Observed Structure of Addresses in IP Traffic

Eddie Kohler, Jinyang Li, *Member, IEEE*, Vern Paxson, *Member, IEEE*, and Scott Shenker, *Fellow, IEEE*

***Abstract*—We investigate the structure of addresses contained in IPv4 traffic—specifically, the structural characteristics of destination IP addresses seen on Internet links, considered as a subset of the address space. These characteristics have implications for algorithms that deal with IP address aggregates, such as routing lookups and aggregate-based congestion control. Several example address structures are well modeled by multifractal Cantor-like sets with two parameters. This model may be useful for simulations where realistic IP addresses are preferred. We also develop concise characterizations of address structures, including *active aggregate counts* and *discriminating prefixes*. Our structural characterizations are stable over short time scales at a given site, and different sites have visibly different characterizations, so that the characterizations make useful "fingerprints" of the traffic seen at a site. Also, changing traffic conditions, such as worm propagation, significantly alter these fingerprints.**

***Index Terms*—Address space, address structures, multifractals, network measurement.**

I. Introduction

THE behavior of individual flows—single connections or streams of packets between the same source and destination—has received extensive analysis for a number of years. However, as the Internet continues to expand in speed and size, the gulf between the behavior of flows and the behavior of large aggregates of flows grows ever wider. Studies of aggregate traffic have focused on questions of behavior at a particular granularity: for example, correlations in packet arrivals seen en masse on a link [1], patterns of backbone traffic when partitioned by directionality, transport protocol, and application [2], [3] or viewed at /8, /16 and /24 prefix granularities [4], or the overall distributions of individual connection characteristics [5], [6]. These studies have made significant progress in understanding the structure of specific types of aggregates. In this paper, however, we focus on how behavior changes *as aggregation increases*. There is clearly a world of difference between an individual TCP connection and a gigabit traffic conglomerate headed from one city to another, but aside from basic statistical multiplexing models, we understand little of how behavior changes as we go from one to another.

We tackle a relatively modest question, one of the simplest conglomerate properties we could investigate: What is the structure of addresses in IPv4 traffic? In particular, how are packets distributed among a conglomerate's component addresses, and how do those addresses aggregate? The answers to these questions are relevant to many models of conglomerates, such as models of how they are routed by the network, and it turns out that addresses in IP traffic exhibit surprisingly rich structure.

We begin with descriptions of our methodology and data sets (Sections III and IV). In particular, we motivate our widespread use of *destination-prefix aggregation*, where two packets are in the same p-aggregate if their destination addresses share a p-bit prefix. We then examine factors contributing to the observed distribution of packets per destination-prefix aggregate, which has a heavy, Pareto-like tail (Section V). This is related to the well-known "mice and elephants" phenomenon, whereby most flows are small, but some flows contain vastly more packets than the average. By applying different types of random shuffling, we show that *address structure*—the arrangement of active addresses in the address space—has a greater effect on aggregate packet counts than the arrangement of packets into flows, at least for medium-to-large aggregates such as /16s. This motivates our investigation of address structure itself.

Under visual examination, the set of addresses in a trace appears broadly self-similar: some structural features reappear at different scales. (For example, see Fig. 1.) We therefore explore fractal address models in Section VI. It turns out that our example address structures are well-described by a two-parameter multifractal model. This parsimonious model captures much, though not all, of the address structure observed in our traces, and provides promise both for synthesizing realistic address structures for simulation, and as an analytic framework for further study. This model is the paper's core result.

In Section VII, we further explore our data sets and our model using concepts and analytic tools designed for analyzing address structures. We finish in Section VIII with a look at how address structure properties vary: over time, from site to site, and for different types of traffic. We find that the structure of aggregates seen at a site is steady over time, that different sites exhibit distinctly different address structures, and that broadly distributed traffic patterns such as the Code Red 1 and 2 worms of July and August 2001 have, not surprisingly, their own striking signature.

An Appendix presents supplementary graphs using additional data sets and parameters.

Manuscript received October 2, 2002; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Rexford. The work of E. Kohler was supported in part by the National Science Foundation under Grant 0230921. This work was largely performed at ICSI. A previous version of this paper appeared in the Second Internet Measurement Workshop, Marseille, France, Nov. 2002.

E. Kohler is with the Computer Science Department, University of California, Los Angeles, CA 90095 USA (e-mail: kohler@cs.ucla.edu).

J. Li is with the Computer Science Department, New York University, New York, NY 10012 USA.

V. Paxson is with the International Computer Science Institute (ICSI), Berkeley, CA 94704 USA, and also with the Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA.

S. Shenker is with the International Computer Science Institute (ICSI), Berkeley, CA 94704 USA, and also with the Computer Science Department, University of California, Berkeley, CA 94720 USA.

Digital Object Identifier 10.1109/TNET.2006.886288

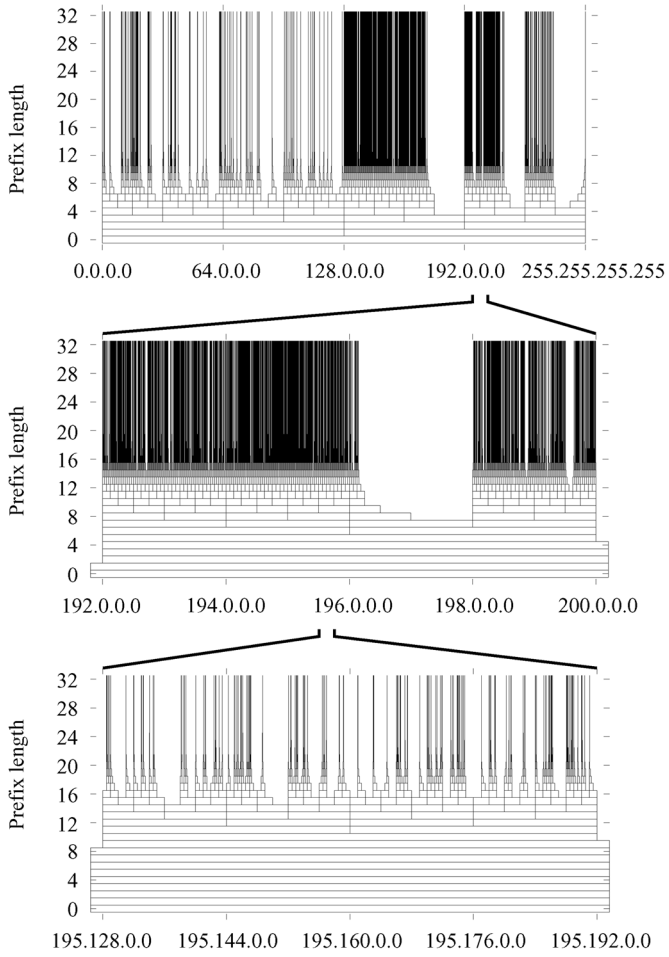

Fig. 1. The address structure of data set U1, with two successive 32× magnifications. We draw a box for every nonempty address prefix; the Y axis is prefix length. A single address would generate a stack of 33 boxes, each half the width of the one below. The topmost boxes are extremely thin!

II. Related Work

We are not aware of similar previous work on characteristics of IP address structure. More broadly, much effort has gone into modeling the structures of traffic bursts in the Internet; like address structure, measured traffic appears to be self-similar [1], [7] and exhibit multifractal characteristics [8]. Krishnamurthy and Wang [9] have previously investigated the properties of client addresses aggregated according to BGP routing prefixes. Their results indicate that these aggregates have a heavy-tailed distribution, like our destination-prefix aggregates. Researchers have begun investigating destination-prefix aggregate properties for aggregate congestion control [10].

III. Destination-Prefix Aggregation

We begin with the fundamental definition of what makes up a traffic aggregate. In this paper, two packets are in the same aggregate iff the first p bits of their destination addresses are equal. Different aggregate sizes use different p. Destination address prefix makes a good aggregate definition for several reasons.

- IP addresses were built for prefix aggregation. The initial IP specification divided addresses into classes based on 1- to 4-bit address prefixes. Depending on class, an 8-, 16-, or 24-bit network prefix determined where a packet should be routed [11]. Classless inter-domain routing (CIDR) [12], which replaced this system as address blocks became scarce, kept the notion of identifying networks by address prefixes, but allowed those prefixes to have any length.
- Likewise, allocation proceeds in prefix-based blocks. IANA delegates short prefixes (which contain many addresses) to other organizations, which then delegate sub-prefixes to their customers, and so forth. This property can relate other aggregate definitions—geographic location or round-trip time, for instance—back to address prefixes.
- IP routers make their routing decisions based on destination address prefix—a longest-prefix-match lookup on all routes keyed by the packet's destination address. Therefore, the characteristics of observed destination-prefix-based aggregates intimately affect the performance of route cache strategies. Other router algorithms that work on aggregates, such as aggregate-based congestion control [10], often define aggregates by destination prefix, since routers already use them for route lookup.

