INTERNET POLICY REVIEW
Journal on internet regulation

Volume 5 | Issue 3

Instability and internet design

Sandra Braman
Department of Communication, Texas A&M University, United States, braman@tamu.edu

Published on 30 Sep 2016 | DOI: 10.14763/2016.3.429

Abstract: Instability - unpredictable but constant change in one's environment and the means with which one deals with it - has replaced convergence as the focal problem for telecommunications policy in general and internet policy in particular. Those who designed what we now call the internet during the first decade of the effort (1969-1979), who in essence served simultaneously as its policy-makers, developed techniques for coping with instability of value for network designers today and for those involved with any kind of large-scale sociotechnical infrastructure. Analysis of the technical document series that was medium for and record of that design process reveals coping techniques that began with defining the problem and went on to include conceptual labour, social practices, and technical approaches.

Keywords: Internet design, Sociotechnical decision-making, Resilience

Article information

Received: 25 Apr 2016 **Reviewed:** 17 Jun 2016 **Published:** 30 Sep 2016
Licence: Creative Commons Attribution 3.0 Germany
Funding: This material is based upon work supported by the US National Science Foundation under Grant No. 0823265. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author.
Competing interests: The author has declared that no competing interests exist that have influenced the text.

URL: http://policyreview.info/articles/analysis/instability-and-internet-design

Citation: Braman, S. (2016). Instability and internet design. *Internet Policy Review*, *5*(3). DOI: 10.14763/2016.3.429

This paper is part of Doing internet governance, a special issue of Internet Policy Review *guest-edited by Dmitry Epstein, Christian Katzenbach, and Francesca Musiani.*

Where convergence was the orienting issue for communication policy-makers in the second half of the 20th century, in the 21st it is resilience in the face of instability, whether from human or natural causes, that has come to the fore (see, e.g., Manzano, et al., 2013; Smith, 2014; Sterbenz et al., 2014; Tipper, 2014). Defining instability here as unpredictable but constant change in one's environment and in the means with which one interacts with it, instability-based problems underlie many of today's internet policy issues.

Among those who must be considered policy-makers for the internet are the computer scientists and electrical engineers responsible for the technical decision-making that brings the network into being and sustains it through constant transformations, expansions, and ever-increasing

complexity. The instabilities faced by early internet designers - those who worked on the problem from when it was first funded by DARPA in 1969 through the close of 1979 - were myriad in number and form. They arose on both sides of this sociotechnical infrastructure, appearing technically in software and hardware, and socially in interpersonal and institutional relations. This was a difficult working situation not only because instabilities were pervasive and unpredictable, but also because the sources of instability and their manifestations were themselves constantly refreshed, unrelenting.

It is these policy-makers who are the focus of this article, which asks: *how did technical decision-makers for what we now call the internet carry on their work in the face of unpredictable but pervasive and ongoing instability in what they were building and what they had to build it with?* It addresses this question by inductively mining the technical document series that served as both medium for internet design and a record of that history (Abbate, 1999).

The analysis is based on a reading of the almost 750 documents in the Internet Requests for Comments (RFCs, www.ietf.org/rfc.html) series that were published during the first decade of the design process (1969-1979). Coping techniques developed during this early period remain important after almost 50 years at the time of writing because such a wide range of types and sources of instability appeared during that period, and because the decisions, practices, and norms of that decade were path determinative for internet decision-making going forward. The document series records a conversation among those responsible for the technical side of the sociotechnical network, but during the first 20 years of the process in particular the discussion included a great deal of attention to social, economic, cultural, legal, and governance issues. Thinking about the design process through the lens of what it took to conceptualise the network and bring it into being under conditions of such instability increases yet again one's appreciation of what was accomplished.

The focus here is on those types of instability that are particularly important for large-scale sociotechnical infrastructure rather than those that appear with any type of endeavour. In bridge-building, for example, it is not likely that the technologies and materials being used will change constantly over the course of the project, but this is a common problem for those working with large-scale sociotechnical infrastructure. Such instability remains a central problem for internet designers today; a draft book on possible future network architectures by David Clark (2016), who has been involved with internet design since the mid-1970s, devotes significant attention to problems of this kind. Other ubiquitous and inevitable decision-making problems, such as value differences among those involved and frustration over time lags between steps of development and implementation processes, were also experienced by internet designers but are beyond the scope of this piece.

Mechanisms developed to cope with instabilities are rarely discussed in scholarly literature. The closest work, although it addresses a qualitatively different type of problem, comes from those in science, technology, and society studies (STS) who examine ways in which scientists transform various types of messiness in the laboratory into the clean details reported as scientific findings (importantly, in the work by Latour & Woolgar [1986], and Star [1989]), and into public representation of those efforts (Bowker, 1994). The research agenda going forward should look in addition at what can be learned from psychology and anthropology.

Internet designer efforts to cope with instabilities began with determining just what constituted stability - in essence, designing the problem itself in the sense of learning to perceive it and frame it in ways that helped solve it. They went on to include figuring out the details (conceptual

labour), getting along (social practices), and making it work (technical approaches).

DEFINING THE PROBLEM AS A TECHNIQUE FOR ITS CURE

Discerning the parameters of instability is an epistemological problem requiring those involved in addressing it to figure out just how to know when the system is stable enough for normal operations to proceed. Internet designers have, from the beginning, required a consensus on the concepts fundamental to such problems. [1] The techniques used to achieve a consensus regarding just what distinguished stability from instability of particular importance included drawing the line between stability and instability, distinguishing among different types of change for differential treatment within protocol (standard) setting processes, and resolving tensions between the global and the local, the universal and the specific.

