Tussle in Cyberspace: Defining Tomorrow's Internet

David D. Clark, *Fellow, IEEE*, John Wroclawski, *Member, IEEE*, Karen R. Sollins, *Member, IEEE*, and Robert Braden, *Member, IEEE*

***Abstract*—The architecture of the Internet is based on a number of principles, including the self-describing datagram packet, the end-to-end arguments, diversity in technology and global addressing. As the Internet has moved from a research curiosity to a recognized component of mainstream society, new requirements have emerged that suggest new design principles, and perhaps suggest that we revisit some old ones. This paper explores one important reality that surrounds the Internet today: different stakeholders that are part of the Internet milieu have interests that may be adverse to each other, and these parties each vie to favor their particular interests. We call this process "the tussle." Our position is that accommodating this tussle is crucial to the evolution of the network's technical architecture. We discuss some examples of tussle, and offer some technical design principles that take it into account.**

***Index Terms*—Competition, design principles, economics, network architecture, trust, tussle.**

I. Introduction

THE Internet was created in simpler times. Its creators and early users shared a common goal—they wanted to build a network infrastructure to hook all the computers in the world together so that as yet unknown applications could be invented to run there. All the players, whether designers, users or operators, shared a consistent vision and a common sense of purpose.

Perhaps the most important consequence of the Internet's success is that the common purpose that launched and nurtured it no longer prevails. There are, and have been for some time, important and powerful players that make up the Internet milieu with interests directly at odds with each other. The Internet is not a single happy family of people dedicated to universal packet carriage. There is contention among the players.

At a minimum these players include users, who want to run applications and interact over the Internet; commercial ISPs, who sell Internet service with the goal of profit; private sector network providers who run a part of the Internet to facilitate their business or other undertaking; governments, who enforce laws, protect consumers, regulate commerce, and so on; intellectual property rights holders, who want to protect their materials on the Internet; and providers of content and higher level services, offered in search of profit or as a public service. The list of stakeholders probably mirrors every aspect of society.[1]

Some examples of contention are very current. Music lovers of a certain bent want to exchange recordings with each other, but the rights holders want to stop them. People want to talk in private, and the government wants to tap their conversations. Conservative governments and corporations put their users behind firewalls, and the users route and tunnel around them. ISPs give their users a single IP address, and users attach a network of computers using address translation. Some examples are so obvious that they are almost overlooked. For the Internet to provide universal interconnection, ISPs must interconnect, but ISPs are sometimes fierce competitors. It is not at all clear what interests are being served, to whose advantage, to what degree, when ISPs negotiate terms of connection.

We use the word "tussle" to describe the ongoing contention among parties with conflicting interests. Different parties adapt a mix of mechanisms to try to achieve their conflicting goals, and others respond by adapting the mechanisms to push back. The Internet, like society in the large, is shaped by controlled tussle, regulated not just by technical mechanism but by mechanisms such as laws, judges, societal opinion, shared values, and the like. There is no "final outcome" of these interactions, no stable point, and no acquiescence to a static architectural model. Today, the Internet is more and more defined by these tussles.

In earlier days, designers might have hoped that the Internet was defined by its technical specification. Engineers attempt to solve problems by designing mechanisms with predictable consequences. Successful engineering yields bridges that predictably do not fall down, planes that predictably do not fall out of the sky, and calculators that give the "right" answer. The essence of engineering is the development and codification of models, techniques and tools that deliver predictable, desirable behavior. But if the reality of the Internet is that there is no final outcome, no predictable result of our design processes, we must recognize the need to think about design differently.

We suggest that the reality of tussle implies the need for network designers to think explicitly about tussle and the design requirements it implies. As a computer science discipline, we focus on design principles that deliver such virtues as performance, robustness, scalability and manageability in the face of complexity, component failures, growth, and other challenges. We need to think about tussle in the same way: as an important and central aspect of design. As we do so, we may come to recognize design strategies driven by the growing tussle among and between different Internet players.

[1]For a detailed discussion of these various players and their impact on the Internet, see [1].

Manuscript received October 7, 2002; revised June 4, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Rexford. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0553 at MIT and agreement number F30602-00-1-0540 at ISI. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation hereon.

D. D. Clark, J. Wroclawski, and K. R. Sollins are with the MIT Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139 USA (e-mail: ddc@csail.mit.edu).

R. Braden is with the University of Southern California Information Sciences Institute, Marina del Rey, CA 90292 USA.

Digital Object Identifier 10.1109/TNET.2005.850224

The challenge facing Internet research and engineering is to recognize and *leverage* this reality—at minimum to accommodate it; if possible, to use it to *strengthen* the technical architecture. In other words, the technical architecture must accommodate the tussles of society while continuing to achieve its traditional goals of scalability, reliability, and evolvability. This expansion of the Internet's architectural goals is a difficult, but central technical problem.

We begin by briefly discussing the nature of tussle—what we can learn from a variety of disciplines, and in particular the role of technology in tussle. We then outline some proposed design principles, that explicitly acknowledge the role of tussle in the system, and recognize the need to accommodate it. We conclude by discussing some tussle spaces, ways in which our principles might guide the technical response to these spaces, and specific technical research that may be of value in accommodating these tussles.

II. What Is New About Tussle?

Any practicing engineer knows that the process of design is not a simple one of being handed a clear specification and building to it. Rather, the design process is one of balancing considerations and resolving tensions to get an acceptable specification for the device being designed. A number of engineering case studies that emphasize the importance of this process are captured by Vinck [2]. An important characteristic of these studies is that in most cases, there is in fact an eventual specification that defines the device—in other words here *the tussle occurs at design time*.

A more sophisticated version of this characterization is that many artifacts are repetitively designed. A car manufacturer makes a new car model each year, and the design evolves as a result of tussle; concerns about reducing the cost of production, user feedback and government regulation are all folded into the next redesign. So *tussle occurs at redesign time*, but only in extreme cases are existing cars recalled and refitted with new designs.

Many computer systems are sold with the idea that they will be configured by the eventual user in major ways at the time they are put to use. In this case, much tussle may occur at *configuration time*.

The cases studied in Vinck are mostly concerned with the design of *components*. The Internet is a *system*, and issues of systems engineering are perhaps more relevant. Hughes [3] contains case studies of large systems projects, ranging from Sage and the ARPAnet to the reconstruction of Boston's central roadway system, locally called the "Big Dig." These stories, especially that of the Big Dig, make clear just how much tussle there is surrounding a project like this, which has great cost, great impact on different communities, and many winners and losers. But in these cases, as in the case of the studies in Vinck, the process of design concludes, and is followed by the process of construction. At some point the design is done; when tussle at build time is very costly, the goal is to get the tussle articulated and resolved at design time.

What is distinctive (though certainly not unique) about the Internet is that the tussle continues in large part while the system is in use. *Tussle occurs at "run time."* In contrast to a project like the Big Dig, where the process of construction solidifies the design in concrete, steel, and tunnels in the ground, for the Internet the process of design and redesign, construction and reconstruction, use and reuse is ongoing. What this paper attempts to consider is not the simple reality of tussle, but the implications of designing, building and using a system where tussle occurs at run time. What are the circumstances that lead to tussle at run time? How can a system be built to best survive and function under these circumstances?

A. Technology, and Our Role in the Tussle

We who are designers of the Internet should not for a moment think we somehow sit outside or above the tussle. Like any other actor in the process, we bring to the table our values and hopes for the Internet and the society it serves, and we advocate for them in the process of design. With rare exception, society does not grant us a special right to impose our values.[2] As designers, we do however play a special role in the tussle. While we have no special rights, we do have a special and unique power, which is the power to create the technology. So we ought to look at the nature of technology to understand what the power is that we hold.

As technologists, we view technology through a particular lens. It is illustrative, and occasionally startling, to look at the perspective of other disciplines.

Bruno Latour, a noted sociologist, writes that *Technology is Society made Durable* [4]. His observation is that it is technology in the fabric of society that provides stability and persistence to its form. In this view, society is not an external framework into which people and technologies are embedded, but is exactly the artifact that results from the ongoing experimentation, tussle or alignment of all the actors that participate in it [5]. In this perspective, termed the *actor network* view by Callon [6], both human and nonhuman actors (including technology) must be given equal attention as shapers of society, and technology, by its durability, provides an important source of structure in these actor-networks.[3]

So as technologists, we have the power (and it is a special power) to create components of the techno-social fabric that potentially structure and stabilize it. It is a paradox of technology that it can both create new, unexpected and revolutionary opportunities, and limit our ability to change things, a theme that is explored by Callon [8]. As technologists, we recognize this limitation to change, which we usually regard with regret because our pleasure is to change things.[4] We observe that a system (like the Internet) is harder to change as it "grows up." We often see this as a drawback or a limitation, but Latour and Callon are

[2]Although at times we are *de facto* able to exercise that ability unchecked, until society "catches up."