One could usefully define aggregates in many other ways, such as by destination geographic area or application protocol, but we only consider aggregates defined by destination address prefixes.

CIDR notation is used for prefixes and aggregates. Given an IP address a and prefix length p, with $0 \leq p \leq 32$, "a/p" refers to the p-bit prefix of a or, equivalently, the aggregate consisting of all addresses sharing that prefix. An aggregate with prefix length p is called a *p-aggregate*, or, sometimes, a "/p". A p-aggregate contains 2^{32-p} addresses, so aggregates with short prefix lengths contain more addresses; the single 0-aggregate contains all addresses and a 32-aggregate is equivalent to a single address. We use the terms "short" and "long" when referring to prefixes, and "small" and "large" when referring to aggregates; short prefixes correspond to large aggregates, and long prefixes to small aggregates.

IV. Data Sets

Our packet traces originate at locations that generally see a lot of traffic aggregation, including access links to universities (U1 and U2) and busy Web sites (W1), ISP routers with peering, backbone, and client links (A1 and A2), and links connecting large metropolitan regions with a major ISP backbone (R1 and R2). The traces date from between 1998 and 2001. Their durations range from 1 to 4 hours, and their packet counts range from 1.4 million to 101 million. We write N for the number of distinct destination addresses in a trace; it ranges from 70 000 to 160 000. Some traces were pseudo-randomly sampled at the packet level. Fig. 2 presents high-level characteristics of these data sets. We believe that traces from sites that see less aggregation, or that draw from a narrower user base, might exhibit different characteristics. Although the properties visible at these locations will have changed over time, a set of external addresses observed at a national laboratory in 2005 demonstrates structural characteristics not far from those observed at that laboratory in 2001 (Section VIII-C).

Trace	Description	Time (hr)	N	Packet count	Sampled?
U1	Access link to a large university	~ 4.0	69,196	62,149,043	no
U2	Access link to a large university	~ 1.0	144,244	101,080,727	no
A1	ISP	~ 0.6	82,678	33,960,054	no
A2	ISP	1.0	154,921	29,242,211	no
R1	Link from a regional ISP	1.0	168,318	1,476,378	1 in 256
R2	Link from a regional ISP	2.0	110,783	1,992,318	1 in 256
W1	Access link in front of a large Web server	~ 2.0	124,454	5,000,000	no

Fig. 2. Characteristics of our traces.

The addresses in many of our traces have been anonymized while preserving prefix relationships. (This kind of anonymization seems to have been introduced by the *tcpdpriv* program's $-A50$ option [13].) A prefix-preserving anonymization function f maintains the property that for any addresses a and b and prefix length p, $a/p = b/p$ iff $f(a)/p = f(b)/p$. All our analysis methodologies are indifferent to this anonymization.

All of our traces are omnidirectional, meaning that they contain all packets passing by the trace location, regardless of whether the packets were heading "towards" or "away from" the trace point. We experimented with algorithms to extract likely unidirectional traces from omnidirectional ones. On seeing a packet with source address a and destination address b, one can assume, modulo spoofing and misrouting, that a is on one side of the link and b is on the other. Running trace R1 through a conservative algorithm based on this insight yielded three address sets: 12% of addresses were "internal", 68% were "external", and 21% could not be classified. (The large number of unclassifiable addresses is partially due to R1's 1-in-256 sampling, which reduces the algorithm's efficacy.) The structural metrics (see Section VIII) of the whole trace follow those of the "external" addresses, probably because there are relatively few "internal" addresses.

Given omnidirectional traces at locations with symmetric routing, we would expect the set of source addresses in the trace to roughly equal the set of destination addresses. This holds true for some, but not all, of our traces. For example, 93% of the addresses in trace U1 appear both as source addresses and as destination addresses, while just 17% of the addresses in trace A1 occur both as sources and as destinations.

It can be useful to develop a general intuition about how address structures look before considering their mathematical properties. Fig. 1 presents one simple visualization of a sample address structure, namely the destination addresses present in trace U1. We draw a box for each aggregate containing at least one address present in the trace. Regions of the address space fall into three visually distinct categories: sparsely populated, such as class A (0.0.0.0 to 128.0.0.0); densely populated, such as class C (192.0.0.0 to 224.0.0.0); and empty, generally address space reserved by the IETF (such as 240.0.0.0 to 255.255.255.255). The address structure appears broadly self-similar, in that structural features recur at different scales. For instance, compare the bottom diagram (195.128.0.0 to 195.192.0.0) with class A in the top diagram (0.0.0.0 to 128.0.0.0). Other traces look generally similar when graphed in this way.

V. Importance of Address Structure

A *packet count distribution* graph shows how the number of packets per group—TCP flow, destination address, or destination-prefix aggregate—varies over the set of all groups. These distributions are significant for congestion control and fairness applications, among others. We are particularly interested in the distributions' rough *shape*—for example, normally distributed, uniform random, or heavy-tailed. Examining these distributions demonstrates the importance of address structure. We see that all three distributions are heavy-tailed, and that address structure is the strongest factor affecting the packet count distribution for medium-sized aggregates.

Relevant characteristics of R1, the trace used throughout this section, are as follows.

Trace duration	1 hour
Sampling ratio	1/256
Number of packets	1 476 378
Number of non-TCP/UDP packets	36 445
Number of TCP/UDP flows	680 663
Number of active addresses (N)	168 318
Number of active 16-aggregates	5785

A. Packet Count Distributions

A random variable X follows a *heavy-tailed distribution* if it is about as likely to exceed a large value as it is to exceed any *larger* value [14]:

$$\lim_{x\to\infty} \frac{P[X > x + y]}{P[X > x]} = 1 \quad \text{for all } y > 0.$$

Thus, the distribution's *tail*—the complement of its distribution function—maintains meaningful probability, no matter how far out that tail is measured. Of course, in any finite distribution the tail is truncated eventually. Heavy-tailed distributions have been frequently observed in natural and artificial phenomena, including the Internet [15], [16].[1] The simplest heavy-tailed distribution is the power-law distribution, where $P[X > x] \sim x^{-\alpha}$ as $x \to \infty$ for some $0 < \alpha < 2$.

Log-log complementary CDF graphs form a well-known test for heavy-tailed distributions. These plots show, for a given x, the fraction of entities that have weight x or more, with both axes in log scale. Power-law distributions appear as straight lines on these graphs for sufficiently large x.

Fig. 3 presents a log-log complementary CDF of the packet counts of TCP/UDP flows, addresses, and 16-aggregates in the R1 data set.[2] The graph's X axis marks the number of packets attributed to an entity—flow, address, or aggregate. (The largest entities in the trace are visible as the endpoints of the lines. The largest flow in the trace contains 3727 sampled packets,

[1]In some cases these observations may be biased by measurement methodology [17].

[2]Appendix Fig. 18 shows similar graphs for other data sets.

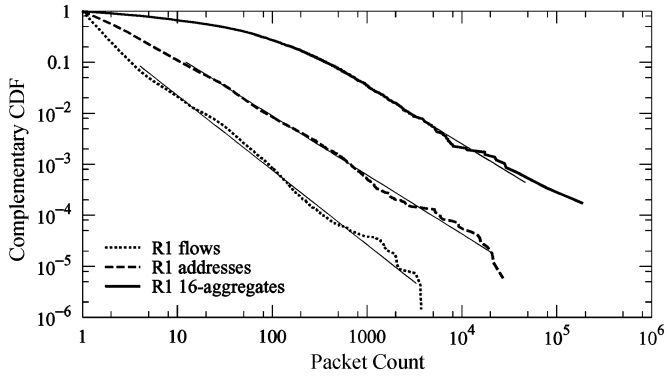

Fig. 3. Log-log complementary CDF of packet counts for R1 flows, addresses, and 16-aggregates. All are consistent with power-law distributions. The fit lines have slopes −1.46, −1.16, and −1.13, respectively.

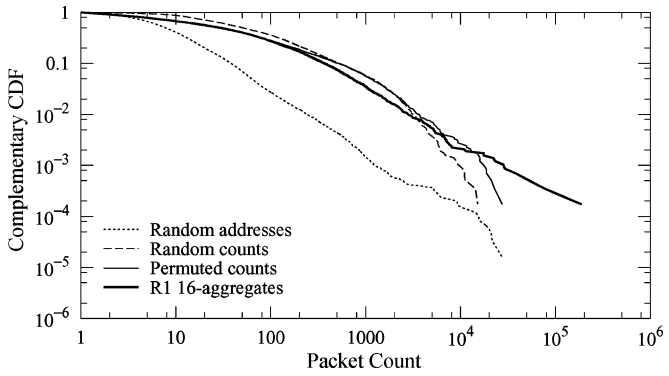

Fig. 4. Complementary CDF of 16-aggregate packet counts for R1 with random addresses, R1 with random address packet counts, R1 with permuted address packet counts (but the same addresses), and R1 itself (line repeated from Fig. 3).

the largest destination address has 27 020 sampled packets, and the largest 16-aggregate has 187 227 sampled packets.) All three distributions appear to have power law tails. That is, the chance that an entity has weight greater than x is proportional to $x^{-\alpha}$ with $0 < \alpha < 2$; here, α is approximately 1.46 for flows, 1.16 for addresses, and 1.13 for 16-aggregates. These values were calculated by least-squares fit to the upper 10% of the distributions' tails, less the last 5 data points. Other traces have similar packet count distributions, although some have lighter tails.

