Vertical and Horizontal End-to-End Arguments in the Internet

Matthias Bärwolff
mbaer@csail.mit.edu

***Abstract*—We propose a novel taxonomy for the end-to-end arguments associated with the internet: a vertical and horizontal version. While the former captures the trade-offs of where to place functions in a vertical network protocol stack, the latter deals with the trade-offs of spreading them across different hosts. Those two versions of the end-to-end argument were effectively one and the same in the early days of the internet. However, this congruence is now thoroughly broken, and it is thus useful to separate the two conceptually. Primarily, this helps framing the broader debate on network design in light of tussles between the internet's various stakeholders and higher order objectives such as user choice and fostering innovation.**

I. Introduction

The thesis of this paper is that there is considerable merit in explicitly distinguishing between two versions of the end-to-end argument, one vertical and the other, possible more important one, horizontal. The implications of the two are materially different. While the former addresses the balance of responsibilities for functions in a vertical protocol stack model and concludes that the shared network layer should carry as few functions as possible, the latter addresses the balance of functions in a horizontal sense, often across stakeholders. Unlike some interpretations of the end-to-end argument that consider it with respect to higher level objectives such as end user choice, empowerment, and innovation, our horizontal version does explicitly allow and call for functions to be delegated to where they maximise the utility functions of the end points initiating and terminating an application.

The paper proceeds as follows. First, we look at the context of the original end-to-end argument, and the second order properties of the end-to-end argument that have quickly come to dwarf its original scope. Against this background, we then go on to elaborate the notions of vertical and horizontal end-to-end arguments. We close the main body of this paper with a discussion of how tussle isolation as a design principle relates to the horizontal end-to-end argument. Finally, we point to related work, possible further research, and conclude.

II. The Original End-to-End Argument

In the early 1980s the internet as we know it today did not exist. Its scope was almost microscopic, there was no WWW, the ISO/OSI protocol framework was still contesting TCP/IP, and most of the useful and openly accessible interconnections took the form of email, NNTP (network news), and BBS (bulletin board system) application layer gateways rather than IP routers. However, the Arpanet had been around for more than a decade, and some experience was beginning to evolve about what to expect from a packet network.

With respect to the end-to-end arguments [1],[1] the first and foremost realisation in the early Arpanet was that making the what was then called subnetwork provide a reliable transport service was next to impossible. The ARPA contractor BBN who were charged with building the network part of the Arpanet found it beyond their means to deliver such a network performance given the myriad of factors affecting the success of a packet transfer over a network. The Arpanet completion report [3] concludes:

> It was originally thought that the ARPANET would lose a message so seldom that there was no point in hosts ever bothering with message retransmission. Unfortunately, resolving various possible lockups has required the subnetwork to discard a message occasionally, and the topology of the network has evolved into long series of machines and lines that increase the probability of involuntary message loss. However, the host-to-host protocol followed the initial thought and did not provide for message retransmission. Given the realities of the probability of message loss in the network and given the host-to-host protocol which is inordinately sensitive to any abnormality, the host-to-host protocol (and protocols based on it) has proven quite unreliable. (pp. III-52)

It was upon this realisation that the research sites connected to the Arpanet were compelled to build protocols that would implement a reliable data transport over an inherently unreliable network, an effort which resulted in the implementation of TCP, a very complex common transport protocol at the end systems [4], later split into IP [5] and TCP [6]. The best effort service of the internet may thus be interpreted as an unintended result of the failure to build a reliable subnetwork rather than a conscious design effort.

It is no coincidence that the example elaborated in the original paper on end-to-end [1] is that of reliable file transfer, an application which had evidently been impossible without end-to-end reliability checks. The objective of the original formulation of the end-to-end principle had been to solve the engineering trade-off of building a reliable network from an unreliable subnetwork, and not to mandate each and every

[1] As an aside, an earlier version of the 1984 paper [1] was presented 1981 in Paris, France [2].

