Observed Structure of Addresses in IP Traffic

Eddie Kohler* Jinyang Li† Vern Paxson* Scott Shenker*

*ICSI Center for Internet Research †MIT Lab for Computer Science
kohler@icir.org, jinyang@lcs.mit.edu, {vern, shenker}@icir.org

Abstract

This paper considers the structure of addresses contained in IP traffic. Specifically, we investigate the structural characteristics of destination IP addresses seen on Internet links, considered as a subset of the address space. These characteristics may have implications for algorithms that deal with IP address aggregates, such as routing lookups and aggregate-based congestion control. We find that address structures match a constructive multifractal model with two parameters, which may be useful for simulations where realistic IP addresses are preferred. We also develop concise characterizations of that structure, including *active aggregate counts* and *discriminating prefixes.* We find that for a given site, our structural characterizations are stable over short time scales, 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".

1 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 such "micro-flows" and their combined behavior when aggregated grows ever wider. To date, studies of aggregate traffic have focused on questions of behavior at a particular level of granularity: for example, correlations in packet arrivals seen en masse on a link [12], patterns of backbone traffic when partitioned by directionality, transport protocol, and application [21, 15] or viewed at /8, /16 and /24 prefix granularities [2], or the overall distributions of individual connection characteristics [5, 17]. These studies have made significant progress in understanding the structure of specific types of aggregates, but the question of how behavior changes *as aggregation increases* has received little attention beyond basic statistical multiplexing models. Yet there is clearly a world of difference between an individual TCP connection and a Gbps backbone stream from one city to another.

Ultimately, we would like to build towards a theory of traffic aggregation. For example, what do we get when we merge together two already-large conglomerates, say for traffic engineering purposes? The work described here is modest in scope compared with this goal. We look at one of the simplest conglomerate properties we could investigate: how a conglomerate's packets are distributed among its component addresses, and how those addresses aggregate. However, these properties form an important part of any model of the routing behavior of conglomerates in the network; and it turns out that even these simple properties exhibit surprisingly rich structure.

The paper body begins with descriptions of our methodology and data sets (Sections 3 and 4). We then examine the factors that give rise to an interesting property of aggregates, namely that the distribution of packets per destination prefix aggregate has a heavy, Pareto-like tail (Section 5). This is related to the well-known "mice and elephants" phenomenon, whereby some flows contain vastly more packets than others. By applying different types of random shuffling, however, we show that *address structure*—the arrangement of active addresses in the address space—has a greater effect on aggregate packet counts than the per-flow packet distribution, at least for medium-to-large aggregates such as /16s. This motivates our investigation of address structure, since we must understand it before we can understand the independently important property of aggregate packet counts.

When examined spatially, as in Figure 2, the structure of the set of addresses in a trace appears broadly self-similar: some structural features reappear at different scales. We therefore explore fractal address models in Section 6. It turns out that real address structures may usefully be analyzed using 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 as a means for accurately synthesizing address structures for simulation purposes, and for providing an analytic framework for further exploring aggregation properties. This model is the core result of the paper.

In Section 7, we further explore our data sets and our model using concepts and analytic tools designed for analyzing address structures. We finish in Section 8 with a look at how address structure properties vary: over time, from site to site, and for different types of traffic. We find

Trace	**Description**	**Time (hr)**	$\boldsymbol{N}$	**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	1/256
R2	Link from a regional ISP	2.0	110,783	1,992,318	1/256
W1	Access link in front of a large Web server	∼ 2.0	124,454	5,000,000	no

Figure 1—Characteristics of our traces.

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.

The appendix presents supplementary graphs using additional data sets and parameters.

2 Related Work

We are 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; measured traffic appears to be self similiar [22, 12] and exhibit multifractal characteristics [7]. Attempts have also been made to model other aspects of the Internet, such as the the power law relationship of the Internet topology [6]. Krishnamurthy and Wang [11] have previously investigated the properties of client addresses aggregated according to BGP routing prefixes. Their results indicate that client cluster size has a heavy-tailed distribution. Recently, researchers have started to investigate IP address prefix based aggregate properties for aggregate congestion control [13].

3 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 [19]. Classless inter-domain routing [8], 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.

- 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 route cache strategies.

- Other router algorithms that work on aggregates, such as aggregate-based congestion control [13], often define aggregates by destination prefix, since routers already use them for route lookup.

- Address 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.

Nevertheless, one could usefully define aggregates in many other ways, such as by destination geographic area or application protocol.

We use CIDR notation for prefixes and aggregates. Given an IP address a and prefix length p, with $0 \le p \le 32$, "a/p" refers to the p-bit prefix of a or, equivalently, the aggregate containing 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.

4 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 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; 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. Figure 1 presents high-level characteristics of these data sets.

