Confidence-Weighted Marginal Utility Analyses for Improved Internet Mapping

Craig Prince and Danny Wyatt

December 10, 2004

Abstract

Many techniques have been proposed for discerning internet topology, but not so many of them have been studied for their utility: the returns in information that they provide as their number of measurements increases. We define a probabilistic model for representing a map built on uncertain observations. We propose new utility metrics and reuse existing metrics in combination with this model to examine the marginal utility of observations for maps made with three different tools: Skitter, Scriptroute, and Rocketfuel. We define classes of observations and quantify their relative utilities and discuss how these results can influence future mapping techniques.

1 Introduction

At a high level, internet mapping is a simple concept: create an accurate, complete map of all or part of the Internet. Some mapping attempts seek to find the internet's physical topology of routers and cables between them, while other attempts seek to find logical interfaces and the virtual links between these interfaces. In both cases the goal is the same: to create a graph representing the interconnections between nodes on the internet.

There are many reasons one might want to create maps of the internet. The first is for research. By studying the topology of the internet we can study how the internet changes, and speculate about its development. In addition to basic research, another use for accurate maps of the internet is in the creation of equally accurate simulations. By having a real topology, new protocols and hypotheses can be tested and validated in real-world conditions. Understanding the topology of a network also allows for easier diagnosis of problems in the network. Finally, overlay networks can benefit from knowing the topology of the networks over which they operate.

Unfortunately, mapping the internet is not a simple matter of observation. The topology of the internet is not directly observable in its entirety. It can only be seen by sending individual traceroutes across it. Each single probe gives us an indirect view of part of the underlying topology and with enough such probes a semi-complete map can be created. The success of this approach depends on a number of factors: the traceroute source, the traceroute target, the conditions of the network, routing policies in the network, and the network topology itself. For mapping, we are at the mercy of the behavior of the network itself and only have under our control the sources and targets of our traces. The challenge is to build an accurate map under this constraint.

Even if one is able to gather enough probes to build a complete map, this approach is still hampered by the fact that the internet is a dynamic system where links and routers are constantly being added and removed. Since we cannot complete all traceroute probes instantaneously, there is a period of time over which the mapping must be conducted. Many changes to topology and routing can occur during this period, which can introduce errors to the ultimate topology that is discerned.

Two types of errors can occur: false positives and false negatives. False negatives can occur when our traceroutes do not traverse some part of the actual topology—we have failed to map some part of the internet. False negatives also occur when nodes along a path do not respond to a traceroute. False positives occur for a number of reasons. First, a link or node could be removed from the topology after having been observed. That is, we may observe a link or node that later disappears, causing our map to be out of date. Routing changes during a traceroute can also cause a false positive. Because traceroutes incrementally probe each node along a route, if routing changes mid-traceroute then we "discover" an edge that does not actually exist. Thus, traceroute is a "noisy sensor" and the data gathered via traceroutes should be considered noisy data.

In order to make maps that are as complete as possible, many mapping tools add more traceroutes from more sources to more targets in the hopes of making a more thorough map. Intuitively, this technique hopes to maximize discovery—the number of new nodes and edges seen—at the expense of collecting more redundant data. Does data bear this intuition out: how redundant are these tools?

At least one mapping tool, Rocketfuel, has attempted to use heuristics to choose which traceroutes to perform in order to reduce redundant data collection without sacrificing discovery. But such pruning is at odds with a desire to reduce the noise in the data—normally, *more* data is needed to overcome noise, not less. Does Rocketfuel indeed have less redundancy, and if so, what is the trade off between that reduction and uncertainty due to noise?

To answer these questions, we will examine the *marginal utility* of observations for maps made with three different tools: Skitter, Scriptroute, and Rocketfuel. Marginal utility is the contextual usefulness of an observation, given all of the observations that have preceded it.

This paper makes a number of contributions to the analysis of internet mapping. First, we analyze how efficiently different mapping tools make use of their traceroutes to build topology maps. Second, we investigate and quantify whether certain traceroute measurements are more valuable than others. Third, we are the first to perform such an analysis while treating traceroute as a noisy sensor. Finally, given the uncertainty in our traceroute observations, we quantify how different mapping methods cope with this uncertainty.

Paper Outline Section 2 discusses our methodology for modeling and evaluating the production of maps by each tool. Section 3 describes the three data sets that we used. Section 4 describes our evaluation metrics and applies them to the data to yield the raw analyses. Section 5 discusses the results of the analyses. Sections 6 and 7 discuss related and future work, respectively, and Section 8 concludes.

2 Methodology

2.1 Modeling and Evaluating a Map

In order to quantify the relative coverage of different maps, we model a map as a directed graph $G = (V, E)$, where V and E denote the usual sets of nodes and edges, respectively. Each node and edge also has an associated count of the number of times it was observed during the construction of the map. A map is constructed over time as an aggregation of discrete *measurements*. For our study, a measurement is simply a single source or target. Each measurement returns a partial map, which is added to a growing intermediate map. The intermediate map is the union of all measurements' partial maps, with their observation counts summed appropriately. The final map is the intermediate map left after the last measurement. We measure the utility of adding measurements in terms of changes to the intermediate map.

There is some hypothetical map $\hat{G}$ that captures the true topology of the internet at a given moment, but without total omniscience across all ISPs, $\hat{G}$ can never be constructed. Instead, each G that we induce is an approximation of $\hat{G}$. Furthermore, it is an approximation in two distinct ways. First, if there are no false positives in the traceroute observations, the nodes and edges of G will clearly be only subsets of the true nodes and edges of $\hat{G}$. Secondly, if there are false positives, then V and E may be erroneous supersets of $\hat{V}$ and $\hat{E}$.

An ideal evaluation of any map would be a comparison of the constructed map to $\hat{G}$, and an ideal evaluation of any mapping tool would compare additional measurements in terms of the resulting intermediate map's approximation to $\hat{G}$. Since we do not have a way to compare maps to $\hat{G}$, we instead use the final map produced by a tool as that tool's best approximation to $\hat{G}$. Each additional measurement is evaluated by comparing the new intermediate map to the final map. This is the same method used in [1], so our results may be compared to theirs.

2.2 Confidence Weighting

As mentioned, observations of internet topology are not direct and there can be errors in the observations that divorce them from reality. To model this uncertainty, the observation counts associated with each node and edge are used to compute the probability that the node or edge actually exists. To represent the fact that the observations are only indirect evidence of the existence of a node or edge, we model each traceroute as a noisy sensor. There is a probability of error, d (for "doubt") associated with the sensor. The probability $P(v)$ that a node v exists is $1 - d^{n(v)}$, where $n(v)$ is the number of times v was observed. The same formula applies for edges, with $n(e)$ substituted. $P(v)$ can be thought of as a measure of confidence in the existence of v. $P(v) = 1$ indicates absolute certainty, and $P(v) = 0$ indicates total doubt. $1 - d$ can then be thought of as a "confidence increment." Each observation moves confidence $1 - d$ percent closer to certainty.

Three facts should be noted about the effect of d on $P(v)$.

$$P(v) = 0 \qquad \forall n(v) \text{ when } d = 1 \tag{1}$$
$$P(v) = 1 \qquad \forall n(v) \text{ when } d = 0 \tag{2}$$
$$\lim_{n(v)\to\infty} P(v) = 1 \qquad \text{when } 0 < d < 1 \tag{3}$$

Equation 1 is the obvious assertion that under total doubt no amount of evidence will ever increase confidence. Equation 2 shows that when no error is assumed, confidence moves to absolute certainty after a single observation and stays there for all future observations. This is the model previously used to measure performance of mapping tools, so our study can reproduce earlier results by setting d to 1. Equation 3 shows that in the limit of infinite evidence confidence will always converge to certainty for any non-zero error probability. This also has the corollary that each successive observation increases $P(v)$ less than any earlier observation. Thus redundant observations only help up to a point. Even when accounting for uncertainty, there is still a diminishing return for redundant information.

A single error parameter assumes symmetric false positive and false negative error probabilities. Such an assumption is clearly an oversimplification, but we want only to examine the trade-off between redundancy and confidence so it suffices for our study. By varying the value of d used during an analysis, we can determine what level of doubt a tool can support. This allows us to quantify the trade-off between efficiency and certainty when given a certain amount of redundancy.

2.3 Evaluation

We use four different metrics to evaluate the utility of adding new measurements to a map: cumulative coverage, novel discovery ratios, entropy, and Kullback-Leibler divergence. (Each metric is described individually in Section 4.) We divide measurements into new sources or new targets, and apply each metric to each of the three data sets under each division.

3 Data

We analyzed data produced by three different mapping tools. Each tool also has access to a different set of sources and uses a (potentially) different list of targets. More importantly, each tool has a different strategy for generating traceroutes and aggregating them into a map. Since all three tools rely on traceroute for their observations, they all share a natural division of their measurements into per source or per target groups.

