Introduction

Every society has its diagram(s).
—GILLES DELEUZE, *Foucault*

This book is about a diagram, a technology, and a management style. The diagram is the *distributed network,* a structural form without center that resembles a web or meshwork. The technology is the digital *computer,* an abstract machine able to perform the work of any other machine (provided it can be described logically). The management style is *protocol,* the principle of organization native to computers in distributed networks. All three come together to define a new apparatus of control that has achieved importance at the start of the new millennium.

Much work has been done recently on theorizing the present historical moment and on offering periodizations to explain its historical trajectory. I am particularly inspired by five pages from Gilles Deleuze, "Postscript on Control Societies," which begin to define a chronological period after the modern age that is founded neither on the central control of the sovereign nor on the decentralized control of the prison or the factory. My book aims to flesh out the specificity of this third historical wave by focusing on the controlling computer technologies native to it.

How would control exist after decentralization? In former times control was a little easier to explain. In what Michel Foucault called the sovereign societies of the classical era, characterized by centralized power and sovereign fiat, control existed as an extension of the word and deed of the master, assisted by violence and other coercive factors. Later, the disciplinary societies of the modern era took hold, replacing violence with more bureaucratic forms of command and control.

Deleuze has extended this periodization into the present day by suggesting that after the disciplinary societies come the *societies of control.* Deleuze believed that there exist wholly new technologies concurrent with the societies of control. "The old sovereign societies worked with simple machines, levers, pulleys, clocks," he writes, "but recent disciplinary societies were equipped with thermodynamic machines[1] . . . control societies operate with a third generation of machines, with information technology and

---

Epigraph: Gilles Deleuze, *Foucault,* trans. Seán Hand (Minneapolis: University of Minnesota Press, 1986), p. 35.

1. "Thermodynamic machines" refers primarily to steam and internal combustion engines and to nuclear power.

computers."[2] Just as Marx rooted his economic theory in a strict analysis of the factory's productive machinery, Deleuze heralds the coming productive power of computers to explain the sociopolitical logics of our own age.

According to Critical Art Ensemble (CAE), the shift from disciplinary societies to control societies goes something like this:

> Before computerized information management, the heart of institutional command and control was easy to locate. In fact, the conspicuous appearance of the halls of power was used by regimes to maintain their hegemony. . . . Even though the monuments of power still stand, visibly present in stable locations, the agency that maintains power is neither visible nor stable. Power no longer permanently resides in these monuments, and command and control now move about as desired.[3]

The most extensive "computerized information management" system existing today is the Internet. The Internet is a global distributed computer network. It has its roots in the American academic and military culture of the 1950s and 1960s.[4] In the late 1950s, in response to the Soviet Sputnik launch and other fears connected to the Cold War,[5] Paul Baran at the Rand Corpo-

---

2. Gilles Deleuze, "Postscript on Control Societies," in *Negotiations,* trans. Martin Joughin (New York: Columbia University Press, 1990), p. 180; an alternate translation is available as "Postscript on the Societies of Control" in *October: The Second Decade, 1986–1996,* ed. Rosalind Krauss et al. (Cambridge: MIT Press, 1997).

3. Critical Art Ensemble, *Electronic Civil Disobedience and Other Unpopular Ideas* (New York: Autonomedia, 1996), pp. 7–8, 9.

4. Katie Hafner and Matthew Lyon dispute this in their book *Where Wizards Stay Up Late: The Origins of the Internet* (New York: Touchstone, 1996), arguing instead that the Internet was derived from the altruistic concerns of a few academics rather than the strategic interests of the Department of Defense. Yet they equivocate, writing on the one hand that "[t]he project had embodied the most peaceful intentions—to link computers at scientific laboratories across the country so that researchers might share computer resources. . . . the ARPANET and its progeny, the Internet, had nothing to do with supporting or surviving war—never did" (p. 10); yet on the other hand they admit that Paul Baran, the man who has contributed most to the emergence of protocol, "developed an interest in the survivability of communications systems under nuclear attack" (p. 54).

5. American anxiety over Soviet technological advancement was very real after the Sputnik launches. "The launching of the sputniks told us," wrote John Dunning for *The New York Times*

ration decided to create a computer network that was independent of centralized command and control, and would thus be able to withstand a nuclear attack that targets such centralized hubs. In August 1964, he published an eleven-volume memorandum for the Rand Corporation outlining his research.[6]

Baran's network was based on a technology called packet-switching[7] that allows messages to break themselves apart into small fragments. Each fragment, or packet, is able to find its own way to its destination. Once there, the packets reassemble to create the original message. In 1969, the Advanced Research Projects Agency (ARPA) at the U.S. Department of Defense started the ARPAnet, the first network to use Baran's packet-switching technology. The ARPAnet allowed academics to share resources and transfer files. In its early years, the ARPAnet (later renamed DARPAnet) existed unnoticed by the outside world, with only a few hundred participating computers, or "hosts."

All addressing for this network was maintained by a single machine located at the Stanford Research Institute in Menlo Park, California. By 1984 the network had grown larger. Paul Mockapetris invented a new addressing scheme, this one decentralized, called the Domain Name System (DNS).

The computers had changed also. By the late 1970s and early 1980s personal computers were coming to market and appearing in homes and offices. In 1977, researchers at Berkeley released the highly influential "BSD" flavor of the UNIX operating system, which was available to other institutions at

---

Magazine in 1957, "that a great despotism is now armed with rockets of enormous thrust, and guidance systems that could deliver a hydrogen warhead of one or more megatons to any spot in the United States." See John Dunning, "If We Are to Catch Up in Science," *New York Times Magazine,* November 10, 1957, p. 19.

6. Baran tells us that these memoranda "were primarily written on airplanes in the 1960 to 1962 era." See Paul Baran, Electrical Engineer, an oral history conducted in 1999 by David Hochfelder, IEEE History Center, Rutgers University, New Brunswick, NJ, USA.

7. A term coined instead by British scientist Donald Davies who, unknowing of Baran's work, also invented a system for sending small packets of information over a distributed network. Both scientists are credited with the discovery; however, because of Baran's proximity to the newly emerging ARPA network, which would be the first to use Baran's ideas, Davies's historical influence has diminished.

virtually no cost. With the help of BSD, UNIX would become the most important computer operating system of the 1980s.

In the early 1980s, the suite of protocols known as TCP/IP (Transmission Control Protocol/Internet Protocol) was also developed and included with most UNIX servers. TCP/IP allowed for cheap, ubiquitous connectivity. In 1988, the Defense department transferred control of the central "backbone" of the Internet over to the National Science Foundation, who in turn transferred control to commercial telecommunications interests in 1995. In that year, there were 24 million Internet users. Today, the Internet is a global distributed network connecting billions of people around the world.

At the core of networked computing is the concept of *protocol.* A computer protocol is a set of recommendations and rules that outline specific technical standards. The protocols that govern much of the Internet are contained in what are called RFC (Request For Comments) documents.[8] Called "the primary documentation of the Internet,"[9] these technical memoranda detail the vast majority of standards and protocols in use on the Internet today.

The RFCs are published by the Internet Engineering Task Force (IETF). They are freely available and used predominantly by engineers who wish to build hardware or software that meets common specifications. The IETF is affiliated with the Internet Society, an altruistic, technocratic organization that wishes "[t]o assure the open development, evolution and use of the Internet for the benefit of all people throughout the world."[10] Other protocols are developed and maintained by other organizations. For example, many of the protocols used on the World Wide Web (a network within the Internet) are governed by the World Wide Web Consortium (W3C). This international consortium was created in October 1994 to develop common protocols such as Hypertext Markup Language (HTML) and Cascading Style Sheets. Scores of other protocols have been created for a variety of other purposes by many

---

8. The expression derives from a memorandum titled "Host Software" sent by Steve Crocker on April 7, 1969, which is known today as RFC 1.

9. Pete Loshin, *Big Book of FYI RFCs* (San Francisco: Morgan Kaufmann, 2000), p. xiv.

10. "Internet Society Mission Statement," available online at http://www.isoc.org/isoc/mission/.

different professional societies and organizations. They are covered in more detail in chapter 4.

Protocol is not a new word. Prior to its usage in computing, protocol referred to any type of correct or proper behavior within a specific system of conventions. It is an important concept in the area of social etiquette as well as in the fields of diplomacy and international relations. Etymologically it refers to a fly-leaf glued to the beginning of a document, but in familiar usage the word came to mean any introductory paper summarizing the key points of a diplomatic agreement or treaty.

However, with the advent of digital computing, the term has taken on a slightly different meaning. Now, protocols refer specifically to standards governing the implementation of specific technologies. Like their diplomatic predecessors, computer protocols establish the essential points necessary to enact an agreed-upon standard of action. Like their diplomatic predecessors, computer protocols are vetted out between negotiating parties and then materialized in the real world by large populations of participants (in one case citizens, and in the other computer users). Yet instead of governing social or political practices as did their diplomatic predecessors, computer protocols govern how specific *technologies* are agreed to, adopted, implemented, and ultimately used by people around the world. What was once a question of consideration and sense is now a question of logic and physics.

To help understand the concept of computer protocols, consider the analogy of the highway system. Many different combinations of roads are available to a person driving from point A to point B. However, en route one is compelled to stop at red lights, stay between the white lines, follow a reasonably direct path, and so on. These conventional rules that govern the set of possible behavior patterns within a heterogeneous system are what computer scientists call protocol. Thus, protocol is a technique for achieving voluntary regulation within a contingent environment.

These regulations always operate at the level of coding—they encode packets of information so they may be transported; they code documents so they may be effectively parsed; they code communication so local devices may effectively communicate with foreign devices. Protocols are highly formal; that is, they encapsulate information inside a technically defined wrapper, while remaining relatively indifferent to the content of information

contained within. Viewed as a whole, protocol is a distributed management system that allows control to exist within a heterogeneous material milieu.

It is common for contemporary critics to describe the Internet as an unpredictable mass of data—rhizomatic and lacking central organization. This position states that since new communication technologies are based on the elimination of centralized command and hierarchical control, it follows that the world is witnessing a general disappearance of control as such.

This could not be further from the truth. I argue in this book that protocol is how technological control exists after decentralization. The "after" in my title refers to both the historical moment after decentralization has come into existence, but also—and more important—the historical phase *after* decentralization, that is, after it is dead and gone, replaced as the supreme social management style by the diagram of distribution.

What contributes to this misconception (that the Internet is chaotic rather than highly controlled), I suggest, is that protocol is based on a *contradiction* between two opposing machines: One machine radically distributes control into autonomous locales, the other machine focuses control into rigidly defined hierarchies. The tension between these two machines—a dialectical tension—creates a hospitable climate for protocological control.

Emblematic of the first machinic technology, the one that gives the Internet its common image as an uncontrollable network, is the family of protocols known as TCP/IP. TCP and IP are the leading protocols for the actual transmission of data from one computer to another over the network. TCP and IP work together to establish connections between computers and move data packets effectively through those connections. Because of the way TCP/IP was designed, any computer on the network can talk to any other computer, resulting in a nonhierarchical, peer-to-peer relationship.

As one technical manual puts it: "IP uses an anarchic and highly distributed model, with every device being an equal peer to every other device on the global Internet."[11] (That a technical manual glowingly uses the term "anarchic" is but one symptom of today's strange new world!)

Emblematic of the second machinic technology, the one that focuses control into rigidly defined hierarchies, is the DNS. DNS is a large decentralized

---

11. Eric Hall, *Internet Core Protocols: The Definitive Guide* (Sebastopol, CA: O'Reilly, 2000), p. 407.

database that maps network addresses to network names. This mapping is required for nearly every network transaction. For example, in order to visit "www.rhizome.org" on the Internet one's computer must first translate the name "www.rhizome.org," itself geographically vague, into a specific address on the physical network. These specific addresses are called IP addresses and are written as a series of four numbers like so: 206.252.131.211.

All DNS information is controlled in a hierarchical, inverted-tree structure. Ironically, then, nearly all Web traffic must submit to a hierarchical structure (DNS) to gain access to the anarchic and radically horizontal structure of the Internet. As I demonstrate later, this contradictory logic is rampant throughout the apparatus of protocol.

