There is more to Internet invariants than meets the eye

Chris Misa[1], Walter Willinger[2], Ramakrishnan Durairajan[1,3], Reza Rejaie[1]

1. University of Oregon 2. NIKSUN, Inc. 3. Link Oregon

Abstract

A rich body of literature assembled over the last 30 years shows that traffic traversing wide-area Internet links is consistent with self-similar (temporal) scaling behavior and that sets of observed addresses have multifractal (spatial) scaling behavior. In view of this empirical evidence, these behaviors cannot be viewed as mere mathematical curiosities but should justifiably be called *invariants* of measured Internet traffic (Internet invariants, for short). At the same time, it is fair to say that the early architects of the Internet were largely unaware of these properties and certainly did not intend to design a network so that the traffic traversing its links would exhibit self-similar scaling in time or multifractal scaling in the IP address space.

In this paper, we resolve this apparent disconnect between architectural intentions and observed behaviors by applying the perspective of Highly Optimized Tolerance (HOT). In particular, we take inspiration from studies on the origins of (temporal) self-similarity in measured Internet traffic but focus on a fundamentally new approach to understanding multifractal (spatial) scaling behavior. Specifically, we examine whether this invariant can be viewed as a visible hallmark of underlying but largely unknown robust design efforts, and explore a reverse-engineering approach to determine the concrete nature of the constrained optimization problems that these robust designs solve. Based on the insights gained from such reverse-engineering efforts, we demonstrate with illustrative examples the benefits of subsequent attempts at forward-engineering—systematically leveraging the identified robust designs in order to provide scientifically sound intellectual foundations and practical principles for designing future networked systems.

1 Introduction

As articulated eloquently more than 20 years ago in [19], *"the Internet is a moving target: it is subject to major changes in how it is used, with new applications sometimes virtually exploding on the scene and rapidly altering the lay of the land."* In such an environment where change seems to be the only constant, networking researchers have been struggling to ensure that their research efforts or prototype implementations don't become obsolete in the blink of an eye but have long-term relevance. The authors in [19] (concerned mainly with faithfully simulating even just parts, or specific aspects of the Internet) argue that one way to cope with this constant change is to look for *invariants*; facets of Internet behavior that have been empirically shown to hold in a wide range of environments and be largely immune to changes up and down the protocol stack and across this global-scale network.

Despite their potential utility, in this work we argue that there is more to Internet invariants than meets the eye. With the exception of patently human-related behavior (*e.g.,* diurnal patterns of activity, Poisson session arrivals [19]), Internet invariants are rarely the outcomes of explicitly specified design objectives and definitely not the result of self-organized criticality [6]. This leads directly to fundamental questions such as *"If a given invariant is the observable manifestation of an underlying but unknown robust design effort, then what is the problem that is being solved by that design effort?"* Answering such questions involves a process called *reverse-engineering* to formulate the optimization problem that the underlying design effort is attempting to solve. Applying the ensuing solutions requires a process of *forward-engineering* to systematically study the solutions' properties as a function of alternate optimization criteria, resource constraints, or sources of uncertainty. As a framework for realizing such processes, we leverage ideas of Highly Optimized Tolerance (HOT) [10, 15, 11], where deliberate robust designs aim for specific levels of tolerance to uncertainty which must be traded off against the cost of some compensating resources. Leveraging HOT-based understanding of Internet invariants potentially yields powerful means to develop future network designs that focus on first-principles approaches to fundamental questions (see [13] and references therein), rather than seeking "quick and dirty" solutions to transient problems.

To illustrate the potential of HOT-enabled approaches to comprehending Internet invariants, in this work we focus in particular on Internet *traffic* invariants; that is, well-documented and ubiquitous properties of real-world Internet traffic that hold irrespective of where in the wide-area Internet, at what point in time, or under what networking conditions the traffic was measured (but see the caveat below). The

existence of such traffic invariants provides researchers with much-needed grounds to assess existing or study new approaches to Internet-specific traffic processing systems, protocols, applications, or services.

Internet traffic naturally decomposes into a *temporal* dimension and a *spatial* dimension. While the former is concerned with temporal aspects and properties of measured packet traffic on a link within the network (*e.g.,* packet per second), the latter refers to the structure of how sets of observed addresses in measured packet traffic on a link within the network manifest in the address space (*e.g.,* observed addresses per prefix). Study of the temporal dimension of Internet traffic unfolds in a rich literature shaped by the discovery some 30 years ago [27] of a facet commonly referred to as *self-similar scaling* (or self-similarity, for short) [27, 32]. In Section 2, we briefly review the current understanding of its root causes, especially the connection to heavy-tailed files or flows (i.e., mice-elephant paradigm) and recall a prior but largely overlooked HOT-based effort from the early 2000s at reverse-engineering that identifies self-similarity as an ubiquitous outward manifestation of a robust and fundamental, but not explicitly stated Internet design principle [47].

In this work, we explore how taking inspiration from prior work on temporal self-similarity leads to a fundamentally new approach to characterizing the *spatial* traffic invariant of multifractal scaling. Despite its potentially equal relevance to present-day and future Internet design problems, the spatial dimension of Internet traffic (*i.e.,* the structure of how sets of observed addresses in measured packet traffic on a link within the network populate the address space) has received only cursory attention to date and lacks the understanding we require from an invariant so it can serve, in the words of [19], *"as a point of stability upon which we can then attempt to build."*

The main goal of this paper, then, is to lay the groundwork for future investigation of this spatial dimension of Internet traffic by starting with what is currently known about how observed addresses in measured Internet traffic are arranged in the IP address space. Our point of departure is a preliminary empirical finding, first reported some 20 years ago in [24, 25, 7], that observed addresses exhibit multifractal scaling behavior. Unfortunately, this work produced inconclusive results due to reliance on pictorial rather than statistical evidence and use of poorly-understood analysis techniques such as the histogram method [17, 36]. Critically, from the perspective articulated above, it did not perform sufficiently deep of an analysis to truly understand and leverage multifractal scaling as an Internet traffic invariant; that is, it did not seek to identify an underlying robust design effort that could explain why multifractal scaling is widely observed and can hence be rightly called an invariant. The importance of such a foundational understanding was not lost on the authors of the original study [24] who state succinctly: *"without a convincing description of how [multifractal] address structure arises, the results of the explorations [in their paper] must be considered preliminary."*

By advocating for a comprehensive view of Internet invariants, our current effort advances understanding of apparent multifractal address structure in measured Internet traffic along the following two fronts:

- We revive study of the spatial aspects of measured Internet traffic but approach the problem from a new "networking first, statistical inference second" perspective. In particular, Section 3 presents new results on constructing and empirically validating an evocative mathematical model in the form of a finite conservative cascade. This model accounts for the underlying Internet mechanisms and processes by which IP addresses are allocated and used, explaining why sets of IP addresses that are responsible for the traffic traversing a link in the network are necessarily multifractal in nature.

- Section 4 pushes beyond "what meets the eye" to propose reverse- and forward-engineering efforts to understand *(i)* in what sense this new Internet invariant is also the ubiquitous outcome of robust designs concerning how IP addresses are actually allocated to various Internet stakeholders, *(ii)* what looking through the multifractal lens can tell us about the ingenuity of the original Internet designers who were first and foremost concerned with providing the necessary stability for the network to continue to grow in size and *(iii)* what are the possible ramifications of this new invariant for designing future networked systems.

Caveats: While this work demonstrates how empirical study of an observed Internet invariant combined with consideration of the highly engineered nature of the Internet yields fundamental insights beyond what is explicitly documented in RFCs or standard textbooks, we hasten to mention that it describes work-in-progress, is admittedly incomplete, and prone to misunderstandings. For example, our focus in this paper is on Internet invariants that concern the wide-area Internet and manifest in traffic over access or backbone links that is assumed to be generated by heterogeneous sets of hosts, applications, and services. In particular, we are not claiming that Internet invariants are encountered in any perceivable network environment, under any traffic conditions, or at all times. For example, we are not saying that specialized environments such as intra-datacenter networks [45, 8] or Industry 4.0 industrial networks [38] will observe traffic invariants such as (temporal) self-similar scaling or (spatial) multifractal scaling. At the same time, our pursuit of HOT-based approaches enables a path to understand what the absence or presence of a given traffic invariant might says about the design principles at work for a given network environment.

