Rigorous Measurement of the Internet Degree Distribution

Matthieu Latapy[1]
Élie Rotenberg[1,2]
Christophe Crespelle[2]
Fabien Tarissan[2]

[1] *Université Paris-Sorbonne, UPMC Univ. Paris 06, CNRS LIP6 UMR 7606, 4 place Jussieu 75005 Paris*

[2] *Université Claude Bernard Lyon 1, DANTE/INRIA LIP UMR CNRS 5668, ENS de Lyon, Université de Lyon*
Firstname.Lastname@lip6.fr

The degree distribution of the internet, that is, the fraction of routers with k links for any k, is its most studied property. It has a crucial influence on network robustness, spreading phenomena and protocol design. In practice, however, this distribution is observed on partial, biased and erroneous maps. This raises serious concerns about the true knowledge we actually have of this key property. Here, we design and run a drastically new measurement approach for the reliable estimation of the degree distribution of the internet, without resorting to any map. It consists of sampling random core routers and precisely estimating their degree with probes sent from many monitors scattered over the internet. Our measurement shows that the true degree distribution significantly differs from classical assumptions: it is heterogeneous but it decreases sharply, in a way incompatible with a heavy-tailed power law.

1. Introduction

The internet has become a crucial infrastructure sustaining our social, economic, cultural and scientific lives at both local and worldwide scales. Despite this, our understanding of its structure remains very limited. To gain more insight, the internet is often modeled as a network where nodes represent routers and links represent direct connections between them (wires, satellites, etc.). The degree distribution (i.e., for each integer k, the fraction p_k of nodes having k links) of this network is particularly important: it plays a key role for resilience to failures and attacks [1, 2], cascade and spreading phenomena [3, 4], and protocol and network design [5, 6]. As a consequence, it is an essential building block of most modern models of the internet [7–10].

https://doi.org/10.25088/ComplexSystems.26.1.1

However, current knowledge of this degree distribution is far from satisfactory, and it is at the core of a lively scientific controversy [11–18]. Indeed, the degree distribution is known only from internet maps obtained through intricate measurement procedures giving partial, biased and erroneous views. These measurements generally rely on the use of the traceroute tool, which provides in principle a route in the network from the monitor running the measurement to a given target. By collecting and merging many such routes, one obtains a map of the internet. See Figure 1(a) for an illustration. However, the traceroute tool is prone to numerous errors [16, 19–21] and, most importantly, the procedure itself is intrinsically biased [11, 14, 22, 23].

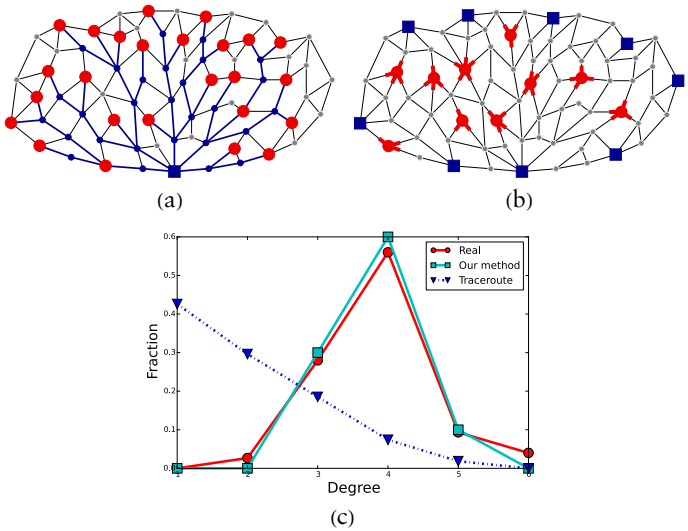


Figure 1. Comparison of our method to the classical traceroute method. (a) A traceroute measurement from one monitor (square node) toward 25 targets (bullet nodes). This measurement needs 97 probes. (b) Measurement with our method from nine monitors (square nodes) toward 10 targets (bullet nodes). Figure 2 details how the links of each target are discovered. This measurement needs 90 probes. (c) The true degree distribution of the network together with the estimates obtained by both methods.

For all these reasons, much effort is devoted to the design of more accurate internet measurement tools and to the collection of larger and larger maps [20, 24–30]. However, as measurement capabilities remain limited and as the internet evolves faster than our ability to measure it, this may very well be a dead end.

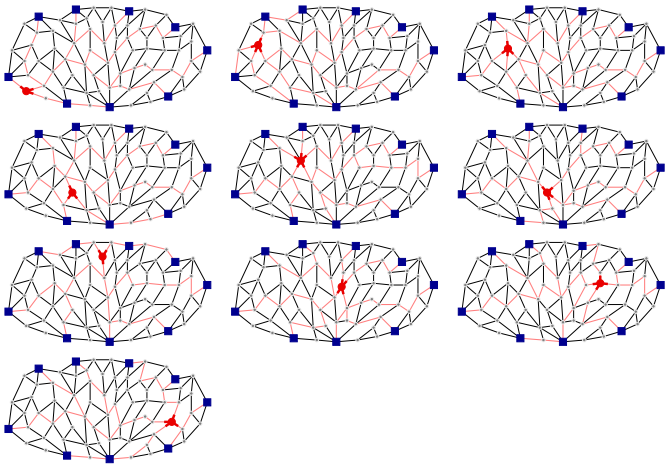


Figure 2. Measurement of the degree of 10 targets using our method. We display 10 copies of the network, one for each target measurement. On each copy we show the routes followed by the probes sent from our nine monitors (square nodes) toward the corresponding target (bullet node).

We present here a drastically new approach able to reliably estimate the degree distribution of the internet without resorting to any map. We probe randomly chosen routers from monitors scattered all over the internet and obtain an accurate estimate of their degree. We infer from these degrees a rigorous estimate of the internet degree distribution, far more reliable than previous knowledge. See Figure 1(b) and (c) for an illustration. This methodological shift raises challenging questions, which we address here. We conclude that, contrary to what most current studies assume, the degree distribution is heterogeneous but is not a heavy-tailed power law.

2. Our Measurement Method

A machine on the internet (a router or an end host) may have several interfaces, each corresponding to a connection to a neighbor machine. Each interface has its own address, and the degree of a router is nothing but its number of interfaces/addresses.

Let us consider an address t, which we call *target*, and let us denote by $r(t)$ the node (router or end host) to which t belongs. Internet protocol specifications [31, 32] state that when a monitor m sends a packet to destination t on an unallocated port, then $r(t)$ should answer m with an error packet (ICMP Destination Unreachable, Code 3/Port Unreachable). An important detail is that the source of this

error packet is in principle the address of the interface i by which $r(t)$ sent it; see Figure 3.

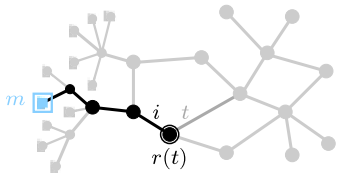


Figure 3. Monitor m sends a packet to destination address t on an unallocated port; the node $r(t)$ answers with an error packet with source address i, and thus m discovers interface i of $r(t)$.

Let us temporarily assume that $r(t)$ implements this feature correctly (we handle other cases in Section 3). Now consider a set M of monitors that all send such a probe toward address t. If for each interface i of $r(t)$ there is a monitor m in M that $r(t)$ answers using i, then one obtains the set of all interfaces of $r(t)$, and so its degree. This constitutes our basic measurement primitive: (1) from each monitor of a set M, we send a packet to an unallocated port of target address t; and (2) we collect the set $M(t)$ of all addresses used by $r(t)$ to answer monitors in M.

Depending on the target t and on the set of monitors M, this measurement primitive may succeed or fail to discover all the interfaces of $r(t)$. In particular, one has to distinguish between two very different kinds of targets: (1) the target node $r(t)$ is in the *core* internet (Figure 4(b)); or (2) the target node $r(t)$ is in the *border* (Figure 4(c)).

As illustrated in Figure 4(c), when the target address belongs to a border node, our measurement primitive may miss many of its interfaces and most likely discovers only the interface directed toward the core. The situation regarding core interfaces of core routers is quite different (see Figure 4(b)). Indeed, such interfaces route traffic toward a non-negligible part of the internet, and one may therefore expect that a reasonably large and well-distributed set M of monitors discovers them. Of course, this highly depends on the considered set of monitors, and we explain in supplementary material how to assess the quality of a monitor set in practice.

