

# Making the Unaccountable Internet: The Changing Meaning of Accounting in the Early ARPANET

A. Feder Cooper  
Cornell University  
Department of Computer Science  
Ithaca, NY, USA  
afc78@cornell.edu

Gili Vidan  
Cornell University  
Department of Information Science  
Ithaca, NY, USA  
gv232@cornell.edu

## ABSTRACT

Contemporary concerns over the governance of technological systems often run up against narratives about the technical infeasibility of designing mechanisms for accountability. While in recent AI ethics literature these concerns have been deliberated predominantly in relation to machine learning, other instances in the history of computing also presented circumstances in which computer scientists needed to un-muddle what it means to design accountable systems. One such compelling narrative can frequently be found in canonical histories of the Internet that highlight how its original designers' commitment to the "End-to-End" architectural principle precluded other features from being implemented, resulting in the fast-growing, generative, but ultimately unaccountable network we have today. This paper offers a critique of such technologically essentialist notions of accountability and the characterization of the "unaccountable Internet" as an unintended consequence. It explores the changing meaning of accounting and its relationship to accountability in a selected corpus of requests for comments (RFCs) concerning the early Internet's design from the 1970s and 80s. We characterize four ways of conceptualizing accounting: as *billing*, as *measurement*, as *management*, and as *policy*, and demonstrate how an understanding of accountability was constituted through these shifting meanings. We link together the administrative and technical mechanisms of *accounting* for shared resources in a distributed system and an emerging notion of *accountability* as a social, political, and technical category, arguing that the former is constitutive of the latter. Recovering this history is not only important for understanding the processes that shaped the Internet, but also serves as a starting point for unpacking the complicated political choices that are involved in designing accountability mechanisms for other technological systems today.

## CCS CONCEPTS

• **Social and professional topics** → **History of computing; Network access control; Socio-technical systems; Computing profession;** • **Information systems;** • **Security and privacy** → *Social aspects of security and privacy*;

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

FACCT '22, June 21–24, 2022, Seoul, Republic of Korea

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-9352-2/22/06...\$15.00  
<https://doi.org/10.1145/3531146.3533137>

## KEYWORDS

Accountability, Accounting, Accountable systems, Internet governance, Resource sharing

### ACM Reference Format:

A. Feder Cooper and Gili Vidan. 2022. Making the Unaccountable Internet: The Changing Meaning of Accounting in the Early ARPANET. In *2022 ACM Conference on Fairness, Accountability, and Transparency (FACCT '22)*, June 21–24, 2022, Seoul, Republic of Korea. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3531146.3533137>

## 1 INTRODUCTION

There has recently been a revival of interest in the value of accountability in technical systems, especially those that involve machine learning. Already in 1996, Nissenbaum [93] recognized the unique challenges of accountability in a computerized society; from a philosophical perspective, she discussed how the ascendance of computerization in important societal contexts was poised to introduce novel barriers to accountability. 25 years later, one can observe the difficulties Nissenbaum [93] raised playing out in recent AI ethics scholarship, illustrating that accountability is a slippery term with multiple valences [33]. For example, several scholars have imported a definition of "public accountability" from Bovens [13] into research concerning algorithmic accountability [64, 127], while others have attempted to institute cultures of accountability through *ex ante* engineering standards of care [45, 89] or *post hoc* audits and impact assessments [3, 87, 90, 108, 120]. These works draw from philosophy, computer science, and the law, and focus almost exclusively on accountability in relation to *algorithms*. This emphasis on algorithms has left broader systems and institutions (in which algorithms are situated) unattended in terms of their relationship to accountability [32].

In this paper, we provide a complement to prior conceptual technological accountability scholarship. We take a historical approach and examine accountability as it pertains to a specific computer system, situated in a time and a place, including its architectural components, protocols, and institutional actors. Rather than taking a top-down approach in which we posit our own definition of accountability to ground our analysis [67, 70, 71, 127], we investigate how an understanding of unaccountability can emerge and evolve over time during a system's development. For our subject, we concentrate on the early Internet and its predecessor—the ARPA-controlled ARPANET.<sup>1</sup> Today, the Internet is often described

<sup>1</sup>ARPA, the Advanced Research Projects Agency, has gone back and forth on including a D for "Defense" at the start of its name. The agency settled on DARPA in 1996, but we refer to it as ARPA throughout for clarity, since our subject, the ARPANET (sometimes referred to as the ARPAnet), was named after ARPA [77, p. 103] [31].

as lacking endemic features for accountability, fixing the perception it is an *unaccountable* system [31, 109, 124]. We contend that identifying how this perception developed can help locate what it means for a system to be accountable, and can enable us to interrogate the dynamics that lead a research community to develop an unaccountable system. While our work is a deep dive into the particular origins of the unaccountable Internet, it is also a case study that elicits broader lessons concerning the unaccountability of complex computer systems, and thus serves as a cautionary tale for the design of contemporary systems.

## 1.1 Retelling Internet Histories: From End-to-End to Accounting

Over the past two decades, Internet historians have pointed to the institutional context of the network's protocols and standards' development [2, 18, 39, 115, 128], explored how commercial actors, user groups, and regulators also shaped its trajectory [50, 83, 119], and contrasted the US-origins of the Internet with networks developed in different political contexts [85, 100]. We follow these studies by looking at how network engineers and administrators, as a research community and governing institution, defined the goal of networking and often contrasted it with changing meanings of "accounting" and "accountability." This paper surveys the early years of the ARPANET project, what Braman [17] has called the "framing years," and goes through the transition of the ARPANET to the Internet. During this decades-long period, the governing bodies and structures of the networks in question were changing, even if many of the same individuals developing network computing remained involved. We therefore refer to this community as "network engineers" or ARPANET contractors in its early years, and as its more formalized afterlives—the Internet Activities Board (IAB) and the Internet Engineering Task Force (IETF).<sup>2</sup>

These institutional perspectives on the history and politics of networks contrast with other accounts that focus on the architectural novelty of the network's design. At the center of these architectural histories stands the now foundational "End-to-End principle." In the early 1980s, Jerome Salzer, Dave Reed, and David Clark, three members of the ARPANET project team, formalized the End-to-End principle in a paper, arguing that a communication network should implement most complex functions at the endpoints rather than within the network itself, leaving the network's design relatively simple and focused on routing data traffic [116, p. 278].<sup>3</sup> Through various retellings of the Internet's origin myth, End-to-End became its defining feature, used to explain the network's exponential, rapid growth and celebrated for its architectural simplicity and even its political commitment to decentralization, individual autonomy, and freedom—and later, a reason behind some of the Internet's discontents [12, 109].<sup>4</sup> The ubiquity of End-to-End as a stand-in for the Internet as a whole was noted by Tarleton Gillespie, who argued that such a cultural uptake of one specific term both obfuscated

its contentious meaning and created a sense in which the architecture of the Internet became a fixed material object [46].<sup>5</sup> This End-to-End ethos has arguably carried forward to the present more generally in computing, with Zittrain [130] recasting End-to-End as the "procrastination principle" to capture the idea that "most problems could be solved later and by others."<sup>6</sup> One can see this in several places, for example: invoked in Google's now-retired motto, "Don't be evil," a rhetorical move that could be read as punting accountability for the behavior of its products to individual engineers; in evolving conversations concerning balancing accountability and unconstrained user behavior with respect to platform content moderation [27, 47, 68, 69]; and in corporate and institutional ethics statements targeted at individual engineers to use machine learning and artificial intelligence technology responsibly.

Among the many possible narratives that get lost in the End-to-End telling is the story of a difficult administrative, technical, economic, and political problem at the heart of the ARPANET: distributed accounting. In the late 1980s, David Clark, one of the trio behind the original 1980s End-to-End paper, reflected back on the experience of designing the ARPANET and Internet protocols and described a set of goals that guided the design principles of the network. The top-level goal was to create a means of interconnecting existing computer networks, supporting the connection of different devices and pre-existing communication protocols [28, pp. 1-2]. It is in relation to this top-level goal that End-to-End was often celebrated as an elegant solution. But Clark went on to articulate a list of seven "secondary goals," last among which read: "resources used on the network must be accountable" [28, pp. 1-2]. Clark refers to this as the goal of "accountability" and cites earlier work on the ARPANET by Vint Cerf and Bob Kahn that already noted the need for a network to provide such features in 1974. The paper in question, however, only makes mention of "accounting" for traffic in relation to the potential to charge for network use [26, p. 2].

In this paper, we propose to take seriously the slippage between Clark's 1988 use of "accountability" and the subordinate, discounted goal of accounting for resources. We argue that matters of accounting were never *merely* about charging and billing, nor were they straightforward—they raised a variety of questions that form a theory of accountability in a networked environment. We trace how the historical actors designing the technical system that is the Internet routinely debated, deferred, and discounted accounting as a feature of the network, and we see how, over time, this community came to define the meaning of accounting as constitutive of accountability.<sup>7</sup> In tracing the shifting meaning of accounting in

<sup>2</sup>Russell [114, pp. 50-53] provides a brief but insightful overview of the process of expansion and formalization of the network research and development community that this paper examines.

<sup>3</sup>The principle was a tacit assumption of the ARPANET project (among other research efforts) for over a decade by the time the paper made it explicit [114, p. 49] [116].

<sup>4</sup>Legal scholars attempting to theorize the relationship between network architecture and political order in particular zeroed in on End-to-End as a way to tie the two together [30, 50, 78, 103, 129].

<sup>5</sup>This cultural uptake had broader reverberations among users of the network. In November 1988, the Morris Worm caused failures that brought down the majority of nodes on the Internet [110]. The network seemed to be in crisis. That such widespread failure could occur suggested the existing, End-to-End-guided architecture was perhaps not suited to the network's broader use and needs. (A retrospective analysis of the Worm is out of the scope of this paper. Please refer to Slayton [117].) Nevertheless, the culture of End-to-End prevailed. After the initial response to excise the Worm from the network, what followed shortly after was a series of ethics statements from revered institutions, including the IAB, MIT, and the NSF [7, 34, 58, 59, 91]. These statements centered the *end-user* as the site at which to locate the responsibility for appropriate network use, rather than calling for proposals to imagine what designing for accountability could mean for internals of the network.

<sup>6</sup>Zittrain discusses the principle in terms of the network in this blog post, and discusses its broader import for software engineering in his book [129].

<sup>7</sup>Daniel Neyland has argued that in the context of machine learning algorithms, there is an intersection between the two registers of the term "account," suggesting that

the first two decades of network computing, we offer a retelling of Internet histories, locating the unresolved problem of accounting as the obverse of the End-to-End elegant solution story.

## 1.2 Research Method and Contribution

We enlist a variety of secondary sources, including work by historians and Science and Technology Studies (STS) scholars. Our main focus is primary sources drawn from the self-described “old boys network” [2, p. 54] that participated in the architecture discussions and implementation of the ARPANET, the Internet’s US predecessor during the 1970s. We follow this community’s writings through the 1980s—the early days of the Internet—bringing in contemporary retrospectives of Internet history written by networking researchers like David Clark, as well as early networking conference proceedings, journal articles, and technical reports. However, we primarily focus on a corpus of Internet Requests for Comments (RFCs): a chronologically-ordered series of documents posted publicly and hosted online by the Internet Society. The RFCs present multiple histories and can be read for their technical content, a community’s self-reflection on its own history, and as source materials for analysts of governance formation [18, p. 66]. The status and means of distributing the RFCs was also subject to change over the period we examine.<sup>8</sup> Our use of the RFCs was therefore a helpful way to map the contours of the community’s conversations over accounting and accountability.<sup>9</sup>

We scraped and filtered the entire corpus to find instances of RFCs that mention “accounting” and “accountability.” Our process was not just a matter of collecting and counting all documents from the first RFC to mention “accounting” [37, p. 3],<sup>10</sup> to the first RFC that uses the word “accountability” [44, p. 2].<sup>11</sup> Rather, we performed a detailed reading to produce an understanding of how accountability evolved conceptually from accounting over time. By our accounting, *accounting* moved through being understood as *billing* (Section 2), as *measurement* (Section 3), as *management* (Section 4), and as *policy* (Section 5), a point at which *accounting* was understood as constitutive of *accountability*, and the lack of accounting features developed within the network’s design an impediment to the network’s operations. The rest of the paper is organized around these four notions, with some nonlinear, overlapping chronological progression. We begin with documents dating

calls to make algorithms more accountable should take note of system features that make technical systems account-able [92]. We similarly focus on the intersection of these two registers, but, by taking a historical approach, we go beyond suggesting a connection to tracing its development.

<sup>8</sup>Originally, the RFCs were necessarily circulated on paper through the physical mail system, and then later were hosted at Stanford Research Institute (SRI) [2]. The current *rfc-editor* tool is available at <https://www.rfc-editor.org/>.

<sup>9</sup>We, however, did not consider the RFCs to be the sole means through which such conversations took place (see Appendix C). For more on how media, legal, and policy scholars have approached reading RFCs see: Milne [88, pp. 40-43] and Braman [16].

<sup>10</sup>RFC 33 represents the first mention of accounting; it added a notion of accounting to the earlier host-host protocol, originally defined in 1969 in RFC 11 [38].

<sup>11</sup>While RFC 721 in 1976 is the first to use the word “accountability,” arguably, it was RFC 808 in 1982, documenting a January 10, 1979 meeting at BBN, which was the first to distinguish between “accounting” and “accountability” explicitly [106, p. 3]: “There was some general discussion of the impact of personal computers on mail services. The main realization being that the personal computer will not be available to handle incoming mail all the time. Probably, personal computer users will have their mailboxes on some big brother computer (which may be dedicated to mailbox service, or be a general purpose host) and poll for their mail when they want to read it. There were some concerns raised about accountability and accounting” (emphasis added).

from before the first RFC—prior to the proposal of the ARPANET project [111]—and trace the changing meaning of accounting and accountability through 1990, when the ARPANET was decommissioned [2, p. 195] [25]. We scope our project to this period because, while concerns about accountability clearly extend through to present day discussions of Internet governance [27, 31, 68], we found that it was during this time period that a notion of an unaccountable network first came about and took hold as a defining characterization of the Internet.

