ARTICLE

PROTOCOL LAYERING AND INTERNET POLICY

CHRISTOPHER S. YOO†

An architectural principle known as protocol layering is widely recognized as one of the foundations of the Internet's success. In addition, some scholars and industry participants have urged using the layers model as a central organizing principle for regulatory policy. Despite its importance as a concept, a comprehensive analysis of protocol layering and its implications for Internet policy has yet to appear in the literature. This Article attempts to correct this omission. It begins with a detailed description of the way the five-layer model developed, introducing protocol layering's central features, such as the division of functions across layers, information hiding, peer communication, and encapsulation. It then discusses the model's implications for whether particular functions are performed at the edge or in the core of the network, contrasts the model with the way that layering has been depicted in the legal commentary, and analyzes attempts to use layering as a basis for competition policy. Next the Article identifies certain emerging features of the Internet that are placing pressure on the layered model, including WiFi routers, network-based security, modern routing protocols, and wireless broadband. These developments illustrate how every architecture inevitably limits functionality as well as the architecture's ability to evolve over time in response to changes in the

† John H. Chestnut Professor of Law, Communication, and Computer & Information Science and Founding Director of the Center for Technology, Innovation and Competition (CTIC), University of Pennsylvania. I would like to thank the Milton and Miriam Handler Foundation, the New York Bar Foundation, and CTIC for their financial support for this project. I would also like to thank the participants in a workshop conducted at the Center for Law and Information Policy at Fordham Law School; the *Duke Law Review*'s 41st Annual Administrative Law Symposium; the workshop on Technology Policy, Law and Economics co-sponsored by the Swiss Federal Institute of Technology (ETH Zürich) and the University of Zürich; and the *University of Pennsylvania Law Review*'s symposium, "The Evolving Internet," held at the University of Pennsylvania Law School, as well as Adam Aviv, Steve Bellovin, Matthew Blaze, Vint Cerf, Bob Kahn, Howard Shelanski, Larry Solum, Jonathan Smith, Konstantinos Stylianou, Rick Whitt, and Jonathan Zittrain for their comments on earlier drafts. All errors are the responsibility of the author.

technological and economic environment. Together these considerations support adopting a more dynamic perspective on layering and caution against using layers as a basis for a regulatory mandate for fear of cementing the existing technology into place in a way that prevents the network from innovating and evolving in response to shifts in the underlying technology and consumer demand.

INTRODUCTION

One of the most striking developments of the past two decades is the emergence of the Internet both as the dominant medium of communication and as a dynamic engine of innovation. Policymakers and commentators typically attribute the Internet's success to key architectural principles incorporated into its design.[1] Among the most frequently cited of these principles is a concept known as protocol layering,[2] which was first developed for the International Standards Organization's (ISO) Open System Interconnection (OSI) Reference model during the late 1970s.[3] Layering has become so widely accepted that it now represents the central framework around which most textbooks on network engineering are organized.[4]

---

1 *See, e.g.*, Preserving the Open Internet, Broadband Industry Practices, 25 FCC Rcd. 17905, 17909 para. 11 (2010) (report and order) [hereinafter Open Internet Order] (calling aspects of the Internet's architecture "critical to its unparalleled success"); *Net Neutrality: Hearing Before the S. Comm. on Commerce, Sci. & Transp.*, 109th Cong. 54 (2006) (prepared statement of Lawrence Lessig, C. Wendell and Edith M. Carlsmith Professor of Law, Stanford Law School) ("[T]he innovation and explosive growth of the Internet [has been] directly linked to its particular architectural design."); *id.* at 9 (prepared statement of Vinton G. Cerf, Vice Pres. & Chief Internet Evangelist, Google Inc.) ("The remarkable success of the Internet can be traced to a few simple network principles—end-to-end design, layered architecture, and open standards . . . .").

2 *See, e.g.*, HAL ABELSON ET AL., BLOWN TO BITS 312-13 (2008) (arguing that technological convergence means that "laws and regulations should respect layers" rather than treating each medium, such as telephony, cable, and radio, as its own silo); COMM. ON THE INTERNET IN THE EVOLVING INFO. INFRASTRUCTURE ET AL., NAT'L RESEARCH COUNCIL, THE INTERNET'S COMING OF AGE 4, 36-38 (2001) (discussing the importance of the "hourglass" structure of the Internet, where innovation occurs "at the edge of the network"). The other principal architectural feature is known as the end-to-end argument. I have offered my initial thoughts on this principle elsewhere. *See generally* Christopher S. Yoo, *Would Mandating Broadband Network Neutrality Help or Hurt Competition? A Comment on the End-to-End Debate*, 3 J. ON TELECOMM. & HIGH TECH. L. 23 (2004). I plan to offer a more detailed analysis in my future work.

3 On the OSI Reference Model's adherence to layering, see Meeting of Int'l Org. of Standards Technical Comm., ISO/TC 97/SC 16 N34, Provisional Model of Open-Systems Architecture (1978), *reprinted in* COMPUTER COMM. REV., July 1978, at 49, 49-56; and Hubert Zimmermann, *OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection*, 28 IEEE TRANSACTIONS ON COMM. 425, 426-27, 429-30 (1980). On the connections between layering in the OSI Reference Model and the Internet model, see M.A. Padlipsky, *A Perspective on the ARPANET Reference Model* 4, 8 (Internet Eng'g Task Force (IETF) Network Working Grp. Request for Comments (RFC) No. 871, 1982) [hereinafter RFC 871], *available at* http://tools.ietf.org/pdf/rfc871.

4 *See, e.g.*, 1 DOUGLAS E. COMER, INTERNETWORKING WITH TCP/IP 161-63 (5th ed. 2006); JAMES F. KUROSE & KEITH W. ROSS, COMPUTER NETWORKING: A TOP-DOWN

Indeed, belief in the layered model has become so strong that it is often widely regarded as the "proper" way to modularize a network.[5]

There is widespread agreement that the incorporation of protocol layering into the Internet's architecture has yielded substantial benefits. Layering allows those working on one layer to ignore most of the inner workings of the other layers, which reduces coordination costs and accelerates product development times by permitting parallel testing and innovation. Layered architectures also provide a stable configuration of network resources and interfaces around which actors can focus their efforts. The current architecture has also proven incredibly resilient. Despite originally being designed for a much smaller scale and a more limited technological and economic context, the Internet now integrates a larger number and greater variety of uses and technologies than its designers ever imagined.[6]

Protocol layering has also found its way into discussions of Internet policy.[7] Early commentators offered it as a technologically agnostic alternative to the regime established by the Communications Act of 1934,[8] which subjected communications to distinct regulatory regimes based on whether

---

APPROACH 48-56 (5th ed. 2010); ANDREW S. TANENBAUM, COMPUTER NETWORKS (4th ed. 2003). Even textbooks organized along different lines still mention protocol layering prominently. *See, e.g.*, LARRY L. PETERSON & BRUCE S. DAVIE, COMPUTER NETWORKS: A SYSTEMS APPROACH 19-30 (4th ed. 2007).

5 *See* David Clark, *Foreword to the First Edition of* PETERSON & DAVIE, *supra* note 4, at ix, ix ("All good computer scientists worship the god of modularity . . . . The field of network protocols is perhaps unique in that the 'proper' modularity has been handed down to us in the form of an international standard: the seven-layer reference model of network protocols from the ISO."); David D. Clark, *Modularity and Efficiency in Protocol Implementation* 24 (IETF Network Working Grp. RFC No. 817, 1982) [hereinafter RFC 817], *available at* http://tools.ietf.org/pdf/rfc817 (noting the "tempt[ation] to think that a layer boundary . . . is in fact the proper boundary to use in modularizing the implementation").

6 *See* STUART MINOR BENJAMIN ET AL., TELECOMMUNICATIONS LAW AND POLICY 713-21 (3d ed. 2012).

7 For leading discussions of layering in the legal literature, see LAWRENCE LESSIG, THE FUTURE OF IDEAS: THE FATE OF THE COMMONS IN A CONNECTED WORLD 23-25 (2001); BARBARA VAN SCHEWICK, INTERNET ARCHITECTURE AND INNOVATION 46-57, 83-105 (2010); JONATHAN ZITTRAIN, THE FUTURE OF THE INTERNET AND HOW TO STOP IT 67-69 (2008); Lawrence B. Solum & Minn Chung, *The Layers Principle: Internet Architecture and the Law*, 79 NOTRE DAME L. REV. 815, 823-849 (2004); Adam Thierer, *Are "Dumb Pipe" Mandates Smart Policy? Vertical Integration, Net Neutrality, and the Network Layers Model*, 3 J. ON TELECOMM. & HIGH TECH. L. 275, 279-83 (2005); Kevin Werbach, *A Layered Model for Internet Policy*, 1 J. ON TELECOMM. & HIGH TECH. L. 37, 57-64 (2002); Richard S. Whitt, *A Horizontal Leap Forward: Formulating a New Communications Public Policy Framework Based on the Network Layers Model*, 56 FED. COMM. L.J. 587, 601-09, 614-36 (2004); and Timothy Wu, Essay, *Application-Centered Internet Analysis*, 85 VA. L. REV. 1163, 1164, 1189-92 (1999).

8 Pub. L. No. 416, 48 Stat. 1064 (codified as amended in 47 U.S.C.).

they were transmitted over telephone wires, coaxial cables, or spectrum.[9] Others argued that the layered model remained properly agnostic about the content of the rules, but argued that problematic practices arising in one layer be addressed only through regulations directly targeted at that layer—rather than through regulations designed to curb that behavior by targeting another layer or the system as a whole.[10]

Other analyses have drawn stronger policy inferences from the layered model. For example, some commentators argued that the layered model could support competition policy by providing "natural boundaries" for defining markets,[11] noting that each layer is subject to different sources of market power.[12] Others went further, suggesting that the economics of the lower layers made them particularly susceptible to market power, although they acknowledged the possibility of deregulating the lower layers once they became more competitive.[13] Others argued that layering promotes "fair

---

9 *See* Douglas C. Sicker & Joshua L. Mindel, *Refinements of a Layered Model for Telecommunications Policy*, 1 J. ON TELECOMM. & HIGH TECH. L. 69, 72 (2002) (explaining the "existing policy" of regulating different services under different titles of the Communications Act); Werbach, *supra* note 7, at 64-65 (discussing how the distinct regulation of telephone and cable led to inconsistent treatment between similar technologies, such as DSL and cable modem services); Whitt, *supra* note 7, at 615-17 (surveying critiques of the current silo approach that uses the underlying technology as the basis for regulation rather than concepts of layering).

10 *See* Solum & Chung, *supra* note 7, at 849, 853 (defining the "layers principle" as the need to "respect the integrity of the layers" and advocating "minimization of layer-crossing regulation"); Douglas C. Sicker, Further Defining a Layered Model for Telecommunications Policy 13 (Oct. 3, 2002) (paper presented at the 30th Telecomms. Policy Research Conf.), *available at* http://www.learningace.com/doc/1675669/875a17ec13593859fdd613067974f72b/tprc_layered_model ("[P]olicy issues at one layer should be recognized as different from policy issues at another layer.").

11 Robert Cannon, *The Legacy of the Federal Communications Commission's Computer Inquiries*, 55 FED. COMM. L.J. 167, 195 (2003).

12 *See* Robert M. Entman, *Transition to an IP Environment* ("Higher degrees of competition may be more feasible and desirable at some layers than others. Therefore, policymakers should recognize that a pro-competitive policy may need to treat different layers differently."), *in* ASPEN INST., TRANSITION TO AN IP ENVIRONMENT: A REPORT OF THE FIFTEENTH ANNUAL ASPEN INSTITUTE CONFERENCE ON TELECOMMUNICATIONS POLICY 1, 2-3 (2001), *available at* http://www.aspeninstitute.org/sites/default/files/content/docs/cands/TRANSITION_BK.PDF; Michael L. Katz, *Thoughts on the Implications of Technological Change for Telecommunications Policy* (noting that "the assessment of market power should largely take place at each layer separately" and discussing how the sources of market power at the transport layer differ from the sources of market power at the applications layer), *in* ASPEN INST., *supra*, at 25, 37-38.

13 *See, e.g.*, Werbach, *supra* note 7, at 66 ("If the physical and logical infrastructure layers in the relevant markets were sufficiently competitive, ILECs would not be able to gain unfair advantage over competitors at the application and content layers."); Whitt, *supra* note 7, at 592, 649, 653, 667 ("[W]hen applied in the telecommunications industry context, the Network Layers Model targets the lower network layers for discrete regulation based on the existence of significant market power, rather than legacy service or industry labels.").

and open competition" among providers offering services at each layer.[14] Still others equated layering with innovation[15] and advocated regulations mandating that the interfaces between layers remain open.[16]

More recent analyses have relied on the existing layered architecture as the foundation for proposals to implement the Open Internet Order. For example, some commentators argue for using consistency with the existing layered architecture as the first screen for determining whether a traffic management practice is reasonable,[17] a position endorsed by certain policy advocates.[18] Others advocate a nondiscrimination rule that maps onto the layered architecture, arguing that lower layers should be forbidden from discriminating on the basis of any information contained in the upper layers.[19]

The Internet's success should not obscure, however, that every architecture necessarily has limitations as well as strengths. David Clark, who was the chief protocol architect for the ARPANET during the 1980s, offered

---

14 Ashish Shah et al., Thinking About Openness in the Telecommunications Policy Context 13 (Sept. 1, 2003) (unpublished manuscript), *available at* http://ssrn.com/abstract=2060641.

15 *See* Solum & Chung, *supra* note 7, at 816 ("The role of the Internet in enabling innovation is not accidental; rather it flows from the Internet's [layered] architecture."); Whitt, *supra* note 7, at 629 (noting "the strong correlation between robust, ends-oriented innovation and the architecture of the Internet"); Wu, *supra* note 7, at 1192-93 (claiming that the layered architecture of the Internet supports "an astoundingly large set of possible applications").

16 Entman, *supra* note 12, at 16; *see also* Werbach, *supra* note 7, at 66-67 (analyzing conditions under which rules preventing the closure of interfaces between layers would be desirable).

17 Scott Jordan, *Four Questions that Determine Whether Traffic Management Is Reasonable*, 2009 IFIP/IEEE INT'L SYMP. ON INTEGRATED NETWORK MGMT. 137, 138 (2009); *see also* Scott Jordan & Arijit Ghosh, *A Framework for Classification of Traffic Management Practices as Reasonable or Unreasonable*, 10 ACM TRANSACTIONS ON INTERNET TECH. 12:1, 12:7 (2010) (proposing an initial screening test for whether a traffic management technique is applied at the right location for the layer).

18 *See* Comments of Google Inc. 25-26, 69-70 (Jan. 14, 2010), *available at* http://fjallfoss.fcc.gov/ecfs/document/view?id=7020378725 (arguing that "the physical layer provider is uniquely positioned to impede, hinder or deter consumer access to other applications providers," whereas "[a]pplications layer providers obviously do not have a comparable ability," and that "[t]his stark functional difference warrants government scrutiny of lower layer activities" and endorsing Jordan's proposal that "[n]etwork congestion techniques also should be consistent with Internet layers architecture"), *commenting on* Preserving the Open Internet, 24 FCC Rcd. 13064, GN Docket No. 09-191 (2009) (notice of proposed rulemaking); Reply Comments of Ctr. for Democracy & Tech. 18-19 (2010), *available at* http://fjallfoss.fcc.gov/ecfs/document/view?id=7020437353 (supporting traffic-management practices where they focus on transmission and do not affect caching and paid peering), *commenting on* Preserving the Open Internet, *supra*.

19 Barbara van Schewick, Network Neutrality and Quality of Service: What a Non-Discrimination Rule Should Look Like 54 box 18 (Stanford Law Sch. Ctr. for Internet and Soc'y, unnumbered working paper, 2012), *available at* http://cyberlaw.stanford.edu/files/publication/files/20120611-NetworkNeutrality_0.pdf. The layering principle prohibits network operators from either accessing or acting on information contained in higher-level protocols. Under van Schewick's proposal, network providers can access and analyze this information, but cannot act on it. *Id.* In other words, they can look, but they cannot touch.

some observations about implicit tradeoffs in layering that, despite being made with respect to an earlier architecture and concerns that did not fully mature, still reflect some basic insights. Clark recognized that the centrality of layering in the engineering literature "tends to suggest that layering is a fundamentally wonderful idea which should be a part of every consideration of protocols."[20] Such a perspective overlooks the fact that layering provides "both a benefit and a penalty."[21] While "[a] visible layer boundary, with a well specified interface, provides a form of isolation between two layers" that permits modifications to one layer without interfering with other layers, "a firm layer boundary almost inevitably leads to inefficient operation."[22] Hiding much of the technical complexity behind layer boundaries prevents other layers from taking advantage of the full functionality of the underlying technology, which in turn increases the resources needed to perform the desired task.[23] Thus, the "tempt[ation] to think that a layer boundary . . . is in fact the proper boundary to use in modularizing the implementation" is "a potential snare."[24] The tradeoff between generality and efficiency is "rarely acknowledged in the computing literature," however.[25] A small but important body of work exists in the engineering literature exploring how protocol layering can harm network performance.[26]

---

20 RFC 817, *supra* note 5, at 24.

21 *Id.* at 16.

22 *Id.*

23 *Id.* at 17. Clark continues:

> In fact, layering is a mixed blessing. Clearly, a layer interface is necessary whenever more than one client of a particular layer is to be allowed to use that same layer. But an interface, precisely because it is fixed, inevitably leads to a lack of complete understanding as to what one layer wishes to obtain from another. This has to lead to inefficiency.

Id. at 24.

24 *Id.*; *see also* COMER, *supra* note 4, at 169 (observing that "strict layering can be extremely inefficient" by sometimes preventing a layer "from optimizing transfers"); RFC 871, *supra* note 3, at 11 (listing efficiency concerns arising from layering).

25 Jean-François Blanchette, *A Material History of Bits*, 62 J. AM. SOC'Y FOR INFO. SCI. & TECH. 1042, 1047 (2011).

26 For classic analyses of the potential downsides of protocol layering, see Greg Chesson, *Protocol Engine Design*, 1987 PROC. USENIX SUMMER CONF. 209; Jon Crowcroft et al., *Is Layering Harmful?*, IEEE NETWORK, Jan. 1992, at 20, 23-24; David Tennenhouse, *Layered Multiplexing Considered Harmful*, *in* PROTOCOLS FOR HIGH-SPEED NETWORKS 143, 144-45 (H. Rudin & R. Williamson eds., 1989); David D. Clark & David L. Tennenhouse, *Architectural Considerations for a New Generation of Protocols*, COMPUTER COMM. REV., Sept. 1990, at 200, 205-07; and Randy Bush & David Meyer, *Some Internet Architectural Guidelines and Philosophy* 7-12 (IETF Network Working Grp. RFC No. 3439, 2002) [hereinafter RFC 3439], *available at* http://tools.ietf.org/pdf/rfc3439.

In addition to its impact on efficiency, protocol layering can also have an adverse impact on innovation that is often overlooked. Although protocol layering promotes innovations that are consistent with the architecture, at the same time it impedes innovations that are inconsistent with the design hierarchy.[27] Moreover, any changes that require a reconfiguration of the design hierarchy require coordinating with actors operating at the layers both above and below the locus of the innovation, which makes such innovations all the more difficult to implement.

The existing policy debate based on protocol layering largely ignores the extent to which it is something of a mixed blessing from the standpoint of innovation. On the one hand, to yield any benefits, an architecture must be relatively stable and change only rarely.[28] Indeed, the natural temptation for computer scientists to optimize for particular applications[29] or to redesign the entire system from scratch means that any calls for a fundamental redesign of the entire architecture should be greeted with a healthy amount of skepticism.[30]

On the other hand, to say architectural changes should be infrequent is not to say that they should never occur. Even the strongest proponents of the layered model recognize that the architecture can and should evolve over time.[31] Major changes transforming the Internet environment—including the growing heterogeneity of end users, the advent of Internet-based video and cloud computing, and the emergence of wireless broadband and the smartphone operating system as the relevant platforms—raise the possibility that circumstances may have changed sufficiently to justify a change in the architecture.[32] Indeed, the emergence of wireless broadband as an important medium of transmission has spawned a growing literature on cross-layer design exploring new architectures that deviate from the

---

27 *See infra* notes 76-78 and accompanying text.

28 *See* Blanchette, *supra* note 25, at 1054.