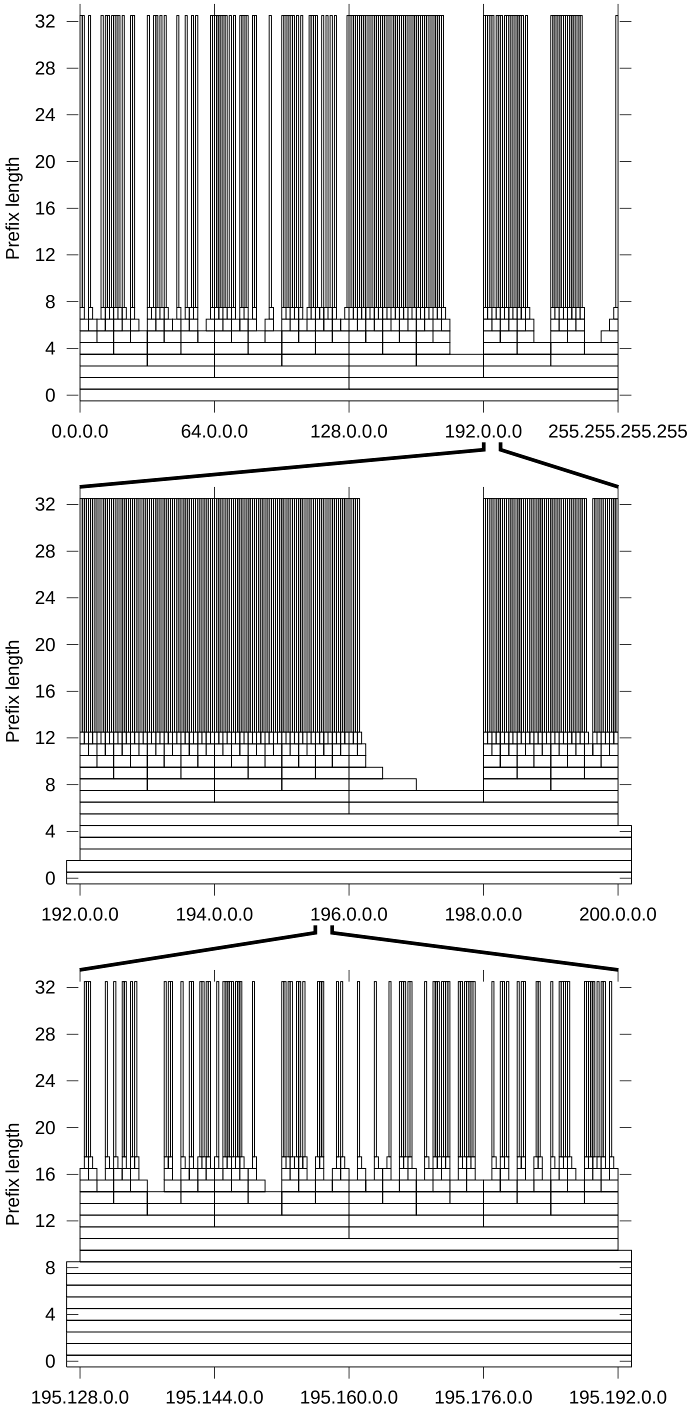

Figure 2—The address structure of data set U1, with two successive 32× magnifications. We draw a box for every nonempty address prefix; the Y axis is prefix length. A single address would generate a stack of 33 boxes, each half the width of the one below. The topmost boxes are extremely thin!

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	5,785

Figure 3—Characteristics of trace R1.

Many of our traces have been anonymized as if by *tcpdpriv –A50* [16]. This applies an anonymization function f to every IP address in the trace. The function preserves prefix relationships, so given addresses a and b and any prefix length p, $a/p = b/p$ iff $f(a)/p = f(b)/p$. All our analysis methodologies are indifferent to this kind of anonymization.

All of the traces are omnidirectional. That is, each trace contains all packets passing by the trace location, regardless of whether the packets were heading "towards" or "away from" the trace point. This choice was mandated by the anonymization of some of our traces. However, we experimented with algorithms to extract likely unidirectional traces from omnidirectional ones. On seeing a packet with source address a and destination address b, we 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 structural metrics (see Section 8) 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. Still, we examine only destination addresses.

Figure 2 shows the destination addresses present in trace U1. We draw a box for each aggregate containing at least one address present in the trace. Other traces look generally similar.

5 Importance of Address Structure

We now turn to the distributions of the number of packets per TCP/UDP flow, destination address, and destination address aggregate for trace R1. These packet count distributions are significant for congestion control and fairness applications, for example. We see that all three distributions are heavy-tailed, and demonstrate that address structure is the most important factor affecting aggregate packet count distributions for medium-sized aggregates.

Figure 3 summarizes relevant characteristics of trace R1.

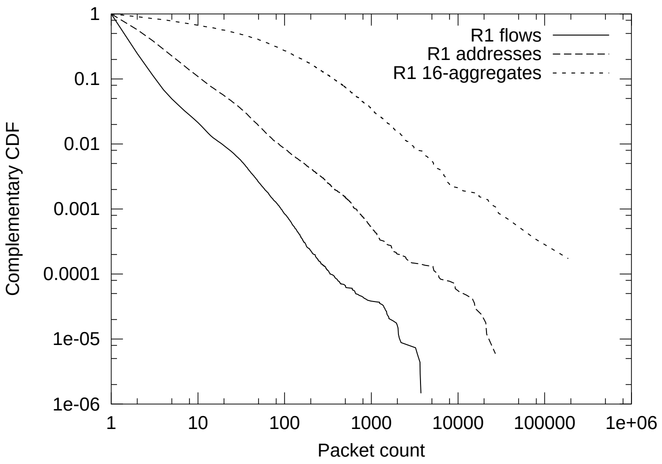

Figure 4—Log-log complementary CDF of packet counts for R1 flows, addresses, and 16-aggregates. All are consistent with power-law distributions. The fit lines have slopes −1.46, −1.16, and −1.13, respectively.

5.1 Packet count distributions

Log-log complementary CDF graphs form a well-known test for heavy-tailed, or power-law tail, 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 for sufficiently large x.

Figure 4 presents a log-log complementary CDF of the packet counts of TCP/UDP flows, addresses, and 16-aggregates in the R1 data set. 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 3,727 sampled packets, 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 $0 < \alpha \leq 2$; 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 points. Other traces have similar packet count distributions, although some have less heavy tails.

We might have expected TCP/UDP flow packet counts to appear heavy-tailed, as they in fact do. Prior work has shown that Web flow weights follow a heavy-tailed distribution [4], and 70% of R1's packets, and 89% of its flows, use ports 80 (http) or 443 (https). 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; the sum of so many finite distributions would tend to converge, however slowly, to a normal distribution. This is not what we see in Figure 4. Why does the 16-aggregate packet count distribution appear, if anything, *more* heavy-tailed than the flow packet count distribution?

5.2 Factors affecting aggregate packet counts

Conceptually, aggregate packet counts depend 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, address structure measures where packets are headed without differentiating 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?

Obviously, the per-address packet count distribution will dominate the packet counts of small aggregates. A 30-aggregate, for example, can contain at most four addresses, so address structure and correlation have minimal impact on aggregate packet count. But what about medium-to-large aggregates, such as /16s?

We can determine the relative importance of the three factors by altering each factor in turn, then comparing the resulting aggregate packet count distributions with those of the real data R1.

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 keeping address structure the same. (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.

Figure 5 shows the results for 16-aggregates. 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 address packet counts.[1]

[1] For 20-aggregates and smaller, "random counts" matches less well than "random addresses"—at first for the largest aggregates, then eventually, with increasing prefix length, for almost all aggregates.