The process of converting domain names to IP addresses is called *resolution.* At the top of this inverted tree are a handful of so-called "root" servers holding ultimate control and delegating lesser control to lower branches in the hierarchy. There are over a dozen root servers located around the world in places like Japan and Europe, as well as in several U.S. locations.

To follow the branches of control, one must parse the address in reverse, starting with the top-level domain, in this case "org." First, the root server receives a request from the user and directs the user to another machine that has authority over the "org" domain, which in turn directs the user to another machine that has authority over the "rhizome" subsection, which in turn returns the IP address for the specific machine known as "www."

Only the computer at the end of the branch knows about its immediate neighborhood, and thus it is the only machine with authoritative DNS information. In other words resolution happens like this: A new branch of the tree is followed at each successive segment, allowing the user to find the authoritative DNS source machine and thus to derive the IP address from the domain name. Once the IP address is known, the network transaction can proceed normally.

Because the DNS system is structured like an inverted tree, each branch of the tree holds absolute control over everything below it. For example, in the winter of 1999, a lawsuit was brought against the Swiss art group Etoy. Even though the basis of the lawsuit was questionable and was later dropped, the courts would have been able to "turn off" the artist's Web site during the course of the trail by simply removing DNS support for "etoy.com." (Instead the artists were forced to pull the plug themselves until after the trial was over.)

A similar incident happened at The Thing, an Internet service provider based in New York who was hosting some of Etoy's agitprop. After some of this material was deemed politically questionable by the Federal Bureau of Investigation, the whole server was yanked off the Internet by the telecommunications company who happened to be immediately upstream from the provider. The Thing had no recourse but to comply with this hierarchical system of control.

The inventor of the World Wide Web, Tim Berners-Lee, describes the DNS system as the "one centralized Achilles' heel by which [the Web] can all be brought down or controlled."[12]

If hypothetically some controlling authority wished to ban China from the Internet (e.g., during an outbreak of hostilities), they could do so very easily through a simple modification of the information contained in the root servers at the top of the inverted tree. Within twenty-four hours, China would vanish from the Internet.

As DNS renegade and Name.Space founder Paul Garrin writes: "With the stroke of a delete key, whole countries can be blacked out from the rest of the net. With the "." [root file] centralized, this is easily done. . . . Control the "." and you control access."[13] Since the root servers are at the top, they have ultimate control over the existence (but not necessarily the content) of each lesser branch. Without the foundational support of the root servers, all lesser branches of the DNS network become unusable. Such a reality should shatter our image of the Internet as a vast, uncontrollable meshwork.

Any networked relation will have multiple, nested protocols. To steal an insight from Marshall McLuhan, *the content of every new protocol is always another protocol.* Take, for example, a typical transaction on the World Wide Web. A Web page containing text and graphics (themselves protocological artifacts) is marked up in the HTML protocol. The protocol known as Hypertext Transfer Protocol (HTTP) encapsulates this HTML object and allows it to be served by an Internet host. However, both client and host must abide by the TCP protocol to ensure that the HTTP object arrives in one piece. Finally, TCP is itself nested within the Internet Protocol, a protocol

---

12. Tim Berners-Lee, *Weaving the Web* (New York: HarperCollins, 1999), p. 126.

13. Paul Garrin, "DNS: Long Winded and Short Sighted," *Nettime,* October 19, 1998.

that is in charge of actually moving data packets from one machine to another. Ultimately the entire bundle (the primary data object encapsulated within each successive protocol) is transported according to the rules of the only "privileged" protocol, that of the physical media itself (fiber-optic cables, telephone lines, air waves, etc.). The flexible networks and flows identified in the world economy by Manuel Castells and other anchormen of the Third Machine Age are not mere metaphors; they are in fact built directly into the technical specifications of network protocols. By design, protocols such as the Internet Protocol *cannot be centralized.*

Protocol's native landscape is the distributed network. Following Deleuze, I consider the distributed network to be an important *diagram* for our current social formation. Deleuze defines the diagram as "a map, a cartography that is coextensive with the whole social field."[14] The distributed network is such a map, for it extends deeply into the social field of the new millennium. (I explore this point in greater detail in chapter 1.)

A distributed network differs from other networks such as centralized and decentralized networks in the arrangement of its internal structure. A centralized network consists of a single central power point (a host), from which are attached radial nodes. The central point is connected to all of the satellite nodes, which are themselves connected only to the central host. A decentralized network, on the other hand, has *multiple* central hosts, each with its own set of satellite nodes. A satellite node may have connectivity with one or more hosts, but not with other nodes. Communication generally travels unidirectionally within both centralized and decentralized networks: from the central trunks to the radial leaves.

The distributed network is an entirely different matter. Distributed networks are native to Deleuze's control societies. Each point in a distributed network is neither a central hub nor a satellite node—there are neither trunks nor leaves. The network contains nothing but "intelligent end-point systems that are self-deterministic, allowing each end-point system to communicate with any host it chooses."[15] Like the rhizome, each node in a distributed network may establish direct communication with another node,

---

14. Deleuze, *Foucault,* p. 34.

15. Hall, *Internet Core Protocols,* p. 6.

without having to appeal to a hierarchical intermediary. Yet in order to initiate communication, the two nodes must *speak the same language.* This is why protocol is important. Shared protocols are what defines the landscape of the network—who is connected to whom.

As architect Branden Hookway writes: "[d]istributed systems require for their operation a homogenous standard of interconnectivity."[16] Compatible protocols lead to network articulation, while incompatible protocols lead to network disarticulation. For example, two computers running the DNS addressing protocol will be able to communicate effectively with each other about network addresses. Sharing the DNS protocol allows them to be networked. However, the same computers will not be able to communicate with foreign devices running, for example, the NIS addressing protocol or the WINS protocol.[17] Without a shared protocol, there is no network.

I turn now to Michel Foucault to derive one final quality of protocol, the special existence of protocol in the "privileged" physical media of *bodies.* Protocol is not merely confined to the digital world. As Deleuze shows in the "Postscript on Control Societies," protocological control also affects the functioning of bodies within social space and the creation of these bodies into forms of "artificial life" that are *dividuated,*[18] sampled, and coded. "Artificial life" is a term I use in chapter 3 to describe protocol *within the sociopolitical theater.* Artificial life simply means the active production of vital forms by other vital forms—what Foucault calls the "work of the self on the self."

I later suggest that Foucault's relationship to life forms is a protocological one. This is expressed most clearly in his later work, particularly in the twin concepts of biopolitics and biopower. Foucault defines biopolitics as "the endeavor, begun in the eighteenth century, to rationalize the problems presented to governmental practice by the phenomena characteristic of a

---

16. Branden Hookway, *Pandemonium: The Rise of Predatory Locales in the Postwar World* (New York: Princeton Architectural Press, 1999), p. 77.

17. WINS, or Windows Internet Name Service, is an addressing technology developed by Microsoft for distributed networks; NIS, or Network Information Service, is a similar technology developed by Sun Microsystems.

18. Deleuze's neologism comes from the word "*in*dividuate." Dividuation would thus be the opposite: the dissolving of individual identity into distributed networks of information.

group of living human beings constituted as a population: health, sanitation, birthrate, longevity, race."[19] Thus one can assume that technologies like biometrics and statistical analysis—from the Bertillon identification system, to the Social Security Act of 1935, to the tabulation of birth rates by the Children's Defense Fund—all fall into the category biopolitics.

Further, he writes that biopolitics "tends to treat the 'population' as a mass of living and coexisting beings who present particular biological and pathological traits and who thus come under specific knowledge and technologies."[20] Biopolitics, then, connects to a certain statistical knowledge about populations. It is a species-knowledge (an expression that sounds less ominous if one considers an allusion to Marx's utopian concept of "species-being").

Still, Foucault puts equal stress on "technologies" and "knowledge" in his definition of biopolitics. But which technologies in particular would correspond to Foucault's biopolitical scenario? I argue here that they are the distributed forms of management that characterize the contemporary computer network and within which protocological control exists.

In *The History of Sexuality, Volume 1,* Foucault contrasts the older power of the sovereign over life (one characterized by the metaphysical concern of either the absence or presence of life) to a new mode in which life is either created or destroyed: "One might say that the ancient right to *take* life or *let* live was replaced by a power to *foster* life or *disallow* it to the point of death."[21] He continues: "The old power of death that symbolized sovereign power was now carefully supplanted by the *administration of bodies* and the *calculated management of life.*"[22] Foucault's treatment of biopower is entirely protocological. Protocol is to control societies as the panopticon is to disciplinary societies.

While protocol may be more *democratic* than the panopticon in that it strives to eliminate hierarchy, it is still very much structured around command and control and therefore has spawned counter-protocological forces.

---

19. Michel Foucault, *Ethics: Subjectivity and Truth,* ed. Paul Rabinow (New York: New Press, 1997), p. 73.

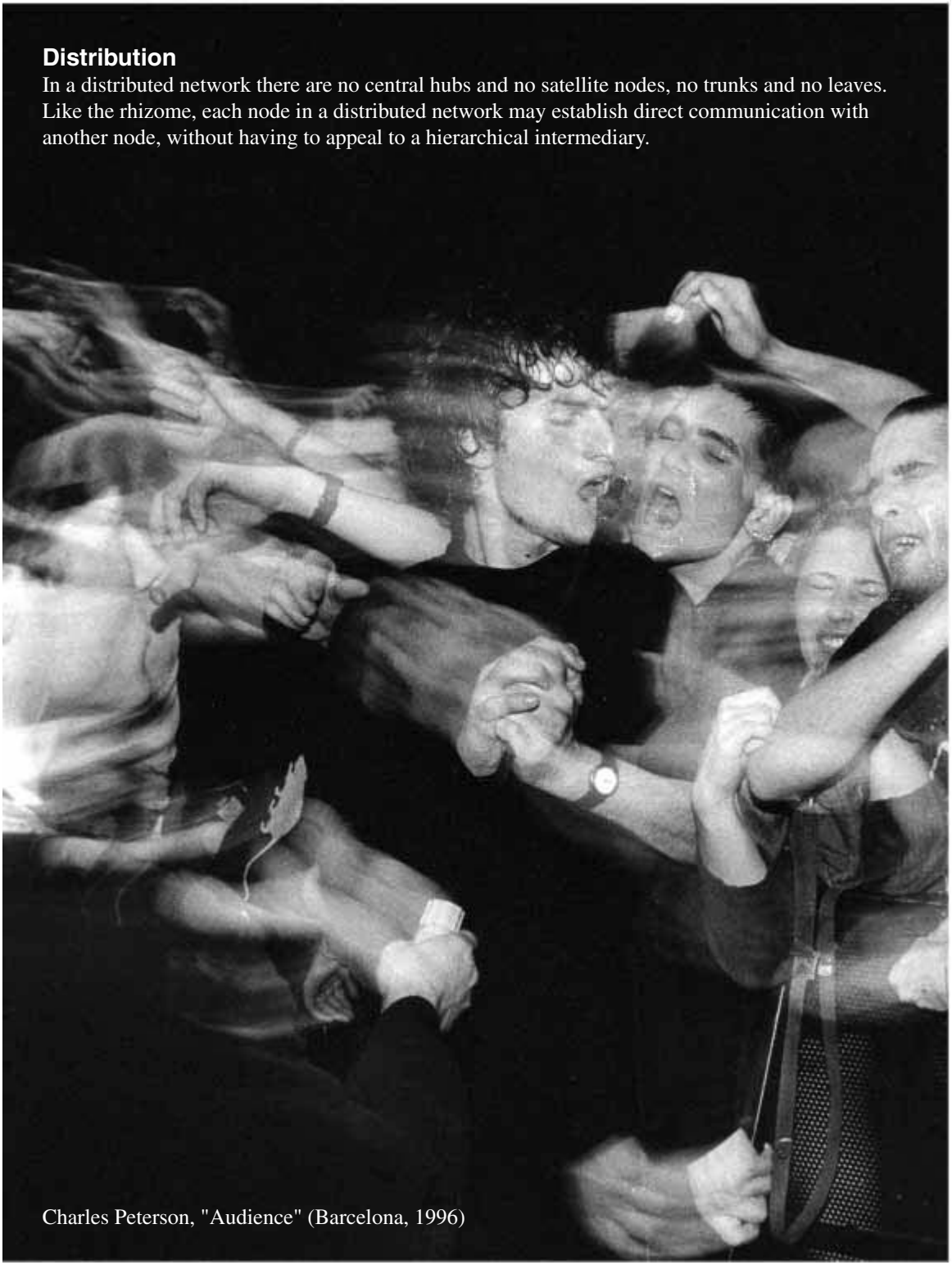
20. Foucault, *Ethics,* p. 71.