29 D.L. Parnas, *Information Distribution Aspects of Design Methodology*, 1 INFO. PROCESSING 71: PROC. IFIP CONG. 71, at 339, 342 (1972).

30 *See* Blanchette, *supra* note 25, at 1055 (noting that developers "often fantasize" about a "clean slate" but that effective "infrastructural change proceeds just as much through improvisation, bricolage, and drift").

31 *See* Solum & Chung, *supra* note 7, at 865 ("As the Internet evolves, it is possible that superior architectures may be conceived."); Werbach, *supra* note 7, at 66 ("If the physical and logical infrastructure layers in the relevant markets were sufficiently competitive, ILECs would not be able to gain unfair advantage over competitors at the application and content layers."); Whitt, *supra* note 7, at 619 ("At its core, the layers principle is a pragmatic tool . . . [and] policymakers should take care not to enshrine it as either definitive or dispositive in each and every situation.").

32 *See* CHRISTOPHER S. YOO, THE DYNAMIC INTERNET 4 (2012) ("The dramatic shift in Internet usage suggests that its founding architectural principles . . . may no longer be appropriate today.").

existing layered stack.[33] While cross-layer design has many proponents, it has proven controversial, with many other engineers contending that it is an unnecessary deviation.[34]

These technological developments are leading both the engineering community and policymakers to try to determine the optimal rate of architectural change and to develop analytical frameworks for recognizing the circumstances under which it should occur. This requires a better understanding of the tradeoffs implicit in protocol layering. Absent such an understanding, regulations that embody a particular vision of the architecture risk effectively locking the existing implementation into place without taking into account the contingencies that may necessitate changes in the underlying architecture. Policymakers should understand the underlying tradeoffs and risks before enshrining any architecture into law, no matter how successful it has proven in the past. Without such an understanding, the invocation of engineering concepts can provide a veneer of technological legitimacy to what is more properly regarded as a normative claim.[35] It also prevents the balancing of other considerations that may favor greater integration and coordination.[36]

This Article seeks to fill that gap. Part I introduces the basic principles underlying protocol layering. Part II reviews the evolution of the layered architecture underlying the Internet. Part III critiques the conceptions of protocol layering in the legal literature, including attempts to invoke it as a guide to competition policy. Part IV identifies developments that are putting pressure on the current layered stack. The conclusion offers an assessment of layering's affirmative implications, determining that, until policymakers understand the principles underlying layering, the invocation

---

33 *See infra* note 243 and accompanying text.

34 *See, e.g.*, Vikas Kawadia & P.R. Kumar, *A Cautionary Perspective on Cross-Layer Design*, IEEE WIRELESS COMM., Feb. 2005, at 3, 7-8 (providing examples of negative and unintended system performance consequences that could result from cross-layer design).

35 *See* Marjory S. Blumenthal, *End-to-End and Subsequent Paradigms*, 2002 L. REV. M.S.U.-D.C.L. 709, 710 ("Although the embrace of engineering principles . . . appears to impart a legitimacy to certain kinds of advocacy, that advocacy reaches beyond the engineering to the ideology long associated with the Internet.").

36 *See* COMM. ON THE INTERNET IN THE EVOLVING INFO. INFRASTRUCTURE ET AL., *supra* note 2, at 145-46 (considering the benefits of vertically integrated companies and possible coordination benefits from combining applications and content); Timothy F. Bresnahan & M. Trajtenberg, *General Purpose Technologies: "Engines of Growth"?*, 65 J. ECONOMETRICS 83, 94-96 (1995) (showing how greater vertical coordination can help internalize positive externalities generated by general purpose technologies); David J. Teece, *Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy*, 15 RES. POL'Y 285, 288, 291-92 (1986) (suggesting that innovation requires close coordination of "complementary assets" to protect against inequalities in bargaining power and to encourage relationship-specific investments).

of engineering principles will obscure rather than promote sound policy analysis.

I. THE CONCEPTUAL UNDERPINNINGS OF PROTOCOL LAYERING

As described above, understanding the relative merits of a layered architecture as well as the circumstances under which it can and should change requires understanding the theory underlying the principle. Because layering is widely recognized as a particular form of modularity,[37] Section A offers a basic introduction to modularity theory. Section B moves past modularity in general to discuss protocol layering in particular. Section C analyzes the advantages and disadvantages to layering suggested by the theory.

To deal first with some preliminary matters of nomenclature, the computers with which end users connect to the Internet are called *hosts*, and the various programs running on any particular host comprise a number of *processes*. Because the Internet is a network of networks, some of these computers are located within a network, while others serve as gateways between networks. Nodes that route traffic within a network are typically called *switches*, while nodes that route traffic between networks are called *routers*.[38]

A particular convention for formatting, interpreting, and reacting to a communication is called a *protocol*.[39] The functions of a protocol are well illustrated by the protocol used in the traditional postal system. Effective transmission of the mail requires agreement between those sending and carrying mail as to where to locate the relevant information. By convention, the return address for letters is located in the upper left-hand corner, the postage in the upper right-hand corner, and the destination address in the middle. The convention for postcards is somewhat different. For many postcards, the return address, the destination address, and the postage are all located on the right-hand side of the card.

Mail systems must also agree on how to interpret the content of the information conveyed in these locations, such as the significance of particular ZIP codes, street addresses, and bar codes. Standardizing where important

---

37 For discussions of layering as a unique form of modularity, see VAN SCHEWICK, *supra* note 7, at 46, 379; Blanchette, *supra* note 25, at 1046; Crowcroft et al., *supra* note 26, at 23; RFC 871, *supra* note 3, at 7; Douglas C. Sicker & Lisa Blumensaadt, *Misunderstanding the Layered Model(s)*, 4 J. ON TELECOMM. & HIGH TECH. L. 299, 305 (2006); Philip J. Weiser, *Law and Information Platforms*, 1 J. ON TELECOMM. & HIGH TECH. L. 1, 4 (2002); Werbach, *supra* note 7, at 59 n.85; Wu, *supra* note 7, at 1190.

38 PETERSON & DAVIE, *supra* note 4, at 253.

39 KUROSE & ROSS, *supra* note 4, at 9.

information is located greatly facilitates the mail system's ability to process the mail while simultaneously flagging for the carrier what information may safely be ignored, such as the personal message written on the left-hand side of post cards. American conventions are by no means the only feasible formats. Indeed, many mail systems in Asia format addresses in the reverse order, with the state being listed first, followed by the city, and then the street address.[40] Despite these differences, these systems will remain interoperable so long as each mail system is able to interpret and convert addresses in the other format.

In addition, mail systems must share an understanding of how to handle particular situations. Some of these features control the tasks internal to one actor, such as how they should treat hold orders and change-of-address notices. The actors must also agree on what happens if the post office attempting to deliver a piece of mail cannot locate the destination address. For first-class mail, post offices return undeliverable mail to the location listed in the return address. For lower classes of mail, however, the post office simply discards the mail.

Understanding how other actors are expected to behave under particular circumstances provides each actor with a guide to interpreting and reacting to what has happened. To use the example described above, if a piece of first-class mail is not returned to sender, the person sending it may assume that it was successfully delivered.[41] The different treatment of lower classes of mail means that senders cannot infer anything from the fact that an article sent via a lower class was not returned. And at a far end of the spectrum, some classes of service require the post office to send a confirmation of delivery back to the sender once mail is delivered, whereas most classes of mail do not. If the sender knows that the post office is supposed to send a delivery confirmation, it may regard the failure to receive a confirmation within a reasonable amount of time as an indication that the letter never arrived. Based on this inference, the sender can take whatever action it deems appropriate, whether that means resending the letter, choosing a different mode of communication, or abandoning attempts to convey the information altogether.

---

40 *Appendix V International Address Formats*, MICROSOFT DEVELOPER NETWORK, http://msdn.microsoft.com/en-us/library/cc195167.aspx (last visited Apr. 4, 2013).

41 This inference is not conclusive, as it is always possible that the mail was lost or destroyed. Whether the sender should take additional means to verify delivery depends on the likelihood of an adverse event as well as the value of what was sent.

A. *Modularity Theory*

Modularity is one of the principal mechanisms for managing complex systems.[42] When the tasks constituting a system are highly interdependent, changes to one task will necessarily affect a wide range of other tasks. Anyone seeking to change one part of the system must then analyze the effect of that change on all of the other interdependent tasks. Modularity seeks to reduce the number of interdependencies that must be analyzed by identifying which tasks are highly interdependent and which ones are not. Highly interdependent tasks are grouped within modules.[43] The points of relatively low interdependence become the natural locus for interfaces between modules.[44]

Limiting the number of interdependencies between modules greatly simplifies the number of permutations that must be tested in order to verify that a change to one module is not adversely affecting the complex system.[45] Predefining the way in which different modules interact with one another also reduces coordination costs.[46] Modular architectures ensure that other modules take into account only those interdependencies permitted by the design through a technique known as *information hiding*.[47] Information needed to support the interdependencies with other modules contemplated by the architectural design must be included in the interface; information relating solely to interdependencies within the module is omitted from the interface and thus hidden from other modules.[48] With respect to these hidden parameters, other blocks may regard this information as a "black box."[49]

---

42 The discussion that follows is based on the more comprehensive discussion of modularity theory in Christopher S. Yoo, Modularity and Internet Policy 5-12 (Sept. 23, 2012) (paper presented at the 40th Telecomms. Policy Research Conf.), *available at* http://ssrn.com/abstract=2032221.

43 *See id.* at 8 ("A well designed module . . . is a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units." (internal quotation marks omitted)).

44 *Id.* at 7.

45 *Id.* at 8, 13 (citing Edsger W. Dijkstra, *The Structure of the "THE"-Multiprogramming System*, 11 COMM. ACM 341, 343 (1968)).

46 *See id.* at 14 ("[M]odularity facilitates the division of labor by enabling autonomous innovation that requires little coordination among modules." (citing Richard N. Langlois & Paul L. Robertson, *Networks and Innovation in a Modular System: Lessons from the Microcomputer and Stereo Component Industries*, 21 RES. POL'Y 297, 302 (1992))).

47 *Id.* at 10 (citing Parnas, *supra* note 29, at 342).

48 *Id.* at 11 (citing 1 CARLISS Y. BALDWIN & KIM B. CLARK, DESIGN RULES: THE POWER OF MODULARITY 72-73 (2000)).

49 *Id.* at 11 (quoting BALDWIN & CLARK, *supra* note 48, at 91).

Consider, for example, the advent of printers using USB ports. Certain aspects of printer design are intimately tied to whether the printer is a laser or an inkjet printer. The creation of a modular interface allows computers connected to those printers to ignore almost all of the details about how any particular printer operates. As long as the computer provides the data in the correct format, the printer should operate without any problems. Conversely, as long as the printer remains ready to process any data submitted in the correct format, the printer's design can be changed without having any impact on the overall system.

Despite the design architect's best efforts, modular systems can rarely be defined a priori. Many aspects of how tasks interact with one another can be understood only after the architect experiments with different solutions.[50] Consequently, modular systems more often result from "improvisation, bricolage, and drift" than from some theoretical conception of the ideal architecture.[51]

B. *Peer Communication and Encapsulation*

Layering represents a very particular form of modularity, in which different parts of the overall system are arranged into parallel hierarchies. In the typical Internet transaction, a process generates a message and transfers it to the operating system running on the host. The operating system divides the message into packets configured for the Internet and hands them off to the first-hop router of a communications network. The sending communications network will convey these packets to the receiving communications network, which in turn passes them to the receiving host's operating system. The operating system then passes them to the process running on the receiving host.

The type of modularity enforced by layering has several distinct characteristics. First, the modules are arranged into a series of client-server relationships, where "each layer is a server to the layer above, and a client to the layer below."[52] Second, this arrangement is strictly hierarchical; every

---

50 *See* BALDWIN & CLARK, *supra* note 48, at 254 ("Given such a high degree of complexity, it simply is not possible for designers to know enough about the system to eliminate all uncertainty. Thus each new design is fundamentally an experiment."); Sendil K. Ethiraj & Daniel Levinthal, *Modularity and Innovation in Complex Systems*, 50 MGMT. SCI. 159, 172 (2004) (noting that designers "lack omniscience").

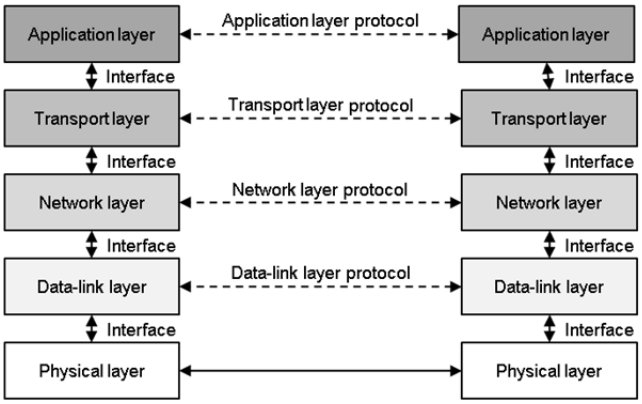
51 Blanchette, *supra* note 25, at 1055.

52 *Id.* at 1046.

layer interacts exclusively with the layers above and below with respect to every communication without bypassing either.[53]

Third, layering differs from other modular schemes in its focus on establishing communication between peers.[54] Unlike the example of the USB port given above, in which a personal computer could establish a connection directly with a printer, layered architectures require that connections be established only between parallel elements in the hierarchy. For example, applications communicate with other applications; operating systems communicate with other operating systems; routers communicate with other routers.

Figure 1: Layering as Peer Communication[55]



Layering ensures that peers communicate only with peers operating at the same level through a practice known as *encapsulation*. Under this approach, each layer takes the data packet provided by the layer above, extracts all of the information that the next layer will need to perform its functions, places that information into a new packet's header, and then places the entirety of the packet it received from the higher layer in the payload of the new packet.[56] Since layering requires that each layer examine only the information contained in the header and prohibits it from examining

---

53 *See* RFC 871, *supra* note 3, at 9 (explaining that protocols operate in a hierarchy with a strict "chain of command").

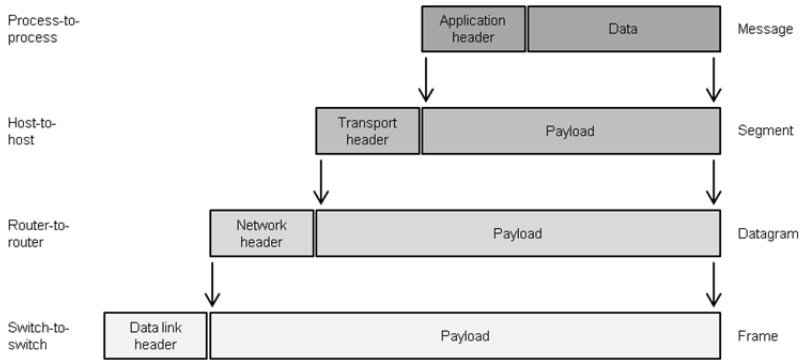
54 *See* TANENBAUM, *supra* note 4, at 26-30 (explaining how layering protocols facilitate communication between peers).

55 *See id.* at 27 fig.1-13.

56 KUROSE & ROSS, *supra* note 4, at 55-56.

or altering any of the information contained in the payload, the content of the packet remains hidden.[57]

Figure 2: Layering as Encapsulation[58]



An analogy to the postal system is illustrative.[59] Suppose that Vint wants to send a memo to a colleague named Bob, who is working in a different branch office of the same company. Vint takes the memo and places it in an interoffice envelope with Bob's name and department written on the front. The company's internal mail system brings the interoffice envelope to the mail room and, without opening it, places the entire contents inside a larger envelope, using the information on the outside of the interoffice envelope to inscribe the mailing address, formatted in the manner required by the U.S. Postal System, on the outside of the envelope. The mail room then gives the postal envelope to a U.S. Postal Service letter carrier who places it inside a shipping container with a bar code on its outside. After the shipping container arrives at a post office in the city where Bob's branch office is located, the post office opens the shipping container, removes the postal envelope, reads the address on its outside, and delivers it to the branch office. The mail room of the branch office opens the postal envelope, reads the address on the interoffice envelope, and delivers it to Bob. Finally, Bob opens the interoffice envelope and reads the content

57 *See* TANENBAUM, *supra* note 4, at 448 (describing "the most fundamental rule of protocol layering: layer k may not make any assumptions about what layer $k + 1$ has put into the payload field").

58 For a similar figure showing an example of encapsulation of an IP datagram, see COMER, *supra* note 4, at 78.

59 The example is adapted from TANENBAUM, *supra* note 4, at 56. *See also* VAN SCHEWICK, supra note 7.

of the message from Vint—a message no one but the two of them had any occasion to see.

Note the key features of this system. Each of the peers receives the exact same communication. After the memo is removed from the interoffice mail envelope, Bob receives the exact message sent by Vint. Once the U.S. mail envelope is opened, the receiving mail room receives precisely the same interoffice mail envelope and contents as the one sent by the sending mail room. After the shipping container is unpacked, the two post offices exchange precisely the same envelope containing the same message as well. The fact that each actor encapsulates the entirety of the communication in a larger envelope ensures that at each step the original message can be recovered unaltered. In addition, refusing to look inside the envelope until it is de-encapsulated ensures that lower-layer protocols cannot make any assumptions about objects handed to them by the upper-layer protocols.

Using an example more closely related to the Internet, consider what occurs when an end user sends an email message. The end user's email client, such as Microsoft Outlook, first extracts the source and destination addresses and places that information in the Simple Mail Transfer Protocol (SMTP) header, which is the general format used for email. Then the email client encapsulates the email in an SMTP message by placing the entire email in the payload of the packet and passes it to the next layer, the transport layer. The transport layer, in turn, reads the information it needs from the SMTP header and places that information in the Transmission Control Protocol (TCP) header, places the segment in the payload, and passes the segment to the network layer. The network layer reads the source and destination addresses from the TCP header, places the necessary information in the Internet Protocol (IP) header, encapsulates the segment into an IP datagram, and passes the datagram to the data-link layer. The data-link layer reads the information it needs from the IP header, places the relevant information in the frame header, and encapsulates the datagram into a data-link layer frame. The data-link layer will use the header information to move the frame hop by hop.

As long as each hop remains within both the same network and the same data-link technology, there is no need to de-encapsulate the frame to reveal information. But when the hop reaches a gateway to another network, the technology may change. Because moving to another network may involve shifting to a different data-link technology, the gateway de-encapsulates the data-link layer frame and passes it to the next network as an IP datagram. The next network will re-encapsulate it in a frame appropriate for its data-link technology and will pass it along in this manner until it reaches the border of another network, when the de-encapsulation process begins again.

Eventually, the packet will reach the receiving host. The host will de-encapsulate the data-link header and pass the IP datagram to the network layer. The network layer will remove the IP header and pass the segment to the transport layer. The transport layer will strip off the TCP header and pass the SMTP message to the application layer. Finally, the application layer will strip off the SMTP header and pass the email to the program that the receiving end user is using to read the message.

Although it is the email clients that are exchanging messages, no direct transfers of data occur between them. Instead, each layer passes the data and control information down through the stack until it reaches the physical layer, which provides the only direct connection. After the communication arrives at the receiving host, the data passes up through the layered stack until it reaches the same layer as the peer sending the communication, at which point it can act on the same information. For this reason, all connections above the physical layer are simply virtual.[60]

It is often said that layered architectures ensure that every entity in the receiving hierarchy receives the exact same object as the one sent by its peer in the sending hierarchy.[61] As is the case with almost every generalized network engineering principle, there are important exceptions. To return to the postal example, one post office may use a postmark to cancel a stamp, in which case the object received will differ in a small way from the object sent. Similar changes occur in the email example. For example, intermediate mail transfer agents will include information in email messages noting that they were received. In addition, the IP header contains a counter that is reduced by one every time a packet traverses a hop, with routers ceasing to route a packet further when the counter hits zero to ensure that packets do not wander around the Internet forever. Nonetheless, the generalization remains a useful concept as a reference model.

A related principle of layering is that lower layers can make no assumptions about the nature of the communications in the upper layers.[62] Because all of the information that any particular layer needs is supposed to be included in the packet's header, no layer is expected to look inside the payload of any packet that it processes. Any attempt to do so is regarded as

---

60 TANENBAUM, *supra* note 4, at 27, 30.

61 *See, e.g.*, COMER, *supra* note 4, at 164 ("Layered protocols are designed so that layer *n* at the destination receives exactly the same object sent by layer *n* at the source."); *id.* at 165 ("Thus, the layering principle states that the packet received by the transport layer at the ultimate destination is identical to the packet sent by the transport layer at the original source.").