2 There is more to self-similarity ...

We first consider the well-documented and ubiquitously observed self-similarity property of measured Internet traffic, listed in [19] as a "promising candidate" for an Internet invariant. We briefly summarize in this section our present-day understanding of this candidate and discuss ways in which this understanding can be solidified and improved beyond what has been reported in the existing literature on this topic.

2.1 The known pieces of the puzzle

The initial empirical studies of observed self-similar scaling in measured Internet traffic were published in [27, 32]. Almost immediately, the focus shifted from quantifying the extent of self-similar scaling in measured traffic traces to trying and answer the question "Why self-similar?" This question was subsequently answered in [42, 14] where the authors provide an empirically-validated physical explanation that leveraged a generative mathematical construct known as the Mandelbrot process (*i.e.,* superposition of IID renewal reward processes with heavy-tailed rewards/durations). This construction identified the ubiquitously observed heavy-tailed nature of individual flows or file transfers that make up the aggregate self-similar packet traffic as root cause for the observed self-similar scaling behavior [32, 42, 14, 4]. Subsequently, this root cause has also been termed the mice-elephant property of Internet traffic - while most flows are small in size and short-lived in time (mice), a few large and long-lasting flows (elephants) are responsible for most of the traffic. Importantly, answering the question "Why self-similar?" with "Because of heavy-tailed flow sizes/file transfers!" triggered a yet more profound question in the form "Why does Internet traffic have the mice-elephant property?"

This question was answered in large part in [47], where the authors expanded on the theme of Highly Optimized Tolerance (HOT) which argues that heavy tails are the ubiquitous outcome of robust designs of complex engineered systems and not the result of some phase transition as is promoted by advocates of Self-organized Criticality (SOC) but which lacks any engineering relevance. More precisely, the HOT approach described in [47] treats the layout of a Web site as an optimal design problem where the design objective is assumed to be the minimization of the average file size (which roughly translates into minimizing the latency that the user experiences in downloading files while browsing the Web site), subject to navigability constraints (*e.g.,* the total number of files on the Web site). When formulating and solving Web layout designs as such constrained optimization problems, the authors of [47] report analytic and simulation results that show that the obtained solutions invariably produce heavy-tailed file sizes. This particular reverse-engineering effort then suggests that heavy tails and, in turn, self-similarity, are a permanent and omnipresent feature of Internet traffic, and not an artifice of current applications or user behavior. In fact, as argued in [47], the obtained findings suggest that *"[self-similar] traffic has some aspects which are intrinsic to at least the current dominant application, the WWW, and may be even more intrinsic to any application which organizes information for human consumption."*

2.2 Some missing pieces of the puzzle

The insight that the described HOT-based approach provides for recognizing the temporal self-similarity invariant of Internet traffic (or equivalently, its ubiquitous mice-elephant property) as outward sign or observable manifestation of a fundamental underlying robust Internet design principle is profound; the identified design principle calls for applications to organize information for human consumption. Nevertheless, the type of reverse-engineering efforts motivated by this approach have received surprisingly little attention from the community in the last 25 years. A notable exception are related studies that aim at developing a mathematical theory of network architectures by considering "layering" as optimization decomposition (see [13] and references therein). Below, we discuss a number of topics that explicitly concern the design of future networked systems and require further reverse- or forward-engineering efforts (beyond the scope of this paper) to substantiate our stated hypotheses that are largely informed by our current understanding of the corresponding past reverse-engineering efforts.

Why is it important for applications to organize information for human consumption? One important insight gained from [47] is that the heavy tailed characteristics of Web traffic are likely to be an invariant of much of the future network traffic, regardless of the dominant applications. Intuitively, since most human-oriented data communication processes involve both active navigation and ultimately the transfer of large objects, they split much of the traffic naturally into mice (*e.g.,* thumbnails, titles, abstracts) and elephants (*e.g.,* high-resolution picture or video, full paper or entire book). As a result, this "encoding" can be expected to persist, irrespective of the applications or services considered. At a more fundamental level, we hypothesize that the root cause for this ubiquitous encoding or organization of information for human consumption is the human brain's highly constrained information processing capability—it works at a mind-numbing slow pace of just around 10 bits per second [46]. The reverse-engineering efforts considered in [47] provide a promising roadmap for rigorously proving these intuitive arguments and hypotheses.

But future AI-based applications and services will certainly change everything! Or will they? Looking towards the future [26], while the insights gained from this line of research caution for continued vigilance with respect to the validity of the time-tested split of modern Internet traffic into elephants and mice, they also suggest that even with

the emergence of new AI-driven applications and services (*e.g.,* LLM inference), the mice-elephant split of Internet traffic will largely persist and will remain an invariant of future Internet traffic because even this new generation of applications can reasonably be viewed as striving to organize information for human consumption. However, to rigorously establish such futuristic scenarios, alternative HOT formulations that result in constrained optimization problems with potentially different design objectives and resource constraints (*e.g.,* different from minimizing the average download time and with different navigability constraints) may need to be explored to see whether they support our current understanding or identify explicit design objectives that expand on or differ from the notion of organizing information for human consumption.

There are no signs of self-similar traffic in my network! Even though [47] provides a promising blueprint for successful reverse-engineering (*i.e.,* formulating the constrained optimization problem that is being solved by a given robust network design), we are not aware of significant subsequent work that leverages the insights obtained from this reverse-engineering and applies them to the process of forward-engineering—*i.e.,* systematically studying what kind of network designs result from purposefully modifying either the objective function or the constraints of the constrained optimization problem at hand. Such forward-engineering efforts would be invaluable for answering questions such as why network environments such as a datacenter serving an application- or service-specific type of traffic and being operated by a single entity that decides what portion of that traffic traverses each link may not see self-similarity in its traffic [5], what other properties such environments may exhibit [43], and why futuristic scenarios where human end users are largely replaced by cyborgs that can easily process information at Mbps-Gbps speeds may require robust designs that result as solutions to very different constrain optimization problems compared to those formulated in [47].

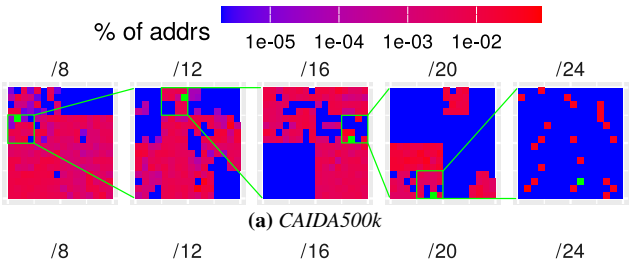
3 A recent invariant: Multifractal scaling of observed addresses

Unlike the self-similar invariant of Internet traffic's temporal dimension which has received significant attention from the research community, establishing the multifractal invariant of Internet traffic's address space dimension requires fundamentally new approaches described in this section.

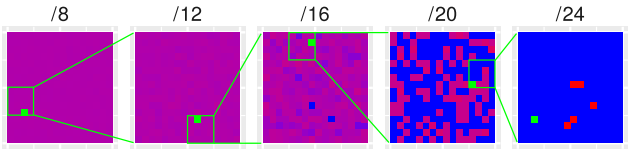
3.1 A picture is worth a thousand words

To study aspects of the spatial structure of observed IP addresses in measured Internet traffic, we consider datasets such as CAIDA500k that was obtained by using the first 500,000 unique source IPv4 addresses observed in the CAIDA-dir-A trace [9].[1] To develop intuition about how sets of such observed addresses populate the IP address space, we leverage the geometric interpretation of observed sets of IPv4 addresses either as point sets in the discrete 1D interval $[0, 2^{32}-1)$ or in the discrete 2D square $[0, 2^{16}-1) \times [0, 2^{16}-1)$.[2] In Figure 1, we use the well-known Hilbert curve-based visualization method that maps the 1D IPv4 address space into a 2D square-image [2, 1, 3] and has the property that prefixes always appear as squares next to their siblings and within their parents (under the usual CIDR interpretation of bit-wise address prefixes).