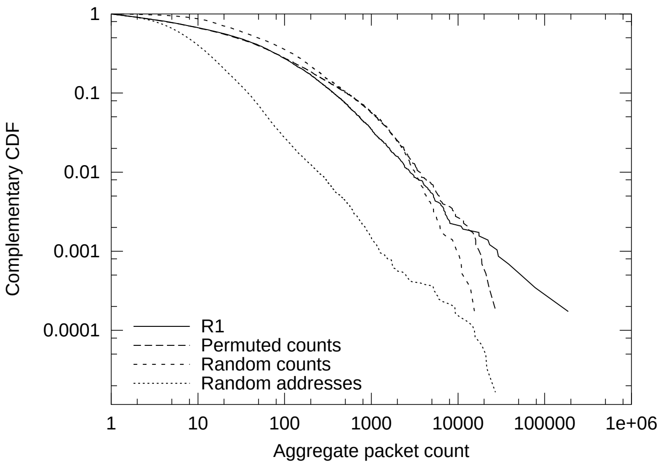

Figure 5—Complementary CDF of 16-aggregate packet counts for R1, R1 with random address packet counts, R1 with random addresses, and R1 with permuted address packet counts (but the same addresses).

6 Multifractal Model

Figure 2 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. Of course, true fractals have structure down to infinitely small scales, while addresses bottom out at prefix length 32. Nevertheless, this is enough depth to make fractal models potentially valuable.

6.1 Fractal dimension

An address structure can be viewed as a subset of the unit interval $I = [0, 1)$, where the subinterval $A_a = [a/2^{32}, (a+1)/2^{32})$ corresponds to address a. Considered this way, address structure might resemble a Cantor dust-like fractal [14, 18]. Cantor dusts have fractal dimension between 0 and 1. What would be the dimension of our address structure?

The *lattice box counting* fractal dimension metric naturally fits with address structures and prefix aggregation. Lattice box counting dimension measures, for every p, the number of dyadic intervals of length 2^{-p} required to cover the relevant dust. These dyadic intervals correspond exactly to our p-aggregates.

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 ($0 \le p \le 32$). 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 $n_p \le n_{p+1} \le 2n_p$. Using this notation, lattice box counting dimension is defined as

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

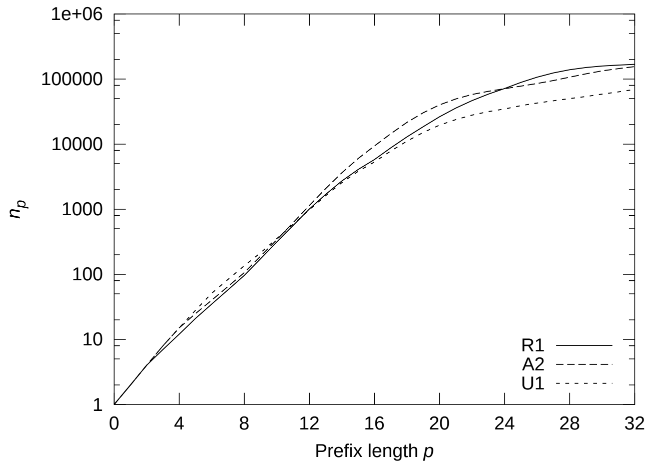

Figure 6—n_p as a function of prefix length for several traces, with a least-squares fit line for R1's $4 \le p \le 14$ region (fit slope 0.79).

In other words, 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). Figure 6 shows a log plot of n_p as a function of p; we find that, for a reasonable middle region $4 \le p \le 14$, 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 $D = 0.79$.

6.2 Multifractality

Adaptations of the well-known Cantor dust construction can generate address structures with any fractal dimension. Starting with the unit interval, one repeatedly removes the middle portion of all subintervals. The relative size h of the removed portion determines the Hausdorff dimension of the resulting set:

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

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

Such Cantor dusts can capture the global scaling behavior of aggregate counts. However, real address structure is more complicated than what they can predict. Dusts 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 Figure 2. This results in different local scaling behavior, 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* [18]. Let $\mu_p(a)$ denote the "mass" associated with the dyadic interval of length 2^{-p} containing a. For us, this is the probability that a randomly chosen

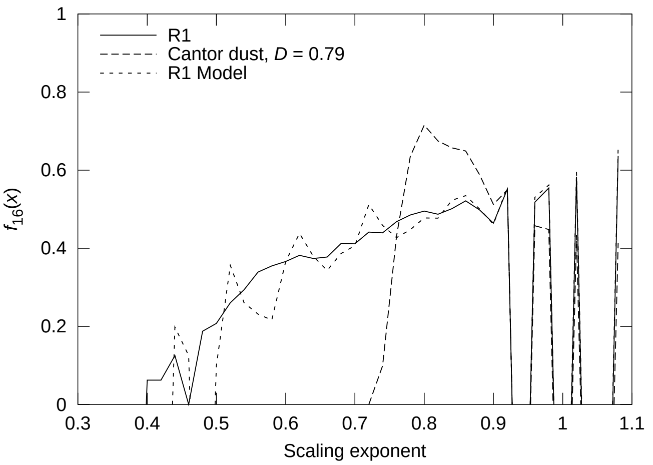

Figure 7—Multifractal spectra for R1 and Cantor dusts, $p = 16$.

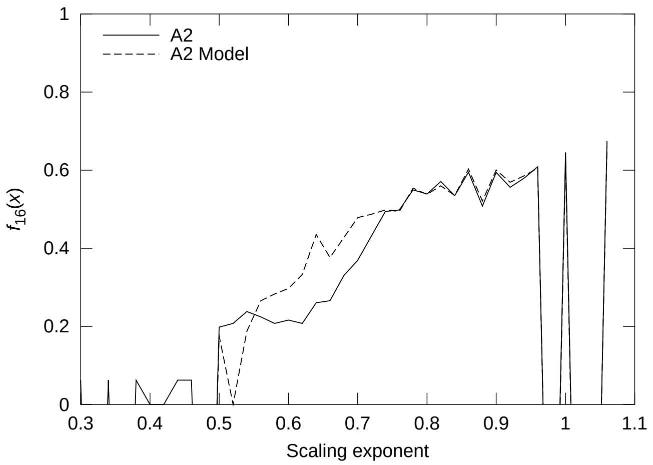

Figure 8—Multifractal spectra for A2 and its model, $p = 16$.

active IP address is contained in the aggregate a/p. Let $\sigma_p(a)$ denote the number of active addresses in the aggregate a/p; then $\mu_p(a) = \sigma_p(a)/N$. When $\mu_p(a) > 0$, the *local scaling exponent* $\alpha_p(a)$ is defined as follows:

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

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.[2] 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 Figure 7 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 dust with fractal dimension 0.79, the same as R1's nominal fractal dimension. 168,318 addresses were sampled, giving the dust the same number of addresses as R1. The resulting structure's multifractal spectrum is narrow compared to that of R1.

