COLUMN: Standards

Paradoxes of Internet Architecture

Srinivasan Keshav
University of Waterloo

The architectural elements of the Internet that led to its great success are now, paradoxically, the source of its severest problems. For example, the use of autonomous systems to distribute governance has allowed rapid growth and scaling but has made the network unmanageable and unable to provide end-to-end quality of service. This article examines this and other key design elements of Internet architecture and shows how they have contributed to its success and how they now constrain it. I then use this framework to identify some key challenges that need to be addressed in the next phase of Internet research.

The Internet is one of the great technical success stories of our age. Starting with only four nodes in 1969, used only by researchers, and only for a very limited suite of text-oriented applications,[1] it has grown to be a world-spanning infrastructure providing sophisticated services that play a critical role in the daily lives of literally billions of people. As we look back at the last nearly 50 years of Internet evolution, it is impossible not to be astonished at its amazing growth and the degree to which it has satisfied its highly ambitious goal of providing access to any information to anyone, anywhere, at any time. Indeed, anyone in the world today, even if that person is in some rural backwater, as long as he or she has a smartphone and wireless Internet, has more access to information than presidents, kings, and emperors of generations past!

Given this success, it is tempting to ask the question, are we done? Is it time for computer-networking researchers to pack up their bags and look for another research area? Given the forum where this article is being published, it will not be a surprise to learn that my answer is no. Indeed, the Internet suffers from some deep-rooted problems that will take considerable additional research to solve. Moreover, as others have also observed, such as in the context of the US National Science Foundation's FIND (Future Internet Design) project,[2] the same structural elements of the Internet that led to its success also lie at the root of these problems. Hence, solving them will require us to re-architect the Internet, while still being legacy compatible.

The rest of this paper is laid out as follows. First, I discuss three significant problems with the Internet today. Next, I present the design principles of the Internet, as laid out by Clark in his seminal paper from ACM SIGCOMM 1988.[3] Then, I discuss how these design principles led to the problems I mentioned. Finally, I propose some steps toward a future Internet architecture.

Published by the IEEE Computer Society
1089-7801/18/$33.00 ©2018 IEEE

THREE PROBLEMS WITH THE INTERNET

Despite its considerable success, there are at least three fundamental problems with the Internet today, as I discuss next. (Other researchers have highlighted issues such as the lack of support for mobile endpoints[4] and the focus on packets rather than data.[5] These have received considerable research attention, so they are not considered here.)

Spam

The Internet has made it possible for unwanted messages to be delivered to email inboxes, website comments, and even telephones at an unprecedented and, in some cases, unsustainable rate. On the Internet, since the marginal cost of an email message is nearly zero, it is profitable for fly-by-night commercial entities to send out millions of messages to get even one sale. This wastes not only network capacity but also the time of the recipients. Despite many efforts, spam continues to be a problem worldwide, with every spam-control measure seemingly blocked by a countermeasure by ever-wilier spammers.

Why can't we get rid of spam once and for all?

Privacy and Security

The topic of Internet privacy and security (or the lack, actually) is quite complex, so I will focus on a single problem, which is that packets, once given to an Internet service provider (ISP), can be stored, analyzed, modified, replayed, or dropped with impunity. This clearly impacts both the security and privacy of Internet users.

From the perspective of privacy, standard Internet users reveal both packet contents and destination addresses to every ISP on the path to the destination. Although TLS and HTTPS can be used to protect packet contents, and VPNs and relay systems such as Tor can be used to protect packet destinations, these additional protections typically are not turned on by default, thus violating user privacy, for example, to make it possible to display highly targeted advertising. Even with these protections, privacy can be obtained only at the cost of substantial performance degradation.

From the perspective of security, ISPs are allowed and, in some cases, required, to make copies of every packet they carry. These can be used, for example, to carry out packet replay attacks.

Why does the Internet provide such little support for security and privacy?

Quality of Service

The dominant application on the Internet today is noninteractive video delivery from services such as NetFlix and YouTube. While this form of video data does not have real-time service requirements, only a minimum throughput requirement, the Internet does not meet even the rather relaxed service requirements of these applications. It certainly does not always meet the stringent throughput and latency requirements for two-way interactive voice and video services such as Skype and WebEx. Given that the telephone network, a special-purpose communication network, could provide excellent voice quality using technology from the 1960s, it is surprising that voice quality on the state-of-the-art Internet is worse that it was half a century ago.

Why can't the Internet support quality of service (QoS)?