Prior work has shown that Web flow weights follow a heavy-tailed distribution [15], and 70% of R1's packets, and 89% of its flows, use ports 80 (http) or 443 (https). Thus, the heavy-tailed nature of the TCP/UDP-flow packet count distribution comes as no surprise. However, we might also have expected large aggregates to appear less heavy-tailed than flows or addresses. Each 16-aggregate can contain tens of thousands of flows, and the sum of so many finite distributions would tend to converge, however slowly, to a normal distribution. We see no significant convergence in our data, however: the 16-aggregate packet count distribution appears, if anything, *more* heavy-tailed than the flow packet count distribution. Why might this be so?

B. Factors Affecting Aggregate Packet Counts

The number of packets in a particular address aggregate can be analyzed as depending on three factors:

1) *Address packet counts*: How many packets are there per destination address?
2) *Address structure*: How many active addresses are there per aggregate? (We call a destination address *active* when its packet count is at least one. Thus, our definition of address structure does not differentiate between popular and unpopular destinations.)
3) The *correlation* between these factors: Do addresses with high packet counts tend to cluster together in the address space? Or do they tend to spread out? Or neither?

We can empirically evaluate the relative importance of these factors by altering each factor in turn, then comparing the resulting aggregate packet count distributions with those of the real data R1. To this end we transform the R1 data set in three ways.

1) "Random counts": This transformation replaces all address packet counts in the data set with numbers drawn uniformly from the interval [0, 17.54]. This destroys address packet counts and correlation while preserving address structure. (17.54 is twice R1's mean address packet count.)
2) "Random addresses": To alter address structure, we randomly choose 168 318 addresses from the address space, then assign R1's address packet counts to those addresses. This preserves the address packet count distribution while destroying address structure and correlation.
3) "Permuted counts": To destroy any correlation between the two distributions while preserving the distributions themselves, we keep the original addresses, but randomly permute their packet counts.

Per-address packet counts dominate the packet counts of small aggregates. That is, for 24-aggregates and smaller, the aggregate packet count distribution of "random addresses" resembles that of the real data, while that of "random counts" does not. This makes intuitive sense. A 30-aggregate, for example, can contain at most four addresses, so address structure and correlation can have minimal impact on 30-aggregate packet counts.

For medium-to-large aggregates, however, the story is quite different. Fig. 4 shows the results for 16-aggregates.[3] All three generated sets differ from the real data, but unlike "random counts" and "permuted counts", the "random addresses" line differs significantly across the entire range of values. This underlines the importance of address structure: for medium-to-large aggregates, address structure has a greater effect on aggregate packet counts than per-address packet counts. In order to understand aggregate packet counts, we must understand how addresses aggregate.

VI. Multifractal Model

Fig. 1 shows that real address structures look broadly self-similar: meaningful structure appears at all three magnification levels. We now validate that intuition by presenting a multifractal model for observed address structures. Although address structures bottom out at prefix length 32, whereas true fractals have structure down to infinitely small scales, this is still enough depth to make fractal models potentially valuable.

[3]Appendix Fig. 19 shows similar graphs for 8- and 24-aggregates.

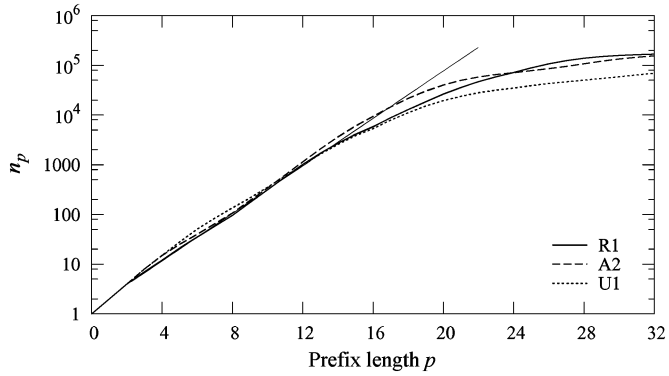

Fig. 5. n_p as a function of prefix length for several traces, with a least-squares fit line for R1's $4 \leq p \leq 14$ region (fit slope 0.79).

A. Fractal Dimension

An address structure can be viewed as a subset of the unit interval $I = [0,1)$, where the subinterval $[a/2^{32}, (a+1)/2^{32})$ corresponds to address a. Considered this way, address structure might resemble a Cantor set-like fractal [18], [19]. The classic Cantor set is created by repeatedly removing the open middle third from each of a set of line segments, where the set is initialized with the unit interval [0,1]. The result is an uncountably infinite set of points that nevertheless contains no continuous interval. The Cantor set has topological dimension 0 (since it consists of isolated points), but it is also a fractal—a set with interesting structure at all scales—and has an intermediate fractal dimension, namely ≈0.63. Like the Cantor set, address structures are sets of points with structure at many scales (although unlike the Cantor set, they are finite). Considered this way, what would be the dimension of our address structure?

The *box-counting* fractal dimension metric, or *Kolmogorov capacity*, fits naturally with address structures and prefix aggregation. If $n(\epsilon)$ is the number of closed boxes of side length ϵ required to cover some set, then the set's box-counting dimension equals

$$D = -\lim_{\epsilon \to 0} \frac{\log n(\epsilon)}{\log \epsilon}.$$

The same measure may be obtained usinerg *dyadic intervals*, which arise from repeated bisection of the unit interval. A dyadic interval with length $1/2^p$ occupies $[n/2^p, (n+1)/2^p]$ for p and n non-negative integers with $n < 2^p$, and thus corresponds to a p-aggregate in our address structure model. Given a trace, let n_p be the number of p-aggregates that contain at least one address present in the trace as a destination $(0 \leq p \leq 32)$. Any nonempty trace will have $n_0 = 1$, since the single 0-aggregate covers the entire address space, and $n_{32} = N$ is the number of distinct destination addresses present in the trace. Furthermore, since each p-aggregate contains and is covered by exactly two disjoint $(p+1)$-aggregates, we know that $n_p \leq n_{p+1} \leq 2n_p$. Box-counting dimension may then be evaluated as

$$D = \lim_{p \to \infty} \frac{\log n_p}{p \log 2}.$$

This definition of course bounds D between 0 and 1.

If address structures were fractal, $\log n_p$ would appear as a straight line with slope D when plotted as a function of p. We would actually expect to see startup effects for low p (higher slope than the true dimension) and sampling effects for high p (lower slope than the true dimension, because there's not enough data to fill out the fractal). Fig. 5 shows a log plot of n_p as a function of p; we find that, for a reasonable middle region $4 \leq p \leq 14$, n_p curves do appear linear on a log-scale plot. For R1, a least-squares fit to this region gives a line with slope 0.79. Thus, R1's nominal fractal dimension is $D = 0.79$.

B. Multifractality

Adaptations of the Cantor set construction can generate address structures with any fractal dimension. The relative size h of the portion removed from each line segment determines the dimension of the resulting set:

$$D = -\frac{\log 2}{\log \frac{1}{2}(1-h)}.$$

For the canonical Cantor set, $h = 1/3$ and $D = \log 2 / \log 3$. Any address interval containing a point of the resulting set could represent an active address.

Such Cantor-like sets can capture the global scaling behavior of aggregate counts. However, real address structure is more complicated than what they can predict. Cantor-like sets have the same local scaling behavior everywhere in the address space, modulo sampling effects. Traces, on the other hand, populate portions of the address space quite differently, as can be seen in Fig. 1. This results in different local scaling behavior, where points cluster more strongly in some regions than others—the essence of multifractality.

To test if a data set is consistent with the properties of multifractals, we use the *Histogram Method* to examine its *multifractal spectrum* [19]. This method evaluates *local scaling exponents*, which measure the approximate scaling behavior near a given point in the structure. In a monofractal, local scaling exponents will all approximate the fractal dimension, but in a multifractal, they vary considerably. Let $\sigma_p(a)$ denote the number of active addresses in the aggregate a/p. Then since address structures treat all active addresses identically, $\mu_p(a)$, the "mass" or probability associated with a/p, equals $\sigma_p(a)/N$. When $\mu_p(a) > 0$, the *local scaling exponent* $\alpha_p(a)$ is defined as follows:

$$\alpha_p(a) = \frac{\log \mu_p(a)}{\log 2^{-p}} = -\frac{\log\left(\sigma_p(a)/N\right)}{p \log 2}.$$

To calculate a multifractal spectrum, first compute a histogram of α_p. That is, decide on a set of evenly sized histogram bins, and for each bin B_i, calculate F_i, the number of aggregates a/p whose $\alpha_p(a)$ value lies within that bin. The multifractal spectrum plots $f_p(B_i) = \log F_i / p$ versus the binned scaling exponents.[4] For multifractal data, this spectrum will collapse onto a single curve for sufficiently large p. Our data sets are dominated by sampling effects for large p, however, so we examine medium p instead. The solid line in Fig. 6 shows R1's multifractal spectrum at $p = 16$; spectra at nearby prefixes are similar. It covers a wide range of values. The dashed line corresponds to an address structure sampled from a Cantor-like set

[4]Strictly speaking, the multifractal spectrum is continuous; this is a binned approximation.

with fractal dimension 0.79, the same as R1's nominal fractal dimension. 168 318 addresses were sampled, giving the set the same number of addresses as R1. The full Cantor-like set has a single fractal dimension, but this single dimension appears cleanly only in the limit; at any individual aggregation level, such as that used in Fig. 6, multiple scaling exponents are visible. Nevertheless, R1's multifractal spectrum is significantly wider than that of the Cantor-like set, indicating that R1 demonstrates multifractal-like behavior.

