Evolvable network architectures: What can we learn from biology?

Constantine Dovrolis
College of Computing
Georgia Tech
dovrolis@cc.gatech.edu

J. Todd Streelman
School of Biology
Georgia Tech
todd.streelman@biology.gatech.edu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

There is significant research interest recently to understand the evolution of the current Internet, as well as to design clean-slate Future Internet architectures. Clearly, even when network architectures are designed from scratch, they have to evolve as their environment (i.e., technological constraints, service requirements, applications, economic conditions, etc) always changes. A key question then is: what makes a network architecture evolvable? What determines the ability of a network architecture to evolve as its environment changes? In this paper, we review some relevant ideas about evolvability from the biological literature. We examine the role of robustness and modularity in evolution, and their relation with evolvability. We also discuss evolutionary kernels and punctuated equilibria, two important concepts that may be relevant to the so-called ossification of the core Internet protocols. Finally, we examine optimality, a design objective that is often of primary interest in engineering but that does not seem to be abundant in biology.

Categories and Subject Descriptors: C.2.5 [Computer Communication Networks]: Internet

General Terms: Theory

Keywords: Robustness, Modularity, Evolvability, Evolutionary Kernels, Punctuated Equilibria, Internet

1. INTRODUCTION

The Internet has been evolving in a self-organized manner, without centralized control, over the last 40 years: from the initial studies of packet switching in the 60's, the development of TCP/IP and Ethernet LANs in the 70's, the introduction of DNS and TCP congestion control in the 80's, the adoption of BGP and CIDR in the early 90's and of NATs in the late 90's, the introduction of new access technologies in the last mile during the last decade (broadband, wireless LANs, WiMAX, etc) and of course the persistent expansion of application-layer protocols and services (peer-to-peer, VoIP, video streaming, multiplayer gaming, online social networks etc).

Recently, the evolution of the Internet architecture has become a major research issue for two reasons. First, there is a concern that the current Internet architecture has been ossified, meaning that it does not evolve as quickly as it used to, especially at the core protocols (network and transport layers). Several architectural extensions to the current Internet, such as IP multicast, IPv6, QoS or new transport protocols such as XCP, have not been adopted in practice, perhaps as a result of this ossification. Second, there is a significant interest in designing clean-slate architectures for a Future Internet, without aiming to make these architectures backwards compatible with the current Internet. A central objective in these design exercises is evolvability, i.e., to ensure that an architecture will have the ability to evolve under new environments and requirements without getting ossified itself a few years later.

Evolvability is a central concept in biology [8]. Evolvability explains how life manages to generate dramatically novel phenotypes and complex behaviors from random genetic variations, in relatively short time scales. Biologists have developed a rich domain of experimental and theoretical knowledge to understand evolvability and to identify the design principles that make living organisms evolvable. These principles include robustness and modularity, two general design objectives that are closely related and that have deep implications for the evolvability of any system.

In this paper, we review the notions of robustness, modularity and evolvability in the biological domain, and discuss how they may be also relevant to the evolution of the Internet architecture as well as to the design of Future Internet architectures. We also identify two important concepts in evolution, evolutionary kernels and punctuated equilibria, which can explain why the core protocols in the Internet architecture appear as ossified: they may be constrained to evolve at a much lower rate so that they form a stable foundation for all other protocols and mechanisms that depend on them. Finally, we comment on another design objective, optimality, that is often of central importance in engineering designs, but that does *not* appear to be abundant in biology.

2. THE BIOLOGICAL ANALOGY AND THE ROLE OF SELECTION VERSUS DRIFT

Analogies and metaphors can be misleading. One may argue that the evolution of living organisms has very little in common with the evolution of a technological artifact such as the Internet. This argument, however, misses the point that the evolutionary process is independent of the underlying substrate, and it is equally applicable to biological,

This research is supported by the National Science Foundation under Grant No. 0831848.

natural, technological or computational systems, as long as few key principles hold. Herbert Simon was among the first to argue convincingly that the design of artificial systems should, and often is, subject to the same principles with natural and biological systems [11].