6.3 Model

The original Cantor construction can be easily extended to a multifractal Cantor measure [10, 20]. 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 $\mu_k(I_{\varepsilon_1 \dots \varepsilon_k}) = m_{\varepsilon_1} \times \dots \times m_{\varepsilon_k}$ (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 a number of addresses so that 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 Figure 7 shows its multifractal spectrum. 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. Figure 8 shows the multifractal spectra for A2 and its model at $p = 16$.

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

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

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.

6.4 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.

7 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 describe 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 characterizations: *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.

7.1 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 6.1, 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 address visible in the trace as a destination. A model of active aggregate counts might affect the design of algorithms that keep track of aggregates by showing 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. Figure 9 shows the values of γ_p for R1, A2, and our multifractal model tuned for R1; Figure 10 additionally shows the model for A2. 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 in γ_p years after the introduction of CIDR [9].

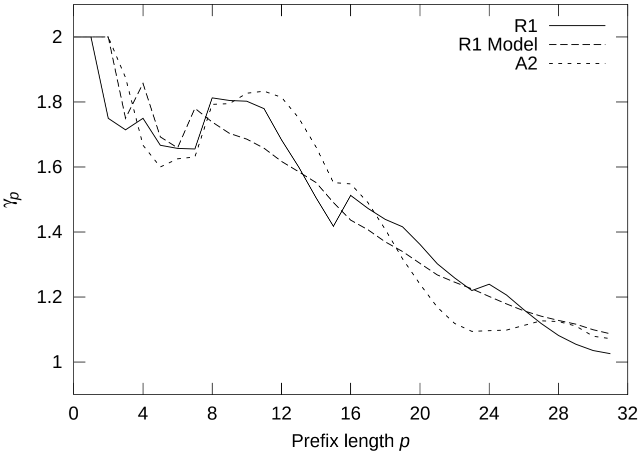

Figure 9—γ_p, aggregation ratios.

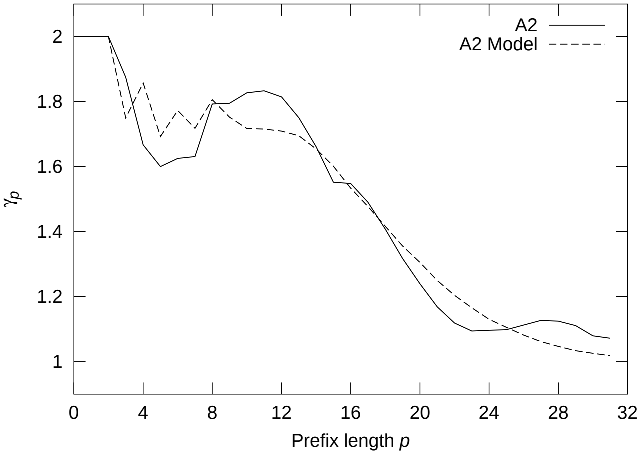

Figure 10—γ_p for A2 and its model.

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.[3]

[3] We note that 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 com-

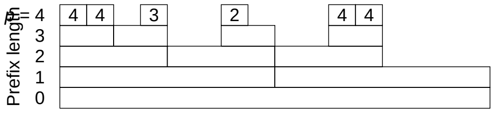

Figure 11—Discriminating prefix example with 4-bit addresses. The top boxes are active addresses; lower boxes represent active aggregates, as in Figure 2. Each active address's discriminating prefix is shown inside its box.

7.2 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, so all addresses are equally separated, and cases where some 16-aggregates are fully populated and others are sparsely populated, so 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. Figure 11 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'd expect aggregates to contain small numbers of active addresses.

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, $\sum \pi_p = N$.

Figure 12 graphs π_p for R1, A2, and our R1 model. 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.

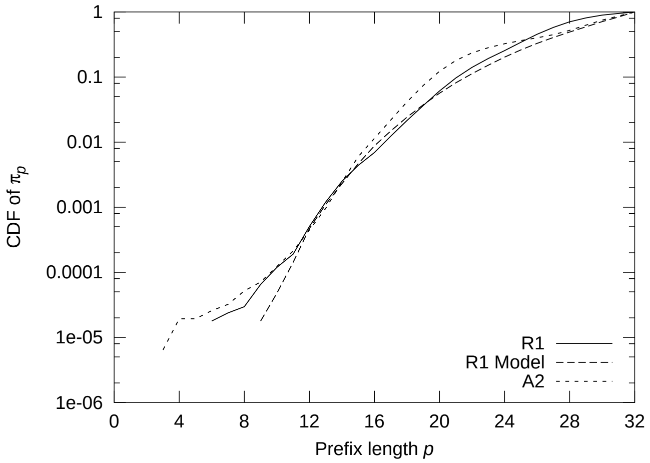

Figure 12—CDF of address discriminating prefix counts π_p.

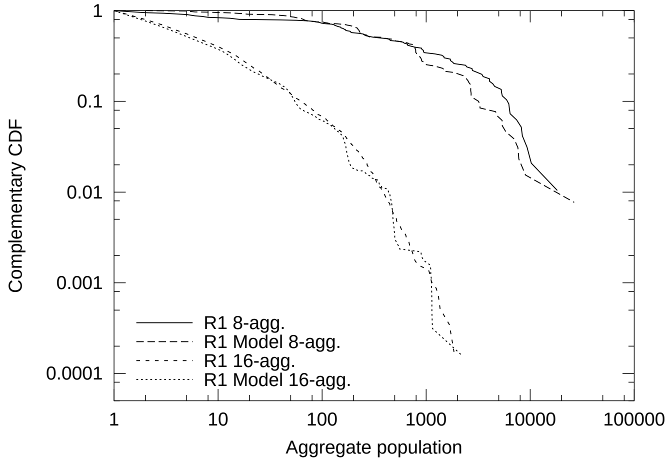

Figure 13—8- and 16-aggregate population distributions for R1.