C. Model

The original Cantor construction can be easily extended to a *multifractal Cantor measure* [20], [21]. Begin by assigning a unit of mass to the unit interval I. As before, split the interval into three parts where the middle part takes up a fraction h of the whole interval; call these parts I_0, I_1, and I_2. Then throw away the middle part I_1, giving it none of the parent interval's mass. The other subintervals are assigned masses m_0 and $m_2 = 1 - m_0$. Recursing on the nonempty subintervals I_0 and I_2 generates four nonempty subintervals I_{00}, I_{02}, I_{20}, and I_{22} with respective masses m_0^2, $m_0 m_2$, $m_2 m_0$, and m_2^2. Continuing the procedure defines a sequence of measures μ_k where $\mu_k(I_{\varepsilon_1 \ldots \varepsilon_k}) = m_{\varepsilon_1} \times \cdots \times m_{\varepsilon_k}$ (each ε_i is 0, 1, or 2); these measures converge weakly towards a limit measure μ. To create an address structure from this measure, we choose N addresses where the probability of selecting address a equals $\mu(A_a)$. If $m_0 = m_2 = 1/2$, this replicates the Cantor construction. If m_0 and m_2 differ, however, the measure μ is multifractal. Although the set of mathematical points with nonzero mass equals the original Cantor set, and has the same basic fractal dimension, the measure's unequal distribution of mass causes the sampled set of addresses to exhibit a wide spectrum of local scaling behaviors.

We constructed another set of addresses, the "R1 Model", by generating 168 318 addresses according to a Cantor measure with basic fractal dimension $D = 0.79$ and with $m_0 = 0.8$ (chosen to fit the data). The dotted line on Fig. 6 shows its multifractal spectrum.[5] The measure is partially deterministic—the Cantor construction's excluded middle means that some addresses will never be chosen—but not entirely. Nevertheless, several samples of the measure led to similar results. The single parameter m_0 is sufficient to make the model match real data fairly well at all scaling exponents.

We created similar models for several other traces, using fractal dimensions and m_0 as follows:

Trace	D	m_0	Trace	D	m_0
R1	0.79	0.80	A2	0.80	0.70
U1	0.73	0.72	W1	0.83	0.75

Each trace's fractal dimension D was measured as the slope of the least-squares fit line on a graph of $\log_2 n_p$ versus p for $4 \le p \le 14$. Each trace's mass proportion m_0 was chosen so that the model's multifractal spectrum covered a similar range as that of the trace. Fig. 7 shows the multifractal spectra for A2 and its model at $p = 16$.[6]

[5]Appendix Fig. 20 shows spectra for R1 and its model at $p = 15$, 17, and 18.

[6]Appendix Fig. 21 shows multifractal spectra at $p = 16$ for all data sets; Appendix Fig. 22 compares the spectra for U1 and W1 to those for their models.

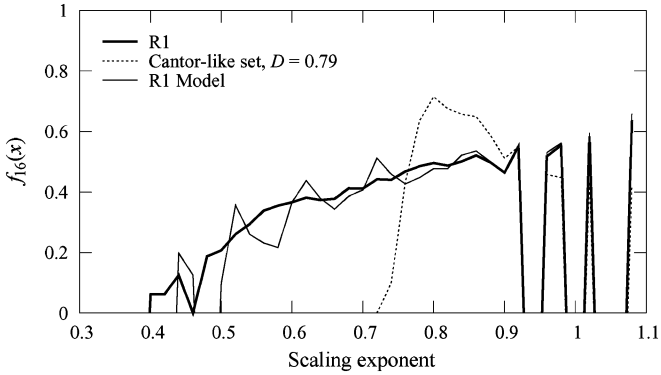

Fig. 6. Multifractal spectra for R1 and Cantor sets, $p = 16$.

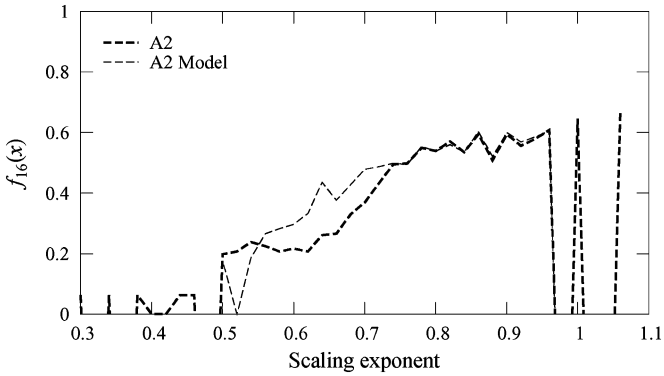

Fig. 7. Multifractal spectra for A2 and its model, $p = 16$.

All of these models broadly match the real data's multifractal spectra. The trace spectra cover different ranges of scaling exponents, but modifying m_0 seems sufficient to capture this variation. In particular, raising m_0 increases the range of scaling exponents on the spectrum, as one would expect. We also experimented with fixing m_0 at our optimal guess and varying D. As D rose above the measured dimension, the model's fractal spectrum fragmented into more spikes; as it lowered below the measured dimension, the model's spectrum smoothed out, but also covered a narrower range of scaling exponents and fell below the real spectrum.

D. Causes

Why might IP addresses appear to be multifractal? This area needs more investigation, but there is an attractive, intuitive explanation. Multifractals can be generated by a *multiplicative process* or *cascade* that fragments a set into smaller components recursively—for example, taking out the middle subinterval as in a Cantor set—while redistributing mass associated with these components according to some rule—for example, a higher probability of further populating the resulting left subinterval. This brings to mind the way IP addresses are allocated: ICANN assigns big IP prefixes to the regional registrars, the registrars assign blocks to ISPs, who further assign sub-prefixes to their customers, and so forth. For social and historical reasons, many of these allocation policies may share a simple basic rule—for example, left-to-right allocation. Together, these processes would generate a cascade, and multifractal behavior.

The model presented above is by no means the only way to generate a set of addresses consistent with multifractal behavior. For example, one can repeatedly divide the unit interval I in

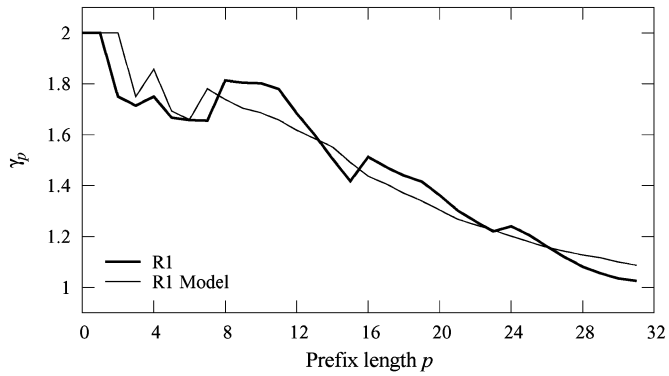

Fig. 8. γ_p, aggregation ratios.

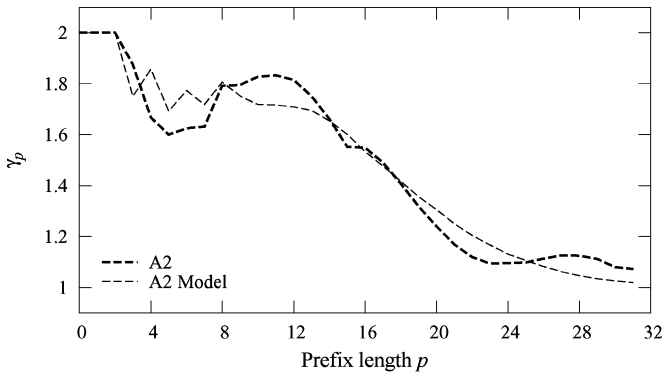

Fig. 9. γ_p for A2 and its model.

half, each time associating random variables, possibly with log-normal or other nonuniform distribution, with the two halves. Points would be chosen according to the resulting probability distribution, which, unlike the multifractal Cantor measure described above, assigns a nonzero probability to every address. Preliminary experiments indicate that this kind of "random cascade" can match the multifractal spectrum of real data, although we had less success matching the metrics described below.