In particular, the two rows in Figure 1 show this visualization for the dataset CAIDA500k (Figure 1a) and for a set of 500k distinct addresses sampled from a uniform distribution (Uniform500k, Figure 1b). The coloring of each square indicates the relative number of actual IP addresses (*i.e.,* /32 prefixes) observed in the corresponding prefix (e.g., red (blue) prefixes contain one or more (no) observed IP addresses). For each row, the first visual shows the IPv4 space at /8 granulairy, and each subsequent visual "zooms-in" on the marked prefix in the previous visual at longer prefix length until individual source addresses (in a particular /24) manifest in the last visual of a row.



(a) *CAIDA500k*



(b) *Uniform500k.*

Figure 1: *Illustration contrasting the structure of IP addresses observed in real-world Internet traffic (a) and synthetic IP addresses from a uniform distribution (b).*

The stark visual difference apparent between Figure 1a and Figure 1b illustrates the complex intrinsic "cluster-within-cluster" structure of real-world IP addresses, a telltale sign of multifractal scaling identified in previous work [24, 25, 7, 29]. In the case of CAIDA500k (Figure 1a), the zoom-in's for the intermediate stages look qualitatively similar (with the exception of artifacts in the first and last visuals due to known reserved blocks of IPs and discreteness effects, respectively) and exhibit visually apparent cluster-within-

[1] We experimented with datasets of different sizes and assembled over different periods and observed no qualitative differences.

[2] A similar exercise can be conducted for IPv6.

cluster behavior with no "typical" cluster size (*e.g.,* in terms of their area or their number of addresses as indicated by color). In the case of Uniform500k (Figure 1b), we also observe qualitatively similar-looking zoom-ins across the different visuals, but the appearance of the zoom-ins is noticeably less "intermittent"—the cluster-within-cluster behavior is more regular and allows for discerning a high degree of uniformity (in terms of color) and homogeneity (with respect to cluster size).

The analysis in [29] uses newly developed robust statistical techniques to rigorously show that this pictoral evidence is indeed consistent with multifractal scaling and holds across a wide range of different past and present real-world Internet traffic traces.

Given this ubiquitous presence of multifractal scaling of observed addresses in measured Internet traffic, we next seek to explain the underlying processes through which such multifractal scaling comes about in these observations. We advance the hypothesis that observed multifractal scaling in measured Internet traffic arises through the combination of a conservative cascade process (initially proposed in [7, 37]) that captures how IP address spaces are divided up and allocated (§ 3.2 - 3.3) and a process that accounts for the uncertainty of encountering allocations associated with different types of organizations or Internet stakeholders depending on where in the Internet traffic is measured (§ 3.4).

3.2 The baseline conservative cascade

A conservative cascade is a random cascade model where at each recursive step, a given "mass" (defined as number of observed distinct addresses) associated with parent prefixes is divided between child sub-prefixes. Under the CIDR interpretation of IP address "prefixes", this cascade forms a binary tree with *(i)* the "mass" associated with each non-leaf node corresponding to the number of IP addresses observed in each particular prefix, *(ii)* leaves with unit mass corresponding to observed IP addresses (i.e., /32s), and *(iii)* the number of recursive steps taken as prefix length or *level* in the tree. The splitting of mass between parent and child nodes is governed by a random variable denoted W and called the cascade's generator, where W has a particular dependence structure that forces "mass" to be conserved at each step of the cascade.

To formalize this view of the IP address space, consider the bit-level representation of a $/l$ address prefix $x = \langle x_1, x_2, \ldots, x_l \rangle$ where $x_i \in \{0,1\}$ for $1 \leq i \leq l$. The **conservative cascade model** with generator W (where W takes values in $[0,1)$, has mean 1/2 and is symmetric about its mean) is a set of random variables $\{W_{l,j} : l > 0 \text{ and } 1 \leq j \leq 2^l\}$ where *(i)* the $W_{l,j}$'s between different levels l are independent and identically distributed, and $W_{l,j}$ determines how much "mass" is split to the j-th child prefix at level l, and *(ii)* the $W_{l,j}$'s at level l conserve "mass", that is they have a dependence structure given by $W_{l,j+1} = 1 - W_{l,j}$ for odd j's[3]. If the total "mass" is given by Z, then (by independence of the $W_{l,j}$'s between different levels l) the measure or "mass" of a given x at level l is

$$\mu_l(x) = Z \cdot \prod_{1 \leq i \leq l} W_{i,j_i}.$$

where $j_i = 1 + \sum_{1 \leq k \leq i} x_k 2^{i-k}$ specifies the prefixes at levels $i \leq l$ that contain x (*i.e.,* the path from x to the root of the binary tree). To illustrate, Figure 2 shows a simplified view of the first four levels and the "mass" of a particular /4 prefix.

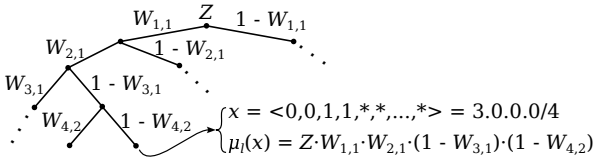


Figure 2: *Example of four levels of a simple conservative cascade showing the "mass" of $x = 3.0.0.0/4$ as a function of the cascade generator W.*

In the mathematical literature, conservative cascades are usually treated in the limit as $l \to \infty$ where the "mass" associated with each branch is a real number and the tree structure shown in Figure 2 is understood as representing dyadic partitions of an interval such as the unit interval $[0,1)$ on the real line [17, 36]. Under certain technical conditions, these limiting objects can be shown to define multifractals (more precisely, multifractal measures) that can be described in terms of well-defined theoretical properties such as the fractal dimension of their support [31, 33] or their multifractal spectrum. However, as we will see in § 3.5, the finite discreteness of IP addresses requires several non-trivial modifications to this traditional view, including the need to focus on "pre-limit" behavior and work with finite approximations of objects that exist and are defined as mathematical limits.

3.3 Address allocation and the conservative cascade

Despite the apparent correspondence between the CIDR prefix organization of IP addresses and a baseline conservative cascade model, to leverage this model to explain the processes behind observed multifractal scaling, we must connect the model to real-world Internet-related processes. We make this connection through the following two observations: **Observation 1** (§ 3.3.1): The hierarchical organization of networks (reflected in address allocations) leads to a non-trivial number of explicit and implicit cascade levels (more than three, less than 32 for IPv4 or 128 for IPv6). **Observation 2** (§ 3.3.2): The lack of distinctive prefix lengths of blocks of allocated addresses requires the bit-by-bit CIDR-based cascade construction.

[3]Note that because of the properties of W, $W_{l,j+1} = 1 - W_{l,j}$ are identically distributed as W.

Datasets and preprocessing. Although we cannot directly observe the mechanisms and policies for all IP address allocation decisions (*e.g.,* for private corporate networks), several data sources capture the outcomes of these decisions and, hence, serve as a proxy for understanding how allocation policies carve up the IP address spaces. In particular, IP address allocation decisions (*i.e.,* which ranges of addresses can be used by which organizations) are kept in public records by the Internet Assigned Numbers Authority (IANA) as a list of global allocation decisions [23, 22], and Regional Internet Registries (RIRs), as "WHOIS" databases. We focus on RIR allocation records obtained as bulk WHOIS data from four of the five RIRs[4] because these records cover the "pre-limit" range of prefixes whereas IANA focuses on shorter prefix lengths (*e.g.,* /8 of IPv4). Overall, our bulk WHOIS dataset contains ∼8.7M IPv4 and ∼1.3M IPv6 network records which we filter to avoid irrelevant records (*e.g.,* placeholders for the global IP address space or for ranges not managed by each particular RIR). We also modify a small number (∼0.7%) of records that describe address ranges not aligned with prefix boundaries by filling them in with a list of the largest possible covering prefixes.