7.3 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 6.2, we expressed this as $\sigma_p(a)$.) All p-aggregates might have similar populations, meaning addresses are spread evenly among the active aggregates. Given our experience with the other metrics, however, 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 isn't as much room for variability.)

Figure 13 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 5.1.) As expected, aggregates exhibit a wide range of populations. The multifractal model echoes the real data, particularly in the tail region.

munication among A2's customers. Intuitively, such a communication pattern—for example, if several of A2's customers were different campuses of a single organization—might correlate with closely-packed active addresses.

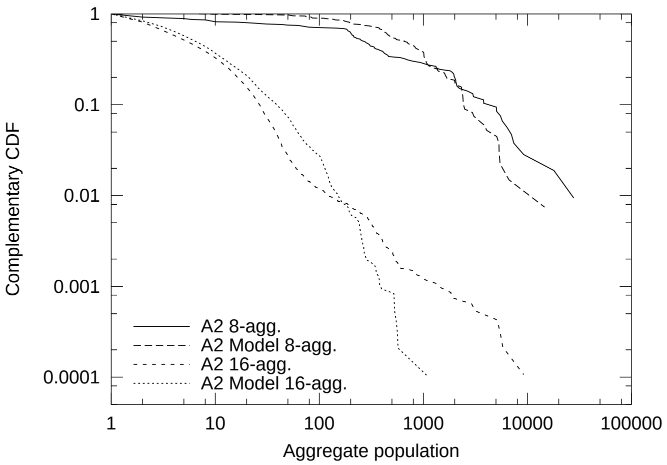

Figure 14—8- and 16-aggregate population distributions for A2.

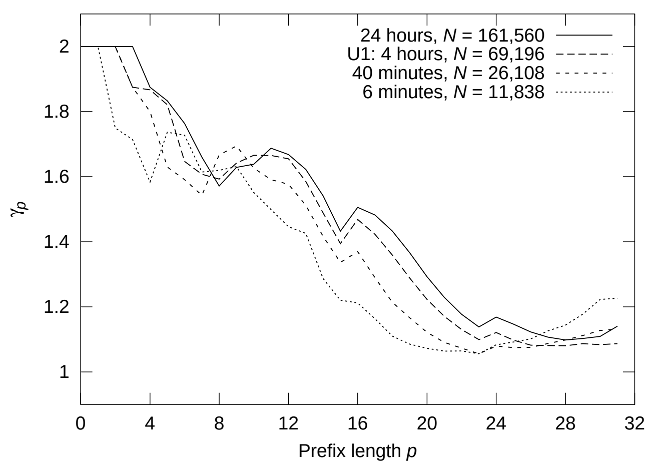

Figure 15—γ_p for U1, and for longer and shorter traces from the same data.

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. Figure 14 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. 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 Figure 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 $p > 27$.

8 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 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?

8.1 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. N, then, 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?

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 the same underlying probability distribution, assuming the distribution didn't change significantly over the 24-hour period.[4]

Figure 15 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.[5] The other curves, however, resemble one another, and differ visibly from other data sets. (Compare Figure 9, for example.)

8.2 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, A1, and A2 into sequential nonoverlapping segments, each containing 32,768 addresses. That is, we process the

[4] The distribution almost certainly does change but, as Figure 15 shows, not enough to affect the argument.

[5] 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.

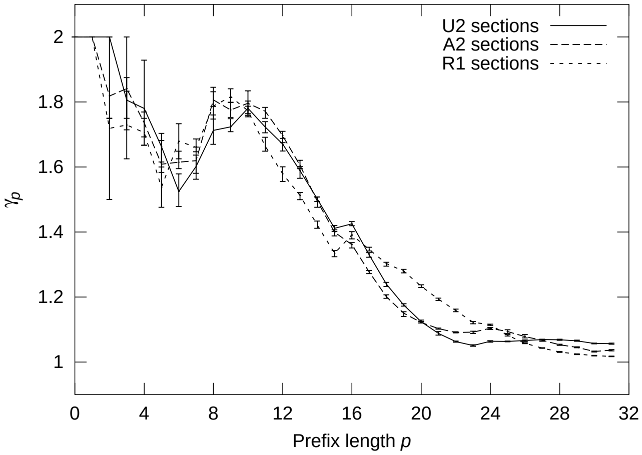

Figure 16—Variations of γ_p over time for different traces. The error bars indicate the range of variations of γ_p.

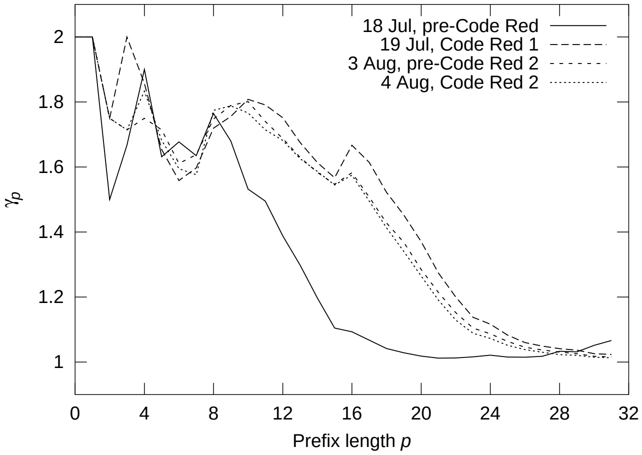

Figure 17—γ_p for external addresses before and after Code Red 1 and 2.

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, 5.5 minutes for A1, and 7.5 minutes for A2. We would like sections from the same trace to resemble one another, and to differ from sections 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, Figure 16 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.

8.3 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 [3] 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.