VII. Metrics

We have seen that a surprisingly simple model of address structure captures the multifractal behavior of real data. Now, we test that model against generic structural metrics that measure how addresses are aggregating. Our goal is to test whether the multifractal model matches real data in simple summary metrics with real-world relevance, in addition to the multifractal spectrum. We introduce three metrics: *active aggregate counts*, which measure where nontrivial aggregation takes place; *discriminating prefixes*, which measure the separation between aggregates; and *aggregate population distributions*, which show how addresses are spread across aggregates.

A. Active Aggregate Counts (n_p and γ_p)

One measurement of how densely addresses are packed is simply how many aggregates there are. A trace containing 10 000 distinct destination addresses might have a single active 16-aggregate, if the addresses were closely packed, or 10 000 different 16-aggregates, if they were maximally spread out. The active aggregate counts n_p, introduced in Section VI-A, capture this notion by counting the number of active p-aggregates for every p. For instance, n_{16} is the number of active 16-aggregates: the number of /16s that contain at least one active address. This measure is relevant to the design of algorithms that keep track of aggregates, since it shows how many aggregates there are on average.

The ratio $\gamma_p = n_{p+1}/n_p$ is often more convenient for graphing than n_p itself.[7] Fig. 8 shows the values of γ_p for R1, and our multifractal model tuned for R1; Fig. 9 shows A2 and the model for A2.[8] γ_p drops vaguely linearly from 2 to 1, corresponding to exponential growth in aggregate counts that gradually flattens out as prefixes grow longer. (γ_p always lies between 1 and 2.) The models' plots are smoother than the real data for $p \geq 6$ or so, but they do match in broad outline. For example, note how the plots for A2 and its model dip lower than those for R1 and its model at $p > 18$. The bumps in γ_p at $p = 8, 16$, and 24 are probably caused by traditional class-based address allocation, still visible years after the introduction of CIDR [22].

Some properties of trace locations may be inferred from graphs of γ_p. For example, A2's γ_p is lower than R1's around $p = 18$ to 24, but higher for $p > 26$. This means that more of A2's aggregation takes place at long prefixes: active addresses are closer to one another than in R1. We hypothesize that A2's location, at an ISP with both peering and customer links, accounts for this; maybe A2's direct customers have relatively many closely packed active addresses.[9]

B. Discriminating Prefixes

Active aggregate counts measure address density, but cannot always characterize address *separation*. An address might be the only active address in its half of the address space, in which case we would call it well-separated from other addresses, or it might be part of a completely populated 16-aggregate. The n_p and γ_p metrics cannot always distinguish between cases where all 16-aggregates (say) are equally populated and cases where some 16-aggregates are fully populated and others are sparsely populated, meaning some addresses are more separated than others. To measure address separation, we introduce a new metric, *discriminating prefixes*.

The discriminating prefix of an active address a is the prefix length of the largest aggregate whose only active address is a. Thus, if the discriminating prefix of an address is 16, then it is the only address in its containing 16-aggregate, but the containing 15-aggregate pulls in at least one other active address. Fig. 10 demonstrates this concept on an example set of 4-bit-long addresses. If many addresses have discriminating prefix less than 20, say, then active addresses are generally well separated, and we would expect aggregates to contain small numbers of active addresses.

[7]Nevertheless, Appendix Fig. 23 graphs of n_p for all data sets and models.

[8]Appendix Fig. 24 graphs of γ_p for all data sets and models, and Appendix Fig. 25 compares γ_p for U1 and W1 to their models.

[9]Our algorithm for identifying "internal" and "external" addresses in omnidirectional traces, which classified 79% of R1's addresses, was able to classify only 21% of A2's addresses. This might indicate a complex conversation pattern, such as high levels of communication among A2's customers. Intuitively, such a communication pattern might correlate with closely packed active addresses—for example, if several of A2's customers were different campuses of a single organization.

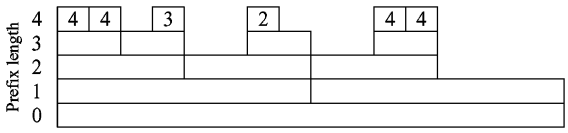

Fig. 10. Discriminating prefix example with 4-bit addresses. The top boxes are active addresses; lower boxes represent active aggregates, as in Fig. 1. Each active address's discriminating prefix is shown inside its box.

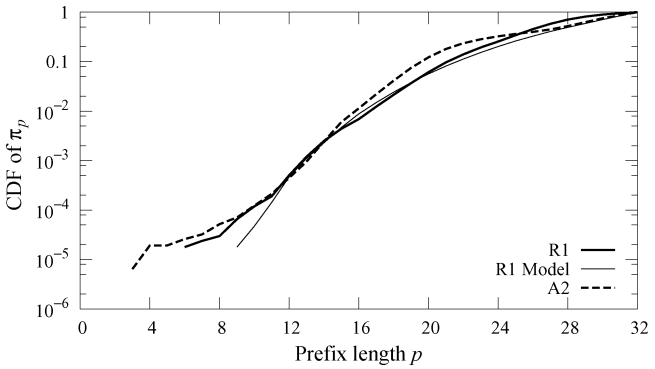

Fig. 11. CDF of address discriminating prefix counts π_p.

We turn discriminating prefixes into a metric by calculating π_p, the number of addresses that have discriminating prefix p, for all $0 \leq p \leq 32$. Since every address has exactly one discriminating prefix, $\sum \pi_p = N$.

Fig. 11 graphs π_p for R1, A2, and our R1 model.[10] The traces' discriminating prefixes range widely, indicating wide variability in address separation. Discriminating prefixes get surprisingly low: one R1 address has a discriminating prefix of 6 (since $\pi_6 > 0$), meaning that some active 6-aggregate contains exactly one active address. (However, the majority of addresses have discriminating prefix 26 or higher.) The model captures this range in discriminating prefixes, although it does not create discriminating prefixes as low as the real data. Simpler models, such as random address assignment, sequential address assignment, and a monofractal Cantor construction, create much narrower ranges of discriminating prefixes.

C. Aggregate Population Distributions

Aggregate population distributions provide a more fine-grained measurement of how addresses are aggregating at a given prefix length. The *population* of an aggregate is the number of active addresses contained in that aggregate; in Section VI-B, we expressed this as $\sigma_p(a)$. Given our experience with the other metrics, we would expect p-aggregates to exhibit a wide range of populations for short-to-medium p. Longer-prefix aggregates contain fewer addresses, so there is not as much room for variability.

This expectation is confirmed by the data. Fig. 12 graphs 8- and 16-aggregate population distributions for R1 and our R1 model on a log-log complementary CDF: for a given x, the Y axis measures the fraction of aggregates with population at least x. This is the same kind of graph as the aggregate packet count distributions in Section V-A. As expected, aggregates exhibit a

[10]Appendix Fig. 26 graphs π_p for all more data sets and models.

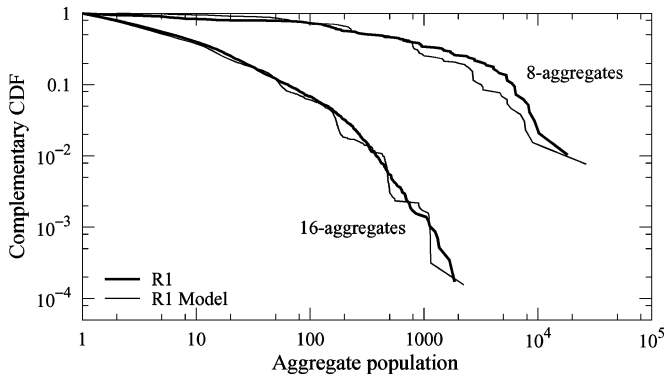

Fig. 12. 8- and 16-aggregate population distributions for R1.

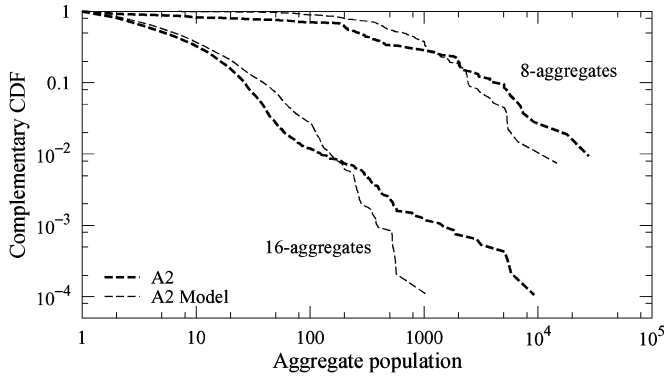

Fig. 13. 8- and 16-aggregate population distributions for A2.

wide range of populations. The multifractal model echoes the real data, particularly in the tail region.

