

Protocols

John Doyle

(Excerpts from “Robustness and the Internet”)

February 28, 2002

Abstract

This article uses the Internet as a starting point to illustrate universal aspects of complex systems throughout technology and biology. Complexity in most systems is driven by the need for robustness to uncertainty in their environments and component parts far more than by basic functionality. Protocols organize highly structured and complex modular hierarchies to achieve robustness, but also create fragilities to rare or neglected perturbations. All of this complexity is largely hidden, deliberately creating the illusion of superficially simple systems, which encourages development of specious theories. We claim these are also the most important and universal features of complex systems.

1 Introduction and motivation

The subject of complex systems is in desperate need of both sound theoretical ideas and concrete, canonical examples, and the Internet is in many ways ideal. The Internet shows unpredictable emergent phenomena and the fragility-complexity-robustness spiral characteristic of evolution in many complex systems. The Internet has elaborate (vertical) multiscale protocols interconnecting transistors, wires, and fiber through to computers and routers on which run a sophisticated collection of software to provide applications for the user and control of the network. Each level itself is complex and (horizontally) distributed. Thus evaluating theoretical claims is challenging, but because we know how all the parts work and are interconnected, and we can make detailed measurements, it is always possible to unambiguously diagnose and “reverse engineer” any claims or phenomena after the fact. This lets us separate sound from specious claims and theories.

We will first compare features of the design and evolution of the Internet with other complex systems familiar from everyday life. This is the first of two themes. If we have identified truly universal and necessary features of complex systems, then they should exhibit themselves everywhere, certainly in the various networks of communication, computing, energy, financial, transportation, etc, that surround us, but even in our laptops, toys, clothes, consumer appliances and electronics, automobiles, and homes. This theme requires no prior technical knowledge, but it would be useful to have a familiarity with the complexity and robustness features of Internet technology. It does require us to look carefully at familiar systems to see past the simple illusions that their complexity creates to their true but sometimes hidden nature. Some of the most cherished and appealing of traditional theoretical concepts from science and engineering are entirely misleading when applied to advanced technological systems. The central issue here is that advanced technologies (and biology) use protocols and feedback to create what amount to deliberate illusions regarding their systems-level behavior. Thus theories that do not fundamentally address protocols and feedback, which is essentially all theory in science and most of engineering, simply do not address the complexity aspects of systems, though they may be essential to understanding how the components function. While having a variety of concrete examples is useful in quickly discarding specious theories, they still leave open the question of necessity versus contingency. That is, what is essential, and what is an historical artifact. For that we need not just counterexamples, but correct theory.

While control and communications theory has played a crucial role throughout in designing aspects of the Internet, a unified and integrated theory of the Internet as a whole has only recently become a practical and achievable research objective. Dramatic progress has been made recently in analytic results that provide for the first time a foundation for a rigorous, coherent, and complete mathematical theory underpinning Internet technology. This new theory addresses directly the performance and robustness of both the “horizontal” decentralized and asynchronous nature of control

in TCP/IP as well as the “vertical” separation into the layers of the TCP/IP protocol stack from application down to the link layer. These results generalize notions of source and channel coding from information theory as well as decentralized versions of robust control. Perhaps most importantly for this collection, the new theoretical insights about the Internet combine with our understanding of its origins and evolution to provide a rich source of ideas about complex systems in general. Most surprisingly, our deepening understanding from genomics and molecular biology has revealed that at the network and protocol level, cells and organisms are strikingly similar to technological networks, despite having completely different material substrates, evolution, and development/construction.