Although the subject of what internet designers knew empirically about how the network was actually functioning is beyond the scope of this article, it is worth noting that comprehending and responding to the sources of instability was made even more problematic by a lack of information:

> [E]ven those of us presumably engaged in 'computer science' have not found it necessary to confirm our hypotheses about network operation by experiment an [sic] to improve our theories on the basis of evidence (RFC 550, 1973, p. 2).

Indeed, design force was explicitly preferred over empirical knowledge:

> If there are problems using this approach, please don't 'code around' the problem or treat your [network interconnection node] as a 'black box' and exxtrapolate its characteristics from a series of experiments. Instead, send your comments and problems to . . . BBN, and we will fix the . . . system" (RFC 209, 1971, p. 1).

STABILITY VS INSTABILITY

For analytical and pragmatic purposes, instability as understood here - unpredictable but constant change in one's environment, including the ways in which one interacts with and is affected by it whether directly or indirectly - can usefully be distinguished from other concepts commonly used in discussions of the internet. Instability is *not* the same thing as ignorance (lack of knowledge about something specific), uncertainty (lack of knowledge about the outcome of processes subject to contingency or opacity, or otherwise unknowable), or ambiguity (lack of clarity regarding either empirical realities or intentions). Indeed, instability differs from all of these other terms in an important way: ignorance, uncertainty, and ambiguity are about what is known by those doing the design work, *the maker*. Instability, on the other hand, is about unpredictable mutability in *that which is being made and the tools and materials available to make it*.

For designers of what we now call the internet, goals during the first decade of the design process *re* network stability were humble. They sought protocols that could last for at least a

couple of years, fearing that if this level of stability could not be achieved it would be hard to convince others to join in the work (RFC 164, 1971). It was considered a real improvement when the network crashed only every day or two (RFC 153, 1971), a rate neither widely nor commonly experienced. According to RFC 369 (1972), no one who responded to a survey had reported a mean-time-between-failure of more than two hours and the average percentage of time with trouble free operation was 35%.

Network designers defined stability operationally, not theoretically. The network is unstable when it isn't functional or when one can't count on it to be functional in future barring extraordinary events. Concepts long used in the security domain to think about those forces that can make a system unstable can be helpful in thinking about instabilities and the internet design process. Those involved with national security distinguish between system *sensitivity* and *vulnerability*. Sensitivity involves system perturbations that may be annoying and perhaps costly but are survivable; hacking into the Democratic National Committee information systems (Sanger & Schmitt, 2016) was a perturbation, but hasn't brought the country down (as of the time of writing). Vulnerability entails those disturbances to a system that undermine its survival altogether; if malware such as Conficker (Kirk, 2015) were used to shut down the entire electrical network of the United States, it would generate a serious crisis for the country. Vulnerability has long been important to the history of telecommunications networks, being key to stimulating the growth of a non-British international telecommunications network early in the 20th century (Blanchard, 1986; Headrick, 1990); the push for greater European computational capacity and intelligent networks in the 1980s (Nora Minc, 1980; Tengelin, 1981); and in discussions of arms control (Braman, 1991) and cybersecurity (Braman, 2014). Factors that cause network instability are those that present possible vulnerabilities.

TECHNICAL CHANGE

The phenomenon of fundamental and persistent change was explicitly discussed by those involved in the early years of designing what we refer to today as the internet. The distinction between incremental and radical change was of particular importance because of the standard-setting context.

It can be difficult for those of us who have been online for decades and/or who were born "digital natives" to appreciate the extent of the intellectual and group decision-making efforts required to achieve agreement upon the most fundamental building blocks of the internet. Even the definition of a byte was once the subject of an RFC and there was concern that noncompliance with the definition by one user would threaten the stability of the entire network (RFC 176, 1971).

For the early internet, *everything* was subject to change, all the time: operating systems, distinctions among network layers, programming languages, software, hardware, network capacity, users, user practices, and on. Everyone was urged to take into account the possibility that even command codes and distinctions among network layers could be redefined (RFC 292, 1972). Those who were wise and/or experienced expected operational failures when ideas were first tried under actual network conditions (RFC 72, 1970). Operating by consensus was highly valued, but it was also recognised that a consensus once achieved might still have to be thrown out in response to experience or the introduction of new ideas or protocols. Instituting agreed-upon changes was itself a source of difficulty because use of the network was constant and maintenance breaks would therefore be experienced as instability (RFC 381, 1972), a condition ultimately mitigated but not solved by regular scheduling of shutdowns.

Looking back from 2016, early perceptions of the relative complexity and scale of the problem are poignant:

> Software changes at either site can cause difficulties since the programs are written assuming that things won't change. Anyone who has ever had a program that works knows what system changes or intermittent glitches can do to foul things up. With two systems and a Network things are at least four times as difficult. (RFC 525, 1973, p. 5)

RFC 525 (1973) also repeats the point that changes by a user at a local site can cause difficulties for the network as a whole. RFC 528 (1973) makes the opposite point: changes in the network could impede or make it impossible for processes at local user sites to continue operating as they had (RFC 559, 1973; RFC 647, 1974); one author complained about the possibility of a situation in which servers behave erratically when they suddenly find their partner speaking a new language (RFC 722, 1976). Interdependencies among the technologies and systems involved in internet design were complex, often requiring delay in implementation of seemingly minor changes because each would require so many concomitant alterations of the protocols with which they interact that all are better left until they can be a part of a major overhaul package (RFC 103, 1971).