INTERNET DESIGN PHILOSOPHY

The thesis of this article is that the three problems I discussed earlier arise due to the design philosophy of the Internet itself. In other words, had the Internet been designed differently, these problems would not have arisen (although others would). To understand this better, it is necessary to examine the design philosophy of the Internet. Fortunately, this was laid out by David Clark in 1988:

> *The top level goal for the DARPA Internet Architecture was to develop an effective technique for multiplexed utilization of existing interconnected networks.*[3]

This goal is then further elaborated to encompass the following seven subgoals:

> *1. Internet communication must continue despite loss of networks or gateways.*
>
> *2. The Internet must support multiple types of communications service.*
>
> *3. The Internet architecture must accommodate a variety of networks.*
>
> *4. The Internet architecture must permit distributed management of its resources.*
>
> *5. The Internet architecture must be cost effective.*
>
> *6. The Internet architecture must permit host attachment with a low level of effort.*
>
> *7. The resources used in the Internet architecture must be accountable.*
>
> ...
>
> *It is important to understand that these goals are in order of importance, and an entirely different network architecture would result if the order were changed.*[3]

I now discuss the impact of these goals on the success of the Internet and on the three problems I presented earlier.

PARADOXES OF INTERNET DESIGN

The Cost–Performance Tradeoff

One of the main contributors to the success of the Internet was that it was much cheaper than telephony: for instance, sending email is much cheaper than communicating the same message over a voice call. This is because unlike telephone circuits, packet-based Internet communication allows links to be statistically multiplexed, reducing the cost of message transmission (the top-level goal and subgoal 5).

The cost reduction that comes from statistical multiplexing, however, comes at a price. With circuit switching, once a call is admitted, throughput is guaranteed and end-to-end delay is tightly constrained. In contrast, with packet switching, and especially in packet networks lacking admission control such as the Internet, there is no throughput guarantee, and the only (loose) bound on end-to-end delay comes from the size of router buffers. Thus, it is challenging, if not impossible, to provide end-to-end latency bounds when network links are overloaded (i.e., the short-term arrival rate exceeds the link service rate).

There have been literally thousands of research projects that have attempted to add QoS guarantees to packet-switched networks. Broadly speaking, these fall into four categories:

1. *Overprovisioning*. The idea here is to keep all network links at low loads, so that there are no packet losses and queueing delays are bounded. Although effective, this approach is expensive and makes the Internet less cost-effective than it could otherwise be.
2. *Admission control*. With admission control, overloaded links can refuse to admit more "calls" (TCP connections or UDP [User Datagram Protocol] packets). Implementing

this is impossible in the current Internet because a TCP connection does not characterize the amount of traffic that it expects to carry. The IntServ[6] RSVP protocol[7] did carry this information, but widespread deployment of RSVP requires close cooperation of ISPs and a mechanism for settlement of payments. But these contradict subgoal 4, which led the Internet to be partitioned into a set of minimally cooperating autonomous systems.

3. *Source throttling*. If sources could learn of overloaded links on their path, they could scale back their transmission rate. This is indeed the approach used in TCP, where packet losses are used to signal link overload, resulting in throttling back of the source rate. The problem with this approach is that it does not necessarily provide throughput or delay guarantees. Moreover, there is no way to enforce source compliance.
4. *Differential service quality*. Unlike the IntServ proposals, the DiffServ proposals for QoS proposed that certain flows could be given better performance, at the cost of the others.[9] While simple and easy to implement, they ran afoul of the same problem as RSVP: autonomous systems (ASs) have no reason to respect service priority requests from other ASs unless the requests are accompanied by a payment. Lacking a reliable accounting system (subgoal 7 was disavowed even in the 1988 paper), this simply did not work out.

In summary, the design decision to focus on multiplexed links did reduce costs, but at the cost of performance. Worse, the partitioning of the Internet into autonomous systems, whose only requirement was packet delivery, has all but ensured that even in the future, the Internet will be unable to provide end-to-end throughput or latency guarantees. I discuss this next.

Autonomous Systems with Narrow Interfaces

To address the fourth subgoal of distributed management, the Internet is physically partitioned into ASs that have a considerable degree of freedom in their operations. Moreover, to achieve the third subgoal (accommodate a variety of networks), an AS is required to support the lowest common denominator, which is best-effort packet delivery. Thus, the interface between two ASs is defined only by addressing and packet transfer (defined by IPv4 and IPv6) and routing-information transfer (defined by BGP [Border Gateway Patrol]). Other than that, ASs can make no assumptions about each other. These two design decisions have led to the Internet being constructed from loosely coupled heterogeneously managed ASs.