The evolutionary process, in its most general form, requires the following components: a) a population (a *species*) of elements (*individuals*) with each element specified by a *genotype*, b) a mechanism through which the genotype of existing individuals is transferred, subject to variation (various types of *mutations* or genetic *recombination*), into new individuals that can replace the old, c) a mechanism (*natural selection*) that gives a reproductive advantage to genotypes that perform better, i.e., that have *a phenotype with higher fitness* in the given *environment*.

Which are the previous three components in the case of the Internet? First, the Internet, viewed as a species, consists of millions of independent but interconnected computer networks (starting from home networks all the way to global tier-1 providers) - these networks are the individuals. Each network is characterized by its genotype: the protocols and operating systems it uses, deployed technologies, the applications that users run, and so on. Second, existing networks are constantly replaced by new networks, with the genotype of the latter strongly influenced by the genotype of their predecessors (e.g., if a home network's owner switches from DSL to cable access, the rest of the network's genotype would be identical to that of the original network). Finally, depending on the broader environment (available technologies and applications, economic costs, existing protocol stacks, etc), a given network genotype may lead to higher fitness than another genotype (e.g., the fitness of the X.25 protocol today would be much less than that of IPv4 - this was not necessarily true in the 80s).

Biologists recognize four evolutionary forces that can change the frequency of gene variants in a population. *Mutation* provides the raw material, in the form of genetic variation, for evolution to occur. Without genetic variation among individuals there can be no evolutionary change. Migration is the movement of individuals (and their genes) from place to place this is also called *gene flow*. Darwin recognized *natural selection* as the main driving force of evolution. Natural selection is a deterministic process, by which traits that increase the survivorship or reproductive output of individuals (*fitness*), increase in frequency over time. *Genetic drift* is a stochastic process, by which gene variants in a population change in frequency due to random sampling and chance. Whereas natural selection will increase the frequency of advantageous gene variants, genetic drift can cause advantageous, neutral or deleterious variants to become more common.

There is much debate about the relative roles of natural selection vs. genetic drift in evolution. What is clear is that *the size of a given population* determines the efficiency with which natural selection can act - random effects of genetic drift are stronger in small populations. This is similar to the sampling effects of flipping a biased coin 10 vs. 10,000 times the role of chance will be more dramatic in a small sample.

The issue of selection vs. genetic drift may be quite relevant to the evolution of the Internet. Back in the early days of the Internet, when its population was just a handful of networks, it was easy for the architecture to "drift" one way or another, simply due to chance. For instance, the adoption of early TCP header fields such as the rarely used URGENT pointer, can be explained as the result of drift (given that its practical use was and still is quite low). Today, the Internet consists of millions of networks and there is immense pressure to only adopt changes if they would have positive impact on the networks fitness, i.e., selection strongly dominates over drift in the current Internet. It should not be surprising that some recent architectural proposals (such as the deployment of IP multicast or QoS) have not been adopted in the current Internet, as they do not provide increased fitness to the network(s) that would first deploy them.

3. ROBUSTNESS

Robustness is one of the most prevalent design principles in biology. It is typically defined as the *invariance of certain phenotypic characteristics in the presence of wide, but bounded, internal or external perturbations.* Internal variations are caused by genetic variability, such as mutations, or by variations in the biochemical parameters of cellular mechanisms. External variations are caused by fluctuations in the environmental conditions (e.g., temperature or concentration of nutrients). Robustness is always defined in terms of specific phenotypic characteristics - it is not a generic attribute of a system. Also, robustness is defined over a range of perturbations - no design would be capable to tolerate any perturbations.

Robustness is often described as stability, homeostasis, or canalization, but these properties are really only special cases of robustness. For instance, a developmental process should be robust even though its dynamic nature means that it cannot be stable. Also, robustness requires the *invariance of function*, not the invariance of structure. A robust organism can be significantly plastic or variable, modifying its internal structure to cope with the given perturbations.

One important point that we can learn from biology is that even though there are many ways to provide a given functionality, very few of these pathways are actually robust for a wide range of perturbations. This has been shown, for instance, in the case of bacterial chemotaxis [1]. Biologists use robustness as a criterion for selecting between different models that can explain the same biological effect: if one model leads to robustness with respect to some key parameters, while another model requires fine tuning of those parameters, the former may be more plausible in certain circumstances.