Convergent evolution throughout biology and technology results in systems with universal features: elaborate hierarchies of protocols and modularity and layers of feedback regulatory networks, driven by demand for robustness to uncertain environments and often sloppy components. This complexity makes them robust to the uncertainties for which such complexity was selected, but also makes the resulting system potentially vulnerable to rare or unanticipated perturbations. This complexity is also largely cryptic and hidden in idealized laboratory settings and in normal operation, becoming conspicuous acutely when contributing to rare cascading failures or chronically through fragility/complexity evolutionary spirals. These puzzling and paradoxical features are an ongoing source of confusion to experimentalists, clinicians, and theorists alike, and have led to a rash of specious theories both in biology and more recently about the Internet. However, these “robust yet fragile” features are neither accidental nor artificial and derive from a deep and necessary interplay between complexity and robustness, modularity, feedback, and fragility.

The concept of Highly Optimized Tolerance (HOT) was introduced to focus attention on the robust yet fragile character of complexity. This is the most essential and common property of complex systems in biology, ecology, technology, and socio-economic systems. HOT offers a new and promising framework to study not only network problems, but also put networks in a bigger context. This will be important both with the convergence of existing communication and computing networks and their widely proposed role as a central component of vast enterprise and global networks of networks including transportation, energy, logistics, etc. Research within the HOT framework addresses many complementary aspects of the multifaceted area of networked complex systems. Issues such as robustness, scalability, verifiability and computability can now (and must) be investigated and understood within a common framework. We believe that these different requirements are not only compatible, but can be combined together in a very natural fashion. Our technologies have a priori emphasized these as separate issues, and the promise of a unified approach to simultaneously handle these critical aspects is of paramount importance. While seeking decomposition of hard problems into simpler subproblems will continue to be an important design strategy and is at the heart of protocols’ role, an integrated and unified theory is required to do this in a rigorous and robust manner.

2 Universal features of complexity and robustness

What emerges from the study of the design, evolution, and future challenges of the Internet and related technologies is that the evolution of spiraling complexity, feedback regulation, robustness, fragility, and cascading failures are heavily intertwined, as is well known to biologists and engineers alike. Organization and design of advanced technologies suggest that there are universal principles linking protocols, modularity, and feedback control with the robust yet fragile nature of complex systems. Truly universal principles should manifest themselves in at least limited ways in scale model (toy) systems, as well as other aspects of our everyday lives, and this section will focus on familiar everyday examples in addition to the Internet.

The universal principles that we will focus on begin with the idea that a system’s complexity is driven far more by the need for robustness to uncertainty in its environment and component parts than by basic functionality. Thus protocols organize highly structured and complex modular hierarchies to achieve robustness, but also create fragilities to rare or ignored perturbations. The evolution of protocols can lead to a robustness/complexity/fragility spiral where complexity added for robustness also adds new fragilities, which in turn leads to new and thus spiraling complexities. The most powerful and also dangerous protocols involve feedback control, which also has the most mathematical and thus least widely understood theoretical foundations. Finally, all of this complexity is largely hidden and deliberately creates the illusion of superficially simple systems, encouraging development of appealing and accessible but completely wrong explanations and theories.

As we attempt to unravel this hidden complexity and robustness, the basic protocols and modularity are the most easily observable characteristics of complex systems generally, and not just the Internet. While meaning varies, modules generally are components, parts, or subsystems of a larger system that contain some or all of the following features: (a) identifiable interfaces (usually involving protocols) to other modules, (b) can be modified and evolved

somewhat independently, (c) facilitate simplified or abstract modeling, (d) maintain some identity when isolated or rearranged, yet (e) derive additional identity from the rest of the system. Internet modules are links, routers, hosts, and the various software components that implement features at various levels and locations throughout the Internet. Perhaps the most familiar software modules are for web browsing and serving and exchange of email.

We claim that protocols are far more important to complexity than are modules. Protocols and modularity are intrinsically complementary and intertwined concepts, but are important to distinguish. In everyday usage, protocols are rules designed to manage relationships and processes smoothly and effectively. If modules are ingredients, parts, components, subsystems, and players then protocols describe the corresponding recipes, architectures, rules, interfaces, etiquettes, and codes of conduct. Protocols here are rules that prescribe allowed interface between modules, permitting system functions that could not be achieved by isolated modules. Protocols also facilitate the addition of new protocols and organization into collections of mutually supportive protocol suites. A good protocol is one that supplies both robustness and evolvability. The TCP/IP protocol suite is the paradigmatic example.