3.1 Skitter

The first data set comes from the CAIDA Skitter tool [2]. Skitter is a collection of 24 sources and 5 (potentially overlapping) target lists. The target lists range in size from 133,000 to 824,000 with a total of 1,119,373 unique targets. Each source uses one or more of the target lists and continually loops through its chosen lists sending traceroutes to each target. We selected 3 days worth of data from December 2002 so that they would overlap with the data described in Section 3.2. Altogether, the Skitter data contains 36,976,237 separate traceroutes.

Skitter is not specifically a mapping tool, but is a larger sensor intended to continually gather data about the internet. Many maps have been made from Skitter data, though, and thus our analysis is not of Skitter in its entirety but of its use as a source of mapping data. When building a map, we take each of the 24 sources as a new measurement and add all traces from that source to the intermediate map. The sources are ordered randomly. Since targets are not standardized across sources, there are unstable results caused by the uncontrolled variable. Nevertheless, any map built with Skitter data will have to use all of the data from a source or none of it and thus any map builder could benefit from our analysis. Because of complications from overlap within the target lists, we were not able to analyze the Skitter data rigorously per target.

3.2 Scriptroute

The second data set was originally presented in [4]. It was gathered over December 18, 19, and 20, 2002. It uses 70 distinct sources, and has 251,518 unique targets chosen from 125,000 network prefixes advertised via BGP. It contains 11,132,254 different traceroutes. These traceroutes have been post-processed to remove the first and final few hops. Thus they contain "backbone-only" paths more likely to be free of gateway routers and end hosts.

Like Skitter, Scriptroute [6] is not specifically a mapping tool, but rather a framework from which to run distributed internet measurements. Each source in this data set is a Scriptroute host, but the data we analyze was gathered specifically to map the internet. Each source sent traceroutes to all of the target prefixes and finished a complete run in approximately 6 hours. One run was launched per day.

For per source studies of the Scriptroute data we ordered the sources randomly. Because of the large number of targets and total traceroutes, we could not perform an exhaustive per target analysis. We instead selected a random sample of 33,000 targets from the first day and analyzed the map constructed with just these targets.

3.3 Rocketfuel

The final data set was originally presented in [5]. It was gathered from December 27, 2001 through February 1, 2002. It uses 837 distinct vantage points and 59,994 unique targets. (A target in this data set is a block of IP addresses from which one representative is chosen.) Altogether, it comprises 1,049,235 different traceroutes.

Rocketfuel is specifically designed to minimize the number of traces required for building a complete map of an ISP. It attempts to discern the entry and exit points for routes traversing the ISP and select only those routes predicted to maximize the amount of new information gathered. As such, it is not an "internet-wide" mapping tool as much as it is an ISP specific mapping tool. This data set contains the aggregate data gathered to map 10 different ISPs.

This data set was gathered with publicly available traceroute servers. To minimize the load on these shared resources, each source was only asked to perform one traceroute every five minutes—hence the month required to complete the entire mapping.

We analyzed this data completely both by source and by target. Both measurement sets are ordered randomly. Since Rocketfuel chooses sources and targets together, there is some uncontrolled variance in targets when different sources are selected (and vice versa).

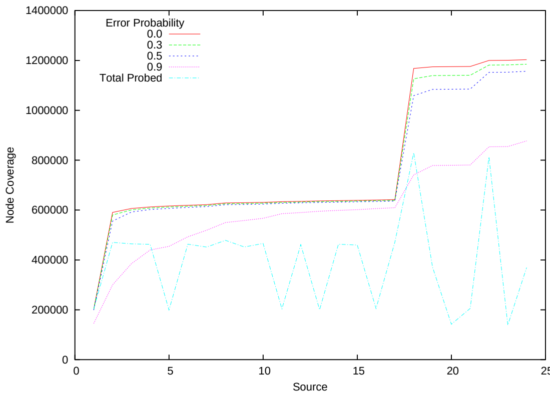
4 Analyses

Each subsection below describes a metric and presents the results of evaluating that metric across all three mapping tools.

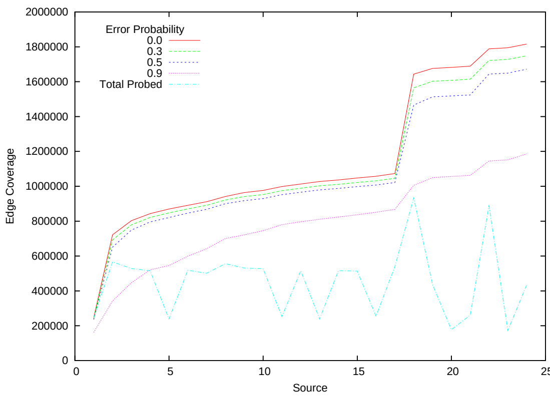
Each plot shows the metric as it was applied to four different map representations, where the value of d varies between each representation. The first line is always for $d = 0.0$, which is zero error and is thus the same assumption as earlier studies. The other lines are for values of 0.3, 0.5, and the extreme 0.9. We chose to show 0.3 because it represents a confidence level that reaches 0.97 after 3 observations and 0.99 after 4 observations. Requiring 3 or 4 observations to raise belief acceptably high is an initial heuristic suggested by [7]. We chose 0.5 and 0.9 because they show increasingly less credible extremes of doubt—traceroute is probably not in error 50% of the time, and certainly not 90% of the time. But some of the tools produce maps with enough redundancy to support error probabilities that high and we want to make that evident.

4.1 Node and Edge Coverage

The first, obvious metric for determining the value of a adding new measurement to a map is the increase in cumulative coverage that



(a) Node coverage



(b) Edge coverage

Figure 1: Per source coverage, Skitter

the measurement brings. We define the node coverage of a map as the mean probability of all possible nodes.

$$\frac{\sum_{v \in V} P(v)}{|V|}$$

V is the set of nodes from the final map to which we are comparing all intermediate maps. If a node has not been observed, its probability is zero. And if d is 0, then the node coverage is simply the ratio of nodes seen so far, and is exactly equivalent to the node coverage defined in [1]. Edge coverage is defined similarly.

The net increase in coverage for each new measurement is an indication of that measurement's utility to the final map relative to the size of the final map. A consistent increase in cumulative size as measurements are added would show that each new measurement is contributing new information to the graph. A decline in the growth of the cumulative map would show that new measurements are becoming redundant.

The coverage plots in this section have an additional line running along the bottom that shows the total number of unique node or edge observations for a given measurement. If there were no redundancies, the line for zero error probability would simply be the running sum of the values on the total observation line.

4.1.1 Skitter

Figures 1(a) and 1(b) show Skitter's cumulative node and edge coverage per source. The most obvious feature of these graphs is that the coverage rises to a stable level after only 2 source, then stays steady, then shoots up after the 18th source, before staying steady again. We do not believe that this is the result of a particularly advantageous source. The 18th source in this ordering has a set of target lists that contain 68% more targets than the entire number of targets probed up to that point. The gain in coverage for that source is a function of the increase in targets, not an additional source. Further support for that conclusion is the large number of total nodes probed for the 22nd source. It too has a much larger target list than all other sources except the 18th, but its list still overlaps with the target lists of the 18th source. That there is not an equivalent change in coverage for the 22nd source suggests that the spike is from targets. We further test this hypothesis by reordering a subset of the data. The results of that reordering are presented in Section 5.

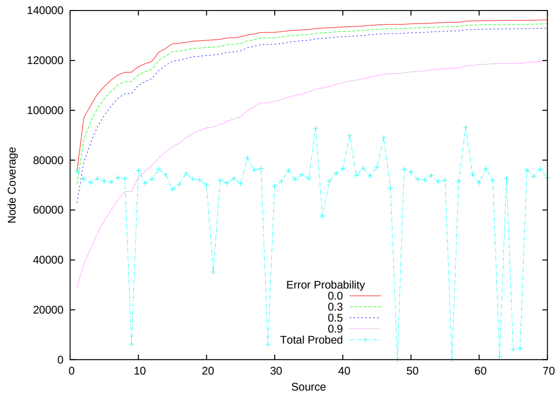
Additionally, note how close together the 0.0, 0.3, and 0.5 error probability lines are for node coverage. That there is no appreciable spread between them shows that the data is redundant enough to push them into the limits of very diminished utility. The spread is greater for edge coverage, however, and indeed, edge coverage continues to increase while node coverage has steadied. There is a greater utility in additional sources for discovering edges than there is for discovering nodes. (It is difficult to compare the confidence weights here due to differences in scale. They will become clearer under normalized metrics, particularly in Section 4.3.)