INCREMENTAL VS RADICAL

A particularly difficult problem during the early years of the internet design process was determining when what was being proposed should be considered something new (a radical change) or a modification (incremental change) (RFC 435, 1973). The difference matters because systems respond differently to the two. Both types of change were rife during the internet design process, manifested in explicit discussions about whether something being discussed in an RFC should be treated as an official change or a modification if ultimately agreed upon and put into practice. As the question was put in RFC 72 (1970), what constitutes official change to a protocol, given that ideas about protocols go through many modifications before reaching solutions acceptable to all?

Translation of value differences into an objective framework was one means used to try to avoid tensions over whether something involved an incremental or radical change. Describing the design of algorithms as a "touchy" subject, a “Gordian knot”, for example, one author proposing a graphics protocol notes, “There are five or ten different criteria for a ‘best’ algorithm, each criterion different in emphasis” (RFC 292, 1972, p. 4). The coping technique used in response to this problem in RFC 292 was to simply order the commands by level and number them. If several commands at the same level came into conflict, some attempt would be made to encode variations of meanings in terms of bit configurations.

MACRO VS MICRO

There are two dimensions along which distinctions between macro-level and micro-level approaches were important in network design: the global vs the local, and general function vs specific function. These two can be aligned with each other, as with the local and specific treatment of a screen pixel trigger in an early graphics protocol that was determined to be so particular to a given configuration of technologies that it should not be included in internet protocols (RFC 553, 1973). The two dimensions of globality and generality, however, need not operate in tandem. In one example, sufficient universality on the network side was ensured by

insisting that it could deal with all local variations encountered (e.g., RFC 184, 1971; RFC 529, 1973).

GLOBAL VS LOCAL

The tension between the universal and the local is fundamental to the nature of infrastructural systems. Indeed, as Star and Ruhleder (1996, p. 114) put it, infrastructure - however global - only comes into being in its local instances. The relationship between the two has long been important to telecommunications networks. In the 1880s, long-time AT&T president Theodore Vail and chief engineer J. J. Carty, who designed the company's monopoly-like and, for the era, ubiquitous network, encountered it:

> 'No one knows all the details now,' said Theodore Vail. 'Several days ago I was walking through a telephone exchange and I saw something new. I asked Mr. Carty to explain it. He is our chief engineer; but he did not understand it. We called the manager. He didn't know, and called his assistant. He didn't know, and called the local engineer, who was able to tell us what it was. (Casson, 1910, p. 167)

Early internet designers phrased the problem this way: "Should a PROTOCOL such as TELNET provide the basis for extending a system to perform functions that go beyond the normal capacity of the local system" (RFC 139, 1971, p. 11). Discussion of ways in which a single entity might provide functions for everyone on the network that most other hosts would be unable to provide for themselves reads much like ruminations on a political system characterised by federalism (in the US) or subsidiarity (in Europe): ". . . to what extent should such extensions be thought of as Network-wide standards as opposed to purely local implementations" (*Ibid.*). The comparison with political thinking is not facile; a tension between geopolitical citizenship and what can be called "network citizenship" runs throughout the RFCs (Braman, 2013).

Drawing, or finding, the line between the universal and the local could be problematic. Decisions that incorporated that line included ensuring that special-purpose technology- or user-specific details could be sent over the network (RFC 184, 1971), treating transfer of incoming mail to a user's alternate mailbox as a feature rather than a protocol (RFC 539, 1973), and setting defaults in the universal position so that they serve as many users as possible (RFC 596, 1973). Interestingly, there was a consensus that users needed to be able to reconnect, but none on just where the reconnection capacity should be located (RFC 426, 1973).

GENERAL PURPOSE VS SPECIFIC PURPOSE

The industrial machines for which legal and policies were historically crafted were either single-purpose or general-purpose. As this affected network policy a century ago, antitrust (competition) law was applied to the all-private US telecommunications network because, it was argued, being general purpose - serving more than one function, carrying both data and voice - was legally problematic as unfair competition. The resulting Kingsbury Commitment separated the two functions into two separate companies and networks that could interconnect but not be the same (Horwitz, 1989).

The internet, though, was experienced as a fresh start in network design. When the distinction between general and special purpose machines came up in the RFCs, it was with pride about having transformed what had previously been the function of a special purpose process into one available for general purpose use:

> With such a backbone, many of the higher level protocols could be designed and implemented more quickly and less painfully -- conditions which would undoubtedly hasten their universal acceptance and availability" (RFC 435, 1973, p. 5).

It was a basic design criterion - what can be considered, in essence, a constitutional principle for network design - that the network should serve not only all kinds of uses and all kinds of users, but also be technologically democratic. The network, that is, needed to be designed in such a way that it served not only those with the most sophisticated equipment and the fastest networks, but also those with the most simple equipment and the slowest networks (Braman, 2011). [2]

With experience, internet designers came to appreciate that the more general purpose the technologies at one layer, the faster and easier it is to design and build higher level protocols upon them. Thus it was emphasised, for example, that TELNET needed to find all commands "interesting" and worthy of attention, whether or not they were of kinds or from sources previously known (RFC 529, 1973, p. 9). In turn, as higher level and more specialised protocols are built upon general purpose protocols, acceptance of (and commitment to) those protocols and to design of the network as general purpose are reinforced (RFC 435, 1973).