3.3.1 Observation 1: Allocations form a non-trivial cascade process

Allocation[5] of the IP address space has come to be understood through a "three-level story" as illustrated in Figure 3. While this three-level story implies a trivial conservative cascade (whose simplicity is insufficient to explain the complexity of observed clustering), we observe that allocation records actually exhibit a much richer hierarchical structure. This structure has both *explicit* and *implicit* manifestation in the arrangement of allocation records in the address space and implies there are **∼2-5 (∼3-7) additional "hidden" levels for IPv4 (IPv6)** that augment the three-level story.

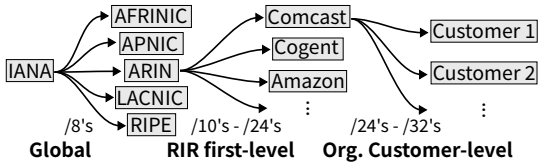


Figure 3: *High-level "cascade" process of Internet address allocations.*

The explicit cascade process in WHOIS allocation records. We quantify the number of explicit cascade steps in the RIR-level address allocation process by constructing the *prefix-inclusion tree*. Each WHOIS network record is a node in this tree, and if a record is a direct sub-prefix of another record (*i.e.,* with no other sub-prefixes in between), then it is that record's child. To measure the number of recursive steps, we leverage the standard definition of tree depth (*i.e.,* a node's depth is the number of edges from that node to the root of the tree—in this case, the top-most network record—and directly corresponds to the number of cascade steps required to reach that node). Figure 4 shows that depth of nodes in the prefix-inclusion trees is significantly long-tailed with median at 1 (2), 95-th percentile at 2 (4) and maximum at 9 (15) for IPv4 (IPv6).[6] Manual inspection of samples of 50 records at depth less than two and two or greater confirmed intuitions about how different depths reflect different types of organizations (*e.g.,* deeper networks tend to be smaller regional entities like hospitals or local businesses).

The implicit cascade process in WHOIS allocation records. Particular address regions assigned to "end-user" organizations in WHOIS databases also exhibit clear clustering along prefix boundaries that reveal evidence of hierarchic structure in organization-internal allocation policies, thus adding additional recursive steps to the conservative cascade. To estimate the number of steps added, we define an *approximate maximum aggregation* method on a prefix-inclusion tree by identifying prefixes where the percentage of address space covered by descendant WHOIS records falls below a fixed threshold (which we set at 51% by default based on observation of our dataset) on the path down the tree.

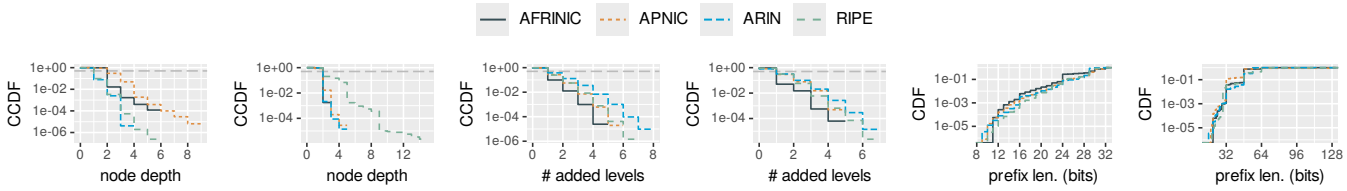
To quantify the number of cascade steps potentially contributed by approximate max aggregate prefixes over the entire address space, we compute how many approximate max aggregate prefixes are generated on the path between each WHOIS leaf record and its parent record. Figure 5 shows that these distributions are long-tailed for both IP versions with median at 1, 95-th percentile between 2 and 3, and maximum between 5 and 7 across different RIRs. Except for AFRINIC (due to prevalent covering of internal WHOIS records), in the remaining RIRs between 25% and 37% of leaves have two or more approximate max aggregate prefixes between them and their parent WHOIS record (again for both IP versions) indicating the presence of substantial implicit hierarchic organization of leaf records.

Summing the steps. Combining the medians and 95-th percentiles for the explicit cascade (1-2 (2-4) for IPv4 (IPv6)) and the implicit cascade (1-3 for both IPv4 and IPv6) yields a total of 2-5 (3-7) additional recursive cascade steps for IPv4 (IPv6), exposing a much richer cascade construction (compared to the "three-level" story) in real-world allocation practice.

[4] Similar to [21], we attempted to obtain the LACNIC bulk WHOIS dataset, but failed due to administrative impasse.

[5] Note that *allocations* are normally understood to refer to intermediate decisions in this process (*i.e.,* address ranges to be further divided by a down-stream entity) whereas *assignments* are terminal decisions (*i.e.,* to be used for the infrastructure of a particular end-user entity) [34, 40].

[6] Note Figure 4's log-scale y-axis and dashed horizontal line at $y = 0.5$ (median).



(a) *IPv4* **(b)** *IPv6*

Figure 4: *Distribution of depth in per-RIR prefix trees over all records.*

(a) *IPv4* **(b)** *IPv6*

Figure 5: *# levels added by approximate max aggregate prefixes.*

(a) *IPv4* **(b)** *IPv6*

Figure 6: *Distribution of the prefix length over all bulk WHOIS records.*

3.3.2 Observation 2: Allocations have no inherent prefix lengths

Although **Observation 1** establishes that address allocations form a deeper cascade with up to eight total levels, this still falls short of the cascade depth assumed by the CIDR-based model that splits mass at each prefix length (from /0 to /31 for IPv4 or /0 to /128 for IPv6). In particular, if there were "typical" prefix lengths for allocation records, it may be the case that the internal cascade process actually splits mass among more than two children at these eight or so distinct prefix lengths. This would require a more complex k-ary cascade model, for *e.g.,* involving additional constraints on the correlation structure of the $W_{l,j}$'s.

We eliminate this possibility by showing the distribution of prefix lengths over all records in our bulk WHOIS datasets in Figure 6. This result indicates that the prefix lengths of WHOIS records are spread out evenly between /12 and /28 for IPv4 and /18 and /48 for IPv6.[7] Though there are minor modes apparent across all RIRs (*e.g.,* for IPv4 at /16 and /24, perhaps echos of the old class-full address organization), the size of these modes relative to the overall number of records is relatively small (*e.g.,* IPv4 /24 records account for ∼21% of AFRINIC and between 5-6% of the other RIRs). The distribution of prefix length for the "approximate maximum aggregate" prefixes described in § 3.3.1 show a similar pattern (not shown due to limited space) with no apparent "typical" prefix length.

From "steps" to "bits". The empirical observation that WHOIS allocation records have no "typical" prefix lengths implies that a bit-by-bit conservative cascade model (*i.e.,* taking each address bit as a level in the cascade) *can* capture the observed behavior without the need for further constraining the $W_{l,j}$'s or modifying the cascade structure. Rather, the "missing" recursive steps of the cascade between particular allocation records can be understood as cases where the $W_{l,j}$ sends *all* of the address mass to the left ($W_{l,j} = 1$) or right ($W_{l,j} = 0$) child causing all addresses to fall in the same child prefixes.

While the above observations establish a process-level correspondence between the conservative cascade model and how addresses are actually *allocated*, they do not provide an explanation for *why* the conservative cascade is a meaningful model for characterizing particular *observed* sets of addresses in measured Internet traffic. This is because we have thus far not considered the weight of each prefix as measured by the number of *distinct* IP addresses observed in that prefix at the vantage point in question. In particular, it still could be the case that address "weight" is split evenly between children at each level of the cascade (*i.e., W* is a constant 1/2).

3.4 An evocative mathematical model

To demonstrate that the conservative cascade of IP address allocation inevitably features imperfect balance between children (*i.e.,* the support of the distribution of W does not consist of the single point 1/2)—and hence the inevitability of multifractal scaling of observed IP addresses—we show through a simple first-principles argument that particular vantage points will observe different allocations with different weight. This argument unfolds through the following three steps.

(i) Consider a particular arbitrary network **X**. **X** is assigned addresses and uses them for different purposes. Given the common client-server architecture, we can group these purposes as being client-oriented (associated with initiating connections) or server-oriented (associated with accepting connections from clients).