2.1 LEGO protocols and robustness

Consider the ubiquitous LEGO toy system. The signature feature of LEGO is the patented snap connection for easy but stable assembly of components. The snap is the basic LEGO protocol, and plays the role in LEGO analogous to what IP plays in the Internet. LEGO bricks are the basic modules and correspond to Internet links, hosts, and routers. The snap forms the thin waist of a LEGO hourglass protocol stack where at the bottom a huge variety of alternative brick modules (link and physical layers) can be interconnected via the snap protocol to create at the top a virtually unlimited number of toys (applications). "Everything on IP, IP on everything" becomes in LEGO, "All toys are built using the snap, the snap is implemented in all components." Thus we can use LEGO in two complementary ways. One is to broaden the discussion of the Internet by illustrating the universality of the features of Internet protocols. The other is as a gentle introduction to the concepts of protocols themselves, and thus instead as a starting point for studying Internet protocols.

The obvious parallel between LEGO and TCP/IP is that the most critical elements of the protocol stack, the snap and IP, form the thin waist of an hourglass-like protocol stack. This waist mediates between broad layers of lower level physical modules (bricks, physical and link layer) and equally broad higher level functionality (applications and toys). Note also that in normal operation, the snap and IP are largely hidden. A finished LEGO toy is obviously composed of brick modules but the snaps in between brick are completely hidden from view. Only unused and unnecessary snaps remain visible. Similarly, Internet applications are obviously run on computer hardware, which are in turn obviously connected by various link technologies, but the role of TCP/IP is completely hidden in normal operation. One can imagine a variety of specious theories of LEGO that ignore the role of the snap protocol, just as there are popular and specious theories of the Internet that ignore TCP/IP.

Similar to the Internet, LEGO exhibits multilayer robustness, from components and toys to the product line. LEGO bricks and toys are robust to trauma, reusable, and the snap is versatile, permitting endless varieties of toys from an array of components. The modularity of LEGO is analogous to both the modularity of the link/physical layer technologies and the application technologies as well as the basic use of packets. All confer robustness in the form of flexibility to changing demands and needs. This modularity makes both a given LEGO collection and the entire toy system evolvable to changes in what one chooses to build, to the addition of new LEGO-compatible parts, and to novel toy designs. Evolution here is simply robustness to (possibly large) changes on long time scales. Low cost of modules and the popularity of the system confer other forms of robustness and evolvability, as lost parts are easily replaced, and enthusiasts constantly design new modules and toys.

The LEGO protocol also creates fragilities. Superficially minuscule damage to the snap at a key interface can cause an entire toy to fail. This is very unlikely in normal use, but it is a potential target for malicious attack or a source of rare but catastrophic failure (the consequence of which may be minor, this is after all a toy). In contrast, non-interfacing parts of bricks may be heavily damaged with minimal impact, and even a fatally damaged component is easily replaced once it is identified. The success of LEGO means that any new, even superior snap would not be easily adopted, and the difficulty in evolving to IPV6 is mirrored in the difficulty that would accompany a change in the LEGO snap. Selection pressures thus preserve a protocol in two ways: Protocols facilitate evolution and are difficult to change.

As systems become more complex, protocols facilitate layering of additional protocols, particularly involving feedback and signaling. In analogy with the proposed evolution of networking, as well as biological evolution, suppose

we want to make a LEGO structure incrementally more useful and versatile by "evolving" it to be (1) mobile then (2) motorized then (3) able to avoid collisions in a maze of obstacles. The first increment is easy to achieve, with LEGO protocol-compatible axles and wheels. Motorizing toys involves a second increment in complexity requiring protocols for motor and battery interconnection as well as a separate protocol for gears. All can be integrated into a motorized protocol suite, to make modular subassemblies of batteries, motors, gears, axles, and wheels. These are available, inexpensive additions. This is roughly analogous in the Internet to the evolution of new link layer technologies, including faster routers and links, which in turn facilitate new applications that utilize the higher available bandwidth.