By identifying the changing meanings of accounting in the ARPANET and early Internet, we offer the following three contributions for the study of accountability in technical systems: first, we link together the administrative and technical mechanisms of *accounting* for shared resources in a distributed system and an emerging notion of *accountability* as a social, political, and technical category, arguing that the former is constitutive of the latter. Second, we characterize a research dynamic among the technical community we studied that deprioritizes the development of administrative tools and accounting infrastructure, treating it as existing somewhere beyond the scope of their work as a matter of “policy” or as subordinate to their core research objectives. Third, this retelling of the early history of the Internet offers not only a corrective for how we view its particular development, but also provides significant lessons about the role of having institutional structures in place and designing for and around their administrative needs when building new accountable technological systems.

## 2 ACCOUNTING AS BILLING: THE DOUBLE-BIND OF SHARING NETWORKED RESOURCES

“ARPA will not pay for the coffee and pastry being served, so please chip in to help me pay for it” [40, p. 2]. This was early-Internet engineer Steve Crocker’s introductory remark to the Network Working Group (NWG) of the ARPANET project in its November 16, 1970 meeting in Houston, Texas. While this may seem like an inane detail about NWG bookkeeping, it in fact serves to highlight the moment when there began a major shift concerning who was paying for using ARPA resources, and what exactly constituted the use that needed to be paid for. That is, while ARPA’s Information Processing Techniques Office (IPTO) had comprehensively funded its researcher-contractors’ logistical and capital expenses since its inception in 1962, by 1970, IPTO-funded labs felt the purse strings tighten. Low-level, seemingly incidental operational costs like a working group’s coffee bill warranted space in the official NWG meeting record. Amid the gravitas of what its members even at the time recognized would be pivotal discussions concerning the architecture of the first distributed computing network, the question of who pays for the coffee was never too far from the question of who pays for computing [40, p. 18].<sup>12</sup>

<sup>12</sup>Starting in 1962, IPTO, situated within the broader ARPA umbrella, essentially funded computer science research centers across the US (including at MIT, UCLA, and Carnegie Mellon), “often outspending universities significantly” in terms of research support [2, pp. 36-37, 44, 56]. These research centers followed the local time-sharing computing paradigm [23]. Also in 1962, J. C. R. Licklider and Welden E. Clark wrote their piece on “On-line man-computer communication” [79] that is frequently cited as the first work to discuss communicating, interconnected computers that can share data and programs [76, 77]. For more on the 1960s development of a notion of an

Talking about the cost of resources was new for beneficiaries of IPTO's funding. Such operational details had not been a concern when there was no network through which computing resources were to be shared. IPTO had purchased computers for its contracted computing sites, which were working on local, site-specific projects, such as automated theorem proving at Stanford Research Institute (SRI) and natural language processing at Bolt, Beranek, and Newman (BBN) [112, p. 548] [2, p. 44].<sup>13</sup> Having paid for these computers up front, IPTO was not particularly concerned about the low-level specifics of how they were used. From the perspective of IPTO-contract-site researchers, this hands-off policy enabled conditions of unrestricted, free usage; it seemed like contracting with ARPA was easy money, as the funding seemed to come with "few strings attached" [2, p. 77].

However, this status quo of unchecked use was not to last. IPTO's original mandate included the goal of eventually connecting its funded sites, even prior to the specific proposal of the ARPANET [77, 82, 112].<sup>14</sup> By the end of 1971, when ARPA was completing the first phase of the ARPANET's construction to connect 15 Interface Message Processors (IMPs), IPTO's vision became a funding precondition [54]. At this time, IPTO appealed to its contractors to no longer just use their computers as local lab resources, but rather to exercise their connection to the network [2, pp. 44–46, p. 55] [112].<sup>15</sup> All 15 of these sites housed other ARPA-funded computing projects. As a result, even if networking was not the focus of every site's individual projects, each site was expected to participate [2, p. 50, pp. 77–78, p. 161] [112, p. 548]. In other words, IPTO needed its contractors to utilize the network they had invested in building, in order to test the network's potential for distributed computing—for two or more remote nodes to effectively work together to complete computational tasks. So, starting in late 1970, the perception of "no strings attached" funding began to crumble [36, 40, 53, 104, 123]. It was becoming clear that ARPA's funding did in fact come with a particular yoke: Contractors did not just have to connect to the ARPANET; they also had to use it.

This pivotal moment, nearly 10 years after IPTO's founding, marked when it was possible to move theory to practice—to empirically validate IPTO's commitment to resource-sharing via the early ARPANET. Historian Janet Abbate discusses how this promise

interconnected computer network, both by Licklider and among other researchers, see Abbate [2, Ch. 1], Turner [119], and Aspray and Ceruzzi [6].

<sup>13</sup>IPTO/ARPA also leased communication lines from common carriers, to serve as the physical connection medium between remote nodes [111] [115, pp. 167–8].

<sup>14</sup>The ARPANET proposal ultimately covered more specific goals: a network for load sharing, a message service, a remote service, data and program sharing, specialized systems software and hardware, and scientific communication. It also described the operation of such a network to "foster the 'community' use of computers" [111, p. 2].

<sup>15</sup>This notion of resource-sharing was, at least at first, considered the distributed analogue of the current local time-sharing computing paradigm, which had its own accounting concerns: "The goal of the computer network is for each computer to make every local resource available to any computer in the net in such a way that any program available to local users can be used remotely without degradation. That is, any program should be able to call on the resources of other computers much as it would call a subroutine. The resources which can be shared in this way include software and data, as well as hardware. Within a local community, time-sharing systems already permit the sharing of software resources. An effective network would eliminate the size and distance limitations on such communities" [112, p. 543]. See also Carr et al. [23, p. 589]: "However, early time-sharing studies at the University of California at Berkeley, MIT, Lincoln Laboratory, and System Development Corporation (all ARPA sponsored) have had considerable influence on the design of the network. In some sense, the ARPA network of time-shared computers is a natural extension of earlier time-sharing concepts." Also, see generally Marill and Roberts [82].

faded rather quickly: "the decline of the ideal of resource sharing," she argues, came as the result of ARPANET's usability issues [2, p. 104] [101, p. 5]. While connecting to the network had been a grueling engineering task, it ultimately was just the beginning of ARPANET's resource-sharing challenges. Once connected, it was difficult to locate specific resources in the network and lingering interoperability issues meant that, even once a resource was found, it often remained unclear how to access it [40, pp. 5–6] [101, p. 6] [97]. Abbate thus concludes that resource-sharing seemed more onerous than it was worth, such that the "demand for remote resources fell," leaving "many sites rich in computing resources...looking for users" and the ARPANET a technology in search of an appropriate application [2, p. 104].

But the usability of a network is itself constructed through choices regarding its administrative infrastructure. Seeking to unpack Abbate's invocation of "usability," we argue that the challenges of resource-sharing, and the changes it produced in IPTO-funded computers' use, can also be ascribed to seemingly mundane (but in fact very difficult) issues of bookkeeping. Even though ARPA continued to foot the entire bill for the ARPANET, both in terms of capital and communication costs [2, p. 85, p. 161], from the perspective of individual research sites, resource-sharing constituted a sacrifice—a loss of the unrestricted, free local computer use that had been the status quo.<sup>16</sup> That is, by requiring sites to reallocate a portion of their computing resources for distributed use, resource-sharing seemed akin to ceding control of one's own local computing budget to remote users with their own respective, perhaps even competing, needs [2, p. 50].

As a result, even though IPTO required them to use the ARPANET, many contractors exhibited unwillingness to do so, wondering how to prioritize local and remote use. Richard G. Mills, the director of MIT's information processing services, succinctly captured this hesitancy, saying: "There is some question as to who should be served first, an unknown user or our local researchers" [2, p. 226]. ARPA was no longer covering all operational costs: It was not paying for the coffee and pastries, and it was not compensating for the loss of previously unrestricted local resources now being shared with others in the network. Inducing resource-sharing therefore exposed a fundamental, underlying tension in distributed computing. Individual labs may not have wanted to give up their local resources, but they also recognized the potential value of being able to use other labs' resources. One suggested solution to this tension was for site administrators to bill remote users in order to recoup losses or disincentivize remote use [40, p. 7]. As J. Pickens asserted in RFC 369, "if distributed computing is allowed, then distributed billing is a necessity" [101, p. 6].

For labs to charge for the use of their resources represented a fundamental shift in the management paradigm of the ARPANET. Until this point, there was no need to do low-level accounting of individual line items of resource usage—of who was using (or even misusing) specific parts of the network—because ARPA was managing the overarching cost. Billing, however, now imposed a new burden of operational costs on individual sites, in which low-level

<sup>16</sup>It is possible that this was just a *perceived* sacrifice of local resources for collective external use, with no actual scarcity of computing resources for those that wanted it. Nevertheless, even if not an actual sacrifice, as we discuss, there was a reluctance to even agree to share local resources with the network.

bookkeeping was going to become crucial for the first time. In the remainder of this section, we show how the challenges presented by the need to develop accounting mechanisms structured the ambivalence towards resource-sharing in the early ARPANET. Accounting posed a non-trivial problem. Local time-sharing computers already “possess[ed] elaborate and definite accounting and resource allocation mechanisms” [37, p. 5], but these did not naturally extend to the distributed resource-sharing environment.<sup>17</sup> The non-triviality of distributed accounting posed challenges that researchers neither knew how to solve, nor really desired to spend time solving in place of performing their individual research.

## 2.1 Explicit and implicit billing: The case of “free” file transfer

The early ARPANET debate over “free” file transfer gives an intuition for the complexity of accounting in the distributed resource-sharing environment. File transfer was (and remains) one of the basic functions of resource-sharing over networked computers. It facilitates the transfer of files from one node to another, enabling sharing among remote users. The File Transfer Protocol [9] first described this capability, which requires resource usage, including memory and CPU utilization, and thus made it a prime candidate feature for billing.<sup>18</sup> Nevertheless, despite the apparent necessity for billing to support the network, many individual users wanted to avoid payment. When using FTP, they leveraged a loophole: the MAIL FILE feature, which allowed for bypassing login on the remote host that housed the file of interest, and made it possible for a user to mail the file to themselves, in place of transferring it via TELNET-based connections.

In response to the popular use of FTP’s accounting loophole, Rob Bressler, a network protocol developer at BBN, issued an RFC calling to codify a more appropriate accounting-free FTP use pattern: the implementation of free, loginless file transfer, which would give users the free access they wanted without abusing FTP’s intended use [21]. Rather than using MAIL FILE, his proposal expressly allowed for users to bypass authentication via a loginless facility and, without logging in, it would not be possible to account for who was transferring the file. The transfer would be “free,” as it would not be possible to bill an account for it.<sup>19</sup> Bressler’s proposal for an intentional, free file transfer feature immediately raised questions about what it meant for resource usage to be “free” [99]. Notably,

<sup>17</sup>As RFC 504 asked, “If you employ accounting Procedures that require cost recovery, how, if at all, should they be modified to work in a network resource sharing environment?” [118, p. 4]. This was not at all straightforward for key technical reasons because inherent issues with consistency between nodes in a distributed system further complicate the mechanics of correct accounting. While consistency in distributed databases is now known to be a topic of fundamental importance in distributing computing, it seems that this issue was first noted in RFC 677 [63] in relation to maintaining duplicate databases for correct, consistent Terminal Interface Processor (TIP) accounting. RFC 677 notes that its contents go beyond “ARPA-like networks” and “are generally applicable to distributed database problems” [63, p. 1], talking about issues of partition tolerance [63, p. 3] and consistency [63, p. 4], as well as how timestamps can be useful for maintaining consistency because they are monotonically increasing (noting, however, that this can be complicated by clock skew between nodes). For a more contemporary treatment of these issues, refer to Abadi [1].

<sup>18</sup>To enable billing, RFC 385 added an account (ACCT) command to the FTP protocol, in order to distinguish accounts used for resource accounting as serving a different function from users logged onto the network [10].

<sup>19</sup>On hardware for which such loginless access was not possible, such as TENEX machines, Bressler proposed adding an account named “FREE,” which could be used by users to avoid billing to specific accounts [21].

fellow BBN network engineer, Ken Pogran rebuffed Bressler’s RFC for making “sweeping assumptions...about the nature and use of accounting mechanisms [102, p. 1].” Pogran resolved to “un-muddle” so-called ‘free file transfer’, making the case that the resource usage involved in FTP (deemed by Bressler as negligible) was in fact quite costly. While CPU utilization for transferring one file might seem negligible, over time such costs add up and certainly cannot be called “free” [102, p. 4]. In short, Pogran argued, nothing is free if you are the one who has to worry about costs.