(ii) Different networks have different balances of clients and servers. Since the cascade process of allocation lacks a "preferred" prefix length of allocations (*i.e.,* there exists no "typical" size of prefix allocation), the "client weight" (respectively "server weight") is clustered across the address space, thus implying that there is not a uniform or homogeneous mix of clients and servers w.r.t. the address space.

(iii) When we observe traffic at a particular vantage point, whether or not we see addresses assigned to **X** depends on **X**'s client/server balance and the position of the vantage point within the Internet. If the vantage point is near a server-heavy region, it is more likely to see **X** if **X** is client-heavy. On the other hand, if the vantage point is near a client-heavy region, it is more likely to see **X** if **X** is server-heavy. In this way, the client-server weight of the address space translates

[7] Note that in Figure 6, instead of focusing on the distributional tail, the log-scale y-axis normalizes for the exponentially increasing total number of potential records at each level.

directly into the "number of addresses per prefix" weight considered in our conservative cascade model (with the position of the vantage point balancing between client weight and server weight).

For example, consider the simplified scenario shown in Figure 7 where three networks, **X**, **Y**, and **Z** use the same number of addresses, but have different numbers of servers and clients exchanging traffic through the Internet. Suppose also that each client independently selects one of the five servers uniformly at random and communicates only with that server. The probability[8] that traffic is exchanged between addresses in two different networks $i, j \in \{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$ is given by $P(i \leftrightarrow j) = (1-(1-s_j/\sum_k s_k)^{c_i}) + (1-(1-s_i/\sum_k s_k)^{c_j}) - (1-(1-s_j/\sum_k s_k)^{c_i}) \cdot (1-(1-s_i/\sum_k s_k)^{c_j})$ where s_k is the number of servers and c_k is the number of clients in network $k \in \{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$. By plugging in the numbers of clients and servers shown in Figure 7, it can be seen that because of different server-client composition, the probability of seeing an address from network **Y** is high ($\sim$0.95) at network **X** and significantly lower ($\sim$0.74) at network **Z**.

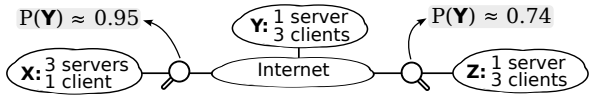


Figure 7: *Example of how client-server weighting translates to different probabilities of seeing addresses from a particular network at different vantage points.*

3.5 How well does the model fit the data?

To quantify how well the imperfectly balanced conservative cascade model described above fits real-world observed sets of IP addresses, we develop an approach to fitting the distribution of the cascade generator W to observed addresses.

The basic approach to such fitting involves computing first the empirical "weights" given by $w_{l,j} = \mu_l(x_1,\ldots,x_j)/\mu_{l-1}(x_1,\ldots,x_{j-1})$ and then estimating the distribution of W by considering the $w_{l,j}$'s to be i.i.d. observations of W.

To illustrate the challenges of this approach, we compute the $w_{l,j}$'s for the CAIDA500k dataset and show their distribution for three example prefix lengths (i.e., $l = 8, 16, 24$) in Figure 8. Although the $w_{l,j}$'s are clearly symmetric (*i.e.,* mean 1/2), they also have two features that complicate the fitting task. First, the shape of the distribution depends strongly on l and second, there are strong modes that begin to appear at 0.0 and 1.0 for longer prefix lengths.

Limitations of prior approaches. Prior approaches to fitting conservative cascades to observed sets of IP addresses [7, 37] used a beta distribution to model W and filtered the $w_{l,j}$'s to only consider a limited range of prefix lengths l (in particular, $0 \le l \le 8$). Since beta distributions

[8] Given by simple composition of the complement and union probability rules and independence between clients.

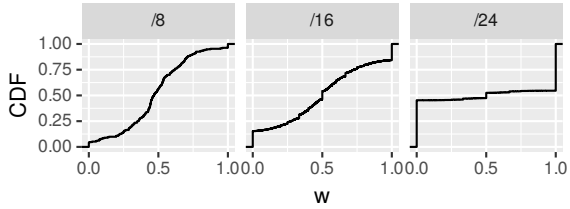
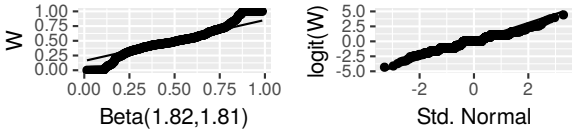


Figure 8: *Distribution of the raw values of $w_{l,j}$ for the CAIDA500k dataset at various prefix lengths.*

can be defined to take values in the unit interval $[0,1]$ and be symmetric around $1/2$, they are, in theory, well suited for modeling the "weights" $w_{l,j}$'s. In practice, however, fitting beta distributions while ensuring symmetry and obtaining confidence intervals can be challenging. Moreover, although removing all prefixes longer than $l = 8$ avoids quantization effects when the number of observed addresses per prefix becomes small, it omits a significant range of scaling behavior and is inconsistent with the notion of "pre-limit behavior" required when applying conservative cascades to IP addresses (§ 3.2).

To illustrate, we apply the method proposed in [7] to the CAIDA500k, fitting a beta distribution to $w_{l,j}$ for $0 \le l \le 8$ and show a Q-Q plot[9] comparing the $w_{l,j}$ (over $0 \le l \le 8$) to the theoretic quantiles of the fit beta distribution in Figure 9a. The significant deviations from linearity at small and large quantiles indicate that trying to fit a beta distribution to these weights does not capture key distributional features.



(a) *Beta,* $0 \le l \le 8$ **(b)** *Our method*

Figure 9: *Q-Q plots comparing the fit of the beta distribution method of prior work [37, 7] and our pre-processing + Logit-normal method.*

Our method. We address these limitation of prior approaches by modeling W as symmetric Logit-normal which admits high-confidence fitting of a single parameter σ and by preprocessing the $w_{l,j}$'s so that scaling information over a wider range of l can be used.

The symmetric Logit-normal distribution has a similar shape as the Beta distribution, but affords a simplified estimation of its single parameter σ through the correspondence with the Normal distribution. In particular, if W follows a Logit-normal distribution, then applying the logit function $(\log(W/(1-W)))$ yields a Normal distribution which, in turn, allows for applications of the standard, Gaussian-based parameter estimation theory. Moreover, the single parameter σ controls the strength of the distribution's mean: smaller values of σ produce a narrow distribution around the mean and as σ increases, the distribution spreads out in $(0,1)$.

[9] Q-Q plots are a standard non-parametric tool for comparing two distributions and produce a straight line when the distributions match.

Our preprocessing approach leverages the following three key ideas:

- We ignore prefixes with a single address since they force $w_{l,j} = 1$ or 0 and hence do not contribute information about how "mass" is "split".
- We correct for "rounding" effects in cases where all address "mass" moves exclusively to the left or right child by replacing $w_{l,j} = 0$ with $1/2n$ and $w_{l,j} = 1$ with $1 - 1/2n$ where n is the number of addresses in the parent prefix.
- We ignore prefixes that cover reserved address space in order to avoid artificial inflation of the variance of W. In cases where the input addresses are anonymized (*e.g.,* with prefix-preserving methods [18]), we approximate the reserved regions in the anonymized space as any empty /8.

Figure 9b shows a Q-Q plot comparing the pre-processed and logit-transformed weights to the Standard Normal distribution. The nearly linear shape of the plot is indicative of a significantly improved fit compared to Figure 9a.

3.6 Key implications

The conservative cascade model that explains why observed sets of IP addresses exhibit multifractal scaling has several important implications.