Standardisation was key. It was understood that a unified approach would be needed for data and file transfer protocols in order to meet existing and anticipated network needs (RFC 309, 1972). Designing for general purpose also introduced new criteria into decision-making. Programming languages and character sets were to be maximised for flexibility (RFC 435, 1973), for example, even though that meant including characters in ASCII set that were not needed by the English language users who then dominated the design process (RFC 318, 1972).

FIGURING OUT THE DETAILS

The importance of the conceptual labour involved in the internet design process cannot be overstated, beginning with the need to define a byte discussed above through the most ambitious visions of globally distributed complex systems of diverse types serving a multitude of users and uses. Coping techniques in this category include the art of drawing distinctions itself as well as techniques for ambiguity reduction.

CONCEPTUAL DISTINCTIONS

Early recognition that not all information received was meant to be a message spurred efforts to distinguish between bit flows intended to as communications or information transfer, and those that were, instead, errors, spurious information, manifestations of hardware or software idiosyncrasies, or failures (RFC 46, 1970; RFC 48, 1970). Other distinctions had to be drawn between data and control information and among data pollution, synchronicity, and network "race" problems (when a process races, it won't stop) (RFC 82, 1970).

The need for distinctions could get very specific. A lack of buffer space, for example, presented a very different type of problem from malfunctioning user software (e.g., RFC 54, 1970; RFC 57, 1970). Distinctions were drawn in ways perhaps more diverse than expected: people experienced what we might call ghost communications when BBN, the consulting firm developing the technology used to link computers to the network during the early years, would test equipment

before delivery by sending messages received by others as from or about nodes they didn't think existed (RFC 305, 1972). And there were programmes that were perceived as having gone "berserk" (RFC 553, 1973).

Identifying commonalities that can then become the subject of standardisation is a critically important type of conceptual labour. The use of numerous *ad hoc* techniques for transmitting data and files across ARPANET was considered unworkable for the most common situations and designers knew it would become more so (RFC 310, 1972). Thus it was considered important to identify common elements across processes for standardisation. One very basic example of this was discussion of command and response as something that should be treated with a standard discipline across protocols despite a history of having previously been discussed only within each specific use or process context (RFC 707, 1975). The use of a single access point is another example of the effort to identify common functions across processes that could be standardised for all purposes (RFC 552, 1973).

Drawing conceptual distinctions is a necessary first step for many of the other coping techniques. It is required before the technical labour of unbundling processes or functions into separate functions for differential treatment, one of the technical tools discussed below, for example, and is evident in other techniques as well.

AMBIGUITY REDUCTION

Reducing ambiguity was highly valued as a means of coping with instability. One author even asserted this as a principle: "words which are so imprecise as to require quotation marks should never appear in protocol specifications" (RFC 513, 1973, p. 1). Quotation marks, of course, are used to identify a word as a neologism or a term being used with an idiosyncratic and/or novel meaning. This position resonates with the principle in US constitutional law that a law so vague two or more reasonable adults cannot agree on its meaning is unconstitutional and void.

Concerns about ambiguity often arose in the course of discussions about what human users need in contrast to what was needed for the non-human, or daemon users such as software, operating systems, and levels of the network, for which the network was also being designed (Braman, 2011). It was pointed out, for example, that the only time mail and file transfer protocols came into conflict was in naming conventions that needed to serve human as well as daemon users (RFC 221, 1971).

GETTING ALONG

The history of the internet design process as depicted in the internet RFCs provides evidence of the value of social capital, interpersonal relationships, and community in the face of instability. Valuing friendliness, communication, living with ambiguity, humour, and reflexivity about the design process were all social tools for coping with instability visible in the RFCs from the first decade. Collectively, we can refer to such tools as "getting along".

FRIENDLINESS

In addition to the normative as well as discursive emphasis on community consensus-building discussed elsewhere (Braman, 2011), the concept of friendliness was used explicitly. Naming sites in ways that made mnemonic sense to humans was deemed usefully user-friendly, allowing humans to identify the sources of incoming messages (RFC 237, 1971). Friendliness was a criterion used to evaluate host sites, both by network administrators concerned also about

reliability and response time (RFC 369, 1972) and by potential users who might have been discouraged by a network environment that seemed alien (RFC 707, 1975). Interpersonal relations - rapport among members of the community (RFC 33, 1970) - were appreciated as a coping technique. The effects of one's actions on others were to be considered: "A system should not try to simulate a facility if the simulation has side effects" (RFC 520, 1973, p. 3).

The sociotechnical nature of the effort, interestingly, shines through even when discussing interpersonal relations:

> The resulting mixture of ideas, discussions, disagreements, and resolutions has been highly refreshing and beneficial to all involved, and we regard the human interaction as a valuable by-product of the main effect. (RFC 33, 1970, p. 3)

At the interface between the network and local sites, internet designers learned through experience about the fundamental importance of the social side of a sociotechnical system. After discussing how network outsiders inevitably become insiders in the course of getting their systems online, one author noted,

> [I]f personnel from the several Host[s] [sic] are barred from active participation in attaching to the network there will be natural (and understandable) grounds for resentment of the intrusion the network will appear to be; systems programmers also have territorial emotions, it may safely be assumed. (RFC 675, 1974)

The quality of relations between network designers and those at local sites mattered because if the network were perceived as an intruder, compliance with protocols was less likely (RFC 684, 1975).