The third increment to create a more autonomous, collision avoiding robot, increases cost and complexity by orders of magnitude, requiring layers of protocols and modules for sensing, actuation and feedback controls plus subsidiary but essential ones for communications and computing. All are available, but it is here we begin to see the true complexity of advanced technologies. Unfortunately, we also start to lose the easily described, intuitive story of the basic protocols. Minimal descriptions of advanced LEGO features enabling sensing and feedback control literally fill books, but the protocols also facilitate building elaborate, robust toys, precisely because this complexity is largely hidden from users. This is consistent with the claim that complexity generally is dominated not by minimal function, but by the protocols and particularly regulatory feedback loops that provide robustness and evolvability. Imagine that a LEGO robot was a prototype for a single toy that dispensed entirely with the LEGO modules in favor of custom implementation. This toy could easily have much more robustness to trauma, be faster, and navigate more complex obstacles, but at the expense of limited part reuse. The modules and lower level protocols would be completely different, yet we might claim that the essence of the toy, and what the prototype aimed to capture, remained. That essence involves the protocols that organized the sensors, actuators, and feedback control system that enables the obstacle avoidance, and contributes almost the entire cost and complexity.

The added complexity of feedback control is absolutely necessary for robust collision avoidance, but also unavoidably creates new and potentially extreme fragilities. While removing a collision-avoiding toy's control system might cause reversion to mere mobility, a small change in an otherwise intact control system could cause wild, catastrophic behavior. For example, a small software bug might easily lead to collision *seeking*, a fragility absent in simpler toys. Note also that a simple test of mobility, without the additional challenge of robust collision avoidance, would not distinguish a highly complex robot from a much simpler one without collision avoidance. Furthermore, disabling collision avoidance would not produce any "phenotype" in a test of mere mobility. It would not imply that collision avoidance components were in any sense "redundant," but merely that they supplied a form of robustness not required by a simple mobility test. These issues arise to some extent in TCP in the feedback control of congestion, but will become much more acute if networks are used in embedded applications for control of physical systems, particular those that involve critical infrastructure.

The snap protocol is only concretely instantiated in LEGO modules, but it is also easy to identify the protocol itself as a useful and informative abstraction. Similarly, TCP/IP may be instantiated only in hosts and routers, but we have no difficulty in thinking of the protocol as an object of study. The snap protocol is thus much more fundamental to LEGO than are any individual modules. Similarly, we have no trouble distinguishing the many higher-level protocols that organize LEGO sensing and feedback from the hardware modules themselves. Good protocols allow new functions to be built from existing components, and new components to be added or to evolve from existing ones, powerfully enhancing both engineering and evolutionary "tinkering." Protocols enable modularity and robustness but are in turn sources of fragility. Successful protocols become highly conserved since they both facilitate evolution and are difficult to change.

The snap protocol itself severely constrains the interconnection of bricks, but the set of useful toys is even more severely constrained and highly structured. Consider the set of all the possible interconnections of a given collection of Lego parts. This is a (combinatorial) huge set. The set of interesting toys is also a large set, but an infinitesimally small subset of what is possible. They are very special and highly structured. Similarly, among the potential toy systems that could conceivably be created using the same basic plastic material, LEGO is highly structured and finely tuned. At the component level, the snap coupling is very precisely machined. Robustness of the type exhibited by LEGO toys and the LEGO system is achieved by fine tuning of highly structured components and interfaces at every level. This becomes most acute in the software used to control a LEGO robot. Thus "robust" and "fine-tuned" are in no sense opposites. Indeed, quite the contrary, highly robust systems are necessarily finely tuned to achieve exactly this robustness.