We would expect this behavior to completely change address structure observable at the edge of the Internet. 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 = 2,332); 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. Figure 17 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 Figure 18, changes as well; it drops, since many aggregates have been added that contain only unsuccessful probes. Figure 18 may also demonstrate a distinction between Code Red 1 and Code Red 2. There are more medium-sized aggregates,

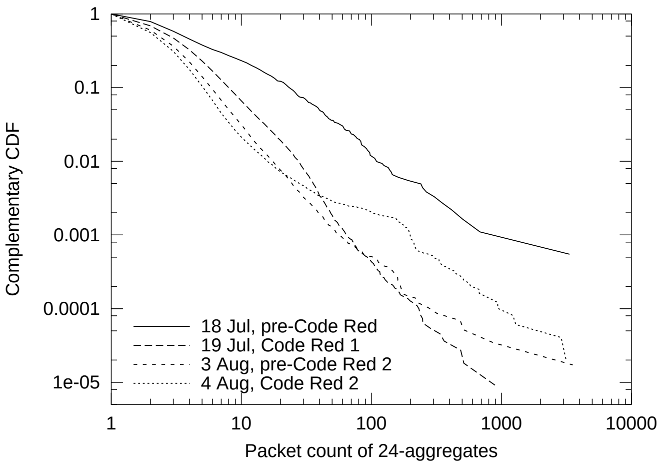

Figure 18—Aggregate packet count distribution for 24-aggregates before and after Code Red 1 and 2.

perhaps because Code Red 2's locality means that networks near the lab in IP space tend to probe it more often.

9 Conclusion

This paper demonstrates that address structure is key to understanding interesting properties of large aggregates, such as their packet count distributions. We presented a multifractal model of observed addresses, and showed that it well models many properties of the address structures we collected. We developed specific structural characterizations to examine how addresses aggregate at different levels. Finally, we demonstrated that address structure differs between sites, yet is relatively insensitive to sample size and stable over short time scales.

Acknowledgments

We are deeply grateful to David Donoho for his comments, guidance, and generosity; he led us, for example, to the multifractal model. Dick Karp was also a generous and thoughtful collaborator. We thank Walter Willinger, Chuck Blake, Robert Morris, and several anonymous reviewers for comments on previous drafts.

References

[1] Chapman & Hall, New York, 1998.

[2] Supratik Bhattacharyya, Christophe Diot, Jorjeta Jetcheva, and Nina Taft. POP-level and access-link-level traffic dynamics in a tier-1 POP. In *ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[3] CAIDA. CAIDA analysis of Code-Red, 2001. `http://www.caida.org/analysis/security/code-red/`.

[4] 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* [1], chapter 1, pages 3–26.

[5] P. Danzig, S. Jamin, R. Cáceres, D. Mitzel, and D. Estrin. An empirical workload model for driving wide-area TCP/IP network simulations. In *Internetworking: Research and Experience*, volume 3, pages 1–26, 1992.

[6] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *Proc. ACM SIGCOMM '99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 251–262, August 1999.

[7] 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 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 42–55, October 1998.

[8] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR): An address management and aggregation strategy. RFC 1519, Internet Engineering Task Force, September 1993. `ftp://ftp.ietf.org/rfc/rfc1519.txt`.

[9] B. Halabi. *Internet Routing Architectures.* Cisco, 1997.

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

[11] Balachander Krishnamurthy and Jia Wang. On network-aware clustering of Web clients. In *Proc. ACM SIGCOMM 2000 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, August 2000.

[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic (extended version). In *IEEE/ACM Trans. on Networking*, pages pp. 1–15, 1994.

[13] Ratul Manajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker. Controlling high bandwidth aggregates in the network. *ACM Computer Communication Review*, to appear.

[14] B. B. Mandelbrot. *Fractals, Form, Chance and Dimension.* San Francisco, CA, 1977.

[15] Sean McCreary and k claffy. Trends in wide area IP traffic patterns: A view from Ames Internet Exchange. In *ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management*, September 2000.

[16] Greg Minshall. Tcpdpriv: Program for eliminating confidential information from traces, 1997. `http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html`.

[17] V. Paxson. Growth trends in wide-area TCP connections. *IEEE Network*, 8(4):8–17, July 1994.

[18] H. O. Peitgen, H. Jurgens, and D. Saupe. *Chaos and Fractals.* Springer-Verlag, 1992.

[19] J. Postel (editor). Internet Protocol. RFC 791, Internet Engineering Task Force, September 1981. `ftp://ftp.ietf.org/rfc/rfc0791.txt`.

[20] Rudolf H. Riedi. Introduction to multifractals. Technical report, Rice University, October 1999.

[21] K. Thompson, G. Miller, and R. Wilder. Wide area Internet traffic patterns and characteristics. In *IEEE Network*, volume 11, pages 10–23, November 1997.

[22] 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* [1], chapter 1, pages 27–53.

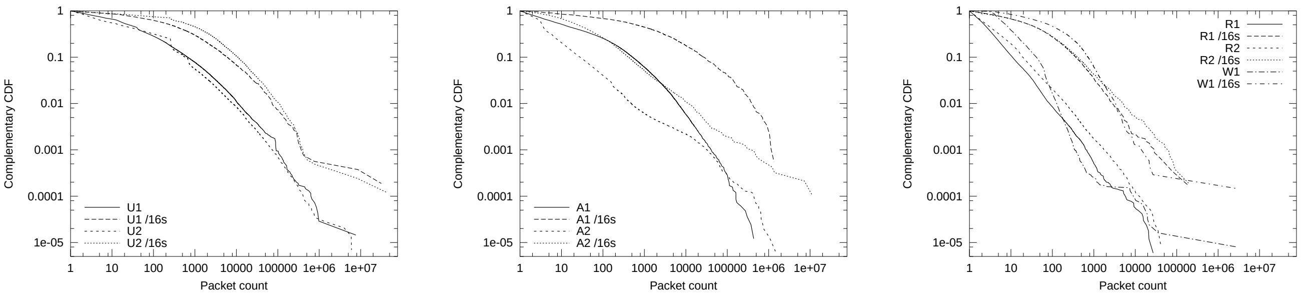

Figure 19—Log-log complementary CDFs of packet counts for addresses and 16-aggregates in all traces.