62 TANENBAUM, *supra* note 4, at 448; RFC 871, *supra* note 3, at 16.

deep packet inspection.[63] However, many widely deployed technologies represent exceptions to this general principle against deep packet inspection.

C. *The Tradeoffs Inherent in Protocol Layering*

Protocol layering yields a number of substantial benefits.[64] Like other modular systems, layering breaks down complex systems into subparts small enough for a single person or a small group of people to describe and understand.[65] In addition, by predetermining how each layer interacts with other layers, layering allows for the development of individual subparts in isolation without having to understand the architecture as a whole.[66] Only those interactions recognized by the design need be taken into account; any details associated with interdependencies encapsulated within modules can be safely ignored.[67] Even more importantly, cabining the number of possible interdependencies between modules reduces the combinatorial explosion of system variants that must be tested when changes are made to the system.[68]

Segregating different functions and defining how they interact with one another also permit development of each layer to proceed independently and limit the impact that changes to individual components have on the system as a whole.[69] Eliminating the need for direct coordination facilitates the division of labor across workgroups[70] and firms.[71] Furthermore, it promotes competition by creating entry points for new firms.[72]

---

63 Under a strict layering principle, each layer is allowed to invoke the services only of the layer located immediately below it. More relaxed versions permit a layer to invoke the services of any layer below it even if it is not immediately adjacent to it. VAN SCHEWICK, *supra* note 7, at 47.

64 This discussion is based on Yoo, *supra* note 42, at 12-17.

65 *See* Blanchette, *supra* note 25, at 1046; Herbert A. Simon, *The Architecture of Complexity*, 106 PROC. AM. PHIL. SOC'Y 467, 474, 477 (1962).

66 *See* D.L. Parnas et al., *The Modular Structure of Complex Systems*, PROC. 7TH INT'L CONF. ON SOFTWARE ENG'G 408, 410 (1984).

67 *See* BALDWIN & CLARK, *supra* note 48, at 91 (noting that modularity "accommodates uncertainty" in the design process).

68 *Id.* at 273-75; *see also* Dijkstra, *supra* note 45, at 344 (concluding that a hierarchical structure prevented the number of potential states from "explod[ing] to such a height that exhaustive testing would have been an illusion").

69 D.L. Parnas, *On the Criteria to Be Used in Decomposing Systems into Modules*, 15 COMM. ACM 1053, 1054 (1972); Simon, *supra* note 65, at 477.

70 *See* Parnas, *supra* note 69, at 1054 ("[D]evelopment time should be shortened because separate groups would work on each module with little need for communication . . . ."); Langlois & Robertson, *supra* note 46, at 301-02 (discussing the benefits of division of labor into different groups).

Layering can also hasten innovation by allowing experiments with different solutions in different layers to proceed in parallel.[73] In addition, protocol layering accommodates uncertainty by making it easy to incorporate subsequent improvements into the existing system.[74] As real-option theory indicates, the ability to postpone such choices can be an important source of value, provided that the interfaces are clearly defined and remain stable.[75]

Like any modular system, protocol layering embodies a precommitment about the types of information permitted to pass between modules. Preventing adjacent modules from acting on certain types of information reduces the complexity of the system by constraining the number of interdependencies that must be taken into account. At the same time, prohibiting modules from taking into account all of the possible information inevitably limits both the efficiency and functionality of the resulting system.

In terms of efficiency, the generality of the layers necessarily means that certain customized solutions tailored to particular situations must be foregone.[76] The layers and the interfaces connecting them predefine and limit the way those layers interact.[77]

---

71 *See* Carliss Y. Baldwin & Kim B. Clark, *Managing in an Age of Modularity*, HARV. BUS. REV., Sept.–Oct. 1997, at 84, 85 ("Different companies can take responsibility for separate modules and be confident that a reliable product will arise from their collective efforts.").

72 Langlois & Robertson, *supra* note 70, at 301.

73 Baldwin & Clark, *supra* note 71, at 91; *see also* Karl Ulrich, *The Role of Product Architecture in the Manufacturing Firm*, 24 RES. POL'Y 419, 435 (1995) ("For the modular architecture, detailed design of each component can proceed almost independently and in parallel.").

74 Baldwin & Clark, *supra* note 71, at 91.

75 *See* BALDWIN & CLARK, *supra* note 48, at 234-37 (applying real option theory to show how modularity permits industries to postpone having to commit to any particular technological solution); *id.* at 284-93 (discussing the option value of hidden modules).

76 *See* RFC 817, *supra* note 5, at 16, 24 ("[A]n interface, precisely because it is fixed, inevitably leads to a lack of complete understanding as to what one layer wishes to obtain from another. This has to lead to inefficiency."); RFC 871, *supra* note 3, at 20-22 (discussing the tradeoff inherent in the fact that fixed layers lead to less design flexibility). Blanchette discusses the loss of efficiency resulting from generalization:

> [T]he most efficient programs are hand-tailored, providing no generalization whatsoever; conversely, highly general abstractions will result in significant loss in efficiency. This is because the specification of an abstraction (the interface) general enough to accommodate a wide range of implementations necessarily involves trade-offs, between the freedom that the abstraction provides and the efficiency of possible implementation.

Blanchette, *supra* note 25, at 1046-47 (internal quotation marks omitted).

77 *See* TANENBAUM, *supra* note 4, at 27 ("The interface defines which primitive operations and services the lower layer makes available to the upper one."); Parnas, *supra* note 29, at 339 ("The connections between modules are the assumptions which the modules make about each other.").

Engineers have long recognized that hiding information can both harm and promote innovation. Specifically, innovation that depends upon the sharing of particular information cannot proceed if that information is held in another layer and if the particular form of modularity imposed by the architecture does not permit that information to pass through the protocol stack. The aforementioned information hiding means that layering "hide[s] vital information that lower layers may need to optimize their performance" and requires "that the optimization of each layer . . . be done separately," which can "conflict with efficient implementation of data manipulation functions."[78]

In other words, design hierarchies represent something of a mixed blessing from the standpoint of innovation.[79] On the one hand, they facilitate innovation that is consistent with the hierarchy. Indeed, the existing layered architecture has proven incredibly robust. On the other hand, predetermining the locus of the interfaces and the information that can pass between layers discourages innovations that are inconsistent with the hierarchy.[80]

Protocol layering also limits the network's ability to evolve. Any system of modularity necessarily envisions that change to the basic architecture will occur relatively slowly.[81] For the most part, the stability of the architecture yields benefits. Predefining the interactions between particular components makes it easier to make changes to individual components without disturbing the system as a whole. Without a high degree of stability, actors could not innovate in individual layers with any confidence.[82]

At the same time, however, this stability can impede the network's ability to evolve into a fundamentally different architecture. Economic theory has long recognized that having an installed base in a network industry can

---

78 RFC 3439, *supra* note 26, at 7-8; *see also* Crowcroft et al., *supra* note 26, at 23 ("[T]he flip side to modularization and data-hiding is that tuning the efficiency of the data path for transfer of data becomes difficult . . . . Vertical partitioning emphasises the discontinuities in the data path, which then obstruct the application from receiving the quality of service it requires.").

79 Christopher S. Yoo, *Product Life Cycle Theory and the Maturation of the Internet*, 104 NW. U. L. REV. 641, 655-56 (2010).

80 *See* Kim B. Clark, *The Interaction of Design Hierarchies and Market Concepts in Technological Evolution*, 14 RES. POL'Y 235, 246 (1985) (using automobiles as an example and noting that "[o]nce choices about core concepts in engines were established, innovative effort moved down into subsidiary parameters").

81 *See* COMM. ON THE INTERNET IN THE EVOLVING INFO. INFRASTRUCTURE, *supra* note 2, at 38 (arguing that innovation at the "center of the network" is slow because building new features into the existing network requires coordinating the actions of multiple developers); Parnas et al., *supra* note 66, at 409 (explaining that changes to modular interfaces should be limited to changes that are unlikely to be needed).

82 *See* Blanchette, *supra* note 25, at 1054.

lead to technological lock-in.[83] Placing such considerations in a design hierarchy amplifies this effect, as any change to the architecture requires parallel changes in the levels both above and below.[84] This process provides a new perspective on what is sometimes termed "Internet time."[85] Although innovation within layers proceeds at breakneck speeds, innovation in the architecture proceeds at a glacial pace.[86]

The sociology of technology provides another drag on innovation.[87] The emergence of a design hierarchy establishes a technical agenda for a product's development, directing further innovation along particular lines.[88] Established technological paradigms guide research along innovation avenues consistent with the incumbent design hierarchy.[89] These paradigms become ingrained in the institutional filters that organizations use to manage information, which tends to further reinforce the status quo.[90] The engineering literature is replete with complaints that the ossification of the Internet is preventing the architecture from evolving.[91] The technological paradigm established by a design hierarchy even extends into the education system. Indeed, some computer science researchers have reportedly expressed reluctance to pursue research inconsistent with the TCP/IP stack.[92]

---

83 *See, e.g.*, Joseph Farrell & Garth Saloner, *Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation*, 76 AM. ECON. REV. 940, 942 (1986) (finding that the existence of a large installed base can cause "excess inertia" and make new technology less likely to be adopted).

84 Yoo, *supra* note 79, at 656.

85 *See generally, e.g.*, MICHAEL A. CUSUMANO & DAVID B. YOFFIE, COMPETING ON INTERNET TIME: LESSONS FROM NETSCAPE AND ITS BATTLE WITH MICROSOFT (1998) (discussing and investigating the pace of competition in the Internet age).

86 *See* Blanchette, *supra* note 25, at 1054 (arguing that, contrary to popular conceptions, computing infrastructure evolves slowly).

87 Yoo, *supra* note 79, at 651-55.

88 Blanchette, *supra* note 25, at 1054.

89 *See* Giovanni Dosi, *Technological Paradigms and Technological Trajectories*, 11 RES. POL'Y 147, 152 (1982); Devendra Sahal, *Technological Guideposts and Innovation Avenues*, 14 RES. POL'Y 61, 71, 78-79 (1985).

90 Philip Anderson & Michael L. Tushman, *Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change*, 35 ADMIN. SCI. Q. 604, 618 (1990).

91 *See, e.g.*, Paul Laskowski & John Chuang, A Leap of Faith? From Large-Scale Testbed to the Global Internet 2 (Sept. 2009) (paper presented at the 37th Telecomms. Policy Research Conf.) (collecting sources indicating that "[t]he predominant view from within the research community is that the internet is incapable of significant architectural change" and that "the network architecture is described as stagnant, even ossified").

92 *See* Vint Cerf et al., FIND Observer Panel Report 2 (Apr. 9, 2009) (unpublished manuscript), *available at* http://www.nets-find.net/FIND_report_final.pdf (citing a concern among some faculty that "architectural research that is not incremental might be considered 'too risky'").

Technological and economic changes can often pressure high-tech industries to evolve toward a fundamentally different architecture.[93] Examples include the desktop PC's absorption of functions that used to be provided by standalone peripheral devices (such as hard disks, modems, and WiFi cards[94]) and the advent of last-mile broadband networks (such as DSL and cable modem systems), both of which undercut the rationale for a standalone regional ISP.[95] In any layered stack, however, the danger is that modularity may inhibit systemic innovation by creating economic pressures and organizational structures that lock the existing interfaces into place. It is thus too simplistic to suggest that modularity and protocol layering present a simple tradeoff between short-run efficiency and long-run evolvability.[96] Indeed, promoting evolvability falls on both the modular and nonmodular sides of the balance. Moreover, although the existing regulatory regime is often criticized as creating vertical, technology-specific silos that ignore the extent to which different means of transmission compete with one another, protocol layering risks creating silos of its own.[97]

Whether any particular architecture strikes the correct balance depends on context. Modular structures reflect the number and location of task interdependencies as well as a particular vision of which interdependencies multiple layers should be permitted to take into account. Changes in technology and end-user demand for network services, however, can cause the nature and relative importance of particular interdependencies to

---

93 *See* Carliss Y. Baldwin, *Where Do Transactions Come from? Modularity, Transactions, and the Boundaries of Firms*, 17 INDUS. & CORP. CHANGE 155, 180 (2008) (noting that "there is no process of technological determinism at work driving the task network toward ever-higher levels of modularity" and that changes in strategies, knowledge, and technologies can cause "task networks to become more integral (i.e., less modular) over time"); Michael G. Jacobides & Sidney G. Winter, *The Co-Evolution of Capabilities and Transaction Costs: Explaining the Institutional Structure of Production*, 26 STRATEGIC MGMT. J. 395, 405 (2005) (noting how the emergence of new productive structures and new knowledge bases can cause the pattern of increasing specialization and vertical disintegration to reverse).

94 *See, e.g.*, Transamerica Computer Co. v. IBM Corp., 698 F.2d 1377, 1382-83 (9th Cir. 1983) (discussing the increasing integration between the CPU and certain peripherals); Cal. Computer Prods., Inc. v. IBM Corp., 613 F.2d 727, 743-44 (9th Cir. 1979) (discussing how the new IBM computer "integrated the disk control function into the CPU"); ILC Peripherals Leasing Corp. v. IBM Corp., 448 F. Supp. 228, 231-32 (N.D. Cal. 1978) (describing how IBM abandoned removable disk drives in favor of nonremovable drives with the head-disk assembly integrated into the computer itself that provided greater storage capacity), *aff'd sub nom.* Memorex Corp. v. IBM Corp., 636 F.2d 1188 (9th Cir. 1980); Telex Corp. v. IBM Corp., 367 F. Supp. 258, 342 (N.D. Okla. 1973) (describing how IBM integrated memory and control units into its CPUs), *rev'd on other grounds*, 510 F.2d 894 (10th Cir. 1975).

95 Yoo, *supra* note 2, at 33-34.

96 *See* VAN SCHEWICK, *supra* note 7, at 370-71.

97 *See* Jacobides & Winter, *supra* note 93, at 404.

change over time. These changing priorities pressure the prevailing modular architecture to change. The intensity and the stability of the demands being placed on the network may also place a higher premium on efficiency.

Indeed, an IETF document known as RFC 1958, which is often cited by policy advocates as laying out the basic architectural principles underlying the Internet,[98] recognized that "fundamentally new requirements might lead to a fundamentally new protocol."[99] RFC 1958 continues:

> In searching for Internet architectural principles, we must remember that technical change is continuous in the information technology industry . . . In this environment, some architectural principles inevitably change. Principles that seemed inviolable a few years ago are deprecated today. Principles that seem sacred today will be deprecated tomorrow. The principle of constant change is the only principle of the Internet that should survive indefinitely.[100]

The document thus squarely rejected the existence of any "dogma about how Internet protocols should be designed."[101]

The question, then, is not whether any particular implementation of protocol layering should ever change. It surely will eventually, if infrequently. Instead, the proper questions are better framed as trying to determine the optimal rate of architectural change and how to recognize the circumstances under which such changes are warranted. Policymakers should therefore develop heuristics for determining the circumstances under which such change might be appropriate. Given the key role that stability plays in fostering innovation and the costs of making such transitions, changes should occur infrequently and should be approached with considerable caution.[102]

---

98 *See, e.g.*, LESSIG, *supra* note 7, at 36; VAN SCHEWICK, *supra* note 7, at 105.

99 *Architectural Principles of the Internet* 3 (IETF Network Working Grp. RFC No. 1958, B. Carpenter ed., 2006) [hereinafter RFC 1958], *available at* http://tools.ietf.org/pdf/rfc1958.pdf.

100 *Id.* at 1.

101 *Id.* at 2.

102 Timothy F. Bresnahan, *New Modes of Competition: Implications for the Future Structure of the Computer Industry* ("It is neither desirable nor even possible to have frequent [pieces of radical change]; the costs of all that change are considerable."), *in* COMPETITION, INNOVATION AND THE MICROSOFT MONOPOLY: ANTITRUST IN THE DIGITAL MARKETPLACE 155, 161 (Jeffrey A. Eisenach & Thomas M. Lenard eds., 1999).

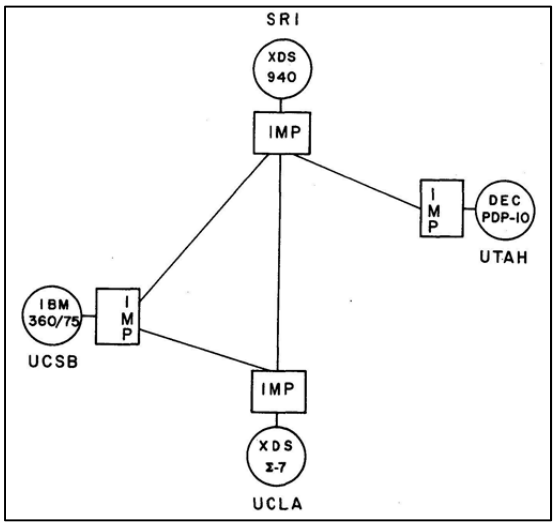
II. THE INTERNET AS AN EXAMPLE OF A LAYERED ARCHITECTURE

Perhaps the best way to understand the way protocol layering works and the advantages it provides is to examine the history of the ARPANET, which was operated by the Defense Department from 1967 to 1990 and which is widely recognized as the precursor to the modern Internet.[103]

A. *Connecting Heterogeneous Hosts*

One of the major challenges facing the ARPANET's designers was to find a way to interconnect the huge variety of computers that end users were using as hosts.[104] The scope of this problem is depicted in Figure 3, which depicts the first four ARPANET sites, each of which used a different type of computer as its host. These hosts operated on fundamentally incompatible design principles.[105]

Figure 3: The ARPANET as of December 1969[106]



103 *See* JANET ABBATE, INVENTING THE INTERNET 44, 195 (1999) (chronicling the ARPANET's initial funding, in 1967, and its final decommissioning, in 1990).

104 *Id.* at 51-53.

105 *Id.* at 48.

106 C. Stephen Carr et al., *HOST-HOST Communication Protocol in the ARPA Network*, 36 AFIPS CONF. PROC. 589, 590 fig.1 (1970).

The solution was to require each host to attach to the network through a minicomputer known as an Interface Message Processor (IMP), which served as that host's contact point with the rest of the network and buffered the network against the heterogeneity of the hosts.[107] All early IMPs were made by the same company from the same technology, ran the same software, and interconnected with one another through the same transmission protocol.[108] The fact that all IMPs were constructed from the same components and ran the same software simplified the challenge of managing the interactions between IMPs and helped organize them into a large, integrated system. Interestingly, the sending-and-receiving IMPs established a virtual circuit that confirmed that there was enough room at the destination to permit transmission of the message and maintained the flow below the maximum rate that the receiving host and the networks supporting the connection could accommodate.

Like the postal service, the ARPANET needed protocols for its network to function properly. One protocol dictated communication at the IMP level. So long as the hosts presented packets to the IMPs formatted in accordance with the protocol, IMPs could accept packets from any type of host without knowing anything about the technological principles on which that particular host was based.[109] At the same time, standardization had the reflexive property of relieving the hosts from needing any knowledge of the details of the underlying network.

The ARPANET's architects also needed to create host-to-host protocols, the most important of which was known as the Network Control Protocol (NCP).[110] Each host typically ran more than one program at the same time.[111] Thus, every sending host needed some way to sift through the return data streams and to route the incoming data to the correct process. The difficulty was that each host employed its own scheme for naming

---

107 *See* F.E. Heart et al., *The Interface Message Processor for the ARPA Computer Network*, 36 AFIPS CONF. PROC. 551, 551 (1970). Each IMP was initially connected to a single host, although the design permitted each IMP to serve up to four hosts. *Id.* at 553. The architecture was later redesigned to permit IMPs to serve larger numbers of hosts. *See* S.M. Ornstein et al., *The Terminal IMP for the ARPA Computer Network*, 40 AFIPS CONF. PROC. 243, 244-45 (1972).

108 All IMPs were built by Bolt, Beranek & Newman around a Honeywell DDP-516. *See* Heart et al., *supra* note 107, at 557-58.

109 ABBATE, *supra* note 103, at 52-53.

110 Carr et al., *supra* note 106; Alex McKenzie & Jon Postel, *Host-to-Host Protocol for the ARPANET*, *in* ARPANET PROTOCOL HANDBOOK 5, 11 (Elizabeth Feiner & Jonathan Postel eds., 1978).