2.2 Protocols and robustness in computer technology

LEGO Mindstorm owners can build sophisticated robots that use computer vision and other sensors to perform complex feedback control tasks such as avoiding collisions with some objects while seeking out and moving other objects. The brains of a LEGO robot is a microprocessor module that has various protocols that interface with sensors, actuators, and an infrared transmitter that allows communication with remote computers. This communication can be used for a variety of functions, including remote or cooperative sensing and actuating, or for downloading programs using software written on a standard PC. It is possible for a network of LEGO robots distributed around the world to communicate via infrared connections to PCs which in turn can communicate via the Internet.

A huge variety of LEGO, PC, and Internet protocols and modules are used in allowing remote communication between robots. For example, in the PC the central separation is into hardware and software and further into an hourglass protocol stack with the operating system (OS) at the waist. The OS is intended to allow a huge variety of applications software to interact transparently with vast range of hardware components. In principle, new software and hardware need only be compatible with the OS protocols and they immediately interface with all existing modules for free. Hardware and software are further layered with protocols. For example, in PC hardware the control and interconnection of the CPU, memory, and the huge variety input/output (I/O) devices is handle by a chip set controller. The memory is layered into a speed/size/cost hierarchy of caches and storage technologies. I/O is organized by a variety of protocols interfacing to an almost unlimited collection of components. At the lowest level, digital circuitry is built from an analog substrate. This digital abstraction allows the circuit functionality and interconnection to be separate from the physical analog substrate, facilitating parallel and independent evolution of each. Software is similarly modularized using a variety of programming protocols established for this purpose.

Just listing in the briefest terms all the LEGO, PC, and Internet protocols and modules that would be used in allowing remote communication between robots would fill a book, and the details would fill a library. As the sophistication of users and their robots increases, the emphasis shifts from LEGO as a mechanical toy system to LEGO as a programming and control system. While it is possible to manually program some functionality directly into the LEGO microprocessor, there are a variety of tools that allow the user to develop detailed virtual prototypes on their PCs of both the mechanical and software systems to be built from LEGO components. The total time and cost of creating a real LEGO robot (or almost any engineering system) can be greatly reduced using virtual prototyping, and the engineering of most complex systems, from autos to planes to refineries to space vehicles now relies heavily on virtual prototyping. While the total costs go down, the percentage of costs and complexity that is present in the virtual prototype can be substantial and come to dominate the design of a complex LEGO robot, to the point where actual construction and operation of the physical robot can be a vanishingly small part of the whole effort.

The essential challenge in design a complex robot is robustness, and when using virtual prototyping that the physical implementation behaves like its idealized virtual prototype. This is perhaps the most important lesson to draw from these examples, and its implications can be crystallized by a simple thought experiment. Suppose someone has already built a sophisticated and robust robot that performs some interesting task, and it is our job to reverse engineer this robot. Suppose we can observe the robot's behavior and can study the interconnection of LEGO parts, but do not have access to the software implementing the control system or any hints on the control system design. It will be completely straightforward by inspection to make a copy of the entire robot except for the control software.

To reverse engineer the control software it will be very useful to explore alternatives using a virtual prototype on a PC, and here is where the complexity will be revealed. It will often be quite straightforward to produce a convincing virtual simulation of the robot, yet have the resulting downloaded robot software fail miserably. That is because the complexity is dominated by robustness, and not the obvious basic functionality. It is possible to have a working virtual prototype that fails when implemented because of inadequate robustness, or works intermittently or even most of the time, but has occasional catastrophic failures not present with the original. Control theory is such a complicated and mathematically intensive subject primarily because it aims to address this issue in a systematic way. Its goal is to provide mathematical and software tools that insure that real systems are robust, and work like their virtual prototypes. It has played an important role in Internet technology, and we expect this role to expand.