This decision has been critical in allowing the rapid growth of the Internet. By distributing management decisions and requiring little coordination between ASs (other than the weak control imposed by ICANN [Internet Corporation for Assigned Names and Numbers] and IANA [Internet Assigned Numbers Authority]), it is highly scalable. These decisions also allowed the nascent Internet to co-opt existing networks, such as the European PTTs (postal telegraph and telecommunications administrations). Moreover, it permitted the piecemeal incorporation of a heterogeneous set of underlying networking technologies into the Internet. Less obviously, this lowest-common-denominator approach makes it possible for ASs to independently evolve. Thus, one AS can be based on long-latency satellite links, while another can use high-speed multilambda fibers, and they can still interoperate. Arguably, an early IPv4 from the mid-1970s would still continue to work, were it plugged into the Internet today!

However, this decentralized and best-effort architecture comes at a price. To begin with, it does not allow end-to-end QoS guarantees, as we have already observed. A more challenging problem is that it permits an AS to employ arbitrary complexity as long as packet delivery is preserved. Thus, ASs can use complex layerings (such as IP on VPN on MPLS [Multiprotocol Label Switching] on ATM [asynchronous transfer mode] on virtual PHY on PHY). This makes the overall system very hard to debug. Arguably, the Internet is unmanageable by design, in that no single entity is aware of the entire topology or its physical instantiation! Moreover, ASs have no constraints on packet capture, modification, and replay as long as they provide packet transport.

We can now see why spam will continue to be a pernicious problem on the Internet. The inter-AS interface carries only packets and routing information. Thus, while one might complain to one's ISP about spam, the ISP has no way to request a peer AS to trace back the spammer. The situation is even worse with DDoS (distributed denial of service) attacks, where the IP address of each attacking bot may be known, but the decentralized nature of the Internet makes it nearly impossible to unearth the identity of the bot master, who may be several AS hops away and operating from an unfriendly political jurisdiction.[9]

Attachment and Identity

To allow rapid deployment, the sixth subgoal is to allow hosts to be attached to the network with little effort. It is instructive to compare the effort it takes to attach a host to the Internet with the effort it takes to attach, say, a mobile phone to the telephone network. In the former case, all the host needs is an IP address, a network mask, and the IP address of a router that is willing to accept a link to it. Even these can be automatically provided using zero-configuration technologies such as DHCP (Dynamic Host Configuration Protocol).[10] In contrast, for a mobile phone, it is necessary to obtain a tamper-proof ID such as an IMEI (International Mobile Equipment Identity), as well as, in most cases, provide credible proof of identity such as with a credit card.

It is immediately obvious that the ease of attachment to the Internet, with no checking or authentication of identity, is what allows it to be very successful, in that endpoints can be created on the fly. On the other hand, this reduction in attachment effort leads to a situation where Internet endpoints are not bound to identities. Hence, malicious actions that originate from these endpoints cannot be bound to entities subject to legal restraints. This is another reason why spam and DDoS attacks are rife on the Internet, in contrast to mobile-phone networks. In the latter, if an endpoint were to send out spam, the authorities could take action against the bound identity. No such thing is possible in the Internet, especially given the degree to which endpoints have been surreptitiously converted to bots. The lack of reliable identity also makes it challenging, if not impossible, to provide end-to-end QoS guarantees, because it is impossible to bill an end user for higher-quality service.

TOWARD A NEW INTERNET ARCHITECTURE

The thesis of this article is that troubling issues such as spam, lack of privacy and security, and lack of QoS are not accidental byproducts of the Internet but arise from its very design. Thus, if these are to be tackled, it will be necessary to re-architect the Internet.

This thesis is not particularly novel. There has already been more than a decade of work on future Internet architectures[2] that focuses on issues of "clean-slate" Internet design, essentially proposing to re-architect the Internet. Although a full review of these designs is beyond the scope of this article, a general characterization of this work is that it focuses on foundational issues of naming, addressing, and routing, as well as, to some extent, privacy and security. It also tackles the lack of QoS and the prevalence of communication spam. Early examples of work in this area include I3[11] and Plutarch.[12] A survey of recent work in this area can be found in the July 2014 *ACM SIGCOMM Computer Communication Review*, which includes an overview of the effort[13] as well as papers on XIA (Expressive Internet Architecture),[14] NDN (Named Data Networking),[5] ChoiceNet,[15] MobilityFirst,[4] and NEBULA.[16]

Perhaps the most ambitious redesign effort is from the SCION (Scalability, Control, and Isolation on Next-Generation Networks) research group at ETH Zurich, which has proposed a new Internet architecture that fully incorporates security and trust.[17,18] SCION allows packet paths to be traced, but without compromising end-user privacy. It is also legacy compatible, and with its SIBRA (Scalable Internet Bandwidth Reservation Architecture) extension[19] lets hosts make end-to-end bandwidth reservations, thus permitting end-to-end QoS guarantees and defense against DDoS attacks.