4.1.2 Scriptroute

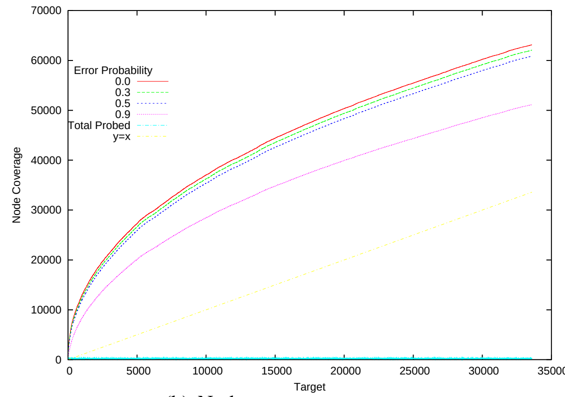
Figures 2(a) and 2(c) show Scriptroute's cumulative node and edge coverage per source. Predictably, its node coverage rises quickly and then levels to a much shallower rate of increase. The later sources with appreciably higher numbers of nodes probed do not cause correspondingly appreciable increases in coverage. Note also that it sees half of all nodes from the very first source.

The edge coverage rises more steadily, displaying a better return per additional source. Increases in the number of edges seen cause increases in cumulative coverage, and the difference between error probabilities is more pronounced. For this data as well there is more utility in new sources for edge coverage than for node coverage.

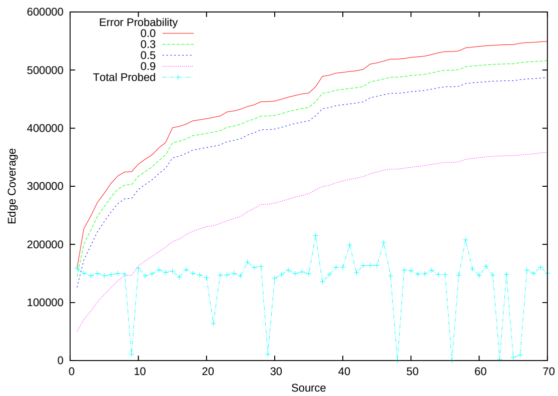
Figures 2(b) and 2(d) show Scriptroute's cumulative node and edge coverage per target. Since it is expected that a new target adds at least one new node and edge (the target itself and the edge leading to the target) we have added the line $y = x$ to the graphs for comparison. Even though it rises quickly in the beginning, coverage continues to increase through additional targets more than it does through additional sources. While its slope becomes linear it is still greater than one, showing that each new target still adds more than just itself to the coverage.



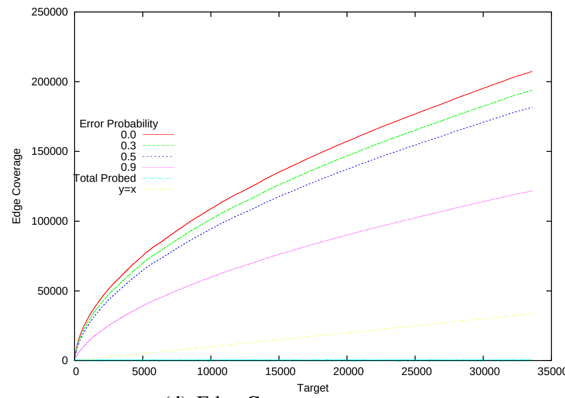
(a) Node coverage per source



(b) Node coverage per target

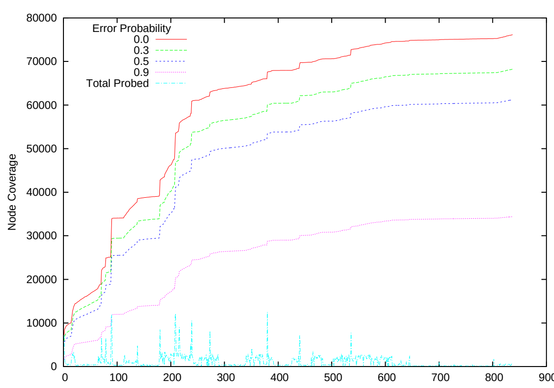


(c) Edge coverage per source

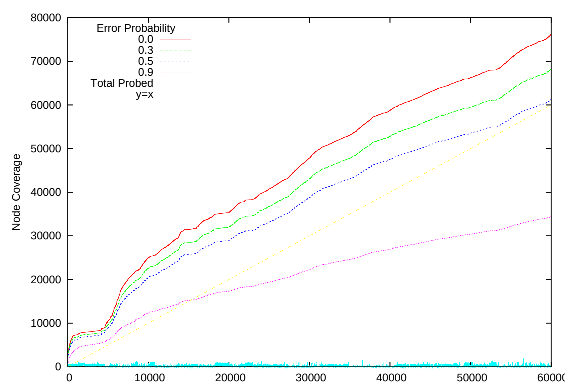


(d) Edge Coverage per target

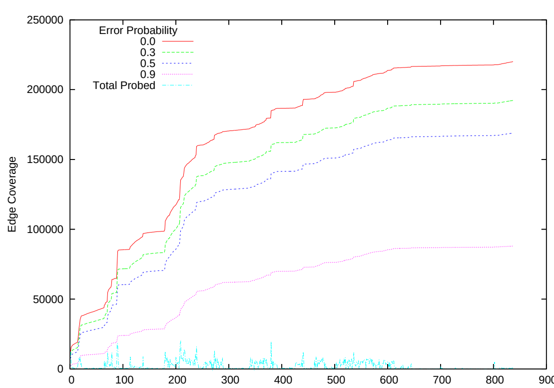
Figure 2: Cumulative coverage, Scriptroute



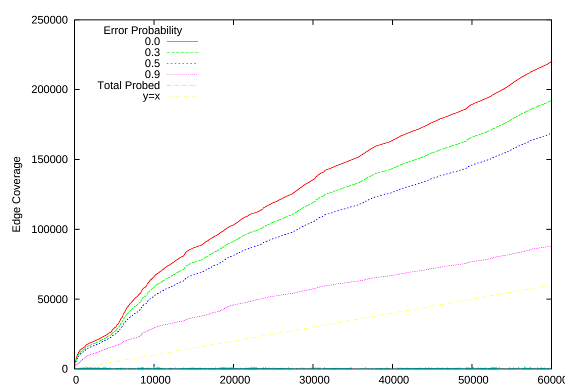
(a) Node coverage per source



(b) Node coverage per target



(c) Edge coverage per source



(d) Edge coverage per target

Figure 3: Cumulative coverage, Rocketfuel

4.1.3 Rocketfuel

Figures 3(a) and 3(c) show Rocketfuel's cumulative node and edge coverage per source. Its coverage increases quickly with the first measurements, but more erratically than the other tools. It also does not decrease in gain as fast as the others. These figures suggest that Rocketfuel lives up to its promise of increased return for each measurement. But only to a point: the final fourth of the sources do not add as much as the first three-fourths but they also do not probe as many total nodes or edges. The normalized metrics later will help distinguish this fact.

Figures 3(b) and 3(d) show Rocketfuel's cumulative node and edge coverage per target. These results are the most impressive seen so far. Additional targets yield the same increase in coverage almost independent of how many targets have been added previously.

4.2 Novel Ratios

Looking at only cumulative coverage treats all measurements equally, ignoring differences in size between measurements. A redundant but large measurement may increase the cumulative coverage more than a smaller measurement with a better individual efficiency. If the size of a measurement is indicative of the cost of performing that measurement, then efficient smaller measurements may provide higher marginal utility than inefficient larger measurements.

Novel ratios are computed exactly as one would expect: the number of new nodes (or edges) found divided by the total number probed. Obviously, the novel ratio for the first measurement is always one, so all of the graphs begin with the second measurement.

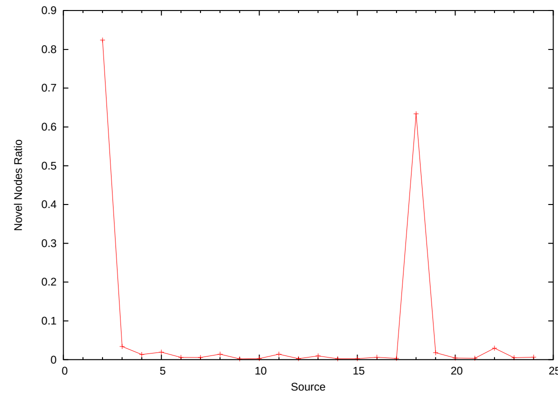
4.2.1 Skitter

Skitter's novel ratios (Figures 4(a) and 4(b)) do not provide many surprises given its cumulative coverage results. Excepting the 18th and 22nd sources, all of its novel ratios after the 3rd source are below 11% for edges and 2% for nodes.