We focus here on the core, which is the key part of the network: it performs the nontrivial routing of packets from one point to another point of the network, while the border is a set of trees connected to this core, where packets are just forwarded up or down the tree (see Figure 4(a)). We therefore discard target addresses that belong to border nodes; see Section 3.

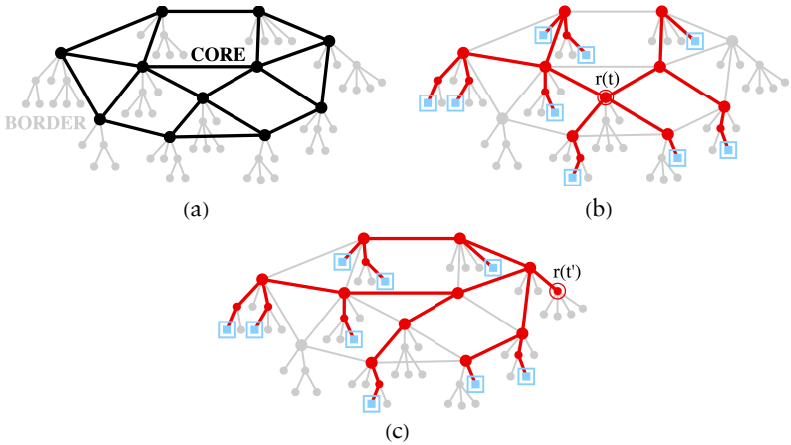


Figure 4. (a) The core and the border of the network; the border is the set of all trees connected to the network; the core is the part remaining when one removes these trees. (b) A set of monitors (the squared nodes) sends probes toward a target address t belonging to a core router $r(t)$ and obtains its four core interfaces of $r(t)$. (c) The same monitors send probes toward another target t' belonging to a border router $r(t')$ and miss most interfaces of $r(t')$.

In summary, we expect a good enough set of monitors M to be able to discover all or almost all core interfaces of any core router, leading to an estimate of its degree in the core internet. Once this measurement primitive is implemented, one may use it to observe the degree of all targets in a set T. If T is a set of core routers sampled uniformly at random (which means that all core routers have the same probability to appear, independently from their degree), then the distribution of degrees observed for T is an estimate of the degree distribution of core routers.

3. Measurement

We present in this section practical measurements we conducted following our approach. We describe the whole procedure step by step, as well as the obtained dataset.

We first built an initial *target set* by sending (from a machine in our lab) a probe to addresses corresponding to 32-bit integers sampled uniformly at random. We stopped this process when we obtained correct answers (i.e., ICMP Destination Unreachable (Code 3/Port Unreachable) error packets) from three million such targets (we considered that no answer would arrive after one minute). This took approximately 10 hours.

Our initial *monitor set* was composed of the approximately 700 machines of the PlanetLab platform [25], which is a distributed infrastructure provided to researchers to conduct network experiments. Some of these potential monitors are of little interest (they have very poor connections, for instance, or they belong to networks that filter our probes) and some are colocated, therefore providing redundant information in our measurement. However, we present in the supplementary material several assessments of this monitor set, which all show that it fits our needs.

Given these initial target and monitor sets, we uploaded the target set to each monitor and remotely asked the monitors to send probes to all targets (in a random order to avoid situations where targets would receive many probes in a short period of time). This lasted approximately four hours (and so each target received at most 700 probes during this period, which is a reasonable load). In order to explore the stability of our measurements, we repeated this operation three times in a row. The whole measurement (building the target set and probing each target from each monitor three times) took less than 24 hours, with a very reasonable load for targets and monitors. At this stage, we obtained for each target its answers to the probes from all monitors (repeated three times), which we gathered onto a local machine for analysis.

We then applied a drastic filtering process (detailed in Section 6) in order to ensure we kept only data relevant to our needs: we removed monitors and targets that behaved incorrectly, as well as border nodes. We also conducted an auxiliary measurement able to obtain the set of all border interfaces visible from our monitors. Thanks to this, we were able to keep only target addresses that were core interfaces of core routers answering our probes correctly. Unsurprisingly, most target addresses belonged to border nodes. We finally obtained for each of our three measurements approximately 5600 targets belonging to reliable core routers. The output of our measurements is the observed degree of these routers, from which we will estimate the degree distribution of internet core routers.

We provide our measurement tools (source code and documentation), as well as the raw dataset at rmidd.complexnetworks.fr.

4. Unbiased Estimation

Based on the preceding procedure, we can achieve the crucial point of our method, namely estimating the degree of core routers sampled uniformly at random. Note that until now, we only sampled uniformly at random the addresses of their interfaces, not core routers themselves. Indeed, one has k possibilities to sample a router with k interfaces, so

high-degree routers appear in our target list with probability higher than low-degree ones. In order to correct this bias a posteriori, we discard from the result of the measurement the core routers whose address t present in the target set turns out to be the address of one of their border interfaces. After this discarding step, the probability that a core router has been sampled is proportional to its number k of core interfaces (which is precisely what we measure). Then, the observed fraction p'_k of routers of core degree k sampled with this bias is proportional to k times the fraction p_k of routers of core degree k sampled uniformly at random: $p'_k \sim k \cdot p_k$. As a consequence, we obtain:

$$p_k = \frac{p'_k}{k} \cdot \frac{1}{\sum_{i>1} \frac{p'_i}{i}},$$

where the second term is nothing but a normalization constant to ensure that $\sum_k p_k = 1$.

We then use this formula to estimate the true degree distribution p_k from the observed one p'_k.

5. Obtained Degree Distribution

The degree distributions observed from our three measurements after bias correction following the preceding formula are given in Figure 5(a). We plot the inverse cumulative distributions in Figure 5(b).

First, notice that the results of all measurements are very similar, which confirms that our results are stable in this setup. We present in Section 6 several other assessments of the quality of our final observation, all confirming that the obtained distributions are good approximations of the true one.

Obtained distributions show clearly that low-degree core routers are prevalent: approximately 75% of them have degree 2 only, and almost 20% have degree 3. This is not surprising, as we observe core interfaces only: these routers certainly have other interfaces connected to border routers and/or end hosts. The number of interfaces they use to actually route traffic in the core internet, however, is very low.

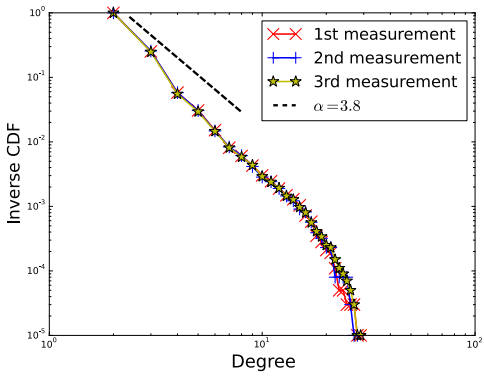
On the other hand, some core routers have much larger degrees, and the highest one we observe is 29. We may possibly miss a few interfaces of this router, but there is little chance that the true largest degree is much higher: we perform measurements from a much larger number of monitors, and so the fact that observed degrees are bounded by this number plays no role. Of course, core routers with degree significantly higher than 29 may exist, and they probably do.

There are, however, none in our random target set, and we therefore expect them to be extremely rare (which is reinforced by the sampling bias toward high-degree routers explained in Section 4).

	Measurement		
Degree	1st	2nd	3rd
2	0.74770	0.74371	0.75214
3	0.19434	0.19838	0.19258
4	0.02727	0.02727	0.02585
5	0.01551	0.01588	0.01486
6	0.00708	0.00640	0.00644
7	0.00206	0.00224	0.00230
8	0.00175	0.00196	0.00147
9	0.00127	0.00131	0.00145
10	0.00057	0.00044	0.00052
11	0.00056	0.00052	0.00047
12	0.00040	0.00044	0.00047
13	0.00020	0.00023	0.00017
14	0.00025	0.00031	0.00031
15	0.00032	0.00009	0.00017

	Measurement		
Degree	1st	2nd	3rd
16	0.00014	0.00025	0.00024
17	0.00023	0.00018	0.00015
18	0.00007	0.00007	0.00007
19	0.00007	0.00009	0.00009
20	0.00002	0.00000	0.00002
21	0.00008	0.00015	0.00008
22	0.00006	0.00000	0.00004
23	0.00000	0.00000	0.00002
24	0.00002	0.00000	0.00002
25	0.00000	0.00005	0.00002
26	0.00000	0.00002	0.00002
27	0.00002	0.00000	0.00002
28	0.00000	0.00002	0.00000
29	0.00002	0.00000	0.00001

(a)



(b)

Figure 5. (a) The degree distributions obtained from our three measurements (after bias correction): for each degree k, we give the estimated fraction p_k of core routers with degree k. (b) Plot of the inverse cumulative degree distributions obtained from our three measurements, after bias correction: for each value x on the horizontal axis, we plot the fraction of core routers having degree higher than or equal to x (log-log scale). We also plot the power law of exponent $\alpha = 3.8$ to show that obtained distributions are incompatible with a power law of exponent lower than this.