This example also illustrates the dangers in naive reliance on simulation. It is possible to produce simulations of complex systems that are superficially similar but entirely fail to capture the complexity and robustness of the real system. This is because the complexity is largely hidden and is only *for* robustness. For an Internet example, suppose we are browsing a remote website on our laptop. It is easy to download the entire website from some remote location and then disconnect the laptop from the Internet, and browse the website purely using the local copies. This will be largely indistinguishable from the process of browsing the original website remotely, until something changes or

we want to link to some other website (this process is the idea behind web caching). In a sense, we have created an extremely faithful simulation of a complex interaction with the Internet, but now without any of the complex protocols and modules that make up the Internet. While it might be tempting to imagine we had captured some simple essence of the Internet, nothing could be further from the truth. We would have merely captured some trivial and superficial features of one application that runs on the Internet.

2.3 Clothing, money, options, and crashes

The universal features of complexity, protocols, and spiraling robustness and fragility can be seen in every aspect of our modern lives. Our various energy, transportation, consumer, financial and social networks provide astonishing functionality but are also vulnerable to large cascading failures often initiated by small malfunctions or deliberate disruption by a small number of attackers. We read of precision weapons capable of hitting individual rooms or vehicles thousands of miles from the launch site, but which are equally sensitive to errors in targeting protocols and can thus destroy friendly objects as well. Medical marvels are balanced by the horrors of new viruses and antibiotic resistant bacterial pathogens.

More mundane examples surround us. For example, the process of creating clothing is organized into an hourglass-like protocol stack. The thin waist is supplied by the process of sewing, which can integrate various physical layer cloth technologies at the bottom, into a variety of garments at the top. This can create the illusion of a single, “seamless” garment, hiding the protocols turning raw fibres into thread, threads into cloth and seams, and the resulting sewn elements into garments and ensembles. The seam is typically the source of greatest fragility, and can sometimes be unravelled with a minimal perturbation. This protocol stack facilitates the creation of robust garments, but also introduces entirely new fragilities. In particular, it enables fashion, which can cause a garment to become obsolete long before it has ceased to perform its basic function.

Money is the waist in an hourglass protocol stack connecting, say, widely varied interests of consumers at the top with an equally vast choice of commodities at the bottom. Compared to a straight barter economy, money provides numerous benefits, but also creates fragilities, such as the relative ease of counterfeiting money compared with tradeable goods. This in turn drives the development of highly complex currencies that are hard to counterfeit. Money also facilitates the creation of currency markets and other vast financial markets connecting investors with investment opportunities. This evolution of complexity can exhibit dramatic robustness and fragility spirals. Derivatives such as futures and options allow investors and producers to hedge their positions, for example against currency fluctuations, and more effectively manage risks. They also allow aggressive corporate managers to manipulate their apparent financial status, which can mislead investors. In an attempt to control such practices, accounting rules have undergone an exponential explosion that parallels that in information technology. The evidence suggests, however, that the resulting complexities of this system have led to greater and not less obfuscation, and may have contributed to recent dramatic bankruptcies such as Enron and Global Crossing. This complexity is exacerbated by the increasingly intertwined nature of financial institutions and accounting and management consulting.

Our energy use is organized into an hourglass protocol stack with a huge variety of user devices such as appliances, heating and cooling systems, consumer electronics, office equipment, at the top all using the small waist of a common currency of 110 volt, 60 hertz alternating current and a standard plug format. This can in turn be supplied via a transmission and distribution network, which is governed by its own protocols. Finally, the energy can be generated by a variety of energy sources, provided they follow the appropriate physical layer protocols. Gasoline provides a similar common currency for connecting energy producers with consumers. These efficient energy protocols have facilitated massive deployment of vehicles and other devices in homes and factories, but have created myriad new fragilities. Perhaps most ominous is the impact the pollution from energy sources is having on the global ecosystem.

