What would Darwin Think about Clean-Slate Architectures?

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

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 It has NOT been peer-reviewed.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

ABSTRACT

As significant resources are directed towards clean-slate networking research, it is imperative to understand how clean-slate architectural research compares to the diametrically opposite paradigm of evolutionary research. This paper approaches the "evolution versus clean-slate" debate through a biological metaphor. We argue that evolutionary research can lead to less costly (more competitive) and more robust designs than clean-slate architectural research. We also argue that the Internet architecture is not ossified, as recently claimed, but that its core protocols play the role of "evolutionary kernels", meaning that they are conserved so that complexity and diversity can emerge at the lower and higher layers. We then discuss the factors that determine the deployment of new architectures or protocols, and argue, based on the notion of "auto-catalytic sets", that successful innovations are those that become synergistic components in closed loops of existing modules. The paper closes emphasizing the role of evolutionary Internet research.

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

General Terms: Theory

Keywords: NSF FIND, GENI, FP7-ICT FIRE, Internet.

1. INTRODUCTION

In the last few years, a major new trend in networking research is to focus on "clean-slate" architectures. The clean-slate paradigm aims to address fundamental problems and limitations of the Internet, without being constrained by the architecture or protocols currently used. As significant resources are directed towards clean-slate research, through NSF's FIND [11] and ICT's FIRE programs [6], it is imperative (or at least healthy) to understand how *clean-slate architectural research* compares to the diametrically opposite paradigm of *evolutionary research*. The latter also aims to address fundamental Internet problems, but without breaking the architecture that is in place today, and respecting constraints that relate to partial deployment, backwards compatibility, and implementation feasibility.

To further understand the key difference between the two paradigms, here is how the clean-slate approach is described in NSF's FIND initiative [11]: "FIND invites the research community to consider what the requirements should be for a global network of 15 years from now, and how we could build such a network if we are not constrained by the current Internet - *if we could design it from scratch.*" Two examples of clean-slate architectural research are XCP [9] and NIRA [14]. IPv6 is an earlier example of clean-slate design, even though significant efforts have focused on making it capable to co-exist with IPv4.

Instances of previous evolutionary developments that have had a major impact on the Internet include TCP congestion control, CIDR, MPLS network provisioning, BGP extensions (such as route reflectors), network intrusion detection systems, as well as several application-layer protocols that provide enhanced capabilities (say VoIP or video streaming) despite the limitations of underlying protocols.[1]

In this position paper, we approach the "evolution versus clean-slate" question through a biological metaphor. Nature has proved in practice that the evolutionary process is able to adapt and survive in radically changing environments and to produce surprising innovation, without ever needing a clean-slate restart.[2] As the evolution of living species is determined by few basic mechanisms, such as genetic heritage, mutations and natural selection, it may be that the evolution of large-scale internetworks is similarly determined by some simple principles. For a discussion of the similarities between the evolution of biological and complex technological systems we refer the reader to [10]. So far, the term "Internet evolution" has been used rather casually in the literature, and practices such as partial deployment or backwards compatibility have been viewed as "necessary evils" for non-disruptive, evolutionary growth. This paper argues that there may be a much deeper correspondence between biological evolution and the evolution of a complex, large-scale and distributed technological artifact such as the Internet. If this hypothesis is true, then we may be able to learn something about the evolution of the Internet from the discipline of biological evolution.

The main arguments in this paper are summarized as follows. First, even though a clean-slate design can always

---

[1] An interesting historical question is whether the invention of packet switching represents an example of clean-slate or evolutionary thinking. The author believes the latter, which becomes more clear when we consider the evolutionary path from circuit switching to synchronous multiplexing and then to asynchronous multiplexing. However, this is certainly a good subject for further debate and historical perspectives.