Going further, we observe that the first values of the obtained distribution (p_k for $k < 10$) are reasonably well fitted by a power law (a straight line in a log-log plot of the distribution). After that, the distribution experiences a sharp decrease. The first values are the ones that our method estimates best, and so one may ask if the obtained distribution is compatible with a power law. As highest degree may be underestimated, this may even be in accordance with the shape of the whole obtained distribution.

In order to explore this question, we compute a lower bound α for power-law exponents compatible with the first values (the most reliable ones). It is the slope of a straight line fitting the distribution in log-log scale. The exponent would clearly be larger than $\alpha = 3.8$; see Figure 5(b), which discards the usual assumption of an exponent close to 2. This also shows that if the true degree distribution is a power law, it is hardly distinguishable from an exponential decrease in practice [33], even for a system the size of the internet.

6. Supplementary Material

6.1 Proof of Concept

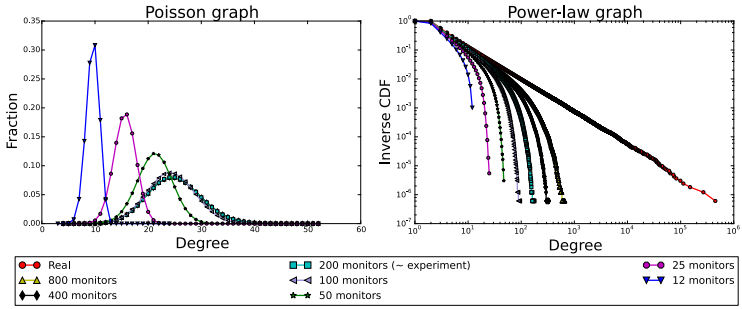
In order to assess the relevance of our approach, we conducted a comprehensive set of simulations, which we present in this section. Assuming that we are able to build appropriate sets of monitors and targets, the key questions we want to answer are, what is the risk that our estimate of a node's degree is different from its true degree? and how many monitors do we need to have an accurate estimate of the degree distribution?

To investigate this, we have conducted simulations as follows (see [34] for more details): we considered different kinds of artificial graphs to represent the network, we used as monitors random nodes with degree one (representing end hosts), and we used all core targets (i.e., nodes in the graph obtained by iteratively removing degree-one nodes). We then assumed that each target answers probes from each monitor using one (randomly chosen) of its interfaces that starts a shortest path from the target to the monitor. We used two different kinds of graphs: one with Poisson degree distribution, which is a typical homogeneous distribution, and one with a power-law degree distribution, which is a typical heterogeneous distribution. These two kinds of distributions are considered as extreme cases for what the true degree distribution may be.

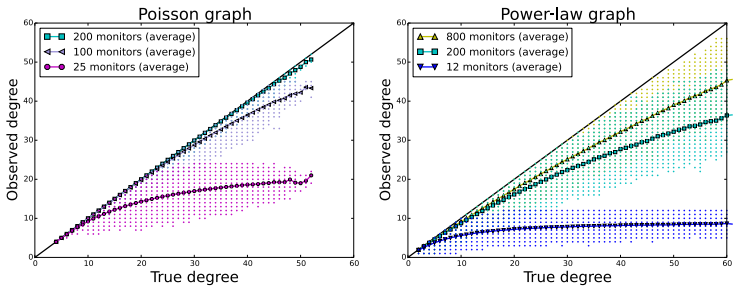
Figure 6 shows the results of the simulations for Poisson and power-law graphs of 2.5 million nodes. Figure 6(a) presents the degree distribution observed with, respectively, 12, 25, 50, 100, 200, 400 and 800 monitors. As one could expect, with 12 monitors the degree distribution is poorly estimated in the two cases. Nevertheless, it is remarkable that, even with this poor level of quality, the nature of the distribution (i.e., homogeneous or heterogeneous) appears clearly. When the number of monitors grows, so does the quality of the observed degree distribution.

With 200 monitors in particular, the observed and the true distributions become visually indistinguishable in the homogeneous case (left). For the heterogeneous case (right), one can observe a cutoff for

very large degrees. This comes from a limitation of our method: the observed degree cannot exceed the number of monitors, and more generally, the estimate becomes inaccurate for targets whose degree is close to the number of monitors. On the other hand, for reasonably low-degree targets, up to approximately 20, the observed distribution and the true one are visually indistinguishable with 200 monitors.



(a) Observed degree distribution



(b) Scatter plot of the true degree versus the observed degree

Figure 6. Simulations with different numbers of monitors (12, 25, 50, 100, 200, 400 and 800) over graphs of $2.5 * 10^6$ nodes whose degree distribution follows either a Poisson law with average degree 25 or a power law with exponent 2.1.

These last statements are strengthened by the plots in Figure 6(b), which shows the scatter plot of true degree (on the x axis) and observed degree (on the y axis) for all targets and for the two kinds of graphs. We can see that with 200 monitors, the estimate degree of each node is quite close to its true degree for the Poisson graphs, thus proving that our method performs very well on this kind of graph. As regards power-law graphs, we can see that using 200 monitors, the estimate degree of low-degree nodes is quite close to the true one. More than 95% of degree-2 nodes are correctly observed, and this proportion drops to 85% when considering all nodes whose degree is

lower than 10. This shows that, for this type of node at least, our method also performs very well on power-law graphs.

Therefore, the only limitation of our method in this theoretical setup seems to be the estimation of the degree of high-degree nodes in power-law graphs. Indeed, an intrinsic limitation of our method is that we cannot obtain a degree estimate larger than the number of monitors $|M|$. However, this limitation has to be put in perspective, as Figure 6(b) shows that, even if poorly estimated, they still cannot be confused with low-degree nodes. Whatever the number of monitors, the worst estimation (lower point on the y axis) grows as the true degree grows.

In conclusion, both for Poisson graphs and power-law graphs, the nature and the shape of the degree distribution are correctly observed even with a low number of monitors. In addition, the observed distribution quickly converges to the true one when the number of monitors grows. The true degree of low-degree nodes is correctly observed (also true for high-degree nodes in the homogeneous case), and a high-degree node is never observed as a low-degree node.

One may wonder if these results still hold for graphs of different sizes and with different parameters, average degree for Poisson graphs and exponent for power-law graphs. These questions were investigated in [34], as well as the influence of some other parameters of the simulations. It turns out that the conclusions we derive here are still valid for different sizes and parameters. In particular, [34] shows that the size of the graph has very little importance, if any, for the quality of the observation with a given number of monitors. Then, the conclusion obtained by simulations on graphs of a few millions of nodes still holds for graphs of the size of the internet.

6.2 Comparison with Traceroute Measurements

In this section, we deepen the comparison between our method and the classical traceroute method with regard to two criteria: the correctness of observed degree distribution and the load induced on the network by the measurement (number of probes sent). We simulate measurements with our method as in Section 6.1 but using only a restricted set of targets. To simulate traceroute measurements, we follow the method of [14]: we give to each link a weight $1 + \epsilon$, where ϵ is uniformly randomly chosen in $\left[-1/n, 1/n\right]$, which ensures with very high probability that there is a unique shortest path between any two nodes. Then, for each monitor we compute the shortest path tree from this monitor to all the other nodes of the network using Dijkstra's algorithm. From these trees, we extract the set of shortest paths from all monitors to all targets in the target list and we aggregate all these paths together into one graph, which is the map resulting from

the traceroute measurement, and on which the degree distribution is then observed.

