The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture (extended version)[*]

Saamer Akhshabi
College of Computing
Georgia Institute of Technology
sakhshab@cc.gatech.edu

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

ABSTRACT

The Internet protocol stack has a layered architecture that resembles an hourglass. The lower and higher layers tend to see frequent innovations, while the protocols at the waist of the hourglass appear to be "ossified". We propose *EvoArch*, an abstract model for studying protocol stacks and their evolution. *EvoArch* is based on a few principles about layered network architectures and their evolution in a competitive environment where protocols acquire value based on their higher layer applications and compete with other protocols at the same layer. *EvoArch* produces an hourglass structure that is similar to the Internet architecture from general initial conditions and in a robust manner. It also suggests a plausible explanation why some protocols, such as TCP or IP, managed to survive much longer than most other protocols at the same layers. Furthermore, it suggests ways to design more competitive new protocols and more evolvable future Internet architectures.

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

General Terms: Theory

Keywords: Internet Architecture, Future Internet, Layering, Network Science, Evolutionary Kernels, Evolution.

1. INTRODUCTION

Why does the Internet protocol stack resemble an hourglass? Is it a coincidence, intentional design, or the result of an evolutionary process in which new protocols compete with existing protocols that offer similar functionality and services? The protocol stack was not always shaped in this way. For instance, until the early nineties there were several other network-layer protocols competing with IPv4, including Novell's IPX, the X.25 network protocol used in Frame Relay, the ATM network layer signaling protocol, and several others. It was through a long process that IPv4 eventually prevailed as practically the only surviving protocol at layer-3, creating a very narrow waist at the Internet architecture hourglass (see Figure 1).

Another important question is: *why do we tend to see more frequent innovations at the lower or higher layers of the protocol hourglass, while the protocols at the waist of the hourglass appear to be "ossified" and difficult to replace?* During the last 30–40 years we have seen many new physical and data link layer protocols created and surviving. And of course the same can be said about applications and application-layer protocols. On the other hand, the protocols at the waist of the hourglass (mostly IPv4, TCP and
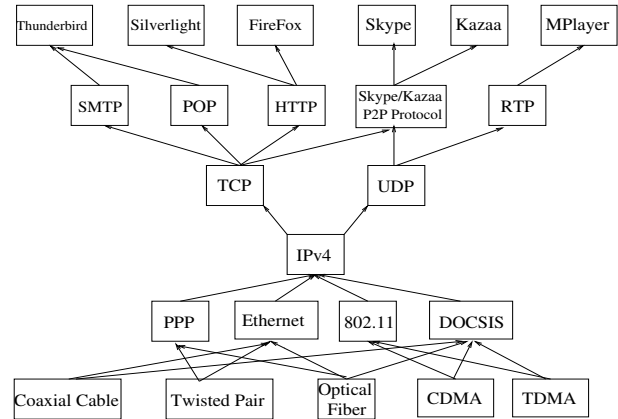

Figure 1: An (incomplete) illustration of the hourglass Internet architecture.

UDP) have been extremely stable and they have managed to outcompete any protocols that offer the same or similar functionality. *How can a new protocol manage to survive the intense competition with those core protocols at the waist of the Internet hourglass?* In fact, *the ossification of the hourglass waist* has been a major motivation for "clean-slate" efforts to design a novel future Internet architecture [16]. There are two important questions in that context. First, *how can we make it more likely that a new (and potentially better) protocol replaces an existing and widely used incumbent protocol?* And second, how can we make sure that a new architecture we design today will not be ossified 10–20 years later? In other words, *what makes a protocol stack or network architecture evolvable?* The previous questions have generated an interesting debate [9, 10, 19].

In this paper, we attempt a first effort to study protocol stacks (and layered architectures, more generally) as well as their evolution in a rigorous and quantitative manner. Instead of only considering a specific protocol stack, we propose an abstract model in which protocols are represented by nodes, services are represented by directed links, and so a protocol stack becomes a layered directed acyclic graph (or network). Further, the topology of this graph changes with time as new nodes are created at different layers, and existing nodes are removed as a result of competition with other nodes at the same layer.

The proposed evolutionary model, referred to as *EvoArch*, is based on few principles about layered network architectures in which an "item" (or service) at layer-X is constructed (or composed) using items at layer-(X-1). These principles capture the following: (a) the source of *evolutionary value* for an item,

(b) the *generality* of items as we move to higher layers,

(c) the condition under which two items compete,

(d) the condition under which one item causes the death or removal of a competing item.

Perhaps surprisingly, these few principles are sufficient to produce hourglass-shaped layered networks in relatively short evolutionary periods.

As with any other model, *EvoArch* is only an abstraction of reality focusing on specific observed phenomena, in this case the hourglass structure of the Internet protocol stack, and attempting to identify a parsimonious set of principles or mechanisms that are sufficient to reproduce the observed phenomena. As such, *EvoArch* is an *explanatory model* (as opposed to black-box models that aim to only describe statistically some observations). *EvoArch* deliberately ignores many aspects of protocol architectures, such as the functionality of each layer, technological constraints, debates in standardization committees, and others.[1] The fact that these practical aspects are not considered by *EvoArch* does *not* mean that they are insignificant; it means, however, that if the evolution of network architectures follows the principles that *EvoArch* is based on, then those aspects are neither necessary nor sufficient for the emergence of the hourglass structure.

EvoArch is certainly *not* going to be the only model, or "the correct model", for the emergence of hourglass-shaped network architectures. It is likely that there are other models that can produce the same hourglass structure, based on different principles and parameters. Additionally, *EvoArch* does not aim to capture every aspect of the Internet architecture; it only focuses on the emergence of the hourglass structure, and so it may be the wrong model to use for other purposes (e.g., to study the economics of new protocol deployment). G.Box wrote that *"all models are wrong but some models are useful"* [3]. We believe that *EvoArch* is a useful model for (at least) the following ten reasons:

1- It gives us a new way to think about protocol stacks and network architectures and to study their evolutionary properties based on few fundamental principles (§2).

2- *EvoArch* provides a plausible explanation (but certainly not the only explanation) for the emergence of hourglass-like architectures in a bottom-up manner (§3).

3- *EvoArch* shows how the location and width of the hourglass waist can follow from certain key parameters of the underlying evolutionary process (§5).

4- *EvoArch* can be parameterized to produce a structure that is similar to the TCP/IP protocol stack, and it suggests an intriguing explanation for the survival of these protocols in the early days of the Internet (§5.4).

5- *EvoArch* suggests how to make a new protocol more likely to survive in a competitive environment, when there is a strong incumbent (§5.5).

6- *EvoArch* provides recommendations to designers of future Internet architectures that aim to make the latter more evolvable (§5.5).

7- *EvoArch* predicts that few protocols at the waist (or close to it) become *ossified*, surviving much longer than most other protocols at the same layer, and it shows how such ossified protocols can be eventually replaced (§6).

8- When we extend *EvoArch* to capture the effect of different *protocol qualities*, we find that the lower part of the hourglass is significantly smaller than the upper part (§7.1).

9- The most stable protocols at the waist of the architecture are often *not* those with the highest quality (§7.2).

10- Finally, *EvoArch* offers a new way to think about the competition between IPv4 and IPv6 and to understand why the latter has not managed to replace the former (§7.3).

The rest of the paper is structured as follows. In Section 2, we describe *EvoArch* and explain how the model relates to protocol stacks and evolving network architectures. In Section 3, we present basic results to illustrate the behavior of the model and introduce some key metrics. Section 4 is a robustness study showing that the model produces hourglass structures for a wide range of parameter values. The effect of those parameters is studied in Section 5 focusing on the location and width of the waist. Section 6 examines the *evolutionary kernels* of the architecture, i.e., those few nodes at the waist that survive much longer than other nodes. Section 7 generalizes *EvoArch* in an important and realistic manner: what if different protocols at the same layer have different qualities (such as performance or extent of deployment)? We review related work in Section 8, present some criticism in Section 9, and conclude in Section 10.

2. MODEL DESCRIPTION

In *EvoArch*, a protocol stack is modeled as a directed and acyclic network with L layers (see Figure 2). Protocols are represented by nodes, and protocol dependencies are represented by directed edges. If a protocol u at layer l uses the service provided by a protocol w at layer $l-1$, the network includes an "upwards" edge from w to u.[2] The layer of a node u is denoted by $l(u)$. The incoming edges to a node u originate at the *substrates* of u, represented by the set of nodes $S(u)$. Every node has at least one substrate, except the nodes at the bottom layer. The outgoing edges of a node u terminate at the *products* of u, represented by the set of nodes $P(u)$. Every node has at least one product, except the nodes at the top layer.

The substrates of a node are determined probabilistically when that node is created.[3] Specifically, each layer l is associated with a probability $s(l)$: a node u at layer $l+1$ selects independently each node of layer l as substrate with probability $s(l)$. We refer to $s(l)$ as the *generality* of layer l. $s(l)$ *decreases* as we move to higher layers, i.e., $s(i) > s(j)$ for $i < j$. The decreasing generality probabilities capture that *protocols at lower layers are more general in terms of their function or provided service than protocols at higher layers*. For instance, in the case of the Internet protocol stack, a protocol at layer-1 offers a very general bit transfer service between two directly connected points; this is a service or function that almost any higher layer protocol would need. On the other extreme, an application-layer protocol, such as SMTP, offers a very specialized service and it is only used by applications that are related to email exchanges. Note that if node u does not select any substrate from layer l we connect it to one randomly chosen substrate from that layer.

Each node u has an *evolutionary value*, or simply *value* $v(u)$ that is computed recursively based on the products of u,

$$v(u) = \begin{cases} \sum_{p \in P(u)} v(p) & l(u) < L \\ 1 & l(u) = L \end{cases} \quad (1)$$

---

[1] The reader can see some of the criticism raised by anonymous reviewers in Section 9.

[2] In practice, the principle of strict layering is occasionally violated through tunnels or other forms of virtual networks. For the most part, however, layering is the norm in protocol architectures rather than the exception. Considering architectures without strict layering is outside the scope of this paper and an interesting subject for future research.