[3]To give equal attention is not to imply that humans and nonhumans are equal in terms of rights or intentions. We can still ascribe intentions to humans, and to technology only the expression of that intention, or agency. Of course, technology sometimes seems to have a mind of its own, which leads to frustration, and to papers with titles such as *Technology as Traitor*, concerning an attempt to deploy SAP [7].

[4]A common exception is one who is benefitting from a technically driven monopoly in some area important to society, and wishes to keep it.

pointing out the advantage—that society may benefit from its stability and predictable character.

This idea that "the network gets harder to change as it grows up" is precisely the implication of the actor network model. Technology in isolation, not embedded in any network of human and nonhuman actors, has nothing to stabilize. It is the whole actor network (as distinct from the Internet as a network of technology) that becomes stable, as all the human and nonhuman actors align and harmonize themselves to common (socio-technical) interfaces. So in the Internet, we look at the protocols, the ISPs, the providers of Internet applications, the users, the laws and the lawmakers, and so on, and we see this whole network becoming more durable to the extent that the actors commit to each other, with the technology as a central anchor in this network. Thus an important question, to which we return below, is to consider what forces make the Internet more or less susceptible to change. For the moment, we point out that this sociological notion of technology as a structuring force of society leads directly to a key theme of this paper: that technologists are, in fact, creating playing fields for the tussles of society to play out in. To understand this better, we first consider several different perspectives on tussle, and then turn to the question of how this observation affects the technical principles on which our systems are architected.

B. Perspectives on Tussle

Many disciplines have some perspective on the nature of tussle, specifically in the context of technology, standards, and networks of actors. Writing from the perspective of business innovation, Christensen [9] describes a very similar phenomenon to the actor network. In *The Innovator's Dilemma*, he describes how incumbents with existing technology get locked into an actor network (though Christensen does not use this term) consisting of their existing customers, the marketing department that listens to those existing customers, engineers who understand existing technology, and managers who stick to existing markets to sustain the bottom line. The durability and rigidity of this actor-network keeps them from pursuing or benefiting from radical disruption. His analysis provides a hint to sort out the paradox of disruption and stability—disruptive technology does not initially succeed by de-stabilizing an existing actor network (or value chain, in the language of the business school). Instead, innovators step outside the existing value chain, and find new customers and new markets, and build up their stability outside the existing network. Only when they have enough durability (stable production and markets) do they then have the potential to overthrow the existing producers.[5]

From another perspective, that of law, Lessig [10] discusses the relative role of law and technology in the stability of the Internet. He does not see the Internet as a "value-neutral" design, but rather as a design that expresses strong values embedded by its designers. His fear is that the forces of government and commerce will shift the whole fabric of the Internet (the actor network in which the technology is embedded) to an outcome with very different values—control, regulation and loss of freedom—and this actor network will be as durable or more than the Internet we see today. His book is a call to act, and to act wisely. It is a call to partake of the tussle.

An economics perspective on tussle in the context of technology and actor networks can be found in [11]. This paper (which uses the term "virtual network" to describe the set of actors) considers the actions of providers in a system with significant externalities and significant opportunities for competition. In the situation they consider, the motivation of the players is usually to stabilize the value chain (the actor network). The paper describes a range of techniques an actor can take, such as pricing strategies, pre-announcement of products, use of property rights, and careful use of open interfaces to signal the intention of the providers. Given the focus of the paper on the goal of stabilizing the value chain, this does beg the question of what we might undertake to keep the value chain fluid, and in what ways this is important.

A more formal model of tussle is provided by the discipline of *game theory*, whose goal is to describe, and more recently, prescribe rules of the underlying tussle "game" that occur during the interactions between actors in the networks. A game represents an abstraction of the underlying tussle environment, and can range from purely conflicting games (so called zero-sum games) where the values of actors in the network are in direct conflict, to coordination games where actors have a common goal but fail to coordinate their actions due to incentive problems. In addition to creating an ontology of tussle environments as games, the theory aims to analyze and construct socially desirable resolutions to each tussle class by specifying contracts that each actor in the network is committed to and has an incentive to follow. The classic theory, first formalized by the seminal zero sum games work of von Neumann and Morgernstern [12] and later extended by the works of John Nash to general sum games [13], achieves much of its power by modeling micro structure of each actor in the network. Actors' information, actions and values are modeled as beliefs, strategies and payoffs respectively and the steady state/s (or equilibria) of the resulting actor network for the given tussle game are then analyzed given these model primitives.

Another dimension of computer technology (and the people who create it) is amplification—the ability to coordinate and advance the actions of a society by spreading ideas from a small group of creators to the larger audience. Wolgar, again from the sociological perspective, explores the metaphor of an IT artifact as text [14], created/written by one set of people and used/read by another. This metaphor reminds us that one set of writers can influence many readers, underlining the ability of technical design and technical artifacts to affect and guide tussle.

System design perspectives on tussle. Designers of distributed and networked systems have historically responded to the existence of tussle in several ways.

One common response is to ignore the issue, or assume that the tussle can be resolved outside the technical system. Current TCP congestion control provides an example. TCP congestion control "works" when and only when the majority of end-systems both participate and follow a common set of rules. This

[5]Earlier, we suggested that it is part of our responsibility to understand who is disenfranchised and who are the outcasts and misfits. This is an important admonition. It is also Christensen's advice when seeking opportunities for disruptive innovation.

strategy places great weight on social pressure to "resolve" the tussle outside the scope of the technical system. It is important to note that the strategy can be effective in many situations—although there is great concern, and widespread belief, that the current situation cannot hold in the Internet, it has worked acceptably well to date. The reason is that the combination of social pressure, standards pressure, and most individual players' inability to make technical modifications has overcome the potential willingness of these players to benefit at others' expense. Should this balance change, the technical design of the system will do nothing to bound or guide the resulting shift.

A second response is to preserve the notion there is "one right answer," but build technical systems that are more resistant to those that percieve the answer differently. This model is time-honored in the distributed system community, and has been applied to the larger space of the Internet. Perlman [15] considers network routing in the presence of byzantine failures. Such a system can be viewed as highly resistant to attempts by players, even small groups of players, to place their interests over the values chosen by the designers. More recently, Savage [16] applies the same strategy to the problem of explicitly uncooperative players in active network measurement (see also [17]), robust congestion signaling (see also [18]), and IP traceback (see also [19], [20]). This work acknowledges that current solutions exist to each of these problems, but that those solutions are dependent on a model of cooperation that no longer exists universally in the network. Savage makes the point that for each of these functions there exist alternative approaches, albeit qualitatively different from each other, that allow for solutions in an uncooperative network.

Recently, systems have been proposed that capture differing user interests using "policy languages." Representative examples of this approach include P3P [21], KeyNote [22], based on PolicyMaker [23], and the policy language embedded in the Common Open Policy Service or COPS protocol [24] of the IETF. This approach explicitly recognizes run-time tussle, and attempts to accommodate it.

Policy languages serve two functions. Explicitly, they allow actors to express their own constraints and requirements within a larger actor space. Implicitly, by imposing an ontology on what can be expressed, they *bound* the tussle that can be expressed within defined limits. This effect can be beneficial, by structuring tussle along natural boundaries as is discussed later in this paper. It can also be defeating, if it prevents the system from capturing and acting on tussles that were not anticipated or seen as important by the language designers. More importantly, the existance of a policy language does nothing to *resolve* tussles, and it does nothing to address the problem of strategic players, malicious users, liars. etc. It simply provides a first step toward accommodation.

Recently, researchers have considered the application of results from game theory to practical distributed and networked systems. This can be seen as an attempt to *reduce or eliminate* tussle from actor networks, by aligning the incentives of the various actors. The main body of the classic game theory is descriptive, focusing on how to model and analyze a given tussle for a static set of actors who are well informed and able to act perfectly. This idealized classical theory has extended over time in two directions: prescriptive mechanism design and bounded rationality/evolutionary game theory. Both are relevant to our topic.