[2] One may argue that *extinction events* represent examples of clean-slate restarts. On the other hand, it is safe to say that there are no signs of an upcoming Internet extinction, and so clean-slate research may need a different justification.

provide a more optimized solution to a given set of objectives than an evolutionary design, the latter is typically less costly and thus, it has *a higher chance for selection in a competitive environment*. Second, we argue that *evolutionary designs are often more robust*, as they have evolved to survive a wider range of environments and objectives, instead of being optimized for a single (typically assumed or predicted) environment. Third, the widely quoted observation that the Internet architecture is ossified may be misleading. Biology teaches us that some core mechanisms (think of photosynthesis) or elements (such as the ATP molecule) need to be "super-conserved" in order for complexity and diversity to emerge around them. In the networking context, it may be that the IP protocol is an *evolutionary kernel*, meaning that it should change very slowly, if any, so that the protocols above and below it can keep evolving on a stable common framework. Fourth, we discuss issues that relate to the deployment of new architectures and protocols. The proponents of testbeds, such as GENI, argue that some excellent architectural ideas do not get adopted by the "real-world" because they lack experimental validation and early deployment at a testbed. We argue that the adoption of architectural innovations may have little to do with whether there is a working implementation or not. In fact, several past architectural "failures" were implemented and deployed in testbeds (think of RSVP or the MBONE).[3] Instead, we argue that the key issue is whether an innovation has the appropriate links in the network of synergistic and antagonistic interactions with other proposed innovations and legacy infrastructures. Relying on the notion of *Auto-Catalytic Sets* (ACS), inspired from theoretical studies of biological evolution, we argue that the innovations that do get deployed and succeed are those that become a component in closed loops of synergistic interactions with existing modules. Finally, we emphasize the significance of evolutionary research as an essential mechanism for the creation of "survivable mutations" in the evolution of the Internet.

The author has no doubt that many readers will disagree with the positions expressed here. His hope is that this article will initiate a healthy debate on the pros and cons of evolutionary versus clean-slate Internet research.

2. THE BIOLOGICAL METAPHOR

In this section, we describe an analogy between an evolving computer network and an evolving biological species. The discussion in the following sections is largely based on this analogy. We first need to agree on what constitutes the "species" and what forms the "environment" in which the species lives in, in the context of an evolving computer network. In our view, the species is the system that emerges from all software and hardware artifacts that form the network under study. This system is based on a conceptual organization or architecture (e.g., datagram internetworking), uses certain protocols (e.g., IP, TCP, 802.11), is implemented on various technologies (e.g., electronic routers, optical transmission), and it supports several applications (e.g., Web, P2P). Note that this view of the network-species goes beyond the architectural or protocol layers; we include implementation technologies and applications, and so we consider purely technological or application-layer changes as valid evolutionary steps.

The network-species lives in an "environment", which is everything outside that species that somehow interacts with the latter. To name the most important environmental components, there is a large population of heterogeneous users, economic structures and conditions, service needs, and potential threats. A key point is that the *environment is highly dynamic and uncertain.* Each of the previous components changes with time in a way that is largely unpredictable. For instance, network security was not a big environmental factor when the original Internet was designed, but it now represents one of the key motivations for change. It is this dynamic nature of the environment that pushes the network species to evolve. If the environment was somehow static, then it is likely that we could design the optimal network for that environment and stay with that artifact for a long time. This point argues that a clean-slate network architecture designed today would also have to survive in a dynamic environment, and so it would also be subject to evolution soon after its original design.

To go a step further, we note that *there is often a cycle between network capabilities and the demands that the environment presents.* Specifically, some of the environmental changes (mostly new service needs and application requirements) at a time period T are not random or externally imposed, but they depend on what the network was able to do just prior to T. To illustrate, until a few years ago the Internet was mostly used for file transfers, while VoIP was considered an ill-fated application. As many networks moved towards overprovisioning and the residential access capacities increased (for reasons that were not related to VoIP), some networks became capable to provide low delay and loss rate, opening the road for the first successful VoIP applications. At that point the need to support VoIP became stronger, and the environment changed in a way that it now requires most networks to be provisioned for VoIP traffic. In other words, a new environmental/application requirement emerged as a result of a capability that the network had developed for unrelated reasons. *This cycle between network capabilities and environmental demands is a crucial element of network evolution.* If there was no such cycle, and the environment was posing requirements that are completely foreign to the existing network capabilities, the evolutionary process would be much more challenging and it is possible that we would need a clean-slate redesign of the Internet every few years.

The biological analogy with a species that lives in a dynamic environment calls for *evolution* as the only way for that species to survive. Biological evolution is based on three key facts of life: genetic heritage, variation (through mutations, gene duplication, or other genetic mechanisms), and natural selection. We next argue that these three facts have close counterparts in the evolution of computer networks. First, *genetic heritage* refers to the well-known property of *backwards compatibility.* Network offspring typically inherit most (but not all) of the architecture, protocols, underlying technologies and supported applications of their predecessors.