[3] Of course in practice substrates are never chosen randomly. The use of randomness in the model implies that a realistic mechanism of substrate selection is not necessary for the emergence of the hourglass structure.
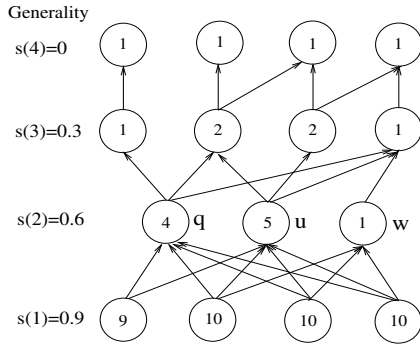
Figure 2: A toy network with four layers. The value of each node is shown inside the circle.

Figure 3: The death probability for three values of the mortality parameter z.

The value of the top-layer nodes is assumed to be fixed; in the simplest version of *EvoArch* it is equal to one. So, the model captures that the value of a protocol u is driven by the values of the protocols that depend on u. For instance, TCP has a high evolutionary value because it is used by many higher layer protocols and applications, some of them being highly valuable themselves. A brand new protocol on the other hand, may be great in terms of performance or new features, but its value will be low if it is not used by important or popular higher layer protocols.

The value of a node largely determines whether it will survive the competition with other nodes at the same layer that offer similar services. Consider a node u at layer l. Let $C(u)$ be the set of *competitors* of u: this is the set of nodes at layer l that share at least a fraction c of node u's products, i.e.,

$$w \in C(u) \text{ if } l(w) = l(u) \text{ and } \frac{|P(u) \cap P(w)|}{|P(u)|} \geq c \quad (2)$$

The fraction c is referred to as the *competition threshold*. In other words, a node w competes with a node u if w shares a significant fraction (at least c) of u's products, meaning that the former offers similar services or functions with the latter. Note that the competition relation is not symmetric: w may provide a generic service, having many products, and thus competing with several protocols at the same layer; the latter may not be competitors of w if they provide more specialized functions and have only few products.

Given the set of competitors of a node u, we can examine whether u would survive the competition or die. The basic idea is that u dies if its value is significantly less than the value of its *strongest* (i.e., maximum value) competitor. Specifically, let $v_c(u)$ be the maximum value among the competitors of u

$$v_c(u) = \max_{w \in C(u)} v(w) \quad (3)$$

If u does not have competitors, $v_c(u)$ and the death probability for u are set to zero. Otherwise, we introduce the *death probability ratio* $r = \frac{v(u)}{v_c(u)}$. The death probability $p_d(r)$ is then computed as follows:

$$p_d(r) = \begin{cases} e^{\frac{-z\,r}{1-r}} & 0 < r < 1 \\ 0 & r \geq 1 \end{cases} \quad (4)$$

The death probability function $p_d(r)$ is shown in Figure 3 for three different values of the *mortality parameter z*. This parameter captures the intensity of the competition among protocols. As z decreases, the competition becomes more intense and it is more likely that a protocol will die if at least one of its competitors has higher value than itself.
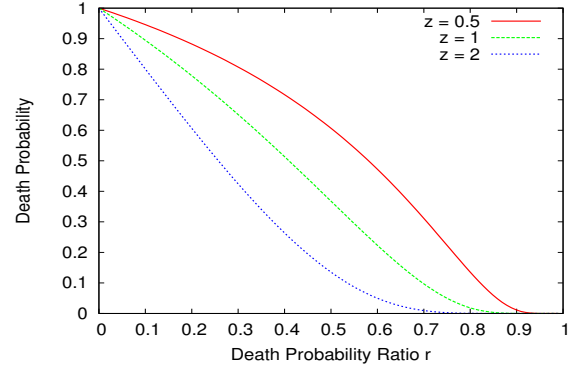
When a node u dies, its products also die *if their only substrate is u*. This can lead to a *cascade effect* where the death of a node leads to the death of several nodes in higher layers.

To illustrate the previous concepts, Figure 2 shows a toy network with $L = 4$ layers. The generality probability for each layer is shown at the left of the corresponding layer. Note that, on average, the number of products per node decreases as we move to higher layers because the generality probability decreases in that direction. Assuming that $c = 3/5$, nodes u and q are competitors of node w in layer-2. It is likely (depending on the parameter z) that w would soon die because its value is much less than that of its maximum-value competitor, u. u is also a competitor of q but this competition is much less likely to be lethal for the latter because its value is comparable to that of u.

EvoArch captures the inherent competition between nodes at the same layer, and specifically, between nodes that offer about the same service. For instance, FTP and HTTP are two application-layer protocols that can both be used for the transfer of files. The large overlap of the services provided by HTTP with the services provided by FTP (i.e., HTTP is a competitor of FTP) and the fact that HTTP acquired over the years a larger evolutionary value from its own higher layer products (applications such as web browsers) leads to the extinction of FTP. On the other hand, TCP and UDP are two transport layer protocols that offer largely different services. Their competition, in terms of products (i.e., application layer protocols), is minimal and the two protocols have coexisted for more than 30 years.

In the simplest version of *EvoArch*, the creation of new nodes follows *the basic birth process*. Specifically, the number of new nodes at a given time is set to a small fraction (say 1% to 10%) of the total number of nodes in the network at that time, implying that the larger a protocol stack is, the faster it grows. Each new node is assigned randomly to a layer. In Section 6, we also examine a *death-regulated birth process*, in which the frequency of births at a layer depends on the death rate at that layer.

EvoArch is a discrete-time model. By t_k, we denote the *k'th round*. In each round, the model execution includes the following steps in the given order:
a) birth of new nodes and random assignment to layers,
b) examine each layer l, in top-down order, and perform three tasks:
b.1) connect any new nodes assigned to that layer, choosing substrates and products for them based on the generality probabilities $s(l-1)$ and $s(l)$, respectively,
b.2) update the value of each node at layer l (note that the value of a node in the k'th round can be affected by nodes added in that same round),

b.3) examine, in order of decreasing value in that layer, whether any node should die (considering the case of cascade deaths).

Initially, we start with a small number of nodes at each layer, and form the edges between layers as if all nodes were new births. Unless noted otherwise, the execution of the model stops when the network reaches a given number of nodes.[4] We refer to each execution as an *evolutionary path*.

We have mathematically analyzed a significantly simpler version of *EvoArch* (See Appendix). Those simplifications are: static analysis (i.e., a non-evolving network), a node can compete only with the maximum-value node at that layer, each layer has the same number of nodes n, and $n\,s(L-1) \gg 1$. Under the previous assumptions, we derived a mathematical expression for the death probability ratio $r(l)$ for a node with the average number of products at layer l. Unfortunately the expression for $r(l)$ is not mathematically tractable and it does not allow us to examine whether it has a unique minimum. Numerically, however, that expression suggests that the ratio $r(l)$ has a unique minimum at a certain layer $\hat{l}$ that only depends on the generality probabilities and the competition threshold. Because the death probability decreases monotonically with $r(l)$ (see Figure 3), the previous observation means that the death probability has a unique maximum at layer $\hat{l}$, and it decreases monotonically at layers above and below $\hat{l}$. It is this death probability pattern that pushes, over several evolutionary rounds, the network to take the shape of a (generally asymmetric) hourglass with a waist at layer $\hat{l}$. The interested reader can find these derivations and further numerical results in the Appendix.

In the rest of the paper, the results are generated from discrete-time simulations of the *EvoArch* model. The benefits of such computational analysis are threefold: first, we do not need to make further simplifying assumptions. Second, we can examine the dynamics of the model, focusing on how the shape of the network changes with time. And third, we can quantify the variability of the results across many different evolutionary paths, instead of only looking at expected values.

3. BASIC RESULTS

In this section, we illustrate the behavior of the *EvoArch* model focusing on the width of each layer across time. We also introduce the main metrics we consider, and the default values of the model parameters.

The default values of *EvoArch*'s parameters are: $L = 10$ layers, $s(l) = 1 - l/L$ (i.e., the generality decreases as $0.9, 0.8, \ldots, 0.1, 0$, as we go up the stack), $c = 0.6$ (i.e., at least 3 out of 5 shared products), and $z = 1$ (see Figure 3). Each evolutionary path starts with 10 nodes at every layer, the average birth rate at each round is 5% of the current network size, and an evolutionary path ends when the network size reaches 500 nodes (but not sooner than 100 rounds). Unless noted otherwise, we repeat each experiment 1000 times, while the graphs show the median as well as the 10th, 25th, 75th and 90th percentiles across all evolutionary paths. We emphasize that the previous default values do *not* correspond, obviously, to the characteristics of the Internet stack. A parameterization of the model for that specific architecture is given in Section 5.4. *EvoArch* is a general model for layered protocol stacks and it does not aim to *only* capture the existing Internet architecture.

Figure 4 shows the width of each layer at the 50th and 100th rounds of the evolutionary process (typically, the network reaches
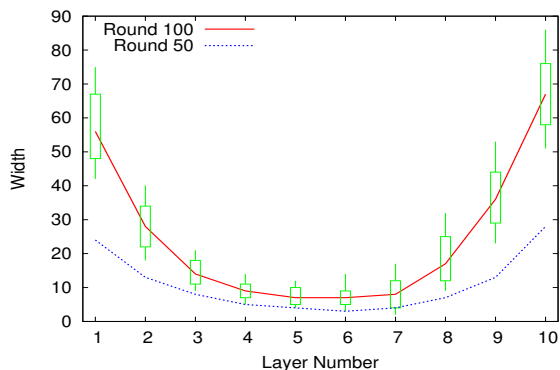
---

[4]We have also experimented with a termination condition based on the number of rounds, instead of the number of nodes. There is no significant difference as long as the network can evolve for at least few tens of rounds.


Figure 4: The median width of each layer at round-50 and round-100 (percentiles are only shown for the latter).

500 nodes in about 100–150 rounds). Note that, at least in terms of the median, the width decreases as we move from the bottom layer to a middle layer, around layer 5, and then it increases again as we move towards the top layer. There is some variability across evolutionary paths however, and so we further examine if the network structure has the shape of an hourglass in every evolutionary path.