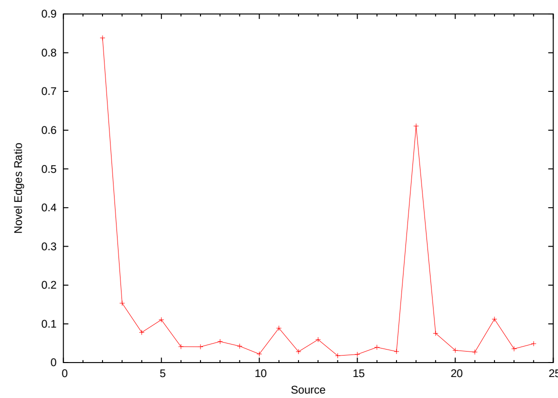
4.2.2 Scriptroute

The novel ratios of the Scriptroute data (Figure 5) are similar to those for the Skitter data. Novelty decreases quickly after the first few measurements. Interestingly, sources 63 and 66 have uncharacteristically large novel ratios *and* are two of the smaller sources in the data set (note their corresponding dips in the lowest line in Figures 2(a) and 2(c)). Compare them to source 36: it also has uncharacteristically high novel ratios but is one of the larger sources in the set. Coverage continues to increase through both 63 and 66 as much as it does through 36. This supports the intuition that efficient small measurements can be as useful as larger measurements.

The per target ratios also confirm the coverage results: additional targets have a diminished utility, but they are more valuable than additional sources.



(a) Ratio of novel nodes to all probed nodes

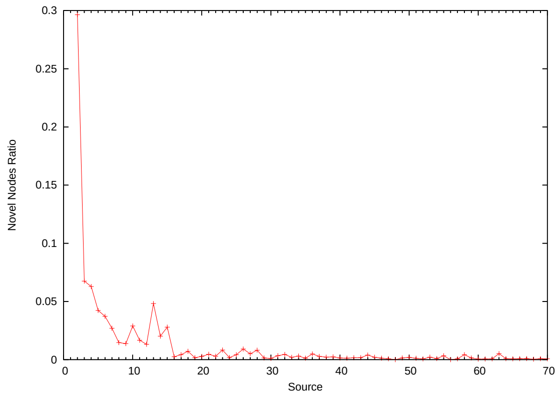


(b) Ratio of novel edges to all probed edges

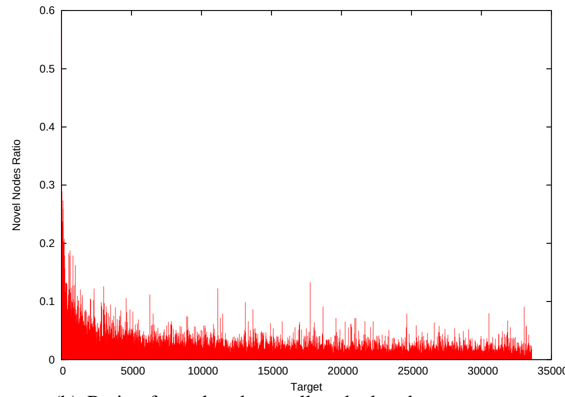
Figure 4: Novel ratios per source, Skitter

4.2.3 Rocketfuel

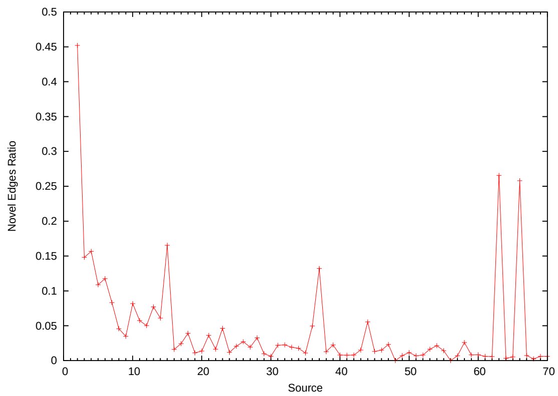
The Rocketfuel novel ratio results shown in Figure 6 are much more surprising than the Skitter and Scriptroute results. The decrease in novelty per source is not as sharp as the others, but sources still show diminishing returns. Most surprising is the steadiness of the per target ratios. 641 targets have novel node ratios over 40%, and 2584 targets have novel edge ratios over 40%. 16 targets see 100% novel nodes and 77 targets see 100% novel edges.



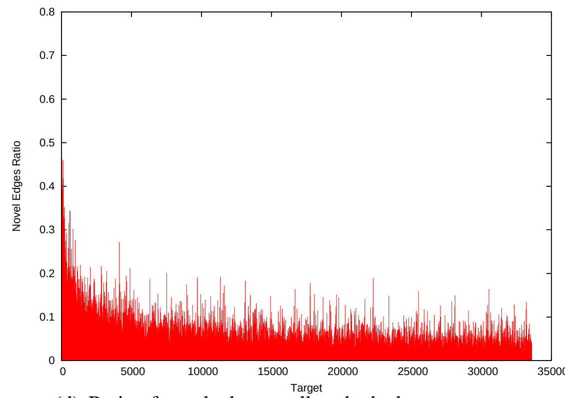
(a) Ratio of novel nodes to all probed nodes, per source



(b) Ratio of novel nodes to all probed nodes, per target

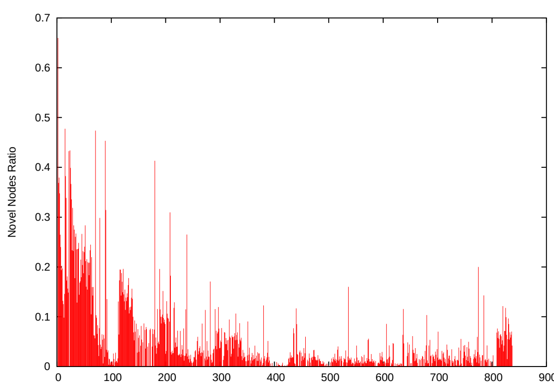


(c) Ratio of novel edges to all probed edges, per source

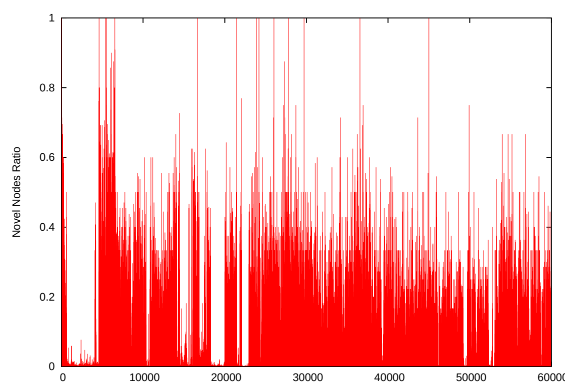


(d) Ratio of novel edges to all probed edges, per target

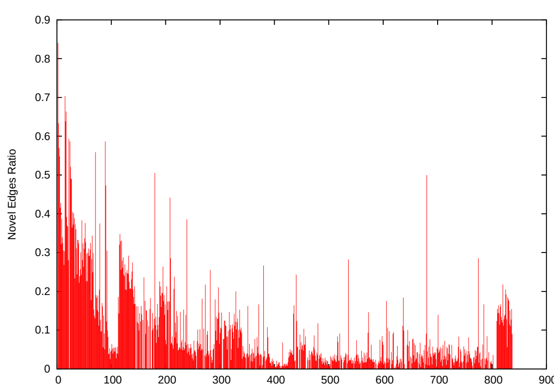
Figure 5: Novel ratios, Scriptroute



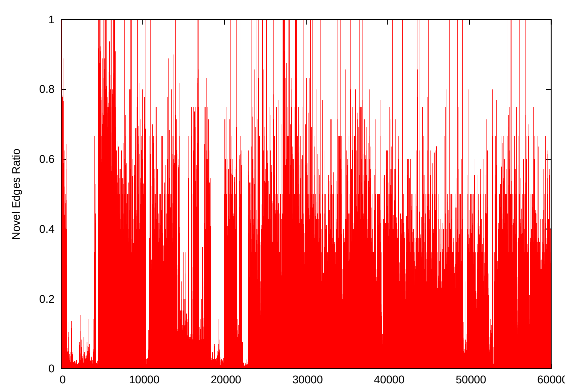
(a) Ratio of novel nodes to all probed nodes, per source



(b) Ratio of novel nodes to all probed nodes, per target



(c) Ratio of novel edges to all probed edges, per source



(d) Ratio of novel edges to all probed edges, per target

Figure 6: Novel ratios, Rocketfuel

4.3 Entropy

Cumulative coverage measures the contribution of each measurement to the final map, and novel ratios estimate the internal utility of each measurement. But neither of those metrics is easily comparable across maps of different sizes built from different data sets with different numbers of measurements. [1] introduces the idea of an information theoretic analysis for quantifying the contribution of individual measurements to a map. After normalizing for differences in size, information theoretic metrics express the value of measurements as expected bits of information to be gained. These bit values are comparable across any map from any tool. [1] focuses on relative entropy between two maps (there is more on this below, in Section 4.4), and we simplify that analysis by also considering the individual entropy of a single map.