978-1-4244-3437-4/09/$25.00 ©2009 IEEE

function of the network and its applications to be implemented by host level protocols. It was explicitly realised that the actual choice of where to put a given function had to be made weighing the respective costs of that function, and considering the commonality of the function in question for applications building on the network layer. David Clark thus notes that it is not the end-to-end argument that militates against functions at the internet layer, but rather the lack of those functions that gave rise to the end-to-end arguments:

> The discovery of packets is not a consequence of the end-to-end argument. It is the success of packets that make the end-to-end argument relevant. [7, slide 31]

Put differently, it is not primarily because not all applications need perfectly reliable data transfer, but much rather because providing such a function at the shared network layer is ludicrously expensive, that the function of reliability sits best with the end hosts.[2] In a related publication, Clark argues that the scope of the original end-to-end argument must not be overstretched:

> There is no reason to believe that the original reasoning about an unreliable communications subsystem makes any sense at the application level. [8, p. 15]

This application level, it turns out, is precisely what we mean by horizontality. And, Clark's observation gives us a first indication that the reasonings that apply to a low-level notion of end-to-end considerations must not by confused with those that apply to horizontal issues involving applications and policy negotiations that stretch across multiple players with incompatible values and intentions.

III. User Choice and Innovation

Over time the end-to-end argument became closely associated with "second-order properties" that go well beyond the original considerations of reliability and flexibility to include "concerns of maintaining openness, increasing reliability and robustness, and preserving the properties of user choice and ease of new service development" [9]. However, this new set of objectives has proven more elusive in its implications about the placement of functions than the more tractable and straightforward reliable file transfer example that very much captures the early end-to-end argument.

By way of analogy with the original "narrow" end-to-end argument it has been proposed that an end-to-end argument serving higher level objectives such as innovation and end user sovereignty be conceived as a "broad version" of the end-to-end argument [10]. The broader scope of this interpretation, it has been argued, entails a much stricter application of the end-to-end argument's preference for having application specific functions sit with the end points [10]. One well-known incarnation of this argument is the position of network neutrality, with the metaphor of neutrality very much determining the agenda, leading to rather strict conclusions about the role of intermediaries between end points. However, the notion of network neutrality has virtually nothing to do with internet design principles or end-to-end arguments per se [11].[3] Again, it is worth quoting Clark who states that applying low level end-to-end arguments to application level issues, may, in fact, "be nonsense" [8, p. 15].

IV. Vertical and Horizontal End-to-End Arguments

We are now ready to draw towards the main point of this paper, the assertion that there is an important distinction between two versions of the end-to-end arguments, one vertical and the other horizontal, each with a different scope and different implications.

A. Vertical End-to-End Arguments

The vertical end-to-end argument addresses the separation of functions in a vertical protocol stack in which one layer acts as a universal spanning layer. It argues — effectively as a complement to the explicit goal of meeting the lowest common denominator of as many as possible types and shapes of networks —[4] that functions shall only be placed at the internet layer if they allow as many different applications as possible to be based on it. The somewhat counter-intuitive conclusion here is that preferably there be as few functions as possible, for any function that is not needed for the operation of the internet layer itself is deemed to be specific to certain applications and may thus be implemented by those applications at the application layer.[5] In fact, even if those functions were provided in full by the network layer, the fact that they are partly implemented on entities that are physically, and often administratively, distinct from the two logical end points, renders them from the viewpoint of those ends potentially futile. The operations of the internet layer cannot be trusted to be performed to the standards of the end points, thus they have to implement them themselves, anyway, should they require them.

For the purpose of our argument, the important point is that the vertical version of the end-to-end arguments addresses the question of which functions to provide at the network layer, the only module in the internet protocol stack that by definition is shared among all its participants. It concludes that those functions should be as few as possible, and even those should not be trusted by the end points that run applications on the internet.

[2] The corollary of this statement is that if the network, for whatever reason, was more reliable than the end hosts, and the cost and overhead of providing perfect reliability was small enough, then that function would better sit with the network.

[3] Arguably, the network neutrality position may be best characterised not as a rigorous academic exercise, but as a useful starting point in a bargaining game.

[4] This argument applies specifically to QoS concerns, for meeting QoS parameters such as delay, jitter, and throughput will inevitably depend on the capabilities of the lower layers close to the physical medium of networking (Nature of Service, or NoS). Hence there is little reason for specifying such functions at a common internet layer that aims at allowing the internetworking of as broad a variety of networks as possible [12, pp. 43 f.].

[5] While experience and common sense have proven that it is sensible for applications to delegate many of these functions to what has become the transport layer [13], logically these functions are part of the application, not the network layer.