To do so, we introduce a metric that quantifies the resemblance of a layered network structure to an hourglass. Let $w(l)$ be the width of layer l, i.e., the number of nodes in that layer at a given round. Let w_b be the minimum width across all layers, and suppose that this minimum occurs at layer $l = b$; this is the *waist* of the network (ties are broken so that the waist is closer to $\lfloor L/2 \rfloor$). Consider the sequence $X = \{w(l)\}, l = 1, \ldots b$ and the sequence $Y = \{w(l)\}, l = b, \ldots L\}$. We calculate the normalized univariate *Mann-Kendall statistic for monotonic trend* on the sequences X and Y as coefficients τ_X and τ_Y respectively [12]. The coefficients vary between -1 (strictly decreasing) and 1 (strictly increasing), while they are approximately zero for random samples. We define $H = (\tau_Y - \tau_X)/2$; H is referred to as the *hourglass resemblance* metric. $H = 1$ if the network is structured as an hourglass, with a strictly decreasing sequence of b layers, followed by a strictly increasing sequence of $L - b$ layers. For example, the sequence of layer widths $\{10,6,8,2,4,7,10,12,9,16\}$ (from bottom to top) has $w_b = 4$, $\tau_X = -0.67$, $\tau_Y = 0.81$ and $H = 0.74$. Note that we do *not* require the hourglass to be symmetric, i.e., the waist may not always be at the middle layer.

Figure 5 shows H (median and the previous four percentiles) as function of time. Note that it only takes few rounds, less than 10, for the median H to exceed 80%. By the 100th round, the median H is almost 95% and even the 10th percentile is more than 80%. This illustrates that *EvoArch* generates networks that typically have the shape of an hourglass. Even though the accuracy of the hourglass structure improves with time, the basic hourglass shape (say $H > 0.8$) is formed within only few rounds. Figure 6 shows the location of the waist as function of time, and the associated 10th and 90th percentiles across 1000 evolutionary paths. With the default parameter values, the median waist is almost always located at layer-6, while the 10th and 90th percentiles correspond to layers-5 and 7, respectively. So, even though there is some small variability in the exact location of the waist across time and across different evolutionary paths, the narrowest layer of the hourglass does not fluctuate significantly.

Figure 7 shows the median width of the typical waist (layer-6), as well as the median width of layers-1 and 9, as functions of time. Even though all three layers start from the same number of nodes, layers-1 and 9 become significantly wider with time, implying a
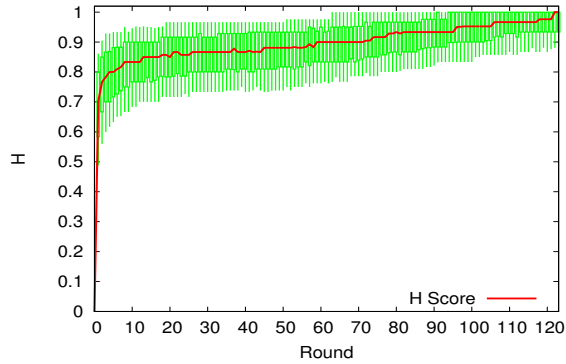
Figure 5: The hourglass resemblance metric H over time.

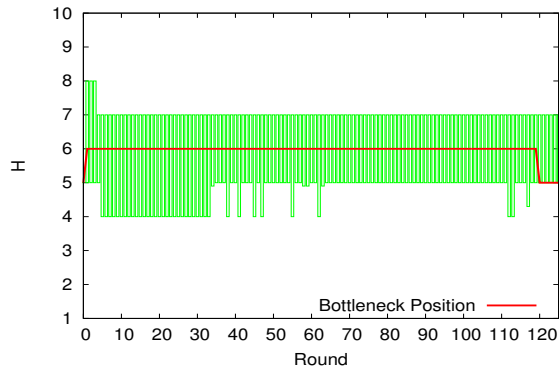**Figure 7: The median width of three layers over time.**

Figure 6: The location of the waist of the hourglass over time.

low death probability. On the other hand, the width of the waist remains relatively low compared to other layers. It typically decreases significantly in the first few rounds, as several of the initial nodes are "unlucky" in terms of products and die soon. It then slowly increases because higher layers become much wider, the birth rate increases with the size of the network, and few additional nodes at the waist can acquire significant value compared to the maximum-value node in that layer.

Obviously, the major question is: *why does EvoArch generate hourglass-shaped networks?* Let us discuss separately what happens at layers close to the top, close to the bottom, and close to the waist.

Because the generality probability $s(l)$ is quite low at layers close to the top, those nodes typically have a small number of products. This means that they rarely compete with each other, and so the death probability is close to zero. For instance, in the application-layer a new protocol can compete and replace an incumbent only if the former provides a very similar service with the latter (e.g., recall the example with FTP and HTTP).

At layers close to the bottom, the generality probability is close to one, and so those nodes have many shared products and thus several competitors. Their value is often similar however, because those nodes typically share almost the same set of products. Thus, the death probability at layers close to the bottom is also quite low.

At layers close to waist, where the generality probability is close to 50%, the variability in the number of products is maximized—recall that the variance of a Bernoulli random variable $X(p)$ is maximum when p=50%. So, few nodes in that layer may end up with a much larger number of products than most other nodes in the same layer, and so with a much higher value. Those nodes would compete with most others in their layer, often causing the death of
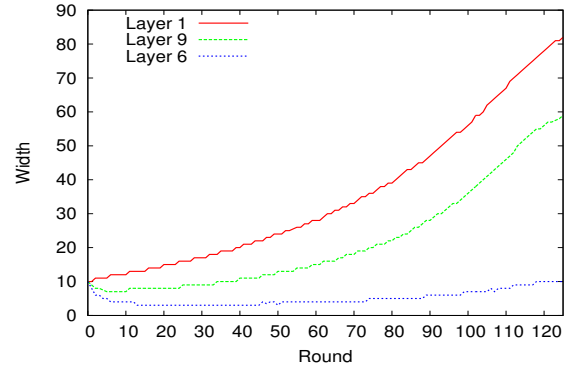
their competitors. In other words, the death probability at bottom and top layers is quite low, while the death probability close to the waist is higher. The birth rate, on the other hand, is the same for all layers, and so the network's middle layers tend to become narrower than the bottom or top layers.

The reader should not draw the conclusion from the previous simplified discussion that the waist is always located at the layer with $s(l)$=0.5. As will be shown in Section 5, the competition threshold c also affects the location of the waist. Also, it is not true that the node with the maximum value at the waist never dies. Section 6 focuses on these "extraordinary" nodes, showing that, even though they live much longer than almost all other nodes in their layer, under certain conditions they can also die.

4. ROBUSTNESS

In this section, we focus on the robustness of the hourglass resemblance metric H with respect to the parameters of the *EvoArch* model. The robustness study has two parts. First, we show that wide deviations from the default value, for a single parameter at a time, do not cause significant changes in H. Second, we show that even if we simultaneously and randomly vary all *EvoArch* parameters, the model still produces hourglass-like structures with high probability.

Let us first focus on the three most important *EvoArch* parameters: the competition threshold c, the generality probability vector s, and the mortality parameter z. We have also examined the robustness of H with respect to the number of layers L, the birth rate, the number of initial nodes at each layer, or the stopping criterion, but those parameters have a much smaller impact on H.

Figure 8-a shows the median H score (together with the previous four percentiles) as we vary c between 0 and 1. The value $c = 0$ corresponds to "global" competition meaning that two nodes of the same layer compete with each other even if they do not share any products. When the competition threshold is so low, the death probability becomes significant even at higher layers, as those nodes start competing even without sharing many products. Thus, the upper part of the network deviates from the hourglass shape.

When $c = 1$, on the other hand, a node u competes with node w only if the former shares *all* products of w. This means that nodes rarely compete, and so most layers grow randomly, without a significant death probability. There is a wide range of c (between 0.1 and 0.9) in which we get reasonably good hourglass structures ($H > 0.8$) in most evolutionary paths. The best results, however, are produced in the range $0.5 < c < 0.8$. *In that range, almost all evolutionary paths produce $H > 0.9$.*

To study the robustness of the model with respect to the generality vector s, we consider a function $s(l)$ comprising of two lin-
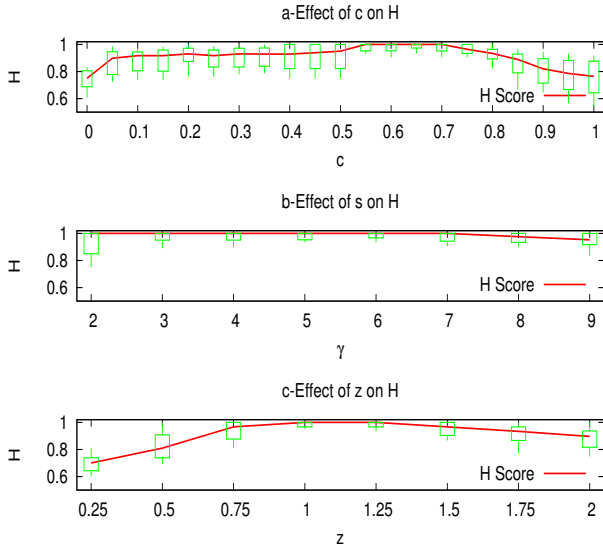
Figure 8: The hourglass resemblance score H as a function of the competition threshold c, the layer γ at which the generality is 50%, and the mortality parameter z.

ear segments that satisfy the constraints: $s(1)$=0.9, $s(\gamma)$=0.5 and $s(L)$=0, where γ is any layer between layer-2 and layer-(L-1). This function allows us to place the layer γ at which the generality probability is 50% at any (interior) layer of the architecture. Figure 8-b shows H as we vary the layer γ. The model is extremely robust with respect to variations in the generality vector s and layer γ.

Figure 8-c shows H as we vary the mortality parameter z. We limit the range of z to less than 2.0 so that the death probability is almost zero only if the value ratio r is close to one; this is not true for higher values of z (see Figure 3). Note that H is typically higher than 0.9 when $0.75 < z < 1.5$. For lower values of z, the death probability becomes so high that only the most valuable node tends to survive in several layers. When z is higher than 1.5, the death probability becomes too low and several layers grow randomly.

In the previous experiments, we varied one parameter at a time. We now examine the robustness of the model when we randomly sample each parameter value simultaneously from a certain range (a Monte Carlo method). Together with the previous three parameters (c, s and z), we also consider here variations in the number of layers L, the random number of initial nodes n_0 separately at each layer, the birth rate μ, and the maximum network size N_{max} at the end of an evolutionary path. A subtle point here is that, as L increases, we need to make sure that N_{max} is also increased (with a larger number of layers the network should be allowed to grow larger). For this reason we set $N_{max} = \eta L$, and vary the factor η instead of N_{max}. We consider the following range for each parameter:
$0.25 \le c \le 0.75$, $3 \le \gamma \le L-2$, $0.75 \le z \le 1.5$, $5 \le L \le 15$, $1\% \le \mu \le 10\%$, $5 \le n_0 \le 20$, and $25 \le \eta \le 55$.