- Even if *every single address* in the address space is actively allocated, the traffic traversing particular Internet vantage points will still exhibit multifractal scaling similar to that observed in Figure 1a. This follows because the allocation cascade described in § 3.3 has no distinct prefix lengths, and the observation process described in § 3.4 implies that the weights $W_{l,j}$ are non-deterministic and dependent on the location of the vantage point in the Internet graph relative to different types of hosts.
- Because we observe similar allocation policy and communication patterns for both IPv4 and IPv6, the conservative cascade model applies equally to both cases regardless of their relative densities (*i.e.,* extreme density of IPv4 address space utilization vs. extreme sparseness of IPv6 address space utilization).
- The dependence of the cascade weights on vantage point location (§ 3.4) implies that the structure of IP addresses captured by the conservative cascade depends directly on the vantage point itself and hence can be used as a digest or "fingerprint" of communication patterns at that vantage point.
- The method of fitting the distribution of the generator W (§ 3.5) implies an efficient approach to generate synthetic sets of IP addresses that closely mimic those encountered in measured Internet traffic, enabling verification and evaluation of existing and future technologies [29, 37].

4 There is more to multifractality ...

Section 3 demonstrates how mathematical modeling efforts that respect the highly engineered nature of networked systems such as the Internet combine with empirical studies bolstered by rigorous statistical inference to arrive at an empirically validated physical explanation for why sets of observed IP addresses in measured Internet traffic exhibit, perforce, multifractal scaling behavior. In this section, we ask (and answer) a yet more fundamental question: Is this inevitable multifractal scaling behavior the visible manifestation of some underlying design efforts, and if so, what are these efforts trying to achieve with respect to how IP addresses are allocated and used in practice?

4.1 New Findings

The IP address allocation processes analyzed in Section 3 resulted from decades of applied experience and fundamental insights into address allocation procedure design. These efforts trace back to the pioneering work of John Postel and resulted, among other achievements, in the inception of the Internet Assigned Numbers Authority (IANA) that is responsible for the global coordination of the Internet protocol addressing systems [20]. To understand *why* conservative cascade-like processes appear to underlie real-world address allocations, we apply the HOT perspective to distill the Internet address allocation problem in terms of several networking objectives and constraints and propose a generic allocation procedure that directly seeks to satisfy these objectives and constraints in the face of uncertainty in the environment. Based on empirical simulation, we show that our HOT-inspired procedure effectively balances the competing goals of maximizing address space utilization while minimizing fragmentation and yields allocation size distributions that exhibit heavy-tailed characteristics, irrespective of the precise nature of how uncertainty is quantified (*e.g.,* the distribution of addresses requested by the different Internet stakeholders can be light-tailed or heavy-tailed).

4.1.1 Internet address allocation: A HOT perspective

To study the fundamental principles at play in Internet address allocation processes, we apply the HOT theme and consider a simplified constrained optimization problem formulation that consists of *(i)* a finite set of addresses to allocate,

(ii) a finite set of "clients" that request addresses for their infrastructure, *(iii)* a procedure to determine which particular address to allocate for each client request, and *(iv)* a model for quantifying the uncertainty in the environment (i.e., address space requests by different organization.) Without loss of generality, we model addresses as fixed bit-width integers and consider finite sequences of allocation requests or allocation "scenarios". Moreover, we quantify uncertainty by modeling the number of addresses requested by each Internet stakeholder as following a given distribution (e.g., exponential) and randomly interleaving requests from all clients. This general formulation allows us to simulate and empirically measure the performance of different address allocation design procedures that process the stream of requests from all clients, responding with an address allocation for each request (or responding that no more requests can be processed) before processing the next request (until all possible addresses have been allocated or the stream of requests terminates). Once allocated, addresses cannot be revoked or reassigned.

Although this formulation drastically simplifies many real-world aspects of IP address allocation, we hold that it represents several key generic constraints that generalize a wide range of real-world situations. First, considering a finite set of addresses captures the fact that many network protocols leverage fixed-size header fields to ensure efficient and deterministic per-packet processing overheads. Second, requiring the allocation procedure to make decisions for each client request without future knowledge of how many addresses that client will eventually need captures the inherent uncertainty in Internet address allocation (*e.g.,* the popularity, and hence scale, of a particular organization cannot be known definitively when the organization first requests addresses). Finally, the simplistic formulation proposed here can easily be extended in future work to explore specific real-world phenomena. For example, churn in the set of client organizations could be modeled by imposing a more detailed request shuffling scheme that bounds client requests to particular positions in the sequence and returning blocks of addresses to the allocation process when a client is considered to no longer exist.

Next, we map the key goals of Internet allocation—maintaining stability while scaling to large numbers of address requests and requesting clients—to this simplified HOT-inspired problem formulation. On the one hand, the goal of stability is inherently captured by assuming address allocations last the entire scenario duration. In particular, this prevents an allocation procedure that reacts to new client requests by reorganizing previously made allocations. On the other hand, the goal of scaling to large numbers of address requests and requesting clients requires further elaboration below.

We posit that, rather than the number of addresses that can be allocated or the number of clients that can receive allocations, a key scalability objective of address allocation is to minimize the sizes of routing tables that result from a given allocation. Because our problem formulation does not consider the structure of the network graph, we use address space fragmentation, measured as the number of disjoint allocations per client, as a proxy for estimating the impact of allocations on routing tables sizes. To understand why minimizing routing table size and fragmentation are critical for enabling scalability, consider a trivial allocation procedure that enumerates all possible addresses (*e.g.,* in increasing order) and responds to each allocation request with the next available address in this enumeration. Although this procedure can effectively allocate all possible addresses to any number of requesting clients, the resulting address allocation structure requires routing tables with one entry for each individual address, leading to extremely inefficient forwarding lookup operations at each intermediate network device (*e.g.,* router, switch, etc.). Similarly, consider a relaxation of this trivial procedure that instead partitions the address space into equal-sized blocks and assigns each client organization a dedicated block. While the total number of blocks used realizes an effective upper-bound on the size of routing tables (assuming for simplicity any multi-homed clients are represented by two separate logical client entities), the number of clients that can be serviced is also limited by the same upper-bound and, in cases where a client requests fewer addresses than the block size, the maximum number of addresses that can be allocated may be less than the total number of possible addresses. In the next section, we explore how a primitive hierarchic cascade-like approach can simultaneously address both of these distinct scalability requirements.

4.1.2 Basic cascade-based allocation design procedure

Our key observation is that cascade structure, in conjunction with the abstract division of address space "mass" along prefix-tree boundaries, provides the key ingredients for a generic design procedure that addresses both the goal of maximizing address space utilization and minimizing routing table size, viz., fragmentation. Moreover, we observe that in the face of uncertainty (*i.e.,* number of addresses requested by the different clients), this procedure tends to produce address allocations where the sizes of individual allocations are heavy-tailed even when the client size distribution is not heavy-tailed, implying that the unbalanced nature of IP address allocations is an inherent result of approaches to optimize both goals in the face of uncertainty.

Our procedure interprets fixed-width addresses as a tree of prefixes and associates each client with one "active" prefix at a time. Each request for a new address is handled by the following steps.

- The first requesting client is associated with the global prefix (*i.e.,* 0/0) and receives the address 0.

- Thereafter, if the requesting client is already associated with a prefix that has available addresses, the next sequential address in that prefix is returned.
- If the prefix is full, the list of all prefixes is searched for the prefix with most available addresses, this prefix is split in half, and the requesting client's active prefix is updated to be the right child (*i.e.,* higher-numbered) half while the original prefix's client's active prefix is shrunk to be the left child (*i.e.,* lower-numbered) half. If the prefix with most available addresses is over half-full already, its right half is split recursively until an empty prefix is obtained on the right.
- If the requesting client is not yet associated with a prefix, the same procedure to find and split the prefix with most available addresses is also used to start a new prefix for the client.

Intuitively, this design procedure views the address space as a "mass" which is recursively split as new clients request addresses or previous clients exhaust the prefix they are actively allocating in.

We evaluate this approach by simulating the procedure described above in the constrained optimization problem formulation described in § 4.1.1 with 16-bit addresses and compare against simple heurisitics such as sequential allocation and fixed block-size allocation procedures. To measure how each procedure performs as the number of address requests increases, we use a fixed number of clients (*i.e.,* 1000) and adjust the mean addresses per client to achieve different levels of address space utilization. For example, at utilization 0.5, half of the addresses are requested on average. We report summaries (*e.g.,* box plots) of aggregate results over 100 independent runs of each experiment.