B. Limits of Vertical End-to-End Arguments

The vertical end-to-end argument does not address the issue of how functions should be placed at the application layer itself, who should control them, and where among a set of stakeholders they should be run. In the very early internet those questions were moot, since the stakeholders were mostly aligned in their interests,[6] and most applications — narrowly conceived — consisted of simple file transfer between two end points or remote access from one end point to another. However, even then — in a higher level sense — many application scenarios consisted of more than two end points implementing the application in its entirety.

Most notably, email quickly took a shape that gave emphasis to user agent protocols such as POP and IMAP which would operate horizontally at the application layer and handle many application functions on behalf of the end user [15]. And, today, most applications depart markedly from the textbook model in which an application encompasses but two end points who also assume all the application specific functions without delegating a single function elsewhere. While one may argue that third parties are ends themselves and thus such delegation of functions is perfectly compatible with a lower level notion of end-to-end principles, from the point of view of the actual ends, those third parties are intermediaries in a higher level sense — very much like routers in a lower level context.

C. Towards Horizontal End-to-End Arguments

In light of the issues raised above it is useful to formulate an explicit horizontal version of the end-to-end argument. Such an argument aims not at addressing the question of where to put functions of networking in a vertical protocol stack, but the questions of tussles, empowerment and end user choice associated with having functions and conflicting interests in them spread beyond the ultimate end points.

Note that we are not implying that the notion of horizontal end-to-end arguments is a new one; in fact, horizontal concerns had been addressed by the original paper [1], too. Note, also, that we are in no way challenging the principal conclusion of the vertical end-to-end argument: statistical aggregation of data traffic entails inherently unreliable networks (no buffer is infinite) which in turn leads to the conclusion that functions such as reliable transport are best implemented by host protocols above the internet layer.

However, we feel that an explicit conceptual distinction between the two versions allows us to more clearly address questions of horizontal tussles and interdependencies that were simply not relevant back in the early 1980s. Without a horizontal notion of end-to-end arguments we are ill-equipped to address the host of issues that arise when layers above IP are increasingly shared — either by application design, or by packet inspection on the part of third party intermediaries — across stakeholders with possibly diverging interests [16].[7]

The core of the horizonal end-to-end argument is the trade-off between putting application level functions with the end points or rather with intermediaries in the network that are in the path (physical or logical) between the two ultimate end points. It is not adventurous to premise the futility of insisting on all application level functions (approaching, effectively, the entirety of functions there are) to be under the immediate control of the ultimate end points. Thus the questions arise: where to put them, whom to put in charge, and which remedies to apply?

Returning briefly to our example of email as an early application with horizontal scope beyond two ultimate end points, the vertical end-to-end argument would argue that email is in perfect fit with the end-to-end argument's normative emphasis on having functions in the higher layers, and thus address none of our questions. A horizontal interpretation, however, would come to a diametrically opposite conclusion: email has vital functions not with the actual end points but with intermediaries in between the two eventual end points.

Does this mean that the application design of email is in violation of end-to-end principles? We postulate that *as long as the intermediaries perform their functions to the standard of the ultimate ends, with the expected net benefit from doing so exceeding that of internalising the functions by those ends*, the answer will be no. The functions that are logically between the two ends can be regarded as having been delegated by those ends. Thus we can reduce the relevant trade-offs to a simple utility function of the ends that use the application. Note that this is not possible with the vertical version of the end-to-end arguments, due to the potentially large externalities imposed upon other applications by optimising the shared internet layer for specific applications. The horizontal end-to-end argument, on the other hand, does not involve such externalities, since functions in between two end points and at layers above IP do not affect other applications built on IP in the same way that adding functions to the IP protocol itself would. From a horizontal viewpoint we are thus free to migrate functions away from the ends.[8]

D. Tussles and the Horizontal End-to-End Arguments

In an already classic paper Clark et al. [21] argue that tussles, the "ongoing contention[s] among parties with conflicting interests" (p. 462), have, by virtue of its openness, become a defining characteristic of the internet and its applications that had better be accounted for in their design: Where there is no scope for resolving tussles, the conflicts between the diverse stakeholders in the internet will spill across module boundaries

[6]Even though sometimes congruence of interests was a matter of tight control rather than laissez faire [14], many of today's areas of conflict did not affect the early internet.