William Vickrey [25], in a seminal work, outlined the beginnings of a theory to generatively design and prescribe actor networks that exhibit a desirable apriori set of properties, over a class of tussle game where actors are only informed about their own preferences and are uncertain of other's preferences—the so called assymetric information games. This theory, later modified and extended by others, showed how to construct rules of a game that guaranteed tussle-free actor networks for a given class of problem revolving around revealing truthful information. With this theory in hand designers begin to have a blueprint for construction of actor network systems that are, within this limited problem domain and model, tussle-free. A key benefit is that with tussle reduced or eliminated in the information subgame, it becomes simpler to reduce or guide tussle in the larger overall game. Recently, there has been an influx of interest from a systems perspective on the computational [26] and network [27] underpinnings of such systems.

Another direction of advancement in theory has been the realization that actors in a network are not, in fact, well informed and perfect optimizers as classic theory requires. In fact actors are often ill-informed (over their own state as well as that of others), myopic and act to satisfy some poorly defined objective. In response, a body of work has arisen (exemplified, for example, by the works of Binmore [28]) that attempts to account for these circumstances.

C. Why is Run-Time Tussle Possible?

While technology may be a durable component of society, much of the Internet's tussle (as we claim) is at "run time," so the Internet seems less durable than, say, the Big Dig with its tunnels through the bedrock. It is well understood that IT artifacts, to some extent made out of software, are more plastic than purely physical artifacts. It is well understood that open interfaces allow for replacement of components. But these simple observations do not tell the whole story. Run-time tussle occurs in a variety of ways in the Internet. It occurs through the process of design and redesign, in the standards arena. It occurs as users and operators pick and choose which bits of technology to deploy, and how to configure and connect them. It occurs as users choose with whom they prefer to interact, and obtain service from. And indeed, the mix of durability and plasticity is shaped at design time by the specification (or not) of open interfaces, and the design (or not) of one or another protocol. At a meta level, this mix itself is somewhat fluid.

Using the model of Callon or Latour, one of the reasons why the Internet is still changeable is that the actor network surrounding the Internet has not become totally consistent. That is, all the tussles—the disagreements and conflicts of interest—have not been driven out of it. These tussles arise, among other reasons, because the open architecture of the Internet allows the continuous entry of new players into the actor network. The entrance of new actors, with fresh perspectives and values, creates continuous churn in the actor network. These actors can be individual users, or new applications and their creators, or (most potent as actors) players that come to

the Internet already embedded in an actor network of their own, perhaps a very solidified one. So when the creation of voice over IP (VoIP) causes the Internet to collide with the "telephone system," the key issue is not a collision of technologies, but a collision between large, heterogeneous actor networks.

This observation allows us to make a somewhat more complex claim about the durability versus the plasticity of the Internet. It is not just that the open nature of the Internet allows new applications and capabilities to be added to the network. It is that the new applications bring *new actors* to the actor network, which keeps the actor network from becoming frozen, which in turn permits change to occur.

If tussle occurs (and is managed) primarily at design time, then the institutions of design will be the venue for resolution. If Internet tussles all occurred at design time, we would look to the Internet Engineering Task Force, and in particular the Internet Engineering Steering Group, which sets direction for the IETF.[6] But since tussle occurs at run time, it occurs in many places and in many ways. The venue for tussle is *heterogeneous*, a manifestation of the complex actor network. Lobbyists, lawyers and legislators tussle over laws, police and courts tussle over their enforcement, competitors tussle in the marketplace and so on. Ongoing tussle does occur at design time, as protocols and standards are changed, augmented and replaced. It occurs as users pick and choose what technology and standards to exploit, and how to connect them together. This heterogeneity limits the influence that any one institution can have in resolving tussle.

There is an open question, which no discipline seems to have the tools to answer in advance, as to whether the tussle will be driven out of the actor network, the actors will be forced into alignment, or whether this dynamic semi-stable system of today will persist. This question is of great interest, even if we can only speculate on the range of outcomes. But the previous discussion offers at least a hint. When new applications and user groups cease to come to the Internet, and the set of actors in the actor network becomes fixed, then we can assume that the tensions and tussles in the network will begin to be resolved, and this will imply a freezing of the actor network, and a freezing of the Internet. So we should look for a time when innovation slows, not just as a signal but also as a pre-condition of a durably formed and unchangeable Internet.

We as authors recognize that we have only scratched the surface of the multidisciplinary literature on tussle, and specifically tussle in the context of technology, systems engineering, and standard setting. Our focus for the rest of this paper is on the technical design process in the context of run-time tussle—how might designers make both value decisions and technical design decisions in this context?

III. Our Job as Designers

In this societal context, Internet technologists do their work—they design and redesign, configure and deploy. And in doing so, they play two intertwined roles at the same time. First, they are actors themselves in the tussle. Second, they can shape the nature of the tussle that comes later. Designers face a choice. If they have strongly held values and objectives, either economic (a design that favors their firm or exploits their patent, for example) or social (a design that favors individual actors or the rights of the state), they can attempt to embed those values into the design in a way that is hard to change. This option can lead to great success (an intentional freezing of the network in favor of some stakeholder), or total failure if the design is rejected, worked around, outlawed or otherwise ignored. The other choice is not to impose a rigid form to the design, but to design for choice at a later time, at run time. Knowing that they are not the final arbiter of tussle, but that tussle will occur at run time after they are done, designers can try to frame the context for that tussle in ways that are benign, in that subsequent tussle can occur without causing harm to the overall design of the network.

Design for change is an option. It is the choice to forebear from imposing a fixed view of the desired outcome. And here we as authors express our own values; it is our opinion that design for choice—design that accommodates tussle rather than attempting to preclude it—has been a beneficial option in the past. It has preserved the option for evolution, it has preserved the option for innovation and the creation of new value, and it has allowed the Internet to keep pace with the computer industry as that industry evolves.

IV. Principles

In this section we offer some design principles to deal with tussle. Based on our preference that we accommodate tussle rather than preclude it, our highest-level principle is:

- Design for tussle—for variation in outcome—so that the outcome can be different in different places, and the tussle takes place within the design, not by distorting or violating it. Do not design so as to dictate the outcome. Rigid designs will be broken; designs that permit variation will flex under pressure and survive.

Second, we identify a principle that strengthens the ability of an architecture to accommodate tussle, and assists in the task of design for change:

- Modularize the design along tussle boundaries, so that one tussle does not spill over and distort unrelated issues.

We discuss the second, more concrete principle first.

A. Modularize Along Tussle Boundaries

Systems designers know to break complex systems into modular parts. Modularity is typically used to manage complexity, allow for independent implementation and component reuse, or meet other technical goals. But "tussle isolation" is perhaps a new principle.

- Functions that are within a tussle space should be logically separated from functions outside of that space, even if there is no compelling technical reason to do so. Doing this allows a tussle to be played out with minimal distortion of other aspects of the system's function.

The design of the DNS provides an example. The current design is entangled in debate because DNS names are used both to name machines and to express trademark. In retrospect, since

[6]There has been much written on tussle in the setting of standards, and competition over standards. See, for example, [29] and [30].

it was (or should have been) obvious that fights over trademarks would be a tussle space, names that express trademarks should be used for as little else as possible. In particular, one might imagine separate strategies to deal with the issues of trademark, naming mailbox services, and providing names for machines that are independent of location (the original and minimal purpose of the DNS). One could then try to design these latter mechanisms to try to duck the issue of trademark.

- Solutions that are less efficient from a technical perspective may do a better job of isolating the collateral damage of tussle.

In contrast, the current design of IP QoS tries to isolate tussles. The use of explicit ToS bits to select QoS, rather than binding this decision to another property such as a well-known port number, disentangles what application is running from what service is desired. It can be anticipated that there will be tussles about what applications a user can run, and separately tussles about what service qualities can be provided. The designers felt that it was better to separate these ideas. This modularity allows tussles about QoS to be played out without distortions, such as demands that encryption be avoided simply to leave well-known port information visible or the encapsulation of applications inside other applications simply to receive better service.

B. Design for Choice

An important aspect of designing for variation in outcome is *design for choice*. Network protocols are designed so that different parties on the network can communicate with each other, consumers can make use of the resources of providers, and providers can interconnect with each other to provide service. It is important that protocols be designed in such a way that all the parties to an interaction have the ability to express preference about which other parties they interact with. Protocols must permit all the parties to express choice.