We present here the results for two graphs on five million nodes of the same kind as those of Section 6.1: a Poisson graph of average degree 25 and a power-law graph of exponent 2.1. For clear comparison, we use the same set of monitors for both methods, composed of 200 monitors for Poisson graphs and 800 monitors for power-law graphs. Our method always uses 5000 targets, which is close to the number of correct core routers in the real-world measurement presented in Section 3. We simulate traceroute measurements with various numbers of targets, resulting in a different number of probes sent on the network (from the same number of probes as our method to a number 2000 times larger).

Figure 7 shows results for the Poisson graph. First, notice that our method accurately estimates the true degree distribution, while traceroute with the same number of probes completely fails. The distribution obtained with traceroute even looks closer to a power-law distribution than to a Poisson distribution, which is a known bias of the traceroute method [11, 14]. Using 10 times more probes does not significantly improve this situation. Only when using 500 times more probes than the number used by our method does the distribution observed by traceroute start to look like a Poisson distribution, even though it remains far from the true one. Still, even with a load 2000

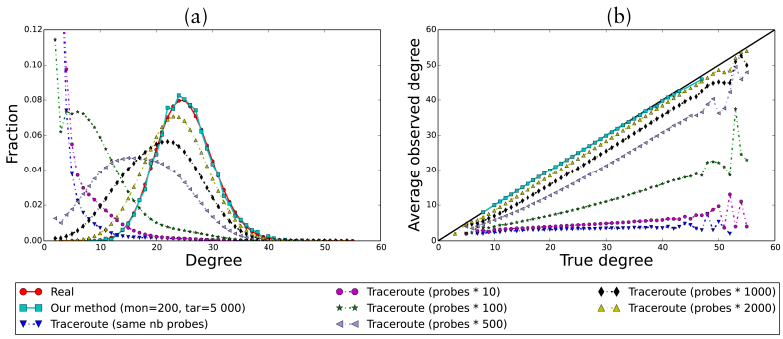


Figure 7. Comparison between our method and traceroute on a five-million-node Poisson graph of average degree 25. We use 200 monitors, and the comparison is done with regard to the number of probes sent. Our method requires 1 000 000 probes for 5000 targets (similarly to our real-world measurement). We compare it to the traceroute method when it is allowed to send the same number of probes (which results in 986 targets) and up to 2000 times more probes (which results in 1 972 000 targets). (a) True degree distribution and estimates obtained by both methods. (b) Average observed degree (y axis) as a function of true degree (x axis).

times larger, that is, probing about 40% of all nodes of the network, which is hardly possible to achieve in practice, the degree distribution observed by traceroute is clearly distinct from the true one.

Figure 7(b) explains this situation: it gives the average observed degree (y axis) for nodes of given true degree (x axis). The average degree observed by our method is almost indistinguishable from the true degree. Instead, degrees observed by traceroute measurement with up to 10 times more probes are barely correlated to the true degree: the average observed degree remains very low almost independently of the true degree. When the number of probes grows, the situation gradually improves, but even with a load 2000 times larger than the one of our method, traceroute still is less accurate than our method, therefore explaining that the distribution itself is not correctly observed.

Going further, let us mention that we pushed the number of targets used by traceroute up to 90% of all nodes. For this huge value only, which is infeasible in practice, the traceroute method performs as well as our method: average error made on observed degree of a node is 0.03 (0.04 with our method) and 97% of all nodes have their degree perfectly measured (96% with our method). With this number of targets, traceroute uses a number of probes 4500 times larger than our method (and the same number of monitors).

Figure 8 shows results for the power-law graph. Our method observes the degree distribution accurately for degrees up to 60, whereas traceroute obtains a much poorer estimate, even with up to 100 times more probes. However, when traceroute is allowed 500 times more probes than our method, it obtains a better estimate, which is visually almost perfect. Surprisingly, using even more probes then reduces the quality of the estimate, which finally becomes less accurate than our method. This is explained by Figure 8(b): even when traceroute obtains a good estimate of the distribution, this is not the consequence of an accurate estimate of individual node degrees. Therefore the good performance of the traceroute method is for some specific values of the number of targets only, and it is a side effect of its own bias. One cannot rely on such artifacts to properly estimate the distribution.

To deepen this, we show in Figure 8(c) the converse statistics: for each observed degree (x axis) we plot the average true degree (y axis) of nodes that were observed with this degree. Figure 8(d) gives the ratio between the average true degree and the observed degree. For traceroute with a load 500 times higher than our method, this shows that nodes of a given degree in the observed distribution have a true degree that is on average 2.5 to 3 times higher: nodes observed with degree 5 have on average a true degree above 12, nodes observed with degree 10 have on average a true degree above 25, and so forth. Our

method performs much better: the ratio between the average true degree and the observed degree remains close to 1 for most nodes, in particular for those having degree up to 20.

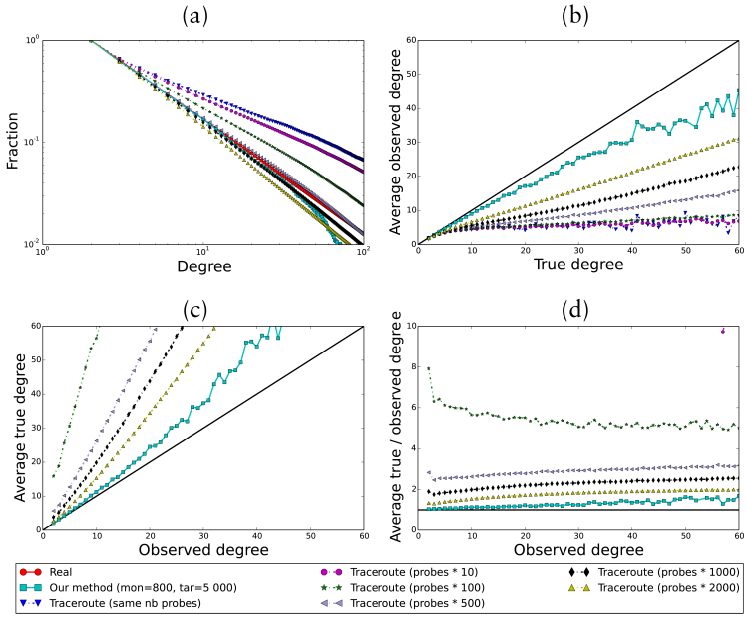


Figure 8. Comparison between our method and traceroute on a five-million-node power-law graph of exponent 2.1. We use 800 monitors, and the comparison is done with regard to the number of probes sent. Our method requires 4 000 000 probes for 5000 targets (similarly to our real-world measurement). We compare it to the traceroute method when it is allowed to send the same number of probes (which results in 1602 targets) and up to 2000 times more probes (which results in 3 204 000 targets). (a) True degree distribution and estimates obtained by both methods. (b) Average observed degree (y axis) as a function of true degree (x axis). (c) Average true degree (y axis) as a function of observed degree (x axis). (d) Ratio between average true degree and observed degree (y axis) as a function of observed degree (x axis).

When traceroute uses more than 500 times as many probes as our method, the obtained degrees become more accurate; see Figure 8(b). However, the accuracy of the observed distribution decreases at the same time. This is due to the fact that, even with 64% of the nodes as targets (i.e., a number of probes 2000 times larger than our method), the quality of degree estimates remains poor, and the ratio between average true degree and observed degree remains close to 2 for most observed degrees. Then, nodes of observed degree d with traceroute have a very different true degree. This is yet another demonstration of the issues of traceroute for degree distribution estimation.

Returning to the comparison between the two methods, let us mention that even when traceroute targets 90% of all nodes (i.e., 4 500 000 nodes here) it does not reach the accuracy of our method (although both use the same monitors, and our method uses only 5000 targets). For nodes of degree at most 60 (which represents 98% of all nodes), the average error made by traceroute on the degree of individual nodes is 1.52 and it perfectly measures the degree of 55% of these nodes. The average error for our method is only 0.74 and it perfectly measures the degree of 71% of these nodes.

6.3 Core versus Border

Intuitively, the border of the internet is the part of the network made of all trees connected to the rest of the network, which is called the core. More formally, the core, also called 2-core in graph theory, may be defined as follows. Consider the pruning process that iteratively removes all nodes of the network having degree exactly one, until there remains no such node. Border routers are the nodes removed during this process when it is applied to the physical internet graph, while core routers are the nodes that remain when the process terminates.