111 *See* Carr et al., *supra* note 106, at 590-91 (describing the independent uses and time-sharing system governing the network's computers).

internal processes, and many of these naming schemes were incompatible with one another.[112]

NCP solved this problem by creating a standardized, intermediate scheme for naming processes called sockets, which were a series of virtual ports (represented by a socket number) that organized incoming and outgoing traffic.[113] To ensure that both the sending and receiving hosts knew which socket to use for a given session, the sending host would send a command, known as a request for connection, that specified the socket it would like to use for sending the traffic as well as the socket it would like to use for receiving the traffic associated with that connection.[114] The receiving host that accepted the connection would respond by identifying the sockets it planned to use to send and receive traffic.[115] This exchange became known as the opening handshake.[116] After the communication was completed, the sending and receiving hosts exchanged messages containing a close command, notifying both hosts that the sockets could be released and made available for other processes.[117] The use of sockets thus permitted each host to use its own system for mapping its internal scheme for naming processes onto particular sockets.[118]

NCP also played an important role in flow control and reliability. In order to prevent a flood of incoming messages from a faster source, or from the simultaneous arrival of flows from multiple sources that exceeded an IMP's or host's ability to process them, NCP prevented the IMP connected to the sending host from forwarding the next packet in a communication until the destination host's IMP successfully sent a return message, known as a Request for Next Message (RFNM). This return message confirmed that the previous packet had been successfully delivered and that the incoming link was now unblocked and available for additional traffic.[119]

---

112 *Id.* at 591.

113 *Id.* at 591-92.

114 *Id.* at 592-93.

115 The request for connection used to initiate a connection was called a sender-to-receiver (STR) command, whereas the message accepting a request for connection was known as a receiver-to-sender (RTS) command. McKenzie & Postel, *supra* note 110, at 15.

116 *See Requirements for Internet Hosts—Communication Layers* 93 (IETF Network Working Grp. RFC No. 1122, R. Braden ed., 1989) [hereinafter RFC 1122], *available at* http://tools.ietf.org/pdf/rfc1122 (describing a connection attempt as a "three-way handshake").

117 McKenzie & Postel, *supra* note 110, at 17.

118 Carr et al., *supra* note 106, at 591-92. The specification of this proposal appears as S. Crocker et al., *New HOST-HOST Protocol* (IETF Network Working Grp. RFC No. 33, 1970), *available at* http://tools.ietf.org/pdf/rfc33.

119 Carr et al., *supra* note 106, at 590; Heart et al., *supra* note 107, at 553. NCP was later modified to allow the number of packets in transit to vary. McKenzie & Postel, *supra* note 110, at 18-20.

Interestingly, the ARPANET placed responsibility for RFNMs on the IMPs connected to the hosts rather than on the hosts themselves.[120] Some engineers presciently suggested that responsibility for this function properly resided with the receiving host rather than with its IMP.[121] Moreover, the ARPANET ensured reliability on a hop-by-hop basis, as each IMP retained a copy of the data until it received confirmation that the downstream IMP had received it successfully.[122]

NCP was not the only host-to-host protocol. The ARPANET's protocol architects designed other host-to-host protocols to meet different needs, such as the Network Voice Protocol (NVP) designed to support packet voice.[123] Because real-time communications are more tolerant of packet loss than other transmissions, NVP did not require IMPs to retransmit lost packets.[124] Moreover, because both the source and destination hosts were

---

Instead of waiting to send an RFNM until the network is not congested, another approach is to allow the receiving host to instruct its IMP to send a request to stop sending. The sending host would then hold any additional traffic until the receiving host sent a message authorizing the sending host to resume its transmission. S. Crocker, *Protocol Notes* 2 (IETF Network Working Grp. RFC No. 36, 1970), *available at* http://tools.ietf.org/pdf/rfc36. The protocol designers would obviate this requirement by having the sending hosts maintain a counter that increased or decreased based on commands sent by the receiving host. Instead of sending a specific request to stop sending, the receiving host could accomplish the same end simply by refusing to send the command to increment the counter. Steve Crocker et al., *An Official Protocol Proffering* 2 (IETF Network Working Grp. RFC No. 54, 1970), *available at* http://tools.ietf.org/pdf/rfc54.

120 Carr et al., *supra* note 106, at 590; Heart et al., *supra* note 107, at 554.

121 *See* M. Elie, *Comments on Memory Allocation Control Commands CEASE, ALL, GVB, RET) and RFNM* 1 (IETF Network Working Grp. RFC No. 68, 1970), *available at* http://tools.ietf.org/pdf/rfc68 ("[T]here is no reason why the RFNM could not be initiated by the receiving host as an acknowledgment of the correct reception of the message . . . ."); J. Kreznar, *Some Questions Re: HOST-IMP Protocol* 1 (IETF Network Working Grp. RFC No. 17, 1969), *available at* http://tools.ietf.org/pdf/rfc17 (quoting Stephen Crocker as asking, "Can a HOST, as opposed to its IMP, control RFNM's?"); *accord* J.H. Saltzer et al., *End-to-End Arguments in System Design*, 2 ACM TRANSACTIONS ON COMPUTER SYS. 277, 282 (1984) ("Another strategy for obtaining immediate acknowledgments is to make the target host sophisticated enough that when it accepts delivery of a message it also accepts responsibility for guaranteeing that the message is acted upon by the target application.").

122 Heart et al., *supra* note 107, at 554.

123 Danny Cohen, *Specifications for the Network Voice Protocol (NVP)* (IETF Network Working Grp. RFC No. 741, 1976), *available at* http://tools.ietf.org/pdf/rfc741; Dan Cohen, *A Protocol for Packet Switching Voice Communication*, 2 COMPUTER NETWORKS 320, 326-29 (1978) (discussing early development of NVP and the establishment of "real-time high quality voice conferencing"); Robert F. Sproull & Dan Cohen, *High-Level Protocols*, 66 PROC. IEEE 1371, 1377-80 (1978) (describing the purposes and challenges of NVP).

124 *See* Sproull & Cohen, *supra* note 123, at 1379 ("No retransmission to remedy errors is required because a lost message is not catastrophic . . . . Moreover, retransmission would introduce highly variable delays that cause worse perceptual damage than the loss of the message. The only effect of a lost message is an audible error for the period represented by the missing

operating in real time, NVP also omitted RFNM-based flow control associated with NCP.[125] Interestingly, it is the unreliable service associated with NVP, informed by Louis Pouzin's experience with a similar French network known as CYCLADES,[126] that would serve as the model for IP, the protocol that would knit the entire Internet together.

The original solicitation of formal bids on the ARPANET project divided the network into two parts: a user subnet that would support functions used by hosts (and thus by end users) and a communications subnet that managed the functions of IMPs.[127] The ARPANET's designers began to refer to these subnets as two different layers.[128] From this perspective, the ARPANET can be disaggregated into two distinct layers: a host layer responsible for organizing the functions provided by the hosts and a communications layer consisting of the IMPs and the leased telephone lines.[129] As is true generally speaking, the primary advantage of layering within the ARPANET was that it allowed the set of functions in one layer to largely ignore the detailed internal mechanics of the functions in the other layers.[130]

Figure 4: ARPANET Protocols as a Two-Layer Stack (circa 1970)[131]

Layer	Location	Functions
Host	Host-to-Host	Handles user activities; initiates and maintains connections between hosts
Communication	IMP-to-IMP	Moves data through the subnet using packet switching; ensures reliable transmission

data."). The mild effect of a lost message in the voice context stands in contrast to the high reliability required in transmitting more sensitive data.

125 *Id.*

126 ABBATE, *supra* note 103, at 125.

127 *See* F. HEART ET AL., REPORT NO. 4799, A HISTORY OF THE ARPANET: THE FIRST DECADE, at III-14 to -24 (1981), *available at* http://www.cs.utexas.edu/users/chris/DIGITAL_ARCHIVE/ARPANET/DARPA4799.pdf (discussing the criteria of the 1968 request for quotes); *see also* Heart et al., *supra* note 107, at 551 ("This approach divides the genesis of the ARPA Network into two parts: (1) design and implementation of the IMP subnet, and (2) design and implementation of protocols and techniques for the sensible utilization of the network by the Hosts.").

128 *See* Stephen D. Crocker et al., *Function-Oriented Protocols for the ARPA Computer Network*, 40 AFIPS CONF. PROC. 271, 271 (1972) (describing the "layers" in the protocols in the ARPANET).

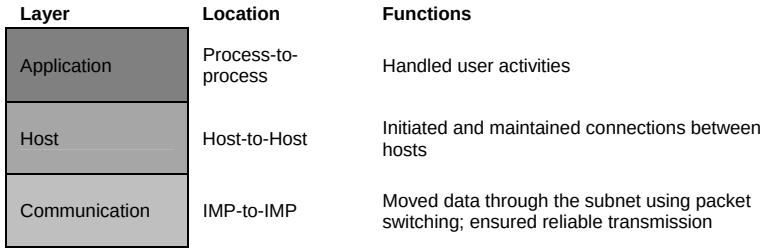
129 Heart et al., *supra* note 107, at 552-53.

130 RFC 871, *supra* note 3, at 3 ("[Layering is designed so that] a given set of related functions . . . should not take special cognizance of the detailed internal mechanics of another set of related functions . . . .").

131 ABBATE, *supra* note 103, at 53.

Combining all host functions into a single layer meant that host-system programmers had to design host-to-host connection systems into every program they wrote, even though many programs required the exact same functions. The protocol designers reduced this redundancy by conceptually disaggregating the host layer into two separate subnetworks. The *host subnetwork* represented the host-to-host functions associated with NCP, which provided general connection services needed by many applications. The *end-user subnetwork* encompassed the user interfaces and functionality employed by particular programs, such as file transfers and remote logins.[132] Although the protocol designers did not yet use the term, a modern observer would characterize this as a decision to divide what was previously considered a single host layer into separate host and applications layers.[133]

Figure 5: ARPANET Protocols as a Three-Layer Stack (circa 1973)[134]



B. *Interconnecting Heterogeneous Transmission Technologies*

In addition to interconnecting heterogeneous hosts, another goal of the Internet project was to enable the interconnection of heterogeneous transmission technologies. The solution to this problem was devised by Vinton Cerf and Robert Kahn.[135] Their solution was to divide the packet switches

---

132 *See* Peggy M. Karp, *Origin, Development and Current Status of the ARPA Network* (comparing the host subnetwork and the user-level subnetwork), *in* DIGEST OF PAPERS: "COMPUTING NETWORKS FROM MINI THROUGH MAXIS—ARE THEY FOR REAL?", PROC. 7TH ANNUAL IEEE COMPUTER SOC'Y INT'L CONF. 49, 49-50 (1973); *see also* RFC 871, *supra* note 3, at 13 (distinguishing the Host-Host layer, which confers interprocess communication functionality, and the Process Level/Applications layer, which contains those protocols that perform resource sharing and remote access functions).

133 ABBATE, *supra* note 103, at 67-68.

134 *Id.* at 68.

135 They laid out the solution in their classic 1974 paper. *See* Vinton G. Cerf & Robert E. Kahn, *A Protocol for Packet Network Intercommunication*, 22 IEEE TRANSACTIONS ON COMM. 637 (1974).

into two types: those that operated entirely within one type of network—later known as interior nodes or switches—and those that operated at the border between two different types of networks—which the authors called gateways and which were later known as routers.[136] The distinction between interior and gateway nodes greatly simplified the management problem by restricting the number of nodes that needed to know information about how to connect different types of networks. Information needed to interconnect with other types of networks, such as addressing, status detection, routing, and fault detection, could be restricted to the gateways.[137]

To minimize the problems of interconnecting disparate networks, Cerf and Kahn proposed that all networks employ a single, uniform addressing scheme.[138] In addition, instead of having each gateway identify the "language" spoken by the next network and translate it, Cerf and Kahn established a single common language that all networks could understand.[139] To facilitate its use by multiple networks, this common language was kept as simple as possible and included only the minimum information needed to transmit the communication.[140] All of this information was placed in an internetwork header that every gateway could read without modifying.[141]

The result was to subdivide what was previously called the communication layer into two distinct layers: one responsible for *inter*network communications—which was then called the Internet layer and would later be called the network layer—and another responsible for *intra*network communications—which was then called the network access layer. The result is the four-layer stack depicted in Figure 6.

136 *See id.* at 638 (introducing the concept of gateways, which serve as the interfaces between different types of networks).

137 ABBATE, *supra* note 103, at 128-29.

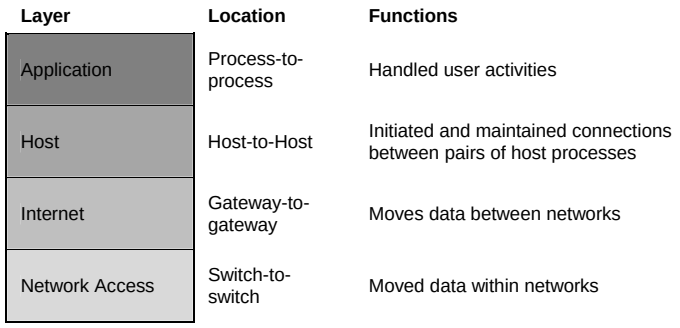
138 *See* Cerf & Kahn, *supra* note 135, at 641 ("A uniform internetwork TCP address space, understood by each gateway and TCP, is essential to routing and delivery of internetwork packets."); *see also* Vinton G. Cerf & Peter T. Kirstein, *Issues in Packet-Network Interconnection*, 66 PROC. IEEE 1386, 1393-99 (1978) (discussing the common internal address structure required for packet-level interconnectivity).

139 Cerf & Kahn, *supra* note 135, at 638-39.

140 *See* Barry M. Leiner et al., *The DARPA Internet Protocol Suite*, IEEE COMM., Mar. 1985, at 29, 31 ("The decision on what to put into IP and what to leave out was made on the basis of the question 'Do gateways need to know it?'.").

141 Cerf & Kahn, *supra* note 135, at 638-39.

Figure 6: ARPANET Protocols as a Four-Layer Stack (circa 1974)[142]



In the initial design, both interdomain routing and the host-to-host functions, such as reliability, were managed by a single large protocol called the Transmission Control Program.[143] After several years of trying to make this architecture work, the protocol architects realized that combining router-to-router and host-to-host functions within a single protocol contravened the basic principles of protocol layering. They decided to split the Transmission Control Program into two separate protocols: a host-to-host protocol called the Transmission Control Protocol and a router-to-router protocol called the Internet Protocol. As a result, the acronym changed from simply TCP to TCP/IP.[144]

The decision to split the protocol into two parts reflected a fundamental insight about the key layer of connectivity. The key to maintaining universal interconnectivity was the Internet Protocol running in the network

142 Leiner et al., *supra* note 140, at 29, 31 fig.3; *see also* RFC 1122, *supra* note 116, at 8-10 (listing and explaining the four-layer stack consisting of application, transport, Internet, and link layers, analogous to those depicted in Figure 6).

143 Cerf & Kahn, *supra* note 135, at 640.

144 *See* Jon Postel, *Comments on Internet Protocol and TCP* 1 (Internet Experiment Note No. 2, Aug. 15, 1977), *available at* http://www.rfc-editor.org/in-notes/ien/ien2.txt ("We are screwing up in our design of internet protocols by violating the principle of layering. Specifically we are trying to use TCP to do two things: serve as a host level end to end protocol, and to serve as an internet packaging and routing protocol. These two things should be provided in a layered and modular way. I suggest that a new distinct internetwork protocol is needed, and that TCP be used strictly as a host level end to end protocol."); *see also* David D. Clark, *The Design Philosophy of the DARPA Internet Protocols*, COMPUTER COMM. REV., Aug. 1988, at 106, 109 (discussing how differences in applications' need for reliability led designers to separate TCP into two host-to-host protocols (TCP and UDP) and a single router-to-router protocol (IP)).

layer.[145] This approach envisioned innovation in the hosts and applications operating in the layers above the network layer as well as in network resources operating in the layers below the network layer. Both sides could ignore any heterogeneity on the other side so long as the various components exchanged information through the Internet Protocol. Moreover, the fields in the interface established by the Internet Protocol limited the functionality of the network by determining what information could pass from the hosts to the routers and switches and vice versa.

Although the ARPANET's success and scope made it the most prominent packet-switched network sponsored by the Department of Defense, it was by no means the only one. ARPA also created the San Francisco Bay Area Packet Radio Network (PRNET). First operational in 1975, the PRNET enabled mobile hosts—including one in a radio-equipped van—to connect to four packet-radio repeaters located at high points around the Bay and to a central control station located at the Stanford Research Institute (SRI).[146] Another was the Atlantic Packet Satellite Network (SATNET), which used a satellite to establish a packet-switched connection to locations in Maryland, West Virginia, England, and Sweden.[147]

The presence of multiple, independent packet-switched networks inevitably led ARPA to look for ways to interconnect them.[148] But interconnecting networks that employed different technologies proved significantly more difficult than interconnecting the technologically identical IMPs associated with the ARPANET. As an initial matter, PRNET and SATNET created the possibility of multipath routing by employing broadcast technologies capable of simultaneously transmitting the same packet to multiple recipients instead of establishing a connection and

---

145 *See* Cerf & Kahn, *supra* note 135, at 638; Leiner et al., *supra* note 140, at 31; RFC 1958, *supra* note 99, at 2-3; *see also* KUROSE & ROSS, *supra* note 4, at 52-53; TANENBAUM, *supra* note 4, at 432.

146 *See* Robert E. Kahn et al., *Advances in Packet Radio Technology*, 66 PROC. IEEE, 1468, 1488-90 (1978) (describing and diagramming the PRNET). The PRNET benefited from the operational experience of a Hawaii-based packet radio network known as ALOHA. *See* Norman Abramson, *The ALOHA System—Another Alternative for Computer Communications*, 37 AFIPS CONF. PROC. 281, 282-85 (1970) (describing the random-access radio communications developed for use within the ALOHA system); R. Binder et al., *ALOHA Packet Broadcasting—A Retrospect*, 44 AFIPS CONF. PROC. 203, 203-15 (1975) (discussing packet broadcasting systems generally and focusing on lessons learned from the ALOHANET).

147 *See generally* Vinton G. Cerf, *Packet Satellite Technology Reference Sources* 2 (IETF Network Working Grp. RFC No. 829, 1982), *available at* http://tools.ietf.org/pdf/rfc829 (describing SATNET as "a packet satellite system which would support the sharing of a common, high speed channel among many ground stations"); Irwin Mark Jacobs et al., *General Purpose Packet Satellite Networks*, 66 PROC. IEEE 1448, 1460-65 (1978) (discussing SATNET's experimental facilities, measurement activities, and results).

148 *See* ABBATE, *supra* note 103, at 121-22.

transferring packets between two specific hosts.[149] In addition, SATNET was subject to propagation delays that were much longer than those associated with the other networks.[150] Spectrum-based transmission technologies like those used in PRNET and SATNET were also much more prone to packet loss than telephone-based transmission technologies. As a result, these networks employed network-based error recovery rather than simply relying on the hosts.[151] The packet switches within these different networks also provided radically different levels of service in terms of packet size, reliability, error correction, in-order packet delivery, and transmission speeds.[152] Finally, each network employed its own distinct scheme for assigning addresses to individual hosts.[153]

The International Network Working Group (INWG), a group working parallel with the ARPANET project, considered various solutions to these problems.[154] One was to require every host to implement every protocol used by other types of networks.[155] Another proposed solution was to allow each network to be aware of all other protocols and to translate the communication whenever it crossed a boundary between networks. The INWG rejected such systems as too cumbersome and prone to failure, particularly when the translation involved protocols employing fundamentally different principles.[156] Moreover, translation programming would also have to be updated whenever a new networking technology was added, with the

---

149 *See* Jacobs et al., *supra* note 147, at 1449 (noting that SATNET was a broadcast technology that could send "at any given time to all earth stations within its transmission coverage area" with only one hop); Kahn et al., *supra* note 146, at 1469, 1480 (noting that the PRNET was based on a single-hop broadcasting system that "is not a particularly efficient mode of operation for two party communications, but it is a very robust way to distribute packets to all parts of the net").

150 *See* Jacobs et al., *supra* note 147, at 1449 (noting that packet satellite networks are subject to propagation delays of roughly 250 milliseconds, making the round-trip delay a total of 500 milliseconds, or half a second).

151 *See id.* (noting that "[f]orward error correction techniques provide an efficient way to improve error performance" with respect to satellite networks); *id.* at 1458 (discussing the use and suppression of network-based reliability in the PODA algorithm employed in SATNET); Kahn et al., *supra* note 146, at 1479, 1492-93 (noting that packet radio networks are subject to higher error rates than wire-based communications and that, as a result, the PRNET used forward-error correction rather than leaving error correction to the hosts).