Why would we expect robustness to be significant for a technological system such as the Internet? If a system operates under a static environment, and its internal components always perform as designed, robustness would be an unnecessary and costly property. Biological systems operate and evolve, however, under unpredictable environments with extremely noisy components. Is the Internet different in that aspect? An artifact with the Internet's size, heterogeneity and distributed control also faces frequent and major changes in both its environmental conditions (e.g., applications mix, offered workload or broader economic conditions, attacks by malicious users) and in its internal elements (e.g., routers and links, protocols). Arguably, a major reason behind the Internet's success is the robustness of the TCP/IP architecture with respect to topological changes and overload conditions.

The link between robustness and evolvability is critical. It may appear at first that these two properties contradict each other: if a system can maintain its function under wide environmental changes, then it will not need to evolve. And if a system is able to evolve when the environment changes, why does it need to be robust? In reality there is no contradiction. Evolution takes place over long timescales, requiring many generations as well as the appearance and/or shuffling of genetic variation. Robustness is important for viability under unfavorable environments, while evolvability allows a system to adapt if/when conditions improve.

How do biological systems achieve robustness? It appears that they deploy mostly three mechanisms. The first is the extensive use of modularity and hierarchical design - this is an important design principle by itself, and is the subject of the next section. The two other mechanisms are feedback control and degeneracy, reviewed next.

3.1 Feedback control

The basic idea of feedback control is to use the output of a mechanism to determine the appropriate input to that mechanism. In the case of negative feedback, the mechanism's input is the difference between the observed output and a desired, reference level, which can vary with time. A system that deploys negative feedback control can achieve robustness to external or internal perturbations if the timescales in which the input varies are larger than the timescales of the feedback controller. A particular type of negative feedback, referred to as *integral feedback*, integrates the error between the output and the reference input over time, and it attempts to minimize it. Integral negative feedback control is used, for instance, in bacterial chemotaxis.

Biological systems also use positive feedback. Positive feedback can force a system to one of several stable states and remain in that state even in the presence of significant noise in the inputs. Such positive feedback loops are common in *network motifs*, i.e., frequently occurring subnetworks of Gene Regulatory Networks (GRNs) that are used in development [9]. For instance, cell differentiation is a process that should result in a distinct and stable type for each cell even if there are major variations in the internal kinetic parameters of the corresponding GRNs. A well-studied instance of such biological robustness is the so-called "segment polarity" network that determines the body segmentation of the fruit fly (*Drosophila*) [12]. The corresponding GRN relies on several nested feedback loops, some of them positive, and it results in robust body segmentation even if the underlying kinetic parameters span several orders of magnitude.

The use of feedback control, especially negative feedback, is common in certain computer network protocols (e.g., TCP congestion control and various AQM mechanisms) but less understood in others (e.g., robustness of online traffic engineering or nested feedback loops that involve both the transport and application layers). Also, the use of positive feedback in enhancing robustness has not been explored enough. On the contrary, positive feedback is often avoided in networking research due to the potential for instability.

3.2 Redundancy and degeneracy

In biological systems, robustness is also provided through extensive redundancy at all levels, from the encoding of the genetic code inside a cell all the way to how animals move or communicate. What is more interesting, however, is that redundancy in biology does *not* simply result from multiple copies of a certain structure or mechanism, as often done in engineering. Instead, biological systems accomplish redundancy via *structurally different mechanisms* that perform the same function (or highly overlapping functions) and that operate independently and simultaneously. To distinguish the two approaches, we adopt Edelman's terminology and refer to redundancy as just the replication of the same structure, while *degeneracy* means that a given function is provided by different structures and mechanisms [3].

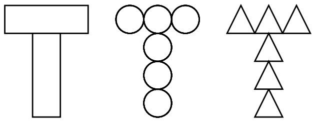

Figure 1: Suppose that we start with three basic building blocks: rectangle, circle and triangle, and we want to create a structure that functions as the letter T. Degeneracy would require that we create different structures with the same function, as shown above.