[7]As an aside, within the notion of horizontal end-to-end arguments we do not depend on any concept of layering altogether, which incidentally makes that framework much easier to apply to ideas of layerless networking that have been floating around in recent years [17], [18].

[8]Some would argue that there are still sufficient externalities that would militate against placing substantial functions away from the ultimate ends [19]; however, others have more plausibly argued that the dynamics of the internet are largely exogenous from the idiosyncrasies of individual end user preferences and applications [20].

in unfortunate ways, much like subsequent changes in a badly modularised software system will affect not only one module, but possibly all of them, making for nightmares in maintaining integrity, changeability and comprehensibility [22].

However, the internet was originally designed with little if any of today's tussles in mind; after all, its central standards were not designed with the anticipation of the internet ever becoming the universal de facto data network standard architecture in the first place [12, p. xiv]. In fact, the mechanisms of the IP protocol have rather been designed for an environment with no tussles at all — a poor fit for any market and competition based evolution of the system.

Plus, in practice, isolating tussles by modularising along tussle boundaries is no trivial endeavour, even if the tussle spaces were well understood. Mapping logical modules to an actual implementation is hard enough for software systems controlled by one player; accordingly, it is even harder for systems that are distributed among a potentially unlimited number of stakeholders. Most importantly, there is a tension between vertical layering of functions, and horizontal interactions between players necessitating layer violations, transparency violations, and ad hoc hacks — often on completely reasonable and innocent technical grounds [16].

Put differently, the essential paradox of the internet with regard to tussles is that the lack of mechanisms in the IP protocol (the neck of the hourglass), while allowing for a great variety of networks, applications, and thus stakeholders to partake in the global internet, severely limits the ability of those stakeholders to negotiate policies at that common layer. Thus, over the years, the internet has developed a formidable control plane that sits above IP and addresses many of the tussles, but also serves to fatten the internet layer logically, in turn undermining its narrowness that gave rise to the tussles in the first place [23].

How can this trade-off be resolved in a most beneficial way? The authors introducing the tussle design notion [21] argue that there should be "value-transfer mechanisms" in order to allow for fine-grained payment rewards for ISPs' investments in network infrastructure and QoS services on the one hand, and a "mechanism whereby the user can exercise choice to select the provider who offered the service" (control over routing) on the other hand (p. 473).

However, designing mechanisms that allow bargaining processes between stakeholders and end user controlled choice of routes among competing network providers is in itself not sufficient to create a normatively desirable market situation. Without a mechanism by which the tussle over *the actual value of the network uses* can be addressed, the situation only resembles a "market" in which all goods are wrapped in perfectly identical boxes — not exactly the kind of efficiency that society tends to value most.

The tussle over the value of using the network originates, of course, at the application layer, for it is applications, not the network by itself, that create value. However, this layer ordinarily leaves no more than opaque payloads at the internet layer. Network providers (and governments, for that matter) are, by virtue of the internet's design, not required and, arguably, not meant to evaluate the content and value that accrues at the application layer. Either way, there is no built in mechanism that allows for tussling on this. Hence deep packet inspection, and hence the desire of network providers to build "next generation networks" which bestow them with greater control over the application layer. Hence encryption of application data, and hence the call for legal sanctions against discriminating applications, aka network neutrality.

Tussles over the surpluses that applications derive from the networks are bound to remain an important feature of the internet.[9] They better be addressed with considerations about the design of mechanisms to channel them within useful boundaries, rather than worries about architectural pureness and a presumption of malign intents of network owners.

The notion of horizontal end-to-end arguments at the application layer helps frame the reasoning about the due trade-offs and design decisions, since the mechanisms that best aid the process of tussling between the various stakeholders will necessarily have to reach up to the application layer, and be horizontal in nature. Plus, it will provide conceptual clarity, and help avoid false analogies that foreclose value-neutral solutions which may involve affording network operators with some leverage over the application layer.

V. Related Work

We are not the first to point to the importance of intermediaries and the horizontal nature of end-to-end arguments. But, to our knowledge, no-one has yet proposed an explicit distinction between vertical and horizontal end-to-end arguments.