For example, the design of the mail system allows the user to select his SMTP server and his POP server. A user can pick among servers, perhaps to avoid an unreliable one or pick one with desirable features, such as spam filters. Users can select what news server they use, perhaps to prevent their children from encountering some of the more colorful news groups. This sort of choice drives innovation and product enhancement, and imposes discipline on the marketplace.

The form that the choice takes for the different parties may be different. A user of mail might choose her SMTP server by configuring a mail-sending program. An ISP might try to control what SMTP server a customer uses by redirecting packets based on the port number.[7]

Providing this sort of choice has a drawback—it adds to the complexity of configuring and using a service. For naïve users, choice may be a burden, not a blessing. To compensate for this complexity, we may see the emergence of third parties that rate services (the on-line analog of Consumers Reports) and parties that provide pre-configured software to relieve the user of dealing with the details of choice.

[7]An over-generalization of the tussle is that service providers exercise control over routing; end-users control selection of other end-points. End-users try to over-rule constrained routing with tunnels and overlay networks.

C. Open Interfaces and Tussle

An important component of design for tussle (but not the only aspect) is the use of open interfaces. Open interfaces have played a critical role in the evolution of the Internet, by allowing for competition among algorithms, implementations, and vendors, and by enabling rapid technical progress through replacement of modular parts rather than entire systems. But open interfaces also allow choice at run-time, not just replacement. If a protocol allows a party to select among alternative providers of service, for example, this usually implies that the interface to that service is well-defined, so that independent versions of the service can be constructed.

Open interfaces at tussle points will differ in character from interfaces that just facilitate replacement and reuse of components. Tussle interfaces need to be designed to take into account the different interests of the parties to the tussle. For example, BGP is used as the routing protocol among ISPs, who interconnect but are business competitors. As we will discuss below, BGP has a different character than a protocol such as OSPF that is designed to be used within a given domain (hopefully a more tussle-free context). The routing arrangements among ISPs are generally not public, even though everyone can see the consequences at the BGP level. A link-state routing protocol requires that everyone export his link costs, while a path vector protocol makes it harder to see what the internal choices are. In the context of tussle, it matters if choices and the consequence of choices are visible.

Interfaces for tussle may benefit from the following sorts of properties, which are not always important in other cases.

- Visible exchange of value.
- Exposure of cost of choice.
- Visibility (or not) of choices made.
- Tools to resolve and isolate faults and failures.

In certain forms of tussle and run-time choice there is often an exchange of value for service. Value need not be "money" but often will be. Napster is a nonmonetary example that illustrates the "mutual aid" aspect of peer-to-peer networking. Whatever the compensation, recognize that it must flow, just as much as data must flow. Sometimes this happens outside the system, sometimes within a protocol. If this "value flow" requires a protocol, design it. (There is an interesting case study in the rise and fall of micro-payments, the success of the traditional credit card companies for Internet payments, and the emergence of PayPal and similar schemes.)

D. A Wide Range of Mechanism is Used for Tussle

We have discussed one tool to facilitate tussle: interfaces that are open, well-specified and crafted for tussle. But interfaces are only part of the story. In many cases, choice at run time requires an explicit protocol for selection. In many cases, the different parties to the tussle use different mechanisms, as noted above, such as restrictions on routing, tunnels and overlays, or intentional perversion of DNS information. The mechanisms of

tussle will depend on the nature of the tussle. In some cases, the interests of the players are simply adverse, and there is no win-win way to balance them. In this case, mechanism choice will be independent and unilateral. But in many cases, players' interests are not adverse, but simply different. A user wants to send data; a provider wants to be compensated for carrying it. While this implies a natural tussle over pricing, in the end both parties realize that they must meet the other's needs. In this case, the choice of mechanism must itself be mutual.

V. Tussle Spaces

In this section we discuss some specific aspects of the Internet in which different players with competing interests come together. In each case, our goal is to examine the nature of the tussle and to illustrate how our principles can be applied in specific cases. We suggest some specific research areas that would benefit from application of our principles.

A. Economics

One of the tussles that defines the current Internet is the tussle of economics. The providers of the Internet are not in the business of giving service away. For most, it is a business, run to make a profit. This means they are competitors, and look at the user, and each other, as a customer and a source of revenue. Providers tussle as they compete, and consumers tussle with providers to get the service they want at a low price.[8]

How can we, as engineers, shape the economic tussle? In fact, we have great power to shape this tussle, but first we have to understand the rules that define it. A standard business saying is that the drivers of investment are fear and greed. Greed is easy to understand—it drove hundreds of billions of dollars worth of investment in telecommunications over the last decade, much of which has been lost in bankruptcy. But fear is more subtle. The vector of fear is competition, which results when the consumer has choice. The tussle among providers and consumers in a competitive landscape is the most basic attribute of a marketplace. Most economists of a "western" bent would argue that competition is good; it drives innovation, disciplines the market, ensures efficiency, and removes the need for intervention and regulation of a market. To make competition viable, the consumer in a market must have the ability to choose. So our principle that one should design choice into mechanism is the building block of competition.

Here are some specific examples, with implications for research and network design:

1) Provider Lock-In From IP Addressing: ISPs would like to find ways to lock in their customers; customers want to preserve the ability to change among providers. This illustrates the basic consumer-producer tussle in a competitive world. For hosts that use static addresses, renumbering is a complex task. Because renumbering hosts can be hard, there is a very explicit tension today between the desire to have addresses reflect topology to support efficient routing and the desire of the customer to change providers easily. Either a customer is locked into his provider by the provider-based addresses, or he obtains a separate block of addresses that is not topologically significant and therefore adds to the size of the forwarding tables in the core of the network. Mechanisms that favor the consumer in this tussle include dynamic host numbering (DHCP) and dynamic update of DNS entries when the host is renumbered.

- A desire for vigorous competition would suggest that the consumer should have the choice to move from ISP to ISP. Given that, the Internet design should incorporate mechanisms that make it easy for a host to change addresses and to have and use multiple addresses. Addresses should reflect connectivity, not identity, to modularize tussle. This would relieve problems with end-node mobility, improve choice in multihomed machines, and improve the ease of changing providers.

2) Value Pricing: One of the standard ways to improve revenues is to find ways to divide customers into classes based on their willingness to pay, and charge them accordingly—what economists call value pricing. An example from another sector is the "Saturday night stay" criterion for airline travel. It costs the airline no more to carry a passenger if she does not stay over Saturday night, but this restriction tends to separate the business and pleasure traveler, which is useful because the business traveler seems to have a greater willingness to pay. Airlines impose Saturday night stay restrictions, and consumers respond by buying multiple tickets, and using only some of the segments of the flight. Airlines respond by declaring this behavior unacceptable. And thus the tussle evolves.

As an example of similar behavior in the Internet, some acceptable use policies for residential broadband access prohibit the operation of a server in the home. To run a server, the customer is required to pay a higher "business" rate. Customers who wish to sidestep this restriction can respond by shifting to another provider, if there is one, or by tunneling to disguise the port numbers being used. The probable outcome of this tussle depends strongly on whether one perceives competition as currently healthy in the Internet, or eroding to dangerous levels.

- This discussion illustrates the point that many design decisions today embody specific social values. The design and deployment of tunnels (or other mechanisms to mask what services are being used by a consumer) shifts the balance of power from the producer to the consumer. Given that value pricing is not a moral wrong, should the consumers be aided in their quest to bypass the controls of the producers? Those who see the consumer as "the little guy" being abused by the "big providers" will design such mechanisms, and this is part of the tussle, not something that happens outside the tussle. What mechanisms get designed, and what standards get approved, are all part of the tussle.

3) Residential Broadband Access: There is concern today that the advent of broadband residential access will be accompanied by a great reduction in competition. Today there are perhaps 5000 dialup Internet service providers. A pessimistic outcome five years in the future is that the average residential customer will have two choices—his telephone company and his

[8]There is now considerable interest in the economics community in the nature of the Internet. Some of the seminal papers are published in [31]. For an overview of the current literature on Internet economics, see the Web site maintained by Mackie-Mason at http://china.si.umich.edu/telecom/net-economics.html.

cable company—because they control the wires. This loss of choice and competition is viewed with great alarm by many, who fear that it may lead to higher prices and restrictions on what the user may do, and there are many forces aligning to fight this loss of competition. Some are regulatory, calling for laws to mandate "open access," to force the owners of the wires to allow multiple ISPs to use them. Economists and regulators hope that multiple providers will install their own cables, to increase competition.[9] However, in a tussle of competition, one cannot compel a potential provider to invest and enter a market.