Degeneracy is more effective than redundancy in terms of robustness, even though it would probably be viewed in engineering as more expensive and complex. If a structure or mechanism that relies on redundancy fails under certain conditions, all replicas of that structure will fail simultaneously when those conditions are encountered. Two important examples of biological degeneracy are the existence of multiple parallel metabolic pathways for the same function, or the activation of multiple immune responses (populations of different antibodies, for instance) in the presence of a pathogen.

The connection between degeneracy and evolvability is intriguing. First, degeneracy may increase evolvability. This is thought to be the case, for instance, when genes are duplicated within genomes and then accumulate independent mutations. Second, degeneracy may naturally result from the evolutionary process. If a function is important for viability, several mechanisms may evolve independently to provide it.

Even though redundancy is commonly used in computer networks, especially to protect physical links or network paths from failures, the use of degeneracy, its differences with redundancy, and the evolutionary consequences of degeneracy for a network architecture, are largely unexplored.

4. MODULARITY

Modularity is an important tool in the design of complex systems. Each module represents a relatively autonomous part of the overall system that is responsible for a certain function or objective. Different modules are connected together to create more complex functions and systems. Topologically speaking, if we think of a design as a network, the density of intra-module edges is much higher than the density of inter-module edges. A closely related concept is that of *hierarchical design*. Two or more modules are combined to form more complex modules at a higher architectural level.

Modularity is also abundant in biology [5]. It has been observed in Gene Regulatory Networks, metabolic networks, protein-protein interactions networks, and networks of species in ecosystems. In the case of GRNs, for instance, an important recent discovery was that complex GRNs can be represented as hierarchical "biological circuits" known as *network motifs* [9]. A network motif is a subgraph that appears in an actual network with much higher frequency than in a random network of the same degree distribution. Important motifs, such as the FeedForward Loop, the BiFan or the Dense-Overlapping Regulon, consist of a small number of interacting genes and they accomplish specific functions (such as noise suppression or pulse generation). Evolution seems to have selected surprisingly few motifs for their functions in building more complex systems.

Modularity is closely related to robustness. Modularity partitions a complex design into roughly independent components, a property also known as *near-decomposability.* Noise, perturbations or failures within one module are not as likely to affect another module. Internal state variations within a module would be irrelevant to another module, unless they affect the relatively few links between the two modules.

Recent modeling work in biology (mostly using computational models of simple evolutionary processes) has provided important insights about the relation between modularity and evolvability. First, it appears that modularity enhances evolvability: modular designs can evolve faster and they can lead to more complex designs than "monolithic" designs [7]. Modular designs often evolve simply by re-arranging the inter-module connectivity, not the internal design of individual modules. In developmental biology, minor "rewiring" of developmental GRNs can cause major phenotypic variations, and it can even explain in some cases the evolution of new species [2].

Second, evolution produces modularity when the environment changes with time, following *Modularly Varying Goals (MVGs)* [7]. Evolution under a constant goal, or under a constant environment, leads to optimized but non-modular designs. The reason is that, under a constant environment, the evolutionary process gradually removes redundant functionality that can be somehow provided by other modules, increasing the density of inter-module linkages. It is those additional linkages that gradually break modularity. The situation is quite different when the environment changes with time, and especially when the environment presents Modularly Varying Goals. With MVGs, the objectives at any point of time can be decomposed into simpler goals and sub-functions; the exact hierarchy of those goals changes with time, but the underlying elementary building blocks remain the same. Then, the evolutionary process selects modular designs in which each underlying module implements an elementary goal, while the time-varying more complex goals are accomplished by rewiring the existing underlying modules. The simulations of [7] have further shown that evolution under MVGs proceeds much faster than evolution under a constant goal or under randomly varying goals.

The previous interactions between modularity, robustness and evolvability are relevant to both the evolution of the existing Internet architecture, as well as to the creation of new but evolvable network architectures. First, we should accept that modularity and hierarchy are necessary tools to deal with complexity. Recent proposals for non-layered network architectures, or for cross-layer optimizations that reduce the modularity of the overall system, are probably moving in the wrong direction. Optimizing one particular metric, such as power consumption or network capacity, can be much less important than the long-term evolvability and robustness of the proposed architecture. Second, network architectures that are designed to cope with a time-varying environment, in terms of constraints, available technologies, costs, application requirements, etc, should be expected to evolve faster when they re-arrange how existing modules are used, compared to architectures that rely on completely new modules and functions every time the environment changes.