Then, by definition, core routers necessarily have more than one interface linking them to other core routers, and we call such interfaces *core interfaces*, their other interfaces being called *border interfaces*. On the other hand, note that any border node has exactly one interface directed toward the core of the network, namely the one that is linked to its unique neighbor when it is removed from the network during the pruning process. We also call core interface this unique interface of a border node and we call border interfaces all its other interfaces. The *core degree* (resp. *border degree*) of a node is its number of core (resp. border) interfaces.

6.3.1 Distinguishing between Core and Border Interfaces

We also conduct an auxiliary measurement in order to obtain for each monitor m the set of border interfaces it may see. To that purpose, monitor m iteratively sends k packets to k random addresses (for a given integer k) with increasing TTLs: the first k packets are sent with TTL 1, the k next packets with TTL 2, and so on. Thanks to the ICMP Time-Exceeded packets issued by the nodes at distance d from m (we discuss later the case of machines that do not send such packets), for each value d of the TTL, m discovers a set of interfaces at distance d from m. We denote this set of interfaces by $I_d(m)$. Let us denote by $\delta(m)$ the smallest d such that $|I_d(m)| > 1$, that is the first TTL at which m discovers more than one interface. We have by defini-

tion $\left|I_{\delta(m)}(m)\right| > 1$ and $\left|I_j(m)\right| = 1$ for all $j < \delta(m)$. The set of border interfaces visible from m is then precisely $\bigcup_{j<\delta(m)} I_j(m)$. All other interfaces visible from m are core interfaces belonging to some core router or belonging to some border node that is not on the path between m and the core of the network. Proceeding in this way for all monitors, we build the set $B(M) = \bigcup_{m\in M} \bigcup_{j<\delta(m)} I_j(m)$ of all border addresses that can be seen from them. Consequently, for each interface seen in the measurement, we are able to determine whether it is a core interface or a border interface: it is in the core if and only if it is not in the set $B(M)$ of border interfaces visible from M.

6.3.2 Recognizing Core Routers

Once we can distinguish between core and border interfaces, we can also distinguish between core and border routers. If a target address t belongs to a border node $r(t)$, our measurements are likely to see only one interface of $r(t)$. In some cases, we may see more than just one interface; see Figure 9. Indeed, $r(t)$ may be a router on the route between some monitor $m \in M$ and the core of the network. In this case, our measurement will also discover the border interface i of $r(t)$ that is directed toward m. By definition, this interface i belongs to the set $B(M)$ of border interfaces that are visible from the set M of monitors (see Section 6.3.1). The key point here is that if $r(t)$ is a border router then, from what precedes, it follows that our measurements see only one interface not in $B(M)$ for $r(t)$. On the other hand, if $r(t)$ is a core router, our measurement will discover at least two core interfaces of $r(t)$ (provided that M is of sufficient quality), which do not belong to $B(M)$ by definition.

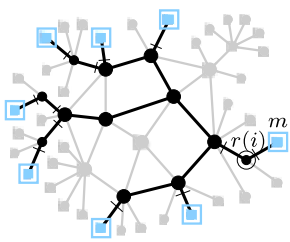


Figure 9. If we target an interface i that belongs to a border router $r(i)$, then our measurements may see more than one interface for $r(i)$, here two. However, only one of them does not belong to $B(M)$, as displayed in this graphic where all interfaces of $B(M)$ are marked with a small dash.

Thus, in the result of our measurement, we are able to distinguish which answering addresses belong to a core router and which of them

belong to a border router. And from Section 6.3.1, we can also determine for each answering address whether it is the address of a core interface or a border interface. This plays a key role for our unbiased estimation, detailed in the next section.

6.4 Uniformly Sampling Core Routers

Being able to sample a core router uniformly at random on the internet (recall that *uniformly at random* means that all possible elements are sampled with the same probability) is at the core of our approach. Unfortunately, there is no direct way to do so. Instead, it is straightforward to get addresses uniformly at random, as they are nothing but 32-bit integers. Of course, sampling such a random integer does not necessarily give a relevant address with regard to our measurement needs: this address may, for instance, be unallocated or belong to an end host or a router that does not answer our probes.

In this section, we first show how to sample uniformly at random an interface of an internet core router that correctly answers our probes, which we call a *correct core router*. From this sampling, which is not a uniform sampling of core routers themselves but only of their interfaces, we rigorously deduce an estimate of the degree distribution of all internet core routers. In other words, from the observed distribution resulting from a uniform sample of the interfaces of core routers, we deduce the observed distribution resulting from a uniform sample of the core routers themselves.

The general scheme of the way we proceed is as follows. We first perform a measurement on a target list uniformly sampled at random among 32-bit integers (see Section 3). Afterward, we select into this target list the addresses of core routers correctly answering our probes, which we determine thanks to the result of the measurement. Then, we restrict these results to the set of target interfaces that belong to correct core routers. Finally, we use the results for this set of targets only to infer the degree distribution of internet core routers.

For the rest of this section, we assume that given an address t, we are able to decide whether t belongs to a host that correctly answers our probes. We show how to do so in Section 6.5. From Section 6.3.2, we are also able to decide based on the result of the measurement whether a given address belongs to a core router or not. Consequently, extracting from our uniformly randomly generated target list the addresses that belong to a host that correctly answers our probes and that is a core router, we obtain a uniform sample of the interfaces of correct core routers.

This is not enough for our goal, as we need a uniform sample of (correct) core routers themselves, not just of their interfaces, as it turns out that when interfaces are uniformly sampled, routers are not.

Indeed, one has k possibilities to sample a router with k interfaces, so high-degree routers appear in our target list with probability higher than low-degree ones. This introduces a bias in the sampling of routers that one can correct if one knows for each router its number of interfaces. Unfortunately, our measurement does not provide this information but instead gives the number of core interfaces of each router (provided that the set of monitors is of sufficient quality).

In order to correct the bias in the number of interfaces of routers introduced by our target selection method, we introduce a supplementary bias in this method: we discard all the target addresses that are not core interfaces of core routers. We are able to do so as we can distinguish between core and border interfaces (see Section 6.3.1). Then the number of possible addresses to select a router so that it will still be in the target list after this last discarding step is no longer its number of interfaces but instead its number of core interfaces. The great benefit here is that, since our measurement determines the number of core interfaces of each core router, we are now able to correct the bias introduced by this target selection procedure.

The observed fraction p'_k of routers of core degree k sampled with this bias is proportional to k times the fraction p_k of routers of core degree k sampled uniformly at random: $p'_k \sim k \cdot p_k$. As a consequence, we obtain:

$$p_k = \frac{p'_k}{k} \cdot \frac{1}{\sum_{i>1} \frac{p'_i}{i}},$$

where the second term is nothing but a normalization constant to ensure that $\sum_k p_k = 1$. We may therefore use this formula to infer the true degree distribution p_k from the observed one p'_k.

In summary, our method to build target sets is as follows. We sample random 32-bit integers and we select the addresses that are core interfaces of core routers that correctly answer our probes. This procedure and the result of its application on our sample measurement are further detailed and discussed in Section 6.5.

6.5 Data Filtering and Processing

In this section, we describe step by step the way we process the raw data obtained from our measurement, containing some irrelevant or inappropriate data, in order to extract from it the part we use to faithfully estimate the degree distribution of internet core routers. The key numbers encountered during the different steps of this processing are summarized in Table 1.

		1st	2nd	3rd
Raw data	Nb running monitors	619	625	622
	Nb answering targets	2 849 740	2 734 548	2 699 642
Step 1	Nb targets giving multiple answers	10 150	9 842	11 048
Step 2	Nb monitors receiving answers from $\leq 80\%$ of targets	198	183	180
	Nb targets answering $\leq 80\%$ of monitors	590 605	527 346	544 252
Step 3	Nb interfaces in $B(M)$	1 040	1 107	1 097
	Nb targets having ≤ 1 interface not in $B(M)$	2 842 481	2 727 422	2 692 135
Step 4	Nb targets t such that $t \notin M(t)$	2 634 226	2 519 320	2 484 483
Processed data	Final number of monitors	421	442	442
	Final number of targets	5 593	5 623	5 619

Table 1. Key post-processing steps for our three measurements.