COMMUNICATION

Constant communication was another technique used in the attempt to minimise sources of instability. Rules were set for documentation genres and schedules (RFC 231, 1971). Using genre categories provided a means of announcing to users how relatively fixed, or not, a particular design decision or proposal was and when actual changes to protocols might be expected - both useful as means of dealing with instability. Today, the Internet Engineering Task Force (IETF), which hosts the RFCs online, still uses genre distinctions among such categories as *Internet Standard*, *Draft Standard*, and *Proposed Standard*, as well as genres for *Best Practices* and others that include those that are *Informational*, *Historic*, or *Experimental*. [3]

Users were admonished to keep the RFCs and other documentation together because the RFCs would come faster and more regularly than would user guides. Still, it was highlighted, it was impossible for users to keep up with changes in the technologies: "It is *almost* inevitable that the TUG [Tip user Guide] revisions follow actual system changes" (RFC 386, 1972, p. 1, emphasis added). Simplicity and clarity in communication were valued; one author's advice was to write as if explaining something both to a secretary and to a corporation president - that is, to both the naiver and to the sophisticated (RFC 569, 1973).

LIVING WITH AMBIGUITY

Although eager to reduce ambiguity wherever possible, early network designers also understood

that some amount of ambiguity due to error and other factors was inevitable (RFC 203, 1971). In those instances, the goal was to learn to distinguish among causal factors, and to develop responses to each that at least satisficed even if that meant simply ignoring errors (RFC 746, 1973).

HUMOUR

Humour is a technique used to cope with instability, as well as with ignorance, uncertainty, and ambiguity, in many environments. Within the internet design process, it served these functions while simultaneously supporting the development of a real sense of community. In RFC 468 (1973), for example, there is an amusing description of just how long it took to define something during the course of internet design. There was an ongoing tradition of humorous RFCs (beware of any published on 1 April, April Fool's Day) (Limoncelli & Salus, 2007).

REFLEXIVITY ABOUT THE DESIGN PROCESS

The final social technique for adapting to instability evident early on was sustaining communal reflexivity about the nature of the design process itself. RFC 451 (1973) highlighted the importance of regularly questioning whether or not things should continue being done as they were being done. It was hoped that practices developed within the network design community would diffuse into those of programmers at the various sites linking into the network (RFC 684, 1975).

MAKING IT WORK

Many of the coping techniques described above are social. Some are technical, coming into play as the design principles that are, in essence, policy for the internet design process (Braman, 2011). A final set of techniques is also technical, coming into use as specific design decisions intended to increase adaptive capacity by working with characteristics of the technologies themselves. Approaches to solving specific technical problems in the face of instability included designing in adaptive capacity, tight links between genre and machinic specifications, delay, and the reverse of delay, making something happen.

ADAPTIVE CAPACITY

General purpose machines begin by being inherently flexible enough to adapt to many situations, but it is possible to go further in enhancing adaptive capacity. The general goal of such features was captured in RFC 524 (1973):

> The picture being painted for the reader is one in which processes cooperate in various ways to flexibly move and manage Network mail. The author claims . . . that the picture will in future get yet more complicated, but that the proposal specified here can be conveniently enlarged to handle that picture too (p. 3).

The problem of adaptation came up initially with the question of what to do with software that had been designed before its possible use in a network environment had been considered. RFC 80 (1970) argued that resolving this incompatibility should get as much attention as developing new hardware by those seeking to expand the research capacity of network users. Another such mechanism was the decision to require the network to adapt to variability in input/output mechanisms rather than requiring programmes to conform with the network (RFC 138, 1971).

Taking this position did not preclude establishing standards for software programmes that interact with the network and making clear that using those standards is desirable (RFC 166, 1971).

Beginning with recuperation of lost messages, and irrespective of the source of error, redundancy has long been a technique for coping with network instability issues. When satellites became available for use in international communications, for example, the US Federal Communications Commission (FCC) required every network provider to continue to invest as much in underseas cables as it invested in satellites (Horwitz, 1989). The early RFCs discuss redundancy in areas as disparate as message transmission (RFC 65, 1970) and the siting of the network directory (RFC 625, 1974). Redundancy in databases was understood as an access issue (RFC 677, 1975).

There are other ways adaptation was technically designed into the early network as a means of coping with instability. RFC 435 (1973) looks at how to determine whether or not a server has an echoing mode during a period in which many hosts could either echo or not echo, but did not have the option to go either way. Requiring fixed socket offsets until a suitable network-wide solution could be found to the problem of identity control at connection points between computers and the ARPANET (RFC 189, 1971) is another example.

There were situations for which reliance on *ad hoc* problem solving was the preferred approach (RFC 247, 1971). At their best, *ad hoc* environments could be used for experimentation, as was done with the mail facility (RFC 724, 1977). A "level 0" protocol was a more formal attempt to define an area in which experimentation could take place; successes there could ultimately be embedded in later protocols for the network itself (RFC 549, 1973). Maintaining a “wild west” zone for experimentation as a policy tool is familiar to those who know the history of radio regulation in the United States, where amateur (“ham”) radio operators have long been given spectrum space at the margins of what was usable. Regulators understood that these typically idiosyncratic individuals were persistent and imaginative inventors interested in pressing the limits of what they could do - and that their tinkering had yielded technical solutions that then made it possible to open up those wavelengths to commercial use over and over again.

Reliance on probabilities was another long familiar technique for situations involving instability as well as uncertainty. RFC 60 (1970) describes a technique apparently used by many larger facilities connected to the network to gain flexibility managing traffic and processing loads. They would falsely report their buffer space, relying on the probability that they would not get into logistical trouble doing so and assuming that statistics would keep them out of trouble should any difficulties occur. The use of fake errors was recommended as a means of freeing up buffer space, a measure considered a last resort but powerful enough to control any emergency.