Second, *variation* refers to specific changes in the existing network-species at the architectural, protocol, technological or application levels. *A mutation is very different from a clean-slate design. The former causes a variation in an existing species, while the latter aims to create a new species.* A difference with biology, of course, is that network

---

[3] This point was further discussed in [5].

mutations do not have to be random. Instead, network researchers, entrepreneurs, creative programmers, can create mutations that have a high probability of survival. It should be emphasized, however, that because of the inherent unpredictability in the network environment, we cannot recognize a priori the best possible mutation. Consequently, it is very important, just as in biology, to *allow for genetic diversity through a plethora of mutations, and then let the environment select the most competitive mutation*, as described in the next paragraph.

Third, *natural selection* is the process through which the environment chooses among mutations. Each mutation is associated with a certain fitness, i.e., the ability of the resulting network to meet the needs of the environment at that time period. In the networking context, we should consider that each mutation is also associated with a cost to deploy or propagate that mutation in the entire species. This cost may not be strictly monetary, and it can be part of a *fitness function* (the higher the cost of the mutation, the lower its fitness). As in the biological context, we argue that network evolution proceeds by selecting the mutation that is associated with the highest fitness. This perspective allows us to understand why the Internet species has not always selected protocols or architectures that were certainly well designed (think of IntServ or IPv6): their fitness function was probably much lower compared to that of competing mutations (overprovisioning or NATs, respectively).

This paper is certainly not the first to note the analogy between the evolution of biological systems and the evolution of large-scale networks. John Doyle and his collaborators, in particular, have contributed extensively in this area (for instance, see [2]). One of their key contributions is the HOT (Highly Optimized Tolerance) framework as a common explanation for the highly-structured and robust (yet fragile in certain aspects) configuration of both biological and technological systems.

3. EVOLUTION VS. CLEAN-SLATE DESIGN

In this section, we first describe a quantitative framework for evolutionary and clean-slate network design. This framework is abstract, and it cannot be used in practice without first knowing many cost parameters. Nevertheless, it is attractive because it leads naturally to two ratios, the "price of evolution" and the "price of clean-slate design", that can be used to compare the two architectural approaches.

Suppose that the network, in its most general representation, can be any element of the set $\mathcal{N}$. During a particular generation k of the evolutionary process, the network at hand is $N_k \in \mathcal{N}$. The network N_k meets certain requirements R_k that are presented by the environment during that generation. For instance, R_k may involve specific security, QoS, or reliability constraints. The cost of network N_k, in some generalized cost units, is $C(N_k)$. This cost includes the design and implementation of network N_k as well as all costs associated with its operation during generation k.

Now, suppose that at some point in time we know the environmental requirements R_{k+1} for the next generation $k+1$ and we need to design a network that meets those requirements. We consider two options: a clean-slate design that will lead to an optimal network $\hat{N}_{k+1} \in \mathcal{N}$ for R_{k+1}, and an evolution-based design $N_{k+1} \in \mathcal{N}$ that results from the mutation of N_k that has the highest fitness (or lowest cost).

The *Clean-Slate (CLS)* design is the minimum-cost network that meets the new requirements:

$$\hat{N}_{k+1} = \arg\min_{N \in \mathcal{N}} \{C(N) \;\text{ such that } R_{k+1}(N) = 1\} \qquad (1)$$

where $R_{k+1}(N) = 1$ means that network N meets the requirements R_{k+1} (and it is zero otherwise).

To define the evolution-based network, we need to consider network mutations and their cost. Let $C_\delta(N, N')$ be the cost of a mutation that transforms a network N to a network N', including all operational costs for N' during generation k. The *Evolution-based (EVO)* design is the network which results from the minimum-cost mutation of network N_k that meets the requirements R_{k+1}:

$$N_{k+1} = \arg\min_{N \in \mathcal{N}} \{C_\delta(N_k, N) \;\text{ such that } R_{k+1}(N) = 1\} \quad (2)$$

Note that the previous cost factors do not need to be constant. As the environmental requirements change with time, the cost factors can also change across generations. For instance, the cost of introducing NAT in a network has decreased over time, both because NAT equipment have become a commodity and because more and more applications become NAT-aware (e.g., Skype).

The previous equations for CLS and EVO designs show a key difference between the two: *CLS produces the optimal (i.e., minimum cost) design for a specific environment.*[4] EVO, on the other hand, selects the minimum cost "mutation" (e.g., an architectural change, a new technology or even a new application capability) that transforms the existing network in order to satisfy a given requirement.

