Describing and Simulating Internet Routes [1]

Jérémie Leguay [2] Matthieu Latapy [3] Timur Friedman [2] Kavé Salamatian [2]

Abstract

This contribution deals with actual routes followed by packets on the internet at IP level. We first propose a set of statistical properties to analyse such routes, which brings detailed information on them. We then use the obtained results to suggest and evaluate methods for generating artificial routes suitable for simulation purposes. This also makes it possible to evaluate various network models. This work is based on large data sets provided mainly by CAIDA's *skitter* infrastructure.

I. INTRODUCTION

Realistic modeling of routes in the internet is a challenge for network simulation. Until now, one had to choose one of the three following approaches to simulate routes: (1) use the shortest path model, (2) explicitly model the internet hierarchy, and separately simulate inter- and intra-domain routing, or (3) replay routes that have been recorded with a tool like `traceroute` [1]. All of these methods have serious drawbacks.

The first method does not reflect reality: routes do not in general have the same properties as shortest paths, as already pointed out for instance by Paxson [2], [3], probably because of routing policies [4], [5] mainly at the autonomous system (AS) level. As described in detail recently by Spring et al. [4], and earlier by Tangmunarunkit et al. [6], [5], this often induces *path inflation*.

The second method is limited by our ability to explicitly simulate the internet hierarchy. Much work has been done to model the internet topology (see for instance [7], [8]), and much progress has been made, but today's topology generators are still capable of being highly inaccurate in capturing some parameters while they strive to adhere to others. (See, for instance, the findings in Li et al.'s Sigcomm 2004 paper [9].) Then, even if one is satisfied with the quality of the topology model, there is the question of simulating dynamic inter- and intra-domain routing. A non-negligible programming effort is required if the choice is made not to use a simulator, such as *ns* [10], that has these algorithms built in. Even here, the modeling issues are challenging.

Finally, the third method is not suitable if routes from a large number of sources are to be simulated. Today's route tracing systems employ at most a few hundred sources. CAIDA's *skitter* [11], [12] infrastructure, for instance, produces an extensive graph suitable for simulations, but it is based on routes from just around thirty sources. Moreover, such data is in general not publicly available, and collecting them is a difficult task.

Despite its well known drawbacks, and because of the lack of more accurate models, the shortest path model is generally used. Examples from recent years include Lakhina et al.'s Infocom 2003 paper [13], Barford et al.'s Sigcomm 2002 paper [8], Riley et al.'s MASCOTS 2000 paper [14], Guillaume et al.'s Infocom 2005 paper [15], and Clauset et al.'s STOC 2005 paper [16]. The *ns* network simulator documentation itself proposes to simulate routes by shortest paths as an alternative to simulate routing algorithms [10, Chs. 26, 29].

---

[1] A reduced conference version of this paper has been published in the proceedings of the international conference *Networking 2005*. This version is much more detailed, contains significantly more results, and we corrected a few mistakes.

[2] LIP6 – CNRS and Université Paris 6 – 4, place Jussieu, 75005 Paris, France – {jeremie.leguay,timur.friedman,kave.salamatian}@lip6.fr

[3] LIAFA – CNRS and Université Denis Diderot – 2, place Jussieu, 75005 Paris, France – latapy@liafa.jussieu.fr

This paper's principal contribution is a new approach to modeling routes in the internet, one that does not share the drawbacks just described. We suggest using an actual measured graph of the internet topology, such as the graph generated by *skitter*. From that topology, we suggest choosing sources and destinations as one wishes from the nodes of the graph. Between these sources and destinations, we then generate artificial routes with a model chosen to reflect statistical properties of actual routes.

Central to this contribution are two specific models for artificial route generation: the random deviation model and the node degree model. These models generate routes with relatively inexpensive calculations, and the routes that they generate better reflect the statistical properties of actual routes than does the shortest path model.

This paper's other contribution is to update measurements of some familiar statistical properties of real routes, notably path length and the hop direction, and to introduce and measure a new statistical property: the evolution of node degree along a route. These properties serve as the standard for evaluating whether simulated routes resemble real routes. By introducing this standard, this paper lays the groundwork for going beyond the work described here through the eventual introduction of yet better models.