Figure 10 shows fragmentation of the resulting address allocation decisions, measured as the number of distinct allocations per client, as a function of address space utilization for each procedure. As expected, the sequential procedure produces highly fragmented allocations with each client requiring up to 60 distinct allocations at 90% utilization. The fixed block-size procedure with two addresses per block requires half as many allocations, but only scales to 50% utilization after which it fails to allocate all requested addresses. Our proposed cascade-based design procedure solves both challenges, successfully allocating all requested addresses while keeping fragmentation below ∼1.9 allocations per client at up to 90% utilization) and requiring only ∼3 allocations per client at 100% utilization.

Finally, we investigate how our proposed cascade procedure relates to the empirical observations discussed in Section 3. In particular, Figure 11 shows the distribution of allocation depth (*i.e.,* the number of times each final allocation was split in half during the cascade procedure). Intuitively, increasing utilization requires additional splitting of prefixes

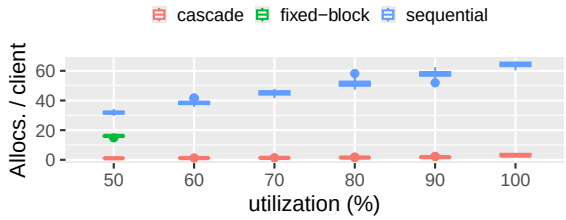


Figure 10: *Impact of address space utilization on fragmentation as measured by the number of distinct allocations per client.*

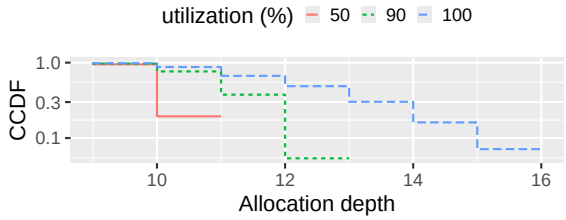


Figure 11: *Distribution of the number of levels (i.e., how many times each prefix was split in half) in allocations made by the cascade procedure for three example utilizations.*

leading to a deeper allocation structure, up to /13 at 90% utilization and all the way to /16 at 100% utilization. This pattern mirrors real-world observed depth of the WHOIS prefix inclusion tree shown in Figures 4 and 5 where RIRs with more allocations tend to have longer-tailed allocation depth distributions (*e.g.,* RIPE has ∼3× more IPv6 allocations compared to the next largest).

4.1.3 Key implications

How does this cascade-based model address the goal of scalability? Hierarchical designs are a natural engineering solution to scalability. In the case of Internet allocations, the close binding of a hierarchical allocation procedure (*e.g.,* the multi-organization hierarchies described in § 3) and a hierarchical organization of the underlying address space (*i.e.,* the CIDR prefix tree) enables maximizing the number of addresses that can be allocated (prefixes can be divided all the way up to individual addresses) and minimizing fragmentation (by avoiding divisions when they are not called for by the concrete address requests and balancing growth of divisions across the address space).

Why is the resulting cascade not perfectly balanced then? Both real-world allocations and our simplified HOT model exhibit imperfectly balanced cascades with long-tailed distributions of prefix depth. Our hypothesis is that this imbalance is a result of not knowing the number of addresses requested by the different Internet stakeholders. In the face of this uncertainty, for most allocations, the heuristic of splitting the prefix with maximum available addresses yields a well-balanced structure. However, in cases where the original client of the split prefix ends up growing to a much larger number of addresses and thus entails deeper splitting elsewhere in the tree, this heuristic is non-optimal.

4.2 Unfinished business

Our HOT-inspired initial generic cascade-based design procedure for Internet address allocation suggests that the kind of cascade structures observed in real-world allocation data (described in Section 3) arise as a result of optimization efforts that try to maximize the number of addresses allocated and minimize fragmentation in the face of uncertain demand from client organizations. Put differently, the ubiquitous multifractal scaling behavior of sets of observed IP addresses in measured Internet traffic can thus be recognized as an outward sign of an underlying (and at best only implicitly pursued) design effort for Internet address allocation that is made explicit by our proposed cascade-based procedure that manifests as a solution of a concrete constrained optimization problem. Below, we briefly illustrate with examples how this new understanding can aid the design of future networks and network applications.

Why is it important to optimize for stability and scalability of IP address allocations? The observation that cascade-based Internet address allocation designs can be seen as engineered efforts to optimize for stability of address allocations and scalability to allocate large numbers of addresses without exploding router table sizes raises deeper questions about the role of addressing in the stability and scalability of the Internet as a whole. Further reverse engineering effort is required to understand the impact of Internet phenomena such as the interplay of address recycling and re-use in cloud settings (and recently also in some ISPs and enterprises [35, 41]) on our notions of stability and on systems like blocklists [39, 44] built on the premise of stable address allocations. Further forward engineering effort is required to leverage models such as those presented here to examine how the drastically increased size of the IPv6 address space may impact future stability and scalability concerns such as router table inflation [12].

But future changes in how the Internet is owned and managed will certainly change everything! Or will they? Permanence of the arguments developed here (linking fundamental address allocation objectives to observed multifractal scaling) through major changes in Internet ownership and management has strong historical support by considering that at a qualitative level, the multifractal scaling behavior observed some 20 years ago—before cloud-based approaches had fully dominated the landscape—is still observed to this day. Insofar as the same objectives of maintaining stability and scalability and the same challenges of dealing with uncertainty in the number of addresses required by any particular client remain present in some form, the same multifractal scaling behavior can be expected to persist. This does not mean that significant changes in the structure of the Internet will have no impact on observed address structure, but that the impact can be expected to be largely quantitative and not qualitative. This, in turn, calls for renewed efforts to discover, optimize, and apply knowledge of multifractal scaling and multifractal analysis tools across the wide range of applications that stand to benefit from such a principled understanding of the structure of observed IP addresses (e.g., see [29] and [30]).

There are no signs of multifractal scaling in my network! In settings where the required elements for our HOT-based approach to obtaining deliberate robust designs are not present, observed addresses may not exhibit multifractal scaling behavior. For example, in data center networks where addresses are allocated by a single organization with complete knowledge of the number of hosts and their connectivity, there is no (or very minimal) uncertainty of how many addresses are required in each logical division of the network (*e.g.,* each prefix). Hence, in these and similar scenarios, in the absence of uncertainty, optimal allocations can easily be determined (*e.g.,* [16]) and can be expected to result in trivial observed address structure. At the same time, based on our new understanding, the absence or presence of multifractal scaling in the observed address structure in a given environment allows us to hypothesize what design efforts with resect to allocating addresses are at work in the given environment.

5 Summary

Originally pioneered and subsequently shown to have real-world applications by B. Mandelbrot [28], self-similar and multifractal scaling behaviors have featured prominently in the networking literature where they have been identified as invariants of measured Internet traffic and observed address structure, respectively. Encountering these Internet invariants motivated us to study in this paper fundamental problems that ask whether these invariants can be recognized as outward signs of some underlying (and at best only implicitly pursued) robust designs, and if so, what is the concrete nature of the problems that are being solved by these robust designs. By pursuing HOT-inspired reverse-engineering efforts, we arrive at surprisingly specific answers. For one, self-similar traffic is the hallmark of robust designs that ensure that the dominant applications and services on the Internet organize information for human consumption. At the same time, multifractal observed address structure is a visible manifestation of underlying robust design efforts that explicitly optimize for stability and scalability in the face of uncertainty in the number of addresses required by each particular Internet stakeholder.

We complement these advances in our fundamental understanding of Internet invariants with proposed forward-engineering efforts that shed light on what the absence or presence of a given Internet invariant says about the design efforts being pursued for a given network environment or show under what circumstances the pertinent design efforts we identified for today's Internet have to be revised for tomorrow's Internet, and if so, how. While a number of our

reported findings and stated hypotheses require more work to further substantiate them or prove them, our hope with this work is that it re-invigorates research efforts that demonstrate the power of reverse-engineering as well as forward-engineering for establishing scientifically sound intellectual foundations and practical principles for designing future networked systems.