21. Michel Foucault, *The History of Sexuality, Volume 1,* trans. Robert Hurley (New York: Vintage, 1978), p. 138.

22. Foucault, *The History of Sexuality, Volume 1,* pp. 138–140, emphasis mine.

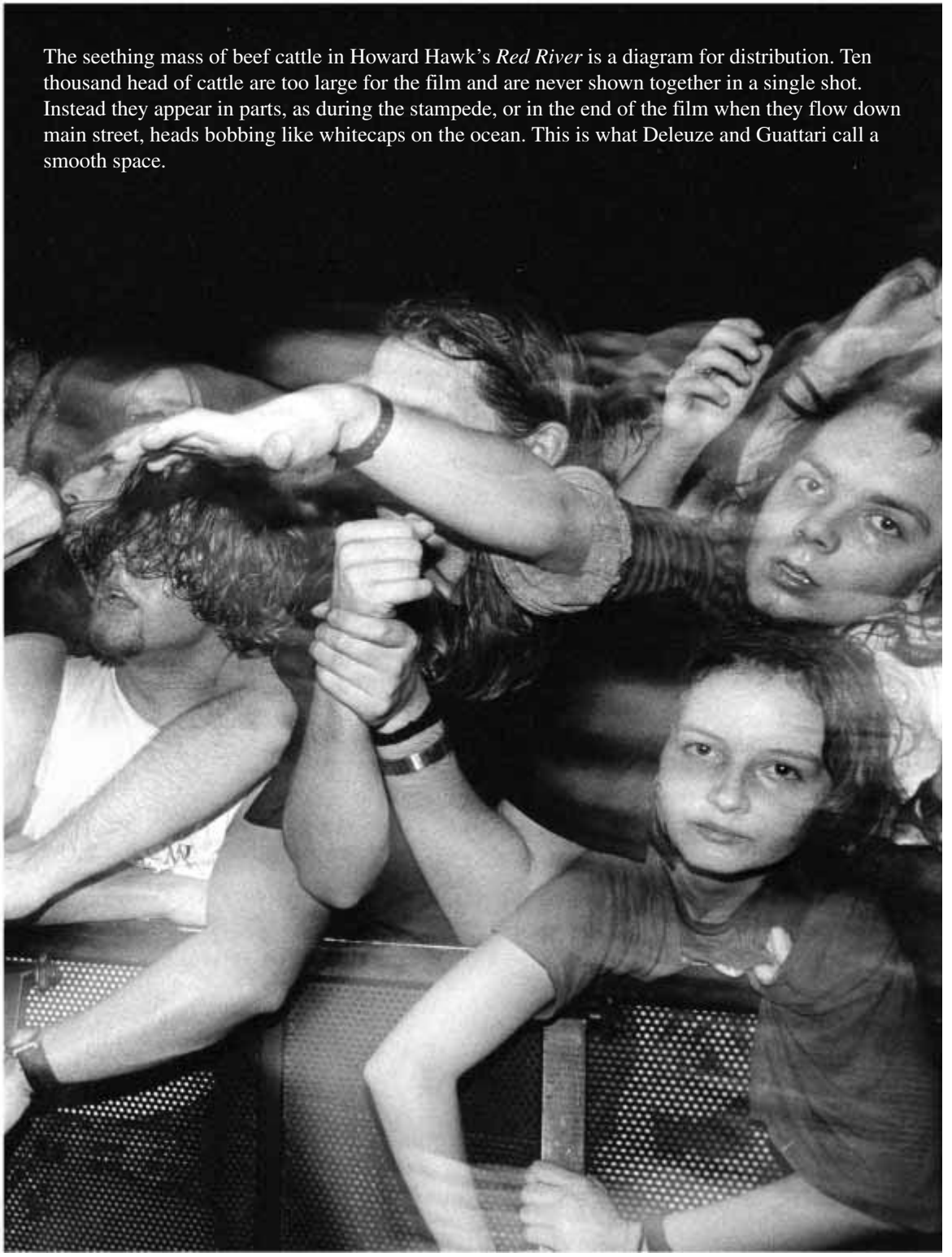
Distribution

In a distributed network there are no central hubs and no satellite nodes, no trunks and no leaves. Like the rhizome, each node in a distributed network may establish direct communication with another node, without having to appeal to a hierarchical intermediary.



Charles Peterson, "Audience" (Barcelona, 1996)

The seething mass of beef cattle in Howard Hawk's *Red River* is a diagram for distribution. Ten thousand head of cattle are too large for the film and are never shown together in a single shot. Instead they appear in parts, as during the stampede, or in the end of the film when they flow down main street, heads bobbing like whitecaps on the ocean. This is what Deleuze and Guattari call a smooth space.



Deleuze recognized this, that the very site of Foucault's biopower was also a site of resistance.

Lest readers overlook its importance, he repeats his realization three times consecutively in an important section of his book *Foucault:* "[1] When power . . . takes life as its aim or object, then resistance to power already puts itself on the side of life, and turns life against power. . . . [2] Life becomes resistance to power when power takes life as its object. . . . [3] When power becomes bio-power resistance becomes the power of life, a vital power that cannot be confined within species, environment or the paths of a particular diagram."[23] Is *life resistance* a way of engaging with distributed forms of protocological management?

Part III of this book, "Protocol Futures," answers yes. While the new networked technologies have forced an ever more reticent public to adapt to the control structures of global capital, there has emerged a new set of social practices that inflects or otherwise diverts these protocological flows toward the goal of a utopian form of unalienated social life.

What is wrong with protocol? To steal a line from Foucault, it's not that protocol is bad but that protocol is *dangerous.* To refuse protocol, then, is not so much to reject today's technologies as did Theodore Kaczynski (the Unabomber), but to direct these protocological technologies, whose distributed structure is empowering indeed, toward what Hans Magnus Enzensberger calls an "emancipated media" created by active social actors rather than passive users.[24]

As Deleuze remarked to Antonio Negri several years ago:

> It's true that, even before control societies are fully in place, forms of delinquency or resistance (two different things) are also appearing. Computer piracy and viruses, for example, will replace strikes and what the nineteenth century called "sabotage" . . . You ask whether control or communication societies will lead to forms of resistance

---

23. Deleuze, *Foucault,* p. 92.

24. Natalie Jeremijenko uses the rubric of "structures of participation" to think about how certain implementations of technology promote active user involvement and understanding while other technologies obfuscate understanding and control user involvement.

that might reopen the way for a communism . . . The key thing may be to create vacuoles of noncommunication, circuit breakers, so we can elude control.[25]

The key here is less the eluding or the breaking or the *non*communication, but simply that Deleuze had the foresight to situate resistive action *within the protocological field.* In the same way that biopower is a species-level knowledge, protocol is a type of species-knowledge for coded life forms. Each new diagram, each new technology, each new management style both is an improvement on the previous one and contains with it a germ that must grow into a still higher form. I am not suggesting that one should learn to love the various apparatuses of control, but rather that, for all its faults, protocological control is still an improvement over other modes of social control. I hope to show in this book that it is *through* protocol that one must guide one's efforts, not against it.

"No more vapor theory anymore," wrote Geert Lovink. Vapor theory tends to ignore the computer itself. The computer is often eclipsed by that more familiar thing, information society. Mine is not a book about information society, but about the real machines that live within that society.

Thus, my study skips direct engagement with the work of Alvin Toffler, Peter Drucker, Daniel Bell, and others who discuss the third phase of capitalist development in social terms.

The large mass of literature devoted to artificial intelligence and speculations about the consciousness (or lack thereof) within man and machine is also largely avoided in this book. Writers like Ray Kurzweil forecast a utopian superfuture dominated by immortal man-machine hybrids. Hans Moravec predicts a similar future, only one less populated by humans who are said to "retire" to the mercy of their ascendant computerized progeny.

Marvin Minsky, Daniel Dennett, John Searle, Hubert Dreyfus, and others have also wrestled with the topic of artificial intelligence. But they are not addressed here. I draw a critical distinction between this body of work, which is concerned largely with epistemology and cognitive science, and the critical media theory that inspires this book. Where they are concerned with

---

25. Gilles Deleuze, "Control and Becoming," in *Negotiations,* trans. Martin Joughin (New York: Columbia University Press, 1990), p. 175.

minds and questions epistemological, I am largely concerned with bodies and the material stratum of computer technology.

My study also ignores the large mass of popular responses to the new technological age, such as Nicholas Negroponte's *Being Digital,* whose gee-whiz descriptions of the incredible *newness* of new technologies seem already dated and thin.

Except for chapter 4, this is largely *not* a book about issues specifically relating to law, Internet governance, state sovereignty, commercial power, or the like. Several books already do an excellent job covering these issues including Milton Mueller's *Ruling the Root.*

While my ultimate indebtedness to many of these authors will be obvious, it is not my goal to examine the social or culturo-historical characteristics of informatization, artificial intelligence, or virtual anything, but rather to study computers as André Bazin studied film or Roland Barthes studied the striptease: to look at a material technology and analyze its specific formal functions and dysfunctions.

To that end this book focuses on distributed computer networks and the protocological system of control present within them. I hope to build on texts such as Friedrich Kittler's groundbreaking *Discourse Networks, 1800/1900,* which describes the paradigm shift from a discourse driven by meaning and sense, to our present milieu of pattern and code. Kittler's two ages, symbolized by the two years 1800 and 1900, correspond structurally (but less so chronologically) to the social periodization supplied by Foucault and Deleuze. The passage from the modern disciplinary societies to those of the control societies, as I have already suggested, is the single most important historical transformation in this book.

Norbert Wiener is also an important character. His books laid important groundwork for how control works within physical bodies. The provocative but tantalizingly thin *Pandemonium: The Rise of Predatory Locales in the Post-war World* from architect Branden Hookway, looks at how cybernetic bodies permeate twentieth-century life. Other important theorists from the field of computer and media studies who have influenced me include Vannevar Bush, Hans Magnus Enzensberger, Marshall McLuhan, Lewis Mumford, and Alan Turing.

I am also inspired by Lovink's new school of media theory known as Net criticism. This loose international grouping of critics and practitioners has grown up with the Internet and includes the pioneering work of Hakim Bey

and Critical Art Ensemble, as well as newer material from Timothy Druckrey, Marina Gržinić, Lev Manovich, Sadie Plant, and many others. Much of this intellectual work has taken place in online venues such as *CTHEORY, Nettime,* and *Rhizome,* plus conferences such as the annual Ars Electronica festival and the Next 5 Minutes series on tactical media.

Although my book is heavily influenced by film and video theory, I include here little discussion of media formats prior to the digital computer.[26] I gain much of my momentum by relying on the specificity of the digital computer as a medium, not its similarity to other visual media. In my estimation, it makes little sense to try to fit non-protocological and nondistributed media such as film and video into this new context—in the same way that it makes little sense to speak of the aura of a Web page, or the essence of a digital text. Nevertheless the history of avant-garde artistic production, from modernist painting to conceptual art, significantly influences my perspective vis-à-vis work being done today.

While lay readers may group all literature dealing with new technologies under the general heading informatization, there is an alternate path that I attempt to follow in this book. This alternate path recognizes the material substrate of media, and the historical processes that alter and create it. It attempts to chart what Manuel DeLanda calls "institutional ecologies." He writes here of the history of warfare, but it could easily refer to digital computing:

> I would like to repeat my call for more realistic models of economic history, models involving the full complexity of the institutional ecologies involved, including markets, anti-markets, military and bureaucratic institutions, and if we are to believe Michel Foucault, schools, hospitals, prisons, and many others. It is only through an honest philosophical confrontation with our complex past that we can expect to understand it and derive the lessons we may use when intervening in the present and speculating about the future.[27]

---

26. For an anthology of recent writing on the confluence of cinematic practices and new media, see Martin Rieser and Andrea Zapp, eds., *New Screen Media: Cinema/Art/Narrative* (London: BFI, 2002).

27. Manuel DeLanda, "Economics, Computers, and the War Machine," in *Ars Electronica: Facing the Future,* ed. Timothy Druckrey (Cambridge: MIT Press, 1999), p. 325.