Yet, while Pogran challenged what it meant for resource usage to be called “free,” he did not claim that such actually not-“free” usage should be disallowed. Rather, he contended that “free” should mean that resource usage was free of charge for a user at a research site—and that such “free” usage should get charged to an overhead “network services account” that ultimately got billed to ARPA [102, p. 3]. In other words, while Pogran seemed to have taken a more nuanced view than Bressler about the costs of resource-sharing, RFC 501 does not ultimately “un-muddle” the issue of distributed accounting. Like Bressler, Pogran also saw merit in avoiding the particulars of accounting; he, too, found it desirable for researchers like himself to not be concerned with the minutiae of how costs got covered. However, unlike Bressler, Pogran made explicit that ignoring costs did not simply make them go away. Rather, he highlighted how explicitly implementing mechanisms to evade accounting corresponded to the status quo of ARPA being on the hook for the bill. The exchange between Bressler and Pogran underscores the same contradiction: Both talked about accounting as necessary to recoup local site losses due to resource-sharing, but both also affirmed the common desire of ARPANET researchers to not pay to use remote resources. The *ad hoc* strategy of “free” accounts would prove infeasible in the longer term, when it was expected for the network to host nodes and users not funded by ARPA [102, p. 3].

## 3 ACCOUNTING AS MEASUREMENT: CONTESTING THE NECESSARY FUNCTIONS OF NETWORKED COMPUTING

The debate over “free” file transfer demonstrates that deciding how to classify what resource usage needed to be accounted for was a challenging and contentious problem.<sup>20</sup> As IPTO pushed for resource sharing more actively, it was not clear what needed to be accounted for. Making the decision to explicitly build workarounds to avoid accounting—pushing some types of resource-sharing costs back to ARPA—ultimately obscured the complex and pervasive role of accounting in a distributed, resource-sharing environment.

The architects of the ARPANET repeatedly punted on designing mechanisms for accounting. Even during the ARPANET’s earliest years, around the completion of the connection of the first part of the network in 1971, accounting had frequently come up as a necessary function, albeit one with unclear requirements [36, 40, 104, 123]. In response, Bob Kahn, co-lead of the ARPANET protocols team, prepared RFC 136: the first unified attempt to clarify accounting’s role in resource-sharing.<sup>21</sup> He raised

<sup>20</sup>See Appendix A for discussion of similar attempts at this classification.

<sup>21</sup>RFC 136 was concurrent with discussions about how to distinguish between billed Research Centers and free (but limited) Service Centers (Appendix A). Notably, this predates the debate over “free” file transfer discussed in Section 2.1, indicating that

ten, as-yet-unanswered crucial questions related to accounting, which ranged from a future of potential private control of the ARPANET through government regulations of network use and to how resource usage should be measured and characterized. These questions, in attempting to clarify what accounting requires, instead serve to clarify just how complicated and expansive accounting is: While accounting clearly involves billing, billing is not the only component of accounting.

Several of Kahn's RFC 136 questions concerning accounting implicated the ability to take measurements of network activity. While questions and engineering activities concerning measurement pre-date RFC 136 [66] [84, Appendix], this was the first time that an RFC unified questions about measurement and accounting within the same scope.<sup>22</sup> Prior to this RFC, accounting and measurement had generally been considered separate—albeit both necessary—functional concerns [66, e.g., p. 19] [84, Appendix], neither of which had been solved by the ARPANET architects. While accounting was conceived of as billing and treated as a nuisance to be kept separate from research (Section 2.1, Appendix A), measurement was afforded the status of being integral to research. Measurement was a necessity for those “interested in the network as an object of study,” [104, p. 2] [40] while accounting was not considered to have such a central role.

The earliest example of measurement's importance as a research function concerns the work of Gerry Cole at UCLA. As late as February 1971, Cole collected and analyzed network data in order to better understand resource usage patterns in the ARPANET's novel distributed environment [123, p. 2].<sup>23</sup> Shortly after these initial measurement efforts, BBN took over the role more formally and set up the Network Control Center (NCC), led by Alex McKenzie, to measure network statistics. The NCC monitored all nodes attached to the ARPANET at this time, and assumed the role of ensuring that the entire network ran smoothly by documenting reliability issues, debugging and diagnosing malfunctions, and monitoring resource usage on the network [2, pp. 64-67,72].<sup>24</sup> The NCC was

the problems RFC 136 raised remained unresolved and carried over into later debates such as FTP.

<sup>22</sup>The method of network operation and the potential for its growth are relevant factors to be considered in formulating a plan for Host accounting. For example, the answers to the following questions provide a useful background for reference: 1. Who or what operates the Network? 2. What is the criteria upon which new sites should be incorporated into the Network? 3. What regulations, if any, apply to the connection of non-ARPA sites? 4. What is the relation, if any, between the ARPA Network and common carrier services? 5. What procedures are required to bring new sites on board and up to speed? 6. What is the most effective way to characterize their Resources? 7. What usage of other Network resources do they anticipate? 8. What procedures will be required for a typical user to obtain access to that Host? 9. What is their charging policy and for what items? 10. Are their rates in accordance with government standards? [65, p. 1]. In short, adding an “account” field to different network functions perhaps could help facilitate accounting, but it was not sufficient on its own to capture the wide-ranging semantics of accounting implicated by Kahn's considerations.

<sup>23</sup>One RFC noting this project read: “Gerry requested that when people are set up to use the Network, they inform him so that he can gather statistics. UCLA will eventually have a program to scan the Network for utilization, but if people could tell him when they were going on to use the Network, it would be easier to measure meaningful things and interpret the data from a knowledge of type of usage” [123, p. 2].

<sup>24</sup>In RFC 101, Kahn was recorded to have mentioned that BBN had an interest in collecting measurements on the network [123, p. 2]. Abbate confirms this, documenting Alex McKenzie's role at BBN: He joined the ARPANET project when BBN's node went online as part of the November 1970 deployment [40, 104], and in 1971 he took charge of the NCC [2, pp. 64-67]: “the NCC acquired a full-time staff and began coordinating upgrades of IMP hardware and software. The NCC assumed responsibility for fixing all operational problems in the network, whether or not BBN's equipment was at fault. Its staff monitored the ARPANET constantly, recording when each IMP, line, or host

not limited solely to conducting research on network utilization, but was responsible for its *de facto* operation, a role we discuss in greater detail in Section 4.

As Bob Kahn suggested in RFC 136, these functions, aside from playing a role in developing an understanding of network use, were an essential component of accounting [65, p. 1]. He made clear that it would not be possible to account for resource usage without having appropriate mechanisms in place to measure resource usage. Accounting-related measurements would not just involve the “who, what, where, and when” of network use [35, p. 3]; they would also involve metrics concerning site performance, such as user response times and frequency of crashes [101].<sup>25</sup> By indicating overlap with functions like measurement, the accounting-related considerations Kahn raised in RFC 136 implicated fundamental questions about what the network should do, and how it should be implemented.

These considerations raised not only practical challenges, but also ideological questions about the network's purpose. The act of performing accounting itself consumes network resources; it costs something to account for costs. The early ARPANET architects considered these costs to be overhead. They were concerned that accounting “costs space” [40, p. 3] and could cause “undue delays in accessing distributed resources” [122, p. 4]. They viewed the resource consumption involved in accounting as an imposition that was in tension with the ARPANET's fundamental, ideal goal to achieve distributed resource sharing [2, pp. 96-97] [112]. Accounting therefore presented a seemingly irreconcilable contradiction: On the one hand, ARPANET architects repeatedly acknowledged that *accounting was critical* to facilitate interconnection between distributed nodes; on the other, they viewed *accounting as hostile* to that very same goal.

This view of accounting as a burden to the network can be understood in relation to the End-to-End architectural principle guiding the construction of the ARPANET. As mentioned in Section 1, from a technical perspective, End-to-End, is a preference toward parsimony in the network—of placing application-specific functionality where it is needed at the end hosts, rather than implementing it inside the network as a feature accessible to all hosts. This principle biases toward only placing the essentials for connectivity inside the network, with the notable exception for features “justified only as performance enhancements” [116, pp. 1,9]. Accounting certainly could not be considered a performance enhancement.<sup>26</sup> In fact, as we have seen, ARPANET architects viewed accounting as a performance hit that should be kept “to a minimum” [40, p. 3]. Abiding

went up or down and taking trouble reports from users. When NCC monitors detected a disruption of service, they used the IMP's diagnostic features to identify its cause. Malfunctions in remote IMPs could often be fixed from the NCC via the network, using the control functions that BBN had built into the network” [2, p. 65]; “By 1976, the Network Control Center was, according to McKenzie, ‘the only accessible, responsible, continuously staffed organization in existence which was generally concerned with network performance as perceived by the user.’ ... The NCC had become a managerial reinforcement of ARPA's layering scheme” [2, p. 66, internal citations omitted].

<sup>25</sup>Metrics would be important for accurately accounting for past resource usage [35, p. 5], and would come to be recognized in the 1980s as important for enabling individual sites to predict future resource usage and corresponding costs [55, pp. 1-3,8,27,44].

<sup>26</sup>The principle, called the End-to-End argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. ... Low level mechanisms to support these functions are *justified only as performance enhancements*” [116, p. 1, emphasis added]. The ability to record measurements, which we note is fundamentally tied to accounting, is essential from the ideological perspective of End-to-End; it is not possible to justify performance enhancements without being able to measure performance.

by End-to-End, it would be “uneconomical” to include accounting within the network, rather than pushing its implementation to end hosts. The reluctance to implement accounting was therefore not just at the level of specific engineers who wanted to evade paying for FTP; rather, it reflected the significant ideological challenges that accounting presented for a network priding itself on its parsimony.

Consistent with an End-to-End approach to accounting, RFC 136 was scoped to “Host Accounting.” Its intent was to address the issue as a matter for the end nodes rather than inside the network. Yet, toward the end, Kahn posed a speculative question for the future, which can be read in direct tension with End-to-End: “Should Host accounting information *eventually flow via the Network?*” [65, p. 4, emphasis added]. However, since accounting was clearly not a “performance enhancement” [116, p. 1], for accounting to not violate End-to-End, it would have to be justified as an essential architectural feature of connecting distributed nodes. An attempt at such a justification would have run contrary to the RFCs at this time, which always list accounting as distinct from the technical requirements of connectivity [36, 40, 104, 123].<sup>27</sup> In the interim, there remained no concrete plan for developing accounting procedures.

### 3.1 The policy of no policy takes hold

In the years following RFC 136, the ARPANET fell short of achieving IPTO’s goal of facilitating resource-sharing [111]. Abbate’s canonical narrative attributes the decline of this ideal to the fact that the ARPANET was very difficult to use, especially for new users trying to join the network. She argues that this created an identity crisis for the ARPANET; it was a technology in search of an application, a role that email was well-posed to assume as it did not suffer from the same usability issues that plagued resource-sharing [2, p. 106, the “smash hit” of email] (Appendix B). While usability issues presented a challenge for the adoption of resource-sharing, we argue that the decline of resource-sharing can also find its roots in the inability to account for resource usage. With no mechanism to account for resource usage, it was not possible to realize the imagined potential of resource-sharing. To reiterate and rephrase Pickens’ assertion, without the necessary functionality of distributed billing, distributed computing was not feasible [101, p. 6].

Accounting was fundamental to the network, but was also fundamentally unresolved. It was non-trivial in all the same ways

<sup>27</sup>Given that the architects commonly recognized the importance of accounting—seen in Kahn’s own RFC 136 and those that preceded it [36, 40, 104, 123]—it was arguably reasonable for Kahn to question whether accounting was fundamental enough to be an in-network feature. One could conceivably read Kahn’s question as a provocation: Will there be a time at which accounting will be important enough to violate End-to-End? This would have been unthinkable in 1971. Instead, Kahn recommended a more gradual, conservative approach: “the implementation of standard automated accounting procedures involving the use of the Network will be deferred until non-automated procedures have been understood and stabilized. Early experimentation in this area is appropriate, however” [65, p. 2]. Even after RFC 136, accounting continued to be punted until some unspecified, but inevitable point in the future. RFC 82 acknowledges that one possibility would be to “worry about accounting when saturation occurs” [40, p. 7]—i.e., when resource usage reached an extent for which it would be absolutely necessary to do accounting, especially since at that time “non-ARPA folks [would] be able to connect” [65, p. 4]. Accounting could also be put off until a future in which ARPA was no longer responsible for the network infrastructure—when another government agency or a private commercial entity assumed the “total cost of operating” [65, p. 2]. It is thus reasonable to conclude that, even as early as 1971, commercialization loomed as a possible future for the ARPANET, at which point accounting would be absolutely necessary for reimbursement “on both a connect and usage basis” [65, p. 4].

that devising protocols for the use of shared resources was non-trivial: inescapably tied to measuring and managing network performance, and implicated in everything from resource allocation to quality of service. These tensions extended well beyond the early 1970s [60, 63, 75, 98, 107, e.g.]. In 1994, one RFC called accounting (grouped with security) “the bane of the network operator,” but admitted that it is the feature “most requested...by those who are responsible for paying the bills” [4, pp. 149-150]. More recently, David Clark referred throughout his latest book to the importance of accounting, and yet also relegated it to a secondary function, saying it “only plays a supporting role compared to the core objective of forwarding data, and researchers like to work on the lead problem, not the supporting role” [31, pp. 46-47]. Importantly, as we argue in the next sections, these accounting tensions carried over to network-wide policy tensions. There were repeated attempts to minimize or redefine the role of accounting, often in the service of deprioritizing its implementation. In the process of attempting to preserve the “free” aspects of the early ARPANET, the network became a free-for-all.

## 4 ACCOUNTING AS MANAGEMENT: TRACING THE BOUNDARIES OF RESPONSIBILITY AND AUTHORITY

The ambivalence toward accounting’s role in the network persisted in the decade that followed the ARPANET’s expansion to more sites. Our discussion of the early years of the network argues that, even during a time of relatively low saturation, accounting emerged as a function with bearing on both billing and measurement. But by the mid-1970s, users’ demands for a more reliable network to support their research work showed how without management the network was not sufficiently operational. Later, the ARPANET’s connection with other networks that constituted the Internet, the “network of networks,”<sup>28</sup> put the possibility of interacting with mistrusted agents front of mind for the network protocol developers. This section explores how accounting, the flexible term for many administrative aspects of network development, was initially contrasted with the core need to create features for network *management*. By the late 1980s, however, descriptions of what would constitute effective network management explicitly referred to accounting as a main component. If the early case of FTP shows how even adding a data field that would allow for future billing carries with it meaningful decisions about the purpose of the network, the 1980s Internet discussions over network management proved accounting raised serious questions about the boundaries of responsibility for and authority over elements of a networked system.