Using the principles of this paper, one could speculate on what sorts of investments are actually likely to be made, and to think about what choice, and what tussle modularity, would improve the outcome of such an investment. One investment option that is gaining momentum now is municipal deployment of fiber, because fiber installed by a neutral party such as a municipality can be a platform for competitors to provide higher level services (e.g., a phone, Internet or television). This requires that the equipment lighting the fiber support multiple service providers. Most of the equipment made today is not "naturally open" in this way, having been designed without consideration of this particular modularity boundary (or indeed with the specific goal of confounding it).

- An important R&D project is to design and demonstrate a fiber-based residential access facility that supports competition in higher-level services. Technical questions include whether sharing should be in the time domain (packets) or color domain, how the fairness of sharing can be enforced and verified, an approach to fault isolation and other operational issues, and how incremental upgrades can be done. This project is motivated both by the principle of "design for choice" and as well by recognition of new tussle boundaries.

Most of today's "open access" proposals fail to balance the interests of concerned parties because they are not modularized along tussle space boundaries. For example, the capital costs and deployment pragmatics of broadband infrastructure differ greatly from those of operating mail and web servers. This creates a natural boundary between the two tussle spaces of broadband facilities provision and ISP services. Proposals that implement open access at this modularity boundary are more likely to benefit the Internet as a whole, because they allow each tussle to play out independently. But they probably will not work to the advantage of those that invest in the fiber.

4) Competitive Wide Area Access: Today, the Internet system does not let the individual customer select his "long distance provider" the way the telephone system does. This is an example of designers failing to appreciate a competitive tussle space.

A requirement for "policy-based routing" was recognized early by Internet designers. Before the Internet was commercialized, the introduction of multiple providers in the NSFnet backbone created a tussle space relating to autonomy, mutual trust, and acceptable use policies [33]. The fundamental question then was who would set routing policy? There were two competing technical proposals answering this in different ways: user control [34] and provider control [33]. The two proposals were shown to have rough equivalence in the set of expressible policies, yet from the tussle viewpoint they had very different consequences.

In the end, this very fundamental choice was made on pragmatic grounds: the user control proposal required changing the data plane (IP protocol), which seemed to be a daunting task even then. On the other hand, provider control required changing only the control plane (inter-domain routing), in the form of the Border Gateway Protocol (BGP) [35]. Furthermore, the providers and their suppliers had the economic incentive to drive the engineering and standardization of BGP, and there was no corresponding economic drive to tilt the playing field toward users control of policies. One can only speculate that a different result might have emerged had the technical community at the time considered the design for tussle.

One could argue, in favor of provider-controlled policy, that there would be sufficient competition in the wide-area market because there were going to be many ISPs directly competing to serve the customer. Letting the local provider enter into a wholesale arrangement to obtain wide area service seemed adequate, because if one local provider made an unsatisfactory choice in wide area provider, the customer could just switch to a new local provider.

But this decision may be having undesirable consequences today. It is possible that customers today would be much more likely to see more service diversity, e.g., a quality of service support for applications, if there were more competition.

- The Internet should support a mechanism for choice such as source routing that would permit a customer to control the path of his packets at the level of providers. A design for such a system must include where these user-selected routes come from or how they are constructed, how failures are managed, and how the user knows that the traffic actually took the desired route. The capability must also be approachable by a broad class of users of varying sophistication. This is a very complex design challenge,[10] but could have a great influence.

This example illustrates another important point about competition. One should be prepared to pay for what one uses, or there is little incentive for a provider to offer it. Today, service providers do not like loose source routes, because ISPs do not receive any benefit when they carry traffic directed by a source route. ISPs enter into business arrangements that determine which traffic they agree to carry across which interfaces, and a source route has the effect of overriding these arrangements. Why should they be enthusiastic about this? Since source routes do not work effectively today, researchers propose even more indirect ways of getting around provider-selected routing, such as exploiting hosts as intermediate forwarding agents. (This kind of overlay network is a tool in the tussle, certainly.) Another, perhaps simpler, approach is to compensate the provider for carrying the packets. But this idea tends to upset designers as well as customers, because they fear they will end up in an

[9]For an analysis of issues in residential broadband access, see [32].

[10]In particular, today's loose source routes, even if widely implemented, would provide only a small portion of what is needed. For discussion of a more complete design using this approach, see [36].

onerous "pay by the byte" situation, which does not seem to have much market appeal.

- The design for provider-level source routing must incorporate a recognition of the need for payment. There must be enough generality in the payment schemes that the market can select an outcome that works for all parties. (Remember, we are not designing the outcome, only the playing field for the tussle.)
- Overlay architectures (e.g., [37]) should be evaluated for their ability to isolate tussles and provide choice. A comparison is warranted between overlay architectures and integrated global schemes to understand how each balances the relative control that providers and consumers have, and whether economic distortion is greater in one or the other.

B. Trust

One of the most profound and irreversible changes in the Internet is that by and large, many of the users do not trust each other. The users of the Internet no longer represent a single community with common motivation and shared trust. There are parties with adverse interests, and some genuine "bad guys" out there. This implies that mechanisms that regulate interaction on the basis of mutual trust should be a fundamental part of the Internet of tomorrow.[11]

Most users would prefer to have nothing to do with the bad guys. They would like protection from system penetration attacks, DoS attacks, and so on. This is a profound tussle, between people who want to be left alone, and people who want to bother them. Since host security today is of variable and mostly poor quality, this desire for protection leads to firewalls. Firewalls change the Internet from a system with transparent packet carriage between all points (what goes in comes out), to a "that which is not permitted is forbidden" network. This is a total reversal of the Internet philosophy, but pure transparency is not what most users long for. For over ten years, Internet purists have been bemoaning the fact that firewalls inhibit innovation and the introduction of new applications (fifteen years ago they were called "mail gateways"), but firewalls have not gone away.

The principle of "design for choice" would imply that users should be able to choose with whom they interact, and users should be able to choose the level of transparency they offer to other users. The principle of "tussle isolation" suggests that these mechanisms should not be overloaded on to any other mechanism, but should be separated. Further, one should consider if, within the broad topic of trust, there are separable issues.

The first topic is control over which parties are willing to exchange packets with each other.

- In the abstract, there is a technical question as to whether each end-node can implement sufficient trust-related controls within itself, or whether delegation of this control to a remote point inside the network is required—a "trust-aware firewall." As a practical matter, the market calls for firewalls. Firewalls that provide trust-mediated transparency must be designed so that they apply constraints based on who is communicating, as well as (or instead of) what protocols are being run and where in the network the parties are. Along with this device must be protocols and interfaces to allow the end node and the control point to communicate about the desired controls.[12] Issues of choice and tussle arise: who gets to pick which firewall a user uses?

Another tussle about firewalls is worth noting. Who gets to set the policy in the firewall? The end user may certainly have opinions, but a network administrator may as well. Who is "in charge"? There is no single answer, and we better not think we are going to design it. All we can design is the space for the tussle. But this illustrates the point about visibility of decision-making. If a system administrator has installed control rules in a firewall that affect an end user, should that end user be able to download and examine these rules? One way to help preserve the end-to-end character of the Internet is to require that devices reveal if they impose limitations on it. However, there is no obvious way to enforce this requirement, so it becomes a courtesy, not a real requirement.

Another dimension of trust is the fact that most users do not trust many of the parties they actually want to talk to. We connect to web sites but are suspicious that they are gathering information on us, stealing our credit cards, not going to deliver what they promised, and so on. In this case, the solution is more complex; we depend on third parties to mediate and enhance the assurance that things are going to go right. Credit card companies limit our liability to $50, or sometimes nothing, in case of dispute. Public key certificate agents provide us with certificates that assure us we are talking to the party we think we are. Web sites assess and report the reputation of other sites. The fact of these third parties contrasts with our simple model of two-party end-to-end communication among trusting parties. Each individual interaction may be two-party end-to-end, but the application design is not.

- An important engineering principle for future applications is that there should be explicit ability to select what third parties are used to mediate an interaction, and to act as an agent for the end-user in improving his trust in the operation. The parties must be able to choose, so they can select third parties that they trust.[13]

Another space in which trust is eroding is that users less and less trust the software they have to run. They suspect their operating system and browser of gathering information on them and passing it on without their knowledge, or turning them in for software license violations. There are web sites that claim to look at the outgoing data stream from the user's machine and detect and remove any information that is leaking out.