GENRE SPECIFICATIONS

Working with the genre requirements described above offered another set of opportunities for coping with instability. The RFC process was begun as an intentionally informal conversation but, over time, became much more formal regarding gatekeeping, genre classification, and genre requirements specific to stages of decision-making. Concomitantly, the tone and writing style of the documents became more formal as well. It is because of these two changes to the RFC publishing process that discussions of social issues within the design conversation declined so significantly after the first couple of decades.

For any RFC dealing with a protocol, what had not been articulated simply didn't exist (RFC 569, 1973). This put a lot of weight on the needs both to provide documentation - and to keep a

technology operating in exactly the manner described in that documentation (RFC 209, 1971). This was not a naive position; in discussion of the interface between the network and host computers, it was admitted that specifications were neither complete nor correct, but the advice was to hold the vendor responsible for technical characteristics as described. In a related vein, RFC authors were advised not to describe something still under experimentation in such a manner that others will believe the technology is fixed (RFC 549, 1973)

This position does, however, create a possible golem problem, in reference to the medieval story about a human-type figure created out of clay to do work for humans, always resulting in disaster because instructions were never complete or specific enough. From this perspective, the expectation of an unambiguous, completely specified mapping between commands and responses may be a desirable ideal (RFC 722, 1976), but could not realistically be achieved.

PUTTING THINGS OFF

The network design process was, by definition, ongoing, but this fundamental fact itself created instabilities: "Thus each new suggestion for change could conceivably retard program development in terms of months" (RFC 72, 1970, p. 2).

Because interdependencies among protocols and the complexity of individual protocols made it difficult to accomplish what were otherwise incremental changes without also requiring so much perturbation of protocols that wholesale revision would be needed (RFC 167, 1971), it was often necessary to postpone improvements that solved current problems until an overhaul took place. This happened with accounting and access controls (*Ibid.*) and basic bit stream and byte stream decisions for a basic protocol (RFC 176, 1971). As the network matured, it became easier to deal with many of these issues (RFC 501, 1973).

There were a number of occasions when the approach to a problem was to start by distinguishing steps of a process that had previously been treated as a single step - unbundling types of information processing, that is, in the way that vendors or regulators sometimes choose or are required to do with service or product bundles. It was realised, for example, that treating "hide your input" and “no echo” as two separate matters usefully permitted differential treatment of each (RFC 435, 1973). Similarly, the official FTP process was broken down into separate commands for data transfer and for file transfer, with the option of further distinguishing subsets within each (RFC 486, 1973). If we think of unbundling the steps of a single process as one way of making conceptual distinctions that provide support for continuing to work in the face of instability as a vertical matter, we might call it horizontal unbundling when distinctions among types of processing involved in a single step are drawn. By 1973 (RFC 520, 1973) it had already been found that having three digits for codes to distinguish among types of replies was insufficient, so a move to five digits was proposed as a short-term fix.

DEMONSTRATION

There were some instances in which designers foresaw a potential problem but could not convince others in the community that it was likely and serious. One technique used in such instances was to make actualize the potential - to make it happen in order to demonstrate the problem in such a way that the community would so appreciate the nature and seriousness of the concern that they would turn to addressing the issue. In 1970, for example, one designer - acting on an insight he had had about a potential type of problem in 1967 - deliberately flooded the network in order to convince his colleagues of the lock-up that results when that happens because of errors in message flow (RFC 635, 1974). This technique is familiar to those who know the literature on the diffusion of innovations. In Rogers’ (2003) synthesis of what has been

learned from thousands of studies of the diffusion of many different types of technologies in a wide range of cultural settings around the world, trialability and observability are among the five factors that significantly affect the willingness of individuals and groups to take up the use of new technologies and practices.

CONCLUSIONS

In today's digital, social, and natural worlds, instability is a concern of increasing importance to all of us as individuals and as communities. Those responsible for designing, building, and operating the infrastructures upon which all else depends - during times of instability just as during times of calm and slow change - confront particular difficulties of enormous importance that may be technical in nature but are of social, political, economic, and cultural importance as well. Insights drawn from discussions about the Internet design process in the Requests for Comments (RFCs) technical document series during the first decade of work on what we now call the internet (1969-1979) regarding how they coped with instability provides insights into coping techniques of potential use in the design, building, and operation of any large-scale sociotechnical infrastructure. The toolkit developed by network designers engaged with all facets of what makes a particular system sociotechnical rather than "just" social or technical: negotiating the nature of the issue, undertaking the conceptual labour involved in figuring out the details, learning how to get along with all of those involved, and incorporating adaptive techniques into the infrastructure itself.

Many of those involved with "ethics in engineering," including the relatively recent subset of that community that refers to itself as studying “values in design,” often start from theory and try to induce new behaviours among computer scientists and engineers in the course of design practice with the hope of stimulating innovations in content, design, or architecture. Here, instead, the approach has been to learn from the participants in the design process themselves, learning from these highly successful technical decision-makers - *de facto* policy-makers for the internet - about how to cope with instabilities in a manner that allowed productive work to go forward.

REFERENCES

Abbate, J. (1999). *Inventing the Internet*. Cambridge, MA: MIT Press.

Below, A. (2012). The genre of guides as a means of structuring technology and community. Unpublished MA Thesis, University of Wisconsin-Milwaukee.

Blanchard, M. A. (1986). *Exporting the First Amendment*. New York: Longman.