### 4.1 Management as failure: The visibility of humans at the Network Control Center

As Section 3 explored, there was no obvious way to measure activity on the network and implementing accounting required tackling various trade-offs (for example, losing data). But over time, ARPANET

<sup>28</sup>In the 1980s, “network of networks” was used interchangeably with “Internet” to emphasize the transition from talking about the ARPANET specifically to the workings of connecting ARPA’s sites with other existing networks. For example, in sketching out a proposal for an interagency research institute, Barry Leiner referred to the “Network of networks’ or Internet model of interconnection” [74, p. 3].

engineers also faced changing expectations of users from the network, dictating new needs for accounting that supported diagnostics. Alex McKenzie, who led the measurement efforts at BBN and established the Network Control Center (NCC) there (Section 3), later reflected on the ARPANET's growing pains as users came to expect greater operational reliability from it: "Once a set of host protocols were defined to allow connections... it was remarkable how quickly all of the sites really began to want to view the network as a utility rather than as a research project" [5, p. 11]. Meaning that, although ARPANET was itself envisioned as an ongoing research project in network communication, it also became the infrastructure that supported other research activities. Allowing that level of reliability required proactive management of use issues.

Between 1972 and 1976, the NCC became the focal unit for troubleshooting issues on the ARPANET, gradually expanding support to additional sites alongside the network's expansion [5, p. 15].<sup>29</sup> It stepped into this role "because the users really needed a single point of contact" in case any component of the network failed.<sup>30</sup> The bind facing the NCC team, as McKenzie put it, was that "even though [their] authority didn't expand, [their] responsibility expanded quite a bit" [5, p. 13]. While the NCC had no authority over how a host on the network in another institution operated, they still found themselves answering for issues that arose with its use. The lack of authority that characterized the NCC's efforts to "control" the network pointed to the need to develop clearer protocols for network management.

While the NCC was committed to addressing network failure through proactive management, some ARPANET engineers believed that management itself was a sign that the research project of ARPANET was a failure. In his 2018 book, David Clark, devoting an entire chapter to the topic, defines management as "those aspects of network operation that involve a person" [31, p. 260]. Defined in this way, through the need for human intervention, management is already set up to contrast with what network engineers might consider the core of their work. Clark, who clearly has come to

view management as an important aspect of networking, points out that some engineers see management as a signal for design failure (as opposed to simply the temporary failure of a component of the network) because "a properly designed network should run itself" [31, p. 260]. In other words, developing the infrastructure for accounting that would aid the NCC's management role was to admit that the ARPANET required such ongoing, labor-intensive human operation.<sup>31</sup>

From his present-day vantage point, Clark describes the network engineers who consider management as a necessary aspect of any robust network as "pragmatists." Throughout this paper, our argument has been that designing accounting mechanisms that would enable accountability is especially key for distributed technical systems: In the case of network failures, accounting would allow for inspection after-the-fact. Recognizing human operators performing "management" as part of the network is, therefore, not giving up on the project of network computing, but acknowledging the range of design decisions that have to be addressed to develop appropriate infrastructure.<sup>32</sup> Throughout the 1980s, developing protocols to support recording data relevant to management became increasingly more mainstream at the IETF. In 1988, Vint Cerf circulated a memo on behalf of the Internet Activities Board recommending the adoption of a common network management framework to avoid incompatibility issues across the network [24, p. 4]. This marked a decade-long shift from implementing a patchwork of accounting and monitoring tools to developing an entire conceptual language around the issues of network management.

The new framework divided the meaning of management into five functional areas: "fault management, configuration management, performance management, accounting management, and security management," later formalized as the FCAPS framework [121, p. 8] [31, p. 261].<sup>33</sup> The framework at once narrowed the role of accounting management, defining it as making "it possible to charge users for network resources used and to limit the use of those resources," [121, p. 8] and expanded its centrality by implicitly recognizing that the tools needed to allow for accounting management were also needed to support a variety of other management functions.

Even in the seemingly discounted role of "accounting management" within the FCAPS framework, we can already see hints to the types of management decisions that went beyond network failure diagnostics. We previously argued that accounting, conceived as billing, already surfaced a variety of questions about what it means to share resources over a network. Accounting as management recognized the potential need to limit access to resources. The developing language around network management in the 1980s marked a transition from the early days of management, consisting

<sup>29</sup>McKenzie later speculated that he was tasked with running the NCC during this period because of his personal belief that the ARPANET should become more of a utility focused on operational issues, rather than an experimental research project that tolerated routine disruptions—to "take it very seriously if anything was broken." For context, the extent to which "things were broken" can be seen in some data in RFC 369, "Evaluation of ARPANET Services," which surveyed a couple of months in early 1972, just preceding McKenzie's role at the NCC. The survey highlighted that the reported mean time between failure was at best 2 hours and at worst 5 minutes. The average of time of "trouble free operation" amounted only to 35%, which the RFC described as "a figure untenable for regular user usage" [101, p. 4]. Later on, McKenzie reported that the NCC considered their efforts to treat the network as a utility successful when providing reliability 98-99% of time, which he conceded was not comparable for what constituted reliability for a utility such as electricity [5, p. 15].

<sup>30</sup>We previously discussed Abbate's argument about the "decline of the ideal of resource sharing" as a matter of neglected usability issues (also see Appendix B). The role that the NCC occupied during this time highlights our point about the ways accounting activities (in this case, measurement data that informed diagnostics) constitute a significant aspect of "usability." But it is also worth noting that the NCC's responses to users' demands did not concern creating interfaces for finding resources or navigating their use, but, more explicitly, issues users had with figuring out whether a specific device or service was down for some reason. McKenzie describes the following scenario: "It's absolutely no use to some geophysicist in Washington who's trying to do something in the DARPA seismic monitoring program, to say, 'Well, you can call the Network Control Center, if that doesn't work call MITRE and ask about their modems, and if that doesn't work call ISI, and ask about their computer.' Nobody would work that way, I wouldn't work that way myself" [5, p. 13]. Even when we focus our attention on the experience of users of the network, rather than its engineers, we see that accounting plays a significant role in the ongoing ability to access services reliably.

<sup>31</sup>By the mid-1970s, the NCC was a 3-person team during East and West Coast business hours, and was further operated by 1-2 people at other times, to offer 24-hour, 7-day-a-week coverage [5, p. 13].

<sup>32</sup>Already in 1981, Jon Postel proposed the Internet Control Message Protocol (ICMP), a means to send back information about a lost datagram, recognizing that the network's protocols are "not designed to be absolutely reliable" [105, p. 1]. ICMP's accounting functions allowed individual users to probe the network to identify when a particular host was down, and is used to this day as a "ping" by both professional network managers and ordinary Internet users [31, p. 263]. But such minimal diagnostics implementations still provided a rather limited amount of management tools.

<sup>33</sup>For details on the "Standards War" that took place concerning this management framework, refer to Russell [114].

of identifying failing components on the network and troubleshooting them, to allowing managers to make decisions about network traffic and resource access, to prioritize some users over others.

## 4.2 The “network of networks”: Management without trust

As the scale of the network grew, not only adding new sites to the ARPANET but also incorporating connections with other existing networks, differences in management across these domains came to the fore.<sup>34</sup> Throughout this transition to a “network of networks,” the applicability of a particular management design captured in the FCAPS framework came to define the boundaries of “Administrative Domains” (ADs)—sectors of a network defined precisely through their shared management [121, p. 8].<sup>35</sup> In our discussion of the developing language around management and accounting above, we have treated the kinds of traffic and resource access issues that occur within an AD. But the introduction of *inter-AD* management raised a new set of concerns and competing interests which needed to be accounted for.

In 1983, Barry Leiner took over directing the Internet research program at ARPA and led a broader effort of organizational restructuring, which both he and Clark have attributed to the growth of the network [114, p. 50] [76, p. 29].<sup>36</sup> In a series of RFCs during his tenure, Leiner became a vocal commentator on a variety of accounting concerns that the new inter-AD age raised. In an “idea paper” circulated as an RFC in 1987, Leiner used the experience of the growing Internet to sketch out the challenges facing any kind of inter-agency research network. The conceptual problem at the core of this type of network remained the lack of a “consistent mechanism to allow sharing of the networking resources” [74, p. 1]. In framing the issues facing a “network of networks” in terms of the needs of different research agencies running their own networks, or ADs, Leiner paid close attention to the possibility that intra-AD priorities might conflict with the inter-AD design: “[to] assure appropriate accountability for the network operation, the mechanism for inter-connection must not prevent agencies from retaining control over their individual network” [74, p. 9]. In 1972, Alex McKenzie found himself responsible for the performance of individual components

of the network without the authority to manage them. Here, over a decade later, Leiner raised the reverse concern—that individual agencies would not be able to be accountable to what occurred on their networks due to a lack of authority over aspects of inter-networking activity. Leiner called for a “management approach” that would allow local control while still sharing resources with users sponsored by other agencies [121, p. 11]. In Leiner’s description, necessary tools for individual network managers included both user access control and privacy, and accounting mechanisms “to support both cost allocation and cost auditing” [74, p. 9]. The shift to a “network of networks,” then, only heightened what has already been a persistent, unresolved need for accounting as part of a network management approach.

At the interface between the different ADs of the “network of networks” sit special host computers with the responsibility to route and ability to translate data across domains, called gateways [74, p. 6] [2, pp. 128-129]. Gateways acted as “buffers” between ADs, simplifying the need for each AD to have working knowledge of others [2, p. 129]. The Exterior Gateway Protocol (EGP), introduced in 1982, was a means of allowing this patchwork of ADs to appear to an end-user as “a single internet” [113, p. 3]. EGP contained mechanisms for discovering “neighboring” ADs and allowed those “neighbors” to exchange reachability information—to know whether the “neighbor” was open to receiving traffic. A key feature of EGP was to enable “each gateway to control the rate at which it sends and receives network reachability information, allowing each system to control its own overhead” [113, p. 5]. The resource-sharing concerns we first encountered in Section 2, now cast against the changing Internet environment that expected increasing numbers of internet-networked ADs, took on a new aspect of accounting as management—a baked-in assumption of mistrust between the different networks. If one could not account for users of other networks, then a buffer between these ADs would be needed.

A few years later, David Clark described the purpose of EGP as follows: “to permit regions of the Internet to communicate reachability information, even though they did not totally share trust” [29, p. 2]. This assumption of mistrust cut in two directions: On the one hand, an AD might wish to limit the amount of information other actors on the network would have access to, using gateways as a buffer; on the other, mistrust also required a higher degree of accounting information to allow for individual AD management decisions. EGP was an imperfect solution. It limited the amount of information different ADs had to share in order to still make use of the “network of networks,” but left under-developed the tools of network management and accounting that would allow for the kind of responsibility Leiner was advocating for.<sup>37</sup>

The network still maintained its original goal: being able to connect a large number of systems of different types of networks—to provide high-performance communication and interoperability among diverse hardware. But it now also had the goal of supporting

<sup>34</sup>Throughout this paper, we focus on the ARPANET as a precursor to the Internet, and the differences between the two, while both technical and administrative, are primarily outside the scope of our discussion of accounting and accountability in the development of network computing. However, during the 1980s, the ARPANET, thanks in part to management efforts such as the NCC, was considered relatively reliable; it was internetworked with other, less reliable networks to constitute the early version of what we may consider as “the Internet.” For more on the architectural and institutional challenges involved in creating the Internet, see “Chapter 4: From ARPANET to Internet” in Abbate [2] and “Chapter 7: Alternative Network Architectures” in Clark [31].

<sup>35</sup>While calling different units of the network “Administrative Domains” (ADs) and at times “Administrative Regions” (ARs) emphasized that conflict or incompatibility on the network often stemmed from the differences in management priorities and systems, over time the term was replaced with “Autonomous Systems” (ASs), a term that obfuscated the type of management considerations we explore in this section.

<sup>36</sup>In Leiner’s 2013 posthumous induction into the “Internet Hall of Fame,” the citation credited him as having “helped set up the bureaucratic structures that developed Internet communication protocols” [94]. These bureaucratic structures included the establishment of the Internet Activities Board (IAB) and the task-force structure for tackling specific technical aspects of the network’s development, which ultimately resulted in the IETF. Historian Andrew Russell argues that the Internet’s technical standards governing structure was itself a political innovation, often overlooked by narratives that focus on the innovative technical aspects of the Internet [114]. It is therefore not surprising for Leiner to emerge as one of the more prolific RFC authors that focused on the administrative and accounting needs of the growing network.

<sup>37</sup>In 1989, the Border Gateway Protocol (BGP) was developed, “built on experience gained with EGP” [81, p. 1]. In his 2018 book, David Clark describes the shift to BGP as necessary to support a commercial network that would allow different Internet Service Providers (ISPs) to have non-hierarchical relationships with one another (as opposed to EGP’s more hierarchical structure, with the ARPANET at the core). When discussing BGP’s limitations, however, Clark raises the issue of its “limited expressive power,” providing different ADs (in this case ISPs) with minimal reachability information that could inform decisions about routing traffic through other ADs [31, pp. 244-245].