Some of the relevant references regarding our horizontal notion of the end-to-end arguments have immediately informed the above discussions. We have considered the notion of tussles [21], particularly how it addresses the horizontal tensions with the vertical protocol layer model of the internet. The notion of trust, and how it relates to the end-to-end arguments is slowly emerging in the literature. There have been considerations of trust as a criterion of where to place functions, both vertically [13] and horizontally [8]. We note that our framework is more general, in that trust is not the ultimate determinant of the place of functions. In our conception, trust is only one of the factors that determine the respective utilities of the various choices for placing functions.

The idea of putting functions in the network, on behalf and under the control of end points, has been put forward and discussed under the heading of active networking [24]. In particular the point has been made that function delegation serves, and not violates, end-to-end principles [24], [25]. The focus of active networking, however, rests largely with augmenting the functions of the IP layer, and it has had no sizeable impact upon networking thus far.

[9] As an aside, nationalising the internet would not only leave the tussles over applications and content in place, but also do away with competition and innovations in internet services, and thus hinder application level innovations and access, too. It is useful to remember that the internet only really took off when ownership and control were ceded over to private parties in the 1990s.

Finally, it should not go unmentioned that considerations about distributed routing [26], error control [27], access control, accounting, etc. have always implicitly dealt with the horizontality of their problems.

VI. Limitations and Further Research

Our framework of vertical and horizontal end-to-end arguments does little more than laying out the basic idea, and could well use some further elaboration. It may be interesting to extend the notion of horizontal end-to-end arguments to explicitly conceive humans as the horizontal end points. Such a conception would emphasise function delegation — notions of nomadic computing spring to mind [28].

Also, it would be interesting to frame questions about the balance of functions by considering the internet not as a core with ends attached to it, but rather as a fully connected mesh of peers. After all, the notion of end hosts under control of end users becomes increasingly futile in a setting where most users cannot even trust their own machines anymore [8].

VII. Conclusion

This paper has introduced a classification of the end-to-end arguments into a vertical and a horizontal version. There are important conceptual differences between the two, that directly affect the implications drawn in various contexts. Specifically, horizontal end-to-end arguments are better suited to inform considerations of tussle isolation among administratively disparate stakeholders in the internet. Also, unlike the vertical end-to-end arguments, they explicitly acknowledge and speak to the the importance of horizontal delegation of functions.

Any meaningful discussion of higher level objectives such as innovation and end user sovereignty will benefit from appreciating the difference between vertical and horizontal end-to-end arguments.

Acknowledgment

I gratefully acknowledge the helpful comments of Kei Ishii, Robert A. Gehring, and the four anonymous reviewers on a very early version of this paper. Sincere thanks to all.

References

[1] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions in Computer Systems*, vol. 2, no. 4, pp. 277–288, 1984, http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf.

[2] ——, "End-to-end arguments in system design," in *Proceedings of the Second International Conference on Distributed Computing Systems, Paris, France, 1981*. IEEE Computer Society, 1981, pp. 509–512.

[3] F. E. Heart, A. McKenzie, J. M. McQuillan, and D. C. Walden, "ARPANET completion report," Bolt Beranek and Newman Inc. (BBN), Report 4799, 1978, http://www.cs.utexas.edu/users/chris/DIGITAL_ARCHIVE/ARPANET/DARPA4799.pdf.

[4] V. G. Cerf and R. E. Kahn, "A protocol for packet network interconnection," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 637–648, 1974, http://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf.

[5] J. Postel, "Internet Protocol: DARPA internet program protocol specification," RFC 791 (Standard), 1981, http://tools.ietf.org/html/rfc791.

[6] ——, "Transmission Control Protocol – DARPA internet program protocol specification," RFC 793 (Standard), 1981, http://tools.ietf.org/html/rfc793.

[7] D. D. Clark, "Application design and the end-to-end arguments," MIT, Presentation at MIT Communications Futures Program Bi-annual meeting, May 30–31, 2007, Philadelphia, PA, 2007, http://cfp.mit.edu/events/may07/presentations/CLARK%20Application%20Design.ppt.

[8] D. D. Clark and M. S. Blumenthal, "The end-to-end argument and application design: The role of trust," in *35th Telecommunications and Communications Policy Research Conference (Papers online)*, no. ID: 748, 2007, pp. 1–24, http://web.si.umich.edu/tprc/papers/2007/748/End%202%20end%20and%20trust%2010%20final%20TPRC.pdf.