- This problem may best be dealt with using nontechnical means—regulation, public opinion and so on [39]. Just

[11]A thoughtful analysis of trust that has shaped our thinking is provided by [38].

[12]The IETF has considered such standards, e.g., the MIDCOM working group.

[13]An interesting debate relevant to this topic emerged during the IETF's chartering of the Open Pluggable Edge Services (OPES) working group, and the resulting IAB deliberation on policy concerns. The IAB has focused on issues of whether one end or both have to concur with the insertion of an intermediate node in the communication, and what tools the user should have to detect and recover from a faulty node.

because a problem manifests in a technical space, it does not mean it has to be solved there. But it is an interesting exercise to consider whether there are technical means to protect a user from software running on their own machine. The history of mandatory security controls and security kernels suggests that this problem is thorny.

1) The Role of Identity: One obvious point about trust is that if communication is to be mediated based on trust, then as a preliminary step, parties must be able to know to whom they are talking. Otherwise, one has little basis for judging how much to trust others.

One could take this as a call for the imposition of a global namespace of Internet users, with attached trust assessments. We believe this is a bad idea. It is hard to imagine a global system that is really trustworthy. More importantly, there are lots of ways that parties choose to identify themselves to each other, many of which will be private to the parties, based on role rather than individual name, etc. What is needed is a framework that translates these diverse ways into lower level network actions that control access. This implies a framework for talking about identity, not a single identity scheme. We suggest that such a framework could usefully share and arbitrate information across many layers of the protocol stack.

The need to know to whom we are talking will challenge a current precept of the Internet, which is that it is permissible to be anonymous on the Internet. There is a fundamental tussle between the ideas of anonymous action, and the idea that in a society where "that which is not forbidden is permitted," one can be held accountable for ones actions. A possible outcome of this tension is that while it will be possible to act anonymously, many people will choose not to communicate with you if you do, or will attempt to limit what you do.[14] A compromise outcome of this tussle might be that if you are trying to act in an anonymous way, it should be hard to disguise this fact. This illustrates the observation that one must think about whether the consequences of choice are visible, or can be hidden.

C. The Tussles of Openness

One of the most profound fears for the Internet today is that it will lose its "open" qualities: the openness to innovation that permits a new application to be deployed, the openness of access that allows a user to point their Web browser at any content they please, the openness that allows a user to select the servers and services that best meet their needs.

The openness to innovation—to new applications and new uses—has perhaps been the most critical success factor for the Internet. But openness is not an unalloyed virtue for service providers. Openness often equates to competition, which creates the fear factor that demands costly investment and drives profits to a minimum. Many telephone company executives remember the good old monopoly days, with a comfortable regulated rate of return and no fear. And many current ISPs may long for a return to those less open, high margin days, if they could only figure out how to get there. The keys are closed or proprietary interfaces, and vertical integration.

[14]An analog is the current situation with Caller ID, where a sender can block the caller's information, but the receiver can refuse to accept calls from a sender that does.

Motivations concerning open versus proprietary systems have much to do with economics. Economists have studied the motivation of providers with various degrees of market power to choose open or proprietary interfaces; see [40]. Industry understands that interfaces, or lack thereof, can shape a market.[15] There is probably a whole paper on the tussles surrounding open versus closed systems. However, as a starting point, the first exercise should be to speculate about whether these various openness tussles can be modularized and disentangled, and what this means for mechanism design.

Vertical integration—the bundling together of infrastructure and higher-level services—requires the removal of certain forms of openness. The user may be constrained to use only certain providers of content, or to pay to run certain protocols, and so on. However, vertical integration has nothing to do with a desire to block innovation. Even in a market with a high degree of vertical integration, innovation that brings new value to the customer is likely to benefit all parties. So it would be wise to separate the tussle of vertical integration, about which many feel great passion, from the desire to sustain innovation.

The technical characteristic of the network that has fostered innovation is transparent packet carriage—the ability to deploy a new protocol without having to modify the inside of the network. But transparency is not the same thing as openness, though they are related. With this brief motivation, we consider some old design principles of the Internet, including the principle that is usually equated with transparency, the end-to-end arguments.

VI. Revisiting Old Principles

A. The Future of the End to End Arguments

One of the most respected and cited of the Internet design principles is the end-to-end arguments, which state that mechanism should not be placed in the network if it can be placed at the end node, and that the core of the network should provide a general service, not one that is tailored to a specific application [44]. There are two general dimensions to the arguments: innovation and reliability.

Innovation: If the core of the network has been tailored to one specific application, this may inhibit the deployment of other applications. If the core of the network must be modified to deploy a new application, this puts a very high hurdle in front of any unproven idea, and almost by definition, a new idea is unproven.

Reliability and robustness: If bits of applications are "in the network," this increases the number of points of failure that can disable the application. The more simple the core of the network, the more reliable it is likely to be.

The simplest application of the end-to-end arguments produces a network that is *transparent*; packets go in, and they come out, and that is all that happens in the network. This simple idea was very powerful in the early days of the Internet, but there

[15]While technical network designers may not think about open interfaces as a tool to drive market structure, industrial players understand this fully. When then Senator Gore announced his vision for a National Information Infrastructure (NII) in the early 1990s, at least two organizations produced requirement documents for the "critical interfaces" that would permit the NII to have a suitable structure [41]–[43].

is much fear that it seems to be eroding, for many of the reasons discussed above:

- The loss of trust calls for less transparency, not more, and we get firewalls.
- The desire for control by the ISP calls for less transparency, and we get application filtering, connection redirection, and so on.
- The desire of third parties to observe a data flow (e.g., wiretap) calls for data capture sites in the network.
- The desire to improve important applications (e.g., the Web), leads to the deployment of caches, mirror sites, kludges to the DNS and so on.

This is a great deal of mechanism, a large potential loss of transparency, and an increasing focus on improving existing applications at the expense of new ones. So what is the future of the end-to-end arguments? We argue that the end-to-end arguments are still valid and powerful, but need a more complex articulation in today's world. The discussion to this point gives us some guidance.

Evolution and "enhancement" of existing, mature applications is inevitable. As applications become popular, lots of players—application service providers, ISPs, equipment providers, etc.—will want to get involved in them, whether as a move toward vertical integration, enhancement of performance or reliability, or some other reason. This will almost certainly lead to increased complexity, perhaps decreased reliability or predictability, and perhaps an evolution of the overall application away from the original vision. We should not imagine that anyone can do much about this. If applications are designed so that the user can control what features "in the network" are invoked, the designer may have done as much as they can.

The most we can do to protect maturing applications is to bias the tussle. If application designers want to preserve choice and end user empowerment, they should be given advice about how to design applications to achieve this goal. This observation suggests that we should generate "application design guidelines" that would help designers avoid pitfalls, and deal with the tussles of success.

Keeping the net open and transparent for new applications is the most important goal. Innovation and the launch of new applications is the engine that has driven the growth of the Internet and the generation of new value. So barriers to new applications are much more destructive than network-based support of proven applications. Since new applications must, almost of necessity, launch incrementally, they most benefit from the transparent simplicity that the end-to-end arguments fostered. By the principle of isolation of tussle, any barriers that are put into the network as a result of the desire to control mature applications or issues of trust should not prevent parties that want transparency from getting it.

Failures of transparency will occur—design what happens then. Today, when an IP address is unreachable, there is little in the way of helpful information about why. A sophisticated user can run traceroute, but today's normal user just gets frustrated. Tools for fault isolation and error reporting would help—the hard challenge is not so much to find the fault but to report the problem to the right person in the right language. That person may be someone who can fix the problem, or someone who can decide to choose a different path or provider—fault reporting is as much a tool of tussle management as it is a tool of technical repair. Of course, some devices that impair transparency may intentionally give no error information or even reveal their presence, and that must be taken into account in design of diagnostic tools.[16]

Peeking is irresistible. If there is information visible in the packet, there is no way to keep an intermediate node from looking at it. So the ultimate defense of the end-to-end mode is end-to-end encryption. End-to-end encryption addresses both the threat that someone wants to steal or modify the information, and the threat that the ISP wants to control what its customers are doing.