Bowker, G. C. (1994). *Science on the run: Information management and industrial geophysics at Schlumberger, 1920-1940*. Cambridge, MA: MIT Press.

Braman, S. (2014). Cyber security ethics at the boundaries: System maintenance and the *Tallinn Manual*. In L. Glorioso & A.-M. Osula (Eds.), *Proceedings: 1st Workshop on Ethics of Cyber Conflict*, pp. 49-58. Tallinn, Estonia: NATO Cooperative Cyber Defence Centre of Excellence.

Braman, S. (2013). The geopolitical vs. the network political: Governance in early Internet design, *International Journal of Media & Cultural Politics*, *9*(3), 277-296.

Braman, S. (2011) The framing years: Policy fundamentals in the Internet design process, 1969-1979, *The Information Society*, *27*(5), 295-310.

Braman, S. (1990). Information policy and security. Presented to the 2nd Europe Speaks to Europe Conference, Moscow, USSR.

Casson, H. N. (1910). *The history of the telephone*. Chicago, IL: A. C. McClurg & Co.

Clark, D. D. (2016). *Designs for an internet*. Available at http://groups.csail.mit.edu/ana/People/DDC/archbook.

Headrick, D. R. (1990). *The invisible weapon: Telecommunications and international relations, 1851-1945*. New York/Oxford: Oxford University Press.

Horwitz, R. B. (1986). *The irony of regulatory reform: The deregulation of American telecommunications*. New York/Oxford: Oxford University Press.

Kirk, J. (2015, Aug. 3). Remember Conficker? It's still around, *Computerworld*, http://www.computerworld.com/article/2956312/malware-vulnerabilities/remember-conficker-its-still-around.html, accessed Sept. 6, 2016.

Latour, B. & Woolgar, S. (2013). *Laboratory life: The construction of scientific facts*, 2d ed. Princeton, NJ: Princeton University Press.

Limoncelli, T. A. & Salus, P. H. (Eds.) (2007). Book on humor in the RFCs. Peer-to-Peer Communications.

Manzano, M., Calle, E., Torres-Padrosa, V., Segovia, J., & Harle, D. (2013). Endurance: A new robustness measure for complex networks under multiple failure scenarios, *Computer Networks*, *57*, 3641-3653.

Nora, S. & Minc, A. (1980). *The computerization of society?* Cambridge, MA: MIT Press.

Rogers, E. M. (2003) *Diffusion of Innovations*, 5th ed. New York: Free Press.

Sanger, D. E. & Schmitt, E. (2016, July 26). Spy agency consensus grows that Russia hacked D.N.C., *The New York Times*, http://www.nytimes.com/2016/07/27/us/politics/spy-agency-consensus-grows-that-russia-hacked-dnc.html, accessed Sept. 6, 2016.

Smith, P. (2014). Redundancy, diversity, and connectivity to achieve multilevel network resilience, survivability, and disruption tolerance, *Telecommunications Systems*, *56*, 17-31.

Star, S. L. (1989). *Regions of the mind: Brain research and the quest for scientific certainty*. Stanford, CA: Stanford University Press.

Star, S. L. & Ruhleder, K. (1996). Steps toward an ecology of infrastructure: Design and access for large information spaces, *Information Systems Research*, *7*(1), 111-134.

Sterbenz, J.P. G., Hutchison, D., Çetinkaya, E.K., Jabhar, A., Rohrer, J.P., Schöller, M., & Tipper, D. (2014). Resilient network design: Challenges and future directions, *Telecommunications Systems*, *56*, 5-16.

Tengelin, V. (1981). The vulnerability of the computerised society. In H. P. Gassmann (Ed.), *Information, computer and communication policies for the 80s*, pp. 205-213. Amsterdam, The Netherlands: North-Holland Publishing Co.

RFCS CITED

RFC 33, New Host-Host Protocol, S. D. Crocker, February 1970.

RFC 46, ARPA Network Protocol Notes, E. Meyer, April 1970.

RFC 48, Possible Protocol Plateau, J. Postel, S. D. Crocker, April 1970.

RFC 54, Official Protocol Proffering, S.D. Crocker, J. Postel,l J. Newkirk, M. Kraley, June 1970.

RFC 57, Thoughts and Reflections on NWG/RFC 54, M. Kraley, J. Newkirk, June 1970.

RFC 60, Simplified NCP Protocol, R. B. Kalin, July 1970.

RFC 65, Comments on Host/Host Protocol Document #1, D.C. Walden, August 1970.

RFC 72, Proposed Moratorium on Changes to Network Protocol, R. D. Bressler, September 1970.

RFC 80, Protocols and Data Formats, E. Harslem, J.. Heafner, December 1970.

RFC 82, Network Meeting Notes, E. Meyer, December 1970.

RFC 103, Implementation of Interrupt Keys, R. B. Kalin, February 1971.

RFC 138, Status Report on Proposed Data Reconfiguration Service, R.H> Anderson, V.G. Cerf, E. Harslem, J.F. Heafner, J. Madden, R.M. Metcalfe, A. Shoshani, J.E. White, D.C.M. Wood, April 1971.

RFC 139, Discussion of Telnet Protocol, T. C. O'Sullivan, May 1971.

RFC 153, SRI ARC-NIC Status, J.T. Melvin, R.W. Watson, May 1971.

RFC 164, Minutes of Network Working Group Meeting, 5/16 through 5/19/71, J. F. Heafner, May 1971.