The remainder of this paper is organized as follows. Sec. II describes the data set that we have used and the context in which our work lies. Sec. III proposes the set of statistical properties to describe routes in the internet. Sec. IV proposes the models we use to simulate routes based on these properties. Sec. V evaluates those models and the assumptions we made, and Sec. VI concludes the paper.

II. The framework

The ideal perspective from which to characterize routes in the internet would be from a snapshot of the routing tables of routers throughout the network. Unfortunately, such a snapshot is impossible to obtain on the scale of the entire network. In this section, we describe the alternative that we opted for, and the hypotheses we made.

A. The internet as a graph

Efforts to map the internet graph take place at three levels as Fig. 1 shows. One is the autonomous system (AS) connectivity graph, which can be constructed from BGP announcements (captured for instance by The Oregon Route Views Project [17] from peering arrangements with roughly 60 network service providers). The others are the router graph, where the nodes are the routers and the links are the physical connections between them, and the IP graph, where the nodes are the IP addresses and the links between them correspond to logical links (*hops* in the routing). Basically, in the IP graph two addresses are linked together if they belong to two routers with a link between them. The IP graph can be obtained using traceroute and similar tools from a number of different points in the network. To our knowledge, *skitter*, which conducts traceroutes from on the order of 30 servers to on the order of a million destinations, is the most extensive ongoing effort at the IP level. The router level has to be inferred from the IP level.

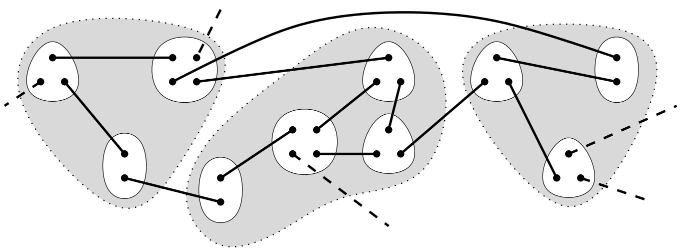

Fig. 1. *Three levels of the internet architecture. Each black dot represents an interface (*i.e.* an IP address). Several interfaces belong to each router, and several routers belong to each AS (the shaded areas). The (plain or dotted) lines correspond to physical links (always between two interfaces). They induce a graph over the set of interfaces, as well as graphs at the router level and at the AS level.*

Note that this separation into three levels is not exhaustive. One may consider the logical links between routers or the physical ones, for instance. One may also consider the physical links between interfaces.

It would also be possible to consider link-layer devices, such as hubs and bridges. The three-levels view however is a good approximation of what happens on the network layer, and will be sufficient for our purpose.

Let us insist on the fact that, because of the fully distributed nature of the internet, these graphs are not directly observable. In order to study them, one has to collect a large amount of information from various sources, and then recompose a (partial and possibly biased) view of the real graph.

Neither level is ideally suited to the task of modeling the behavior of routes at the router level. While the AS graph is directly based upon routing information, it is too coarse-grained to capture the details of path inflation. Moreover, a shortest path at the AS level does not necessarily correspond to a shortest path at the router or IP levels. As mentioned in the Introduction, simulators that do not explicitly model the AS hierarchy have been found by Tangmunarunkit et al. [7] to do better at generating graphs with desirable properties. Since our goal is to help in network simulations, we will therefore focus on the IP and router levels. Similar work should however be done at the AS level, and the comparison of the two would certainly be very interesting.

The main problem when using `traceroute` is that what one actually sees is the IP graph, while the graph of routers would be more relevant. One single node in the router graph appears as several separate nodes, one or more for each of its interfaces, in the IP graph. Moreover, `traceroute` captures *logical* links, which may miss the presence of tunneling, in ATM or MPLS subnetworks for instance. Ideally, then, one would construct the router graph using methods to "disambiguate" IP addresses, such as the alias resolution techniques described by Pansiot et al. [18], and by Govindan et al. [19] for *Mercator*. There are also techniques, such as those used by Spring et al. [20], [21], in *Rocketfuel*, and by Teixeira et al. [22], that take advantage of router and interface naming conventions to infer router-level topology from the IP one. Up to our knowledge, no study deals with the tunneling problem and other sophisticated bias.