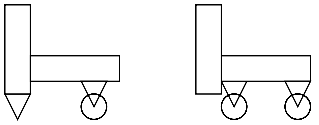

Figure 2: A basic diagram of a plow and a stroller. Even though the two structures have very different function, they consist of the same sub-modules, interconnected in different ways. An environment that presents Modularly Varying Goals (MVGs) can speedup the evolutionary process and it results in modular designs.

5. EVOLUTIONARY KERNELS

There is increasing biological evidence that certain genes, and in particular, certain Gene Regulatory Networks, have seen very few changes over almost half of the duration of life on this planet. For instance, certain developmental GRNs were established in the Early Cambrian (about 510 mya) and they have not evolved significantly since then [2]. These GRNs are referred to as *kernels*, and it is now understood that they are largely responsible for major aspects of animal body plans. The existing phylum-level body plans may have been shaped, to a large degree, from the action of these kernel GRNs. For instance the heart of a fruit fly and the heart of a human, despite distinct morphologies, develop using the same core cardiac GRN.

Why are kernel GRNs resistant to evolutionary change? In the words of Davidson and Erwin, "change in them is prohibited on pain of developmental catastrophe, both because of their internal recursive wiring and because of their roles high in the developmental network hierarchy" [2]. Evolutionary kernels represent a stable basis on which diversity and complexity of higher-level processes can evolve. On the other hand, kernel GRNs constrain the major axes of development and morphology and so, inevitably, they limit the scope of body plans that can subsequently evolve.

Evolutionary kernels may also exist and can play a similar role in the evolution of technological artifacts such as the Internet architecture. IPv4, the central protocol in the Internet, was the first protocol (together with TCP) that was created in the Internet architecture and it has evolved very slowly since then. Even when certain IPv4 header fields were re-defined (such as the transition from the Type-of-Service to the Differentiated Services (DS) field) those changes were always backwards compatible without requiring any change to existing networks.

It is plausible that IPv4 plays the role of an evolutionary kernel in the Internet architecture. It *binds* together the wide range of protocols and technologies at the physical and data-link layers, with the great diversity of protocols, applications and services at the higher layers of the architecture.

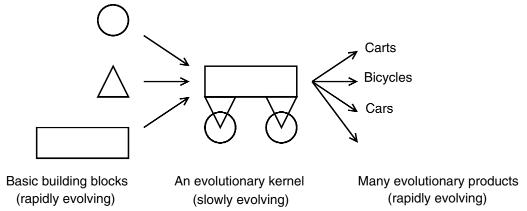

Figure 3: In this example, a frame on top of two wheels represents an evolutionary kernel. Many different components (at the left) can be used to construct this kernel, and many different products (at the right) can be produced using this kernel. Both substrate components and resulting products evolve much more frequently than the evolutionary kernel.

IPv4 can run over any link technology, and presumably any application can run over IP. This would be much harder, if at all possible, if IPv4 was subject to frequent changes, or if there were several protocols at the same layer with IPv4 (as that would create the need for "translation" between them). At the same time, the IPv4 protocol limits the space of possible protocols that can exist below or above it. It would be an interesting research project to identify somehow the boundaries of the services that can be provided over IPv4. If a future Internet architecture aims to replace IPv4, its best bet may be to support applications and services that are outside those IPv4 boundaries, assuming of course that there would be real interest for such services.