Of course, encrypting the data stream has drawbacks. One is that the actions of the ISP might actually be making things better. They might be offering performance improvements or other benefits that the end user actually wants. But this situation is not an issue; if the user has control over whether the data is encrypted or not, the user can decide if the ISP actions are a benefit or a hindrance. The other drawback is that encrypting the stream might just be the first step in an escalating tussle between the end user and the network provider, in which the response of the provider is to refuse to carry encrypted data. It is probably not the case that a commercial ISP would escalate to this level, though some ISPs today refuse to support VPNs without a higher level of payment. In the U.S., competition would probably discipline a provider that tried to block encryption. But a conservative government with a state-run monopoly ISP might. And in that case, policy will probably trump technology in any case. Then the advantage of having the encrypted mode is that it would force the government to be explicit about what their policy was. Forcing the choice to be public and visible is about all that technology can do to moderate this situation.[17]

Note that in a multiway application, where third parties are involved to insure the validity of the transaction, the meaning of "end-to-end" gets more complex, and so does the proper use of encryption.

B. Separation of Policy and Mechanism

Another design principle of great age and uncertain origin[18] is that technologists should design policy-free mechanism, and allow those who use the system (whether literal "users," administrators, etc) to adjust the mechanisms to match their specific needs. This paper challenges this principle as perhaps being too simplistic. True policy-free design is, at best, extremely difficult. Mechanism defines the range of "policies" that can be invoked, which is another way of saying that mechanism bounds the range of choice. So in principle there is no pure separation of policy from mechanism. As we assert above, the choice to forebear from constraint and to leave choice to those who come later is itself a value-laden choice, albeit one we respect.

[16]See the footnote above on the deliberations by the IAB on the charter for the OPES working group.

[17]The next step in this sort of escalation is steganography—the hiding of information inside some other form of data. It is a signal of a coming tussle that this topic is receiving attention right now.

[18]An early articulation of the principle can be found in [45].

However, this analysis does not totally negate the principle. The chief advantage of attempting to separate mechanism and policy is to isolate some regions of the system from tussle. Even if the attempt is not completely successful, these isolation regions can serve to separate different tussles from each other, and can serve as technological 'fixed points' that allow different tussles to play out at different speeds.

- Perhaps the most challenging intellectual puzzle in this design space is to discover parts of mechanism that really can be divorced from policy—which, in other words, actually *are* value-neutral.

One value (or bias) that is shared by many people is user empowerment. This is the preference that the user, rather than the service provider or the software provider, be able to pick what applications to run, what servers and services to use, and so on. User empowerment, to many, is a basic Internet principle, but for this paper, it is the manifestation of the right to choose—to drive competition, and thus drive change.

One could argue that user empowerment is a bias, of the "David and Goliath" sort—a bias imposed on the tussle between the little guy and the provider, who is seen as "big and bad." This view would suggest that to the extent one tries to be value-neutral in the design of mechanism, one should not favor user empowerment. One could also argue that the fundamental design goal of the Internet is to hook computers together, and since computers are used for unpredictable and evolving purposes, making sure that the users are not constrained in what they can do is doing nothing more that preserving the core design tenet of the Internet. In this context, user empowerment is a basic building block, and should be embedded into all mechanism whenever possible. This paper suggests that the latter view is the defensible one, because choice is a basic tool to deal with tussle.

The recognition of tussle as a fundamental behavior does give one further hint at how to try to separate mechanism from policy. If one can find spaces where tussles are unlikely, then (as noted above) the interfaces and mechanisms can be simpler. If one can truly separate tussles, then one can do a better job of matching mechanism to problem. So the instruction to "separate mechanism from policy" is not incorrect, but just requires careful thought to carry out as best one can.

VII. Lessons For Designers

This more complex interpretation of old design principles, and the introduction of new principles, needs to be seen in terms of system synthesis. How can we, as designers, build systems with desired characteristics and improve the chances that they come out the way we want? If we try to design a system that is open, for example, which means we will encounter the tussles surrounding vertical integration and capture of value in exchange for investment, how can we proceed?

One can learn from the past. To some of us in the research community, a real frustration of the last few years is the failure of explicit QoS to emerge as an open end-to-end service. This follows on the failure of multicast to emerge as an open end-to-end service. It is instructive to do a post-mortem on these failures.[19] Here is one hypothesis. For the ISPs to deploy QoS, they would have to spend money to upgrade routers and for management and operations. So there is a real cost. There is no guarantee of increased revenues. Why risk investment in this case? If the consumer could exercise effective competitive pressure in ISP selection, fear and greed might have driven ISPs to invest, but the competitive pressures were not sufficient. On the other hand, if ISPs use the new QoS mechanisms in a closed way, rather than an open way, they greatly enhance revenue opportunities. Thus, for example, if they deploy QoS mechanisms but only turn them on for applications that *they* sell, they reduce the open nature of the Internet and create opportunities for vertical integration. If Internet Telephony requires QoS to work, and they only turn on QoS for *their* version of Internet Telephony, then they can price it at monopoly prices.

One can thus see the failure of QoS deployment as a failure first to design any value-transfer mechanism to give the providers the possibility of being rewarded for making the investment (greed), and second, a failure to couple the design to a mechanism whereby the user can exercise choice to select the provider who offered the service (competitive fear). The argument about choice here is actually subtle. The user had the power to choose the level of QoS needed—that could be expressed in the ToS bits. What was missing was routing, to allow the user to favor one ISP over another if that ISP honored the bits.

- Anyone who designs a new enhancement for the Internet should analyze the tussles that it will trigger, and the tussles in the surrounding context, and consider how they can be managed to ensure that the enhancement succeeds. As noted above, a powerful force is the tussle of competition. Protocol design, by creating opportunities for competition, can impose a direction on evolution.

VIII. Conclusion

As the Internet evolves to become a full component of society, the person most likely to be dismayed is the fabled cypherpunk. Ref. [46] summarizes the cypherpunk view of privacy as follows: "[T]he cypherpunk's credo can be roughly paraphrased as 'privacy through technology, not through legislation.' If we can guarantee privacy protection through the laws of mathematics rather than the laws of men and whims of bureaucrats, then we will have made an important contribution to society. It is this vision which guides and motivates our approach to Internet privacy." Our position is that the laws of men and the so-called whims of bureaucrats are part of the fabric of society, like it or not. They are some of the building blocks of tussle, and must be accepted as such. We, as technical designers, should not try to deny the reality of the tussle, but instead recognize our power to shape it. Once we do so, we acquire a new set of hard, technical problems to solve, and this is a challenge we should step up to willingly.

[19]The case study of the failure to deploy multicast is left as an exercise for the reader.

ACKNOWLEDGMENT

The authors gratefully acknowledge essential discussions with members of the NewArch project (http://isi.edu/newarch), particularly M. Handley, N. Chiappa, T. Faber, and A. Falk. P. Faratin, J. Camp, S. Gillette, and the ACM SIGCOMM and IEEE/ACM TRANSACTIONS ON NETWORKING reviewers provided welcome comments and feedback, shaping our future work as well as this paper. S. Floyd provided invaluable encouragement at a well-chosen moment. Sincere thanks to all.

REFERENCES

[1] M. S. Blumenthal and D. D. Clark, "Rethinking the design of the Internet: The end-to-end arguments versus the brave new world," *ACM Trans. Internet Technol.*, vol. 1, no. 1, Aug. 2001.

[2] D. Vinck, Ed., *Everyday Engineering: An Ethnography of Design and Innovation*. Cambridge, MA: MIT Press, 2003.

[3] T. Hughes, *Rescuing Prometheus*. New York: Pantheon Books, 1998.

[4] B. Latour, "Technology is society made durable," in *A Sociology of Monsters: Essays on Power, Technology and Domination*. ser. Sociological Review Monograph, J. Law, Ed. London, U.K.: Routledge, 1991.

[5] S. Strum and B. Latour, *The Social Shaping of Technology*, 2nd ed. Berkshire, U.K.: Open University Press, 1999, ch. Redefining the Social Link: From Baboons to Humans.

[6] M. Callon, *The Social Construction of Technological Systems*. Cambridge, MA: MIT Press, 1987, ch. Society in the Making, pp. 17–50.

[7] O. Hanseth and K. Braa, "Technology as traitor: Emergent SAP infrastructure in a global organization," in *Proc. Int. Conf. Information Systems*, 1998, pp. 188–196.

[8] M. Callon, "Techno-economic networks and irreversibility," in *A Sociology of Monsters: Essays on Power, Technology and Domination*. ser. Sociological Review Monograph, J. Law, Ed. London, U.K.: Routledge, 1991.