[9] J. Kempf, R. Austein, and IAB, "The rise of the middle and the future of end-to-end: Reflections on the evolution of the internet architecture," RFC 3724 (Informational), 2004, http://tools.ietf.org/html/rfc3724.

[10] B. van Schewick, "Architecture and innovation: The role of the end-to-end argument in the original internet," Dissertation zum Doktor der Ingenieurswissenschaften (Dr. Ing.) (PhD Thesis), Technische Universität Berlin, Germany, 2004, (available from the university library of Technische Universität Berlin, http://www.ub.tu-berlin.de, signature 4TA3016).

[11] M. Mueller, D. Cogburn, J. Mathiason, and J. Hofmann, "Net neutrality as global principle for internet governance," Internet Governance Project, School of Information Studies, Syracuse University Syracuse, NY USA, Research Paper, 2007, http://www.internetgovernance.org/pdf/NetNeutralityGlobalPrinciple.pdf.

[12] J. Day, *Patterns in Network Architecture: A Return to Fundamentals*. Prentice Hall, 2008.

[13] T. Moors, "A critical review of "end-to-end arguments in system design"," in *Proceedings of IEEE International Conference on Communications, 2002 (ICC 2002)*, vol. 2. IEEE, 2002, pp. 1214–1219.

[14] S. Burnham, "Can privacy and computer coexist?" *NY Times*, vol. November 5, 1983, p. 13, 1983.

[15] C. Partridge, "The technical development of internet email," *IEEE Annals of the History of Computing*, vol. 30, no. 2, pp. 3–29, 2008, http://www.net-tech.bbn.com/~craig/email.pdf (pre-publication version).

[16] B. Carpenter and S. Brim, "Middleboxes: Taxonomy and issues," RFC 3234 (Informational), 2002, http://tools.ietf.org/html/rfc3234.

[17] R. Braden, T. Faber, and M. Handley, "From protocol stack to protocol heap: Role-based architecture," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 17–22, 2003, http://conferences.sigcomm.org/hotnets/2002/papers/braden.pdf.

[18] J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, and A. Warfield, "Plutarch: An argument for network pluralism," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 4, pp. 258–266, 2003.

[19] J. Zittrain, *The Future of the Internet—And How to Stop It*. Yale University Press, 2008, http://ssrn.com/abstract=1125949.

[20] S. E. Gillett, W. H. Lehr, J. T. Wroclawski, and D. D. Clark, "Do appliances threaten internet innovation?" *Communications Magazine, IEEE*, vol. 39, no. 10, pp. 46–51, Oct 2001.

[21] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in cyberspace: Defining tomorrow's internet," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. New York, NY, USA: ACM, 2002, pp. 347–356, http://www.sigcomm.org/sigcomm2002/papers/tussle.pdf.

[22] D. L. Parnas, "On the criteria to be used in decomposing systems into modules," *Communications of the ACM*, vol. 15, no. 12, pp. 1053–1058, 1972, http://www.cs.umd.edu/class/spring2003/cmsc838p/Design/criteria.pdf.

[23] R. L. Aguiar, "Some comments on hourglasses," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 5, pp. 69–72, 2008.

[24] D. L. Tennenhouse and D. J. Wetherall, "Towards an active network architecture," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 2, pp. 5–17, 1996, http://ccr.sigcomm.org/online/files/p81-tennenhouse.pdf (2007 SIGCOMM reprint).

[25] S. Bhattacharjee, K. L. Calvert, and E. W. Zegura, "Active networking and the end-to-end argument," in *Proceedings of the 1997 International Conference on Network Protocols (ICNP '97)*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 220–228, http://www.ieee-icnp.org/1997/papers/1997-23.pdf.

[26] P. Baran, "On distributed communications networks," *IEEE Transactions on Communications*, vol. 12, no. 1, pp. 1–9, 1964.

[27] J. Postel, "Internet control message protocol," RFC 792 (Standard), 1981, http://tools.ietf.org/html/rfc792.

[28] L. Kleinrock, "Nomadic computing (keynote address)," *Telecommunication Systems*, vol. 7, no. 1-3, pp. 5–15, 1997.