The entropy of a probability distribution A is the mean number of bits needed to encode each event in the distribution. It is denoted $H(A)$ and computed as

$$H(A) = \sum_{e} -P(e) \log P(e)$$

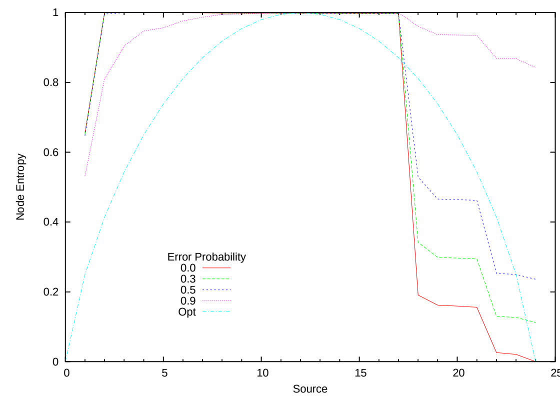
where e ranges over each mutually exclusive event in A. The second term in the product is the information content of a single event. The lower $P(e)$ is, the more negative $\log P(e)$ becomes, which supports the intuition that unlikely events carry more information. The first term weights that raw information content up or down according to the probability of the event, and makes the entire sum a weighted average.

As we have defined them, node and edge coverage are each a distribution over a single boolean variable. We compute the entropy of those distributions to find the node and edge entropies of a map. Since entropy is the mean number of bits needed to encode an event, it is also the expected gain in information after observing a single event. As a map is built, the entropy of the intermediate map reflects the expected amount of information to be gained by adding a new measurement.

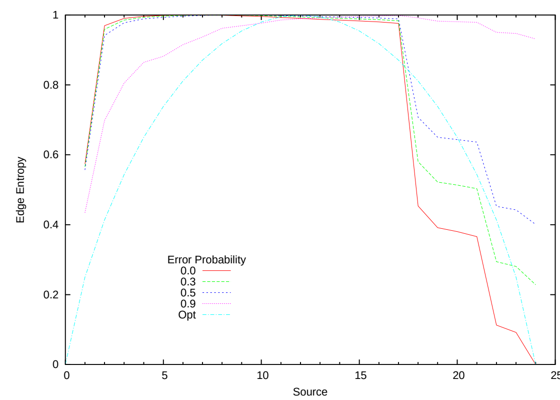
The change in entropy as a map is built proceeds in two phases. First, entropy increases as measurements are added and coverage increases to 50%. This stage can be seen as initially discovering that the internet exists and that there is information to be learned about it. Obviously, we know *a priori* that the internet exists, but we have not included that as a prior for our distributions. As such, the entropy numbers in the first phase are a bit misleading. Nevertheless, the *change* in entropy still reflects how fast information is gained.

In the second phase, entropy peaks when coverage is at 50% and then begins to decline. The moment of peak entropy marks the measurement after which marginal utility begins to diminish. The expected number of bits gained by each measurement declines.

It is here that our confidence weights become most relevant. Clearly, since we use it as our goal, the map built with zero error probability will end with complete coverage and an entropy of zero. The maps built with other error probabilities end with nonzero entropies. Their final entropy values show the number of bits (which would contribute to confidence) that could still be gained from an additional measurement. A higher final entropy reflects a larger trade-off between discovery and confidence.



(a) Node entropy



(b) Edge entropy

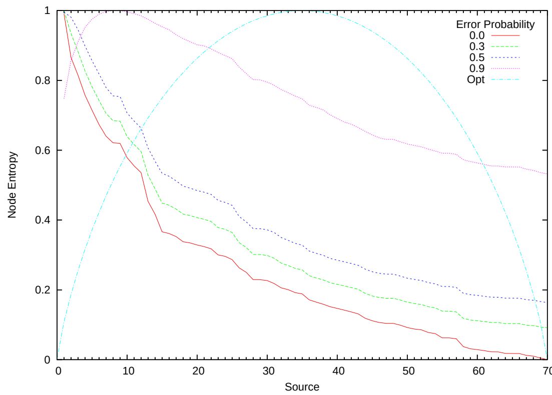
Figure 7: Entropy per source, Skitter

To each plot in this section we have added a smooth arc that represents one theoretical optimum entropy. This arc represents the entropy of the map if all of its information were spread evenly across all measurements so that each measurement had a predictable utility. Note that even under this scenario marginal utility will still decrease because each measurement contributes a continually smaller percentage to the growing intermediate map. What this theoretical optimum maximizes is predictability of return on effort.

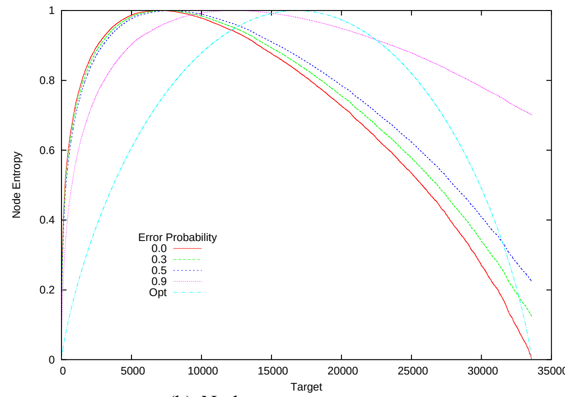
4.3.1 Skitter

Skitter's entropy (Figure 7) clearly displays its low utility for additional sources. A change in entropy means that information has been gained and the underlying distribution has been adjusted. Skitter's entropy stays flat across the majority of its sources, betraying their poor utilities.

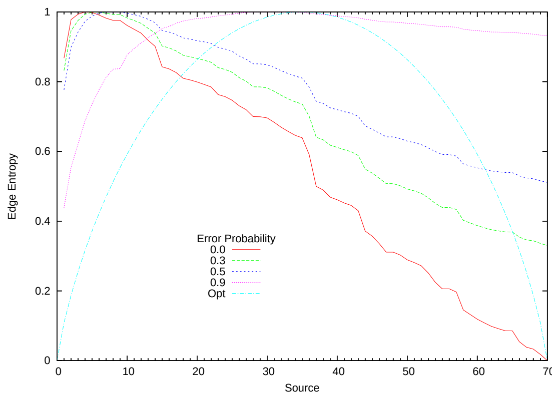
Additionally, note the final entropies of (for example) the map built with error probability 0.3. 0.11 bits of information about nodes and 0.22 about edges are expected to be gained from a new measurement.