The complex "institutional ecology" of modern computing is thus the focus of this book.

Just as Marx descended into the internal structure of the commodity to interpret its material workings within the context of production at large, I must descend instead into the distributed networks, the programming languages, the computer protocols, and other digital technologies that have transformed twenty-first-century production into a vital mass of immaterial flows and instantaneous transactions.

Indeed, I attempt to read the never-ending stream of computer code *as one reads any text* (the former having yet to achieve recognition as a natural language), decoding its structure of control as one would a film or novel.

Periodization

Let me pause for a minute to address something that is taken for granted throughout much of the rest of this book. I refer to the axiom, taken from periodization theory, that history may be divided into certain broad phases, and that the late twentieth century is part of a certain phase that (although it goes by several different names) I refer to alternatively as the postmodern or digital age.

It is no mystery to scholars of critical theory that, while terminology and timelines may differ, a whole series of thinkers have roughly agreed on three broad historical phases, these being the classical era, the modern era, and the postmodern era.[28] This general consensus is what I would like to describe briefly now, not to fetishize its overarching structure, but instead to observe that "periodization *is an initial technique that opens the path and allows us to gain access to history and historical differences.*"[29] While this comparativist approach to periodization theory will undoubtedly land me in somewhat treacherous waters (for who is able to align so many different thinkers chronologically,

---

28. This triad refers primarily to the sociopolitical realm. In the realm of culture, a different triad becomes more important, that of realism, modernism, and postmodernism. See especially Fredric Jameson's extended essay "The Existence of Italy," in *Signatures of the Visible* (New York: Routledge, 1992).

29. This citation is from Michael Hardt and Kathi Weeks's interpretation of Fredric Jameson in *The Jameson Reader* (Oxford: Blackwell, 2000), p. 13, emphasis in original.

much less structurally!), I feel that the overwhelming consensus among many of my theoretical sources must be brought into the light of day before I continue with my own observation—that protocol is a system of management *historically posterior* to decentralization.

Foucault—both in his own writings, and as he has been interpreted by Deleuze—has put forth perhaps the clearest periodization. Foucault was especially interested in the historical shift from what he called the sovereign, or "classical," era of the eighteenth century, and the disciplinary, or "modern," era beginning after the French Revolution and extending into the early part of the twentieth century.

In his persuasive introduction to *Discipline and Punish,* Foucault observes that this historical transformation transpired, at least in the prison system and other systems of socialized punishment, between the years 1750 and 1830. While physical punishment was more dominant during the eighteenth century, "[a]t the beginning of the nineteenth century," writes Foucault, "the great spectacle of physical punishment disappeared . . . The age of sobriety in punishment had begun."[30] At the same time that punishment became more "sober" it also became more diffuse, more immanent to the personal lives and habits of people. Good citizens were now expected to punish *themselves,* to preemptively discipline their own bodies such that the power of punishment originated ultimately from within, not from some outside force.

This historical shift, from sovereign society to disciplinary society, reoccurs throughout the writings of Foucault, particularly in texts such as *Madness and Civilization* and *The History of Sexuality, Volume 1.* One may make the analogy that this transformation is the same as the shift from a centralized diagram (one overseer) to a decentralized diagram (many overseers).

Deleuze reinforces the historical arguments, first presented by Foucault, in his book *Foucault,* as well as in several interviews and incidental texts in the collection *Negotiations.* Deleuze's contribution was to flesh out the later segment of Foucault's periodization, and to suggest that Foucault was as clearly in tune with the second shift from disciplinarity to control as he was

---

30. Michel Foucault, *Discipline and Punish,* trans. Alan Sheridan (New York: Vintage, 1995), p. 14.

with the first shift from sovereignty to disciplinarity. While Deleuze's writings on Foucault may in fact tell readers more about Deleuze's predilections than Foucault's, nevertheless Deleuze has much to contribute, especially by establishing a connection between control society and computers (a word hardly mentioned in Foucault, if at all).

Deleuze defines the relationship between the different social phases and their native machinic technologies very clearly, in two different texts. The first comes from his 1990 interview with Antonio Negri, where he writes: "Each kind of society corresponds to a particular kind of machine—with simple mechanical machines corresponding to sovereign societies, thermodynamic machines to disciplinary societies, cybernetic machines and computers to control societies."[31] A few months later, in his "Postscript on Control Societies," Deleuze says much the same thing: "It's easy to set up a correspondence between any society and some kind of machine . . . The old sovereign societies worked with simple machines, levers, pulleys, clocks; but recent disciplinary societies were equipped with thermodynamic machines . . . ; control societies function with a third generation of machines, with information technology and computers."[32] In Deleuze, therefore, computers are historically concurrent with control societies.

Kittler agrees roughly with this periodization in his book *Discourse Networks, 1800/1900.* Reminiscent of Foucault's genealogies, Kittler's book is a history of knowledge over the last two hundred years. Kittler looks at two years—1800 and 1900—and shows how the state of knowledge changed from a "kingdom of sense" (in 1800) based on understanding and meaning to a "kingdom of pattern" (in 1900) based on images and algorithms.

He defines a discourse network as "the network of technologies and institutions that allow a given culture to select, store, and process relevant data."[33] Discourse networks change, as disciplinary networks changed for Foucault, and it is this transformation that so interests Kittler. He writes:

---

31. Deleuze, *Negotiations,* p. 175.

32. Deleuze, *Negotiations,* p. 180.

33. Friedrich Kittler, *Discourse Networks, 1800/1900,* trans. Michael Metteer and Chris Cullens (Stanford: Stanford University Press, 1990), p. 369.

In the discourse network of 1900, discourse is produced by RANDOM GENERATORS. Psychophysics constructed such sources of noise; the new technological media stored their output . . . The discourse network of 1900 was the first to establish a treasury of the signifier whose rules were entirely based on randomness and combinatorics . . . The discourse network of 1800 played the game of not being a discourse network and pretended instead to be the inwardness and voice of Man; in 1900 a type of writing assumes power that does not conform to traditional writing systems but rather radicalizes the technology of writing in general.[34]

Kittler's 1800 kingdom of sense corresponds roughly to Foucault's sovereign societies: Both are interested in depth, in probing to the heart of a body or an object to derive its essential meaning. 1800 is the year of the signifier.

At the same time Kittler's 1900 kingdom of pattern corresponds roughly to Foucault's disciplinary societies: Both are interested in the patterned affection of bodies and information. In what Kittler calls the "logic of chaos and intervals,"[35] the machinic processes embodied in the patterning apparatus of the typewriter or the phonograph come to the fore. 1900 is the year of the algorithm. Again, one may make the analogy that this transformation is the transformation from centralization (singular meaning) to decentralization (meaning's replication).

In the sociopolitical realm many thinkers have also charted this same periodization. Ernst Mandel uses the concept of Kondratieff waves to examine what he calls the era of late capitalism beginning in approximately 1945. "As far as I can see," writes Fredric Jameson, "the general use of the term *late capitalism* originated with the Frankfurt School; it is everywhere in Adorno and Horkheimer, sometimes varied with their own synonyms (for example, 'administered society')."[36] Jameson states that the concept is ultimately Mandel's: "There have been three fundamental moments in capitalism, each one marking a dialectical expansion over the previous stage. These are market

---

34. Kittler, *Discourse Networks, 1800/1900,* pp. 206, 210, 211–212.

35. Kittler, *Discourse Networks, 1800/1900,* p. 192.

36. Fredric Jameson, *Postmodernism, or, The Cultural Logic of Late Capitalism* (Durham: Duke University Press, 1991), p. xviii.

capitalism, the monopoly stage or the stage of imperialism, and our own, wrongly called postindustrial, but what might be better termed multinational capital,"[37] or to use Mandel's terminology, late capitalism.

Like other social critics of late-twentieth-century life, Jameson looks to the economic crisis of 1973 as a turning point, a moment that "somehow crystallized"[38] these new currents of postmodernity. Jameson admits that Mandel's work "is what made [his] own thoughts on 'postmodernism' possible."[39]

Sociologist Manuel Castells has also documented this transformation out of decentralization into new distributed, flexible economies in his three-volume treatise *The Information Age: Economy, Society and Culture.* Using the term "network society" (rather than Deleuze's "society of control" or Jameson's "late capitalism"), Castells shows with extensive quantitative documentation that today's sociopolitical space is dominated not by robust national economies and core industrial sectors but by "interactive networks" and "flexible accumulation."

Charting the same periodization that I rely on in this book, Castells shows how, for example, corporate business structures have changed in the last several decades from a decentralized "vertical" corporatism to a more distributed "horizontal" meshwork: "The corporation itself has changed its organizational model, to adapt to the conditions of unpredictability ushered in by rapid economic and technological change. The main shift can be characterized as the shift from vertical bureaucracies to the horizontal corporation."[40] This transformation echoes the structural difference that Deleuze and Guattari see between the tree and the rhizome.[41] Trees correspond to vertical bureaucracies, while rhizomes correspond to horizontal meshworks.

While Michael Hardt and Antonio Negri have an almost identical analysis of contemporary economics in their book *Empire,* their analysis of poli-

---

37. Jameson, *Postmodernism,* p. 35.

38. Jameson, *Postmodernism,* p. xx.

39. Jameson, *Postmodernism,* p. 400.

40. Manuel Castells, *The Information Age: Economy, Society and Culture: Volume 1—The Rise of the Network Society* (Oxford: Blackwell, 1996), p. 164, emphasis removed from original.

41. See Gilles Deleuze and Félix Guattari, *A Thousand Plateaus,* trans. Brian Massumi (Minneapolis: University of Minnesota Press, 1987), chapter 1.

tics is more sophisticated. Conscious of their relationship to Foucault and Deleuze's argument described earlier, Hardt and Negri connect the society of control to the new world order they call "Empire."

First, they define the pre-imperial forces of the disciplinary society: "[d]isciplinary society is that society in which social command is constructed through a diffuse network of *dispositifs* or apparatuses that produce and regulate customs, habits, and productive practices."[42] Then, they define the society of control as that society "in which mechanisms of command become ever more 'democratic,' ever more immanent to the social field, *distributed* throughout the brains and bodies of the citizens."[43]

Hardt and Negri specifically address new media in *Empire,* writing that, within the Internet, "[a]n indeterminate and potentially unlimited number of interconnected nodes communicate with no central point of control."[44] In their opinion this "decentralized" architecture is "what makes control of the network so difficult."[45]

While I spend much of this book arguing against such descriptions of the Internet (i.e., I argue that the Internet is distributed not decentralized and that it is in fact highly controlled despite having few if any *central* points of control), this appears to be a nonfatal mistake in their argument. The attentive reader will notice that here Hardt and Negri actually mean *modern* control and not imperial control. For what they say elsewhere about Empire should also be true here about new media. A distributed architecture is precisely that which makes protocological/imperial control of the network so easy. In fact, the various Internet protocols mandate that control may *only* be derived from such a distributed architecture.

---

42. Michael Hardt and Antonio Negri, *Empire* (Cambridge: Harvard University Press, 2000), p. 23. These "dispositifs" allude to the apparatuses of the prison or the hospital observed by Foucault, or even more specifically to the "ideological state apparatuses" and "repressive state apparatuses" observed by Foucault's teacher Louis Althusser, through whose work the term "apparatus" gained prominence in film studies and other critical theory of the late 1960s and 1970s.

43. Hardt and Negri, *Empire,* p. 23, emphasis mine.

44. Hardt and Negri, *Empire,* p. 299.

45. Hardt and Negri, *Empire,* p. 299.

Hardt and Negri confirm this position by writing elsewhere that "the passage to the society of control does not in any way mean the end of discipline [i.e., control]. In fact, the immanent exercise of discipline . . . is extended even more generally in the society of control."[46]

The computer protocol is thus in lockstep with Hardt and Negri's analysis of Empire's logics, particularly the third mode of imperial command, the *managerial* economy of command.[47] This command protocol knows from the start that "[c]ontingency, mobility, and flexibility are Empire's real power."[48] Protocological control mirrors the movements of Empire. In fact one might go so far as to say that *Empire is the social theory and protocol the technical.* Thus Hardt and Negri are accurate in their analysis of the "Symptoms of Passage." An analysis of computer protocols proves this, for it reassigns the former weapons of Leftists—celebration of difference, attack on essentialism, and so forth—as the new tools of Empire: "This new enemy not only is resistant to the old weapons but actually thrives on them, and thus joins its would-be antagonists in applying them to the fullest. Long live difference! Down with essentialist binaries."[49] A distributed network is precisely what gives IP its effectiveness as a dominant protocol. Or to take another example, the flimsy, cross-platform nature of HTML is precisely what gives it its power as a protocological standard. Like Empire, if protocol dared to centralize, or dared to hierarchize, or dared to essentialize, it would fail.