6. PUNCTUATED EQUILIBRIA

Today, most biologists agree that evolution does not proceed at a roughly constant pace (known as *gradualism*), but instead it alternates between long periods of *stasis* and relatively short periods of intense change in the existing species and in the underlying developmental mechanisms that produce those species [4]. This evolutionary pattern is referred to as *punctuated equilibria.* For instance, the "Cambrian explosion" refers to a relatively short time period about 530 mya in which the pace of evolution increased by a factor of ten leading to major species diversification. Recently, Jain and Krishna proposed an evolutionary computational model that shows how the evolutionary process can proceed with punctuated equilibria under very general assumptions, without considering a necessarily biological substrate [6]. Their model considers a coevolutionary network of interactions, both positive and negative, between a number of species. The concept of punctuated equilibria has been also applied in the context of social theory and in the study of technology diffusion.

We pose the intriguing hypothesis here that the Internet architecture is also subject to punctuated equilibria. An empirical study of this hypothesis would require a measurement of the pace with which the Internet architecture has been adopting new protocols or significant modifications to its existing protocols - a difficult, but not impossible task.

What if the hypothesis of punctuated equilibria is true? First, we should not expect that the Internet architecture will be evolving at a roughly constant pace. It is "natural" for a technological artifact such as the Internet to go through long periods of stasis, during which its architecture appears to be ossified. Second, the Jain-Krishna model shows that even though most external stimuli do not initiate significant evolutionary change, there can be rare and minor external events that can trigger major changes in the internal organization of an ecosystem. In the Jain-Krishna model, these are not random events however; they affect the so-called dominant *Auto-Catalytic Set (ACS)* in the ecosystem, either damaging it or shifting it to another set of species. This may be an important consideration for researchers that work on future Internet designs. A new architecture would manage to replace or significantly affect the existing Internet architecture only if it offers a major advantage in those core functions that play the role of the dominant ACS today; otherwise it would not manage to have impact or, in the best case scenario, it would be subsumed by the existing Internet architecture.

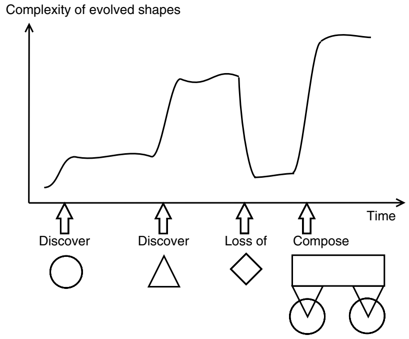

Figure 4: The evolution of a complex ecosystem consists of long periods of stasis separated by relatively short periods of major transformations. The latter can be caused by the creation or destruction of Auto-Catalytic Sets, or by a shift of the dominant ACS from one set of species to another. In this example, a rapid increase in the complexity of the evolved shapes can be due to the introduction of a new component or the formation of a new evolutionary kernel.

7. OPTIMALITY

In the engineering literature, including the study of computer networks, optimality in terms of performance or cost is usually the primary goal - often, the only goal. In only few cases (mostly in topology design), the optimization problem includes some reliability constraints to deal with limited link or node failure scenarios. The more general robustness problem, which aims to maintain the proper functioning of the system under internal or external variations is rarely considered. Other objectives, such as evolvability or modularity are even less frequently mentioned.

On the other hand, biological systems seem to be "designed" mostly for robustness, modularity, and evolvability. If we think of robustness and modularity as mechanisms that facilitate evolvability, it is fair to say that the primary objective of biological systems is the latter. On the contrary, biological systems do not appear to be globally optimal in terms of structure or function [10]; an engineer can usually design something to outperform an organism at a single task. Why is that?

Modular and robust systems are typically not optimal both topologically (i.e., in terms of the number of their internal components and connections), and dynamically (i.e., in terms of their response dynamics). First, the topology of modular systems includes, as previously discussed, redundant functional elements and interconnections. The reason is that each module has to independently perform its function, even if some elements of that function are also available in other modules. The topology of robust systems is also highly redundant, either because some functional elements are replicated or because there are multiple different elements that implement the same function (degeneracy). This topological redundancy increases the structural and operational cost of the resulting system, making it suboptimal. Second, in terms of response dynamics, modular systems are typically slower and they often need to go through more intermediate transient states before reaching an attractor (a point attractor would represent a steady-state, while a limit cycle would represent an oscillatory behavior). Robust systems would also be slower because of all the additional processing steps, checks and feedback loops that robustness requires.