[9] C. M. Christensen, *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Watertown, MA: Harvard Business School Press, 1997.

[10] L. Lessig, *Code and Other Laws of Cyberspace*. New York: Basic Books, 1999.

[11] M. Katz and C. Shapiro, "Systems competition and network effects," *J. Economic Perspectives*, vol. 8, no. 2, pp. 93–115, 1994.

[12] J. von Neumann and O. Morgernstern, *The Theory of Games and Economic Behavior*. Princeton, NJ: Princeton Univ. Press, 1944.

[13] J. F. Nash, "Equilibrium points in n-person games," *Proc. National Academy of Sciences*, vol. 36, pp. 48–49, 1950.

[14] S. Wolgar, "Configuring the user, the case of usability trials," in *A Sociology of Monsters: Essays on Power, Technology and Domination*. ser. Sociological Review Monograph, J. Law, Ed. London, U.K.: Routledge, 1991.

[15] R. J. Perlman, "Network Layer Protocols With Byzantine Robustness," Lab. Computer Science, Massachusetts Inst. Technology, Cambridge, MA, Tech. Rep. MIT-LCS-TR-429, 1988.

[16] S. Savage, "Protocol Design in an Uncooperative Internet," Ph.D. dissertation, Dept. Computer Science, Univ. Washington, Seattle, WA, 2002.

[17] ——, "Sting: A TCP-based network measurement tool," in *Proc. 1999 USENIX Symp. Internet Technologies and Systems*, Boulder, CO, 1999, pp. 71–79.

[18] S. Savage, N. Cardwell, D. Wetherall, and T. Anderson, "TCP congestion control with a misbehaving receiver," *ACM Comput. Commun. Rev.*, vol. 29, no. 5, pp. 71–78, Oct. 1999.

[19] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 295–306.

[20] ——, "Network support for IP traceback," *IEEE/ACM Trans. Networking*, vol. 9, no. 3, pp. 226–237, Jun. 2001.

[21] (2003) P3P 1.0: A New Standard in Online Privacy. World Wide Web Consortium. [Online]. Available: http://www.w3.org/P3P/brochure.html

[22] M. Blaze, J. Feigenbaum, and J. Ioannidis, "The keynote trust-management system version 2," Internet RFC 2704, Sep. 1999.

[23] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proc. 17th IEEE Symp. Security and Privacy*, 1996, pp. 164–173.

[24] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "The COPS (Common Open Policy Service) protocol," Internet RFC 2748, Jan. 2000.

[25] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *J. Finance*, vol. 16, pp. 8–37, 1961.

[26] N. Nissan and A. Ronen, "Algorithmic mechanism design," in *Proc. ACM Symp. Theory of Computing*, Atlanta, GA, May 1999, pp. 129–140.

[27] J. Feigenbaum and S. Shenker, "Distributed algorithmic mechanism design: Recent results and future directions," in *Proc. 6th Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, New York, 2002, pp. 1–13.

[28] K. Binmore, *Essays on the Foundations of Game Theory*. Oxford, U.K.: Basil Blackwell, 1990.

[29] S. Besen and J. Farrel, "Choosing how to compete: Strategies and tactics in standardization," *J. Economic Perspectives*, vol. 8, no. 2, pp. 117–131, 1994.

[30] W. Lehr, "Compatibility standards and industry competition: Two case studies," *Economics of Innovation and New Technology*, vol. 4, no. 2, pp. 97–112, 1996.

[31] *Internet Economics*, L. McKnight and J. Bailey, Eds., MIT Press, 1997.

[32] *Broadband: Bringing Home the Bits*, Computer Science and Telecommunications Board, National Research Council, 2002.

[33] J. Rekhter, "EGP and policy based routing in the new NSFNET backbone," RFC 1092, Feb. 1989.

[34] D. D. Clark, "Policy routing for Internet protocols," Internet RFC 1102, 1989.

[35] K. Lougheed and J. Y. Rekhter, "Border gateway protocol (BGP)," Internet RFC 1105, Jun. 1989.

[36] X. Yang, "NIRA: A new Internet routing architecture," in *Proc. ACM SIGCOMM FDNA 2003 Workshop*, Karlsruhe, Germany, Aug. 2003.

[37] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. 18th ACM Symp. Operating Systems Principles*, Banff, Alberta, Canada, Oct. 2001.

[38] H. Nissenbaum. (2001) Securing trust online: Wisdom or oxymoron. *Boston Univ. Law Rev.* [Online]. Available: http://www.princeton.edu/helen/BU-final-trust.pdf

[39] S. L. Garfinkel. (2004) The Pure Software Act of 2006. [Online]. Available: http://www.technologyreview.com/articles/wo_garfinkel040 704.asp

[40] N. Economides, "The economics of networks," *Int. J. Industrial Organization*, vol. 14, no. 6, pp. 670–699, 1996.

[41] *Perspectives on the National Information Infrastructure: Ensuring Interoperability*, Computer Systems Policy Project, 1994.

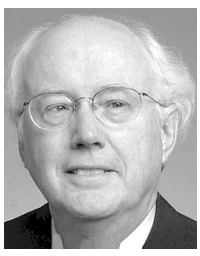
[42] *An Architectural Framework for the National Information Infrastructure*, Cross-Industry Working Team, 1994.

[43] *Realizing the Information Future: The Internet and Beyond*: Computer Science and Telecommunications Board, National Research Council, 1994.

[44] J. Saltzer, D. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Trans. Comput. Syst.*, vol. 2, no. 4, Nov. 1984.

[45] R. Levin, E. S. Cohen, W. M. Corwin, F. J. Pollack, and W. A. Wulff, "Policy/mechanism separation in HYDRA," in *Symp. Operating Systems Principles*, 1975, pp. 132–140.

[46] I. Goldberg, D. Wagner, and E. Brewer, "Privacy-enhancing technologies for the Internet," in *Proc. IEEE COMPCON*, 1997, pp. 103–109.



David D. Clark (F'98) received the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 1973.

He is a Senior Research Scientist at the MIT Computer Science and Artificial Intelligence Laboratory, where his work on the Internet started in 1975. Recent activities include extensions to the Internet to support real-time traffic, pricing and related economic issues, and policy issues surrounding the Internet, such as broadband local loop deployment. His current research looks at re-definition of the architectural underpinnings of the Internet, and the relation of technology and architecture to economic, societal and policy considerations.

Dr. Clark has been a Fellow of the ACM since 2001.

John Wroclawski (M'98) received the B.S. and M.S. degrees from the Massachusetts Institute of Technology (MIT), Cambridge.

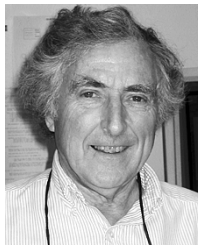
He is a Research Scientist with the Advanced Network Architecture Group at MIT's Computer Science and Artificial Intelligence Laboratory, and with MIT's Communications Futures Program. His interests include the architecture, technology and protocols of large, decentralized communication systems such as the Internet, systems aspects of pervasive computing, and the core principles of self-organizing systems. He is a former editor-in-chief of ACM SIGCOMM *Computer Communications Review*.

Mr. Wroclawski has been a member of the ACM since 1995.

Karen R. Sollins (M'85) received the B.A. degree in mathematics from Swarthmore College, Swarthmore, PA, and the M.S. and Ph.D. degrees in computer science from the Massachusetts Institute of Technology (MIT), Cambridge.

She is a Principal Scientist at MIT, where she does research on network architecture and infrastructure in support of distributed systems and applications. She also spent two years as a Senior Program Director for Networking Research at the National Science Foundation.

Dr. Sollins has been a member of the ACM since 1969.



Robert Braden (M'62) received the B.S. degree in engineering physics from Cornell University, Ithaca, NY, in 1957 and the M.S. degree in physics from Stanford University, Stanford, CA, in 1962.

He joined Carnegie Tech as Acting Assistant Professor at the Computer Center in 1962. Subsequently, he held a variety of software management, development, and research positions at Stanford (SLAC) and the University of California at Los Angeles (UCLA). In 1986, he joined the network research group at the University of Southern California (USC) Information Sciences Institute (ISI), Marina del Rey. He has worked on ARPAnet and Internet protocol design and implementation since 1970, under the sponsorship of the Defense Advanced Research Projects Agency (DARPA) and the National Science Foundation (NSF). His particular interests are end-to-end protocol and architectural issues.

Mr. Braden is a Fellow of the ACM and of ISI.