“Multiple organizations with mutual distrust and policy/legal restrictions” [52, pp. 4-5]. The change in the network environment to one of mistrust unearthed critical architectural problems, as mistrust was fundamentally at odds with central assumptions in the Internet Protocol (IP), the network’s core routing protocol. Routing “determines the series of networks and gateways a packet will traverse in passing from the source to the destination” [29, p. 1]. Embedded in the process is a notion of *how* to determine the series over which traffic should travel, which IP implemented by “minimizing some measure of the route, such as delay” [29, p. 1] and only promising to provide “point-to-point best-effort data delivery” [15, p. 2]. In only needing to satisfy the constraint of minimizing some cost, the selected route could hypothetically use any path to transport a packet from source to destination. This strategy would not be sufficient in an environment of mistrust: A packet traveling between two regions that trusted each other could potentially travel through an untrusted region while in transit. Point-to-point, best-effort service gave ADs no explicit control over how their packets were routed, instead locating best-effort routing decisions in the internals of the network [52, p. 1] [41, p. 5]. This left packet data—and any downstream effects that data might have on the destination’s resources—at risk of interception or tampering.

As the only mechanism for transporting packets between ADs, best-effort routing was therefore in direct tension with the intra-AD management concerns discussed in this section. For an AD’s management goals to be met in the context of a distributed network, it would be important to add additional control mechanisms “to select routes in order to restrict the use of network resources to certain classes of customers” [29, p. 1]. It became clear that administrators needed to be able to set specific *policies* for how to select routes—constraints for how data should traverse the network in order to conform with local resource usage goals. As we discuss in the next section, being able to support policies through network architecture would require fundamental changes. That is, changes for “controlled network resource sharing and transit [would] require that policy enforcement be integrated into the routing protocols themselves and [could] *not be left to network control mechanisms at the end points*” [41, p. 5, emphasis added]. Policies would test the supposed fundamental commitment to End-to-End that had governed the network since its inception [116]. Comprehensive accounting for resource usage would demand placing additional functionality in the network. Being able to connect was no longer enough; ADs needed policies to control how and when that connection occurred.

## 5 ACCOUNTING AS POLICY: FROM ACCOUNTING TO ACCOUNTABILITY

As we have shown in Section 3, in the early days of the ARPANET accounting was considered a “supporting role compared to the core objective of forwarding data” [31, pp. 46-47]. But, as we discussed in Section 4, over the course of the 1980s and growth of the network, accounting, and its role in management, was directly implicated in the network’s core objective of routing. Questions of routing started getting framed as questions of policy, and ARPANET engineers thus began to reckon with the notion that matters of technical design could not be separated from issues like who got to use the network and for what purpose. As David Clark noted back in

1989, “Policy matters are driven by human concerns, and these have not turned out to be amenable to topological constraints, or indeed to constraints of almost any sort” [29, p. 2]. Routing decisions could no longer just be about technical constraints, such as minimizing a cost metric like overall distance; they would also need to incorporate legal and political constraints so that routing could produce “predictable, stable result[s] based on the desires of the administrator” [19, p. 2].

STS scholars have shown that technical and scientific work like this often involves such delineation of social and political aspects as existing beyond the scope of the engineers’ work. Much of the literature produced by these scholars has focused on exploring the inherent embeddedness of technical decisions within social and political structures, clarifying that there is no real way to carve out the *socio* from the *technical* [11, 62, 72]. We can see this insight from STS play out in the network engineers’ attempts to weed policy out of the network’s design—to no avail. Even if policy was a “human concern,” the network still needed a technical architecture in order to implement it.

### 5.1 Policy routing: An implementation for in-network control

By the end of the 1980s, some engineers were arguing that the network needed a “new generation of routing protocols” that would allow each AD “to independently express and enforce policies regarding the flow of packets to, from, and through its resources” [41, p. 1] [43].<sup>38</sup> These new architectural demands to support policies for routing control could not be separated from accounting. Accounting for resources was essential for controlling resources, since it was directly implicated in preventing, tracking, and correcting for unintended use [19, p. 9]. The resulting proposed solution, “Policy Routing” (PR), was the architecture put forth to enable different communication policies between ADs in the network.

A PR consisted of a sequence of gateways from source AD to destination AD; if such a route existed, then the policy associated with that PR was satisfied [20, 29].<sup>39</sup> Such policies came in two flavors: access control and charging, frequently tied to quality of service (QoS) [61, p. 12] [14, p. 35]. Access control policies determined “who [could] use resources and under what conditions” [41, p. 10]. They enabled filtering out traffic considered “administratively inappropriate” [80, p. 6], including blanket policies that defined “users, applications, or hosts” that could never “be permitted to traverse

<sup>38</sup>And ultimately, even though accounting had become more expansive than the act of billing, policy remained inextricably tied to billing: “The discussion of lost packets makes clear an important relationship between billing and policy. If a Policy Route takes packets through a region of known unreliability, the regions preceding it on the path may be quite unwilling to forgive the charges for packets which have successfully crossed their region, only to be lost further down the route. A billing policy is a way of asserting that one region wishes to divorce itself from the reliability behavior of another region. ... The use of a specific policy condition can make clear to the end user which [ADs] do not view themselves as interworking harmoniously” [29, p. 12].

<sup>39</sup>There were various competing proposals on how exactly to implement policy routing. David Clark’s RFC 1102 was among the most-cited, ultimately expanded upon by systems researcher Deborah Estrin [42]. Estrin collaborated on several different policy routing implementation schemes. All proposals, regardless of specifics, had to handle “three design parameters: location of routing decision (i.e., predetermined at the source or hop-by-hop), algorithm used (i.e., link state or distance vector), and expression of policy in topology or in link status” [20, p. 231].

certain segments of the network” [73, p. 9],<sup>40</sup> and finer-grained policies, which could allow a network manager to enforce traffic restrictions on a “particular misbehaving host” [14, p. 34] for a specific period of time. In contrast, charging policies could “be based upon equity (‘fairness’) or upon inequity (‘priority’)” [14, p. 49] of network traffic; they controlled the level of service guarantees for packet-forwarding, ranging from best-effort (with no service guarantees) to prioritized service, in which, for a premium, packets could jump to the head of the forwarding queue.<sup>41</sup> Policies enabling higher-quality service options would be useful for new types of applications, like videoconferencing;<sup>42</sup> such real-time applications are more sensitive to the disruptions common to best-effort service, including unpredictable network latency and delays. Additionally, on a saturated network, the ability to set policies to prioritize service could be useful to ADs that wanted to treat some types of traffic as more important than others. Network engineers were less concerned with discussing these specific policy choices, but rather wanted to ensure that the policy routing architecture was sufficiently flexible to implement a wide range of AD-specific policy requirements [29, p. 5] [41, p. 4].

All of these policies, regardless of access- and QoS-related particulars, fundamentally concerned resource usage, and therefore ultimately implicated accounting. That is, while “Network accounting [was] generally considered to be simply a step that leads to billing,” by the late 1980s it was clear that accounting had much broader utility [73, p. 34], as these same “records [could] also be used to determine usage patterns for the system” [56, p. 28]. Irrespective of whether accounting records were used for billing, they contained useful information concerning resource usage, and therefore were essential for informing policy to control usage [73, p. 35] [29, p. 11] [41, p. 6].<sup>43</sup> For example, records could be used in *post*

<sup>40</sup>While policy routing is not an architectural requirement within today’s Internet, it remains an often-used enterprise solution that can be implemented in routers used by organizations and ISPs. Geoblocking, while similar in purpose to some access-based policies, is a different function for restricting traffic based on a user’s geographical location. Rather than examining the internals of packet flows, traffic is blocked at the source using an IP address (a heuristic for geographic location, since IP addresses are generally assigned by country). Thus, as Clark notes, geoblocking is “supported approximately” on today’s Internet; it is fairly easy to circumvent via using a VPN [31, p. 298]. For a detailed treatment of geoblocking, see Goldsmith and Wu [48]. The recent advent in software-defined networking (SDN) technology indicates that it is possible policy routing will experience renewed interest; SDN could allow for the implementation and enforcement of application-specific policies [31, p. 210] [49].

<sup>41</sup>Even if the division of policy types was clear at a high level, it remained unclear what a reasonable charging policy might be: e.g., even if it was clear *who* was paying for a particular QoS, it was not clear “that the services provided at the network layer [would] map well to the sorts of services that network consumers [were] willing to pay for. ... In the telephone network (as well as public data networks), users pay for End-to-End service and expect good quality service in terms of error rate and delay (and may be unwilling to pay for service that is viewed as unacceptable). In an internetworking environment, the heterogeneous administrative environment combined with the lack of End-to-End control may make this approach infeasible” [19, p. 9].

<sup>42</sup>In the mid 1980s, new technical capabilities for end-node workstations presented the opportunity to develop entirely new classes of applications [15, p. 2]: Videoconferencing could be widely available, not just fodder for flashy demos, as in the late 1960s [8, pp. 138–142, Mother of All Demos]. However, new workstation technology would not alone be sufficient to support such demanding applications; the network would also need to update its protocols. As mentioned above, best-effort routing is not well-suited to real-time applications like videoconferencing; packets get lost or dropped due to congestion; there are unpredictable periods of slow service due to packet queuing delays. Supporting additional qualities of service in network protocols would better accommodate these new hardware- and software-enabled applications.

<sup>43</sup>Cost recovery and billing remained important considerations of policy: “Almost all of the existing Internet has been paid for as capital purchase and provided to the users as a free good. There are limited examples of cost recovery, but these are based on

*hoc* audits to confirm that actual resource usage aligned with set access and QoS policies [56, pp. 49,66,77] [75, p.11].<sup>44</sup> In particular, “unusual accounting records [could] indicate unauthorized use of the system” [56, p. 28]. If a pattern of “malicious” [75, p. 11] use or other of “abuse (e.g., unauthorized use) develop[ed], an accounting system could track this and allow corrective action to be taken, by changing routing policy or imposing access control (blocking hosts or nets)” [19, p. 10]. Routing policy tied together accounting information with the “human concerns” of how data moved around the network and introduced, into the architecture, considerations that conflicted with strict notions of efficiency.

## 5.2 An unaccountable network: Enabling accountability required designing for accounting

This connection between accounting and enforcement of resource control policies brought about the necessary conditions to produce the first working definition of accountability in relation to the network. In RFC 1125, networking researcher Deborah Estrin made the interdependence between accounting, policy, and accountability unimpeachably clear:

One way of reducing the compromise of autonomy associated with interconnection is to implement mechanisms that assure accountability for resources used. Accountability may be enforced a priori, e.g., access control mechanisms applied before resource usage is permitted. Alternatively, accountability may be enforced after the fact, e.g., record keeping or metering that supports detection and provides evidence to third parties (i.e., non-repudiation). Accountability mechanisms can also be used to provide feedback to users as to consumption of resources. ... [I]t becomes more appropriate to have resource usage visible to users, whether or not actual charging for usage takes place [41, p. 6].

In short, achieving accountability in the network meant being able to implement policies for dealing with resource misuse. While some policies emphasized prevention and others concerned identifying, isolating, and mitigating misuse after-the-fact, all policies were ultimately dependent on accounting records [19, p. 9].<sup>45</sup>

an annual subscription fee rather than a charge related to the utilization. *There is a growing body of opinion which says that accounting for usage, if not billing for it, is an important component of resource management. For this reason, tools for accounting and billing must be a central part of any policy mechanism*” [29, p. 11, emphasis added]. And yet, just as the early conversations over billing in the 1970s indicated (Section 2), setting an overarching billing policy is especially challenging in a distributed network, which was made even more difficult when different regions of the network clarified their individual administrative needs (Section 4): “However, precisely because the administrative regions are autonomous, we cannot impose a uniform form of billing policy on all of the regions...The billing problem is thus a very complicated one, for the user would presumably desire to minimize the cost, in the context of the various outstanding conditions” [29, p. 11].

<sup>44</sup>This same data, aside from setting and enforcing policy, also retained its overarching role of allowing “network management personnel to determine the ‘flows’ of data on the network, and the identification of bottlenecks in network resources,” [75, p. 11] meaning it continued to support monitoring and diagnostics for operational reasons.

<sup>45</sup>RFC 1104 tries to distinguish the function of accounting from the function of policy-based routing. In the process, it becomes clear that, while accounting’s role is more expansive than its implications in policy-based routing, policy-based routing entirely depends on accounting: “Accounting vs. Policy Based Routing: Quite often Accounting and Policy Based Routing are discussed together. While the application of both Accounting and Policy Based Routing is to control access to scarce network resources, these are separate (but related) issues. The chief difference between Accounting and Policy Based Routing is that Accounting combines history information with policy information to track network usage for various purposes. Accounting information may in turn drive policy mechanisms (for instance, one could imagine a policy limiting a

In showing the importance of accounting for enabling accountability, Estrin’s RFC 1125 captures the fundamental argument that we have taken up in this paper: The tensions that come up in a resource-sharing, networked computing environment (Sections 2 & 3), in which there is no global authority (Section 4), ultimately reflect tensions concerning accountability and autonomy. While we have shown throughout this paper that issues of resource management were never not contentious, it was at this point in time that the network architects and engineers found accounting-related issues sufficiently important to be considered at the level of architecture, instead of an annoyance. In Estrin’s words, “the lack of global authority, the need to support network resource sharing as well as network interconnection, the complex and dynamic mapping of users to ADs and rights, and the need for accountability across ADs, are characteristics of inter-AD communications which must be taken into account in the design of both policies and supporting technical mechanisms”; “it would be inexcusable to ignore resource control requirements and not to pay careful attention to their specification” [41, pp. 6,7].