Many of our systems are undergoing an explosion in complexity due to advanced controls and embedded computing and networking. A Boeing 777 has millions of parts, 150,000 distinct subsystems, including roughly a thousand computers. An example more familiar and accessible than an airplane is the automobile. Modern cars have dozens of microprocessors, automating and controlling every aspect of vehicle function. They host sophisticated engine controls to meet emissions and fuel-economy standards (spark timing, fuel/air ratio, transmission, etc), provide advanced diagnostics for maintenance, enhanced safety features (control of airbags, braking, and traction), and improved comfort and convenience (cruise control, internal environmental controls, GPS-based navigation, security and alarms, wireless emergency communications, etc). This is just the beginning of a trend to higher levels of automation and control. Obviously, much simpler vehicles without any of these features are available, but they pollute more, are less safe,

require more frequent and expensive maintenance, and will generally deteriorate more quickly. Complex control and computer networking can actually simplify some features such as wiring by using common components and network busses. Manufacturing and design is also simplified by allowing for highly modular design and the use of sloppier and cheaper mechanical components. This is all facilitated throughout by numerous standard protocols for interfacing modules.

Even the user interface in automobiles (steering wheel, pedals, keys, etc) is a protocol that standardizes most human/vehicle interactions, as are the traffic laws, signs, signals, lanes, and other mechanisms that control traffic. Nevertheless, highway traffic relies heavily on the human driver to perform sensing and control of vehicles. As in LEGO, a full automated highway system would require orders of magnitude more complexity than is in place currently. It is likely a fully automated system would greatly reduce the total number of serious injuries and fatalities in accidents, but because there would also be new fragilities subject to liability litigation, it is unlikely that such a system will be deployed anytime soon.

A more immediate example of spiraling fragility and complexity is the electronic control system that measures vehicle acceleration during a crash and determines whether to deploy an airbag. A seat belt simply provides specific restraining forces as a function of position, but the airbag is more complex. It is designed to deploy only when the vehicle dynamic state enters a certain regime that is potentially dangerous to the occupant, and there is an intrinsic tradeoff about exactly when and where to deploy the bag. Both too often and too rarely are dangerous, but for opposite reasons. An airbag that deploys during a relatively minor accident can cause trauma where none would have occurred without the airbag. Airbags still give, on average, an overall benefit, but actually make certain circumstances for certain occupants more dangerous, and have led to some unnecessary deaths. Automotive engineers are improving the airbag system with greater complexity to sense occupant size and status and more finely control airbag deployment. Most such robustness-producing mechanisms, such as anti-lock brakes, traction control, or automatic collision avoidance systems, are even more complex.

2.4 Biological complexity

Biological organisms exhibit extraordinarily elaborate hierarchical organization of protocols and modularity, and use feedback control even more ubiquitously and aggressively than does any technology. We are aware of organism's external behaviors, and molecular biology has catalogued many of the elementary components, but the layers of protocols and modularity that connect the two have been far harder to discern. Again, these layers are the most critical features of complex systems but also the most cryptic and hidden, whereas the highest level behavior and the lowest level components are the most visible. Nevertheless, thinking in terms of protocols, in addition to genes, organisms, and populations, as foci of both natural selection and biological research, may be a useful abstraction for understanding the nature and evolution of biological as well as technological complexity. While biological cellular processes are far more distributed, stochastic, and heterogeneous than, say, VLSI circuits, in both cases, such circuits form merely the components of complex control, communications, and computing systems. Biology not only integrates these functions but also builds them directly at the molecular level.

Organisms and ecosystems also have extreme robust yet fragile characteristics. Life in all forms is remarkably robust to environmental fluctuations and tolerates substantial component uncertainty. Ecosystems can recover from massive change, yet be virtually destroyed by a single exotic species. Organisms have been found to grow and persist in almost every environment, and have evolved complexity that will take perhaps centuries to unravel. The control systems that enable such organisms are both critical and largely hidden except when they fail. Even then, large multicellular organisms are unaffected by the death of individual cells, but certain malfunctions in one or a few cell's control systems can lead to fatal autoimmune diseases or cancer.