We generate 1000 evolutionary paths, each with a randomly chosen value for all previous parameters. The CDF of the hourglass resemblance scores is shown in Figure 9. Even when we vary all parameters randomly in the given ranges, the score H is still higher than 0.9 in 68% of the evolutionary paths, and higher than 0.75 in 90% of the evolutionary paths. We manually examined some evolutionary paths in which the score H is lower than 0.5. They typically result from "bad" combinations of parameter values. For instance, a large value of c in combination with a large value of z severely sup-
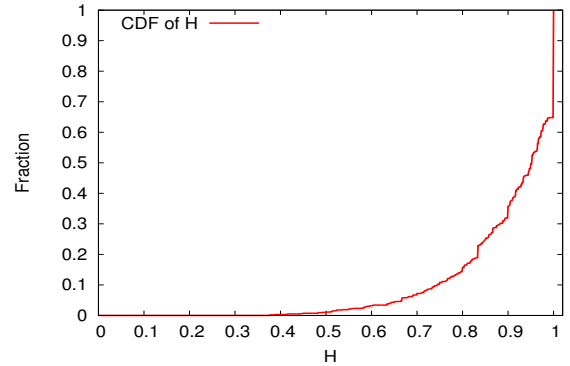

Figure 9: CDF of the hourglass resemblance score H when all parameters vary randomly in a certain range (see text).

press deaths in all layers, allowing the network to grow randomly. Or, a small value of c pushes the waist towards higher layers, while a small γ pushes the waist towards lower layers, causing deviations from the basic hourglass shape (e.g., a double hourglass shape with two waists).

5. LOCATION AND WIDTH OF WAIST

In this section, we focus on the effect of the three major *EvoArch* parameters (competition threshold c, generality vector s, and mortality parameter z) on the location and width of the waist.[5] We also estimate the value of these three parameters in the case of the current Internet architecture (TCP/IP stack), and discuss several implications about the evolution of the latter and its early competition with the telephone network. We also discuss how to design a new architecture so that it has higher diversity (i.e., larger width) at its waist compared to the TCP/IP stack.

5.1 Effect of competition threshold

Figure 10-a shows the location and width of the waist as c increases. Recall from Section 4 that the model produces high values of H when c is between 0.1 to 0.9. As c increases in that range, the waist moves lower and its width increases. The competition threshold c quantifies how similar the services or products of two protocols must be before they start competing. As c increases, it becomes less likely that two nodes compete. Especially at higher layers, where the generality is low and nodes have few products, increasing c decreases the frequency of competition and thus the death probability. This means that those higher layers grow faster. Nodes at lower layers, where $s(l)$ is close to one, have many overlapping products and so they are less affected by c. Thus, *increasing c pushes the waist towards lower layers.* The same reasoning (increasing c decreases the death probability) explains why *the waist becomes wider as c increases.*

5.2 Effect of generality vector

As in Section 4, we focus on a two-segment piecewise linear generality vector: the first segment extends between $s(1) = 0.9$ and $s(\gamma) = 0.5$, and the second extends between $s(\gamma)$ and $s(L) = 0$. This function allows us to control the layer at which the generality is 50% (and the variance of the number of products is maximized) by modifying the parameter γ. Figure 10-b shows the location and width of the waist as γ increases from layer-2 to layer-(L-1). Recall that *EvoArch* produces high hourglass resemblance scores through-

---

[5]Any parameter we do not mention is set to the default value given in § 3.
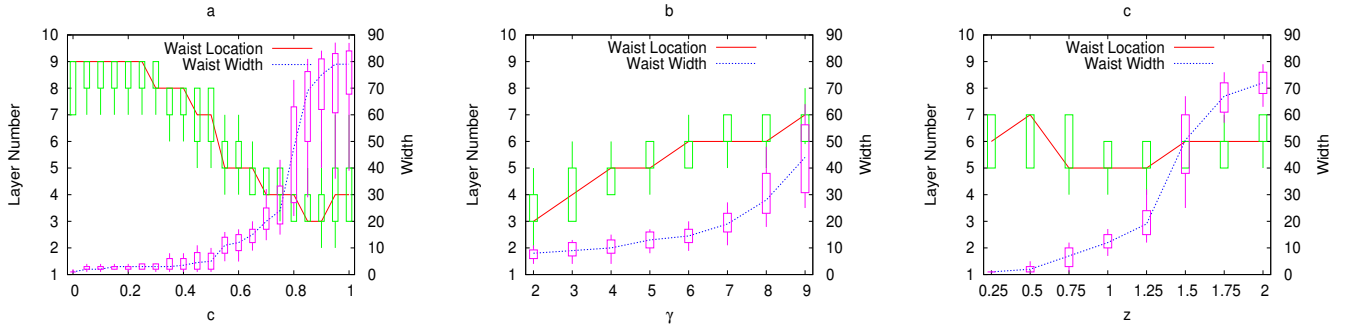
Figure 10: Location and width of the waist as a function of (a) competition threshold c, (b) layer γ with 50% generality, and (c) mortality parameter z.

out that range. The general observation is that *as γ increases, the location of the waist increases.* It is important however that *the location of the waist is* not *exactly equal to γ*; in other words, the variance in the number of products is not sufficient to predict the layer at which the death probability is highest (and the width is lowest). The competition threshold c also influences the location of the waist, as previously discussed.

As γ increases, the width of the waist also increases. The reason is that the location of the waist moves to layers with larger generality. For instance, Figure 10-b shows that when $\gamma = 5$ the median waist is also at layer-5, while when $\gamma = 8$ the median waist is at layer-6. Thus, when $\gamma = 5$, the generality of the waist is 50%, while when $\gamma = 8$ the generality of the waist is approximately 61%. Higher generality, however, means a larger number of products for new nodes at the waist, a higher evolutionary value relative to the node with the maximum number of products in that layer, and thus a higher probability of survival.

5.3 Effect of mortality parameter

Recall that z controls the shape of the death probability (see Figure 3), with lower values of z causing more lethal competition. Figure 10-c shows the location and width of the waist when z varies between 0.25 and 2.0. As expected, *as z increases, the width of the waist increases*—the reason is that the death probability decreases, allowing more nodes to survive even though they compete with other nodes. On the other hand, *the parameter z does not have a significant effect on the location of the waist.*

5.4 Implications for the TCP/IP stack

In the current Internet architecture, the waist is located at the network layer and so it is practically at the mid-point of the protocol stack (see Figure 1). Further, the waist is very narrow: just one dominant networking protocol (IPv4) and two major transport protocols (TCP and UDP). We have estimated a good parameterization of the *EvoArch* model for the case of the TCP/IP stack (based on trial-and-error and also exploiting the trends shown in Figure 10). The values are: $L = 6$ (we distinguish between application-layer protocols such as HTTP at layer-5, and individual applications such as Firefox at layer-6), $c \approx 0.7$, $\gamma = 3$, and $z \approx 0.3$. With these parameter values the waist is almost always located at layer-3 and it consists of only few nodes (typically less than three). The median H score is 1 and the 10-90th percentiles are 0.66 and 1, respectively.[6]

What do these parameter values imply about the evolutionary characteristics of the current Internet architecture? In terms of the

parameter c, a competition threshold around 70% implies that two protocols can co-exist in the TCP/IP stack as long as their relative product overlap (see Equation 2) is no more than about 70%; otherwise at least one of them will compete with the other. A good example of two protocols that co-exist at the same layer with little overlap in their services and functionality are TCP and UDP. The reason is that one of them is mostly used by applications that require reliability, while the other is chosen by a largely non-overlapping set of applications that prefer to avoid TCP's retransmissions, congestion control or byte-stream semantics. It is only few applications (e.g., DNS or Skype) that use both TCP and UDP.

The low value of z (approximately 0.3) implies that competition between protocols at the TCP/IP stack is very intense: *a protocol can survive only if its value is higher than about 90% of the value of its strongest component!* A good survival strategy for a new protocol u would be to *avoid competition* with the highest-value protocol in that layer w. This can be achieved if u has largely non-overlapping products with w; in other words, *the new protocol should try to provide mostly different services or functionality than the incumbent.* The relatively high value of c (70%) means that a significant degree of service overlap would be tolerated, making it easier for the new protocol to also support some of the legacy applications.

The previous point also suggests an intriguing answer to a historical question. *How can we explain the survival of the TCP/IP stack in the early days of the Internet, when the telephone network was much more powerful?* During the 70s or 80s, the TCP/IP stack was not trying to compete with the services provided by the telephone network. It was mostly used for FTP, E-mail and Telnet, and those services were not provided by the incumbent (telephone) networks. So, TCP/IP managed to grow and increase its value without being threatened by the latter. In the last few years, on the other hand, the value of the TCP/IP protocols has exceeded the value of the traditional PSTN and Cable-TV networks, and it is now in the process of largely replacing them in the transfer of voice and video.

In terms of the parameter s, the fact that the waist of the TCP/IP stack is located at the network layer implies that the generality of that layer is close to 50%. This means that a new protocol at the network layer would see the highest variability (i.e., maximum uncertainty) in terms of whether it will be selected as substrate from protocols at the next higher layer. So, from an architect's perspective, the network layer of the TCP/IP stack is the layer at which a new protocol would experience the maximum uncertainty in terms of deployment and ultimate success.

5.5 Future Internet architectures

EvoArch also gives some interesting insights about the evolvability of future Internet clean-slate architectures. Suppose that a net-

---

[6] A corresponding parameterization using the more realistic death-regulated birth process is given in Section 6.

work architect would like to ensure that there is more diversity (i.e., larger width) in the waist of a new architecture compared to the TCP/IP stack—this goal has been suggested, for instance, by Peterson et al. [17]. *How can the network architect increase the likelihood that the evolution of a new architecture will lead to a wider waist, with several surviving protocols?* Based on the previous results, this will happen if we increase z—it is unlikely however that a network architect can control the intensity of competition; that is largely determined by economic and deployment considerations. A second and more pragmatic approach is to design protocols that are largely non-overlapping in terms of services and functionality, as previously discussed, so that they do not compete with each other. This approach was discussed in the previous section.