Further to these many theoretical interventions—Foucault, Deleuze, Kittler, Mandel, Castells, Jameson, Hardt and Negri—are many dates that roughly confirm my periodization: the discovery of DNA in 1953; the economic crisis in the West during the 1970s epitomized by President Richard Nixon's decoupling of the U.S. dollar from the gold standard on August 17, 1971 (and thus the symbolic evaporation of the Bretton Woods agreement); Charles Jencks's claim that modern architecture ended on July 15, 1972, at 3:32 P.M.; the ARPAnet's mandatory rollover to TCP/IP on January 1, 1983; the fall of the Berlin Wall in 1989; the crashing of AT&T's long-distance

46. Hardt and Negri, *Empire,* p. 330.

47. Hardt and Negri, *Empire,* p. 199.

48. Hardt and Negri, *Empire,* p. 200.

49. Hardt and Negri, *Empire,* p. 138.

Table 1

Periodization Map

Period	Machine	Dates	Diagram	Manager
Sovereign society	Simple mechanical machines	March 2, 1757 (Foucault)	Centralization	Hierarchy
Disciplinary society	Thermodynamic machines	May 24, 1844 (telegraph); 1942 (Manhattan Project)	Decentralization	Bureaucracy
Control society	Cybernetic machines, computers	February 28, 1953 (Watson and Crick); January 1, 1983 (TCP/IP)	Distribution	Protocol

telephone switches on January 15, 1990; the start of the Gulf War on January 17, 1991.[50] These dates, plus the many periodization theories mentioned earlier, map together as shown in table 1.

That these dates do not line up in any precise manner is of no concern. Periodization theory is a loose art at best and must take into account that, when history changes, it changes slowly and in an overlapping, multilayered way, such that one historical moment may extend well into another, or two moments may happily coexist for decades or longer. For instance, in much of the last hundred years, *all three social phases described earlier existed at the same time* in the United States and elsewhere. To paraphrase William Gibson: The future is already here, but it is not uniformly distributed across all points in society. At best, periodization theory is an analytical mindgame, yet one that breathes life into the structural analyses offered to explain certain tectonic shifts in the foundations of social and political life. My book implicitly participates in this game, mapping out certain details of the third, "control society" phase, specifically the diagram of the distributed network, the technology of the computer, and the management style of protocol.

---

50. For Jencks, see his *The Language of Post-Modern Architecture* (New York: Rizzoli, 1991); for references to AT&T, see Bruce Sterling's *The Hacker Crackdown* (New York: Bantam, 1993) and Michelle Slatalla and Joshua Quittner's *Masters of Deception* (New York: HarperCollins, 1995).

1

Physical Media

The language of the RFC was warm and welcoming.
—KATIE HAFNER AND MATTHEW LYON, *Where Wizards Stay Up Late*

While many have debated the origins of the Internet, it's clear that in many ways it was built to withstand nuclear attack. The Net was designed as a solution to the vulnerability of the military's centralized system of command and control during the late 1950s and beyond. For, the argument goes, if there are no central command centers, then there can be no central targets and overall damage is reduced.

If one can consider nuclear attack as the most highly energetic, dominating, and centralized force that one knows—an archetype of the modern era—then the Net is at once the solution to and inversion of this massive material threat, for it is precisely noncentralized, nondominating, and nonhostile.

The term *protocol* is most known today in its military context, as a method of correct behavior under a given chain of command. On the Internet, the meaning of protocol is slightly different. In fact, the reason why the Internet would withstand nuclear attack is precisely because its internal protocols are the enemy of bureaucracy, of rigid hierarchy, and of centralization. As I show in this chapter, the material substrate of network protocols is highly flexible, distributed, and resistive of hierarchy.

The packet-switching technologies behind the Internet provided a very different "solution" to nuclear attack than did common military protocol during the Cold War. For example, in 1958 the Royal Canadian Air Force and the U.S. Air Force entered into agreement under the North American Aerospace Defense Command (NORAD). NORAD is a radar surveillance system ringing North America that provides early warnings of missile or other air attacks against Canada and the United States. "The command monitors any potential aerospace threat to the two nations, provides warning and assessment of that threat for the two governments, and responds defensively to any aircraft or cruise missile threatening North American airspace."[1] The NORAD system is a centralized, hierarchical network. It contains regional control sectors, all of which are ultimately controlled by the USSPACECOM Command Center at Cheyenne Mountain in Colorado Springs, Colorado. It functions like a wall, not like a meshwork. Faced with a nuclear attack,

---

Epigraph: Katie Hafner and Matthew Lyon, *Where Wizards Stay Up Late: The Origins of the Internet* (New York: Touchstone, 1996), p. 144.

1. *NORAD: Into the 21st Century,* U.S. Government Printing Office (1997-574-974).

NORAD meets force with force. Once the outer protection zone of the landmass is compromised, the NORAD command is able to scramble defensive air power through a rigidly defined system of command and control that is directed outward from a single source (USSPACECOM), to subservient endpoint installations that help resist attack. The specific location of each radar installation is crucial, as is the path of the chain of command. During the Cold War, NORAD was the lynchpin of nuclear defense in North America. It is a "solution" to the nuclear threat.

The Internet system could not be more different. It follows a contrary organizational design. The Internet is based not on directionality nor on toughness, but on flexibility and adaptability. Normal military protocol serves to hierarchize, to prioritize, while the newer network protocols of the Internet serve to *distribute.*

In this chapter I describe exactly what distribution means, and how protocol works in this new terrain of the distributed network.[2] I attempt to show that protocol is not by nature horizontal or vertical, but that protocol is an algorithm, *a proscription for structure* whose form of appearance may be any number of different diagrams or shapes.

The simplest network diagram is the centralized network (see figure 1.1). Centralized networks are hierarchical. They operate with a single authoritative hub. Each radial node, or branch of the hierarchy, is subordinate to the central hub. All activity travels from center to periphery. No peripheral node is connected to any other node. Centralized networks may have more than one branch extending out from the center, but at each level of the hierarchy power is wielded by the top over the bottom.

---

2. The division of network designs into centralized, decentralized, and distributed appears in Paul Baran's *On Distributed Communications: 1. Introduction to Distributed Communications Networks* (Santa Monica, CA: RAND, 1964), p. 2. Baran's diagrams have been copied by many authors since then.

Following William Evan, John Arquilla and David Ronfeldt suggest a topology even simpler than the centralized network. This is the chain or line network: for example, "in a smuggling chain where people, goods, or information move along a line of separate contacts, and where end-to-end communication must travel through the intermediate nodes." See Arquilla and Ronfeldt, *Networks and Netwars: The Future of Terror, Crime, and Militancy* (Santa Monica, CA: RAND, 2001), p. 7.

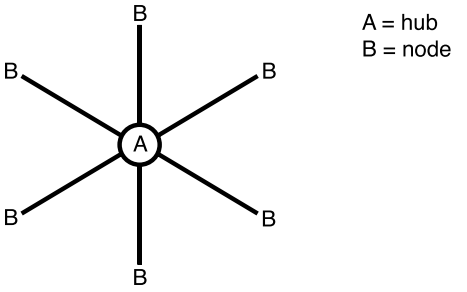


Figure 1.1
A centralized network

The American judicial system, for example, is a centralized network. While there are many levels to the court system, each with its own jurisdiction, each decision of each court can always be escalated (through the appeals process) to a higher level in the hierarchy. Ultimately, however, the Supreme Court has final say over all matters of law.

The panopticon, described in Foucault's *Discipline and Punish,* is also a centralized network. In the panopticon, repurposed by Foucault from the writings of Jeremy Bentham, a guard is situated at the center of many radial cells. Each cell contains a prisoner. This special relationship between guard and prisoner "links the centre and periphery." In it, "power is exercised without division, according to a continuous hierarchical figure" occupying the central hub.[3]

A *de*centralized network is a multiplication of the centralized network (see figure 1.2). In a decentralized network, instead of one hub there are many hubs, each with its own array of dependent nodes. While several hubs exist, each with its own domain, no single zenith point exercises control over all others.

There are many decentralized networks in the world today—in fact, decentralized networks *are the most common diagram of the modern era.*

One example is the airline system. In it, one must always travel through certain centralized hub cities—generally in the Midwest or central areas of the United States. Direct nonstop service is only possible if one happens to

3. Michel Foucault, *Discipline and Punish,* trans. Alan Sheridan (New York: Vintage, 1997), p. 197.

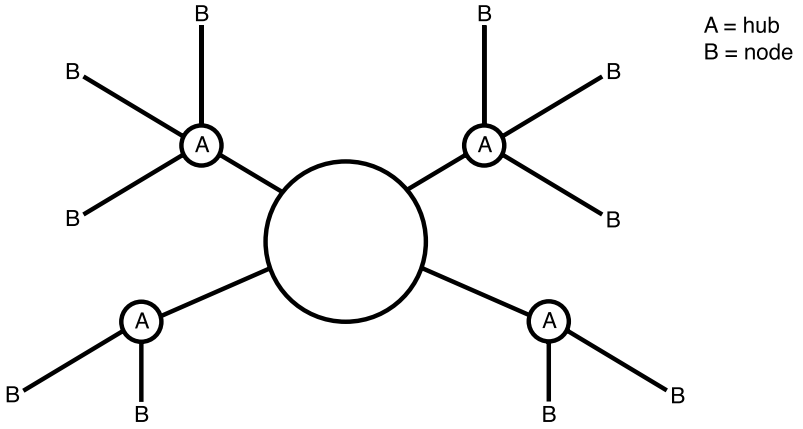


Figure 1.2
A decentralized network

be traveling from one hub to another (or if one pays a premium for special routes).

For the airline system, the decentralized network is the solution to multiplicity, albeit a compromise between the needs of the passenger and the needs of the airlines. There are far too many airports in the country to allow for nonstop service between each and every city; however, it would be inefficient to route every passenger through a single, Midwestern hub (e.g., consider a flight from North Carolina to Maine).

The third network diagram, the one that interests me most here, is called the distributed network.[4] The emergence of distributed networks is part of a larger shift in social life. The shift includes a movement away from central

4. In *Networks and Netwars,* Arquilla and Ronfeldt call this third network topology an "all-channel" network "where everybody is connected to everybody else" (p. 8). However their all-channel network is not identical to a distributed network, as their senatorial example betrays: "an all-channel council or directorate" (p. 8). Truly distributed networks cannot, in fact, support all-channel communication (a combinatorial utopia), but instead propagate through outages and uptimes alike, through miles of dark fiber (Lovink) and data oases, through hyperskilled capital and unskilled laity. Thus distribution is similar to but not synonymous with all-channel, the latter being a mathematical fantasy of the former.

bureaucracies and vertical hierarchies toward a broad network of autonomous social actors.

As Branden Hookway writes: "The shift is occurring across the spectrum of information technologies as we move from models of the global application of intelligence, with their universality and frictionless dispersal, to one of local applications, where intelligence is site-specific and fluid."[5] Computer scientists reference this historical shift when they describe the change from linear programming to *object-oriented* programming, the latter a less centralized and more modular way of writing code. This shift toward distribution has also been documented in such diverse texts as those of sociologist Manuel Castells, American Deleuzian Hakim Bey, and the Italian "autonomist" political movement of the 1970s. Even harsh critics of this shift, such as Nick Dyer-Witheford, surely admit that the shift is taking place. It is part of a larger process of postmodernization that is happening the world over.

What is the nature of these distributed networks? First, distributed networks have no central hubs and no radial nodes. Instead each entity in the distributed network is an autonomous agent.