On a related note, David Clark, who wrote the 1988 paper on Internet design philosophy, has put together a book-length treatment of his ideas on Internet architecture.[20] It makes very interesting reading from the acknowledged master of this field and will be sure to inspire future network architects in tackling the complex issues that arise from statistical multiplexing, narrow inter-AS interfaces, and lack of endpoint identity.

Although this short note is certainly not the place to lay out new architectural principles, it is tempting to observe that the easiest locus for intervention might be the inter-AS interface at a NAP (network access point), allowing QoS metadata, settlement information, and identities to traverse the AS–AS boundary. One can imagine that two geographically contiguous mobile-phone-based Internet providers, such as Tencent and China Unicom, who already have bindings between endpoints and identities, might wish to mutually exchange identity and QoS information to improve their operational security and the QoS received by their customers. This mutually beneficial core can expand to include other ASs and other NAPs over time. Of course, this is only a high-level sketch, but this general direction seems promising.

CONCLUSION

Despite the Internet's great success, Internet research is far from over: we need to address some endemic issues that arise from the design principles of the Internet itself. To do so will require a reexamination of the original design principles and the elucidation of new principles and an associated new architecture. This should be of great interest to the next generation of Internet researchers.

ACKNOWLEDGMENTS

This article is based on my presentation at the ACM TURC conference organized by the ACM China Council in Shanghai, China, in May 2017 (blizzard.cs.uwaterloo.ca/iss4e/wp-content/uploads/2017/08/paradoxes.pdf). My thanks to the organizers for inviting me to present my thoughts in this forum. I benefited from insightful comments from Jon Crowcroft and Adrian Perrig. My sincere thanks to them for their helpful feedback.

REFERENCES

1. B.M. Leiner et al., "A brief history of the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, 2009, pp. 22–31.
2. "NETs FIND Project," National Science Foundation, 2009; http://www.nets-find.net/index.php.
3. D.D. Clark, "The Design Philosophy of the DARPA Internet Protocols," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, 1988, pp. 106–114.
4. A. Venkataramani et al., "Mobility.rst: A mobility-centric and trustworthy internet architecture," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014, pp. 74–80.
5. L. Zhang et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014, pp. 66–73.
6. D.D. Clark, S. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: Architecture and mechanism," *ACM SIGCOMM Computer Communication Review*, vol. 22, no. 4, 1992, pp. 14–26.
7. L. Zhang et al., "Rsvp: A new resource reservation protocol," *IEEE Network*, vol. 7, no. 5, 1993, pp. 8–18.
8. K. Nichols and B. Carpenter, *Definition of differentiated services per domain behaviors and rules for their specification*, technical report, 2001.
9. D.D. Clark and S. Landau, "Untangling attribution," *Harvard National Security Journal*, 2011.
10. D.H. Steinberg and S. Cheshire, *Zero Configuration Networking: The Definitive Guide*, O'Reilly Media, 2005.
11. I. Stoica et al., "Internet in-direction infrastructure," *ACM SIGCOMM Computer Communication Review*, vol. 12, no. 2, 2002, pp. 73–86.
12. J. Crowcroft et al., "Plutarch: an argument for network pluralism," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 4, 2003, pp. 258–266.

13. D. Fisher, "A look behind the future internet architectures efforts," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014, pp. 45–49.
14. D. Naylor et al., "Xia: Architecting a more trustworthy and evolvable internet," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014, pp. 50–57.
15. T. Wolf et al., "Toward an economy plane for the internet," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014, pp. 58–65.
16. T. Anderson et al., "A brief overview of the nebula future internet architecture," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, ACM SIGCOMM Computer Communication Review, 2014, pp. 81–86.
17. D. Barrera et al., "The SCION Internet Architecture," *Communications of the ACM*, vol. 60, no. 6, 2017, pp. 56–65.
18. X. Zhang et al., "SCION: Scalability, control, and isolation on next-generation networks," *Proceedings of the 2011 IEEE Symposium on Security and Privacy* (SP 11), 2011, pp. 212–227.
19. A. Perrig et al., *Scion: A secure internet architecture*, Springer International, 2017.
20. D.D. Clark, "Designs for an Internet," 2017; https://groups.csail.mit.edu/ana/People/DDC/ebook-arch.pdf.

ABOUT THE AUTHOR

Srinivasan Keshav is a professor at the University of Waterloo's School of Computer Science. Contact him at keshav@uwaterloo.ca.