A third approach is to design the architecture so that its waist is located at a layer with a high generality. As we saw in Figure 10-b, as we increase γ, increasing the generality of all layers, the waist moves higher, at a layer with higher generality. This also means that the waist is getting wider, allowing the co-existence of several protocols at that layer. How can this be done in practice? Suppose that we start from a 6-layer architecture X in which the waist is located at layer-3, and we want to redefine the functionality of each layer so that the waist of the new architecture Y is located at a higher layer. We should increase the generality of each layer (but still maintaining that $s(l)$ decreases as l increases) so that the corresponding protocols provide more general services in Y than in X. For instance, instead of defining HTTP as an application-specific protocol that is only used by web browsers, HTTP can be re-defined and used as a very general content-centric delivery protocol. This specific example, actually, has been recently proposed as a rather simple way to provide the benefits of clean-slate content-centric architectural proposals using an existing protocol [18].

6. EVOLUTIONARY KERNELS

It is often said that the core protocols of the Internet architecture (mostly IPv4, TCP and UDP) are "ossified", meaning that they are hard to modify or replace, creating an obstacle for network innovations [17]. At the same time however, they can be viewed more positively as the protocols that form the core of the architecture, creating a common interface between a large number of nodes at lower layers and a large number of nodes at higher layers. This is why we refer to them as *evolutionary kernels*, based on a similar concept about certain genes and gene regulatory networks in biology [10]. What can we learn from *EvoArch* about such ossification effects and evolutionary kernels? *Does the model predict the emergence of long-surviving nodes at the hourglass waist?* What is the reason that those nodes manage to survive much longer than their competitors? Do they ever get replaced by other nodes, and if so, under what conditions?

Let us focus on the waist—under the default parameters ($L = 10$) the waist is typically at layer-6. As previously discussed, the waist has the highest death probability, and so one may expect that it is unlikely to find any long-living nodes at that layer. We generate 1000 evolutionary paths, each lasting 100 rounds. At the end of the evolutionary path, we calculate the *maximum age* among all nodes that were ever born at layer-6. Figure 11 shows the CDF of the maximum age for various subsets of nodes: a) *the node with the maximum age* (we refer to such nodes as *rank-1 kernels* or simply *kernels*), b) the second older node (we refer to them as *rank-2 kernels*), c) all nodes, excluding only rank-1 kernels.

Note that almost all (rank-1) kernels survive for at least 50–60% of the entire evolutionary path. Actually, about 40% of the kernels are still alive at the end of the evolutionary path, meaning that their age is only determined by their birth time. On the other hand, the remaining nodes have much shorter life span. About 40% of them do not survive for more than a round or two and 90% of them survive for less than 20 rounds. The rank-2 kernels have much larger age than most nodes, but there is still a significant gap in the age of rank-1 and rank-2 kernels. So, our first observation is that *EvoArch predicts the emergence of very stable nodes at the waist that tend to survive for most of the evolutionary path.*
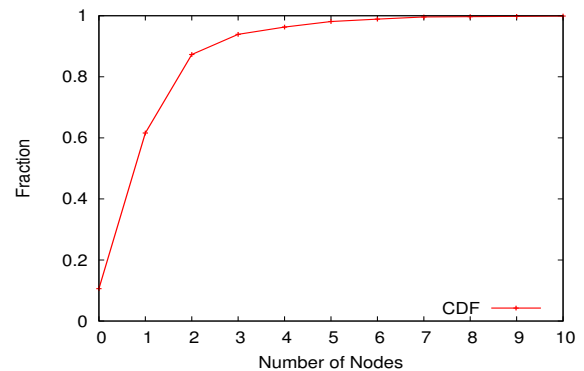
Figure 11 shows that only a small fraction of nodes survive for more than 50 rounds in an evolutionary path. So, let us identify in each evolutionary path those few nodes that survive for at least that long - we refer to them as "higher-rank kernels". Figure 12 shows the CDF of the number n of higher-rank kernels in each evolutionary path. In about half of the evolutionary paths only the rank-1 kernel exists. In almost all cases, $n \leq 4$. So, the number of nodes that can co-exist with the rank-1 kernel for more than 50 rounds is typically at most three. This confirms that it is difficult to survive for long at the waist in the presence of a rank-1 kernel. The nodes that manage to do so either have almost the same set of products with the rank-1 kernel (and thus almost the same value), or they have mostly different products than the rank-1 kernel, not competing with it. We have also examined the birth times of those nodes, and observed that in about 70% of the evolutionary paths rank-1 kernels are born earlier than higher rank kernels.

How large is the value of a kernel, and how can a kernel die? We define the *normalized value* $\hat{v}(u)$ of a node u at a given round as its value $v(u)$ divided by the value that u would have *if it was connected to all products at the next higher layer* in that round. So, $\hat{v}(u) \leq 1$. Note that because the death probability is almost zero when $r > 0.90$ (for the default value of z), if the normalized value
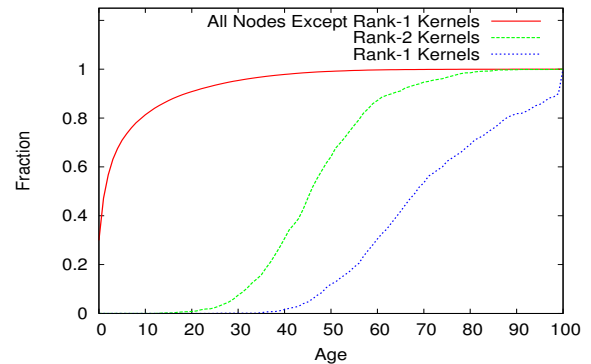

Figure 11: CDF of the age of various node subsets at layer-6.

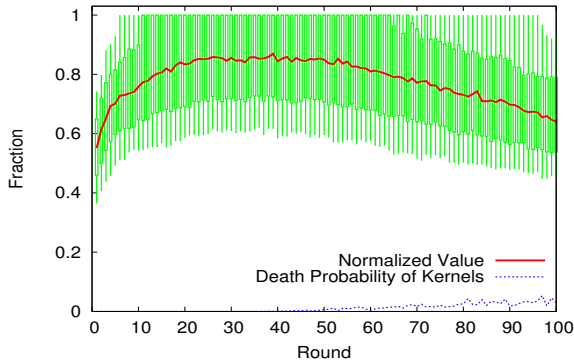Figure 12: CDF of the number of nodes at layer-6 surviving more than half of the evolutionary path.

Figure 13: Normalized value for kernels at layer-6 (basic birth process).

Figure 14: Normalized value for kernels at layer-6 (death-regulated birth process).

of a node is higher than 90% that node cannot die even if it had a higher-value competitor.

Figure 13 shows the normalized value of all (rank-1) kernels, based on 1000 evolutionary paths. In the first 10–20 rounds, the normalized value increases as the upper layer grows larger and the kernels acquire new products. Then, during the next 30–40 rounds their normalized value varies around 80–90%, which means that those kernels are unlikely to die, even if they face competition. During the last 30–40 rounds, however, the normalized value of many kernels gradually drops. To understand this trend, and to explain how a kernel can be replaced, we have to examine the birth process. As time progresses, the upper layers of the network grow larger (recall that the death probability is low at higher layers due to low competition). In the basic birth model, however, the birth rate is proportional to the size of the network, and new nodes are distributed uniformly across all layers. Thus, the birth rate at the waist also increases with time.

The previous increase has two consequences. First, as the layer above the waist increases in size, new potential products appear for both the kernel and its competitors. Each of these new nodes will select the kernel as substrate with probability $s(l_b)$, where l_b is the waist. Second, as the birth rate at the waist increases, it becomes more likely that a new node at that layer will acquire enough new products so that its value becomes comparable, or even higher, than the value of the kernel. In other words, the death of kernels is largely due to the birth of several new nodes at the next higher layer: *if the kernel fails to quickly acquire most of those new potential products at the next higher layer it will experience a decrease in its normalized value, becoming more vulnerable to new or existing competitors at its own layer.*

The previous discussion raises the question: *what if the birth rate is not the same at all layers?* Specifically, what if the birth rate at a layer is negatively correlated with the death probability at that layer? This modification of the model, which we refer to as *death-regulated birth process*, captures situations in which the implementation or deployment of new protocols at a certain layer is discouraged by the intense competition that one or more incumbent protocols create at that layer. Arguably, this is a more realistic model than the basic birth model we considered earlier.

In the death-regulated birth process, we maintain an estimate of the death probability $\tilde{d}(l)$ at layer l since the start of the evolutionary path. As in the basic birth process, the overall birth rate is proportional to the network size and the allocation of births to layers is random. However, a birth at layer l is successful with probability $1 - \tilde{d}(l)$; otherwise the birth fails and it is counted as a death in that layer.
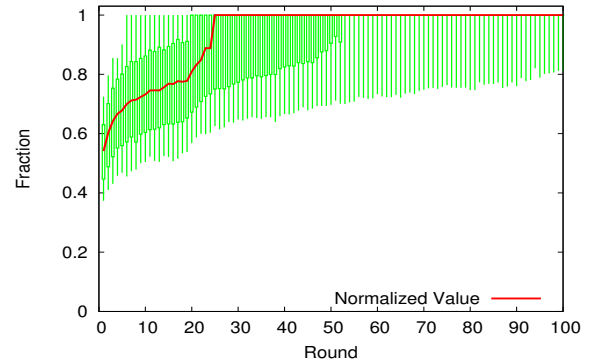
The death-regulated birth process creates a positive feedback loop through which the emergence of one or more kernels at the waist reinforces their position by decreasing the rate at which new nodes (and potential competitors) are created. Figure 14 shows the normalized value of the rank-1 kernel at layer-6, when we switch from the basic birth model to the death-regulated birth model after round-20. Note that the median normalized value (as well as the 25-th percentile) of the kernel becomes 100% within just few rounds. In other words, *with a death-regulated birth process it becomes practically impossible to replace a kernel at the waist of the hourglass.* Even when a new node u is somehow successfully born at the waist, the number of new nodes at the next higher layer is limited (because the birth process is also death-regulated at that layer) and so u's products will be most likely shared by the kernel. This means that u will face the kernel's competition, and so u will most likely die.