Rather than accounting not meeting the parsimony requirements of End-to-End, accounting’s necessary role in accountability meant that the engineers promoting a policy routing architecture—including, notably, End-to-End [116] author and network architect David Clark [29]—were willing to consider accounting features as sufficiently fundamental to incorporate within the network’s foundational routing protocol. This would require placing mechanisms for accounting in the network, instead of just at the end nodes. This perspective marked a significant shift from the early days of the ARPANET, well-characterized in Bob Kahn’s RFC 136 (Section 3). Before, accounting was discounted as not being a part of the network’s core architecture, with respect to the End-to-End principle: It was separated from the function of routing and was not a “performance enhancement” [65, 116], and therefore was not considered a candidate feature of the network’s essential architecture. By 1988, recognizing accounting’s necessity for accountability placed accountability directly at odds with this earlier interpretation of End-to-End. To resolve this tension by choosing to acknowledge accountability as an essential feature, the network’s engineers would be forced to re-imagine the meaning and primacy of End-to-End. For an accountable network, the next evolution of its architecture would need to be shaped by a new, competing ideology.

## 6 CONCLUSION

In this paper, we have traced the changing meaning of the term “accounting” among the research community which developed the ARPANET and early Internet, from the late 1960s and through the end of the 1980s, with a particular focus on the RFC corpus from this time period. We have paid attention both to how accounting was and was not considered part of the set of research problems facing network engineers, and to how the meaning of “accounting” shifted in relation to the changing environment of the network’s deployment and institutional context. We characterized four notions of accounting within ARPANET and early Internet RFCs—accounting as *billing*, accounting as *measurement*, accounting as *management*,

and accounting as *policy*—and argued that the different meanings of accounting and the stakes of debates concerning each theme provide an emerging articulation of accountability. Our analysis places accounting and its administrative associations squarely within the domain of both deep technical questions about the network and the political expectations of accountable technical systems.

This analysis resonates beyond the historiographical considerations of the Internet’s development and the contemporary policy issues of its governance. It demonstrates that accountability is not a pre-given quality of a system that can be designed from the get-go, but rather is a set of emergent attributes that are negotiated throughout the design process—a process that itself is never fully settled in deployed systems like the Internet. From this starting point, we can see why technical systems that are now identified as unaccountable cannot be remedied simply by calls to implement greater accountability, which also run the risk of being mistakenly equated with a call to record everything as an audit best practice. Rather, taking stock of accountability as the result of complex negotiations, in the ways we have done above, means first and foremost recognizing whose interests are accounted for in a given accountability regime.

We close by offering three insights from this discussion that have bearing on research concerning accountability and complex technical systems, which together serve as a cautionary tale for the design of contemporary computer systems. First, the core design question of the story above is about resource sharing and allocation, a motivating problem in many computing applications today, especially in machine learning. The non-trivial questions involved in designing mechanisms to account for resource use in a distributed system are constitutive of the possibility of creating accountability. It is thus worth interrogating how mechanisms for accounting may facilitate or obscure accountability. Second, we argued that developing accounting mechanisms was routinely deferred, not only because of the foundational tensions involved in developing accounting schemes, but because of a dynamic of discounting its significance. Accounting used the administrative language around issues of delineating different actors’ autonomy and negotiating trust, which created a policy of prioritization, allowing such issues to be dismissed as “operational,” beyond the core set of research concerns, even though it was clear that distributed accounting for resource sharing was a fundamental challenge in the field of networking. Finally, by taking an institutional approach to the history of technical objects, we argued that the broader context in which technological systems develop shapes the ways accountability is defined and implemented (or ignored). Taking the administrative needs of this setting seriously, rather than casting them aside as overhead concerns, would help clarify the necessary infrastructure to support concrete notions of accountability in complex technical systems. Even within the same technical system—the Internet—we see how over time institutional changes led to shifts in understandings of accounting and the demands of accountability. Designing a definitive “one-size-fits-all” technical spec for accountability, thus, would offer an abstract solution to a set of complex questions that ought to be addressed within specific institutional contexts.

certain organization to a fixed aggregate percentage of dynamically shared bandwidth). Conversely, policy information may affect accounting issues” [19, p. 9].

## ACKNOWLEDGMENTS

A. Feder Cooper is supported by the Artificial Intelligence Policy and Practice initiative at Cornell University, the Digital Life Initiative at Cornell Tech, and the John D. and Catherine T. MacArthur Foundation. The authors would like to thank the Artificial Intelligence, Policy, and Practice initiative, the Fidelity, Authenticity, and Knowability in Electronic media Lab, and the Qualitative and Interpretive Research Institute at Cornell, FAcCT's anonymous reviewers, and the following individuals for feedback on earlier versions of this work: Harry Auster, Salonee Bhaman, Solon Barocas, Madiha Zahrah Choksi, Fernando Delgado, Abigail Z. Jacobs, Devin Kennedy, Jon Kleinberg, Karen Levy, Pegah Moradi, and Malte Ziewitz.

## REFERENCES

- [1] Daniel Abadi. 2012. Consistency Tradeoffs in Modern Distributed Database System Design: CAP is Only Part of the Story. *Computer* 45, 2 (Feb. 2012), 37–42.
- [2] Janet Abbate. 1999. *Inventing the Internet*. MIT Press, Cambridge, MA, USA.
- [3] Philip Adler, Casey Falk, Sorelle A. Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. 2018. Auditing black-box models for indirect influence. *Knowledge and Information Systems* 54 (2018), 95–122.
- [4] P. Almqvist and F. Kastenholz. 1994. Towards Requirements for IP Routers. RFC 1716. <https://doi.org/10.17487/RFC1716>
- [5] Alexander Anderson. 1990. Oral history interview with Alexander A. McKenzie. <https://hdl.handle.net/11299/107489> Retrieved from the University of Minnesota Digital Conservancy.
- [6] William Aspray and Paul E. Ceruzzi. 2008. *The Internet before Commercialization*. The MIT Press, Cambridge, MA, USA, 9–43.
- [7] Athena Project. 1989. Teaching Students About Responsible Use of Computers. *Communications of the ACM* 32, 6 (June 1989), 704.
- [8] Thierry Bardini. 2001. *Bootstrapping: Douglas Engelbart, Coevolution, and the Origins of Personal Computing*. Stanford University Press, USA.
- [9] Abhay Bhushan. 1972. File Transfer Protocol. RFC 354. <https://doi.org/10.17487/RFC0354>
- [10] Abhay K. Bhushan. 1972. Comments on the File Transfer Protocol. RFC 385. <https://doi.org/10.17487/RFC0385>
- [11] Wiebe E. Bijker, Thomas P. Hughes, and Trevor J. Pinch. (Eds.). 1987. *The Social construction of technological systems: New directions in the sociology and history of technology*. MIT Press, Cambridge, Mass.
- [12] Marjory S. Blumenthal and David D. Clark. 2001. Rethinking the Design of the Internet: The End-to-End Arguments vs. the Brave New World. *ACM Trans. Internet Technol.* 1, 1 (Aug. 2001), 70–109. <https://doi.org/10.1145/383034.383037>
- [13] Mark Bovens. 2007. Analysing and Assessing Accountability: A Conceptual Framework. *European Law Journal* 13, 4 (2007), 447–468. <https://doi.org/10.1111/j.1468-0386.2007.00378.x>
- [14] R. Braden and J. Postel. 1987. Requirements for Internet gateways. RFC 1009. <https://doi.org/10.17487/RFC1009>
- [15] Robert T. Braden, Dr. David D. Clark, and Scott Shenker. 1994. Integrated Services in the Internet Architecture: an Overview. RFC 1633. <https://doi.org/10.17487/RFC1633>
- [16] Sandra Braman. 2010. The Interpenetration of Technical and Legal Decision-Making for the Internet. *Information, Communication & Society* 13, 3 (2010), 309–324.
- [17] Sandra Braman. 2011. The Framing Years: Policy Fundamentals in the Internet Design Process, 1969–1979. *The Information Society* 27, 5 (2011), 295–310.
- [18] Sandra Braman. 2013. Laying the path: Governance in early internet design. *Info (Cambridge, England)* 15, 6 (2013), 63–83.
- [19] Hans-Werner Braun. 1989. Models of policy based routing. RFC 1104. <https://doi.org/10.17487/RFC1104>
- [20] L. Breslau and D. Estrin. 1990. Design of Inter-Administrative Domain Routing Protocols. In *Proceedings of the ACM Symposium on Communications Architectures & Protocols* (Philadelphia, Pennsylvania, USA) (SIGCOMM '90). Association for Computing Machinery, New York, NY, USA, 231–241. <https://doi.org/10.1145/99508.99558>
- [21] Bob Bressler. 1973. Free file transfer. RFC 487. <https://doi.org/10.17487/RFC0487>
- [22] Finn Brunton. 2013. *Spam: A Shadow History of the Internet*. The MIT Press, Cambridge, Massachusetts.
- [23] C. Stephen Carr, Stephen D. Crocker, and Vinton G. Cerf. 1970. HOST-HOST Communication Protocol in the ARPA Network. In *Proceedings of the May 5-7, 1970, Spring Joint Computer Conference* (Atlantic City, New Jersey) (AFIPS '70 (Spring)). Association for Computing Machinery, New York, NY, USA, 589–597. <https://doi.org/10.1145/1476936.1477024>
- [24] Vint Cerf. 1988. IAB recommendations for the development of Internet network management standards. RFC 1052. <https://doi.org/10.17487/RFC1052>
- [25] Vint Cerf. 1989. Requeium for the ARPANET. In *The Internet History*, 7–10. <https://archive.computerhistory.org/resources/access/text/2017/11/102646703-05-01-acc.pdf> Accessed January 9, 2022.
- [26] V. Cerf and R. Kahn. 1974. A Protocol for Packet Network Intercommunication. *IEEE Transactions on Communications* 22, 5 (1974), 637–648. <https://doi.org/10.1109/TCOM.1974.1092259>
- [27] Danielle Keats Citron and Daniel J. Solove. 2022. Privacy Harms. *Boston University Law Review* 102 (2022), 62.
- [28] D. Clark. 1988. The Design Philosophy of the DARPA Internet Protocols. In *Symposium Proceedings on Communications Architectures and Protocols* (Stanford, California, USA) (SIGCOMM '88). Association for Computing Machinery, New York, NY, USA, 106–114. <https://doi.org/10.1145/52324.52336>
- [29] Dave Clark. 1989. Policy routing in Internet protocols. RFC 1102. <https://doi.org/10.17487/RFC1102>
- [30] David Clark and Susan Landau. 2011. Untangling Attribution. *Proceedings of A Workshop on Deterring Cyberattacks: Informing Strategies and Developing Options for US Policy* (10 2011), 25–40.
- [31] David D. Clark. 2018. *Designing an Internet* (1st ed.). The MIT Press, Cambridge, MA, USA.
- [32] A. Feder Cooper, Karen Levy, and Christopher De Sa. 2021. Accuracy-Efficiency Trade-Offs and Accountability in Distributed ML Systems. In *Equity and Access in Algorithms, Mechanisms, and Optimization*. Association for Computing Machinery, New York, NY, USA, Article 4, 11 pages. <https://doi.org/10.1145/3465416.3483289>
- [33] A. Feder Cooper, Emanuel Moss, Benjamin Laufer, and Helen Nissenbaum. 2022. Accountability in an Algorithmic Society: Relationality, Responsibility, and Robustness in Machine Learning. <https://doi.org/10.48550/ARXIV.2202.05338>
- [34] CPSR. 1989. CPSR Statement on the Computer Virus. *Communications of the ACM* 32, 6 (June 1989), 699.
- [35] D. Crocker, N. Neigus, J. Feinler, and J. Iseli. 1973. ARPANET users interest working group meeting. RFC 585. <https://doi.org/10.17487/RFC0585>
- [36] S. Crocker. 1970. Network Meeting. RFC 75. <https://doi.org/10.17487/RFC0075>
- [37] S. Crocker, S. Carr, and V. Cerf. 1970. New Host-Host Protocol. RFC 33. <https://doi.org/10.17487/RFC0033>
- [38] G. Deloche. 1969. Implementation of the Host - Host Software Procedures in GORDO. RFC 11. <https://doi.org/10.17487/RFC0011>
- [39] Laura DeNardis. 2015. The Internet Design Tension between Surveillance and Security. *IEEE Annals of the History of Computing* 37, 2 (2015), 72–83. <https://doi.org/10.1109/MAHC.2015.29>
- [40] Edwin W. Meyer, Jr. 1970. Network Meeting Notes. RFC 82. <https://doi.org/10.17487/RFC0082>
- [41] Deborah Estrin. 1989. Policy requirements for inter Administrative Domain routing. RFC 1125. <https://doi.org/10.17487/RFC1125>
- [42] Deborah Estrin and Martha Steenstrup. 1991. Inter Domain Policy Routing: Overview of Architecture and Protocols. *SIGCOMM Comput. Commun. Rev.* 21, 1 (Jan. 1991), 71–78. <https://doi.org/10.1145/116030.116036>
- [43] Deborah Estrin and Gene Tsudik. 1989. Security Issues in Policy Routing. In *Proceedings of 1980 IEEE Symposium on Security and Privacy*.
- [44] Larry Garlick. 1976. Out-of-Band Control Signals in a Host-to-Host Protocol. RFC 721. <https://doi.org/10.17487/RFC0721>
- [45] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for Datasets. <http://arxiv.org/abs/1803.09010>
- [46] Tarleton Gillespie. 2006. Engineering a Principle: 'End-to-End' in the Design of the Internet. *Social Studies of Science* 36, 3 (2006), 427–457.
- [47] Tarleton Gillespie. 2010. The politics of 'platforms'. *New Media & Society* 12, 3 (2010), 347–364. <https://doi.org/10.1177/1461444809342738>
- [48] Jack L. Goldsmith and Tim Wu. 2006. *Who Controls the Internet?: Illusions of a Borderless World*. Oxford University Press, Oxford, UK.
- [49] Albert Greenberg, David A. Maltz, Jennifer Rexford, Geoffrey Xie, Jibin Zhan, and Hui Zhang. 2019. Reflections on a Clean Slate 4D Approach to Network Control and Management. *SIGCOMM Comput. Commun. Rev.* 49, 5 (Nov. 2019), 90–91. <https://doi.org/10.1145/3371934.3371962>
- [50] Shane Greenstein. 2015. *How the Internet Became Commercial: Innovation, Privatization, and the Birth of a New Network*. Princeton University Press, Princeton, NJ.
- [51] Sally Hambridge and Albert Lunde. 1999. DON'T SPEW A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam<sup>®</sup>). RFC 2635. <https://doi.org/10.17487/RFC2635>
- [52] S. Hares and D. Katz. 1989. Administrative Domains and Routing Domains: A model for routing in the Internet. RFC 1136. <https://doi.org/10.17487/RFC1136>
- [53] J. Heafner and E. Harslem. 1971. Service center standards for remote usage: A user's view. RFC 231. <https://doi.org/10.17487/RFC0231>
- [54] F. E. Heart, R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden. 1970. The Interface Message Processor for the ARPA Computer Network. In