RFC 166, Data Reconfiguration Service: An Implementation Specification, R.H. Anderson, V.G. Cerf, E. Harslem, J.F. Heafner, J. Madden, R.M. Metcalfe, A. Shoshani, J.E. White, D.C.M. Wood, May 1971.

RFC 167, Socket Conventions Reconsidered, A.K. Bhushan, R. M. Metcalfe, J. M. Winett, May 1971.

RFC 176, Comments on 'Byte Size for Connections', A.K. Bhushan, R. Kanodia, R. M. Metcalfe, J. Postel, June 1971.

RFC 184, Proposed Graphic Display Modes, K.C. Kelley, July 1971.

RFC 189, Interim NETRJS Specifications, R.T. Braden, July 1971.

RFC 203, Achieving Reliable Communication, R.B. Kalin, August 1971.

RFC 209, Host/IMP Interface Documentation, B. Cosell, August 1971.

RFC 221, Mail Box Protocol: Version 2, R. W. Watson, 1971.

RFC 231, Service center standards for remote usage: A user's view, J.F. Heafner, E. Harslem, September 1971.

RFC 237, NIC View of Standard Host Names, R.W. Watson, October 1971.

RFC 247, Proffered Set of Standard Host Names, P.M. Karp, October 1971.

RFC 292, Graphics Protocol: Level 0 Only, J. C. Michener, I.W. Cotton, K.C. Kelley, D.E. Liddle, E. Meyer, January 1972.

RFC 305, Unknown Host Numbers, R. Alter, February 1972.

RFC 309, Data and File Transfer Workshop Announcement, A. K. Bhushan, March 1972.

RFC 310, Another Look at Data and File Transfer Protocols, A> K. Bhushan, April 1972.

RFC 318, Telnet Protocols, J. Postel, April 1972.

RFC 369, Evaluation of ARPANET Services January-March, 1972, J.R. Pickens, July 1972.

RFC 381, Three Aids to Improved Network Operation, J.M. McQuillan, July 1972.

RFC 386, Letter to TIP Users-2, B. Cosell, D.C. Walden, August 1972.

RFC 426, Reconnection Protocol, R. Thomas, January 1973.

RFC 435, Telnet Issues, B. Cosell, D.C. Walden, January 1973.

RFC 451, Tentative Proposal for a Unified User Level Protocol, M. A. Padlipsky, February 1973.

RFC 468, FTP Data Compression, R.T. Braden, March 1973.

RFC 486, Data Transfer Revisited, R.D. Bressler, March 1973.

RFC 501, Un-muddling 'Free File Transfer', K.T. Pogran, May 1973.

RFC 513, Comments on the New Telnet Specifications, W. Hathaway, May 1973.

RFC 520, Memo to FTP Group: Proposal for File Access Protocol, J.D. Day, June 1973.

RFC 524, Proposed mail protocol, J.E. White, June 1973.

RFC 525, MIT-MATHLAB meets UCSB-OLS -- an example of resource sharing. W. Parrish, J.R. Pickens, June 1973.

RFC 528, Software checksumming in the IMP and network reliability, J.M. McQuillan, June 1973.

RFC 529, Note on Protocol Synch Sequences, A.M. McKenzie, R. Thomas, R.S. Tomlinson, K.T. Pogran, June 1973.

RFC 539, Thoughts on the Mail Protocol Proposed in RFC 524, D. Crocker, J. Postel, July 1973.

RFC 549, Minutes of Network Graphics Group Meeting, 15-17 July 1973, J.C. Michener, July 1973.

RFC 552, Single Access to Standard Protocols, A.D. Owen, July 1973.

RFC 553, Draft Design for a Text/Graphics Protocol, C.H. Irby, K. Victor, July 1973.

RFC 559, Comments on the New Telnet Protocol and its Implementation, A.K. Bushan, August 1973.

RFC 569, NETED: A Common Editor for the ARPA Network, M.A. Padlipsky, October 1973.

RFC 596, Second thoughts on Telnet Go-Ahead, E.A. Taft, December 1973.

RFC 625, On-line hostnames service, M.D. Kudlick, E.J. Feinler, March 1974.

RFC 635, Assessment of ARPANET protocols, V. Cerf, April 1974.

RFC 647, Proposed protocol for connecting host computers to ARPA-like networks via front end processors, M.A. Padlipsky, November 1974.

RFC 675, _____. 1974.

RFC 677, Maintenance of duplicate databases, P.R. Johnson, R. Thomas, January 1975.

RFC 684, Commentary on procedure calling as a network protocol, R. Schantz, April 1975.

RFC 707, High-level framework for network-based resource sharing, J.E. White, December 1975.

RFC 722, Thoughts on Interactions in Distributed Services, J. Haverty, September 1976.

RFC 724, Proposed official standard for the format of ARPA Network messages, D. Crocker, K.T.

Pogran, J. Vittal, D.A. Henderson, May 1977.

RFC 746, SUPDUP graphis extension, R. Stallman, March 1978.

FOOTNOTES

1. Of course the extent to which this was true shouldn't be overstated. Jon Postel famously simply announced himself as the "naming czar" when he was still a graduate student.

2. In contrast to technological democracy, network neutrality involves regulatory treatment of vendor efforts to differentiate service provision speed to and access by users through pricing mechanisms sometimes, though not always, driven by relations between service and content providers that are also subject to competition (antitrust) law.

3. Other genre distinctions have been found useful by those conducting research on the RFCs. Below (2012), for example, analysed all of the documents identifiable as "guides" by those in the field of technical communication for the ways in which they were used for community-building in a valuable case study for that community of scholars and practitioners.