We have also estimated a good parameterization of the *EvoArch* model for the case of the TCP/IP stack using the death-regulated birth process (again based on trial-and-error and exploiting the trends shown in Figure 10): $L = 6$, $c \approx 0.7$, $\gamma = 3$, and $z \approx 0.5$. With these values the waist is located at layer-3, its median width is one node, the median width of layer-4 is four nodes, while the width of the remaining layers increases with time. The median H score is 1 and the 10-90th percentiles are 0.66 and 1, respectively.

6.1 Kernels in the Internet architecture

There are several interesting connections between what *EvoArch* predicts about kernels and what happens in the Internet architecture. There is no doubt that IPv4, as well as TCP and UDP, are the kernels of the evolving Internet architecture. They provide a stable framework through which an always expanding set of physical and data-link layer protocols, as well as new applications and services at the higher layers, can interoperate and grow. At the same time, those three kernel protocols have been difficult to replace, or even modify significantly. Further, the fact that new network or transport layer protocols are rarely designed today implies that the birth process at those layers is closer to what we call "death-regulated", i.e., limited by the intense competition that the kernel protocols create.

EvoArch suggests an additional reason that IPv4 has been so stable over the last three decades. Recall that a large birth rate at the layer *above the waist* can cause a lethal drop in the normalized value of the kernel, if the latter is not chosen as substrate by the new nodes. In the current Internet architecture, the waist is the network layer but the next higher layer (transport) is also very narrow and stable. So, the transport layer acts as an *evolutionary shield* for IPv4 because any new protocols at the transport layer are unlikely

to survive the competition with TCP and UDP. On the other hand, a large number of births at the layer above TCP or UDP (application protocols or specific applications) is unlikely to significantly affect the value of those two transport protocols because they already have many products. In summary, the stability of the two transport protocols adds to the stability of IPv4, by eliminating any potential new transport protocols that could select a new network layer protocol instead of IPv4.

In terms of future Internet architectures, *EvoArch* predicts that even if these architectures do not have the shape of an hourglass initially, they will probably do so as they evolve. When that happens, the emergence of new ossified protocols (kernels) will be a natural consequence. If the architects of such clean-slate designs want to proactively avoid the ossification effects that we now experience with TCP/IP, they should try to design the functionality of each layer so that the waist is wider, consisting of several protocols that offer largely distinct but general services, as discussed in Section 5.5.

7. QUALITY DIFFERENTIATION

So far we have assumed that the value of a protocol is only determined by the value of its products. It would be more realistic however to consider that the evolutionary value of a protocol also depends on other factors, which we refer to as *quality*. The "quality factor" should be interpreted broadly; it can capture properties such as performance, extent of deployment, reliability or security, clarity of the corresponding specification or other features. The quality factor also allows *EvoArch* to capture the effect of incremental improvements in existing protocols: such improvements do not create a new node in the model, but they increase the quality parameter of an existing node. In the following, we assume that the quality factor of a node is constant—an interesting extension of the model will be to consider time-varying quality factors.

In this section, we conduct a simple extension to the *EvoArch* model so that each protocol u has a certain quality factor $q(u)$. We are mostly interested in two questions. First, *how does this quality differentiation affect the shape of the resulting architecture?* And second, *focusing on the kernel nodes at the waist, do they tend to be nodes with the highest quality factor?* The quality of a node u is represented by a multiplicative factor $q(u)$, uniformly distributed in $[q_{min}, 1]$ (with $0 < q_{min} < 1$). The value of node u is then the product of the quality factor $q(u)$ with the quality-independent value of u:

$$v(u) = \begin{cases} q(u) \sum_{p \in P(u)} v(p) & l(u) < L \\ q(u) & l(u) = L \end{cases} \quad (5)$$

7.1 Effect of quality differentiation

We have repeated the analysis of Section 3 with the previous model extension, with $q_{min} = 0.8$. Figure 15 shows the median width of each layer with the default parameter values, based on 1000 evolutionary paths. *The resulting network continues to have the hourglass shape* (decreasing width up to a certain layer and then increasing width), even when nodes have different quality factors; the median H score is 1.0 and the 10th and 90th percentiles are 0.9 and 1.0, respectively. The location of the waist does not change as a result of the quality factor heterogeneity.

There are two interesting differences however. First, *the network grows at a slower pace*. This is a result of the increased death probability because of nodes with low quality factor. Second, *the lower part of the hourglass, below the waist, is now smaller in size than the upper part of the hourglass.* The reason is that nodes at lower layers have more competitors (due to higher generality at
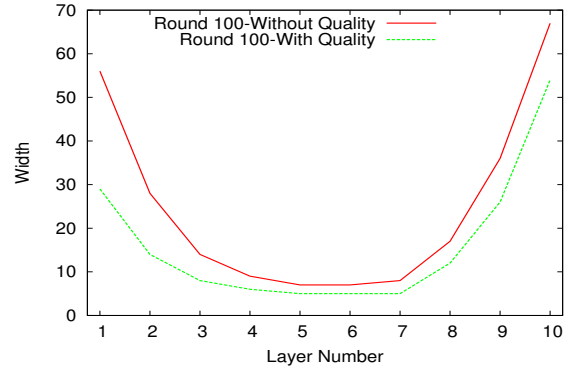

Figure 15: The median width of each layer when nodes have different quality factors (compared to the case that all nodes have the same quality).
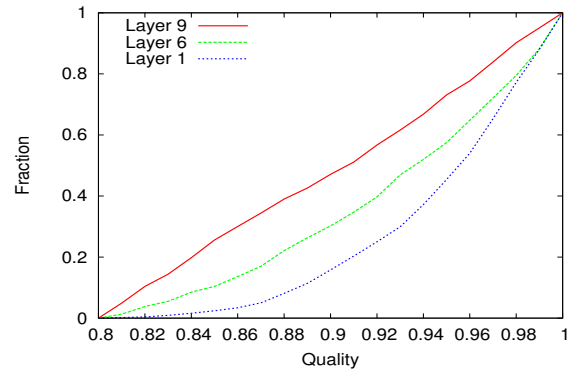

Figure 16: CDF of the quality factor $q(u)$ of the kernel node at the waist (layer-6). The corresponding CDFs for the oldest node at layer-1 and layer-9 are also shown.

those layers) than nodes at higher layers. When all nodes have the same quality, nodes at lower layers compete widely but they usually survive the competition because their product sets are similar, and thus their values are similar. With heterogeneous qualities, on the other hand, the value of nodes at lower layers can be significantly different, increasing the death probability compared to the case of homogeneous qualities. This increased death probability makes the size of the lower part of the hourglass smaller than the upper part.

It is interesting that the TCP/IP stack has a similar asymmetry, with the bottom part of the hourglass being smaller in terms of size than the upper part. *EvoArch* offers a plausible explanation for this effect: the heterogeneity among different protocols at the same layer, in terms of performance, security, extent of deployment, etc, increases the death probability at lower layers more than at higher layers.

7.2 Quality of kernel nodes

Section 6 focused on the evolutionary kernels at the waist of the hourglass. In the case of heterogeneous qualities, *an important question is whether the kernels tend to also have the highest quality factor. In other words, can we expect that the "best" nodes (i.e., highest quality factor) are also the strongest nodes in terms of value?*

Figure 16 shows the CDF of the quality factor of the kernel node at the waist (layer-6), based on 1000 evolutionary paths. As a reference point, we also show the CDF of the quality factor of the oldest node at layer-1 and at layer-9.

If the quality of a node has no effect on its death probability, we would expect that the CDF of the quality factor of the oldest node would be a straight line between 0.8 and 1 (recall that the quality factor varies uniformly in that range). This is the case at layer-9. In that layer, which is almost at the top of the stack, nodes rarely compete with each other because of their low generality and small number of products. So, their quality does not influence their age, and the quality factor of the oldest node varies randomly between 0.8 and 1.

At the bottom layer (layer-1) on the other hand, we observe a strong bias towards higher quality values. As discussed earlier in this section, the nodes at the bottom layer have high generality, many competitors, and so their quality factor strongly affects their value and death probability. *It is mostly high-quality nodes that survive at the bottom layers.*

At the waist, where only few kernel nodes survive for most of the evolutionary path, the bias towards higher quality values is much weaker. In about 30% of the evolutionary paths the quality factor of the kernel is less than 0.9, which means that *those kernels are in the bottom half of all nodes in terms of quality.* Similarly, the probability that a node with high quality factor (say $q(u) > 0.95$) becomes the kernel of the waist is only 40%.

7.3 Implications for IPv4 and IPv6

In the Internet architecture, it would be hard to argue that IPv4 has dominated at the network layer because it was an excellent protocol in terms of design. During the last 30–40 years, several other network-layer protocols have been proposed, and probably some of them were better than IPv4 in several aspects. *EvoArch* gives us another way to understand the success of IPv4, and to also think about future protocols that could potentially replace it. It will help if such future protocols are better than IPv4; that is not a sufficient condition to replace IPv4 however. If the potential replacements attempt to directly compete with IPv4, having a large overlap with it in terms of applications and services but without offering a major advantage in terms of the previous "quality factor", it will remain difficult to replace IPv4.

What does EvoArch suggest about IPv6 and the difficulty that the latter faces in replacing the aging IPv4? We should first note that IPv6 does not offer new services compared to IPv4; it mostly offers many more addresses.[7] This means that IPv6 has, *at most*, the same products with IPv4, and so the latter is its competitor. Further, because IPv6 is not widely deployed, it is reasonable to assume that its quality factor is much lower than that of IPv4. So, even if the two protocols had the same set of products, IPv4 has much higher value and it wins the competition with IPv6. The situation would be better for IPv6 under two conditions. First, if IPv6 could offer some popular new services that IPv4 *cannot* offer—that would provide the former with additional products (and value) that the latter does not have. Second, IPv6 should avoid competition with IPv4, at least until it has been widely deployed. That would be the case if IPv6 was presented, not as a replacement to IPv4, but as "the second network layer protocol" that is required to support previous new services.

8. RELATED WORK

To the extent of our knowledge, there is no prior work in modeling the evolution of protocol stacks or hourglass-like network architectures. At a high level however, our work is related to recent

efforts that develop a rigorous theory of network layering and architecture, mostly using mathematical tools from optimization and control systems [4]. We agree with those authors that network architecture can become the subject of more quantitative and rigorous scientific methods. We have a different view however on how to get there: instead of thinking about each layer as the solution to an optimization problem, we focus on the evolutionary process that shapes a network architecture over time and we emphasize the role of robustness and evolvability instead of optimality. Also relevant is the work of Csete and Doyle [5], which has emphasized the role of hierarchical modularity and evolution in both technological and biological systems. Those authors have also identified the significance of the hourglass (or bowtie) structure in the corresponding network architectures.