- Proceedings of the May 5-7, 1970, Spring Joint Computer Conference* (Atlantic City, New Jersey) (AFIPS '70 (Spring)). Association for Computing Machinery, New York, NY, USA, 551–567. <https://doi.org/10.1145/1476936.1477021>
- [55] Bob Hinden. 1983. Host Monitoring Protocol. RFC 869. <https://doi.org/10.17487/RFC0869>
- [56] Paul Holbrook and Joyce K. Reynolds. 1991. Site Security Handbook. RFC 1244. <https://doi.org/10.17487/RFC1244>
- [57] Philip J. Nesser II. 1999. The Internet and the Millennium Problem (Year 2000). RFC 2626. <https://doi.org/10.17487/RFC2626>
- [58] Internet Activities Board. 1989. Ethics and the Internet. *Communications of the ACM* 32, 6 (June 1989), 710.
- [59] Internet Activities Board. 1989. Ethics and the Internet. RFC 1087. <https://doi.org/10.17487/RFC1087>
- [60] ISO. 1983. ISO Transport Protocol specification. RFC 892. <https://doi.org/10.17487/RFC0892>
- [61] ISO. 1984. Protocol for providing the connectionless mode network services. RFC 926. <https://doi.org/10.17487/RFC0926>
- [62] Sheila Jasanoff. 2004. *States of Knowledge: The Co-Production of Science and the Social Order*. Routledge, Abingdon, Oxon.
- [63] Paul R. Johnson and Robert H. Thomas. 1975. Maintenance of duplicate databases. RFC 677. <https://doi.org/10.17487/RFC0677>
- [64] Severin Kacianka and Alexander Pretschner. 2021. Designing Accountable Systems. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) (FAcCT '21). Association for Computing Machinery, New York, NY, USA, 424–437. <https://doi.org/10.1145/3442188.3445905>
- [65] Robert Kahn. 1971. Host accounting and administrative procedures. RFC 136. <https://doi.org/10.17487/RFC0136>
- [66] P. Karp. 1971. Categorization and guide to NWG/RFCs. RFC 100. <https://doi.org/10.17487/RFC0100>
- [67] Been Kim and Finale Doshi-Velez. 2021. Machine Learning Techniques for Accountability. *AI Magazine* 42, 1 (April 2021), 47–52. <https://ojs.aaai.org/index.php/aimagazine/article/view/7481>
- [68] Kate Klonick. 2020. The Facebook Oversight Board: Creating an Independent Institution to Adjudicate Online Free Expression. *Yale Law Journal* 129 (2020), 82. Issue 2418.
- [69] Jeff Kosseff. 2022. A User's Guide to Section 230, and a Legislator's Guide to Amending It (or Not). *Berkeley Technology Law Journal* 37 (2022), 40. Issue 2.
- [70] Joshua A. Kroll. 2021. Outlining Traceability: A Principle for Operationalizing Accountability in Computing Systems. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) (FAcCT '21). Association for Computing Machinery, New York, NY, USA, 758–771. <https://doi.org/10.1145/3442188.3445937>
- [71] Joshua A. Kroll, Joanna Huey, Solon Barocas, Edward W. Felten, Joel R. Reidenberg, David G. Robinson, and Harlan Yu. 2017. Accountable Algorithms. *University of Pennsylvania Law Review* 165 (2017), 633–705. Issue 633.
- [72] Bruno Latour. 1993. *We have never been modern*. Harvard University Press, Cambridge, Mass.
- [73] Barry Leiner. 1988. Critical issues in high bandwidth networking. RFC 1077. <https://doi.org/10.17487/RFC1077>
- [74] Barry M. Leiner. 1987. Implementation plan for interagency research Internet. RFC 1015. <https://doi.org/10.17487/RFC1015>
- [75] Barry M. Leiner. 1987. Network requirements for scientific research: Internet task force on scientific computing. RFC 1017. <https://doi.org/10.17487/RFC1017>
- [76] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. 2009. A Brief History of the Internet. *SIGCOMM Comput. Commun. Rev.* 39, 5 (oct 2009), 22–31. <https://doi.org/10.1145/1629607.1629613>
- [77] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Lawrence G. Roberts, and Stephen S. Wolff. 1997. The Past and Future History of the Internet. *Commun. ACM* 40, 2 (feb 1997), 102–108. <https://doi.org/10.1145/253671.253741>
- [78] Lawrence Lessig. 2006. *Code version 2.0* ([2nd ed.], ed.). Basic Books, New York.
- [79] J. C. R. Licklider and Welden E. Clark. 1962. On-Line Man-Computer Communication. In *Proceedings of the May 1-3, 1962, Spring Joint Computer Conference* (San Francisco, California) (AIEE-JRE '62 (Spring)). Association for Computing Machinery, New York, NY, USA, 113–128. <https://doi.org/10.1145/1460833.1460847>
- [80] Mike Little. 1989. Goals and functional requirements for inter-autonomous system routing. RFC 1126. <https://doi.org/10.17487/RFC1126>
- [81] K. Loughheed and Y. Rekhter. 1989. Border Gateway Protocol (BGP). RFC 1105. <https://doi.org/10.17487/RFC1105>
- [82] Thomas Marill and Lawrence G. Roberts. 1966. Toward a Cooperative Network of Time-Shared Computers. In *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference* (San Francisco, California) (AFIPS '66 (Fall)). Association for Computing Machinery, New York, NY, USA, 425–431. <https://doi.org/10.1145/1464291.1464336>
- [83] Charlton D. McIlwain. 2020. *Black software: The Internet and racial justice, from the AfroNet to Black Lives Matter*. Oxford University Press, New York, NY.
- [84] Fenwick McKelvey. 2018. *Internet Daemons: Digital Communications Possessed*. University of Minnesota Press, USA.
- [85] Eden Medina. 2011. *Cybernetic revolutionaries: Technology and politics in Al-lende's Chile*. MIT Press, Cambridge, Mass.
- [86] R. Merryman. 1973. UCSD-CC Server-FTP facility. RFC 532. <https://doi.org/10.17487/RFC0532>
- [87] Jacob Metcalf, Emanuel Moss, Elizabeth Anne Watkins, Ranjit Singh, and Madeleine Clare Elish. 2021. Algorithmic Impact Assessments and Accountability: The Co-Construction of Impacts. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) (FAcCT '21). Association for Computing Machinery, New York, NY, USA, 735–746. <https://doi.org/10.1145/3442188.3445935>
- [88] Esther Milne. 2021. *Email and the Everyday: Stories of Disclosure, Trust, and Digital Labor*. MIT Press, Cambridge.
- [89] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasseran, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) (FAT\* '19). Association for Computing Machinery, New York, NY, USA, 220–229. <https://doi.org/10.1145/3287560.3287596>
- [90] Emanuel Moss, Elizabeth Anne Watkins, Ranjit Singh, Madeleine Clare Elish, and Jacob Metcalf. 2021. Assembling Accountability: Algorithmic Impact Assessment for the Public Interest. SSRN preprint.
- [91] National Science Foundation. 1989. NSF Poses Code of Networking Ethics. *Communications of the ACM* 32, 6 (June 1989), 688.
- [92] Daniel Neyland. 2016. Bearing Account-able Witness to the Ethical Algorithmic System. *Science, Technology, & Human Values* 41, 1 (2016), 50–76. <https://doi.org/10.1177/0162243915598056>
- [93] Helen Nissenbaum. 1996. Accountability in a Computerized Society. *Science and Engineering Ethics* 2 (1996), 25–42.
- [94] Internet Hall of Fame. 2013. Barry Leiner. <https://www.internethalloffame.org/inductees/barry-leiner>
- [95] M.A. Padlipsky. 1973. Tentative proposal for a Unified User Level Protocol. RFC 451. <https://doi.org/10.17487/RFC0451>
- [96] M. Padlipsky. 1974. Specification of the Unified User-Level Protocol. RFC 666. <https://doi.org/10.17487/RFC0666>
- [97] M. A. Padlipsky. 1973. Feast or famine? A response to two recent RFC's about network information. RFC 531. <https://doi.org/10.17487/RFC0531>
- [98] M. A. Padlipsky. 1973. Two solutions to a file transfer access problem. RFC 505. <https://doi.org/10.17487/RFC0505>
- [99] M. A. Padlipsky. 1973. What is "Free"? RFC 491. <https://doi.org/10.17487/RFC0491>
- [100] Benjamin Peters. 2008. *How not to network a nation: The uneasy history of the Soviet internet*. MIT Press, Cambridge, Massachusetts.
- [101] J. Pickens. 1972. Evaluation of ARPANET services January-March, 1972. RFC 369. <https://doi.org/10.17487/RFC0369>
- [102] K.T. Pogran. 1973. Un-muddling "free file transfer". RFC 501. <https://doi.org/10.17487/RFC0501>
- [103] David G Post. 2009. *In search of Jefferson's moose: Notes on the state of cyberspace*. Oxford University Press, Oxford ; New York.
- [104] J. Postel. 1970. Network meeting report. RFC 77. <https://doi.org/10.17487/RFC0077>
- [105] J. Postel. 1981. Internet Control Message Protocol. RFC 777. <https://doi.org/10.17487/RFC0777>
- [106] J. Postel. 1982. Summary of computer mail services meeting held at BBN on 10 January 1979. RFC 808. <https://doi.org/10.17487/RFC0808>
- [107] Jonathan B. Postel. 1980. Internet Message Protocol. RFC 759. <https://doi.org/10.17487/RFC0759>
- [108] Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) (FAT\* '20). Association for Computing Machinery, New York, NY, USA, 33–44. <https://doi.org/10.1145/3351095.3372873>
- [109] Jennifer Rexford and Constantine Dovrolis. 2010. Future Internet architecture: clean-slate versus evolutionary research. *Commun. ACM* 53 (September 2010), 36–40. Issue 9. <https://doi.org/10.1145/1810891.1810906>
- [110] Joyce K. Reynolds. 1989. Helminthiasis of the Internet. RFC 1135. <https://doi.org/10.17487/RFC1135>
- [111] Lawrence G. Roberts. 1967. Multiple Computer Networks and Intercomputer Communication. In *Proceedings of the First ACM Symposium on Operating System Principles (SOSP '67)*. Association for Computing Machinery, New York, NY, USA, 3.1–3.6. <https://doi.org/10.1145/800001.811680>
- [112] Lawrence G. Roberts and Barry D. Wessler. 1970. Computer Network Development to Achieve Resource Sharing. In *Proceedings of the May 5-7, 1970, Spring Joint Computer Conference* (Atlantic City, New Jersey) (AFIPS '70 (Spring)). Association for Computing Machinery, New York, NY, USA, 543–549. <https://doi.org/10.1145/1464291.1464336>

- [//doi.org/10.1145/1476936.1477020](https://doi.org/10.1145/1476936.1477020)
- [113] Eric C. Rosen. 1982. Exterior Gateway Protocol (EGP). RFC 827. <https://doi.org/10.17487/RFC0827>
- [114] Andrew L. Russell. 2006. 'Rough Consensus and Running Code' and the Internet-OSI Standards War. *IEEE Ann. Hist. Comput.* 28, 3 (jul 2006), 48–61. <https://doi.org/10.1109/MAHC.2006.42>
- [115] Andrew L. Russell. 2014. *Open Standards and the Digital Age: History, Ideology, and Networks*. Cambridge University Press, Cambridge.
- [116] J. H. Saltzer, D. P. Reed, and D. D. Clark. 1984. End-to-End Arguments in System Design. *ACM Trans. Comput. Syst.* 2, 4 (nov 1984), 277–288. <https://doi.org/10.1145/357401.357402>
- [117] Rebecca Slayton. 2016. *Framing Computer Security and Privacy, 1967–1992*. Association for Computing Machinery and Morgan & Claypool, 287–330.
- [118] Bob Thomas. 1973. Distributed resources workshop announcement. RFC 504. <https://doi.org/10.17487/RFC0504>
- [119] Fred Turner. 2006. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. University of Chicago Press, Chicago, IL, USA.
- [120] Briana Vecchione, Karen Levy, and Solon Barocas. 2021. Algorithmic Auditing and Social Justice: Lessons from the History of Audit Studies. In *Equity and Access in Algorithms, Mechanisms, and Optimization* (–, NY, USA) (EAAMO '21). Association for Computing Machinery, New York, NY, USA, Article 19, 9 pages. <https://doi.org/10.1145/3465416.3483294>
- [121] U. Warrior and L. Besaw. 1989. Common Management Information Services and Protocol over TCP/IP (CMOT). RFC 1095. <https://doi.org/10.17487/RFC1095>
- [122] R. Watson. 1973. Some thoughts on system design to facilitate resource sharing. RFC 592. <https://doi.org/10.17487/RFC0592>
- [123] Richard W. Watson. 1971. Notes on the Network Working Group meeting, Urbana, Illinois, February 17, 1971. RFC 101. <https://doi.org/10.17487/RFC0101>
- [124] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James Hendler, and Gerald Jay Sussman. 2008. Information Accountability. *Commun. ACM* 51, 6 (jun 2008), 82–87. <https://doi.org/10.1145/1349026.1349043>
- [125] James E. White. 1971. Network specifications for UCSB's Simple-Minded File System. RFC 122. <https://doi.org/10.17487/RFC0122>
- [126] James E. White. 1973. Responses to critiques of the proposed mail protocol. RFC 555. <https://doi.org/10.17487/RFC0555>
- [127] Maranke Wieringa. 2020. What to Account for When Accounting for Algorithms: A Systematic Literature Review on Algorithmic Accountability. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) (FAT\* '20). Association for Computing Machinery, New York, NY, USA, 1–18. <https://doi.org/10.1145/3351095.3372833>
- [128] JoAnne Yates. 2019. *Engineering rules: Global standard setting since 1880*. Johns Hopkins University Press, Baltimore, Maryland.
- [129] Jonathan Zittrain. 2008. *The Future of the Internet—And How to Stop It*. Yale University Press, USA.
- [130] Jonathan Zittrain. 2014. The Virtues of Procrastination. <https://www.internetimpossible.org/virtues-of-procrastination/> Accessed January 21, 2022.