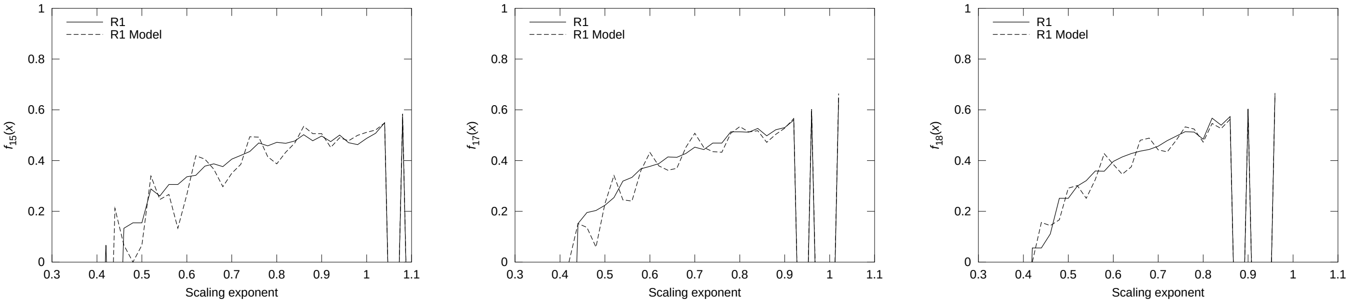

Figure 20—Multifractal spectra for R1 and its model, $p = 15$, 17, and 18.

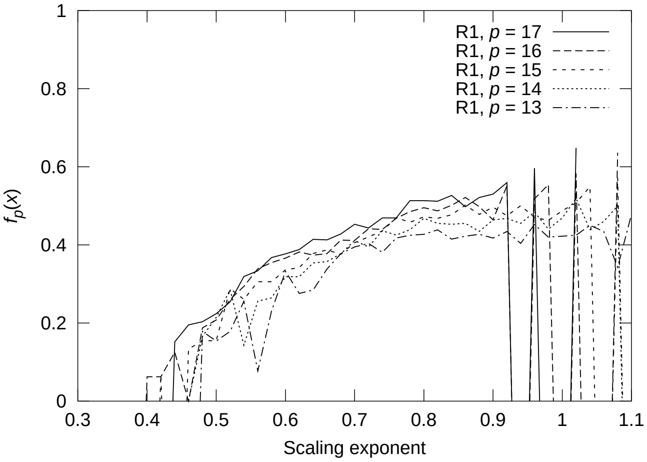

Figure 21—Multifractal spectra for R1 at prefix levels $13 \leq p \leq 17$.

Appendix

U1: -.92, /16: -0.96 U2: -.93, /16: -1.1 A1: -1.07, /16: -0.91 A2: -0.64, /16: -0.70 R1: -1.16, /16: -1.13 R2: -1.07, /16: -0.92 W1: -1.43, /16: -1.60

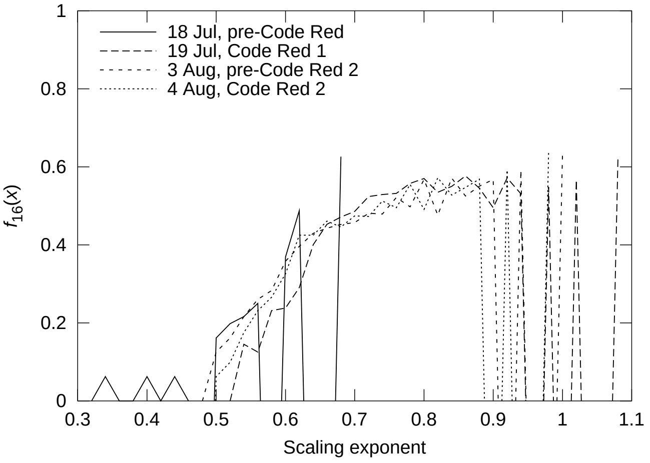

Figure 27—Multifractal spectra at a national laboratory before and after Code Red 1 and 2, $p = 16$.

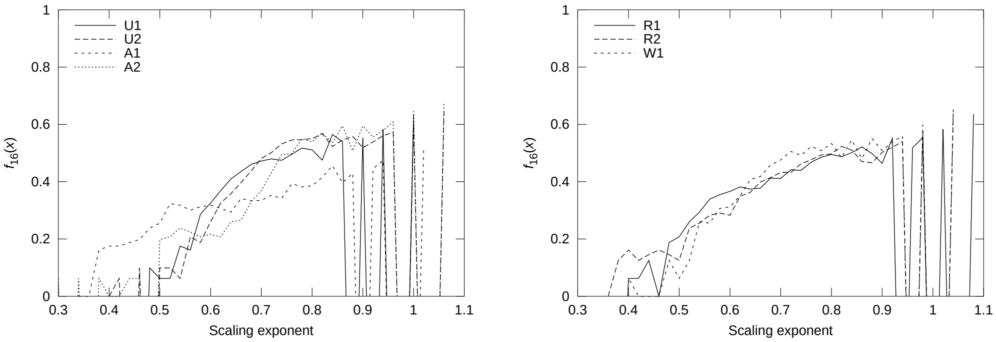

Figure 22—Multifractal spectra for all data sets, $p = 16$.

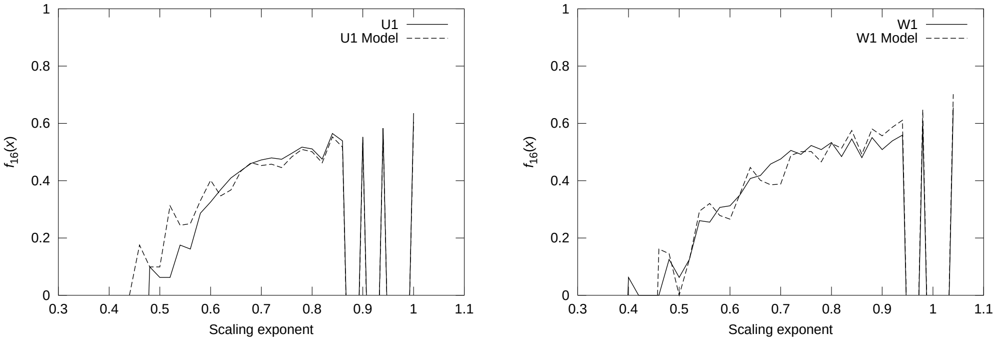

Figure 23—Multifractal spectra for U1 and its model, and for W1 and its model, $p = 16$.