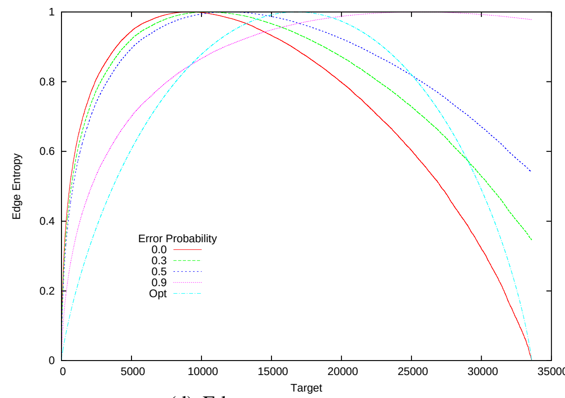
(a) Node entropy, per source



(b) Node entropy, per target

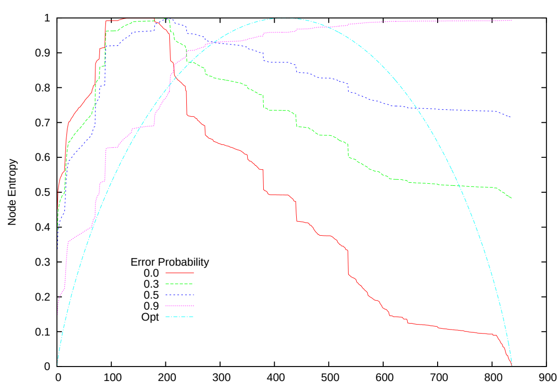


(c) Edge entropy, per source

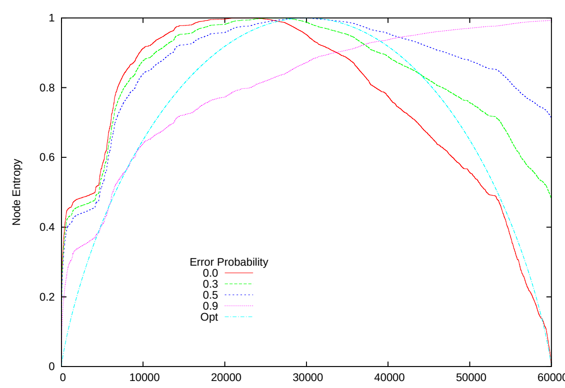


(d) Edge entropy, per target

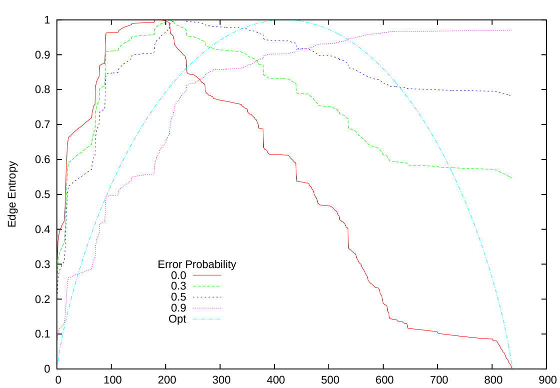
Figure 8: Entropy, Scriptroute



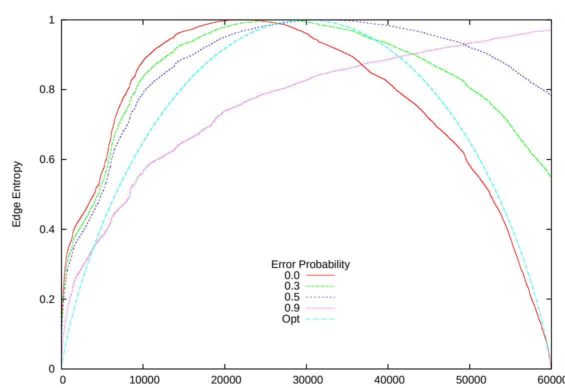
(a) Node entropy, per source



(b) Node entropy, per target



(c) Edge entropy, per source



(d) Edge entropy, per target

Figure 9: Entropy, Rocketfuel

4.3.2 Scriptroute

Scriptroute's per source entropy measurements, shown in Figures 8(a) and 8(c), show that it is more consistent in gaining new information than Skitter, but that it gains half of its information within the first four sources. The gradual decrease in entropy (throughout for edges, after source 14 for nodes) shows that it is slow to gather new information after the initial peak. Lastly, there is high redundancy in node observations but low redundancy for edges, as shown by the final entropies. If the sensor error probability is 0.3, 0.33 bits of information could be gained about edges by adding a new source.

Scriptroute's per target entropies (Figures 8(b) and 8(d)) show better marginal utility than the per source entropies. Maximum entropy is still reached early, but the decrease towards certainty is quicker and has a steepening slope. Because the per target data is a sub-sample of the per source data, the final entropies are not equal, but they are close: an expected 0.35 bits of information could be gained for error probability 0.3.

4.3.3 Rocketfuel

Rocketfuel's entropies show its greater utility and lower redundancy. Per source, its entropy rises and then falls very quickly, except for a stretch of unhelpful sources towards the end. It should be possible to reorder the sources so that the final helpful ones precede the unhelpful ones, and then to evaluate whether pruning those unhelpful ones would ultimately be beneficial. The high final entropies show Rocketfuel's low redundancy. 0.55 bits could be gained about edges from another source, if the error probability is 0.3.

Rocketfuel's per target entropies support the conclusion seen so far that targets have higher utility than sources. Rocketfuel's per target entropy curves come closest to the theoretical optimum, and they do not show any stretches of low information gain like the per source entropies.

4.4 Kullback-Leibler Divergence

The metric used in [1] to estimate the marginal utility of additional measurements is Kullback-Leibler divergence. K-L divergence[1] expresses the average number of extra bits needed to encode a system according to an incorrect probability distribution for the system.

K-L divergence, denoted $KL(A||B)$, is the relative entropy between two different distributions, A and B, over the same set of events. It can be derived from entropy as follows, where A is the true distribution, B is the incorrect distribution, and $P_D(e)$ denotes the probability of event e according to distribution D:

$$H(B) - H(A) \simeq \sum_e -P_B(e)\log P_B(e) - \sum_e -P_A(e)\log P_A(e) \tag{4}$$

$$\simeq \sum_e P_A(e)\log P_A(e) - P_B(e)\log P_B(e) \tag{5}$$

$$= \sum_e P_A(e)\log P_A(e) - P_A(e)\log P_B(e) \tag{6}$$

$$= \sum_e P_A(e)(\log P_A(e) - \log P_B(e)) \tag{7}$$

$$KL(A||B) = \sum_e P_A(e)\log\left(\frac{P_A(e)}{P_B(e)}\right) \tag{8}$$

Recall that the formula for entropy multiplies the information content of an event by the event's probability in order to appropriately weight the event's information against the information of all possible events. From that it follows that the probability used for the weighting must be the true probability. Equation 6 is derived from 5 by replacing, in B's entropy, the incorrect probability with the true probability while keeping the incorrect information estimate. This is also why equations 4 and 5 are only written as approximate equalities. K-L divergence is always positive and it is zero for equal distributions.

For our study we calculate the K-L divergence between the final map that would be constructed if there were zero sensor error and the intermediate map constructed as a measurements are added. This is equivalent to the metric called "offline utility" in [1], except that our addition of non-zero error probabilities means that some maps will not reach zero K-L divergence after all measurements have been added.

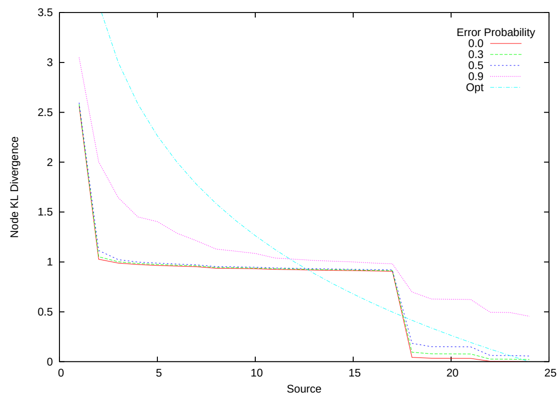
In this context, what K-L divergence shows is the number of bits needed to transform the intermediate distribution into the final distribution. A large drop in K-L divergence after a measurement corresponds to a large gain in information from that measurement. Thus, the marginal utility of a measurement can be estimated by the drop in K-L divergence provided by that measurement. These graphs provide the best "at a glance" estimate of the marginal utility per measurement.

To each of the graphs in this section we have added a line showing the K-L divergence of the same theoretical optimum as the optimum in the entropy graphs: the division of all information equally between measurements. The further below this line the actual K-L divergence dips, the lower the marginal utility of succeeding measurements.

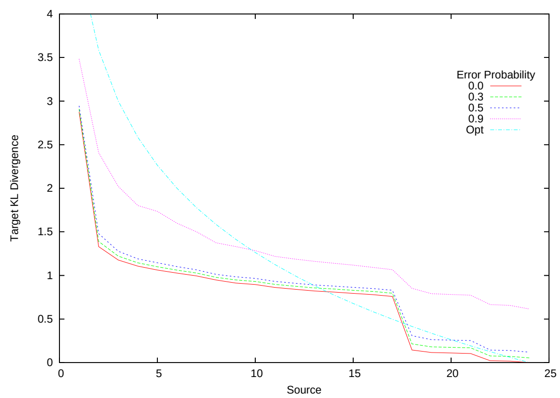
4.4.1 Skitter

Skitter's per source K-L divergences are shown in Figure 10. The level portion has the same interpretation as entropy: no information is being gained. That Skitter's K-L divergence exceeds the optimum shows that its fast early gain in information has tipped

[1] Sometimes called K-L distance, though it is not symmetric and does not obey the triangle inequality, and thus is not a strict distance metric.



(a) Node KL Divergence



(b) Edge KL Divergence

Figure 10: KL Divergence per source, Skitter

beyond inefficiency into wastefulness: utility has diminished so greatly as to not be worth any effort.

4.4.2 Scriptroute

Scriptroute's K-L divergence (Figure 11) clearly shows the utility intuitively expected for such an exhaustive mapping technique. The information that can be gained with each successive measurement falls very rapidly. Note that while that trend is true of both the per source and per target divergences, the per target ones have higher absolute values and are closer, though still below, the optimum curve.

4.4.3 Rocketfuel

The K-L divergence for Rocketfuel shown in Figure 12 shows much greater utility than the divergences for either Skitter or Scriptroute. Rocketfuel's per source divergence shows as much marginal utility as Scriptroute's per target divergence. Even greater than that is Rocketfuel's per target utility. Its divergence curves—particularly its per target edge divergence—are the closest to optimum of all three tools. Rocketfuel appears to live up to its promise of improved efficiency.