3.1 The price of evolution

A first question based on the previous framework is: *what is the price of evolution?* Or, what is the cost of the EVO network at a given generation compared to the CLS network that can meet the same requirements? By definition, the CLS network has lower cost than the EVO network, i.e., $C(\hat{N}_{k+1}) \leq C(N_{k+1})$, because CLS produces the minimum cost design that satisfies the given requirements. So, the price of evolution at generation $k+1$ can be defined by the ratio:

$$\phi_{k+1} = \frac{C(N_{k+1})}{C(\hat{N}_{k+1})} \geq 1 \qquad (3)$$

The crucial question here is *whether this ratio increases with time.* If the sequence $\{\phi_i, i = 1, 2, \dots\}$ diverges away from one, then the EVO network gradually becomes more and more costly than the CLS network. In that case, we can expect that at some point during the evolutionary process a disruption will occur because the EVO network has become too costly, compared to the cost of a new network that is optimized for the current environment. One may argue that such a disruption took place when the datagram-based Internet offered a mechanism for much cheaper data communication compared to the circuit-based telephone network of the seventies, allowing several new applications to emerge.

---

[4] Proponents of clean-slate architectures may argue at this point that their goal is not to design an optimal architecture for a specific environment, but an architecture that targets adaptability and evolvability. Unfortunately, even though we all agree that these are important objectives, it remains unclear how to achieve them "by design", or even how to examine in a quantitative manner whether a given architecture has these properties.

On the other hand, if the ratio ϕ_i does not increase with time, then the EVO network, even though not optimally designed for the given requirements, remains close (in terms of cost, not architecture) to what a clean-slate design would produce at each generation. In that case, it is much harder to justify a clean-slate redesign of the network, given the high cost of the latter.

3.2 The price of clean-slate design

A second interesting question based on the previous framework is: *when is it better to rely on evolution instead of a clean-slate design?* Based on the previous cost terms, the cost of the CLS design is $C(\hat{N}_{k+1})$ while the cost of performing the mutation required by the EVO design is $C_\delta(N_k, N_{k+1})$. So, this last question can be rephrased: under which conditions (i.e., cost structure and sequence of requirements R_i) is it true that the following ratio

$$\chi_{k+1} = \frac{C_\delta(N_k, N_{k+1})}{C(\hat{N}_{k+1})} \qquad (4)$$

is less than one? Intuitively, we expect that an EVO mutation will cost much less than a completely new network (CLS). The reason is that the cost of the former is only the cost of the corresponding mutation. For instance, the cost of installing spam filters at mail servers is arguably much less than moving to a completely new email architecture that can avoid spam altogether, given the wide deployment and investment in the current SMTP-based architecture.

On the other hand, the ratio χ_{k+1} may be larger than one under certain conditions, meaning that the EVO mutation may cost more than the CLS network. That could happen if, in a particular generation x, the requirements R_x become completely unrelated, or much more stringent, than the requirements imposed on the previous generations. Such a sudden shift in the environment may, in theory, force the evolutionary process to select a mutation that is of comparable or even higher cost than a clean-slate design. For instance, let us recall the debate between ATM networks and IntServ-capable networks in the mid-nineties. End-to-end ATM can be now viewed as a clean-slate design that could offer per-flow QoS. RSVP and the IntServ architecture represented an evolutionary change in the Internet protocols that could also offer per-flow QoS. The requirement of per-flow QoS, however, was significantly different than any previous Internet capability. Consequently, the cost of the two approaches was not very different at the end, as they would both require major changes in applications, end-hosts and throughout the network infrastructure. Eventually, the real-world avoided both approaches, and it selected the lower-cost mutation of overprovisioning.

If the price of clean-slate design depends on the "dynamics" of network requirements, *what is the process that determines the sequence of these requirements?* Earlier, we discussed the cycle between the capabilities of a network and the requirements that the environment imposes on the network. If this feedback is strong, meaning that the requirements R_{k+1} are often determined by what the network N_k can do, then it may be that the evolutionary process never breaks. Otherwise, if the requirements change rapidly and significantly at the same time, we can expect that a new network will have to be redesigned from scratch. Looking at the last 20-30 years of Internet history, starting from the development of TCP/IP or even earlier, we cannot identify any rapid change in the series of network requirements that was not quickly addressed through an evolutionary development. The bottomline is that *a clean-slate restart has not been required during the history of the Internet so far.*

4. OPTIMALITY VERSUS ROBUSTNESS

The CLS design paradigm focuses on optimality, given certain requirements. What happens however when the environment changes with time, or when the requirements are not precisely known? We should be also interested in the robustness of a network architecture, especially in the presence of uncertainty and fluctuations in the requirements that that architecture should address.