The future of biology also has many parallels with the future of complex engineering systems. Emphasis is now shifting from components and molecules to the study of the vast networks that biological molecules create that regulate and control life. It is our claim, which this article will not attempt to justify, that while biology and technology have different system-level behavior and even more vastly different component parts, the organizational principles that govern the layers of protocols, modules, and feedback that lie in between are far more alike than is commonly realized. LEGO and the Internet are completely unlike, but they share certain essential features involving robustness, protocols, and feedback control, and these essential features are shared with biology. While there are perhaps many ways in principle to interconnect components to make complex systems, very few that are robust and thus likely to persist and be observed. This is both a recent phenomena and a recent discovery. Only in the last few decades have

technological systems (and toys) begun to approach biology in their level of complexity, and thus provide any solid basis for comparison. At the same time, molecular biologists have identified enough of the components and their interconnection that the nature of the intermediate level protocols have begun to emerge.

Biological complexity is a fascinating and timely topic and is subject to vigorous and expanding research attention. Unfortunately, even scratching the surface is well beyond the scope of this article, which will return to a deeper look at the Internet and related technology. Since the research needs for systems engineering and systems biology are converging, it is fortunate that the mathematical theory and software infrastructure to address these needs is finally an achievable goal. Central to this theory is a growing understanding of the "design principles" of complex networks, and the role of protocols, modularity, interconnection, and feedback in creating robust, evolvable systems.

2.5 Evolving internetworking challenges

Many popular technological visions emphasize ubiquitous control, communications, and computing, with systems requiring high levels of not only autonomy and adaptation, but also evolvability, scalability, and verifiability. With current technology these are profoundly incompatible objectives, and both biology and nanotechnology create additional novel multiscale challenges. A rigorous, practical, and unified theoretical framework will be essential for this vision, but until recently, has proven stubbornly elusive. Two of the great abstractions of science and technology have been the separation, in both theory and applications, of 1) controls, communications, and computing from each other, and 2) the systems level from its underlying physical substrate. These separations have facilitated massively parallel, wildly successful, explosive growth in both mathematical theory and technology, but left many fundamental problems unresolved and a poor foundation for future systems of systems in which these elements must be integrated. This horizontal and vertical isolation of systems is the heart of reductionism. Science has focused almost exclusively on understanding the details of physical substrates, whereas technology has increasingly balanced this with an emphasis on systems, and particularly those aspects largely independent of the physical substrate, such as protocols and software.

Our lives are increasingly dominated by our interaction with a wide variety of networks, in transportation, energy, health, utilities, finance, politics, as well as voice, video, and data, which in turn also interact with our local and global environment. These currently disjoint networks will be increasingly integrated, using ubiquitous embedded computing, into a single convergent network of networks. This creates the opportunity for both unprecedented promise and risk. A lightning strike in one state can cause a power outage in another state far away, a hacker on another continent can deny web access, a single firm can trigger a global financial crisis, a software bug can cause a rocket, airplane, or automobile to crash. Terrorists can turn the power of one network against another, causing tragedies of global proportions. Because the associated networks have remained fairly isolated from each other, such events can have large, but still limited impact. This is changing, and will continue to do so.

While novel human-computer interfaces will transform the way we interact with machines and even with each other via networks, even more applications will also involve devices such as sensors and actuators that interact with the physical environment, with requirements much less forgiving than human users. An extra dimension in this context comes from the problem of designing distributed real-time control to be implemented on networks, adding control *over* networks to the existing substantial challenge of robust control of the network flows themselves. Finally, networks of networks, where communications, computing, and control are deeply embedded, creates new challenges in both cooperative operation, and the containing of catastrophic, cascading failure events. Indeed, perhaps one of our greatest national security threats will be the increasing vulnerability of our critical infrastructure to both cascading failure and deliberate attack.