Step 0. Reserved addresses. As explained in Section 3, before the measurement starts, we build the list of targets by sampling uniformly at random addresses corresponding to 32-bit integers and by keeping the first three million of these addresses that answered the probe we sent to each of them from a single monitor. For the sake of completeness, let us mention that we actually apply one additional filter at this step: if the address sampled at random belongs to a known class of reserved addresses [36], then we simply discard it and pick another one at random. Thus, in the measurement itself, all the targets we use do not belong to such a reserved class of addresses (and they correctly answered the monitor we use in this step).

All the subsequent filters are based on the result of the measurement and are therefore applied afterward. The numbers of targets and monitors they apply to are given in Table 1.

Step 1. Targets giving multiple answers. Some targets in our list behaved incorrectly: they sent several answers to a unique probe sent by one monitor. As these targets do not behave correctly with regard to our measurement primitive, we simply discarded them and kept only those that sent a single answer to the probe of each monitor. The number of discarded targets is given in Table 1.

Step 2. Targets and monitors with only a few answers. Some targets answered a few monitors only, probably because of shutdowns during measurements, very low ICMP rate limiting or other specific reasons. Conversely, some monitors received surprisingly few answers, probably due to a very poor local connection, shutdowns or to the fact that PlanetLab machines may be overloaded (they are shared by numerous users). We plot these numbers in Figure 10, which shows that most monitors received answers from most targets, as we expected. In practice, we discarded monitors that received answers to fewer than 80% of their probes, and conversely all targets that sent answers to fewer than 80% of monitors. See numbers in Table 1.

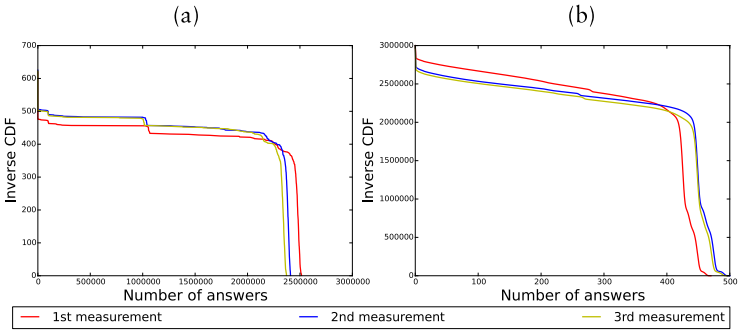


Figure 10. (a) For each number x on the horizontal axis, we plot the number y of targets that sent at least x answers to our probes. (b) For each number x on the horizontal axis, we plot the number y of monitors that received at least y answers to our probes.

Step 3. Recognizing core routers. The aim of this filtering step is to select only the addresses of the target list that belong to correct core routers, with the method presented in Section 6.3: (*i*) we build the set $B(M)$ of the border interfaces visible from our set M of monitors (see Section 6.3.1); and (*ii*) we keep only the target addresses t such that the set of interfaces $M(t)$ observed for t contains at least two interfaces that do not belong to $B(M)$ (see Section 6.3.2). The number of interfaces in $B(M)$ and the number of targets filtered, that is, that do not satisfy condition (*ii*), are given in Table 1.

Step 4. Uniform sampling of core routers. In order to correct the bias due to the fact that we uniformly sample interfaces instead of uniformly sampling routers, we perform a supplementary filter. This filter consists of discarding all addresses of the target list that are not addresses of core interfaces. The effect of this filter is to replace the bias mentioned by another one that we can rigorously correct (see Section 6.4). Note that this filtering step is independent of Step 3: they can be performed in any order, and even simultaneously, on the dataset. Table 1 gives the number of addresses in the target list that are filtered at Step 4, independently of Step 3.

It must be clear that a core router r may give incorrect answers to our probes. In particular, r may give no answer at all, or it may always answer using the same interface independently of the monitor. (Of course, more intricate behaviors are also possible, but they are very unlikely [37] and we ignore them here.) In the former case, there is only very little chance that an address of r is in our target list, as we target only addresses that answered one probe some hours before. Nevertheless, it may still happen that a router behaves this way during our measurement, and in this case it will be removed from the tar-

get list at Step 2. In the latter case, where the router r always answers using the same interface independently of the monitor, it will be filtered at Step 4. Conversely, if an address t of a correct core router r is in our target list, then our measurement sees at least two of the interfaces of r (as long as monitors are reasonably well distributed), and therefore t will successfully pass all filters. Then, our filtering process is successful in the sense that it is able to distinguish between correct core routers and other core routers.

Finally, note that there is no reason to assume that the degree of core routers is correlated to whether they answer our probes correctly or not. Indeed, low-degree core routers may a priori misbehave as well as high-degree ones, and conversely. As a consequence, the degree distribution of correct core routers, which we estimate here, is the same as the degree distribution of all core routers.

6.6 Quality of the Monitor Set

Our method relies on the use of a large set M of monitors scattered over the internet. It is crucial that this set is large enough, since the accuracy of the estimation of the degrees of targets highly depends on this number (see Section 6.1). On the other hand, having several monitors in the same location has limited interest: it is probable that most targets use the same interface to answer probes coming from these monitors (see Figure 11). Assessing the quality of a given set M of monitors (regarding our measurement goals) is therefore crucial, and we propose here three different and complementary approaches to do so.

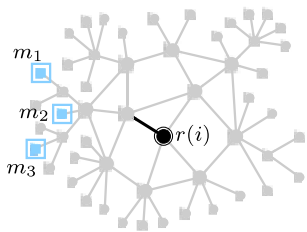


Figure 11. Three monitors m_1, m_2 and m_3 are actually colocated, and therefore they may observe a unique interface for any given target router $r(i)$. They are redundant regarding the quality of the measurement.

6.6.1 Colocated Monitors

When a packet sent from one monitor m, which is an end host, goes through the core of the internet, by definition of the core and the border (see Figure 4(a)), it always enters the core through the same router, which we call the *branching point* of m. Thanks to the auxil-

iary measurement method described in Section 6.3, any monitor may identify its branching point: the unique interface in $I_{\delta(m)-1}(m)$ is the (unique) interface of this branching point, which is directed toward m.

Now, let us consider two monitors m and m' such that $I_{\delta(m)}(m) = I_{\delta(m')}(m')$. In other words, the first time m and m' see several interfaces they see the exact same ones. Then certainly having both m and m' in the monitor set has little interest for our measurements: m and m' enter in the core internet through very close routers (probably through the same branching point; see Figure 11). We say that such monitors are colocated. The number of noncolocated monitors in M is a key value for estimating the quality of M: it basically represents the number of significantly different locations hosting monitors in M.

In the preceding analysis, we ignored machines that do not send ICMP Time-Exceeded packets. Because of them, we may erroneously decide that some monitors are colocated; this means that we underestimate the quality of our monitor set, which has no important consequence in our context: the quality is only underestimated. Similarly, it is possible that two monitors m and m' have different branching points but satisfy $I_{\delta(m)}(m) = I_{\delta(m')}(m')$. Again, this would make us underestimate the quality of the monitor set and therefore we may safely ignore this. Conversely, some monitors m and m' may have different but similar sets $I_{\delta(m)}(m)$ and $I_{\delta(m')}(m')$, indicating that they are not colocated but located close to each other. It may be interesting to use this for a more subtle assessment of the level of distribution of monitors, but we leave this for further work.

We used the method we just described and the auxiliary measurement described in Section 6.3.1 to identify classes of colocated monitors, which provide basically redundant information. We obtained 203 different classes, each containing on average 2.11 monitors. This is consistent with the fact that each institution involved in PlanetLab often contributes several monitors located at the same place. Examination of the DNS names of monitors belonging to a same class confirmed this: they typically match the same *.domain.tld pattern.

6.6.2 Diversity of Views

In this approach, we estimate an intrinsic quality of a monitor set M as the number of different locations hosting a monitor. A complementary view is obtained by evaluating the quality of a measurement from M toward targets in a set T. For instance, one may evaluate the quality of M as the number of distinct interfaces observed from M: $Q_0(M) = \sum_{t \in T} |M(t)$. Clearly, if $Q_0(M') > Q_0(M)$, then M' may be considered as better than M. More subtle quality functions may be

defined. In particular, it is interesting to take into account the fact that interfaces of low-degree routers are easier to observe than the ones of high-degree routers. This leads to the quality function $Q_1(M) = \sum_{t \in T} |M(t)| d(t)$, where $d(t)$ stands for the degree of target router $r(t)$. Of course we do not have the value of $d(t)$ but approximate it using the results of our measurements.