152 ABBATE, *supra* note 103, at 121-22.

153 Cerf & Kahn, *supra* note 135, at 641.

154 ABBATE, *supra* note 103, at 131-32.

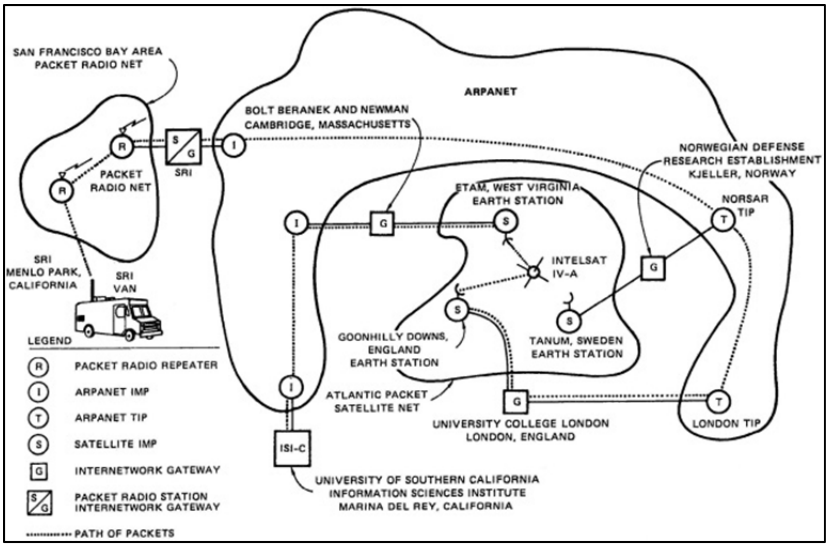
155 *See* Cerf & Kahn, *supra* note 135, at 638.

156 *See* ABBATE, *supra* note 103, at 128 (noting that designers considered requiring every network to be able to translate all existing host protocols, but rejected it because such a scheme would not scale and would not work seamlessly); Cerf & Kirstein, *supra* note 138, at 1399 (discussing the possibility of translating protocols at every boundary between networks, noting that mismatches in concept and the lack of a common address space would limit the functionality of such an approach and the difficulty of sequentially translating across multiple networks).

number of updates increasing combinatorially as the number of protocols increased.

Perhaps the most celebrated demonstration of how gateways could facilitate the interconnection of heterogeneous networks occurred in 1977, when a communication initiated by a truck connected to the PRNET in the San Francisco area was routed through the ARPANET to London, transmitted via SATNET from London to West Virginia, rerouted again through an ARPANET connection in Cambridge, Massachusetts, and delivered to the University of Southern California's Information Sciences Institute in Marina del Rey, California.[157] The initial implementation of the network layer was incomplete, in that the network layer protocol was implemented on a network-by-network basis and discarded at the gateways. In effect, what we now think of as the network layer operated more like an end-to-end transport layer until gateways became more universally deployed.

Figure 7: Interconnecting the ARPANET with the Packet Radio and Satellite Radio Networks[158]



The four-layer stack envisioned by the ARPANET's protocol architects largely ignored the structure below the network layer, lumping all of the

157 *See* ABBATE, *supra* note 103, at 131-32.

158 STANFORD RESEARCH INST., QUARTERLY MANAGEMENT REPORT NO. 15, PROJECT NO. 2325-N5, PACKET RADIO SYSTEM DEVELOPMENT 6 fig.1 (1976), *available at* http://archive.computerhistory.org/resources/text/2009/102686324.05.01.acc.pdf.

functions below it into a single network access layer.[159] In so doing, this model failed to consider that a particular medium can run more than one networking technology (for example, spectrum can run Ethernet, 802.11 (WiFi), 802.16 (WiMax), or legacy circuit-switched technologies associated with telephony). In addition, the same networking technologies could run on different physical media (for example, Ethernet can run on a wide variety of wireline and wireless technologies). To take this additional level of abstraction into account, the Internet Protocol stack borrowed a concept from another stack known as the Open Systems Interconnection (OSI) Reference Model and divided the lowest layer into a data-link layer and a physical layer.[160] The result is the five-layer Internet stack in Figure 8 that appears in every modern textbook on network engineering.

Figure 8: The Modern TCP/IP Reference Model[161]

Layer	Location	Protocols
Application	Process-to-process	SMTP (email), HTTP (web), FTP (file transfer), Telnet (remote login)
Transport	Host-to-Host	TCP (reliable); UDP (unreliable)
Network	Router-to-router	Internet Protocol
Datalink	Switch-to-switch	Ethernet, connection oriented (X.25, ATM, Frame Relay), wireless (802.11, Bluetooth)
Physical	Within network	Twisted pair (telephone), coaxial cable, fiber optics, spectrum

What is perhaps most striking is that although the ARPANET's protocol architects embraced layering as a fundamental principle from the very beginning, the design they ultimately adopted did not adhere to any preconceived notion about how functions should be divided among the different layers. Instead, as is typical of any modular design,[162] the layered model evolved during the design process through experimentation and

159 TANENBAUM, *supra* note 4, at 44, 49.

160 Zimmermann, *supra* note 3, at 430; *see also* Sicker, *supra* note 10, at 10 (distinguishing access providers and transport providers in the physical network).

161 *See, e.g.*, COMER, *supra* note 4, at 161-63; KUROSE & ROSS, *supra* note 4, at 51-53; TANENBAUM, *supra* note 4, at 49.

162 *See supra* notes 50-51 and accompanying text.

compromise rather than through a precommitment to a particular set of principles around which the network architecture should be organized.[163]

In fact, when the TCP/IP Reference Model first emerged, it was far from a monolith. During the late 1980s, many people believed that it was simply a transitional step toward a more general architecture based on the OSI Reference Model championed by the International Standards Organization (ISO)[164] or that the TCP/IP and the OSI models would coexist for a long time.[165] In fact, the battle between TCP/IP and OSI represented not only different philosophies (decentralized vs. centralized control) but also different sponsoring communities (netheads vs. bellheads).[166]

C. *The TCP/IP Reference Model*

The foregoing history established the framework for the existing five-layer protocol stack that underlies the modern Internet. Layers are traditionally numbered from the bottom of the stack, with the physical layer typically called layer 1, the data link layer typically called layer 2, the network layer typically called layer 3, the transport layer typically called layer 4, and the application layer typically called layer 5. For ease of exposition, however, in this Section the model will be presented from the top down.

1. The Application Layer

As was the case in the ARPANET, the topmost layer in the Internet stack is the application layer (typically called layer 5). This layer encompasses a wide variety of protocols, each designed to support particular classes of applications. For example, the HyperText Transfer Protocol (HTTP) is the application protocol that supports browsing the World Wide Web. It supports the entire class of web browsing programs, including

---

163 *See* ABBATE, *supra* note 103, at 51 ("The ARPANET's builders did not start out with a specific plan for how functions would be divided up among layers or how the interfaces and protocols would work. Rather, a layered model evolved as the ARPANET developed."); TANENBAUM, *supra* note 4, at 45 (noting that in the Internet, the protocols preceded the model).

164 TANENBAUM, *supra* note 4, at 45; *see also* Geoff Huston, *10 Years Later*, ISP COLUMN (June 2008), http://www.potaroo.net/ispcol/2008-06/10years.html (declaring the end of any suspicion that the Internet was a "waystop on the road to adoption of the [OSI] framework").

165 D. Clark et al., *Towards the Future Internet Architecture* 3 (IETF Network Working Grp. RFC No. 1287, 1991), *available at* http://tools.ietf.org/pdf/rfc1287.

166 *See generally* T.M. DENTON CONSULTANTS, NETHEADS VERSUS BELLHEADS: RESEARCH INTO EMERGING POLICY ISSUES IN THE DEVELOPMENT AND DEPLOYMENT OF INTERNET PROTOCOLS (Mar. 31, 1999), *available at* http://www.tmdenton.com/pub/bellheads.pdf.; Steve G. Steinberg, *Netheads vs. Bellheads*, WIRED, Oct. 1996, at 145.

Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari, and Opera, to name a few. Similarly, the Simple Mail Transfer Protocol (SMTP) supports all email clients, such as Microsoft Outlook and Mozilla Thunderbird, as well as web-based email systems such as Google's Gmail and Microsoft's Hotmail. Other common application-layer protocols include the file transfer protocol (FTP) and Telnet (for remote logins). Application protocols run exclusively within the end host.[167]

2. The Transport Layer

The next highest layer is the transport layer (typically called layer 4). It performs functions analogous to the ARPANET's host-to-host protocol. Whereas the application-layer protocols establish connections between processes, transport-layer protocols establish connections between hosts.[168]

Although the protocol designers envisioned a multitude of transport protocols, only two are widely used.[169] The first is the Transmission Control Protocol (TCP), which in 1983 replaced NCP as the Internet's primary transport protocol.[170] TCP operates on principles that are similar to NCP in many ways. As an initial matter, TCP uses a series of ports to direct data streams to the appropriate process running on the host.[171] Roughly 330 of the 65,536 existing ports have been preassigned to commonly used programs. HTTP, for example, is assigned to port 80, and FTP to port 21.[172]

TCP is also similar to NCP in that it uses an initial exchange of messages to establish a connection and to inform both hosts as to the port numbers that will be used. TCP likewise uses an exchange of messages to close a connection. In addition, TCP expects to receive a confirmation for every packet it sends. These confirmations are sent by the host, instead of the first-hop router to which that host is connected, and are called acknowledgments (ACKs) instead of RFNMs.[173]

---

167 KUROSE & ROSS, *supra* note 4, at 54-55.

168 TANENBAUM, *supra* note 4, at 42.

169 *See* M Handley, *Why the Internet Only Just Works*, 24 BT TECH. J. 119, 122-23 (2006) ("In short, a new transport protocol is not going to become widespread on a time-scale shorter than a decade, if ever.").

170 ABBATE, *supra* note 103, at 140-42.

171 *See* Cerf & Kahn, *supra* note 135, at 641 (introducing "the notion of *ports* in order to permit a process to distinguish between multiple message streams" and to allow TCP to "demultiplex[] the stream of internetwork packets it receives" and to direct each stream to the appropriate process).

172 KUROSE & ROSS, *supra* note 4, at 204.

173 Cerf & Kahn, *supra* note 135, at 643; *see also* Saltzer et al., *supra* note 121, at 282 (relating that RFNMs were "never found to be very helpful" because they did not indicate "whether or not the target host acted on the message").

TCP and NCP, however, differ in important ways. For example, NCP relied on the IMPs to ensure that every packet was delivered safely, while TCP places responsibility for reliability on the hosts. Specifically, every time TCP transmits a segment, it estimates the time that segment should take to reach the receiving host and return to the sender. If it has not received an ACK within the expected time frame the host duplicates the unacknowledged packet and resends it.[174] For reasons discussed in greater detail below, TCP also interprets a missing acknowledgment as a sign that the network is congested and, in those instances, cuts its sending rate in half.[175]

The other major transport protocol is known as the User Datagram Protocol (UDP), which in some ways is the natural successor to NVP. The protocol designers had originally thought that only one transport protocol was necessary and that all applications would run over TCP. But they soon discovered that certain applications do not run particularly well over TCP. This is because TCP inherently presumes that if a segment is dropped, the application will prefer to use the next available window of bandwidth to resend the old segment rather than to send a new segment. Unfortunately, the delay of waiting for TCP's retransmission timer to expire and for TCP to resend the packet creates an unacceptable delay for real-time interactive media. For example, packet voice works better if the listener asks the speaker to simply repeat the garbled message instead of having the application lock up while waiting for the dropped packet to be resent.[176]

The protocol architects thus created UDP to provide an alternative that would better support applications that are sensitive to latency. UDP uses the same port structure as TCP to ensure that the traffic arrives at the correct process in the receiving host. But unlike TCP, UDP begins transmitting data immediately without awaiting an exchange of messages to open a connection between the hosts. Even more importantly, unlike TCP, UDP simply sends a stream of segments without waiting for acknowledgments from the receiving hosts. As a result, UDP is unable to guarantee reliable delivery of messages. Finally, because UDP does not establish a connection

174 This expected time is actually the estimated round-trip time plus a grace period. The grace period is usually four times an approximation of the standard deviation of the acknowledgment arrival time. KUROSE & ROSS, *supra* note 4, at 251-53; TANENBAUM, *supra* note 4, at 552.

175 *See infra* Section IV.B.

176 Clark, *supra* note 144, at 106, 108-09. The VoIP example is illustrative, although the increase in bandwidth and the ability to interpolate missing samples have largely obviated the need for retransmissions with respect to VoIP.

with the receiving host, it does not need to exchange messages to close a transport-layer connection.[177]

The transport layer thus provides two very different types of services depending on which characteristics applications need the most. Applications that need reliability and can tolerate latency can use TCP to ensure reliable, error-free delivery. On the other hand, applications that are tolerant of packet loss and sensitive to latency are more likely to employ UDP.[178]

3. The Network Layer

The network layer (typically called layer 3) is generally regarded as the "glue that holds the entire Internet together," for this layer provides the uniform basis that each network connected to the Internet uses to transmit data communications across an ever-changing landscape of technologically heterogeneous systems.[179] Unlike application- and transport-layer protocols, network-layer protocols run in both routers and hosts. But because network-layer protocols govern only traffic transiting from one network to another, network-layer protocols need to run only in gateway routers and do not necessarily need to run in the switches operating the inside of a given network.[180]

The network layer protocol that integrates the entire Internet is known as the Internet Protocol (IP).[181] Unlike other layers, which are designed to permit a variety of protocols each designed to fulfill a different need, the network layer can support only a single uniform governing protocol.

To play this role effectively, IP had to strike a delicate balance. On the one hand, it had to include all of the information that any network would need to transmit packets to their destinations even when the packets had to traverse multiple, technologically heterogeneous networks. On the other hand, the desire to minimize the burden of running IP meant that it had to be kept as simple as possible.[182] As a result, IP encompasses only a minimal amount of information, including the source address, the destination

---

177 *See* COMER, *supra* note 4, at 176-77; TANENBAUM, *supra* note 4, at 525-26.

178 Clark, *supra* note 144, at 109.

179 *See* KUROSE & ROSS, *supra* note 4, at 53; TANENBAUM, *supra* note 4, at 432.

180 Leiner, *supra* note 140, at 30-31. For simplicity, many networks use IP-enabled nodes for both routers and switches.

181 Although IP is the most important network layer protocol, there are others. *See* TANENBAUM, *supra* note 4, at 449 (noting additional network layer protocols, including ICMP, ARP, RARP, BOOTP, and DHCP).

182 Leiner, *supra* note 140, at 31.

address, and a small amount of technical information.[183] Most importantly for the purposes of the current Internet policy debate, IP also allows applications to include a type-of-service marker that is intended to support prioritization of certain traffic,[184] although the syntax of this field has since been displaced by a new quality-of-service regime known as Differentiated Services.[185] The central organizing principle is to limit the network layer to the information that routers need to know.

4. The Data-Link Layer

The next layer is known as the data-link layer (typically called layer 2). Data-link layer protocols share with network-layer protocols the responsibility for guiding traffic through the network; as a result, data-link layer protocols necessarily run in switches as well as hosts. Because the data-link layer protocols were designed to govern how packets are transmitted within a single network as opposed to between two different networks, the layer is supposed to control the behavior of interior nodes,[186] although for technical reasons, networks may choose to deploy network-layer routers on interior nodes.[187] Whereas nodes that direct traffic based on network-layer information are typically called routers, nodes that direct traffic on the basis of data-link layer information are typically called switches.[188]

The data-link layer encompasses a wide variety of networking technologies. One of the most popular is Ethernet, which is almost undoubtedly the most widely used protocol in local area networks. Ethernet is basically a broadcast protocol that uses an unreliable, connectionless technology to send every packet it receives to every end user connected to it.[189]

In addition to Ethernet, the data-link layer includes many technologies that are byproducts of the connection-oriented approach to networking—such as Frame Relay and Asynchronous Transfer Mode (ATM)—on which legacy telephone companies have long relied. In fact, ATM was initially touted as a replacement for the Internet approach and came complete with

---

183 Cerf & Kahn, *supra* note 135, at 638-39. The technical information includes whether the packet is IP version 4 or version 6, the length of the header and the accompanying datagram, some instructions to guide the network should it have to divide a packet into smaller fragments, and some information to support error checking. *See* TANENBAUM, *supra* note 4, at 432-36.

184 TANENBAUM, *supra* note 4, at 434.

185 *See generally* S. Blake et al., *An Architecture for Differentiated Services* 12 (IETF Network Working Grp. RFC No. 2475, 1998), *available at* http://tools.ietf.org/pdf/rfc2475.

186 Leiner, *supra* note 140, at 30.

187 KUROSE & ROSS, *supra* note 4, at 491.

188 *Id.* at 491-93.

189 *See generally* TANENBAUM, *supra* note 4, at 65-68, 271-91 (describing the history and function of Ethernet).

its own reference model when launched in the early 1990s. Although it did not displace the basic Internet paradigm, it remains widely used in certain telephone systems.[190]

The data-link layer also encompasses networking technologies designed specifically for wireless systems.[191] These include the 802.11 family of protocols that support the WiFi systems widely used in home routers, the 802.16 protocol used to support WiMax systems like Clear, and the Bluetooth protocols designed to connect mobile phones to other devices. Wireless data-link architectures operate on very different principles. The frequent collisions resulting from the lack of a medium to monitor other users' activities[192] and the inherent unreliability of wave propagation forces data-link protocols that support wireless transmission to adopt different solutions than those used by wireline technologies.[193]

5. The Physical Layer

The final layer is the physical layer (typically called layer 1), which moves individual bits from one node to the next.[194] Any path that can convey a message sequence can constitute a link in this layer, whether physical or not.[195] The means of encoding information varies widely depending on whether the carrier wave is composed of visible light passing through a fiber optics network, an electromagnetic wave passing through a copper wire, or an electromagnetic wave passing through the ether.[196] In addition, different transmission media can use different approaches to modulating the carrier wave, such as varying its amplitude, frequency, or phase.[197] Media also differ in terms of bandwidth, attenuation, susceptibility to interference, and a host of other dimensions.[198]

---

190 *See id.* at 62 ("ATM was much more successful than OSI, and it is now widely used deep within the telephone system, often for moving IP packets.").

191 Note, however, that like the PRNET, the Ethernet developed from the protocols used to run the ALOHANET. *See* Robert M. Metcalfe & David R. Boggs, *Ethernet: Distributed Packet Switching for Local Computer Networks*, 19 COMM. ACM 395, 396 (1976).

192 TANENBAUM, *supra* note 4, at 69. Collisions were also a problem on early wired Ethernet implementations, but modern implementations have mitigated most of these problems. *See id.* at 66-67 (discussing the improved capability of Ethernet to listen for other activity).

193 *Id.* at 292-309; KUROSE & ROSS, *supra* note 4, at 536-58.

194 COMER, *supra* note 4, at 160; KUROSE & ROSS, *supra* note 4, at 51-53; TANENBAUM, *supra* note 4, at 41-44, 48-49.

195 *See* Heart et al., *supra* note 107, at 553 ("A link is a conceptual path that has no physical reality . . . .").

196 PETERSON & DAVIE, *supra* note 4, at 64.

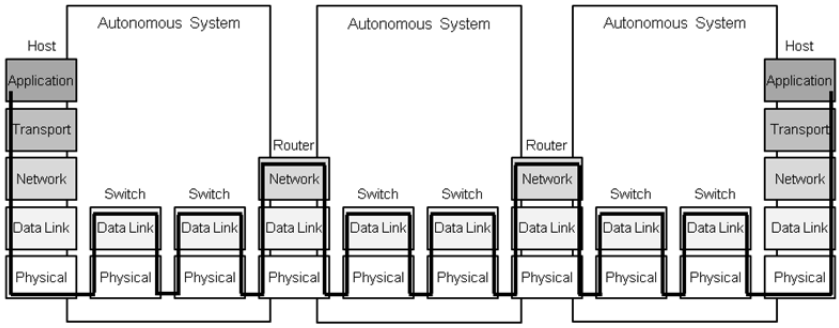
197 TANENBAUM, *supra* note 4, at 126.

198 *See* Christopher S. Yoo, *The Changing Patterns of Internet Usage*, 63 FED. COMM. L.J. 67, 77-78 (2010).

