFC 6555, Internet Engineering Task Force (IETF), 2012; https://tools.ietf.org/html/rfc6555.
5. B. Trammell, C. Perkins, and M. Kuehlewind, "Post Sockets: Towards an Evolvable Network Transport Interface," *2017 IFIP Networking Workshop on Future of Internet Transport*, 2017.
6. L. Chuat, A. Perrig, and Y.-C. Hu, "Deadline-Aware Multipath Communication: An Optimization Problem," *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (DSN 17), 2017.
7. S. Deng et al., "WiFi, LTE, or Both?," *Proceedings of the 2014 Conference on Internet Measurement Conference* (IMC 14), 2014.
8. H.V. Madhyastha et al., "iPlane: An Information Plane for Distributed Services," *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, 2006, pp. 367–380.
9. undefined undefined and undefined undefined, *Application-Layer Traffic Optimization (ALTO) Protocol*, IETF RFC RFC 7285, Internet Engineering Task Force (IETF), 2014; https://tools.ietf.org/html/rfc7285.
10. *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*, IETF RFC RFC 7752, Internet Engineering Task Force (IETF), 2014.

ABOUT THE AUTHORS

Brian Trammell is a senior researcher in the Networked Systems Group and the Network Security Group at ETH Zurich. Contact him at trammell@tik.ee.ethz.ch.

Jean-Pierre Smith is a doctoral student in the Network Security Group at ETH Zurich. Contact him at jean-pierre.smith@inf.ethz.ch.

Adrian Perrig is a professor of computer science and the director of the Network Security Group at ETH Zurich. Contact him at adrian.perrig@inf.ethz.ch.