Given that the environment in which a network evolves is uncertain and to a large degree unpredictable, *we expect that the clean-slate design paradigm cannot be as robust as the evolutionary paradigm.* First, an evolutionary architecture has been tested in practice for a longer time period and in the presence of wide variations in the environment of available technologies and user requirements. Second, any evolutionary process tends to accumulate unnecessary or underutilized "baggage", i.e., various components or properties that were perhaps important in the past, but not now. For instance, we all carry the vermiform appendix in our bodies as an evolutionary remainder of a remote ancestor. Similarly, the TCP protocol has 16 bits in its header for the Urgent Pointer field, even though that field is rarely used. Such components however, can also facilitate and speed up the evolutionary process, because they provide place-holders for future capabilities and extensions. For instance, even though the TOS IPv4 header field was rarely used in the past twenty years, it has recently been redefined as the DSCP header field and is now used in providing differentiated services within some networks.

An interesting related research question is related to the price of evolution (ratio ϕ) and the robustness of the EVO network relative to the CLS network. Is it true that the robustness of EVO increases as the ratio ϕ increases, or can we achieve somehow a "sweet spot" of evolution that is both efficient (i.e., ϕ close to one) and significantly more robust than CLS? Even though it is hard to answer such questions in the abstract framework of this paper, we expect that quantitative modeling should be possible in the case of specific design problems in which the CLS and EVO processes can be accurately specified.

5. OSSIFICATION OR CONSERVATION?

The widely cited paper by Peterson, Shenker, and Turner [12] has become the manifesto for clean-slate architectural research. That paper argues that *the mid-layer protocols of the Internet stack are ossified and that the Internet is at an impasse.* In this section, we argue based on the biological metaphor that the mid-layer protocols of the Internet architecture correspond to "evolutionary kernels" and as such, they need to be conserved with very few changes. Their conservation is necessary so that complexity and diversity can result at lower and higher architectural layers and technologies. The fact that several architectural changes at the network or transport layers have "failed" (i.e., they did not get selected by the real-world) during the last two decades is not a result of an impasse, but is one more evidence that the evolution of the Internet follows similar laws with biological evolution.

In genetics, there are certain "ultraconserved elements" in the human genome that are almost identical to corresponding genes in very remote species [1]. Specifically, it appears that 481 segments of the human genome, each of them longer than 200 base pairs, have been absolutely conserved during the last 300-400 million years! In a related discovery, Davidson and Erwin show evidence, based on the evolution of body plans, that certain small gene regulatory networks referred to as *kernels* evolved about 500 million years ago, and after their initial formation they quickly "locked" development onto a certain path [4]. The role of such evolutionary kernels can be understood using the *bow tie architecture* of Csete and Doyle [3]. In the context of metabolism, the knot corresponds to just a dozen of key molecules, such as the energy-carrier ATP and the glucose 6-phosphate metabolite, and it appears to be conserved with only minor variations throughout all species. The bow tie architecture emphasizes that the conservation of the core components (the knot) may be a necessary condition for the development of complexity and for the continuous evolution at the edges of the bow tie.

Returning to networking and to the Internet architecture, we cannot ignore that networks are designed (and often implemented) in a layered manner. In the case of the Internet, it is often said that the mid-layer protocols, namely IP, TCP/UDP, and the routing protocols (the "waist of the protocol hourglass"), have become *ossified*, meaning that they have practically stopped evolving during the last decade or so. New architectures and protocols for the network/transport layers, such as IP-multicast, IntServ, or XCP, have not been selected despite the fact that they were well studied, implemented, and deployed in testbeds. We argue that the TCP/IP protocol waist corresponds to an evolutionary kernel in the Internet architecture. Their conservation may be a necessary condition for the continuous evolution of the protocols that live at either higher than the waist (mostly application-layer protocols) or lower than the waist (mostly link and physical layers). Without the ultraconserved common base that the protocol waist offers, the evolution of ever-increasing diversity and complexity at the higher and lower layers protocols would not be possible.