## A THE “RESEARCH” VS. “SERVICE” DISTINCTION

It is important to note that Bressler [21] and Pogran [102] were not alone in conceiving of rationales for circumventing accounting; punting on both implementing and performing accounting for certain aspects of network use was a common theme in the early ARPANET. Most notably, there was an overarching attempt to distinguish categories of work on the network, such that some preferred types of work could be “free”—that is, have the bill covered by ARPA. In response to this desire, a binary distinction emerged at the site level—“Research Centers vs. Service Centers” [104, p.1]; or, “free but limited access research sites” used strictly for experimental purposes and “billing sites” [40, p. 3, p. 18]. Early Internet pioneer Jon Postel further refined this classification in relation to hardware and access patterns: “The Service Centers tend to have big machines, lots of users, and accounting problems; while the Research Centers tend to have specialized hardware, a small number of users, and no accounting at all” [104, p.1]. Even with these definitions, it is not entirely clear what “service” meant in terms of function, aside from the common need for accounting in order to support site use. The word “service” remains simultaneously vague and overloaded,

both during the 1970s when ARPANET engineers were teasing out research/service distinction and with respect to contemporary use. For example, in his 2018 book, David Clark notes that “service” is a term that has repeatedly confused him, and that he makes sense of it by reducing it to the following: “A service is something that you sell; it is how you make money” [31, p. 311].

One can attempt to elicit a far more precise understanding of service from Service Center examples, which included the Network Information Center (NIC) at SRI, Multics at MIT [123, p. 13], UCSB's Simple-Minded File System (SMFS, which provided a secondary storage node that anyone on the ARPANET could pay to use; it had limited availability and was not intended to become the storage node for the whole network) [125], and UCSD's FTP service site, which had to bill for usage in order to support itself (each FTP file transfer was billed separately in the accounting system, based on lower-level accounting for processor, I/O, and core usage, and, if used, external storage tapes) [86].

However, these varied examples of site functions, particularly the inclusion of FTP, the costs of which were debated in unique detail (Section 2.1), indicate that what constituted the distinction between research and service was contentious and unclear.<sup>46</sup> Moreover, it was additionally not trivial to consider how billing should be handled when research and service sites resource share. Notably, Larry Roberts acknowledged this, but did not provide a clear idea on how to resolve it: “What happens when a research site talks to a billing site? I think it is do-able” [40, p. 3]. The UCSD FTP site, at least by 1973, had to account for usage and issue bills in order to support itself (it is unclear if this site had an option to use a FREE account for research purposes). However, as the RFCs debating “free” file transfer show, this need for accounting did not extend to all FTP usage, whether it was hosted at a service site or not—at least not immediately [21, 102].

That is, even if accounting was not necessary at research sites to start, it was acknowledged that, as more users joined the ARPANET and wanted access to limited resources, it would eventually be necessary to account and bill for research usage, as well. For example, as early as RFC 82, Douglas Engelbart was recorded as saying that SRI will eventually have to bill as more users come online: “A system will exist in Spring 1971, to allow an agent to insert into a catalog. The dialogue that goes on will determine which way the data base grows. We are pretty sure that eventually SRI will have to charge because of many potential users not at primary sites seeking limit[sic] resources. ... Each site is registered. Any person who gets in on a site's account has its access. We won't worry about accounting until saturation occurs. We would like to encourage use of the agent system to create and use a survey of resources at each site,” [40, p. 7] which included SRI's research theorem prover tools [112, p. 548]. In fact, by 1987, the interagency research Internet proposal made it clear that accounting would absolutely be necessary for research [74, 75].

<sup>46</sup>This contention is explicitly addressed as a difference in “orientation” in RFC 231: “In the network at large, with our research orientation, personnel tend to have a different approach to computing than that required by a service bureau.” Service Centers were believed to be subject to “market-oriented requirements” to rate-limit use, while Research Centers were free from such forces [53, p. 4].

## B THE DECLINE OF THE IDEAL OF RESOURCE SHARING

Janet Abbate argues that, as the network began to spread to more ARPA-contractor sites, the “demand for remote resources fell,” and that “many sites rich in computing resources seemed to be looking for users” [2, p. 104]. Abbate credits this “decline of the ideal of resource sharing” (where this ideal was the vision of the ARPANET articulated in Roberts [111]) to severe usability issues in the ARPANET. Throughout Chapter 3, she discusses the practical difficulties of using the network—even with an action as simple as finding a particular resource, as the network lacked appropriate search tools [2, p. 86].<sup>47</sup>

Even if users got past the initial hurdle of finding a resource, there remained additional steps in order to access it. Abbate acknowledges that this in part had to do with accounting, but discusses the issue as one of usability: The user would have to contact an administrator to set up an account on the remote host in order to access it, which required finding and contacting the appropriate person at the remote site; if the remote site wanted to charge for usage, the user then also usually had to initiate a purchase order at their local institution. Only then could they access the resource, at which point it often remained a challenge to figure out how the resource was supposed to be used [2, pp. 87–88]. This lack of usability, which reflected both technical and administrative issues, ideally should not have been a relevant concern for individual users. However, it was a particularly challenging obstacle for novice users, and thus became a bottleneck in the network’s ability to grow and reach saturation. Abbate therefore reasons that the ARPANET became a technology seeking an appropriate application and an interested user base, as it fell short of its goal to facilitate resource sharing. The ARPANET needed to experience a fundamental shift in “identity and purpose” if it was to be a useful technology [2, p. 109].<sup>48</sup>

Abbate writes that this shift ultimately occurred when the ARPANET found such a “smash hit” application in email [2, p. 106]. In contrast to the technical and administrative usability issues that plagued resource sharing, email was very simple to use; it connected users at remote sites, but was an application that users could access locally. Similarly, local area networks (LANs) also became a popular, unexpected use of the ARPANET at this time, as LANs did not have the same usability issues as more long-range, distributed resource sharing. By 1975, with sites like USC and SRI using the ARPANET as for LANs, 30% of traffic on the network was intra-node, as opposed to inter-node [2, p. 94]. In short, Abbate argues that, while resource sharing struggled to find users, these two ARPANET uses were successes that validated the utility of the network.

Unpacking Abbate’s focus on usability, we argue that it is possible to understand the “decline of the ideal of resource sharing” and its

attendant challenges to the would-be user in terms of the lack of appropriate, fleshed-out accounting mechanisms. While usability was certainly a relevant factor concerning the (lack of) ease of adoption of the ARPANET for resource sharing, it should not be conflated with a lack of desire to resource share altogether. In fact, the debate over “free” file transfer (Section 2.1) and the attempt to classify different sites as research or service centers is evidence that ARPANET users wanted to resource share (with services like UCSB’s SMFS and UCSD’s FTP node being sufficiently utilized to require accounting to recoup costs, see Appendix A). Instead, as we discussed in Sections 2 and 3, accounting was (and remains) an extremely challenging technical and administrative problem, which the early ARPANET architects did not have an appetite to address. As a result, accounting did not become a first-order feature of the network, and was instead a messy patchwork of non-interoperable, ill-defined systems that made recovering the costs of resource sharing intractable.

From this perspective, it is possible to recast Abbate’s examples of unexpected successes of usability—the “smash hit” of email and the proliferation of LANs—as successes owing to their independence from distributing accounting. Email leveraged the distributed network, but it did not have the same distributed accounting problems as resource sharing. Aside from using gateway nodes for routing email to its final destination, email (at least its 1970s iteration) was a local application, using local CPU and local storage. If accounting needed to be done, it could be handled locally at the site level, using local administrative procedures (if there were any), such as those used for time-sharing [37, p. 4] [126, concerning UCLA’s internal billing for email]. In other words, it is possible to view email as a success not just because of ease of use, but also because it was possible to treat email as “free” in a way similar to the initial assumptions about the negligible costs of “free” file transfer, in which local billing could be used for local usage or billing could be punted back to ARPA. Of course, just like “free” file transfer, email was not literally free. Email eventually became quite costly, particularly when sending spam became a growing practice.<sup>49</sup> Spam put a strain on receivers’ local resource usage, which meant that it, too, became an accounting problem [51].<sup>50</sup> Similarly, LANs were intra-node, and also represented a problem of local accounting, as opposed to distributed accounting.

## C COMPUTATIONAL METHODS

In this appendix, we document our procedure for identifying RFCs related to our project. The accompanying code can be found at <https://github.com/pasta41/rfc-scraper>.

**Our rfc-scraper tool.** We developed a tool to scrape the RFC Editor, which pulls down the .txt version of each RFC and associated metadata. We wrote a separate script to filter and map the above data to identify RFCs. The filtering capability is simple:

<sup>47</sup>See also RFC 531, concerning ARPANET usability problems, the creation of a resource notebook to improve these problems, and ultimately the additional gaps the resource notebook highlighted in relation to documentation reliability issues [97].

<sup>48</sup>For more on evaluating the “friendliness” of the network, see RFC 369 [101]; see RFC 451 [95] and RFC 666 [96] for defining the unified user level protocol (UULP)—a proposed solution to ARPANET usability issues, which suggested a command language for “user convenience,” “resource sharing,” “economy of mechanism,” “front-ending...onto existing commands,” “accounting and authorization,” and “process-process functions” [96, pp. 1–2]. In other words, UULP was an attempt to come up with a single protocol that would help with network usability functions.

<sup>49</sup>For a detailed treatment on the history of Internet spam, see Brunton [22].

<sup>50</sup>RFC 2635, “Don’t Spew: A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam\*),” discusses the costs of spam via a comparison with physical mail. It notes that it is easier to send email, so the scale of junk email is much greater. The costs are also quite different: It costs the sender very little to send spam; “the recipient bears the majority of the cost” [51, pp. 3–4]. The RFC thus calls spam “unethical behavior” [51, p. 3], and goes so far as to compare it to a seizure of private property, since it eats up local resources—a “theft of service” [51, p. 4]

The script takes as an argument a search term to grep for (e.g., “account”, treated as a prefix and case insensitive); if there is a match in an RFC, the RFC ID is mapped to its associated metadata (author, title, working groups, etc.) and the results are saved to a .csv file. Separately, each matched instance in each RFC is printed to a .html file (with 5 leading and trailing lines of surrounding context, and each matched word highlighted in color and emphasized).

For more detailed documentation on this code, please refer to <https://github.com/pasta41/rfc-scraper>. The README in the repository has the most up-to-date information related to the required python dependencies (including bash scripting utilities called within python for extra efficiency), and the ANSI HTML adapter package used to develop the color-coded .html search results files. We intend to update this tool with more sophisticated search functionality so that others can use it in the future for additional RFC-related research (e.g., filtering by time range, working group, author; other data manipulation functions).

**Search terms and verification process.** We ran the rfc-scraper tool on several search terms related to our project purpose:

- “account”, which matches account\* and served as our superset search for all accounting terms

- “accounting”, which matches a subset account\*
- “accountable”, which matches a subset of account\*
- “accountability”, which matches a subset of account\*
- “time-shar”, which matches time-shar\* and therefore includes time-share, time-sharing, and similar terms
- “survivabl”, which matches survivable and survivability

We then manually read the .html files, which include leading and trailing text surrounding matches, to determine which RFCs we should read in more detail for our project. Of the 9085 RFCs published at the time of running the tool, this process enabled us to identify a subset of 136 RFCs to read in full. 19 of the 136 were false positives (i.e., they were not ultimately relevant). During detailed reading of the 136 RFCs, we further identified 12 RFCs that our search terms missed, which were relevant to our project. We found these RFCs through references in both the RFCs our tool selected and in the secondary literature we cite. We include our RFC tracker code book in the rfc-scraper repository. As an aside, through this process we learned that our search tool bears a coincidental resemblance to (though is much simpler than) the search-and-filter tools built to identify time datatype issues related to the Y2K in Internet protocols [57].