Recent work has investigated the competition between incumbent and emerging network architectures [13, 14]. Their focus, however, is mostly on the deployment share of each architecture using economic models from the literature of technology diffusion. We think that the deployment effects represent just one of many instances of competition between protocols at the same layer. In the economics literature, several authors have focused on standards, compatibility, and on the diffusion of new technologies (see, for instance, the early work by Farrell and Saloner [11] or the review [7]). That line of work focuses on the positive externalities created by the requirement for compatibility, and how those externalities often lead to the dominance of a single standard or protocol. To the extent of our knowledge, however, there is no work in economics about multi-layer architectures or networks that involve multiple standards, and no studies of the hourglass or bowtie structure that can emerge in such architectures.

In a 2001 IETF talk, Steve Deering alarmed the community that the waist of the Internet architecture is getting bigger, as more functionality (such as QoS or native multicast) was then proposed for IPv4 [8]—*EvoArch* suggests that a wider waist, with different protocols offering largely non-overlapping but general services, is actually a good way to increase the evolvability of an architecture. Popa et al. argue that HTTP (a layer-5 protocol) can become the new narrow waist of the Internet architecture because almost all applications and services today can run over HTTP [18]. We argue that instead of pushing a single-protocol waist from layer-3 to layer-5, a more evolvable network architecture should have a wider waist that does not include only one protocol. Otherwise, HTTP will also be considered "ossified" and an obstacle to innovation in few years from now. Culler et al. argue that the narrow waist of a sensor network architecture should not be the network layer, as in the current Internet, but a single-hop broadcast with a rich enough interface to allow multiple network protocols [6]. *EvoArch* can help designers of such new and special-purpose architectures to think about the ideal location of the hourglass waist considering the generality of the services provided at each layer.

9. CRITICISM

EvoArch does not consider the semantics and various practical aspects of specific layers, protocols or architectures. There are certainly several objections to this abstract modeling approach. We include here some of the concerns expressed by the anonymous reviewers, including a brief response.

One reviewer summarized most of the criticism with the following quote: *"The strength of this work is the simplicity of the model as it makes fairly general assumptions that are not tied into, for instance, the semantics of what each layer does. However, the failure to take semantics into account is its biggest failing as well."* *EvoArch* shows that it is not necessary to consider the layer seman-

---

[7]The original proposals for IPv6 included several novel services, such as mobility, improved auto-configuration and IP-layer security, but eventually IPv6 became mostly an IPv4-like protocol with many more addresses.

tics for the emergence of the hourglass structure; the semantics are probably important for other architectural characteristics but not for the hourglass structure.

Another high-level concern relates to the confirmation-bias risk: *"The problem with a paper like this is that we know the answer the model must produce, so (like Jeopardy) we have to find a model that can be tweaked to give that answer, and then see if we learn anything beyond what we already knew about the answer."* It is true that our objective has been to identify a model that can produce the hourglass structure. Additionally, however, our objective was to identify a general, parsimonious and explanatory model based on a small set of principles about layered and evolving network architectures. More importantly, *EvoArch* leads to several insights and explanations that were not expected from or "built-in" the model formulation.

Some reviewers offer different explanations for the narrow protocol waist at the network layer: *"The need for global addressing is a more important factor in having a single internetworking protocol than the evolutionary dynamics proposed here. IP does almost nothing else besides global addressing, and there is little reason to have two global addressing protocols."* This view is also reflected in the following quote: *"What if the reality is that the lower-most layer that provides end-to-end connectivity is the waist by default because that in effect represents the balance between generality and ease of use? It is hard to roll out an end-to-end connectivity service on top of lower-level primitives, and IP won because it got there (wide-area deployment) first."* Note that the previous plausible explanations for the presence of a narrow waist (with a single protocol at the network layer) do not explain, however, why we observe an hourglass structure. Another reviewer thinks that the hourglass structure is a result of a wide diversity of constraints at the lower layers and a wide diversity of services and applications at the upper layers: *"The more relevant dynamic at the bottom of the architecture is that there are different environmental niches (wireless, optical, etc.) that serve different needs in terms of deployability, bandwidth, and cost. So the model should have idiosyncratic applications at the top and idiosyncratic technologies at the bottom, tied together by layered protocols of varying generalities."* This is actually similar to *EvoArch*, even though it introduces more "semantics" about the function and constraints of lower layer protocols.

A more negative review questioned even whether the protocol stack has the shape of an hourglass: *"This model bears no intuitive relation to reality - in assumptions about the conditions under which protocols are invented, the process by which some are selected over others, or, indeed, that the protocol stack of the Internet is an hourglass (in practice, it is not: VPNs, tunnels, federation with other networks like the phone system, all complicate the picture)."* We believe that despite the former *architectural exceptions* (that are hard to place at a given layer), the Internet architecture is still shaped as an hourglass. Finally, we repeat what should be well-known to any scientist: the beauty and usefulness of a model is that it allows us to understand certain aspects of a system without having to describe or consider all the elements that constitute that system.

10. CONCLUSIONS

A main thesis behind this work is that we can study network architectures in a quantitative and scientific manner, in the same way that we study for instance the performance of transport protocols or the stability of routing protocols. In this spirit, we proposed a model for the evolution of layered protocol stacks. *EvoArch* is based on few principles about the generality of protocols at differ-ent layers, the competition between protocols at the same layer, and how new protocols are created. Even though *EvoArch* does not capture many practical aspects and protocol-specific or layer-specific details, it predicts the emergence of an hourglass architecture and the appearance of few stable nodes (not always of the highest quality) at the waist. Further, *EvoArch* offers some intriguing insights about the evolution of the TCP/IP stack, the competition between IPv4 and IPv6, and the evolvability of future Internet architectures. Possible extensions of *EvoArch* include a dynamic notion of quality (to capture, for instance, how the deployment of a protocol can change with time depending on the protocol's value), a growing number of layers as the complexity of the provided services increases with time, and architectures without strict layering.

Finally, we note that the presence of hourglass (or bowtie-like) architectures has been also observed in metabolic and gene regulatory networks [5, 20], in the organization of the innate immune system [2], as well as in gene expression during development [15]. Even though it sounds far-fetched, it is possible that there are similarities between the evolution of protocol stacks and the evolution of the previous biological systems. We explore these cross-disciplinary connections in on-going work.

Acknowledgements

11. REFERENCES

[1] S. Akhshabi and C. Dovrolis. The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture. In *ACM SIGCOMM*, 2011.

[2] B. Beutler. Inferences, Questions and Possibilities in Toll-like Receptor Signalling. *Nature*, 430(6996):257–263, 2004.

[3] G.E.P. Box. Robustness in the Strategy of Scientific Model Building. Technical report, DTIC Document, 1979.

[4] M. Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle. Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures. *Proceedings of the IEEE*, 95(1):255–312, 2007.

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

[6] D. Culler, P. Dutta, C. T. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, and J. Zhao. Towards a Sensor Network Architecture: Lowering the Waistline. In *USENIX HotOS*, 2005.

[7] P.A. David and S. Greenstein. The Economics of Compatibility Standards: An Introduction to Recent Research. *Economics of Innovation and New Technology*, 1(1):3–41, 1990.

[8] S. Deering. Watching the Waist of the Protocol Hourglass. 2001. www.iab.org/documents/docs/hourglass-london-ietf.pdf.

[9] C. Dovrolis. What would Darwin Think about Clean-Slate Architectures? *ACM SIGCOMM Computer Communications Review*, 38(1):29–34, 2008.

[10] C. Dovrolis and T Streelman. Evolvable Network Architectures: What can we Learn from Biology? *ACM SIGCOMM Computer Communications Review*, 40(2), 2010.

[11] J. Farrell and G. Saloner. Standardization, Compatibility, and Innovation. *The RAND Journal of Economics*, 16(1):70–83, 1985.

[12] M. Hollander and D. A. Wolfe. *Nonparametric Statistical Methods*. Willey Interscience, 1999.

[13] Y. Jin, S. Sen, R. Guérin, K. Hosanagar, and Z.L. Zhang. Dynamics of Competition between Incumbent and Emerging Network Technologies. In *NetEcon*, 2008.

[14] D. Joseph, N. Shetty, J. Chuang, and I. Stoica. Modeling the Adoption of New Network Architectures. In *Proceedings of ACM CoNEXT*, 2007.

[15] A.T. Kalinka, K.M. Varga, D.T. Gerrard, S. Preibisch, D.L. Corcoran, J. Jarrells, U. Ohler, C.M. Bergman, and P. Tomancak. Gene Expression Divergence Recapitulates the Developmental Hourglass Model. *Nature*, 468(7325):811–814, 2010.

[16] NSF-10528. Future Internet Architectures (FIA). *National Science Foundation*, 2010.

[17] L. Peterson, S. Shenker, and J. Turner. Overcoming the Internet Impasse through Virtualization. In *ACM SIGCOMM HotNets*, 2004.

[18] L. Popa, A. Ghodsi, and I. Stoica. HTTP as the Narrow Waist of the Future Internet. In *ACM SIGCOMM HotNets*, 2010.

[19] J. Rexford and C. Dovrolis. Future Internet Architecture: Clean-Slate versus Evolutionary Research. *Communications of the ACM*, 53:36–40, 2010.

[20] J. Zhao, H. Yu, J.H. Luo, Z.W. Cao, and Y.X. Li. Hierarchical Modularity of Nested Bow-Ties in Metabolic Networks. *BMC Bioinformatics*, 7(1):386, 2006.

APPENDIX

The objective of the following analysis is to derive the death probability ratio $r(l)$ for an "average node" at layer l, i.e., for a node with the mean number of products at layer l.

We can do so under several assumptions that simplify the *EvoArch* model significantly:

1. The network is static (i.e., non-evolving).

2. Each layer has the same number of nodes n.

3. $n\, s(L-1) \gg 1$ (recall that $s(L-1)$ is the lowest non-zero generality across all L layers).