D. *Layering's Implications for Where Functions Are Performed*

The structure of the five-layer Internet stack dictates whether particular functions are performed by hosts operating at the network's edge or nodes operating in the network's core. The hosts operating at the edge of the network run all five levels of the stack, whereas the routers operating in the core of the network run only the bottom three layers. In addition, although the bottom three layers of the network stack—the network, data-link, and physical layers—operate in the routers serving as gateways between two networks, the switches operating inside a network run only the bottom two layers of the stack—the data-link and physical layers.

Figure 9: Where Layers Run[199]



Thus, the interface between the transport and the network layer defines a key boundary for determining where functions are performed. Although important exceptions exist, as a general matter services performed at or above the transport layer are provided by hosts, while the network layer defines the upper boundary for services provided by the routers and switches.[200]

III. CHARACTERIZATIONS OF THE LAYERED MODEL APPEARING IN THE LEGAL LITERATURE

Despite the engineering community's embrace of the five-layer Internet stack, discussions of protocol layering appearing in the legal commentary rarely frame the architecture in these terms. The most common approach condenses the layered model into a four-layer stack by compressing the

199 For a similar figure, see Leiner et al., *supra* note 140, at 30 fig.2.

200 COMM. ON THE INTERNET IN THE EVOLVING INFO. INFRASTRUCTURE ET AL., *supra* note 2, at 126.

transport and network layers into a single layer.[201] Others compress the analysis into a two-layer model that emphasizes a single interface separating the upper layers from the lower layers.[202] In either event, these commentators generally propose keeping the interfaces between layers open.[203]

A. *Combining the Transport and Network Layers into a Single Layer*

Combining the transport layer and the network layer into a single layer deprives the layered model of much of its analytical power and misperceives the network layer's function as the basis for universal connectivity.[204] Indeed, the decision to separate the transport from the network layer represents one of the central architectural decisions underlying the Internet; this decision was essential to supporting real-time applications, such as packet voice.[205] For content and application providers, the critical resource is access to the network layer (and thus to the interface between the transport and the network layer).[206] Combining the transport and network layers buries the key interface in the middle of a reconceptualized layer. As

---

201 *See, e.g.*, ZITTRAIN, *supra* note 7, at 67-68; Entman, *supra* note 12, at 2; Craig McTaggart, *A Layered Approach to Internet Legal Analysis*, 48 MCGILL L.J. 571, 582 (2003); Sicker & Blumensaadt, *supra* note 37, at 309-10; Werbach, *supra* note 7, at 59-64; Whitt, *supra* note 7, at 624; *see also* LESSIG, *supra* note 7, at 23-25 (discussing a three-layer model).

202 *See* Cannon, *supra* note 11, at 196-97 (conceiving of the network stack as a physical layer beneath a logical layer); Kevin Werbach, *Breaking the Ice: Rethinking Telecommunications Law for the Digital Age*, 4 J. ON TELECOMM. & HIGH TECH. L. 59, 78 (2005) (describing "the legacy regulatory structure" as "a nascent two-layer framework"); Whitt, *supra* note 7, at 652-53 (identifying the two-layer model as underlying the current regulatory regime and criticizing it for "miss[ing] the importance of interfaces between layers" (citing Werbach, *supra* note 7, at 55-56)); Tim Wu, *Why Have a Telecommunications Law? Anti-Discrimination Norms in Communications*, 5 J. ON TELECOMM. & HIGH TECH. L. 15, 22-23 (2006) (describing proposals for an "implicit, but un-codified two-layer system"); Wu, *supra* note 7, at 1191 ("So while there are actually four layers in the Internet architecture, for many purposes the most important distinction is between the transport layers . . . and the interpretation layers . . . .").

203 *See* Entman, *supra* note 12, at 16 ("Most participants agreed that . . . public policy should keep interfaces open to interconnection at each layer."); Werbach, *supra* note 202, at 81-82 (arguing for reorienting regulation to keep the connective layers open, including the one intermediating between the application layer and the physical layer); Wu, *supra* note 7, at 1192 (focusing on the need for regulation to keep open the interface between the application layer and the transport layer).

204 *See supra* subsection II.C.3.

205 *See supra* notes 176-78 and accompanying text.

206 *See* COMM. ON THE INTERNET IN THE EVOLVING INFO. INFRASTRUCTURE ET AL., *supra* note 2, at 128-29 (noting that inability to access the network layer would "constrain[] user choice and deteriorate[] . . . the quality of products"); RFC 1958, *supra* note 99, at 2 ("The key to global connectivity is the inter-networking layer.").

such, it obscures important issues by making it impossible to examine access at this interface.

Moreover, the transport layer runs exclusively in hosts, while the network layer defines the upper boundary for services provided by the routers and switches.[207] Compressing the layers in this manner thus contradicts the central tenet of layering that limits entities to interacting only with their peers—gateways only with other gateways, hosts only with other hosts.[208] Combining a layer that operates in both routers and hosts with one that runs solely at the edge of the network makes it impossible to map the resulting layered stack onto the end-to-end argument.[209] Finally, if the single-logical-layer argument is combined with an argument for nondiscrimination in the logical layer,[210] users would be prohibited from utilizing one of the core features built into the network layer from the outset: the Type-of-Service flag included in IP to permit prioritization.[211] These problems have led even engineers who are sympathetic to the goals of network neutrality to criticize the manner in which these models have oversimplified the TCP/IP Reference Model.[212]

B. *Dumb Pipes vs. the Hourglass Model*

In addition to permitting the Internet to run a wide variety of applications and host processes seamlessly, the Internet Protocol also enables the network to run an arbitrary variety of transmission technologies. This ability is facilitated by a thin, simple layer in the middle of the protocol

---

207 *See supra* note 200 and accompanying text.

208 *See supra* note 54 and accompanying text.

209 *See* Scott Jordan, *A Layered Network Approach to Network Neutrality*, 1 INT'L J. COMM. 427, 443-44 (2007) [hereinafter Jordan, *Layered Network Approach*]; Scott Jordan, *Implications of Internet Architecture on Net Neutrality*, ACM TRANSACTIONS ON INTERNET TECH., May 2009, at 5:1, 5:16 [hereinafter Jordan, *Internet Architecture*].

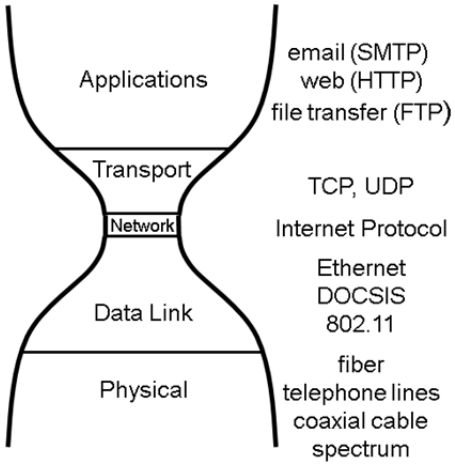
210 *See, e.g.*, VAN SCHEWICK, *supra* note 7, at 73 ("[F]iltering and control mechanisms in the network's core may also violate the layering principle if the mechanisms operate at the Internet layer or at a lower layer but access or modify the message . . . ."); Tim Wu, *Network Neutrality, Broadband Discrimination*, 2 J. ON TELECOMM. & HIGH TECH. L. 141, 168 (2003) ("[A]bsent evidence of harm to the local network or the interests of other users, broadband carriers should not discriminate in how they treat traffic on their broadband network on the basis of inter-network criteria.").

211 *See supra* note 184 and accompanying text. For a description of the Type-of-Service flag, see TANENBAUM, *supra* note 4, at 434.

212 *See* Jordan, *Internet Architecture*, *supra* note 209, at 5:16; Jordan, *Layered Network Approach*, *supra* note 209, at 443; *see also* Kai Zhu, Note, *Bringing Neutrality to Network Neutrality*, 22 BERKELEY TECH. L.J. 615, 634 (2007) (criticizing accounts that "simplify[] the technically complex and elegant TCP/IP into a 'dumb pipe' or a 'code layer'" as "both technically inaccurate and conceptually misleading").

stack to mediate between any heterogeneity in the upper and lower layers. The solution was not to require that all of the lower layers be simple or uniform, as some proponents of general protocols have suggested.[213] Instead, the architecture envisions a wide variety of technologies both above and below the network layer, with many lower layer protocols containing sophisticated functionality, including Frame Relay, Asynchronous Transfer Mode, Active Queue Management, and MultiProtocol Label Switching.

Figure 10: The Hourglass Model of the Internet Protocol Stack[214]



It is for this reason that the classic representation of the TCP/IP Reference Model takes the shape of an hourglass, with the Internet Protocol serving as the thin waist.[215] The hourglass structure emphasizes that only

213 *See* Cannon, *supra* note 11, at 197 (dividing information services into lower-layer basic services and upper-layer enhanced services); Wu, *supra* note 7, at 1191-92 (arguing that his proposed four-layer model can be simplified into two layers).

214 For a similar figure, see Steve Deering, *Watching the Waist of the Protocol Hourglass*, PROC. FIFTY-FIRST INTERNET ENG'G TASK FORCE 2 (2001), *available at* http://www.ietf.org/proceedings/51/slides/plenary-1/sld002.html.

215 *See* COMM. ON THE INTERNET IN THE EVOLVING INFORMATION INFRASTRUCTURE ET AL., *supra* note 2, at 36, 126-28; Walter Willinger & John Doyle, *Robustness and the Internet: Design and Evolution*, *in* ROBUST DESIGN: A REPERTOIRE OF BIOLOGICAL, ECOLOGICAL, AND ENGINEERING CASE STUDIES 231, 242-43 (Erica Jen ed., 2005); RFC 3439, *supra* note 26, at 3; *see also* VAN SCHEWICK, *supra* note 7, at 89 (noting that with "the Internet Protocol as the hourglass's waist," developers can innovate independently on either end); ZITTRAIN, *supra* note 7, at 67-69 (describing the hourglass and explaining, "It is only the middle that is narrow, containing Internet protocol, because it is meant to be as feature-free as possible"). The hourglass model is

the network layer must remain uniform and relatively simple. The services provided by the upper and lower layers of the protocol stack are anything but uniform and can employ complex and elaborate network management practices. It is for this reason that many in the engineering community have rejected attempts to reduce protocol layering into the simple policy inference that lower layers should be kept as dumb as possible.[216]

C. *Layering and Competition Policy*

An additional concern is that analyses based on engineering concepts like protocol layering are sometimes offered as substitutes for conventional tools of policy analysis. For example, debates over Internet policy sometimes focus on whether a particular practice is consistent with layering without addressing whether that practice furthers other policy concerns, such as promoting economic competition. In particular, many have invoked the layers model to justify subjecting the lower layers of the Internet to regulation while largely exempting the upper layers from regulatory scrutiny.[217]

---

usually attributed to John Aschenbrenner in connection with the OSI layered stack. *See* Jean Bartik, *OSI: From Model to Prototype as Commerce Tries to Keep Pace*, DATA COMM., Mar. 1984, at 307, 314-15; B Carpenter & S Brim, *Middleboxes: Taxonomy and Issues* 25 (IETF Network Working Grp. RFC No. 3234, 2002) [hereinafter RFC 3234], *available at* http://tools.ietf.org/pdf/rfc3234.pdf.

216 *See, e.g.*, Jordan, *Internet Architecture*, *supra* note 209, at 5:21; Jordan, *Layered Network Approach*, *supra* note 209, at 446; *see also* Zhu, *supra* note 212, at 634 (offering a technical critique of arguments that TCP/IP mandates "dumb pipe[s]").

217 *See* Mark Cooper, *Open Communications Platforms: The Physical Infrastructure as the Bedrock of Innovation and Democratic Discourse in the Internet Age*, 2 J. ON TELECOMM. & HIGH TECH. L. 177, 180 (2003) (arguing for regulation "preserving an open physical layer within the communications platform" because its owners "are in a unique position" to "employ singular, narrow motives and leverage market power in order to protect existing monopoly rents to achieve domination over neighboring products"); Susan Crawford, *The Internet and the Project of Communications Law*, 55 UCLA L. REV. 359, 375-76 (2007) (surveying literature arguing that upper layers should remain unregulated); Werbach, *supra* note 7, at 59-60 ("In general terms, regulation is more justified at lower layers, because openness at one layer often allows for innovation at higher layers."); Whitt, *supra* note 7, at 592 (noting that "when applied in the telecommunications industry context, the Network Layers Model targets the lower network layers for discrete regulation based on the existence of significant market power" while "leaving otherwise competitive content and applications markets unfettered by regulation"). Both Whitt and Werbach later qualified these statements by acknowledging that anticompetitive problems can arise in any layer. *See* Kevin Werbach, *Bringing Home the Bits*, 4 J. ON TELECOMM. & HIGH TECH. L. 59, 80 (2005) (rejecting approaches that presuppose heavy regulation on lower layers and little to no regulation on the upper layers); Werbach, *supra* note 7, at 60 n.90 ("Under some circumstances, more extensive regulation may be justified at a higher layer, or competition may be sufficient to ensure openness without the need for regulatory intervention."); Whitt, *supra* note 7, at 635 ("Of course, . . . one cannot assume that the exclusive gatekeeper will only exist at the physical layer. Indeed, a recent study solicited by the European Commission explains that Next Generation Networks ('NGNs') likely will contain new 'control points' that can reside in any layer or 'plane' of the network hierarchy." (footnote omitted)).

Other proposals suggest that network management techniques that violate protocol layering be regarded as inherently problematic.[218]

Calls for singling out the physical layer for regulatory scrutiny are both overinclusive and underinclusive. They are underinclusive in that not every aspect of the physical layer is characterized by the type of high fixed costs that tend to create market failure. The FCC, for instance, has rejected regulating backbone providers on the theory that the market is sufficiently competitive.[219] It is only the "last mile"[220] that is the source of concern (and even that aspect is becoming more competitive, particularly as the deployment of 4G LTE makes wireless an increasingly plausible alternative).[221] Hence, many layers commentators argue that the physical layer must be disaggregated into core and last-mile components.[222]

The underinclusive nature of focusing regulatory attention exclusively on the physical layer stems from the fact that market power is possible at the upper layers as well, which is reflected in the recent antitrust cases against Microsoft and Google.[223] Admittedly, upper-layer services are not characterized by the high fixed costs that have been the traditional source of market domination in the telecommunications industry. That said, other economic features such as first-mover advantages, network effects, or intellectual property can create anticompetitive concerns. This is why many who advocate using layering as the basis for regulatory policy acknowledge

---

218 *See, e.g.*, Comments of Google, *supra* note 18, at 69-70 ("Network congestion [management] techniques . . . should be consistent with Internet layers architecture . . . .").

219 *See* Daniel F. Spulber & Christopher S. Yoo, *Mandating Access to Telecom and the Internet: The Hidden Side of* Trinko, 107 COLUM. L. REV. 1822, 1891 (2007) (describing the FCC's policy that in the absence of a dominant backbone player, individual backbones have sufficient incentive to interconnect even absent regulation).

220 The "last mile" refers to the final leg of the delivery of telecommunications services to consumers. Daniel F. Spulber & Christopher S. Yoo, *Rethinking Broadband Internet Access*, 22 HARV. J.L. & TECH. 1, 2 n.3 (2009).

221 *See id.* at 17, 40 (describing how the emergence of competition in last-mile broadband service justified deregulation); *id.* at 9-10, 25-27 (tracing the emergence of wireless as the leading last-mile broadband platform and noting that "the FCC has specifically rejected the conclusion that last-mile broadband services constitute a natural monopoly").

222 *See* Sicker, *supra* note 10, at 11, 16 (dividing physical layer services into "access," i.e. last-mile, and "transport," i.e. the core of the network); Whitt, *supra* note 7, at 623-24 (same).

223 The antitrust authorities' recent interest in Facebook indicates that market power can exist in applications as well. *See* Christopher S. Yoo, *When Antitrust Met Facebook*, 19 GEO. MASON L. REV. 1147, 1148-54 (2012) (describing "network effects," which "exist when the value of a network depends on the number of other users connected to the network," as a source of market power for Facebook and other social networking sites). The cable program access rules indicate that market power can arise with respect to content as well. Werbach, *supra* note 202, at 79 ("The program access rules in the 1992 Cable Act . . . were designed to prevent cable operators from using their dominance of certain high-value content to prevent competition at the physical layer . . . .").

that market power can arise at any layer.[224] It also explains why the FCC's Open Internet Order "reject[ed] proposals to limit [its] rules to actions taken at or below the lower network layer."[225]

Protocol layering is also frequently lauded for its ability to promote competition in another way. As a general matter, enabling actors to connect without asking permission and to innovate within their layers without affecting other layers is likely to promote competition. A closer examination, however, reveals that the combination of open connectability and intra-layer freedom can actually limit competition. Professor Timothy Bresnahan has emphasized the potential benefits of what he calls "divided technical leadership," in which firms with similar technical and marketing capabilities push against the vertical boundaries separating them in an attempt to seize control over key platform elements.[226] Regulation that mandates access would freeze these interfaces and threaten to block these markets from this important source of rivalry.[227] While Bresnahan offered this theory in a somewhat different context, the same insights apply to the Internet with considerable force.

Other studies have emphasized how coordination between input providers can internalize the positive externalities created by general purpose technologies[228] or can allow new entrants to manage the market power wielded by input suppliers before undertaking relationship-specific investments.[229] These countervailing considerations underscore the extent to which the relationship between openness and competition policy is more complex than suggested by the arguments of those who equate layering with promoting competition.

IV. THE IMPACT OF TECHNOLOGICAL CHANGE ON THE LAYERED MODEL

There can be little question that the five-layer TCP/IP Reference Model that frames the structural analysis of the Internet represents a tremendous technical achievement. It has proven remarkably stable and robust, scaling to accommodate a dazzling variety of users, applications, and networking

---

224 Katz, *supra* note 12, at 37-38; Weiser, *supra* note 37, at 13; Werbach, *supra* note 202, at 79; Whitt, *supra* note 7, at 635-36.

225 Open Internet Order, *supra* note 1, at 17,948 n.235.

226 Bresnahan, *supra* note 102, at 166.

227 *See* Christopher S. Yoo, *Beyond Network Neutrality*, 19 HARV. J.L. & TECH. 1, 11 (2005) [hereinafter Yoo, *Beyond Network Neutrality*]; Christopher S. Yoo, *Vertical Integration and Media Regulation in the New Economy*, 19 YALE J. ON REG. 171, 282-85 (2002).

228 Bresnahan & Trajtenberg, *supra* note 36, at 96.

229 Teece, *supra* note 36, at 302.

technologies. Its past success has understandably led many in the policy debate to embrace it as an essential pillar to preserve.

But the success of the model makes it easy to forget that, like all layered architectures, it is the product of a series of tradeoffs determined largely by the nature of the underlying technological interdependencies and the nature of the services being demanded of the network. It is thus appropriate to examine the conditions under which circumstances have changed sufficiently to merit changing the architecture as well.

A. *Reliability*

One area that is undergoing technological change is with respect to the way that the current layered stack ensures reliability. Although reliability in the ARPANET was largely guaranteed by the IMPs operating in the core of the network, the existing layered architecture assigns responsibility for guaranteeing reliable transmission to the transport layer.[230] As a result, the current architecture assigns the responsibility for ensuring reliability to the hosts operating at the edge of the network. However, the advent of wireless transmission technologies has begun to place pressure on the assumption that the transport layer should remain the locus for ensuring reliability.

As noted earlier, when TCP sends a packet, it sets a retransmission timer; if it has not received an acknowledgment by the time the timer expires, it presumes the packet has been lost and resends it.[231]

The rationale generally invoked for this decision is laid out in the landmark article on the end-to-end argument by Jerome Saltzer, David Reed, and David Clark. They argued that performing intermediate error checking provided few additional benefits because the hosts at either end of the communication are likely to perform an end-to-end error check anyway, which necessarily involves information only available at the network's edge.[232] Moreover, implementing reliability in the core would require all applications, even those applications whose tolerance for damaged or lost

---

230 *See* Geoff Huston, *The End of End to End?*, ISP COLUMN (May 2008), http://www.potaroo.net/ispcol/2008-05/e0e2e.html (noting that the predominant approach to digital networking during the 1970s and 1980s required that each switch in a path store a local copy of the data until it received confirmation that the downstream switch had received the data and calling approaches that placed responsibility for reliability on hosts "heresy"); *see also* ABBATE, *supra* note 103, at 125 (calling approaches giving hosts the responsibility for maintaining reliability "unconventional").