References

[1] 62 days + almost 3 billion pings + new visualization scheme = the first internet census since 1982. https://tinyurl.com/34t4dmn8. Accessed: 2025-05.

[2] IPv4 census map - CAIDA. https://www.caida.org/archive/id-consumption/census-map/. Accessed: 2025-05.

[3] Packed to the brim: Analyzing highly responsive prefixes on the internet. https://hrp-stats.github.io/. Accessed: 2025-05.

[4] Martin F Arlitt and Carey L Williamson. Web server workload characterization: The search for invariants. *ACM SIGMETRICS Performance Evaluation Review*, 24(1):126–137, 1996.

[5] Berk Atikoglu, Yuehai Xu, Eitan Frachtenberg, Song Jiang, and Mike Paleczny. Workload analysis of a large-scale key-value store. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, pages 53–64, 2012.

[6] Per Bak. *How nature works: the science of self-organized criticality*. Springer Science & Business Media, 2013.

[7] Paul Barford, Rob Nowak, Rebecca Willett, and Vinod Yegneswaran. Toward a model for source addresses of internet background radiation. In *Proc. of the Passive and Active Measurement Conference*. Citeseer, 2006.

[8] Theophilus Benson, Aditya Akella, and David A Maltz. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 267–280, 2010.

[9] CAIDA. The CAIDA UCSD anonymized Internet traces dataset - 2019. https://www.caida.org/data/monitors/passive-equinix-nyc.xml. Accessed: 2024-02.

[10] Jean M Carlson and John Doyle. Highly optimized tolerance: A mechanism for power laws in designed systems. *Physical review E*, 60(2):1412, 1999.

[11] Jean M Carlson and John Doyle. Complexity and robustness. *Proceedings of the national academy of sciences*, 99(suppl_1):2538–2545, 2002.

[12] Robert Chang, Pradeep Dogga, Andy Fingerhut, Victor Rios, and George Varghese. Scaling IP lookup to large databases using the CRAM lens. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*, pages 127–146, 2025.

[13] Mung Chiang, Steven H Low, A Robert Calderbank, and John C Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, 2007.

[14] Mark E Crovella and Azer Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on networking*, 5(6):835–846, 1997.

[15] John Doyle and Jean M Carlson. Power laws, highly optimized tolerance, and generalized source coding. *Physical Review Letters*, 84(24):5656, 2000.

[16] Jonathon Duerig, Robert Ricci, John Byers, and Jay Lepreau. Automatic ip address assignment on network topologies. *Technical Note FTN200602, University of Utah Flux Group*, 2006.

[17] Carl JG Evertsz and Benoit B Mandelbrot. Multifractal measures. *Chaos and fractals*, 1992:921–953, 1992.

[18] Jinliang Fan, Jun Xu, Mostafa H. Ammar, and Sue Moon. Cryptopan. https://web.archive.org/web/20190430122100/https://www.cc.gatech.edu/computing/Networking/projects/cryptopan/. Accessed: 2024-02.

[19] Sally Floyd and Vern Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on networking*, 9(4):392–403, 2001.

[20] V Fuller and T Li. Rfc 4632: Classless inter-domain routing (cidr): The internet address assignment and aggregation plan, 2006.

[21] Amanda Hsu, Frank Li, and Paul Pearce. Fiat lux: Illuminating ipv6 apportionment with different datasets. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(1):1–24, 2023.

[22] IANA. IPv6 global unicast address assignments. https://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml. Accessed: 2024-01.

[23] IANA. IPv4 address space registry. https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml, 2023. Accessed: 2024-01.

[24] Eddie Kohler, Jinyang Li, Vern Paxson, and Scott Shenker. Observed structure of addresses in ip traffic. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 253–266, 2002.

[25] Eddie Kohler, Jinyang Li, Vern Paxson, and Scott Shenker. Observed structure of addresses in ip traffic. *IEEE/ACM Transactions on Networking*, 14(6):1207–1218, 2006.

[26] Maria Korolov. Ai workloads set to transform enterprise networks. https://www.networkworld.com/article/3963141/ai-workloads-to-transform-enterprise-networks.html, 2025. Accessed: 2025-09.

[27] Will E Leland, Murad S Taqqu, Walter Willinger, and Daniel V Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on networking*, 2(1):1–15, 1994.

[28] Benoit B Mandelbrot. *Fractals and scaling in finance: Discontinuity, concentration, risk*. Springer, 1997.

[29] Chris Misa, Ram Durairajan, Arpit Gupta, Reza Rejaie, and Walter Willinger. The multifractal ip address structure: Physical explanation and implications. *arXiv preprint arXiv:2504.01374*, 2025.

[30] Chris Misa, Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. Leveraging prefix structure to detect volumetric ddos attack signatures with programmable switches. *IEEE Symposium on Security and Privacy (SP) (Oakland)*, 2024.

[31] Mina Ossiander and Edward C Waymire. Statistical estimation for multiplicative cascades. *The Annals of Statistics*, 28(6):1533–1560, 2000.

[32] Vern Paxson and Sally Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Transactions on networking*, 3(3):226–244, 1995.

[33] Sidney Resnick, Gennady Samorodnitsky, Anna Gilbert, and Walter Willinger. Wavelet analysis of conservative cascades. *Bernoulli*, 9(1):97–135, 2003.

[34] RIPE. IPv6 address allocation and assignment policy. https://www.ripe.net/publications/docs/ripe-552/, 2012. Accessed: 2024-10.

[35] Erik Rye, Robert Beverly, and Kimberly C Claffy. Follow the scent: Defeating ipv6 prefix rotation privacy. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 739–752, 2021.

[36] Hadrien Salat, Roberto Murcio, and Elsa Arcaute. Multifractal methodology. *Physica A: Statistical Mechanics and its Applications*, 473:467–487, 2017.

[37] Joel Sommers and John Raffensperger. Efficient and realistic generation of ip addresses. In *4th International ICST Conference on Simulation Tools and Techniques*, 2012.

[38] Gábor Soós, Dániel Ficzere, and Pál Varga. Investigating the network traffic of industry 4.0 applications–methodology and initial results. In *2020 16th International Conference on Network and Service Management (CNSM)*, pages 1–6. IEEE, 2020.

[39] Costa Tsaousis. FireHOL IP lists | IP blacklist | IP blocklists | IP reputation. http://iplists.firehol.org/. Accessed: 2025-01.

[40] Leo Vegoda. Assignments, allocations and temporary transfers. https://ipv4.global/events/assignments/. Accessed: 2024-10.

[41] Liang Wang, Hyojoon Kim, Prateek Mittal, and Jennifer Rexford. Raven: Stateless rapid ip address variation for enterprise networks. *Proceedings on Privacy Enhancing Technologies*, 2023.

[42] Walter Willinger, Murad S Taqqu, Robert Sherman, and Daniel V Wilson. Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on networking*, 5(1):71–86, 1997.

[43] Jackson Woodruff, Andrew W Moore, and Noa Zilberman. Measuring burstiness in data center applications. In *Proceedings of the 2019 Workshop on Buffer Sizing*, pages 1–6, 2019.

[44] www.blocklist.de. Fail2ban reporting service. https://www.blocklist.de/en/index.html. Accessed: 2025-01.

[45] Zihan Yan, Dan Li, Li Chen, Dian Xiong, Kaihui Gao, Yiwei Zhang, Rui Yan, Menglei Zhang, Bochun Zhang, Zhuo Jiang, et al. From atop to zcube: Automated topology optimization pipeline and a highly cost-effective network topology for large model training. In *Proceedings of the ACM SIGCOMM 2025 Conference*, pages 861–881, 2025.

[46] Jieyu Zheng and Markus Meister. The unbearable slowness of being: Why do we live at 10 bits/s? *Neuron*, 113(2):192–204, 2025.

[47] Xiaoyun Zhu, Jie Yu, and John Doyle. Heavy tails, generalized coding, and optimal web layout. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, volume 3, pages 1617–1626. IEEE, 2001.

A Ethics

This work raises no ethical issues beyond the handling of private traces of network traffic. In handling these traces we followed all relevant policies of respective institutions.