4. A node can only have one competitor: the node with the largest number of products at that layer.

The following analysis focuses on two nodes at each layer l: a node u with the mean number of products at layer l, and a node u_m with the expected value of the *maximum* number of products at layer l. Based on assumption (4), the death probability of node u depends on the ratio $r(l) = v(u)/v(u_m)$ between the value of node u and the value of node u_m. Let $\rho(l)$ be the number of products of node u and $\rho_m(l)$ the number of products of node u_m. We need a fifth assumption, regarding nodes u and u_m:

$$5. \quad \frac{v(u)}{v(u_m)} = \frac{\rho(l)}{\rho_m(l)} \tag{6}$$

Based on the previous five assumptions, we derive next the following mathematical expression for the death probability ratio $r(l)$:

$$r(l) = 1 - \left(1 - \frac{\lfloor n\, s(l) \rceil}{\lfloor \rho_m(l) \rceil}\right) \sum_{i=\lceil c \times \lfloor n\, s(l) \rceil \rceil}^{\lfloor n\, s(l) \rceil} \frac{\binom{\lfloor \rho_m(l) \rceil}{i}\binom{n - \lfloor \rho_m(l) \rceil}{\lfloor n\, s(l) \rceil - i}}{\binom{n}{\lfloor n\, s(l) \rceil}} \tag{7}$$

where $\lfloor x \rceil$ represents the "round to closest integer" operator, while

$$\rho_m(l) = \sum_{i=0}^{n} i \left(F(i)^n - F(i-1)^n \right) \tag{8}$$

where

$$F(x) = \sum_{i=0}^{x} \binom{n}{i} s(l)^i \left(1 - s(l)\right)^{n-i}. \tag{9}$$

Unfortunately the previous expression for $r(l)$ is not tractable and it does not allow us to examine whether $r(l)$ has a unique minimum. We observed numerically, however, that the ratio $r(l)$ has a unique minimum at a certain layer $\hat{l}$ that only depends on the generality probabilities and the competition threshold (see Figure 17). Because the death probability decreases monotonically with $r(l)$, the previous observation about $r(l)$ means that the death probability has a unique maximum at layer $\hat{l}$, and it decreases monotonically at layers above and below $\hat{l}$. It is this death probability pattern that pushes, over several evolutionary rounds, the network to take the shape of a (generally asymmetric) hourglass with waist at layer $\hat{l}$.
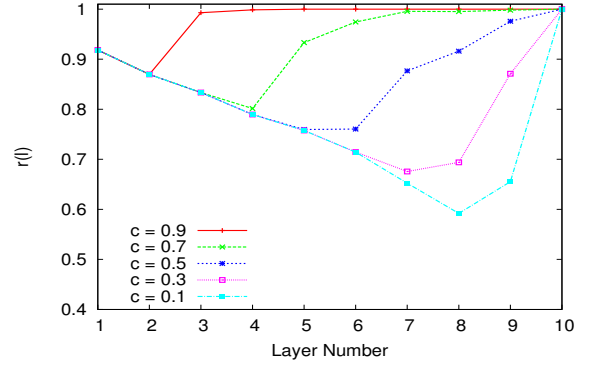

Figure 17: The death probability ratio $r(l)$ as a function of the layer l for different values of the competition threshold c. The parameters are: $L = 10$, $n = 50$, and $s(l) = 1 - l/L$.

In the following, we derive the expression for $r(l)$ given in (7).

Proof

Based on the earlier definition of nodes u and u_m at layer l, the death probability ratio for node u is:

$$r(l) = \begin{cases} \frac{v(u)}{v(u_m)} & \text{if } u_m \text{ is a competitor of } u \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

while based on assumption (5) the previous value ratio is equal to the corresponding number of products ratio,

$$r(l) = \begin{cases} \frac{\rho(l)}{\rho_m(l)} & \text{if } u_m \text{ is a competitor of } u \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

We can rewrite the ratio $r(l)$ as

$$r(l) = P[u_m \text{ is competitor of } u] \times \frac{\rho(l)}{\rho_m(l)} + \left(1 - P[u_m \text{ is competitor of } u]\right) \times 1. \tag{12}$$

So, we need to derive $\rho(l)$, $\rho_m(l)$ and the probability $P[u_m \text{ is competitor of } u]$.

The number of products for a node at layer l is given by the random variable

$$X = \max\{B(n, s(l)), 1\}. \tag{13}$$

where $B(n, s(l))$ is a random variable that follows the binomial distribution with parameters n and $s(l)$. Based on assumption (3), the expected value $n \times s(l)$ of this binomial distribution is much larger than one at all layers, and so the average number of products at layer l can be approximated as:

$$\rho(l) \approx n \times s(l). \tag{14}$$

We derive the expected value $\rho_m(l)$ of the maximum number of products at layer l using ordered statistics.

Specifically, if $X_1, X_2, ..., X_n$ are n I.I.D. random variables with cdf and pmf $F(x)$ and $f(x)$, respectively, the pmf of the k'th ordered statistic $X_{(k)}$ (in increasing order) is given by

$$P(X_{(k)} = x) = \sum_{j=0}^{n-k} \binom{n}{j} \left(\left(1 - F(x)\right)^j \left(F(x)\right)^{n-j} \right.$$
$$\left. - \left(1 - f(x)\right)^j \left(F(x) - f(x)\right)^{n-j} \right). \quad (15)$$

The n'th order statistic represents the maximum of $X_1, X_2, ..., X_n$, and its pmf $f_m(x)$ is:

$$f_m(x) = P(X_{(n)} = x) = F(x)^n - \left(F(x) - f(x)\right)^n. \quad (16)$$

Suppose now that the random variable X_i represents the number of products of a node i at layer l $(i = 1 \ldots n)$, as defined in (13). If $n \times s(L-1) >> 1$, X can be approximated as a binomial random variable with parameters n and $s(l)$, and so

$$f(x) = P(X = x) = \binom{n}{x} s(l)^x \left(1 - s(l)\right)^{n-x} \quad (17)$$

and

$$F(x) = P(X \le x) = \sum_{i=0}^{\lfloor x \rfloor} \binom{n}{i} s(l)^i \left(1 - s(l)\right)^{n-i}. \quad (18)$$

Replacing $f(x)$ and $F(x)$ in $f_m(x)$, and assuming that x is integer, we get that

$$f_m(x) = \left[\sum_{i=0}^{x} \binom{n}{i} s(l)^i \left(1 - s(l)\right)^{n-i} \right]^n$$
$$- \left[\sum_{i=0}^{x} \binom{n}{i} s(l)^i \left(1 - s(l)\right)^{n-i} \right.$$
$$\left. - \binom{n}{x} s(l)^x \left(1 - s(l)\right)^{n-x} \right]^n$$
$$= F(x)^n - F(x-1)^n. \quad (19)$$

$\rho_m(l)$ is then given by

$$\rho_m(l) = \sum_{i=0}^{n} i f_m(i). \quad (20)$$

Replacing (19) in (20), we get (8).

Figure 18 shows $\rho(l)$ and $\rho_m(l)$ as a function of layer l.

The final step is to derive the probability that u_m is a competitor of u. This probability, denoted by $c(l)$, refers to the event that u shares at least a fraction c of its products with u_m at layer l, where c is the competition threshold.

We define the function

$$h_l(x, y, z) = P[\text{A node with } x \text{ products at layer } l \text{ shares exactly}$$
$$y \text{ products with a node that has } z \text{ products}]$$

which is equal to

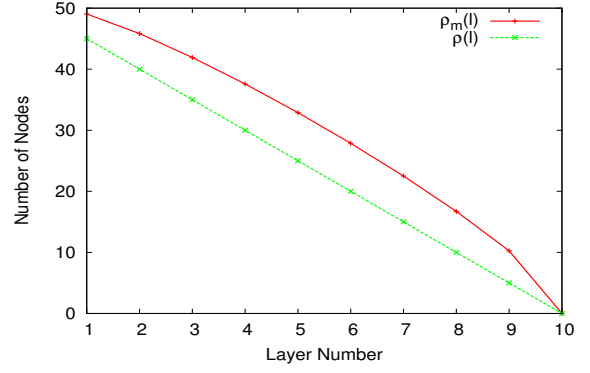$$h_l(x, y, z) = \frac{\binom{z}{y} \binom{n-z}{x-y}}{\binom{m}{x}}. \quad (21)$$

Figure 18: $\rho(l)$ and $\rho_m(l)$ as a function of layer l. **The parameters are:** $L = 10$, $n = 50$, **and** $s(l) = 1 - l/L$.
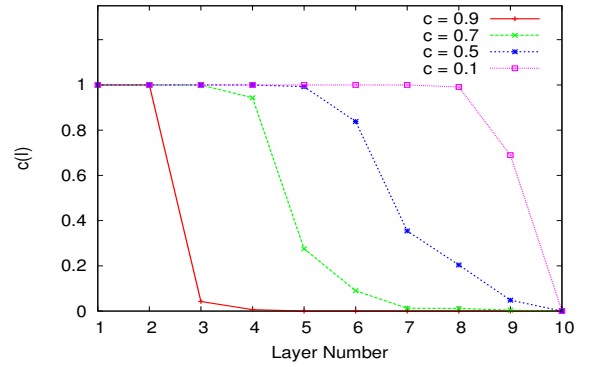

Figure 19: The probability $c(l)$ as a function of layer l for different values of the competition threshold c. The parameters are: $L = 10$, $n = 50$, **and** $s(l) = 1 - l/L$.

$c(l)$ can then be expressed as

$$c(l) = P[u_m \text{ is a competitor of } u]$$
$$= P[\text{a node with } \rho(l) \text{ products at layer } l \text{ shares at least}$$
$$c \, \rho(l) \text{ products with a node that has } \rho_m(l) \text{ products}]$$
$$= \sum_{i=\lceil c \times \lfloor \rho(l) \rfloor \rceil}^{\lfloor \rho(l) \rceil} h_l(\lfloor \rho(l) \rfloor, i, \lfloor \rho_m(l) \rfloor)$$
$$= \sum_{i=\lceil c \times \lfloor \rho(l) \rfloor \rceil}^{\lfloor \rho(l) \rceil} \frac{\binom{\lfloor \rho_m(l) \rfloor}{i} \binom{n - \lfloor \rho_m(l) \rfloor}{\lfloor \rho(l) \rceil - i}}{\binom{n}{\lfloor \rho(l) \rceil}}. \quad (22)$$

Figure 19 shows the probability $c(l)$ as a function of layer l.

Finally, combining the expressions (14), (8), (22) and (12), we get the final result in (7).