A perfect example of a distributed network is the rhizome described in Deleuze and Guattari's *A Thousand Plateaus.* Reacting specifically to what they see as the totalitarianism inherent in centralized and even decentralized networks, Deleuze and Guattari instead describe the rhizome, a horizontal meshwork derived from botany. The rhizome links many autonomous nodes together in a manner that is neither linear nor hierarchical. Rhizomes are heterogeneous and connective, that is to say, "any point of a rhizome can be connected to anything other."[6] They are also multiple and asymmetrical: "[a] rhizome may be broken, shattered at a given spot, but it will start up again on one of its old lines, or on new lines."[7] Further, the rhizome has complete disregard for depth models, or procedures of derivation. As Deleuze and Guattari write, a rhizome "is a stranger to any idea of genetic axis

---

5. Branden Hookway, *Pandemonium: The Rise of Predatory Locales in the Postwar World* (New York: Princeton Architectural Press, 1999), pp. 23–24.

6. Gilles Deleuze and Félix Guattari, *A Thousand Plateaus,* trans. Brain Massumi (Minneapolis: University of Minnesota Press, 1987), p. 7.

7. Deleuze and Guattari, *A Thousand Plateaus,* p. 9.

or deep structure."[8] Trees and roots, and indeed "[a]ll of arborescent culture"[9] is rejected by the rhizome. Summarizing the unique characteristics of the rhizome—and with it the distributed network—Deleuze and Guattari write:

- [U]nlike trees or their roots, the rhizome connects any point to any other point . . .
- The rhizome is reducible neither to the One nor the multiple. . . . It is composed not of units but of dimensions, or rather directions in motion.
- It has neither beginning nor end, but always a middle (*milieu*) from which it grows and which it overspills.
- Unlike a structure, which is defined by a set of points and positions, with binary relations between the points and biunivocal relationships between the positions, the rhizome is made only of lines . . .
- Unlike the tree, the rhizome is not the object of reproduction . . .
- The rhizome is an antigenealogy. It is short-term memory, or antimemory.
- The rhizome operates by variation, expansion, conquest, capture, offshoots.
- The rhizome is an acentered, nonhierarchical, nonsignifying system without a General and without an organizing memory or central automation.[10]

If diagrammed, a distributed network might look like figure 1.3. In a distributed network, each node *may* connect to any other node (although there is no requirement that it does). During a node-to-node connection, no intermediary hubs are required—none, not even a centralized switch as is the case in the telephone network. Point "X" may contact "Y" directly via one of several path combinations.

A distributed network is always caught, to use an expression from Deleuze and Guattari, au milieu, meaning that it is never complete, or integral to itself. The lines of a distributed network continue off the diagram. Any subsegment of a distributed network is as large and as small as its parent network. Distribution propagates through rhythm, not rebirth.

---

8. Deleuze and Guattari, *A Thousand Plateaus,* p. 12.

9. Deleuze and Guattari, *A Thousand Plateaus,* p. 15.

10. Deleuze and Guattari, *A Thousand Plateaus,* p. 21, bulleted format mine.

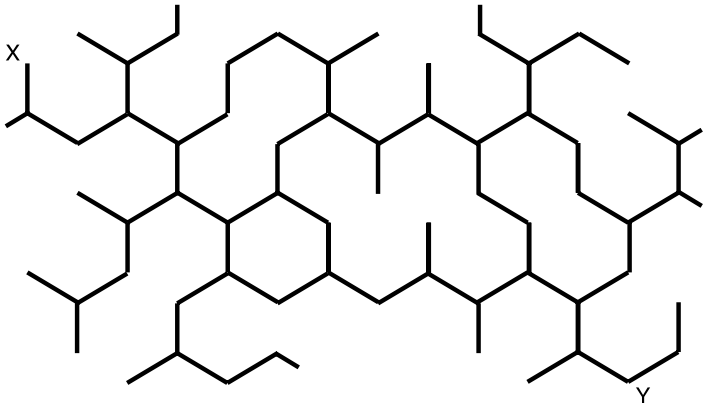


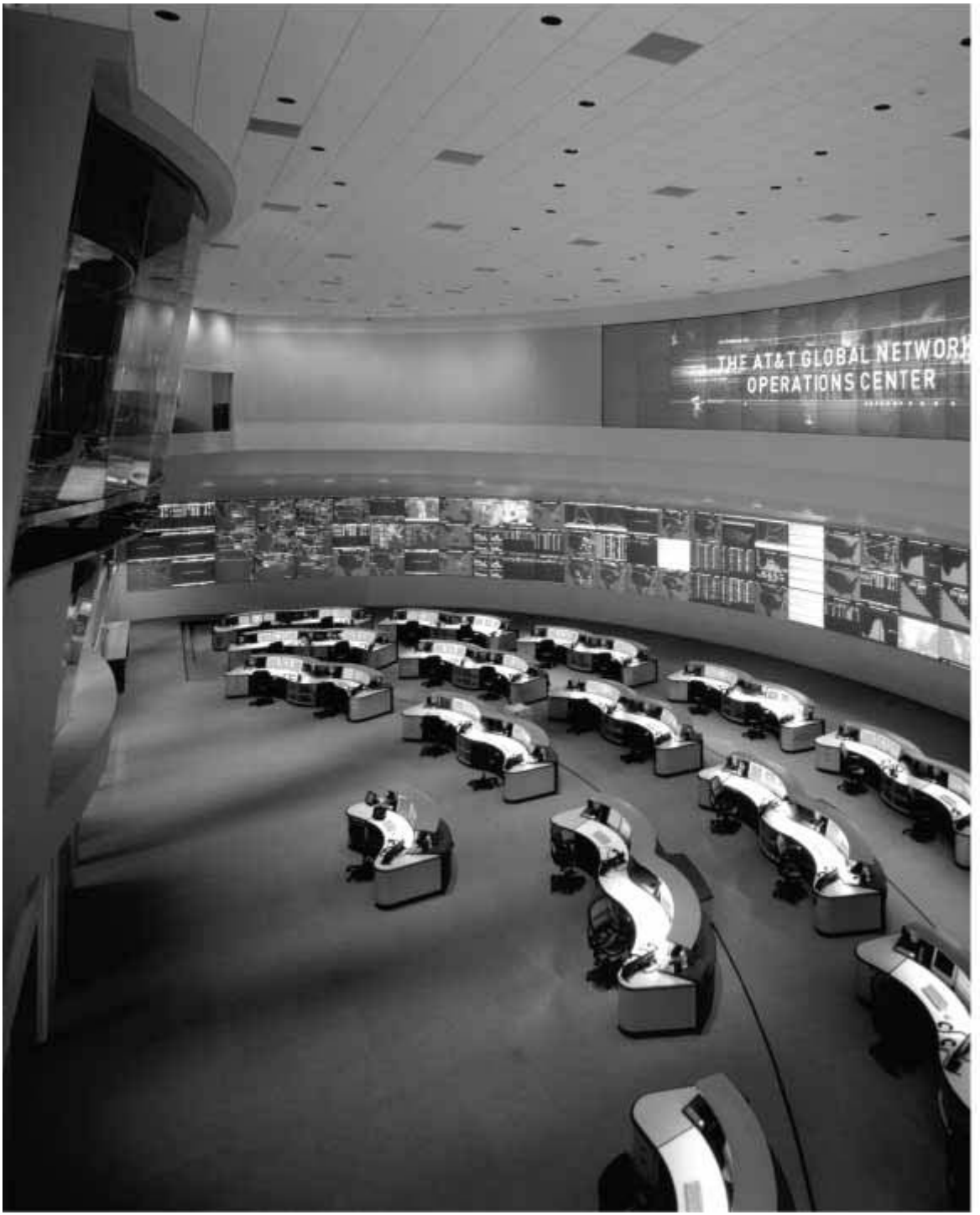
Figure 1.3
A distributed network

One actually existing distributed network is the Dwight D. Eisenhower System of Interstate & Defense Highways, better known as the interstate highway system. The highway system was first approved by Congress immediately following World War II, but was not officially begun until June 29, 1956, when President Eisenhower signed it into law. (This is exactly the same period during which Internet pioneer Paul Baran began experimenting with distributed, packet-switching computer technologies at the Rand Corporation.[11]) The highway system is a distributed network because it lacks any centralized hubs and offers direct linkages from city to city through a variety of highway combinations.

For example, someone traveling from Los Angeles to Denver may begin by traveling on Interstate 5 north toward San Francisco turning northwest on Interstate 80, or head out on Interstate 15 toward Las Vegas, or even Interstate 40 toward Albuquerque. The routes are varied, not predetermined. If one route is blocked, another will do just as well. These are the advantages of a distributed network.

---

11. As Hafner and Lyon write: "Baran was working on the problem of how to build communication structures whose surviving components could continue to function as a cohesive entity after other pieces were destroyed." See Katie Hafner and Matthew Lyon, *Where Wizards Stay Up Late,* p. 56.



AT&T Global Network Operation Center (architect: HOK; photo: Peter Paige)

ntr i tion D ntr i tion

A centralized network consists of a single central power point (a host), from which are attached radial nodes. The central point is connected to all of the satellite nodes which are themselves connected only to the central host. A decentralized network, on the other hand, has *i e* central hosts, each with its own set of satellite nodes. A satellite node may have connectivity with one or more hosts, but not with other nodes. Communication generally travels unidirectionally within both centralized and decentralized networks: from the central trunks to the radial leaves.

Of course the Internet is another popular and actually existing distributed network. Both the Internet and the U.S. interstate highway system were developed in roughly the same time period (from the late 1950s to the late 1970s), for roughly the same reason (to facilitate mobility and communication in case of war). Later, they both matured into highly useful tools for civilians.

What was once protocol's primary liability in its former military context—the autonomous agent who does not listen to the chain of command—is now its primary constituent in the civil context. The diagram for protocol has shifted from the centralized to the decentralized network, and now finally to the distributed network. Distributed networks have no chain of command, only autonomous agents who operated according to certain pre-agreed "scientific" rules of the system.

For the Internet, these scientific rules are written down. Called protocols, they are available in documents known as RFCs, or "Requests for Comments." Each RFC acts as a blueprint for a specific protocol. It instructs potential software designers and other computer scientists how to correctly implement each protocol in the real world. Far more than mere technical documentation, however, the RFCs are a discursive treasure trove for the critical theorist.

The RFC on "Requirements for Internet Hosts," an introductory document, defines the Internet as a series of interconnected networks, that is, a *network of networks,* that are interconnected via numerous interfacing computers called gateways: "An Internet communication system consists of interconnected packet networks supporting communication among host computers using the Internet protocols . . . The networks are interconnected using packet-switching computers called 'gateways.'"[12] Populating these many different networks are hosts, single computers that are able to send and receive information over the network. According to this RFC, "A host computer, or simply 'host,' is the ultimate consumer of communication services. A host generally executes application programs on behalf of user(s), employing network and/or Internet communication services in support of this function. . . . Internet hosts span a wide range of size, speed, and function.

12. Robert Braden, "Requirements for Internet Hosts," RFC 1122, October 1989, p. 6.

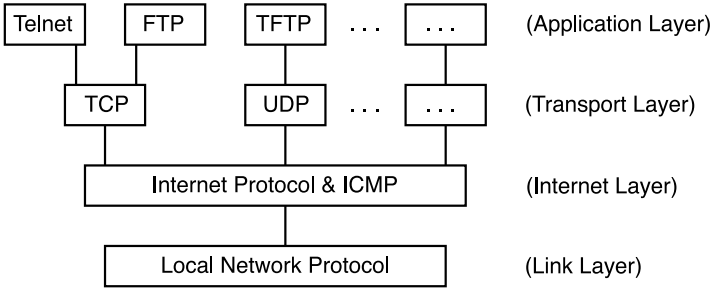


Figure 1.4
Protocol layers

They range in size from small microprocessors through workstations to mainframes and supercomputers."[13] Or, as the RFC on Transmission Control Protocol simply defines it, hosts are "computers attached to a network."[14] If the host is a receiver of information, it is called a client. If it is a sender of information, it is called a server.

In order for hosts to communicate via the Internet, they must implement an entire suite of different protocols. Protocols are the common languages that all computers on the network speak. These component protocols act like layers. Each layer has a different function (see figure 1.4). Considered as a whole, the layers allow communication to happen.

The RFC on "Requirements for Internet Hosts" defines four basic layers for the Internet suite of protocols: (1) the application layer (e.g., telnet, the Web), (2) the transport layer (e.g., TCP), (3) the Internet layer (e.g., IP), and (4) the link (or media-access) layer (e.g., Ethernet).