Given a quality function Q like the ones discussed, one may assess the impact of the addition of a new monitor m to the current monitor set by calculating $Q(M)$ and $Q(M \cup \{m\})$. Ideally, one wants to maximize Q to collect the most accurate set of observed interfaces while keeping M as small as possible to prevent redundant measurements (which may be costly).

In practice, we want to assess a given monitor set M, and to do so we start from an empty monitor set and compute the expected quality improvement when monitors are added one by one, in a random order. The quality is expected to grow with the number of monitors and then to reach a steady or almost steady regime, meaning that adding more monitors would not improve the measurement significantly. Of course, if many monitors are colocated (for instance, if they are all at the same location), the quality will have precisely this behavior (as adding more monitors at the same location does not significantly improve the measurement). This is why this quality function approach is complementary to the colocation-based one: we first perform the colocation and then plot the behavior of the quality function when noncolocated monitors are added.

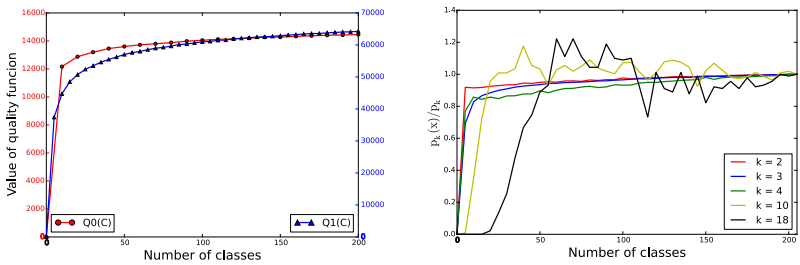
More precisely, once colocated monitors are identified, we proceed as follows: we first estimate the quality of the monitor set when only one colocation class is used, then two colocation classes, and so on, until all colocation classes (and thus all monitors) are used. We add colocation classes in a random order and average the obtained quality. The result is displayed in Figure 12(a). As expected, for both quality functions, the quality grows sharply at the beginning and rapidly converges. This indicates that adding more monitors at more locations would not improve the results much, and so that our monitor set and the number of locations hosting them are reasonable.

6.6.3 Convergence of Observations

Last but not least, a clear way to assess the quality of a given monitor set regarding our measurement objectives is to directly observe how the estimated fraction p_k of routers of degree k converges when the number of monitors grows, for all k. Here again, we expect these fractions to converge rapidly to a steady value, which is our final estimate. This would indicate that the last monitors we added were not necessary, and thus that we have obtained an accurate view. For the

same reasons already discussed, this is complementary to colocation analysis.

In order to examine the impact of adding more monitors at more locations on the estimated fraction p_k of core routers with degree k (which is what we are interested in), we proceed as follows: we add colocation classes one by one and observe how p_k evolves. Results are depicted in Figure 12(b). The estimates for small degrees rapidly converge, which was expected, as only a few monitors (and locations) are needed to correctly estimate them. Interestingly, only very few locations (approximately 10) are needed to obtain an estimate of p_k for $k < 5$ with 80% precision. Increasing the number of monitors rapidly improves the quality of the estimate. Even for large degrees, the estimate rapidly reaches a value comparable to the final one, despite the fact that it only slowly converges after that.



(a) (b)

Figure 12. (a) Evolution of the quality of the monitor set when we add colocation classes. (b) Ratio of the observed fraction $p_k(x)$ of routers of degree k with x colocation classes over the final obtained value p_k (with all classes).

6.6.4 Conclusion

Finally, this work on the monitor set shows that we have around 200 significantly different locations hosting monitors, and that this is sufficient to ensure a reasonable quality for our results. It is clear, however, that increasing the number of monitors and the number of locations hosting them would improve both the accuracy and the reliability of our estimates.

7. Related Work

The physical and IP-level internet structures have been extensively studied since the seminal papers of Pansiot et al. [38] and Faloutsos et al. [39]. The most classical approach consists of building maps from traceroute-like measurements. However, several studies have

shown that obtained maps are intrinsically biased [11, 14, 16–18, 22, 23, 40], and even that traceroute outputs are unreliable [17, 19, 20]. The hope that increasing the size and quality of maps would overcome these issues has led to much effort, but the situation remains far from satisfactory [12, 18, 40].

Conducting precise measurements of the degree of random nodes to obtain a reliable estimate of the degree distribution was first suggested in [14]. We explored the possibility to do so at the IP level in [35] but we only partly succeeded, and we conducted thorough simulations in [34]. Property-driven network measurements are also developed in other contexts, in particular online social networks (OSNs) [41, 42] and P2P overlay measurements [43].

Our work is also closely related to alias resolution (which plays a key role in the building of maps): while we seek all (unknown) interfaces of a given router identified by one of its interfaces, alias resolution aims at identifying in a given set of interfaces the ones that belong to a same router [37, 44–46]. Probes similar to ours are used in this context, in particular by the *iffinder* tool [47], as well as other techniques. Our use of such probes was clearly inspired by these works.

Finally, important efforts are devoted to the deployment of large and distributed measurement infrastructures, which are crucial for this field of research [24, 25, 27, 29, 30]. Some of them distribute monitoring capabilities at a huge scale (typically onto thousands of end hosts) and so are particularly promising for extending the work we present here [29, 30].

References

[1] R. Albert, H. Jeong and A.-L. Barabási, "Error and Attack Tolerance of Complex Networks," *Nature*, **406**, 2000 pp. 378–382. doi:10.1038/35019019.

[2] C. Magnien, M. Latapy and J.-L. Guillaume, "Impact of Random Failures and Attacks on Poisson and Power-Law Random Networks," *ACM Computing Surveys*, **43**(3), 2011 pp. 13:1–13:31. doi:10.1145/1922649.1922650.

[3] A. Barrat, M. Barthélemy and A. Vespignani, *Dynamical Processes on Complex Networks*, New York: Cambridge University Press, 2008.

[4] R. Pastor-Satorras and A. Vespignani, "Epidemic Spreading in Scale-Free Networks," *Physical Review Letters*, **86**(14), 2001 pp. 3200–3203. doi:10.1103/PhysRevLett.86.3200.

[5] A. Akella, S. Chawla, A. Kannan and S. Seshan, "On the Scaling of Congestion in the Internet Graph," *ACM SIGCOMM Computer Communication Review*, **34**(3), 2004 pp. 43–56. doi:10.1145/1031134.1031141.

[6] C. Gkantsidis, M. Mihail and A. Saberi, "Conductance and Congestion in Power Law Graphs," in *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, San Diego, New York: ACM, 2003 pp. 148–159. doi:10.1145/885651.781046.

[7] H. Haddadi, M. Rio, G. Iannaccone, A. Moore and R. Mortier, "Network Topologies: Inference, Modeling, and Generation," *IEEE Communications Surveys and Tutorials*, **10**(2), 2008 pp. 48–69. doi:10.1109/COMST.2008.4564479.

[8] H. Haddadi, S. Uhlig, A. Moore, R. Mortier and M. Rio, "Modeling Internet Topology Dynamics," *ACM SIGCOMM Computer Communication Review*, **38**(2), 2008 pp. 65–68. doi:10.1145/1355734.1355745.

[9] F. Tarissan, B. Quoitin, P. Mérindol, B. Donnet, J.-J. Pansiot and M. Latapy, "Towards a Bipartite Graph Modeling of the Internet Topology," *Computer Networks*, **57**(11), 2013 pp. 2331–2347. doi:10.1016/j.comnet.2013.04.007.

[10] X. Wang and D. Loguinov, "Understanding and Modeling the Internet Topology: Economics and Evolution Perspective," *IEEE/ACM Transactions on Networking*, **18**(1), 2010 pp. 257–270. doi:10.1109/TNET.2009.2024145.

[11] D. Achlioptas, A. Clauset, D. Kempe and C. Moore, "On the Bias of Traceroute Sampling: Or, Power-Law Degree Distributions in Regular Graphs," *Journal of the ACM*, **56**(4), 2009 pp. 21:1–21:28. doi:10.1145/1538902.1538905.

[12] P. Barford, A. Bestavros, J. Byers and M. Crovella, "On the Marginal Utility of Network Topology Measurements," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (IMW '01)*, San Francisco, New York: ACM, 2001 pp. 5–17. doi:10.1145/505202.505204.