It is worth noting that aggregate population distributions are the most effective test we have found to differentiate address structures. For example, before generating our multifractal model, we developed an algorithm that generates a random address structure exactly matching a given set of γ_p values, discriminating prefixes, and even discriminating prefixes for aggregates. Despite the fitting, the aggregate population distributions generated by the model were far off the real data, much farther off than our current multifractal model.

Aggregate population distributions also demonstrate our model's limitations. Fig. 13 shows distributions for A2 and its model. The model is pretty far off. Overall, the models for R1 and W1 match their traces' aggregate population distributions well, while the models for A2 and U1 do not.[11] The most obvious difference between these sets of traces can be seen on plots of γ_p. A2 and U1 have lower amounts of aggregation at medium-to-long prefixes than R1 and W1, but higher amounts of aggregation at long prefixes. In Figs. 8 and 9, for example, A2's γ_p dips appreciably below that of R1 for $18 \leq p \leq 25$, only to rise above it for $p > 27$. Our current multifractal model does not achieve both these properties simultaneously; if a model has low γ_p for $18 \leq p \leq 25$, it has low γ_p for $p > 27$.

VIII. Properties of γ_p

We now turn from the multifractal address model to the γ_p metric itself. In particular, we investigate γ_p's properties as a concise characterization, or "fingerprint", of the traffic visible

[11]Appendix Fig. 27 shows similar graphs for U1 and W1 and their models.

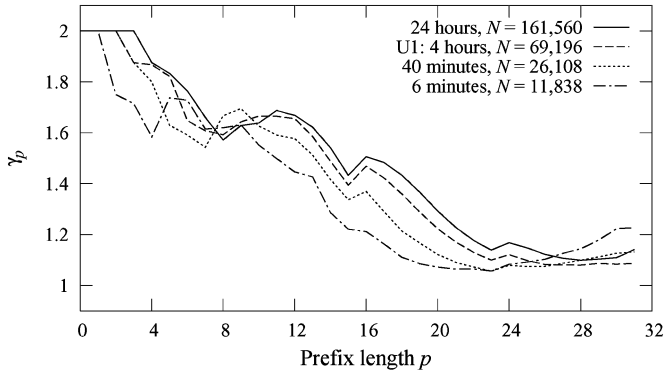

Fig. 14. γ_p for U1, and for longer and shorter traces from the same data.

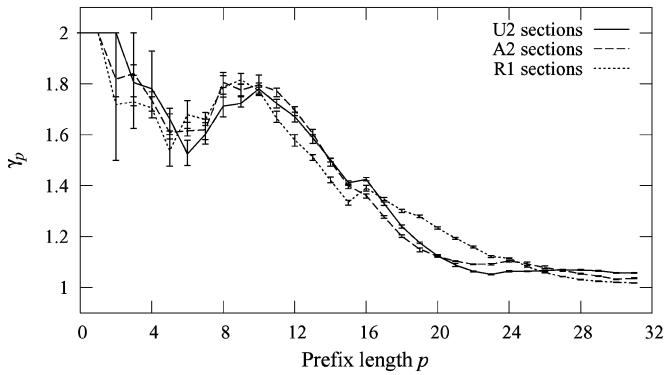

Fig. 15. Variations of γ_p over time for different traces. The error bars indicate the range of variations of γ_p.

at a location. Is γ_p dominated by the sheer number of active addresses (N)? Does the γ_p graph change over short time scales at a single location? And how do unusual events, such as heavy worm propagation, show up in γ_p?

A. Sampling Effects

All of our structural characterizations depend, to some degree, on N, the total number of active addresses observed. Sampling gives a useful analogy. Think of an address trace as a sampling of an underlying discrete probability distribution, where each destination address has a fixed probability. Then N resembles a sample size. How much do n_p and γ_p depend on this sample size? For example, if we sampled shorter or longer sections of a trace, how would that affect γ_p? Too-sensitive dependence on N would make γ_p much less useful as a fingerprint.

We vary N by examining contiguous sections of a 24-hour trace containing U1 as a 4-hour-long subset. These shorter and longer sections effectively represent differently sized samples of an underlying probability distribution, assuming that distribution did not change significantly over the 24-hour period. The distribution almost certainly does change, but our results show its structural characteristics do not change terribly much.

Fig. 14 shows γ_p for U1 traces with durations ranging from 24 hours to 6 minutes. The number of active addresses varies over more than an order of magnitude, from 161 560 to 11 838. We would expect the γ curve to shift downward as N decreases, since N is the product of the γ_ps. For small sample sizes, and the 6-minute trace in particular, the shape of the curve also changes significantly—the characteristic bumps at $p = 16$ and 24 have disappeared and the curve turns up significantly for $p > 24$, a property not visible in any other section.[12] The other curves, however, resemble one another, and differ visibly from other data sets. (Compare Fig. 8, for example.) For this data set, at least, the γ curve displays properties independent of relatively large variations in the sample size N.

B. Short-Term Stability

For address structure characterizations to be useful as traffic "fingerprints", they must not vary too much on the order of minutes or even one hour under normal traffic conditions. We will see that this is indeed the case.

To examine γ_p's stability over time, we break traces U2, A2, and R1 into sequential nonoverlapping segments, each containing 32 768 addresses. That is, we process the traces in temporal order, collecting addresses and packet counts; but just before recording the 32 769th address, we output the current section of the trace and start a new one. The traces break into about 10 sections each. The segments from a given trace all last for about the same duration; the average duration is 6.7 minutes for U2, 7.5 minutes for A1, and 6.6 minutes for R1. We would like sections from the same trace to resemble one another, and to retain their differences from other traces.

First, we calculated the average number of addresses that adjacent sections have in common. If 32 767 addresses are the same, then obviously the sections will have similar characteristics. In fact, about half of the addresses change from section to section; the first and second A1 sections, for example, share just 15 239 addresses.

Despite this major address turnover, Fig. 15 demonstrates that the shape of the γ_p curve remains quite stable, especially for medium-to-large p. Each line shows the average γ_p for the sections of some trace; the error bars on that line show the maximum and minimum γ_p values in any section of that trace. For much of the address space, the error bars from different traces do not even overlap. Note that N is identically 32 768 for every section on the graph: differences between traces are caused purely by address structure.

C. Worms

Up to this point, we have examined the characteristics of address structures under normal network conditions. Now we consider how worm propagation, and specifically the propagation of Code Red 1 and 2, affects address structure.

The Code Red worm [23] exploits a buffer overflow vulnerability in Microsoft's IIS webservers. In order to spread the worm (version 1 and 2) to as many hosts as possible, the worm generates a random list of IP addresses and tries to infect each one in turn. Code Red 1 picks addresses completely randomly. Code Red 2, by contrast, attacks addresses with greater probability that lie within the same aggregates as the infected host. (Three-eighths of the time, it chooses a random address within the same /16; one-half of the time, it chooses within the same /8; one-eighth of the time, completely randomly.) This reduces the time that the worm wastes on dead addresses.

[12]A possible explanation: Like all our traces, U1 contains bidirectional data. At long time scales, the large variety of external sites visited will dominate visible address structure. At short time scales, that variety cannot express itself, so the structural dynamics of internal addresses become more important.

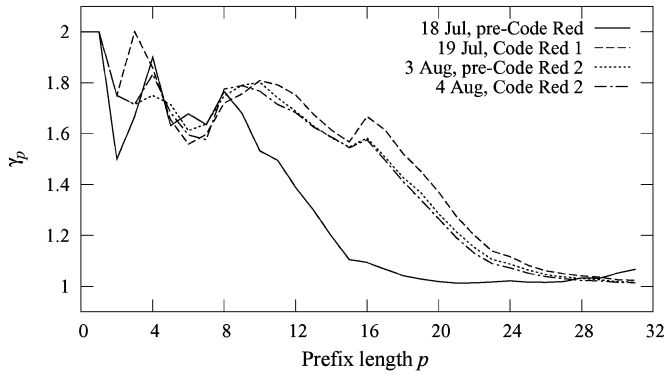

Fig. 16. γ_p for external addresses before and after Code Red 1 and 2.

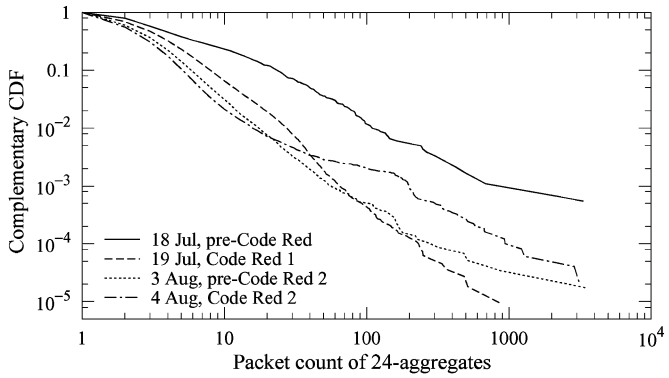

Fig. 17. Aggregate packet count distribution for 24-aggregates before and after Code Red 1 and 2.