These layers are nested, meaning that the application layer is encapsulated within the transport layer, which is encapsulated with the Internet layer, and so on.

This diagram, minus its "layer" captions, appears in RFC 791. The four layers are part of a larger, seven-layer model called the OSI (Open Systems Interconnection) Reference Model developed by the International Organization for Standardization (ISO). Tim Berners-Lee, inventor of the Web, uses a

13. Braden, "Requirements for Internet Hosts," pp. 6–7.

14. Jonathan Postel, "Transmission Control Protocol," RFC 793, September 1981, p. 7.

slightly different four-layer model consisting of "the transmission medium, the computer hardware, the software, and the content." Yochai Benkler, from whom Lawrence Lessig has drawn, uses instead a three-layer model consisting of a physical layer, a code layer, and a content layer. Lev Manovich uses an even simpler, two-layer model consisting of a "cultural" layer comprised of "the encyclopedia and the short story; story and plot; composition and point of view; mimesis and catharsis; comedy and tragedy," and a "computer" layer comprised of computer languages, variables, functions, packets, and other code elements.[15]

Consider an average telephone conversation as an analogy. There are several protocols at play during a telephone call. Some are technical, some social. For example, the act of listening for a dial tone and dialing the desired phone number can be considered to be in a different "layer" than the conversation itself.

Furthermore, the perfunctory statements that open and close a telephone conversation—"Hello," "Hi, this is . . . ," "Well, I'll talk to you later," "Okay, goodbye," "Bye!"—are themselves not part of the normal conversation "layer" but are merely necessary to establish the beginning and end of the conversation.

The Internet works the same way. The application layer is like the conversation layer of the telephone call. It is responsible for the content of the specific technology in question, be it checking one's email, or accessing a Web page. The application layer is a *semantic* layer, meaning that it is responsible for preserving the content of data within the network transaction. The application layer has no concern for larger problems such as establishing net-

---

15. For these references, see Jonathan Postel, "Internet Protocol," RFC 791, September 1981, p. 5; Tim Berners-Lee, *Weaving the Web* (New York: HarperCollins, 1999), pp. 129–130; Yochai Benkler's "From Consumers to Users: Shifting the Deeper Structures of Regulation Toward Sustainable Commons and User Access," *Federal Communications Law Journal* 52 (2000), pp. 561–579; and Lev Manovich, *The Language of New Media* (Cambridge: MIT Press, 2001), p. 46. The critical distinction is that the OSI model, my preferred heuristic, considers everything to be code and makes no allowances for special anthropomorphic uses of data. This makes it much easier to think about protocol. The other models privilege human-legible forms, whose reducibility to protocol is flimsy at best.

work connections, or actually sending data between those connections. It simply wants its "conversation" to work correctly.

The transport layer is one step higher in the hierarchy than the application layer. It has no concern for the content of information (one's email, one's Web page). Instead, the transport layer is responsible for making sure that the data traveling across the network arrives at its destination correctly. It is a social layer, meaning that it sits halfway between the content or meaning of the data being transferred and the raw act of transferring that data. If data is lost in transit, it is the transport layer's responsibility to resend the lost data.

Thus, in our hypothetical telephone conversation, if one hears static on the line, one might interject the comment, "Hello . . . Are you still there?" This comment is *not* part of the conversation layer (unless your conversation happens to be about "still being there"); rather, it is an interstitial comment meant to confirm that the conversation is traveling correctly across the telephone line. The opener and closer comments are also part of the transport layer. They confirm that the call has been established and that it is ready for the conversation layer, and conversely that the conversation is finished and the call will be completed.

The third layer is the Internet layer. This layer is larger still than both the application and transport layers. The Internet layer is concerned with one thing: the actual movement of data from one place to another. It has no interest in the content of that data (the application layer's responsibility) or whether parts of the data are lost in transit (the transport layer's responsibility).

The fourth layer, the link layer, is less important to my study. It is the hardware-specific layer that must ultimately encapsulate any data transfer. Link layers are highly variable due to the many differences in hardware and other physical media. For example, a telephone conversation can travel just as easily over normal telephone wire as it can over fiber-optic cable. However, in each case the technology in question is radically different. These technology-specific protocols are the concern of the link (or media-access) layer.

The different responsibilities of the different protocol layers allow the Internet to work effectively. For example, the division of labor between the transport layer and the Internet layer, whereby error correction is the sole responsibility of the transport layer and routing (the process by which data is "routed," or sent toward its final destination) is the sole responsibility of the Internet layer, creates the conditions of existence for the distributed network.

Thus, if a router goes down in Chicago while a message is en route from New York to Seattle, the lost data can be resent via Louisville instead (or Toronto, or Kansas City, or Lansing, or myriad other nodes). It matters not if the alternate node is smaller or larger, or is on a different subnetwork, or is in another country, or uses a different operating system.

The RFCs state this quality of flexibility very clearly:

> A basic objective of the Internet design is to tolerate a wide range of network characteristics—e.g., bandwidth, delay, packet loss, packet reordering, and maximum packet size. Another objective is robustness against failure of individual networks, gateways, and hosts, using whatever bandwidth is still available. Finally, the goal is full "open system interconnection": an Internet host must be able to interoperate robustly and effectively with any other Internet host, across diverse Internet paths.[16]

As long as the hosts on the network conform to the general suite of Internet protocols—like a lingua franca for computers—then the transport and Internet layers, working in concert, will take care of everything.

The ultimate goal of the Internet protocols is totality. The virtues of the Internet are robustness, contingency, interoperability, flexibility, heterogeneity, pantheism. Accept everything, no matter what source, sender, or destination.

TCP is the most common protocol in the transport layer. It works very closely with the IP to ensure that the data sent via IP arrives correctly. TCP creates a "virtual circuit" between sender and recipient and uses that imaginary circuit to regulate the flow of information. Where IP is blind to the ultimate integrity of the data it transports (more on IP later), TCP constantly checks to see if the message arrives in one piece. As the RFC specifies, "TCP is used by those applications needing reliable, connection-oriented transport service, e.g., mail (SMTP), file transfer (FTP), and virtual terminal service (Telnet)."[17]

TCP is responsible for the "handshake" that happens between two computers at the moment a connection is established.

---

16. Braden, "Requirements for Internet Hosts," p. 8.

17. Braden, "Requirements for Internet Hosts," p. 82.

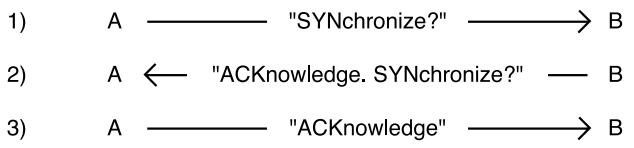


Figure 1.5
Three-way handshake

TCP creates an imaginary circuit between sender and receiver. It "saves state"; that is, it remembers the state of the conversation from moment to moment (something that IP does not do by itself, nor does the other common transport protocol called UDP). This is what the RFC refers to when it describes TCP as "a connection-oriented, end-to-end reliable protocol,"[18] as an example of ongoing "inter-process communication," or as the creation of a "logical circuit" between two computers. The circuit doesn't in fact exist in the real world, but it is created temporarily to connect sender and receiver, in much the same way that a circuit is temporarily created between caller and recipient during a normal telephone conversation (except that with the phone system, the circuit is created by an actual switch, rather than through a distributed connection).

The TCP circuit is created through a three-step process known as a handshake. First, the sender sends a message called a "SYN" (synchronize). Second, the recipient replies with a message called an "ACK" (acknowledge) and initiates its own SYN request. Finally, the original sender acknowledges the recipient's SYN by sending its own ACK (see figure 1.5). After this three-way handshake is complete—(1) "Hello!" (2) "Hi. How are you?" (3) "I'm fine thanks"—the connection is established and normal communication may begin.

The primary value of TCP is its robust quality. TCP allows communication on the Web to be very reliable: Information is monitored during transport and is re-sent if lost or corrupted.

As a system this robustness is achieved by following a general principle: "Be conservative in what you do, be liberal in what you accept from others."[19]

---

18. Postel, "Transmission Control Protocol," p. 1.

19. Postel, "Transmission Control Protocol," p. 14.

This means that TCP hosts should liberally accept as much information as possible from other, foreign devices. But if any of the information is corrupted, the "conservative" host will delete the information and request a fresh copy be re-sent. As the RFC notes, the goal of TCP is "robustness in the presence of communication unreliability and availability in the presence of congestion."[20]

TCP's partner protocol is IP. TCP and IP work together to create a protocol suite, referred to simply as TCP/IP. IP is responsible for one thing: moving small packets of data called "datagrams" from one place to another. As the RFC specifications for IP note, "the internet protocol provides for transmitting blocks of data called datagrams from sources to destinations."[21]

However, in IP there are "no mechanisms to augment end-to-end data reliability, flow control, sequencing, or other services commonly found in host-to-host protocols"[22] such as TCP. This means that IP simply seals up its datagrams and shoots them out into the ether. It does not wait for any SYNs or ACKs, and it receives no certification that the datagrams have been received (since these are all the responsibilities of the transport layer, TCP). The IP knows that, eventually, its datagrams will arrive at their locations, and if they don't, the transport layer will provide all error correction and send requests for the missing datagrams to be re-sent.

IP is like the engine powering a car—the engine moves the car, but it has no faculties for knowing when and where to steer, or knowing when and where to stop or speed up (these are the responsibilities of the driver). The engine cannot recognize the different between a green and red traffic light. It has no business dealing with things that are outside its protocological purview.

Technically, then, IP is responsible for two things: routing and fragmentation. Routing is the process by which paths are selected for moving data across a network. Since networks are heterogeneous and ever-changing, the route between point A and point B is never fixed but must be rethought each time material wishes to pass over it.

---

20. Postel, "Transmission Control Protocol," p. 1.

21. Postel, "Internet Protocol," p. 1.

22. Postel, "Internet Protocol," p. 1.

This flexible routing system is achieved through a "hopping" process whereby data is passed from computer to computer in sequence. None of the computers in the chain of hops knows definitively where the desired destination lies. But they do know in which general direction the destination is. They pass their datagrams to the computer that lies in the "general direction" of the destination. Each computer en route maintains a cache containing information about which of its neighbors lie in which general direction. Each node in the network knows not where the final destination is, but simply which direction, or "next-hop," will get it closer to its destination. If the next-hop proves to be faulty, then the intermediary gateway alerts the source computer and the source computer updates its next-hop cache.

Thus, if Chicago is the next-hop for a message leaving New York en route to Seattle, and Chicago goes down, then Louisville becomes New York's next-hop for Seattle. Later, if Chicago is reinstated and becomes the best routing option again, New York updates its cache accordingly.

The next-hop strategy means that no single node on the Internet knows definitively where a destination is, merely that it is "over there." Each node does know the exact location of every node *it is connected to,* and may pass its messages to whatever machine is closest to "over there." After enough hops in the right direction, the destination machine will no longer be "over there" but will actually be the next-hop for the router currently carrying the data, and the data will be delivered. In this way the message hops around until it arrives in the immediate vicinity of its destination, whereby the exact location of the destination is in fact known and final delivery is possible.

Each datagram is given a number called a "time-to-live." This number designates the maximum number of hops that that datagram is able to take before it is deleted. At each hop, the time-to-live is decreased by one. If the time-to-live reaches zero, the routing computer is obligated to delete the datagram. This ensures that datagrams will not hop around the network indefinitely, creating excess congestion.

The second responsibility of the Internet Protocol is fragmentation. When messages are sent across the network, they are inevitably too large to be sent in one piece. Hence, each message is fragmented, or disintegrated into several small packets, before it is sent. Each small packet is sent over the network individually. At the end, the packets are collected and reassembled to recreate the original message. This process is called fragmentation.

Each physical network has its own personalized threshold for the largest packet size it can accommodate. Thus, no single fragmentation recipe will work for every network. Some, like large freeways, will accommodate large packets, while others, like small back roads, will accommodate only small packets.