231 *See supra* note 174 and accompanying text.

232 Saltzer et al., *supra* note 121, at 278-81.

packets was very high, to bear the delays associated with maintaining reliability.[233]

For these reasons, the Internet's protocol designers shifted responsibility for error recovery from the nodes to the hosts.[234] The nodes in the core of the network became pure store-and-forward routers that no longer had to keep track of what happened to packets after transmission.

A closer analysis of the decision not to implement reliability in the core reveals a decisionmaking process that is somewhat more complex. Assigning responsibility for reliability to the core of the network would have meant that nodes had to track packets already transmitted until the next downstream router confirmed receipt. If a node failed, the only way the network could have recovered from such a loss would have been to restart the entire session from scratch.[235]

More importantly for our purposes, the decision to assign responsibility for maintaining reliability to the hosts was the result of a pragmatic tradeoff. Saltzer, Reed, and Clark explicitly recognized that "it would be too simplistic to conclude that the lower levels should play no part in obtaining reliability" and that "the amount of effort to put into reliability measures within the data communication system is seen to be an engineering tradeoff based on performance, rather than a requirement for correctness."[236] Indeed, they recognized that some networks may be so unreliable as to justify giving the lower layers greater responsibility for ensuring reliability.[237]

The advent of wireless broadband may represent an example of when it may make sense to shift responsibility for ensuring reliability to a lower layer. Unlike wireline networks, which rarely drop packets for reasons other than congestion, wireless networks suffer from much higher loss rates

---

233 *See id.* at 284-85 (citing real-time voice conversations as an example of an application with a high tolerance for damaged packets but a low tolerance for delay). As David Clark observed,

> [T]he most serious source of delay in networks is the mechanism to provide reliable delivery. A typical reliable transport protocol responds to a missing packet by requesting a retransmission and delaying the delivery of any subsequent packets until the lost packet has been retransmitted. It then delivers that packet and all remaining ones in sequence. The delay while this occurs can be many times the round trip delivery time of the net, and may completely disrupt the speech reassembly algorithm. In contrast, it is very easy to cope with an occasional missing packet.

Clark, *supra* note 144, at 109.

234 Cerf & Kahn, *supra* note 135, at 643.

235 *See* Clark, *supra* note 144, at 107-08.

236 Saltzer et al., *supra* note 121, at 280-81.

237 See *id.* at 281 (acknowledging that, at times, "[p]erforming a function at a low level may be more efficient, if the function can be performed with a minimum perturbation of the machinery already included in the low-level subsystem").

caused by the sensitivity of spectrum-based transmission to local conditions.[238] Recovering from packet loss can be quite slow, as the process must wait for a sending host's retransmission timer to expire. Moreover, sending the duplicate packet from the host necessarily consumes additional network resources.

For this reason, modern wireless broadband networks increasingly deploy network-based reliability systems, such as Automatic Repeat reQuest (ARQ), that allow the data-link layer to access information associated with the transport layer in order to improve error recovery. Under ARQ, the data-link layer uses acknowledgments between adjacent switches to achieve faster recovery from errors.[239] In addition, the data-link layer installs an entity known as a snoop agent that sniffs all packets heading toward the receiver to determine whether they are using TCP. If the snoop agent receives the data-link layer acknowledgment, it generates its own TCP acknowledgment and drops the TCP acknowledgment generated by the receiving host.[240]

Shifting functions currently performed by the transport layer into either the data-link layer or routers operating in the core (and allowing the data-link layer to observe information associated with the transport layer) diverges from the TCP/IP reference model. These proposals have been controversial, principally because deviating from the layered model introduces variation, which increases the number of interdependencies that each layer must take into account.[241] For example, there are reports that additional latency introduced by ARQ is exacerbating problems with bufferbloat.[242]

Experiments allowing lower layers to access information associated with higher layers in order to improve network performance are part of the burgeoning literature on cross-layer design in wireless networks.[243] The engineering community has yet to reach consensus on the merits of cross-layer design. That said, the pragmatic nature of network engineering

---

238 Yoo, *supra* note 198, at 77-80.

239 For a detailed explanation of ARQ, see Dzmitry Kliazovich & Fabrizio Granelli, *A Cross-Layer Scheme for TCP Performance Improvement in Wireless LANs*, 2004 PROC. IEEE GLOBAL TELECOMM. CONF. 840, 841.

240 *Id.*

241 Kawadia & Kumar, *supra* note 34, at 7-8.

242 For more discussion of bufferbloat, see generally Mark Allman, *Comments on Bufferbloat*, 43 COMPUTER COMM. REV. 31 (2013); and Jim Gettys & Kathleen Nichols, *Bufferbloat: Dark Buffers in the Internet*, 55 COMM. ACM 57 (2012).

243 For surveys of this literature, see Fotis Foukalas et al., *Cross-Layer Design Proposals for Wireless Mobile Networks: A Survey and Taxonomy*, 10 IEEE COMM. SURVS. & TUTORIALS 70 (2008); Sanjay Shakkottai et al., *Cross-Layer Design for Wireless Networks*, IEEE COMM., Oct. 2003, at 74; Vineet Srivastava & Mehul Motani, *Cross-Layer Design: A Survey and the Road Ahead*, IEEE COMM., Dec. 2005, at 112.

counsels against basing objections on rigid adherence to a fundamentalist principle.[244] A better policy would be to understand how the determinants of layering may change and to develop heuristic guides to determine when a change in the underlying tradeoffs may justify a shift to a different layered architecture.

B. *Congestion*

Congestion management represents another interesting case study of how the protocol architects allocated a particular function to particular layers. The ARPANET placed primary responsibility for congestion management on the IMP connected to the receiving host, which used RFNMs as a form of flow control to ensure that it did not receive more packets than it could handle.[245]

The Internet's initial design similarly followed a core-based approach, having routers send ICMP source quench messages to hosts when they became congested,[246] but that method proved to be ineffective.[247] Adopting a strikingly different approach, the Internet manages congestion by assigning primary responsibility to the sending host. The principles articulated by Saltzer, Reed, and Clark, which note that functions should be implemented where necessary information resides,[248] suggest that the sending host is a problematic locus for managing congestion. Congestion is typically the product of what multiple hosts are doing, but individual hosts generally possess information about only their own activities and lack information about the behavior of other hosts. The nodes in the core of the network are better positioned to observe the flows being generated by multiple users.[249]

---

244 *See The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture* 8, 10 (IETF Network Working Grp. RFC No. 3724, J. Kempf & R. Austen eds., 2004) [hereinafter RFC 3724], *available at* http://tools.ietf.org/pdf/rfc3724 (contending that insistence on a fundamentalist principle is "an unproductive approach").

245 *See supra* Section I.A.

246 *See* J. Postel, *Internet Control Message Protocol: DARPA Internet Program Protocol Specification* 10-11 (IETF Network Working Grp. RFC No. 792, 1981), *available at* http://tools.ietf.org/pdf/rfc792.pdf (explaining ICMP source quench messages); RFC 1122, *supra* note 116, at 103 (requiring hosts to react to these messages).

247 Fernando Gont, *Deprecation of ICMP Source Quench Messages* 3 (IETF Network Working Grp. RFC No. 6633, 2012), *available at* http://tools.ietf.org/pdf/rfc6633.pdf.

248 *See supra* note 232 and accompanying text.

249 *See* Sally Floyd & Van Jacobson, *Random Early Detection Gateways for Congestion Avoidance*, 1 IEEE/ACM TRANSACTIONS NETWORKING 397, 397 (1993) ("The most effective detection of congestion can occur in the gateway itself."); *see also* Handley, *supra* note 169, at 120 ("Congestion is essentially a network-level problem rather than a transport-level problem . . . .").

Thus, the same information-requirements rationale that suggested that reliability should be managed from the edge of the network now suggests that congestion should be managed by the core. Since congestion management is arguably a service required by a broad range of applications and since it requires information that is available only in the core, a core-based solution seems most appropriate.[250]

The decision to deviate from the logical locus for congestion management is best understood in light of the sense of urgency during the mid-1980s, when the decision to manage congestion from the edge was made. In one of the few instances in which network engineers did not stay ahead of major problems, the network "suffered from a series of congestion collapses."[251] Any network-based solution would have required modifying all of the network's core routers, which would have taken a long time and been prohibitively expensive.[252]

Host-based congestion management depends on hosts' exponentially ramping down the amount of traffic they are transmitting whenever the network becomes congested. Without a direct congestion signal, however, hosts needed some basis for inferring from the signals that were visible to them when the network was congested. Van Jacobson and Mike Karels devised an ingenious solution to this problem that required adding only a few lines of code to the next release of UNIX.[253] They noted that networks typically drop packets for only two reasons: either the packet becomes corrupted or it encounters a congested buffer that was full.[254] Because wireline networks rarely corrupt packets, hosts could take the failure to receive an acknowledgment within the expected amount of time as a de facto signal that the network was congested and as an indication that they needed to reduce the amount of traffic they were sending through the network.[255]

---

250 Indeed, Raj Jain proposed a core-based solution that would have enabled routers to notify hosts about congestion at the time the edge-based solution was adopted. RAJ JAIN ET AL., DIGITAL EQUIP. CORP., CONGESTION AVOIDANCE IN COMPUTER NETWORKS WITH A CONNECTIONLESS NETWORK LAYER 6-7 (1997), *available at* http://www1.cse.wustl.edu/~jain/papers/ftp/cr5.pdf (describing a scheme in which "[a]ll routers in the subnet monitor their load").

251 Handley, *supra* note 169, at 120.

252 *See* Van Jacobson, *Congestion Avoidance and Control*, COMPUTER COMM. REV., Aug. 1988, at 314, 319 (noting that a network-based solution would "require[] a . . . modification to *all* existing gateways").

253 *Id.* at 314-15, 321.

254 *Id.* at 319.

255 *Id.*

This regime was deployed relatively quickly and has served as the primary mechanism for managing congestion on the Internet ever since.[256] Assigning responsibility for congestion management to the hosts also made the network more flexible by eliminating the need to knit together different approaches to congestion management employed by heterogeneous networks. Although subsequent mechanisms have been developed to enlarge the core's role in managing congestion, such as Random Early Discard (RED) and Explicit Congestion Notification (ECN),[257] these mechanisms are designed around the scheme devised by Jacobson and Karels. Thus, they are effectively supplements to, rather than replacements for, host-based congestion management.

This host-based approach was never intended to be a permanent solution.[258] As an initial matter, because it depended on an inference from the lack of an acknowledgment, it worked only for traffic based on TCP.[259] The advent of latency-sensitive applications, such as streaming video and VoIP, has placed increasing emphasis on UDP. The increase in the proportion of traffic that does not use acknowledgments has pressured developers and computer scientists to find a better solution.[260] Some scholars have proposed requiring that all UDP implementations be "TCP friendly," consuming the same amount of bandwidth as a typical TCP-based communication.[261] Thus

---

256 TANENBAUM, *supra* note 4, at 547.

257 For more on RED, see Bob Braden et al., *Recommendations on Queue Management and Congestion Avoidance in the Internet* 2-3, 7-8 (IETF Network Working Grp. RFC No. 2309, 1998), *available at* http://tools.ietf.org/pdf/rfc2309.pdf. For more on ECN, see K. Ramakrishnan et al., *The Addition of Explicit Congestion Notification (ECN) to IP* 3, 10-13 (IETF Network Working Grp. RFC No. 3168, 2001), *available at* http://tools.ietf.org/pdf/rfc3168.pdf. Interestingly, problems with bufferbloat strengthen the use case for deploying RED. For a detailed discussion, see Gettys & Nichols, *supra* note 242, at 59-63.

258 *See* Jacobson, *supra* note 252, at 322 (envisioning host-based congestion recovery as an intermediate step toward gateway-based congestion detection).

259 Handley, *supra* note 169, at 120.

260 *See* KUROSE & ROSS, *supra* note 4, at 213 (calling UDP's lack of congestion control "controversial" because its failure to reduce its rate in response to packet loss can cause UDP traffic to "crowd[] out . . . TCP sessions"). Kurose and Rose also note that

> [f]rom the perspective of TCP, the multimedia applications running over UDP are not being fair—they do not cooperate with the other connections nor adjust their transmission rates appropriately. Because TCP congestion control will decrease its transmission rate in the face of increasing congestion (loss), while UDP sources need not, it is possible for UDP sources to crowd out TCP traffic.

Id. at 293.

261 *E.g.*, Sally Floyd et al., *Equation-Based Congestion Control for Unicast Applications*, 30 COMPUTER COMM. REV. 43, 43 (2000); E. Kohler et al., *Datagram Congestion Control Protocol (DCCP)* 6 (IETF Network Working Grp. RFC No. 4340, 2006), *available at* http://tools.ietf.org/pdf/rfc4340; Jamshid Mahdavi & Sally Floyd, TCP-Friendly Unicast Rate-Based Flow Control

the analytical coherence of TCP-friendly solutions has been subject to increasing analytical attack.[262]

The growing popularity of streaming video and other UDP-based applications may also be making the need for core-based congestion management more important. When email and web browsing were the dominant forms of Internet traffic, one could plausibly argue that variation in applications' ability to tolerate congestion militated against incorporating congestion management into lower layers of the protocol stack. The growth of latency-sensitive applications as a proportion of network traffic has increased the justification for returning to a core-based approach to congestion control. Of course, network owners can also reduce congestion by increasing capacity. Whether increasing capacity or returning to a core-based approach is the more efficient solution is largely a matter of relative cost.[263]

The growing importance of wireless broadband technologies is also increasing the pressure on the status quo. Unlike wireline networks, wireless networks often drop packets for reasons other than congestion, such as when atmospheric conditions or reflections create a dead spot that limits the amount of bandwidth available or when a bad handoff between cell sites leads to a dropped transmission.[264] Thus, the advent of wireless broadband further undercuts the inference upon which the current system of congestion management is based. Also, unlike in wireline systems, which can simply increase raw capacity, the limited amounts of spectrum available for wireless broadband restrict bandwidth expansion to measures such as reducing cell size, and these measures are ultimately limited by Shannon's Law.[265]

(Jan. 1997) (unpublished manuscript), *available at* http://www.psc.edu/index.php/component/remository/Networking/Networking-Papers/TCP-Friendly-Unicast-Rate-Based-Flow-Control.

262 *See* Bob Briscoe, *Flow Rate Fairness: Dismantling a Religion*, COMPUTER COMM. REV., Apr. 2007, at 63, 65 ("Flow rate fairness was the goal behind fair resource allocation in widely deployed protocols like . . . TCP-friendly rate control. But it is actually just unsubstantiated dogma to say that equal flow rates are fair." (endnotes omitted)); *id.* ("To be realistic for large-scale Internet deployment, relative flow rates should be the *outcome* of another fairness mechanism, not the mechanism itself. That other mechanism should share out the 'cost' of one user's actions on others . . . ."); Jacobson, *supra* note 252, at 322 (cautioning that while TCP "algorithms at the transport endpoints can insure the network capacity isn't exceeded, they cannot insure fair sharing of that capacity"); M. Mathis, Rethinking TCP Friendly 5 (IETF Congestion Control Research Grp. Internet-Draft, Mar. 2009) (unpublished draft), *available at* http://tools.ietf.org/pdf/draft-mathis-iccrg-unfriendly-00.pdf (questioning the assumptions made by the TCP-friendly model).

263 *See* Yoo, *Beyond Network Neutrality*, *supra* note 227, at 22-23.

264 *See* Yoo, *supra* note 198, at 79 (noting that transmission errors, such as those involving interference problems, are frequent within wireless networks).

265 Shannon's Law holds that "the maximum rate with which information can be transmitted given limited bandwidth is a function of the signal-to-noise ratio." *Id.* at 78; *see also* Claude F.

Moreover, the solutions to managing congestion on wireless networks often deviate from the established layered architecture.[266] Some solutions employ a split TCP connection in which the sending host receives an acknowledgment from the wireless base station rather than the receiving host.[267] Others employ a snoop module solution in the base station (much like ARQ) that sniffs TCP packets as they pass by, sets a shorter retransmission timer than TCP, and then retransmits the packet if it does not receive a quick acknowledgment (making sure to discard any duplicated acknowledgments).[268]

C. *Distributed Optimization*

As noted earlier, one of the virtues of protocol layering is that it renders each layer largely independent of the others. This allows any entity operating at one layer to optimize its behavior without considering the impact on the overall system.[269]

Enabling distributed optimization provides all of the advantages discussed above,[270] but it is not without its drawbacks.[271] There is no guarantee that individual optimization decisions will necessarily lead to a globally optimal solution.[272] A few examples will demonstrate this point.

1. Aggressive TCP Implementations

Leading network engineers have long recognized that because no one feels responsible for thinking broadly about the Internet, actors have incentives to act in ways that promote their selfish best interests without regard to their impact on the overall system.[273] One classic case is aggressive TCP implementations.

Shannon, *Communication in the Presence of Noise*, 37 PROC. INST. RADIO ENGINEERS 10, 20-21 (1949) (explaining the theory).

266 Yoo, *supra* note 198, at 79.

267 *See* Ajay Bakre & B.R. Badrinath, *I-TCP: Indirect TCP for Mobile Hosts*, PROC. 15TH INT'L CONF. ON DISTRIBUTED COMPUTER SYS. 136, 137 (1995) (describing a method of accommodating the special requirements of mobile hosts by splitting connections in two).

268 Hari Balakrishnan et al., *Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks*, 1 WIRELESS NETWORKS 469, 471-72 (1995). Wireless backhaul through satellite, terrestrial microwave, or other spectrum-based technologies remains relatively rare, so such solutions are typically deployed only near the edges of the network.

269 *See supra* note 69 and accompanying text.

270 *See supra* Section I.C.

271 Yoo, *supra* note 42, at 26-27.

272 *See* RFC 3439, *supra* note 26, at 7-8; Crowcroft et al., *supra* note 26, at 23-24.

273 *See* Robert Braden et al., Developing a Next-Generation Internet Architecture 16 (July 15, 2000) (unpublished manuscript), *available at* http://www.isi.edu/newarch/DOCUMENTS/

a. *Refusal to Back Off in the Face of Congestion*

The host-based approach to managing congestion in the current layered architecture means that this regime depends on what amounts to the honor system.[274] To avoid congestion collapse, Jacobson's algorithm requires every host that detects network congestion to reduce exponentially the number of unacknowledged packets it permits to remain outstanding.[275] The lack of any effective governance mechanism gives rise to a classic cartel cheating problem; each host has the incentive to continue to send at a high rate and to depend on others to eliminate congestion by reducing their sending rates.[276] Although the Internet community may once have represented the type of close-knit community that could prevent such deviations from occurring, the rapid expansion of the Internet has undercut its ability to rely on social norms to protect against this type of behavior.[277]

b. *Multiple TCP Sessions*

Network engineers have also long recognized that a host can obtain a larger proportion of the available bandwidth simply by opening multiple

---

WhitePaper.pdf ("There is no commercial provider who believes that they [*sic*] hold the responsibility for the Internet architecture.").

274 *See* Charles L. Jackson, *Wireless Efficiency Versus Net Neutrality*, 63 FED. COMM. L.J. 445, 448-50 (2011) ("Users' systems must act altruistically, sacrificing their network service for the greater good, in order for these congestion control approaches to be effective.").

275 *See* Jacobson, *supra* note 252, at 318.

276 *See* PETERSON & DAVIE, *supra* note 4, at 470 ("[B]ecause the entire congestion-control mechanism is implemented at the sources and [first in, first out] queuing does not provide a means to police how well the sources adhere to this mechanism, it is possible for an ill-behaved source (flow) to capture an arbitrarily large faction of the network capacity."); Bob Braden et al., *Recommendations on Queue Management and Congestion Avoidance in the Internet* 9-11 (IETF Network Working Grp. RFC No. 2309, 1998), *available at* http://tools.ietf.org/pdf/rfc2309.pdf (noting that TCP implementations "can grab an unfair share of the network bandwidth" by aggressively refusing to back off in an attempt "to claim to have a 'faster TCP,'" which would logically lead to "a spiral of increasingly aggressive TCP implementations, leading back to the point where there is effectively no congestion avoidance and the Internet is chronically congested"); D. Papadimitriou et al., *Open Research Issues in Internet Congestion Control* 31 (IETF Network Working Grp. RFC No. 6077, Dmitri Papadimitriou ed., 2011), *available at* http://tools.ietf.org/pdf/rfc6077.pdf ("[C]ongestion control depends on parties acting against their own interests. It is not in a receiver's interest to honestly return feedback about congestion on the path, effectively requesting a slower transfer [or the] sender's interest to reduce its rate . . . if it can rely on others to do so."); M. Mathis, Relentless Congestion Control (IETF Congestion Control Research Grp. Internet-Draft, 2009), *available at* http://tools.ietf.org/pdf/draftmathis/iccrg/relentless-tcp-00.pdf (describing an alternative approach to congestion control that does not back off as much as the Jacobson-Karels algorithm).