5 Discussion

The first conclusion to be drawn from the node and edge coverage analyses is that adding targets yields more additional information than adding sources. However, it is also interesting to note that while node coverage quickly levels off as the number of sources increases, there is always some gain in edge coverage. This implies that adding sources "fills in" the links between nodes in the graph.

It is also interesting to observe that Rocketfuel continues to gain relatively more coverage as sources are added. This result could come about for two reasons. It could be that adding a source gives a different perspective on the network and thus more information. Or, since sources and targets are chosen together in Rocketfuel, it could be that adding a source causes more targets to be explored, and that these new targets are disjoint from the rest. This is a question for future exploration.

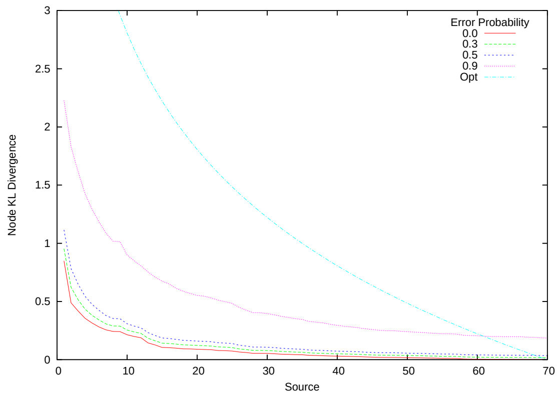
The novel node and edge ratios are also very surprising for Rocketfuel. Each source added seems to be gaining many new nodes and edges. We believe that this is a result of the intelligent route selection mechanism used by Rocketfuel. By considering ingress and egress points, Rocketfuel is able to add, for a source, the targets that it expects will explore parts of an ISP that have not been previously observed. This does appear to be effective. The entropy measures also reflect this conclusion about Rocketfuel.

In general, our analysis of entropy brought up some interesting observations as well. First we notice that each tool gains information quickly in the beginning of its run. This is expected, to some degree, since the first few traceroutes will pick up the local topology and this will always be redundant in future traceroutes. However, even in light of this, it is surprising how close the entropy graphs of Rocketfuel and Scriptroute are to the optimal—especially when considered per target.

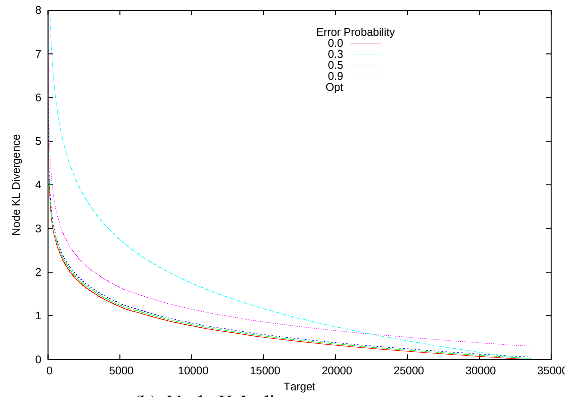
The K-L divergence plots show how poor the Skitter data is. The plot goes above the optimal which means that for a long time it is getting much less information than is expected as sources are added—implying that sources 3 through 17 are all redundant.

Our experiments with confidence weighted maps also provide interesting insights. We can look at the separation between maps built with different error probabilities to get an idea of how redundant the observations are. We see that the Skitter plots remain the least separated under varying error probabilities. This suggests that there is a large amount of overlap in the observations made by Skitter. A conclusion supported by the fact that there are many sources whose observations add nothing new to coverage.

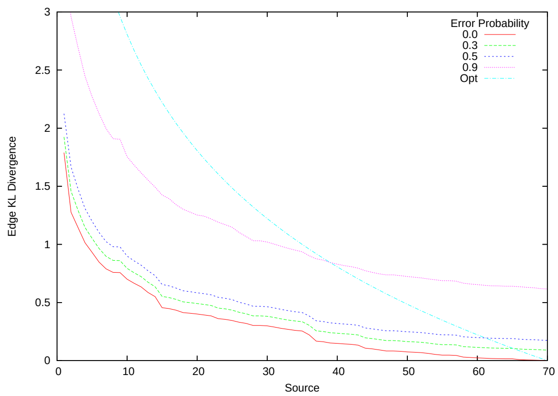
Scriptroute is more spread than Skitter, but less than Rocketfuel. From Figure 2(a) we can see that every Scriptroute source added explores about half of the nodes in the graph and a quarter of the edges. This supports the conclusion that there is a great deal of redundancy.



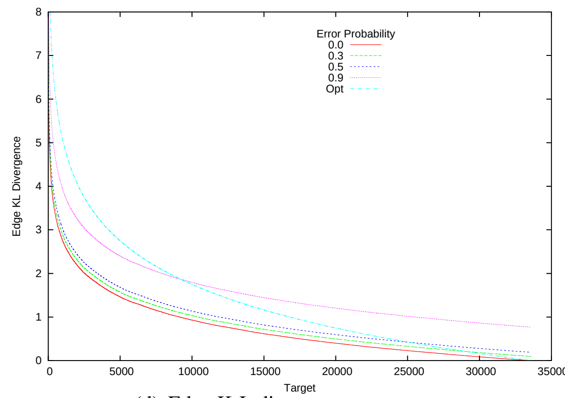
(a) Node K-L divergence, per source



(b) Node K-L divergence, per target

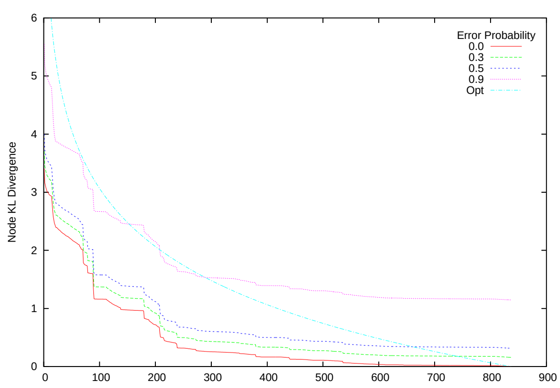


(c) Edge K-L divergence, per source

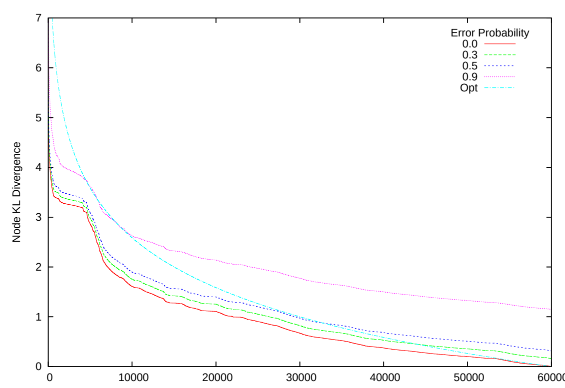


(d) Edge K-L divergence, per target

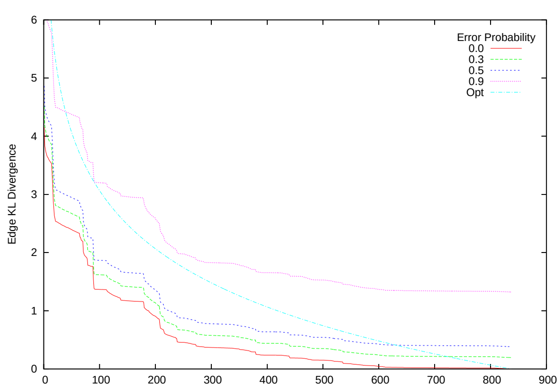
Figure 11: K-L divergence, Scriptroute



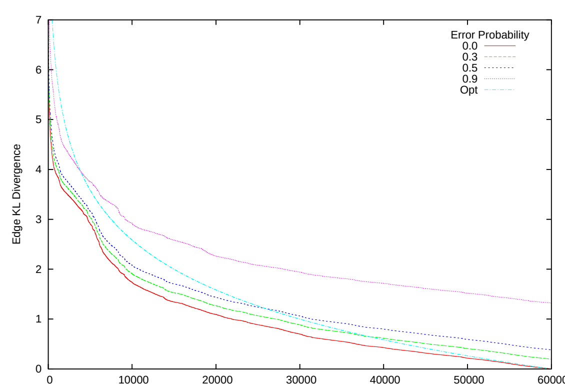
(a) Node K-L divergence, per source



(b) Node K-L divergence, per target



(c) Edge K-L divergence, per source



(d) Edge K-L divergence, per target

Figure 12: K-L Divergence, Rocketfuel

Finally, we see that Rocketfuel's plots are very spread out for different error probabilities. We believe this means there is not a lot of redundancy in its observations. While this is good because it reduces the total number of observations needed, it also reduces the confidence in the final graph. For small enough error probabilities this does not matter, but depending on the accuracy required this could be an issue. If we were attempting to map a noisier network (e.g. something ad-hoc) then the error probability could be high enough to have an effect.