Most of these disambiguation techniques, as applied for example in the *iffinder* tool from CAIDA [23], do not work by simple inspection of the IP graph; they require active probing, preferably simultaneously with graph discovery. This constraint makes extensive disambiguated router-level graphs much harder to obtain than IP graphs. At best, some core network topologies are available in this form thanks to Rocketfuel. But Rocketfuel is untested in stub networks. Finally, it is very difficult to judge the extent to which disambiguation is successful, and incomplete or incorrect disambiguation could introduce unknown biases.

To avoid these difficulties, we have restricted ourselves to the IP graph as obtained from *skitter*, and routes in this graph as obtained directly from `traceroute`. The resulting caveat is that the graph may not be properly representative of the router level graph.

This caveat is however mitigated by the fact that the IP graph nonetheless resembles the router graph in one important respect: except if we encounter tunneling, route lengths are preserved. That is to say that a route that has a given length in the router level graph has the same length in the corresponding IP graph. Furthermore, as Broido et al. note [24], "interfaces are individual devices, with their own individual processors, memory, buses, and failure modes. It is reasonable to view them as nodes with their own connections." Finally, we consider this work as a first step towards the accurate modeling of routes, and therefore prefer to make choices as simple as possible. We will see in Sec. V that these assumptions have little impact, if any, on our results.

B. The data set

This study uses *skitter* data from July 2nd 2003. The data was collected from 23 servers targeting 594,262 destinations, leading to 7,075,189 routes (not all sources probed all destinations) on that day. We obtained a graph by merging all these routes. We then removed invalid IP addresses [4] thus eliminating 3.95% of the edges and 3.25% of the nodes. The resulting graph contains $885{,}438$ nodes and $1{,}266{,}671$ links.

This graph captures well the small-world, clusterized, and scale-free nature of the internet already pointed out in numerous publications, see for instance Jin and Bestavros [26] and [27], [28], [29], [30], [31]. In particular, the average distance is approximately 11.4 hops, and the degree distribution is well fitted by a power law of exponent 1.97, see Fig. 2: the fraction of nodes of degree k is distributed as $k^{-1.97}$. This captures in particular the fact that, though most nodes have a low degree, there is a non-negligible number of nodes with very high degree. This graph also exhibits a high average clustering coefficient [5] of 0.035 (compared to 1.30×10^{-6} for a random graph of the same number of nodes and links). The fact that this graph shares properties common to most complex networks encountered in practice, as described for instance by Albert and Barabási [32] and Newman [33], will be useful for our characterisation of internet routes.

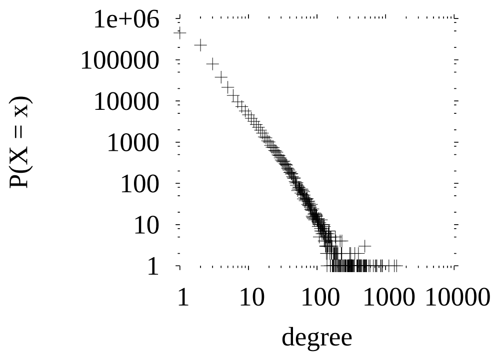

Fig. 2. *Degree distribution in the* skitter *graph.*

Notice that this graph is necessarily incomplete and biased due in particular to probing from a limited number of sources, to route dynamics, to tunneling and to erroneous or absent responses to traceroute probes. Biases of graphs induced by acquisition through a small number of traceroute monitors have been studied for instance by Lakhina et al. [13] and by Clauset et al. [16].

However, recent studies by Dall'Asta et al. [34] and Guillaume et al. [15] show that one may be quite confident of the accuracy, using this kind of exploration, of distances and degrees, which are the main properties that we use here. Moreover, *skitter* data represents the current state of the art in its extent and accuracy. We therefore consider this graph as a good approximate of the IP graph in this study, and will call it the *skitter* graph.