277 YOO, *supra* note 32, at 17-18.

TCP connections.[278] The engineering community attempted to deter such behavior by recommending that hosts open no more than two simultaneous TCP connections,[279] but the lack of enforcement mechanisms meant that compliance depended entirely on the honor system. The problem with this approach is that it is individually rational for each host to deviate from the collectively rational solution.[280]

Browser manufacturers began defecting from this restriction in an attempt to obtain better performance than their rivals. The first was Netscape, which permitted hosts to open as many as eight TCP connections in order to download information in parallel.[281] Other new entrants, such as Mozilla Firefox, Opera, and Apple Safari, also configured their browsers to permit hosts to open eight connections.[282] Google's Chrome browser and

---

278 This problem was suggested in the very first specification of the host-to-host protocol in 1970. *See* Carr et al., *supra* note 106, at 590 (describing the assumption "that a user does not use multiple links to achieve a wide band"). Subsequent publications have frequently noted the problem. *See, e.g.*, KUROSE & ROSS, *supra* note 4, at 293 ("[T]here is nothing to stop a TCP-based application from using multiple parallel connections . . . . When an application uses multiple parallel connections, it gets a larger fraction of the bandwidth in a congested link."); Sally Floyd & Kevin Fall, *Promoting the Use of End-to-End Congestion Control in the Internet*, 7 IEEE/ACM TRANSACTIONS ON NETWORKING 458, 468 (1999) ("[T]he use of concurrent connections increases throughput for those applications that break a TCP connection into multiple connections . . . ."); RFC 2309, *supra* note 276, at 10 ("Note that there is a well-known way to achieve more aggressive TCP performance without even changing TCP: open multiple connections to the same place, as has been done in some Web browsers."); Sally Floyd, *Congestion Control Principles* 4 (IETF Network Working Group RFC No. 2914, 2000), *available at* http://tools.ietf.org/pdf/rfc2914 ("[T]o achieve more aggressive performance without even changing the transport protocol . . . , open multiple connections to the same place . . . . Thus, instead of a spiral of increasingly aggressive transport protocols, we would instead have a spiral of increasingly aggressive web browsers, or . . . applications.").

279 *See, e.g.*, R. Fielding et al., *Hypertext Transfer Protocol—HTTP/1.1*, at 47 (IETF Network Working Grp. RFC No. 2616, 1999), *available at* http://tools.ietf.org/pdf/rfc2616.pdf ("Clients that use persistent connections SHOULD limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy . . . . These guidelines are intended to improve HTTP response times and avoid congestion.").

280 *See* Floyd & Fall, *supra* note 278, at 463, 465 (discussing the lack of incentives to employ "cooperative methods of congestion control"); John Nagle, *On Packet Switches with Infinite Storage* 5 (IETF Network Working Grp. RFC No. 970, 1985) [hereinafter RFC 970], *available at* http://tools.ietf.org/pdf/rfc970 (using a game-theory view of datagram networks to illustrate how there can be a tragedy of the commons "in which the optimal strategy for each player is suboptimal for all players"); John Nagle, *Congestion Control in IP/TCP Internetworks* 2 (IETF Network Working Grp. RFC No. 896, 1984), *available at* http://tools.ietf.org/pdf/rfc896 (observing "suboptimal behavior by host TCP implementations").

281 Nelson Minar & Marc Hedlund, *A Network of Peers: Peer-to-Peer Models Through the History of the Internet*, *in* PEER-TO-PEER: HARNESSING THE BENEFITS OF A DISRUPTIVE TECHNOLOGY 3, 12 (Andy Oram ed., 2001).

282 Dylan Schiemann, *IE8: 6 Connections Per Host*, COMET DAILY (Mar. 5, 2008), http://cometdaily.com/2008/03/05/ie8-6-connections-per-host.

Microsoft's Internet Explorer 8 browser showed greater restraint, limiting hosts to opening six connections.[283] In 2009, Firefox upped the ante, reconfiguring its browser to permit hosts to open up to 15 TCP connections.[284]

c. *Autotuning*

A new form of aggressive flow is associated with a feature commonly known as autotuning. Every TCP implementation signals to sending hosts the maximum data rate it is willing to accept by identifying in an advertised window the available buffer space in every acknowledgment.[285] Because the field in the TCP header for the advertised window is sixteen bits wide, the maximum amount of traffic that the receiving host could acknowledge was long thought to be 2^{16} or 65,536 bytes, which equates to a rate of 5.24 Mbps in a path with a roundtrip time of 100 milliseconds.[286]

Although this size was appropriate for the constraints of the narrowband Internet, the carrying capacity of modern broadband networks has rendered this limitation obsolete. Modern TCP implementations have used autotuning to overcome this limitation without deviating from the semantics of IP.[287] TCP implementations that apply autotuning can use a new feature to increase the size of the advertised window by a factor of 2^{14} (more than sixteen thousand times larger). With that capacity, the receiver can advertise windows of more than 1 billion bytes,[288] which can increase throughput rates on a path with a roundtrip time of 100 milliseconds to increase transmission rates to over 85 Gbps.

Long built into Linux and Apple O/S, autotuning has now been incorporated into Windows Vista and Windows 7.[289] TCP implementations that lack autotuning capabilities will unilaterally constrain the amount of traffic sent to the size of the available buffer advertised by the receiver, which is limited to 65,535 bytes without autotuning.

---

283 *Id.*; *see* Alsciende, Answer to *Max Parallel http Connections in a Browser?*, STACKOVERFLOW (June 12, 2009, 8:59), http://stackoverflow.com/questions/985431/max-parallel-http-connections-in-a-browser (listing the number of permitted connections for various browsers).

284 *See Issue 12066: Match Firefox's Per-Host Connection Limit of 15*, CHROMIUM (May 15, 2009), http://code.google.com/p/chromium/issues/detail?id=12066.

285 COMER, *supra* note 4, at 198; Walter Willinger & John Doyle, Robustness and the Internet: Design and Evolution 11 (2002) (unpublished manuscript), *available at* http://netlab.caltech.edu/publications/JDoylepart1_vers42002.pdf.

286 Joseph Davies, *TCP Receive Window Autotuning*, TECHNET MAG. (Jan. 2007), http://technet.microsoft.com/en-us/magazine/2007.01.cableguy.aspx.

287 V. Jacobson et al., *TCP Extensions for High Performance* 5, 8 (IETF Network Working Grp. RFC No. 1323, 1992) [hereinafter RFC 1323], *available at* http://tools.ietf.org/pdf/rfc1323.

288 *Id.* at 11.

289 Davies, *supra* note 286, at 4-5.

Autotuning raises two basic problems. First, previous TCP implementations naturally constrained bandwidth usage by not allowing more than 65 thousand bytes per session to be in transit. Implementations running autotuning are much less constrained. To the extent that this transmission rate exceeds the carrying capacity of the network, new implementations will increase their sending rates until they create congestion somewhere in the network. Second, to the extent that implementations with and without autotuning operate on the same network, the newer implementations will consume a disproportionate amount of the bandwidth,[290] providing another example of interactions between the transport and the data-link layers.

Curbing these opportunistic behaviors may require a network-based solution operating in the lower layers that identifies and limits the behavior of aggressive transport protocols. Leading examples include Weighted Fair Queuing (WFQ)[291] and flow-valve mechanisms such as the RED "penalty box."[292] To the extent that these solutions require the inspection of transport layer data and penalizing certain transport layer implementations, they are inconsistent with protocol layering.

2. Simultaneous Optimization

Another classic problem arises when two different layers attempt to optimize the same parameter. The fact that both layers attempt to make adjustments at the same time will interfere with the feedback that each is receiving. This is precisely the concern that is the focus of the longstanding literature critical of layering.[293]

---

290 *See* Geoff Huston, *A Decade in the Life of the Internet*, INTERNET PROTOCOL J., June 2008, at 7, 13; Video, *Changing Technology and the Limits of the Layered and End-to-End Models*, CTR. FOR TECH., INNOVATION AND COMPETITION, U. PA. L. SCH. (May 6, 2010), https://www.law.upenn.edu/institutes/ctic/conferences/internet-policy.php. I personally experienced this phenomenon when I added a new machine running Windows 7 to a home network that consisted of machines running Windows XP. Until I disabled the autotuning feature on the new machine, its relative lack of restraint caused downloads to all other machines to slow to a crawl whenever the Windows 7 machine was downloading Internet content.

291 Floyd & Fall, *supra* note 278, at 459.

292 Kenjiro Cho, *Flow-Valve: Embedding a Safety-Valve in RED*, 3 PROC. GLOBAL TELECOMM. CONF. 1753, 1754 (1999).

293 *See supra* note 26 and accompanying text. Barbara van Schewick similarly notes the problems posed by the distributed nature of optimization. VAN SCHEWICK, *supra* note 7, at 43-44. Her subsequent discussion focuses on the potential downsides of more integrated optimization in terms of monopoly profit and innovation effects without discussing the potential efficiency benefits. *Id.* at 152-63.

On some level, the recent dispute between Comcast and BitTorrent can be understood through this lens.[294] BitTorrent attempts to maximize throughput by opening up hundreds of TCP connections and searching for the five with the fastest links.[295] Comcast, for its part, initially attempted to manage congestion by targeting the hosts' transport layers by sending TCP resets.[296] Comcast's current implementation operates exclusively at the data-link layer, detecting when portions of the network are congested and temporarily slowing down the traffic from the heaviest users that are creating the congestion.[297] As a result, both the data-link and the transport layers may both adjust flows simultaneously to optimize throughput. Moreover, layering permits BitTorrent to focus solely on optimizing its own operations, which may or may not coincide with the outcome Comcast was pursuing or the global optimum.

Network engineers have begun to use unified theories of distributed optimization to evaluate the performance of the existing Internet. They have concluded that the existing system of layers is only one of several possible solutions to the problem and have used the distributed optimization framework to identify cross-layer structures that may perform better.[298] These solutions, however, necessarily diverge from the layered architecture by making information visible to other layers that would otherwise remain hidden.

3. Other Considerations

The net result is that the emergence of wireless broadband, the growing importance of UDP-based applications, and attempts by upper layers to avoid interdependencies are putting pressure on the existing layered architecture. Waiting in the wings are new applications such as cloud

---

294 *See* Comcast Corp. v. FCC, 600 F.3d 642, 645 (D.C. Cir. 2010) (adjudicating Comcast's appeal of an FCC ruling that Comcast could not manage its network in such a way as to discriminate against peer-to-peer sharing services using the BitTorrent protocol).

295 R. Penno et al., *LEDBAT Practices and Recommendations for Managing Multiple Concurrent TCP Connections* 4-6, 11 (IETF Transport Working Grp. Internet Draft, 2010), *available at* http://tools.ietf.org/pdf/draft-ietf-ledbat-practices-recommendations-00.pdf.

296 Formal Complaint of Free Press & Public Knowledge Against Comcast Corp. for Secretly Degrading Peer-to-Peer Applications, 23 FCC Rcd. 13028, 13029 para. 3 (2008) (memorandum opinion and order).

297 Letter from Kathryn A. Zachem, Vice Pres., Regulatory & State Legislative Affairs, Comcast Corp., to Dana Shaffer, Chief, Wireline Competition Bureau, and Matthew Berry, Gen. Counsel, FCC (Jan. 30, 2009), *available at* http://apps.fcc.gov/ecfs/document/view?id=6520194593.

298 *See, e.g.*, Mung Chiang et al., *Layering as Optimization Decomposition: Framework and Examples* ("'Layering as optimization decomposition' is a unifying framework for understanding and designing distributed control and cross-layer resource allocation in wired and wireless networks."), *in* IEEE INFO. THEORY WORKSHOP 52, 56 (2006).

computing that may demand different functionality and may require a remodularization of the layered stack.[299]

D. *Security*

Another development pressuring the layered model is the growing need for security. Although IP datagrams include a source IP address in their header, that information is not reliable and can easily be forged.[300] Although solutions to this problem exist, they have not yet been widely deployed.[301] Moreover, the TCP/IP Reference Model does not allow end users to verify the paths along which a particular communication has traveled. Although verification was less problematic when the Internet remained a close-knit community, the growth in the size and heterogeneity of end users is placing increasing emphasis on the importance of trust.[302]

Although network-based features such Internet Protocol Security (IPSec) and Domain Name System Security (DNSSec) exist, security and identity verification have been regarded primarily as the responsibility of the hosts.[303] More recently, responsibility for some security-related functions has begun shifting to third-party proxies operating in the network's core rather than the host.[304] The shift of these functions into the network's core is consistent with the principle of locating particular functions where the information needed to perform those functions resides.[305] Accurate spam detection, for example, often requires the ability to examine email destined for many different users. Modern approaches to botnet detection often examine the DNS queries being submitted by a large number of hosts

---

299 For a discussion of cloud computing, see Yoo, *supra* note 198, at 83-86.

300 *See* David Moore et al., *Inferring Internet Denial-of-Service Activity*, 24 ACM TRANSACTIONS COMPUTER SYS. 115, 118 (2006) (noting that "many attackers forge, or 'spoof,' the IP source address of each packet they send").

301 J. Touch et al., *Problem and Applicability Statement for Better-than-Nothing Security (BTNS)* 7 (IETF Network Working Grp. RFC No. 5387, 2008), *available at* http://tools.ietf.org/pdf/rfc5387.

302 *See* YOO, *supra* note 32, at 17 ("[T]he universe of end users has become less trustworthy, as reflected by the increased frequency of spam, viruses, invasions of privacy, and other forms of malicious behavior."); Marjory S. Blumenthal & David D. Clark, *Rethinking the Design of the Internet: The End-to-End Arguments vs. the Brave New World*, 1 ACM TRANSACTIONS ON INTERNET TECH. 70, 75-76 (2001) (describing the breakdown in trust on the Internet); RFC 3724, *supra* note 244, at 6 ("Perhaps the single most important change from the Internet of 15 years ago is the lack of trust between users.").

303 *See* Scott Bradner, *The End of End-to-End Security?*, IEEE SECURITY & PRIVACY, Mar./Apr. 2006, at 76, 77 (noting that under the current architecture "security and privacy are the responsibilities of the end nodes," not the network); Saltzer et al., *supra* note 121, at 282-83 (arguing that security by encryption is properly the responsibility of the end nodes).

304 YOO, *supra* note 32, at 90.

305 *See supra* notes 248-50 and accompanying text.

to identify the bot master.[306] Increased reliance on third-party proxies stems from the fact that many end users no longer trust their own machines. In what computer scientists Marjory Blumenthal and David Clark call "the ultimate insult," these end users may trust a third-party proxy residing in the network more than they trust the computer sitting on their desk.[307]

These considerations have increasingly led end users to look to network-based solutions to provide security.[308] Many of these security implementations violate the principles of protocol layering. Firewalls are core-based technologies that examine transport-layer information—such as port numbers—to determine which information to filter.[309] Spam detection often requires that the middlebox operating in the network layer pass beyond the transport layer to examine email at the application layer in order to detect spam. In addition, major ISPs routinely sample the traffic passing through their network and use deep packet inspection (DPI) to examine it for security threats.[310] Each of these practices represents a violation of the layering principle prohibiting devices in the network's core from examining information associated with the transport or application layers.

In addition to its security benefits, DPI can also enhance a network's functionality. For example, Plusnet uses DPI to divide the data stream into multiple levels of priority.[311] Prioritizing traffic in this manner has enabled Plusnet to win numerous industry awards for the quality of its network connections and for customer satisfaction.[312] Despite these benefits, however, using DPI to examine content in the core of the network represents a clear violation of layering.

Most radically, the National Science Foundation's Future Internet Architecture initiative places greater emphasis on designing architectures in which identity and path verification are inherent properties of the layered stack rather than a feature added after the fact.[313] A good example of this

---

306 YOO, *supra* note 32, at 91.

307 David D. Clark & Marjory S. Blumenthal, *The End-to-End Argument and Application Design: The Role of Trust*, 63 FED. COMM. L.J. 357, 379 (2011).

308 *Id.* at 375-78.

309 Steven M. Bellovin & William R. Cheswick, *Network Firewalls*, IEEE COMM., Sept. 1994, at 50.

310 Comments of Christopher S. Yoo 6-7 (2010), *available at* http://fjallfoss.fcc.gov/ecfs/document/view?id=7020919810, *commenting on* Preserving the Open Internet, 24 FCC Rcd. 13064, GN Docket No. 09-191 (2009) (notice of proposed rulemaking).

311 *Broadband: All About Traffic Management*, PLUSNET, http://www.plus.net/support/broadband/quality_broadband/traffic_prioritisation.shtml (last updated Dec. 19, 2012).

312 *See Award Winning Broadband and Quality Customer Service*, PLUSNET, http://www.plus.net/press/awards.shtml (last visited Apr. 10, 2011).

313 *See* Press Release, Nat'l Sci. Found., NSF Announces Future Internet Architecture Awards (Aug. 27, 2010), *available at* http://www.nsf.gov/news/news_summ.jsp?cntn_id=117611

limitation is network security. The current architecture does not permit verifiable information about the identity of particular end users to pass through the protocol stack.[314] Despite the growing need for security caused by the increasing size of the Internet and its utilization for increasingly delicate tasks, the network has been slow to adapt to this new reality. The shifting importance of these concerns emphasizes the importance of not regarding any particular layered architecture as if it were a natural construct. Moreover, it underscores the potential dangers of using regulation to enshrine any particular architecture into law.

CONCLUSION

Layering has emerged as a popular way to analyze emerging issues of Internet policy. In addition to providing a more integrated and functionally oriented alternative to the approach enshrined in the Communications Act of 1934, under which each mode of transmission is treated as a regulatory universe unto itself, layering conforms to the manner in which the engineering community views the network. Layering also plays a key role in making the complexities of network management more tractable. Indeed, it is hard to see how one would solve such a complex engineering problem as the Internet without it.

Policymakers should not forget the engineering literature that analyzes circumstances under which layering can lead to suboptimal outcomes. Although layering is designed to facilitate interconnection and promote innovation by modularizing clusters of tasks and thus making them independent, it comes at a cost of reduced functionality and efficiency. Moreover, like all forms of modularity, layering works by information hiding and by minimizing the extent to which interdependencies cross module boundaries. As such, any set of layered protocols reflects a preexisting vision of how layers should interact and which interdependencies matter. Changes in the technology and environment surrounding the Internet are putting new pressures on those commitments. Rather than mandating access to the existing interfaces or opposing practices that deviate from the existing architecture, policymakers should adopt a more dynamic perspective that allows for the possibility that the optimal layered structure may change over time.

The dynamic way that the layered model evolved also underscores that network engineering is an inherently pragmatic enterprise that is ill suited

(announcing awards for projects that, for example, explore ways to "incorporate adequate mechanisms to support secure content-oriented functionality" and to "bridg[e] the gap between human and intrinsically secure identifiers").

314 *See supra* note 300-02 and accompanying text.

to broad, categorical generalizations or claims of inviolability. Indeed, no idealized architecture is inherently superior.[315] The optimal architecture depends instead on the shape of the particular flows passing through the network, on the value that end users place on particular services, and on the relative costs of network resources. The advent of Internet-based video, wireless broadband, new architectural features, and security-related concerns transforms the technological and economic environment surrounding the Internet. This transformation creates natural pressure on the layered stack to evolve in response.

As of now, those participating in policy debates do not have a working understanding of many fundamental principles around which the Internet is organized. For example, policymakers would benefit from having a basic grasp of how congestion is managed on the Internet, which areas the engineering community regards as settled, and which are regarded as controversial. Debates over controversial engineering principles are often heated, as is the case in any academic discipline; as a result, debates about layering at times engender strong assertions of diametrically opposed views. Sensible Internet policy depends on the participants in policy debates having a sufficient appreciation of the issues and positions in these debates to take the full range of views presented in the engineering literature into account.

315 RFC 817, *supra* note 5, at 2 ("[T]here may be no such thing as a successful general purpose protocol implementation.").