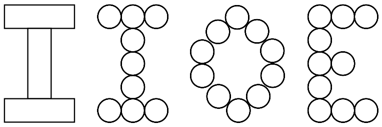

Figure 5: Suppose that the cost of a design is related to the number of components it is built from. The design of the letter I with rectangles is then of lower cost than the design with circles. The latter however is more evolvable, as it can be easily "rewired" to produce different phenotypes or functions.

In few cases, biological systems appear to be optimizing a certain cost-benefit relation, but in a heuristic manner that resembles stochastic search techniques. A good example is the genetic regulatory circuit that controls the production of lactose in E-coli. In that bacterium, there is a clear cost-benefit tradeoff: the cell grows slower the more lactose it produces, while on the other hand, breaking down lactose to use as an energy source helps the cell to grow faster. E-coli has a regulatory mechanism that allows a population of cells to move towards the optimal production level of lactose, i.e., the level that maximizes the cell's growth rate. In other words, a population of E-coli cells is quickly taken over by cells that optimize their lactose production level.

What does the previous discussion imply for network architectures and for the Internet in particular? First, if we accept that a network architecture is not designed only once, but it constantly evolves under the pressure of new application requirements, economic constraints and available technologies, we need to accept that the architecture should be primarily designed for evolvability, not for optimality. Tools that promote evolvability, such as modularity, hierarchy, robustness and degeneracy can be more important than performance or cost. Second, the architectural components that should be optimized, if any, are those that provide an evolutionary advantage to the architecture. Optimizations that jeopardize the evolvability of the architecture should be avoided, even if they result in short-term gains. Third, any optimizations should be adaptive in nature, attempting to heuristically optimize important trade-offs even under unknown and time-varying environments. Optimizations that assume full knowledge of the environment in which the architecture operates, or that require that environment to be constant, may be steps in the wrong direction.

8. CONCLUSIONS

We reviewed some important concepts in biological evolution, especially related to evolvability, and discussed how these concepts may be relevant to the evolution of Internet architectures. An important next step, currently under study, is to apply these concepts in a quantitative framework, through experimental or theoretical approaches, in specific network design problems. For instance, the use of degeneracy, instead of redundancy, the impact of modularly varying goals in the evolvability of a network architecture, or the trade-off between optimality and evolvability are research directions that seem fertile and should be further pursued. We hope that these research directions will lead to more cross-disciplinary work between network scientists and biologists in the future.

9. REFERENCES

[1] U. Alon, M. G. Surette, N. Barkai, and S. Leibler. Robustness in Bacterial Chemotaxis. *Nature*, 397:168–171, January 1999.

[2] E. H. Davidson and D. H. Erwin. Gene Regulatory Networks and the Evolution of Animal Body Plans. *Science*, 311:796–800, 2006.

[3] G. M. Edelman and J. A. Gally. Degeneracy and Complexity in Biological Systems. *PNAS*, 98:13763–13768, 2001.

[4] S. J. Gould. *The Structure of Evolutionary Theory.* Harvard University Press, 2002.

[5] L. Hartwell, J. Hopfield, S. Leibler, and A. Murray. From Molecular to Modular Cell Biology. *Nature*, 402:C47–C52, December 1999.

[6] S. Jain and S. Krishna. Large Extinctions in an Evolutionary Model: The Role of Innovation and Keystone Species. *PNAS*, 99:2055–2060, February 2002.

[7] N. Kashtan, E. Noor, and U. Alon. Varying Environments can Speed Up Evolution. *PNAS*, 104:13711–13716, August 2007.

[8] M. Kirschner and J. Gerhart. Evolvability. *PNAS*, 95:8420–8427, July 1998.

[9] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of Evolved and Designed Networks. *Science*, 303:1538–1542, 2004.

[10] G. A. Parker and J. Maynard Smith. Optimality Theory in Evolutionary Biology. *Nature*, 348:27–33, November 1990.

[11] H. A. Simon. *The Sciences of the Artificial.* The MIT Press, 1996.

[12] G. von Dassow, E. Meir, E. M. Munro, and G. M. Odell. The Segment Polarity Network is a Robust Developmental Module. *Nature*, 406:188–192, 2000.