[13] B. Krishnamurthy, W. Willinger, P. Gill and M. Arlitt, "A Socratic Method for Validation of Measurement-Based Networking Research," *Computer Communications*, **34**(1), 2011 pp. 43–53. doi:10.1016/j.comcom.2010.09.014.

[14] A. Lakhina, J. W. Byers, M. Crovella and P. Xie, "Sampling Biases in IP Topology Measurements," in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM 2003)*, New York: IEEE, 2003 pp. 332–341. doi:10.1109/INFCOM.2003.1208685.

[15] A. Mahanti, N. Carlsson, A. Mahanti, M. Arlitt and C. Williamson, "A Tale of the Tails: Power-Laws in Internet Measurements," *IEEE Network*, **27**(1), 2013 pp. 59–64. doi:10.1109/MNET.2013.6423193.

[16] P. Mérindol, B. Donnet, O. Bonaventure and J.-J. Pansiot, "On the Impact of Layer-2 on Node Degree Distribution," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC 2010)*, Melbourne, Australia, New York: ACM, 2010 pp. 179–191. doi:10.1145/1879141.1879164.

[17] M. Roughan, W. Willinger, O. Maennel, D. Perouli and R. Bush, "10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems," *IEEE Journal on Selected Areas in Communications*, **29**(9), 2011 pp. 1810–1821. doi:10.1109/JSAC.2011.111006.

[18] W. Willinger, D. Alderson and J. C. Doyle, "Mathematics and the Internet: A Source of Enormous Confusion and Great Potential," *Notices of the AMS*, **56**(5), 2009 pp. 586–599. www.ams.org/notices/200905/rtx090500586p.pdf.

[19] B. Donnet, M. Luckie, P. Mérindol and J.-J. Pansiot, "Revealing MPLS Tunnels Obscured from Traceroute," *ACM SIGCOMM Computer Communication Review*, **42**(2), 2012 pp. 87–93. doi:10.1145/2185376.2185388.

[20] F. Viger, B. Augustin, X. Cuvellier, C. Magnien, M. Latapy, T. Friedman and R. Teixeira, "Detection, Understanding, and Prevention of Traceroute Measurement Artifacts," *Computer Networks*, **52**(5), 2008 pp. 998–1018. doi:10.1016/j.comnet.2007.11.017.

[21] Y. Zhang, R. Oliveira, Y. Wang, S. Su, B. Zhang, J. Bi, H. Zhang and L. Zhang, "A Framework to Quantify the Pitfalls of Using Traceroute in AS-Level Topology Measurement," *IEEE Journal on Selected Areas in Communications*, **29**(9), 2011 pp. 1822–1836. doi:10.1109/JSAC.2011.111007.

[22] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez and A. Vespignani, "Exploring Networks with Traceroute-like Probes: Theory and Simulations," *Theoretical Computer Science*, **355**(1), 2006 pp. 6–24. doi:10.1016/j.tcs.2005.12.009.

[23] J.-L. Guillaume, M. Latapy and D. Magoni, "Relevance of Massively Distributed Explorations of the Internet Topology: Qualitative Results," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, **50**(16), 2006 pp. 3197–3224. doi:10.1016/j.comnet.2005.12.010.

[24] CAIDA. "Macroscopic Topology Measurements." (Jan 30, 2017) www.caida.org/projects/macroscopic.

[25] PlanetLab Consortium. "Planetlab: An Open Platform for Developing, Deploying, and Accessing Planetary-Scale Services." (Jan 30, 2017) www.planet-lab.org.

[26] B. Donnet and T. Friedman, "Internet Topology Discovery: A Survey," *IEEE Communications Surveys & Tutorials*, **9**(4), 2007 pp. 56–69. doi:10.1109/COMST.2007.4444750.

[27] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. E. Anderson, A. Krishnamurthy and A. Venkataramani, "iPlane: An Information Plane for Distributed Services," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI '06)*, Seattle, 2006, Berkeley: USENIX, 2006 pp. 367–380. dl.acm.org/citation.cfm?id=1298490.

[28] R. Motamedi, R. Rejaie and W. Willinger, "A Survey of Techniques for Internet Topology Discovery," *IEEE Communications Surveys & Tutorials*, **17**(2), 2015 pp. 1044–1065. doi:10.1109/COMST.2014.2376520.

[29] RIPE NCC. "RIPE Atlas." (Jan 30, 2017) atlas.ripe.net.

[30] Y. Shavitt and E. Shir, "DIMES: Let the Internet Measure Itself," *ACM SIGCOMM Computer Communication Review*, **35**(5), 2005 pp. 71–74. doi:10.1145/1096536.1096546.

[31] F. Baker, ed., *Requirements for IP Version 4 Routers*, US: RFC Editor, 1995.

[32] R. Braden, ed., *Requirements for Internet Hosts—Communication Layers*, US: RFC Editor, 1989.

[33] R. Perline, "Strong, Weak and False Inverse Power Laws," *Statistical Science*, **20**(1), 2005 pp. 68–88. doi:10.1214/088342304000000215.

[34] C. Crespelle and F. Tarissan, "Evaluation of a New Method for Measuring the Internet Degree Distribution: Simulation Results," *Computer Communications*, **34**(5), 2011 pp. 635–648. doi:10.1016/j.comcom.2010.06.006.

[35] C. Crespelle, M. Latapy and É. Rotenberg, "Rigorous Measurement of IP-Level Neighborhood of Internet Core Routers," in *INFOCOM IEEE Conference on Computer Communications Workshops,* New York: IEEE, 2010. doi:10.1109/INFCOMW.2010.5466707.

[36] E. Gerich, *Guidelines for Management of IP Address Space*, US: RFC Editor, 1993.

[37] K. Keys, "Internet-Scale IP Alias Resolution Techniques," *ACM SIGCOMM Computer Communication Review*, **40**(1), 2010 pp. 50–55. doi:10.1145/1672308.1672318.

[38] J.-J. Pansiot and D. Grad, "On Routes and Multicast Trees in the Internet," *ACM SIGCOMM Computer Communication Review*, **28**(1), 1998 pp. 41–50. doi:10.1145/280549.280555.

[39] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '99)*, Cambridge, MA, New York: ACM, 1999 pp. 251–262. doi:10.1145/316194.316229.

[40] M. Latapy and C. Magnien, "Complex Network Measurements: Estimating the Relevance of Observed Properties," in *Proceedings of the 27th Conference on Computer Communications (INFOCOM 2008)*, New York: IEEE, 2008 pp. 2333–2341. doi:10.1109/INFOCOM.2008.227.

[41] M. Gjoka, M. Kurant, C. T. Butts and A. Markopoulou, "Walking in Facebook: A Case Study of Unbiased Sampling of OSNs," in *Proceedings of the 29th Conference on Information Communications (INFOCOM 2010)*, New York: IEEE, 2010 pp. 2498–2506. doi:10.1109/INFCOM.2010.5462078.

[42] M. Kurant, A. Markopoulou and P. Thiran, "Towards Unbiased BFS Sampling," *IEEE Journal on Selected Areas in Communications*, **29**(9), 2011 pp. 1799–1809. doi:10.1109/JSAC.2011.111005.

[43] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen and W. Willinger, "On Unbiased Sampling for Unstructured Peer-to-Peer Networks," *IEEE/ACM Transactions on Networking*, **17**(2), 2009 pp. 377–390. doi:10.1109/TNET.2008.2001730.

[44] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet Map Discovery," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, New York: IEEE, 2000 pp. 1371–1380. doi:10.1109/INFCOM.2000.832534.

[45] M. H. Gunes and K. Sarac, "Resolving IP Aliases in Building Traceroute-Based Internet Maps," *IEEE/ACM Transactions on Networking*, **17**(6), 2009 pp. 1738–1751. doi:10.1109/TNET.2009.2014227.

[46] M. H. Gunes and K. Sarac, "Importance of IP Alias Resolution in Sampling Internet Topologies," in *IEEE Global Internet Symposium, 2007*, New York: IEEE, 2007 pp. 19–24. doi:10.1109/GI.2007.4301425.

[47] B. Huffaker, D. Plummer, D. Moore and K. Claffy, "Topology Discovery by Active Probing," in *Proceedings of the 2002 Symposium on Applications and the Internet (SAINT) Workshops*, Washington, DC: IEEE Computer Society, 2002 pp. 90–96.