The reader may wonder at this point: *if the core protocols need to be conserved and there is no need for major innovation at the waist of the protocol stack, then what is the role of networking research?* The author believes that networking research should have a much broader scope than focusing on the core protocols, such as routing, transport, or naming/addressing. Innovation today mostly takes place at the lower and higher layers of the protocol stack, where new technologies and diverse application requirements interact with legacy protocols and architectures. *By narrowing our focus on the core protocols, we only narrow the significance and potential impact of our research.*

6. WHAT GETS DEPLOYED AT THE END?

Suppose that an ISP deploys a new protocol or service. What determines whether this innovation will be globally adopted or not? It has been claimed recently that the "ecosystem" of ISPs has reached an impasse, partially because ISPs have no incentive to offer a new service if the latter cannot be deployed in an end-to-end basis. It is also claimed that new networking architectures and protocols are not deployed because they lack experimental validation at a wide-area testbed. This last argument is the main motivation for building expensive infrastructures such as GENI. In this section we focus on these issues, and ask a naive but significant question: *among all competing research proposals and proposed innovations, what gets deployed at the end?*

One way to approach this question is to focus on specific protocols and examine the difficulties they encountered as they moved from initial design to deployment. This approach was taken in [13], a very interesting recent Internet Draft that discusses several case studies. Here, we prefer to approach the previous question in a more abstract and general framework. Specifically, we use a model proposed by Jain and Krishna that was originally developed to study the evolution of ecosystems and the adoption of evolutionary innovations [7]. A summary of that model follows. Consider S species (or "agents"). The interactions between agents are captured by a directed weighted graph, where each node represents an agent. The weight $w_{i,j}$ of the link from node j to i is a real number that quantifies the type and strength of the effect that agent j has on agent i. When $w_{i,j} > 0$, the effect is cooperative, and so, at least in terms of population, the more we have of agent j the more will be produced of agent i. If $w_{i,j} < 0$, the effect is antagonistic, and the presence of agent j tends to reduce the population of agent i. Note that the interpretation of the weights does not need to be in terms of population; instead, they can refer to abstract utility functions, monetary power, market shares, or even an abstract deployability metric for a protocol. The Jain-Krishna model also relies on a relation that expresses the population growth rate of agent i as a function of the population of other agents and of the matrix of weights $W = \{w_{i,j}\}$. That relation is domain-specific.

The Jain-Krishna model leads to several interesting results (see also [8]). For instance, the ecosystem can go through periods of "randomness", without any particular structure, followed by long periods of highly organized self-emerging structure, followed by sudden extinction and/or transformation events. These evolutionary dynamics are referred to as *punctuated equilibria.* A key concept behind these effects is the *Auto-Catalytic Set (ACS). An ACS is a set of nodes such that each node has at least one incoming positive link from another node in the set.* Note that a cycle of positive links in the interaction network is an ACS, but in general, an ACS does not need to consist of only a cycle of positive links. An ACS represents a group of agents that can sustain themselves, accumulating population (or utility, power, etc) through their symbiotic relations. It is also interesting that the entire ecosystem can go through an extinction or major transformation event when a "keystone" agent in the dominant ACS loses its positive incoming links for some reason.

The Jain-Krishna model gives us a new perspective about the competition and fate of proposed innovations (new agents in the previous network of agent interactions). *The innovation that eventually succeeds corresponds to the agent that is not only technically solid, but also it forms the stronger ACS in the network of interactions with other innovations and existing infrastructures.* In the context of networking protocols, we should not expect that a new protocol will get deployed simply because it offers a new service. The key question is whether it forms the appropriate synergistic connections with either existing protocols, or with other new protocols that are available at the same time. In the net-

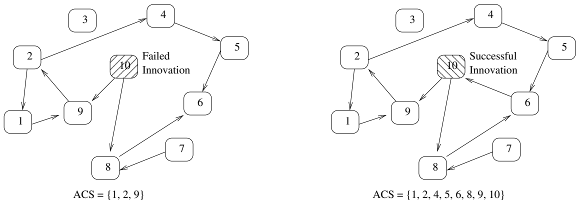

Figure 1: Two different versions of an interaction network after an "innovation" (node 10) has been introduced in the ecosystem. The ACS at the left does not include node 10 and the innovation fails. The network at the right includes only one more link, but that link creates a large ACS that includes the new node and the innovation succeeds.

working research community, we often compare the merit of different innovations strictly in terms of performance. Unfortunately, other factors, such as the nature and strength of the interactions between an innovation and the context in which it will be deployed, are rarely understood or considered.