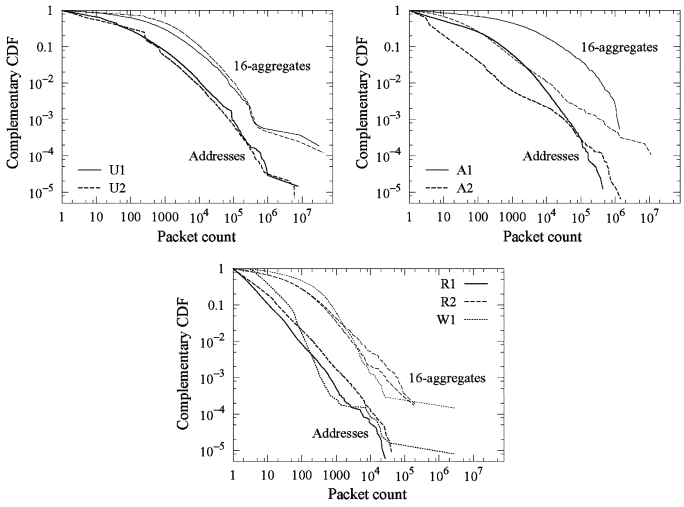

Fig. 18. Log-log complementary CDFs of packet counts for addresses and 16-aggregates in all traces. (See Section V-A.)

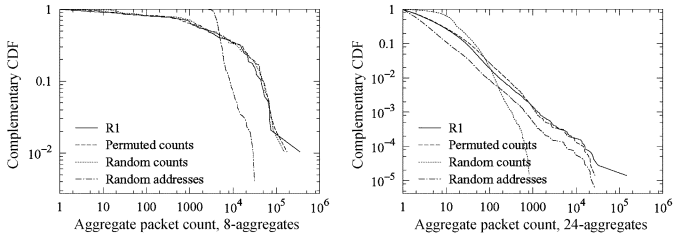

Fig. 19. Complementary CDFs of 8- and 24-aggregate packet counts for R1 and modified traces. (Compare Fig. 4 in Section V-B.) Address structure seems to be the most important factor affecting aggregate packet counts for /8s, but per-address packet counts dominate for smaller /24s.

We would expect this behavior to greatly affect the address structure observed at a given site. Any site has a usual probability distribution for the addresses that might be expected to access it in a given time; Code Red would add all infected hosts to that distribution. Also, the sheer magnitude of Code Red would change the address structure by changing the rate at which new addresses enter the system. We examine the address structure not to advocate its use for worm detection, but to demonstrate network behavior very different from the normal conditions described elsewhere in this work.

We obtained hour-long flow traces from a national laboratory taken the day before Code Red 1 hit (July 18, 2001, $N = 2332$); the first day of Code Red 1's widespread infection (July 19, 2001, $N = 167\,563$); the day before Code Red 2 hit (August 3, 2001, $N = 79\,563$; Code Red 1 was still active); and the first day of Code Red 2's widespread infection (August 4, 2001, $N = 63\,954$). Unlike our other traces, these contain only the addresses of hosts outside the laboratory that attempted to open connections inside the laboratory. This avoids effects from the lab's own infected hosts.

As expected, Code Red wildly changed the structure of addresses seeking to contact the lab. Fig. 16 shows a plot of γ_p for the four traces. The July 18 line is representative for connections predating Code Red: small N, small γ_p. After Code Red, a much broader range of addresses contact the lab, raising N and the aggregate ratio. The aggregate packet count distribution, shown in Fig. 17, changes as well; it drops, since many aggregates have been added that contain only unsuccessful probes. Fig. 17 may also demonstrate a difference between the two Code Reds: Code Red 2 generates more medium-sized aggregates, perhaps because its locality means that networks near the lab in IP space tend to probe it more often.

IX. Conclusion

Address structure is key to understanding some interesting properties of large aggregates, such as their packet count distributions. A multifractal model of observed addresses can echo many properties of the address structures we collected. We developed specific structural characterizations to examine how addresses aggregate at different levels. These structural characterizations differ between sites, yet are relatively insensitive to sample size and stable over short time scales. Without a convincing description of how address structure arises, the results of these explorations must be considered preliminary.

Appendix

Additional Figures

These additional figures (Figs. 18–27) show our data sets in more depth. The main text refers to them in footnotes where appropriate. Notes on particular figures follow.

Fig. 18 shows log-log complementary CDFs of packet counts for addresses and 16-aggregates for all our traces; compare Fig. 3 in Section V-A. We were not able to calculate packet counts for TCP/UDP flows for many of these traces because

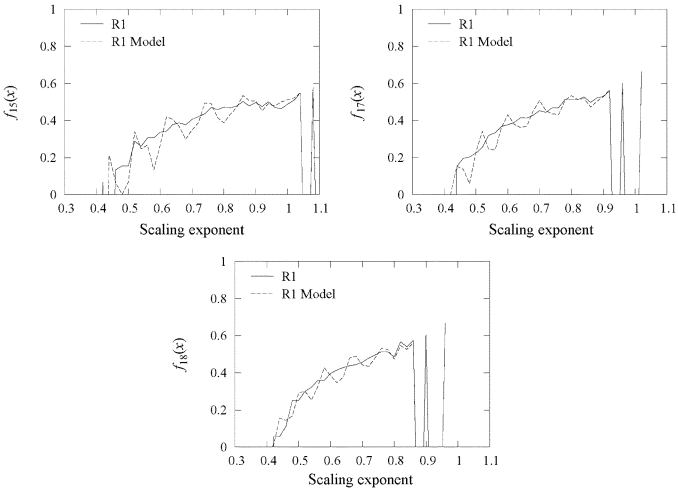

Fig. 20. Multifractal spectra for R1 and its model, $p = 15$, 17, and 18. (See Section VI-C.)

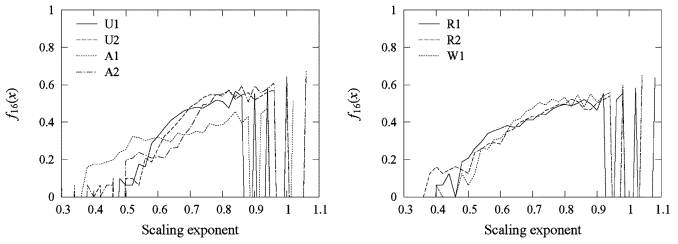

Fig. 21. Multifractal spectra for all data sets, $p = 16$. (See Section VI-C.)

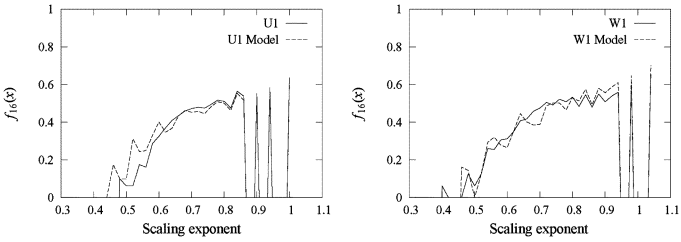

Fig. 22. Multifractal spectra for U1 and its model, and for W1 and its model, $p = 16$. (See Section VI-C.)

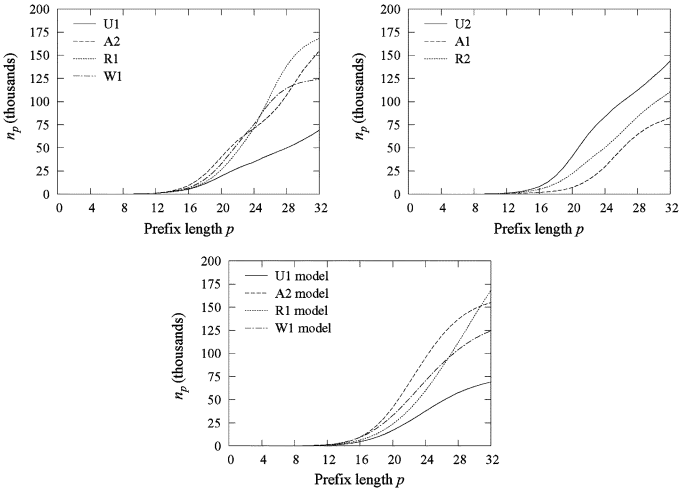

Fig. 23. Aggregate counts n_p for all data sets, and for models of U1, A2, R1, and W1. (See Section VII-A. Note: the Y axis is not log scale.)

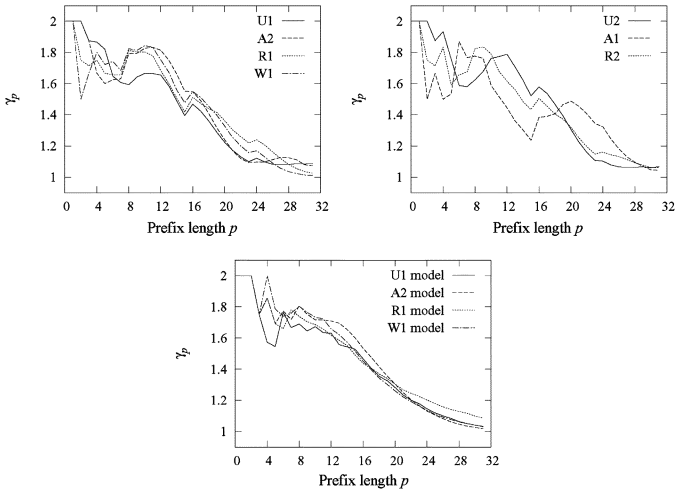

Fig. 24. γ_p for all data sets, and for models of U1, A2, R1, and W1. (See Section VII-A.)