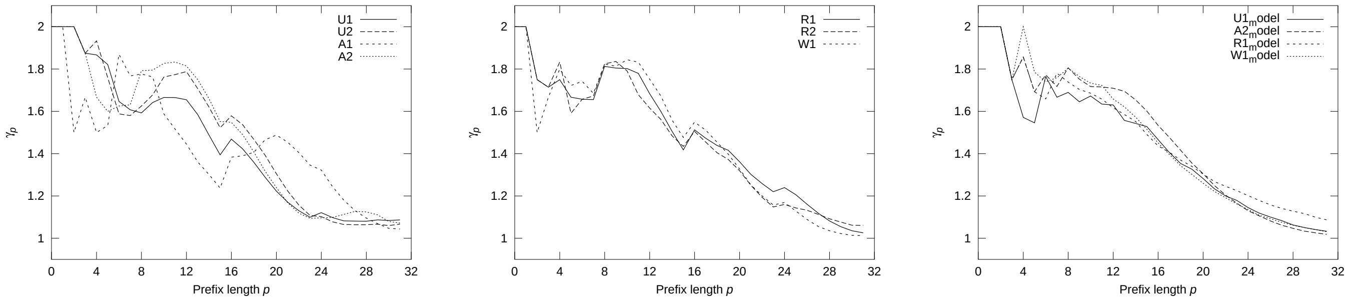

Figure 24—γ_p for all data sets, and for models of U1, A2, R1, and W1.

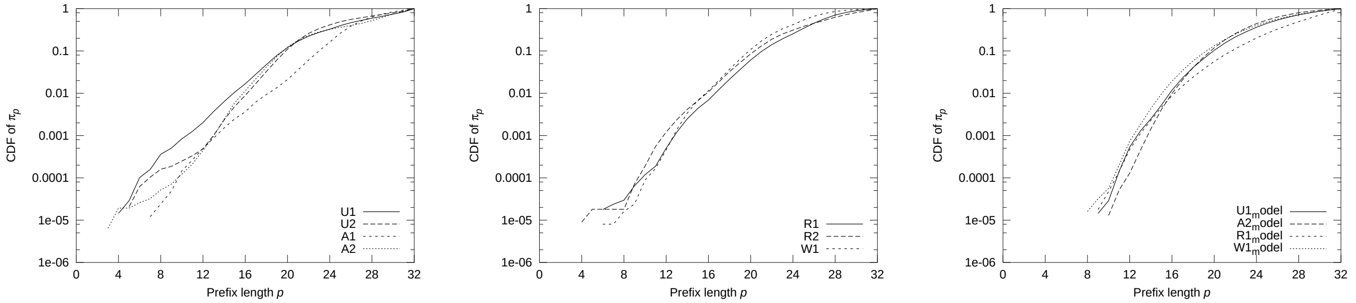

Figure 25—CDF of discriminating prefix counts π_p for all data sets, and for models of U1, A2, R1, and W1.

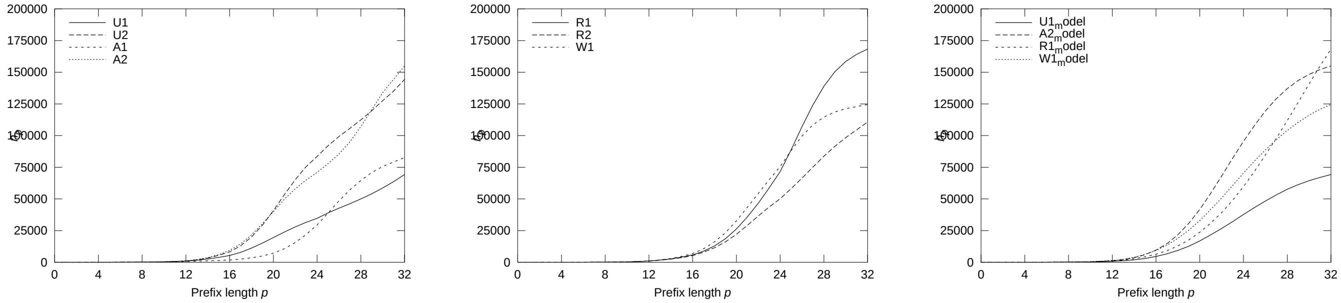

Figure 26—Aggregate counts n_p for all data sets, and for models of U1, A2, R1, and W1.

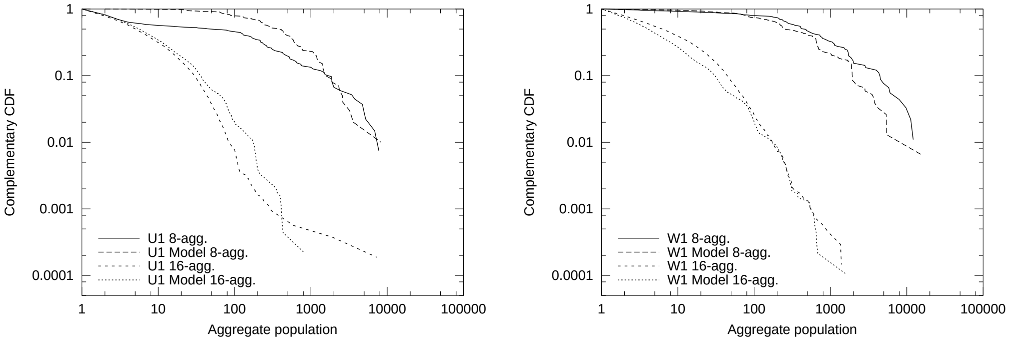

Figure 28—8- and 16-aggregate population distributions for U1 and W1.

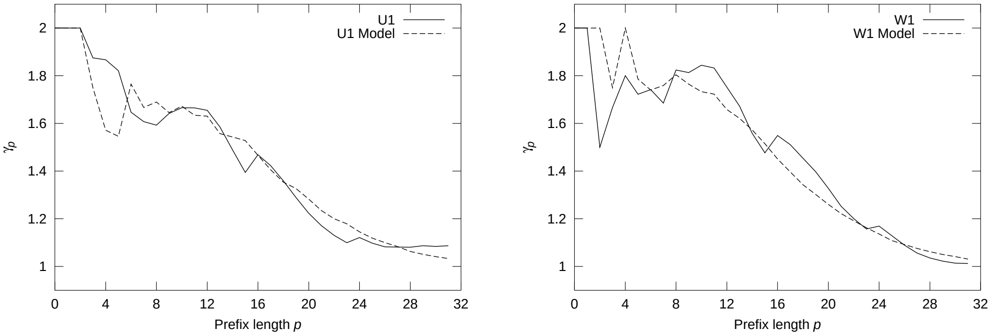

Figure 29—γ_p for U1 and its model, and for W1 and its model.