6 Related Work

There has been quite a bit of previous work related to mapping the internet effectively. All these methods attempt to validate their results by comparison to previous mapping methods. Our work is different in that we simply try to quantify the efficiency of each system in terms of the amount of information they gain with respect to the effort they expend. Spring et al. [6] attempted to produce high level maps of network connectivity for determining path inflation in the internet. While they do map the internet, they are not overly concerned with the efficiency of their mapping.

In [5] the goal was to create complete and accurate maps. However, they only evaluate how accurate their final maps are, ad the reduction in measurements needed to produce the entire map (as compared to Skitter) but they do not examine the utility of the individual measurements that they took.

Even earlier work on internet mapping is that of Pansiot and Grad [3] who discovered that most of their map could be made using only a handful of sources. We confirm this results for two of our data sets. However, we agree with [5] in that the Rocketfuel does gain additional data from adding more sources. Neither of these studies looked at the effect of including confidence weighting in the results.

Our analysis work derives directly from that of Barford, Bestavros, et al., as evidenced by our frequent citation of [1]. They were the first to attempt to quantify the marginal utility of sources and targets for building an internet map. However, their work focuses only on a single set of data (12 Skitter sources), while our work examines data from three different internet mapping technologies that each use a different technique for mapping. We also extend their analysis to include confidence weighting and to show how that weighting effects the marginal utility results.

7 Future Work

7.1 Enhanced Models

Our current mapping model is very preliminary. As in previous efforts, we consider each traceroute as a single, unredundant observation. In actuality, most traceroute implementations contain redundancy of their own—typically three retries per hop. We could integrate the redundancy information from traceroute directly into our confidence calculations.

Also, many times there are missing hops in the traceroute data that need not be entirely ignored. If, for example, on some path node 6 does not respond to traceroute but nodes 5 and 7 do, we can still infer the existence of node 6, and that it has links to nodes 5 and 7. Furthermore, if 5 and 7 are within the same ISP, we can assign to 6 a probability for also being in the same ISP, and we could even use the DNS names of 5 and 7 to attempt to guess the name of 6. So simply ignoring any node that does not return its name and address throws away information.

Now that we have introduced a model that can accept probabilities and incomplete observations, we could take this additional information into account to improve the results of mapping.

7.2 New Analyses

The analyses presented in this study are compatible with those from [1], but with our expanded representation many new ones are now possible. [1] only represented the entire map with the probability that a single node or edge had already been included in the map. We represent each node and edge with its own probability of existence, and represent the map as the mean of these probabilities.

Thus, in addition to computing entropy and K-L divergence calculations over the mean probability, we could compute them separately for each node and edge and then take the mean of those results—in other words, the mean of the entropies, not the entropy of the mean. We could even compute the full joint probability of all nodes (or edges), and take the entropy of that—but summing over the joint would involve $2^{|V|}$ terms. Which leads to the question of whether the joint could be factored according to conditional independences in the node and edge distributions, and what those independences might imply.

Clearly, the greater degrees of freedom provided by a probabilistic model open many more avenues for exploration.

7.3 Time Decayed Confidence

Currently we assume that the total number of observations of a node or edge signify our confidence in its existence. Unfortunately, in the limit of increasing observations, links caused by noise will eventually gain high confidence. Also, we have no way to account for nodes or edges that later permanently disappear. It should be the case that the longer we *do not* see a node or edge, the more our confidence in its existence should *decrease*. In order to add this decay to our confidence it is necessary to collect statistics regarding the *periodicity* of mapping runs. That is, how long it takes to produce a complete map. This is difficult to know since the true map is unknown, but we can still define an approximate periodicity by using our existing information theoretic metrics to determine when coverage is reaching saturation.

To that we could add prior belief about the ephemerality of nodes and edges. The fringes of the map are more dynamic than the core, so we could use our topology results to classify nodes and then use that classification to re-weight our confidence.

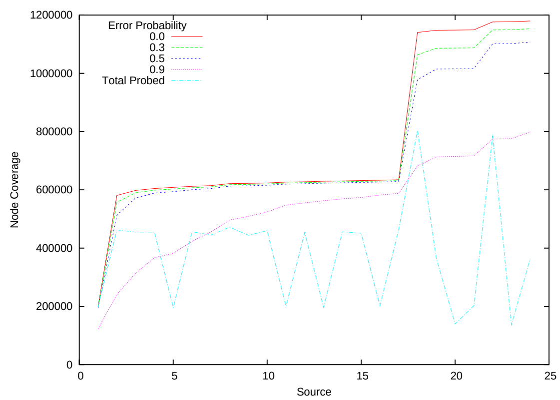


Figure 13: Skitter data, day 1, random ordering

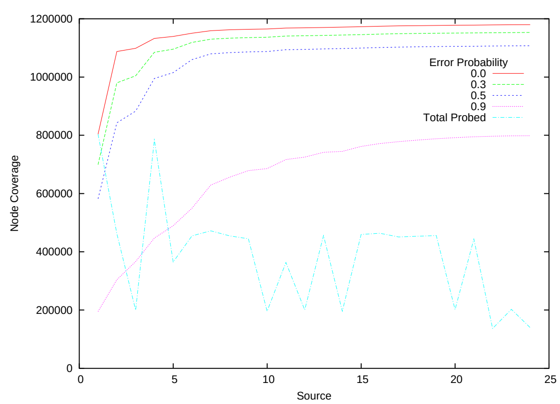


Figure 14: Skitter data, days 2 and 3, greedy heuristic ordering

7.4 Heuristic Reordering

We believe that statistics about the utility of measurements can be used as a prior for selecting and predicting the benefit of future measurements. As a preliminary look at this hypothesis we conducted an experiment that took one day of the Skitter data and examined the node coverage per source. We then reordered the sources for the second and third day of the data according to their contribution to node coverage on the first day.

The resulting effect on performance is impressive. Figure 13 shows the node coverage per source if reordering isn't done and Figure 14 shows the result if reordering is done. We can see that we immediately get almost all the nodes and can greatly reduce the number of measurements by using only a small number of sources.

7.5 Cost and Benefit

Finally, though we alluded to it in Section 4.2, most of our analysis (as well as earlier ones) ignore the cost of measurements. It is assumed that adding a new source is difficult, but this is not always the case. A system such as Scriptroute can gather data on many sources almost as easily as it can on a single source. Additionally, the study for which the Scriptroute data was initially gathered placed a high benefit in approaching 100% edge coverage. For them, the diminished utility was worth the low cost of additional measurements.

For other uses, such as automatically constructed maps used for overlay networks, the point at which additional measurements are not worth their cost may come sooner. Quantifying the cost of measurements in addition to their utilities would also be a useful analysis.

8 Conclusion

Our work has yielded many important observations and results regarding internet mapping.

- We were able to quantify the utility of adding sources and targets for several internet mapping systems.
- We have confirmed past results showing that adding targets is, in general, more beneficial than adding sources. This is true for all of the mapping systems examined. In fact, in the Skitter and Scriptroute data we found that almost all information is gained from the first few sources.
- Unlike in the Skitter and Scriptroute data, adding more sources continues to add information for the Rocketfuel system, with returns diminishing much more slowly.
- We have proposed a confidence weighting scheme to combat the inherent error in mapping observations.
- We have found that there is much more redundancy in the Skitter and Scriptroute data than in the Rocketfuel data. This means that these first two systems are much more robust in the face of error; however, this comes at the cost mapping efficiency.

9 Acknowledgements

Many thanks to Neil Spring for his generous help with this project. Many thanks as well to David Wetherall and Valentin Razmov for their suggestions and support during this project.

References

[1] Paul Barford, Azer Bestavros, John Byers, and Mark Crovella. On the marginal utility of network topology measurements. In *Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement*, pages 5–17. ACM Press, 2001.

[2] CAIDA. *The Skitter project*. http://www.caida.org/tools/measurement/skitter/.

[3] J. Pansiot and D. Grad. On routes and multicast trees in the internet, 1998.

[4] Neil Spring, Ratul Mahajan, and Thomas Anderson. The causes of path inflation. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 113–124. ACM Press, 2003.

[5] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with Rocketfuel. In *Proceedings of the 2002 conference on applications, technologies, architectures, and protocols for computer communications*, pages 133–145. ACM Press, 2002.

[6] Neil Spring, David Wetherall, and Tom Anderson. Scriptroute: A public internet measurement facility. In *4th USENIX Symposium on Internet Technologies and Systems*, 2002.

[7] David Wetherall. 2004. Personal Correspondence $\ddot{\smile}$.