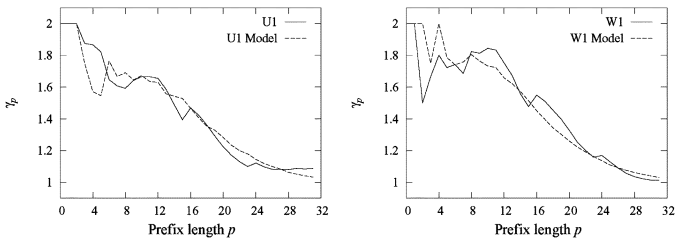

Fig. 25. γ_p for U1 and its model, and for W1 and its model. (See Section VII-A.)

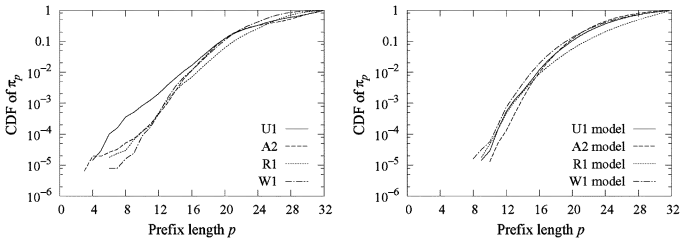

Fig. 26. CDFs of discriminating prefix counts π_p for U1, A2, R1, and W1 and their models. (See Section VII-B.)

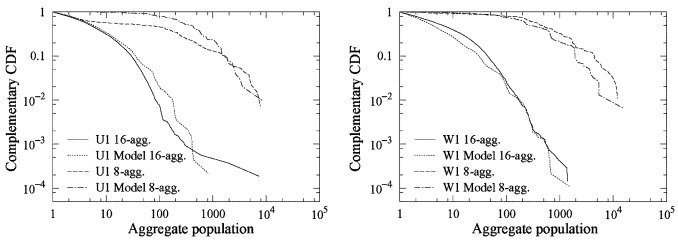

Fig. 27. 8- and 16-aggregate population distributions for U1 and W1 and their models. (See Section VII-C.)

the traces contained no per-flow data. Fits to the upper tails of these curves yield values around 1 for α, the power in a power-law distribution. However, not all distributions seem strongly heavy-tailed; see the lines for A1, for example.

ACKNOWLEDGMENT

The authors are deeply grateful to D. Donoho for his comments, guidance, and generosity; he led them, for example, to the multifractal model. D. Karp was also a generous and thoughtful collaborator. The authors thank W. Willinger, C. Blake, R. Morris, S. Floyd, and several anonymous reviewers for comments on previous drafts. The authors are also very grateful to the contributors of the traces used in this study, who cannot be explicitly identified because the traces must remain anonymous.

REFERENCES

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[2] S. McCreary and K. Claffy, "Trends in wide area IP traffic patterns: A view from Ames Internet Exchange," presented at the ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management, Monterey, CA, Sep. 2000.

[3] K. Thompson, G. Miller, and R. Wilder, "Wide area Internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10–23, Nov. 1997.

[4] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft, "POP-level and access-link-level traffic dynamics in a tier-1 POP," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001.

[5] P. Danzig, S. Jamin, R. Cáceres, D. Mitzel, and D. Estrin, "An empirical workload model for driving wide-area TCP/IP network simulations," *Internetworking: Research and Experience*, vol. 3, no. 1, pp. 1–26, 1992.

[6] V. Paxson, "Growth trends in wide-area TCP connections," *IEEE Network*, vol. 8, no. 4, pp. 8–17, Jul. 1994.

[7] W. Willinger, V. Paxson, and M. S. Taqqu, "Self-similarity and heavy tails: structural modeling of network traffic," in *A Practical Guide to Heavy Tails*. New York: Chapman & Hall, 1998, ch. 1, pp. 27–53.

[8] A. Feldmann, A. C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic," in *Proc. ACM SIGCOMM'98*, Oct. 1998, pp. 42–55.

[9] B. Krishnamurthy and J. Wang, "On network-aware clustering of Web clients," in *Proc. ACM SIGCOMM 2000*, Aug. 2000.

[10] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *ACM Comput. Commun. Rev.*, vol. 32, no. 3, Jul. 2002.

[11] J. Postel, Ed., "Internet Protocol," Internet Engineering Task Force, RFC 791, Sep. 1981 [Online]. Available: ftp://ftp.ietf.org/rfc/rfc0791.txt

[12] V. Fuller, T. Li, J. Yu, and K. Varadhan, "Classless Inter-Domain Routing (CIDR): An Address Management and Aggregation Strategy," Internet Engineering Task Force, RFC 1519, Sep. 1993 [Online]. Available: ftp://ftp.ietf.org/rfc/rfc1519.txt

[13] G. Minshall, Tcpdpriv: Program for Eliminating Confidential Information From Traces. 1997 [Online]. Available: http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html

[14] K. Sigman, "A primer on heavy-tailed distributions," *Queueing Syst.*, vol. 33, no. 1–3, pp. 261–275, Dec. 1999.

[15] M. E. Crovella, M. S. Taqqu, and A. Bestavros, "Heavy-tailed probability distributions in the World Wide Web," in *A Practical Guide to Heavy Tails*. New York: Chapman & Hall, 1998, ch. 1, pp. 3–26.

[16] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. ACM SIGCOMM'99*, Aug. 1999, pp. 251–262.

[17] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore, "On the bias of traceroute sampling: or, power-law degree distributions in regular graphs," in *Proc. 37th ACM Symp. Theory of Computing*, May 2005, pp. 694–703.

[18] B. B. Mandelbrot, *Fractals, Form, Chance and Dimension*. San Francisco, CA: W. H. Freeman, 1977.

[19] H. O. Peitgen, H. Jurgens, and D. Saupe, *Chaos and Fractals*. New York: Springer-Verlag, 1992.

[20] D. Harte, *Multifractals: Theory and Applications*. New York: Chapman Hall/CRC, 2001.

[21] R. H. Riedi, "Introduction to Multifractals," Rice Univ., Houston, TX, Tech. Rep., Oct. 1999.

[22] B. Halabi, *Internet Routing Architectures*. Indianapolis, IN: Cisco, 1997.

[23] CAIDA Analysis of Code-Red. Cooperative Association for Internet Data Analysis (CAIDA), La Jolla, CA, 2001 [Online]. Available: http://www.caida.org/analysis/security/code-red/

Eddie Kohler received S.B. degrees in mathematics with computer science and in music, and S.M. and Ph.D. degrees in electrical engineering and computer science, all from the Massachusetts Institute of Technology, Cambridge.

He is an Assistant Professor of computer science at the University of California, Los Angeles, working largely in operating systems and sensor networks. He is also co-founder and Chief Scientist of Mazu Networks, a network security company.

Dr. Kohler has been a member of the ACM since 1999.

Jinyang Li (M'06) received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge.

She works as an Assistant Professor in computer science at New York University, where she leads the Networks and Wide-area Distributed Systems group. She is currently working on distributed storage systems and wireless mesh networks.

Dr. Li has been a member of the ACM since 1999.

Vern Paxson (M'05) received the B.S. degree from Stanford University, Stanford, CA, and the M.S. and Ph.D. degrees from the University of California at Berkeley.

He is a Senior Scientist at the International Computer Science Institute (ICSI) in Berkeley, CA, and a Staff Scientist with the Lawrence Berkeley National Laboratory. His main active research projects are network intrusion detection in the context of Bro, a high-performance network intrusion detection system he developed; large-scale network measurement and analysis; and Internet-scale attacks, particularly rapidly propagating network "worms".

Dr. Paxson co-founded the ACM/USENIX Internet Measurement Conference, served on the editorial board of IEEE/ACM TRANSACTIONS ON NETWORKING, chaired the Internet Research Task Force, and is presently vice-chair of ACM SIGCOMM. He co-chaired the program committee of the IEEE Symposium on Security and Privacy in 2005 and 2006, and is a two-time recipient of the IEEE Communications Society William R. Bennett Prize Paper Award. He has been a member of the ACM since 1989.

Scott Shenker (M'87–SM'96–F'00) spent his academic youth studying theoretical physics but soon gave up chaos theory for computer science. Continuing to display a remarkably short attention span, his research over the years has wandered from Internet architecture and computer performance modeling to game theory and economics. He currently splits his time between the UC Berkeley Computer Science Department and the International Computer Science Institute (ICSI).