7. EVOLUTIONARY RESEARCH REVISITED

It is unfortunate that evolutionary research is often viewed as incremental, short-sighted, or simplistic. The major difference between evolutionary and clean-slate architectural research is that the former aims to solve problems and enable new services without breaking the existing Internet. In other words, *evolutionary research emphasizes the role of the context or environment in which a proposed solution will be deployed, as opposed to designing protocols in a vacuum.*

Even if we had the chance to redesign the Internet from scratch today, we should recognize that, because of the delays involved in the design, implementation and testing phases, the new protocols will probably not be deployed in less than five-ten years from now. In the meanwhile, and given the fast-paced evolution of the surrounding ecosystem, it is likely that the underlying technologies, constraints and requirements will have significantly changed, outdating any clean-slate architecture that we design today.

In terms of GENI, we should note that deployment or experimentation at a research testbed does not necessarily improve our understanding of a new protocol or service. A testbed, no matter how expensive or fast it is, cannot capture the complex interactions of the Internet ecosystem in terms of competition and synergy between the involved players (users, ISPs, router vendors, etc), it cannot exhibit the diversity and heterogeneity of the technologies and applications used at the real Internet, and it cannot reproduce the closed-loop multiscale characteristics of Internet traffic, among other issues.

Finally, instead of attempting to concentrate the energy and creativity of the networking research community in a focused effort to create the architecture of the Future Internet, *we should encourage and support the diversity of ideas and innovations.* Evolutionary research can be viewed as *the creation of survivable mutations.* We cannot predict the future, and sometimes we do not even accurately know the present. Consequently, instead of attempting to design the best solution to a given problem, *we should invest in the creation of multiple competing solutions, relying on the real-world to select the fittest innovation given the constraints and requirements at that time.*

Acknowledgments

Thanks to Mostafa Ammar and Mukarram Bin Tariq for many long discussions. Sally Floyd and Greg Minshall provided thorough and constructive comments, as well as the HotNets-VI reviewers.

8. REFERENCES

[1] G. Bejerano, M.Pheasant, I.Makunin, S.Stephen, W. James Kent, J.Mattick, and D.Haussler. Ultraconserved Elements in the Human Genome. *Science*, 304:1321–1325, May 2004.

[2] M. Csete and J. Doyle. Reverse Engineering of Biological Complexity. *Science*, 295:1664–1669, March 2002.

[3] M. Csete and J. Doyle. Bow Ties, Metabolism and Disease. *TRENDS in Biotechnology*, 22(9):446–450, September 2004.

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

[5] C. Dovrolis. Four Myths about GENI (and one recommendation). Presented at the DIMACS Tutorial on Algorithms for Next Generation Networks, August 2007.

[6] A. Gavras, A.Karila, S.Fdida, M.May, and M.Potts. Future Internet Research and Experimentation: The FIRE Initiative. *ACM Computer Communication Review (CCR)*, July 2007.

[7] S. Jain and S. Krishna. A Model for the Emergence of Cooperation, Interdependence, and Structure in Evolving Networks. *Proceedings of the National Academy of Science (PNAS)*, 98(2):543–547, January 2001.

[8] S. Jain and S. Krishna. Large Extinctions in an Evolutionary Model: The Role of Innovation and Keystone Species. *Proceedings of the National Academy of Science (PNAS)*, 99(4):2055–2060, February 2002.

[9] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay Product Networks. In *Proceedings of ACM SIGCOMM*, August 2002.

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

[11] National Science Foundation. NSF NeTS FIND Initiative. http://www.nets-find.net.

[12] L. Peterson, S. Shenker, and J. Turner. Overcoming the Internet Impasse through Virtualization. In *Proceedings of ACM Workshop on Hot Topics in Networks (HotNets-III)*, November 2004.

[13] D. Thaler and B.Aboba. What Makes For a Successful Protocol? http://tools.ietf.org/html/draft-iab-protocol-success-01, 2007.

[14] X. Yang. NIRA: A New Internet Routing Architecture. In *Proceedings of ACM SIGCOMM Workshop on Future Directions in Networking Architecture*, August 2003.