But if a message starts its journey as large packets, it cannot be stymied mid-journey if it happens to come upon a foreign network that only accommodates small packet sizes. Refragmentation may be necessary en route. Thus, if a message starts off being fragmented into large packets (e.g., if it is traveling over a fiber-optic cable), it may need to refragment itself mid-journey if it encounters a medium-sized pipe (e.g., a telephone line) somewhere en route. IP can deal with this contingency. Fragmentation allows the message to be flexible enough to fit through a wide range of networks with different thresholds for packet size.

Whenever a packet is created via fragmentation, certain precautions must be taken to make sure that it will be reassembled correctly at its destination. To this end, a header is attached to each packet. The header contains certain pieces of vital information such as its source address and destination address. A mathematical algorithm or "checksum" is also computed and amended to the header. If the destination computer determines that the information in the header is corrupted in any way (e.g., if the checksum does not correctly correlate), it is obligated to delete the packet and request that a fresh one be sent.

At this point, let me pause to summarize the distinct protocological characteristics of the TCP/IP suite:

- TCP/IP facilitates peer-to-peer communication, meaning that Internet hosts can communicate directly with each other without their communication being buffered by an intermediary hub.
- TCP/IP is a distributed technology, meaning that its structure resembles a meshwork or rhizome.
- TCP/IP is a universal language, which if spoken by two computers allows for internetworking between those computers.
- The TCP/IP suite is robust and flexible, not rigid or tough.
- The TCP/IP suite is open to a broad, theoretically unlimited variety of computers in many different locations.

- The TCP/IP protocol, and other protocols like it, is a *result* of the action of autonomous agents (computers).

Each of these characteristics alone is enough to distinguish protocol from many previous modes of social and technical organization. Together they compose a new, sophisticated system of distributed control.

Not every protocol is concerned with the process of peer-to-peer communication as are TCP and IP. DNS, or Domain Name System, is a protocol with a very simple, but different, mandate. DNS is responsible for translating Internet addresses from names to numbers.

While many computer users are familiar with the "dot-com" style of writing Internet addresses (e.g., www.superbad.com or www.rhizome.org), computers themselves use a numerical moniker instead, called an IP address. IP addresses are written as a group of four numbers separated by dots (e.g., 206.252.131.211). While it is very difficult for humans to remember and use such numbers, it is very easy for computers. "The basic problem at hand," writes DNS critic Ted Byfield, is "how we map the 'humanized' names of DNS to 'machinic' numbers of the underlying IP address system."[23] Computers understand numbers more easily, humans understand words. Thus, before each and every transaction on the World Wide Web, one's hand-typed Web address must first be translated to an IP address before the computer can do its work:

```
www.rhizome.org ↔ 206.252.131.211
```

This translation is called "resolution" and it is the reason why DNS exists. If DNS had never been developed, Internet addresses would look more like long telephone numbers or postal codes. Instead they look like long words.

Prior to the introduction of DNS in 1984, a single computer, called a *name server,* held all the name-to-number conversions. They were contained in a single text file. There was one column for all the names and another for all the numbers—like a simple reference table. This document, called HOSTS.TXT,

---

23. Ted Byfield, "DNS: A Short History and a Short Future," *Nettime,* October 13, 1998.

lived in Menlo Park, California, at the Network Information Center of the Stanford Research Institute (SRI-NIC).[24] Other computers on the Internet would consult this document periodically, downloading its information so that their local reference tables would carry the most up-to-date data. The entire system of naming referred to in this file was called the *name space.*

This early system was a centralized network, par excellence, with SRI-NIC at the center. However as the Internet grew larger this single, central node became incompatible with the nature of the network: "The toll on SRI-NIC, in terms of the network traffic and processor load involved in distributing the file, was becoming unbearable. . . . Maintaining consistency of the files across an expanding network became harder and harder. By the time a new HOSTS.TXT could reach the farthest shores of the enlarged ARPAnet, a host across the network had changed addresses, or a new host had sprung up that users wanted to reach."[25]

To solve this problem, computer scientist Paul Mockapetris designed a new system, a decentralized database of name/number mappings called DNS (see figure 1.6). The new system, still in place today, operates like an inverted tree:

> The domain name space is a tree structure. Each node and leaf on the tree corresponds to a resource set (which may be empty). . . . The domain name of a node or leaf is the path from the root of the tree to the node or leaf. By convention, the labels that compose a domain name are read left to right, from the most specific (lowest) to the least specific (highest).[26]

The tree structure allows Mockapetris to divide the total name space database into more manageable and decentralized zones through a process of hierarchization. As Mockapetris writes, "approaches that attempt to collect a consistent copy of the entire database will become more and more expensive

---

24. See Paul Albitz and Cricket Liu, *DNS and BIND, Third Edition* (Sebastopol, CA: O'Reilly, 1998), p. 3.

25. Albitz and Liu, *DNS and BIND,* pp. 3–4.

26. Paul Mockapetris, "Domain Names—Concepts and Facilities," RFC 882, November 1983, p. 6.

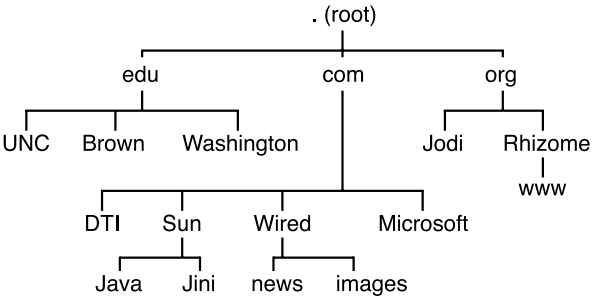


Figure 1.6
Domain Name System (DNS)

and difficult, and hence should be avoided."[27] Instead each portion of the database is delegated outward on the branches of the tree, into each leaf.

At the top of the inverted tree sit the so-called root servers, represented by a single dot (".") They have authority over the *top-level domains* (TLDs) such as "com," "net," "edu," and "org." At each branch of the tree, control over a different zone of the name space is delegated to a server that is lower on the tree. Thus, in order to resolve the address "www.rhizome.org," one must first ask the root server where to find the "org" zone. The root server replies with an authoritative answer about where to find the "org" name server. Then, the "org" name server is queried and replies with the answer for where to find the "rhizome" host within the "org" zone. Finally, the "rhizome" name server is queried, and replies with the numerical address for the "www" computer that lives within the "rhizome" domain.

Like this, the process starts at the most general point, then follows the chain of delegated authority until the end of the line is reached and the numerical address may be obtained. This is the protocol of a decentralized network.

In DNS, each name server can reply only with authoritative information about the zone immediately below it. This is why the system is hierarchical. But each name server can *only* know authoritative information about the zone immediately below it. The second, or third, or even fourth segment down the branch has been delegated to other name servers. This is why the system is decentralized.

---

27. Mockapetris, "Domain Names—Concepts and Facilities," p. 2.

The more central name servers that are closer to the root of the tree cannot tell you authoritative information about the computers at the ends of the branches, but they *can* tell you who they have delegated such information to and where to find the delegates.

As mentioned in the introduction to this book, protocol is based on a contradiction between two opposing machinic technologies: One radically distributes control into autonomous locales (exemplified here by TCP and IP), and the other focuses control into rigidly defined hierarchies (exemplified here by DNS). There are other important conclusions that one may derive from the preceding discussion of protocol.

First, as the discussion of DNS suggests, protocol is a universalizing system. Ted Byfield writes that what is unique to the DNS is

> its historical position as the first "universal" addressing system—that is, a naming convention called upon . . . to integrate not just geographical references at every scale . . . but also commercial language of every type (company names, trademarks, jingles, acronyms, services, commodities), proper names (groups, individuals), historical references (famous battles, movements, books, songs), hobbies and interests, categories and standards (concepts, specifications, proposals) . . . the list goes on and on.[28]

DNS is the most heroic of human projects; it is the actual construction of a single, exhaustive index for all things. It is the encyclopedia of mankind, a map that has a one-to-one relationship with its territory. Thus, as I demonstrate in chapter 2, DNS is like many other protocols in that, in its mad dash toward universality, it produces sameness or consistency where originally there existed arbitrariness. As the saying goes, apples and oranges are not comparable in the "real world," but in the DNS system they are separated by a few binary digits. DNS is not simply a translation language, *it is language.* It governs meaning by mandating that anything meaningful must register and appear somewhere in its system. This is the nature of protocol.

Second, as the discussion of TCP/IP shows, protocol is materially immanent. That is, protocol does not follow a model of command and control that places the commanding agent outside of that which is being commanded. It

---

28. Byfield, "DNS."

is endogenous. (This is a departure from the more hierarchical definition of protocol used by the military where control is exercised from without.)

For example, the protocological manipulation of an HTML object by an HTTP object begins first with the parsing of the HTML object:

```
<html>
<body>
Hello World!
</body>
</html>
```

The creation of a special HTTP *header* that derives from the original object is attached to the beginning of it and describes it in various ways:

```
HTTP/1.1 200 OK
Date: Sun, 28 Jan 2001 20:51:58 GMT
Server: Apache/1.3.12 (Unix)
Connection: close
Content-Type: text/html

<html>
<body>
Hello World!
</body>
</html>
```

The header contains various pieces of information about the HTML object such as the date the file was last modified (line 2), the make and model of the server offering the file (line 3), and the type of content it is (in this case, it is text-based HTML [line 5]).

The HTTP object, then, is simply the HTML object plus its new HTTP header, all wrapped up into a new form and separated by a blank line. The new header is prefixed to the original content, becoming part of its material body. But, since the HTTP header is nothing but a description of the material contents of the HTML object, the larger protocol (HTTP) is simply a way of rewriting the smaller one (HTML)—the smaller data object is encapsulated by the larger one. In doing so, the HTML object is immanently

transformed—*its actual data is prefixed by another unit of data*—to function within the larger context of HTTP.

Another conclusion is that, while protocol is immanent to a particular set of data, *protocological objects never contain their own protocol.* Thus, TCP/IP houses HTTP, which houses HTML, which houses ASCII text, etc. New headers are added at each level, but in terms of content, protocols are never continuous with themselves.

At each phase shift (i.e., the shift from HTML to HTTP, or from HTTP to TCP), one is able to identify a data object from the intersection of two articulated protocols. In fact, since digital information is nothing but an undifferentiated soup of ones and zeros, data objects *are nothing* but the arbitrary drawing of boundaries that appear at the threshold of two articulated protocols.[29] In order to see HTML, one must actually view it as it intersects with HTTP. Otherwise, one looks at HTML and sees nothing but its own internal protocols: text and markup tags.

A last point, something that should be of particular interest to critical theorists, is that protocol is *against interpretation.* This is to say that protocol does little to transcode the meaning of the semantic units of value that pass in and out of its purview. It encodes and decodes these values, yes, but such transformations are simply trivial mathematics and do not affect meaning in the same way that a Hollywood film may affect the meaning of femininity, or a police officer walking the beat may affect the meaning of power in public space. Protocols do not perform any interpretation themselves; that is, they encapsulate information inside various wrappers, while remaining relatively indifferent to the content of information contained within.

The consequences of this are legion. It means that protocological analysis must focus not on the sciences of meaning (representation/interpretation/reading), but rather on the sciences of possibility (physics or logic), which I address in more detail in chapter 5 on hacking.

The limits of a protocological system and the limits of *possibility* within that system are synonymous.

---

29. This is similar to Manovich's principle of "modularity" in which every new media object is made up of independent parts, which themselves are unique independent objects. It is, in a sense, objects all the way down. See Lev Manovich, *The Language of New Media,* pp. 30–31.

To follow a protocol means that everything possible within that protocol is already at one's fingertips. Not to follow means no possibility. Thus, protocological analysis must focus on the possible and the impossible (the envelope of possibility), not a demystification of some inner meaning or "rational kernel" within technology. *Protocol is a circuit, not a sentence.*

In this chapter on physical media I have tried to describe protocol from the perspective of its real material substrate. I described the distributed network and positioned protocol as a unique governing principle within that network. I highlighted the TCP/IP suite of Internet protocols and DNS as the two most important theoretical moments for protocol—one protocol radically distributes control into autonomous agents, the other rigidly organizes control into a tree-like decentralized database.

Next, I move beyond the hard science of protocol and begin to consider it from the perspective of form. That is: How does protocol function, not as a material machine, but as an entire formal apparatus? What techniques are used by and through protocol to create various cultural objects? How can one define protocol in its most abstract sense?

These are the fundamental questions contained in chapter 2 on form, to which I now turn.