III. Statistical properties of routes

This section presents a set of properties for the statistical description of internet routes. These properties motivate the models of Sec. IV. Several properties have already been studied in previous works, and the work here serves to evaluate, update and complete them.

A. Route lengths

It is well known that routes are not shortest paths: they are not optimal in general. Fig. 3 shows the distributions of route lengths in our data set, and of the corresponding shortest paths. It also shows the distribution of the difference (*delta*) between the length of a route and the corresponding shortest path.

---

[4] We consider an address invalid if it belongs to the following subset of the special-use addresses described in RFC 3330 [25]: the private IP address blocks 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16, the link-local addresses in 169.254.0.0/16, the TEST-NET addresses in 192.0.2.0/24, the "this network" block 0.0.0.0/8, the loopback address block 127.0.0.0/8, the 6to4 relay anycast address block 192.88.99.0/24, the benchmark testing block 198.18.0.0/15, the multicast address block 224.0.0.0/4, and the reserved address block formerly known as the Class E addresses, 240.0.0.0/4, which includes the LAN broadcast address, 255.255.255.255.

[5] The clustering coefficient of a node is the probability that two randomly chosen neighbors of this node are linked together.

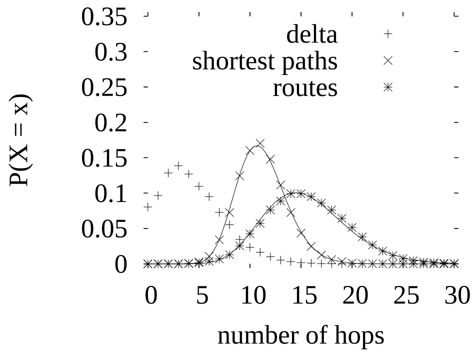

Fig. 3. *Length distributions of routes and shortest paths, and distribution of the difference between the length of each route and the corresponding shortest path length.*

These distributions are compiled as follows. For each route i obtained by `traceroute`, we compute its length ℓ_i and the length s_i of a shortest path between the source of the route and its destination. We also compute the difference, $\delta_i = \ell_i - s_i$.

The mean length of 15.57 hops for routes in this data set fits closely Paxson's observations [3], [2] on a data set that is older by nine years. The shortest paths have a mean length of 11.4 hops.

The distributions are well centered on their mean value: no route has a length more than twice the average. However, route lengths vary more around their mean, with a standard deviation $\sigma = 3.99$, than do shortest paths ($\sigma = 2.62$).

The delta distribution confirms Tangmunarunkit et al's observation [6], [5], mentioned at the beginning of this paper, that roughly 80% of routes are not shortest paths. In this particular data set, 19.34% of routes are shortest paths. Notice that, since the data is incomplete, there are undiscovered links, which implies that 19.34% is an overestimate: at least 80.66% of the considered routes are indeed longer than shortest paths in the true IP graph.

Route lengths and shortest path lengths are both well fitted by gamma distributions. Shortest paths have an estimated shape parameter of $k = 21.18$ and an estimated scale parameter of $\theta = 0.53$. Routes have $k = 14.56$ and $\theta = 1.07$.

Tangmunarunkit et al. also observed that 20% of routes were at least 50% longer than shortest paths. We find a somewhat larger portion: 33.4%. Again, this is a lower bound, and therefore the larger value may be due to a more accurate exploration.

One might wonder if the value of δ is correlated to the length of the shortest path, which would seem natural. For instance, routes between sources and destinations that are further apart may have a larger δ. We examine more closely the shortest path lengths between 9 and 16, which represent more than 85% of the cases. In this range, the mean value of δ is best fitted by the line $y = 0.13x + 1.46$ with an asymptotic standard error for both parameters under $\pm 13\%$, see Fig.4. Given this low slope and this standard error, it may be seen as almost flat, which contradicts the intuition: the value of delta does not depend significantly on the actual distance between the considered sources and destinations. Notice however that the mean hides considerable variations, which can be observed in the quantile plots in Fig. 4.

B. Hop direction

When a packet travels from one router to another, it may move closer to its destination, but it may also move further, or even stay at the same distance from the destination. Likewise, the distance from the source may increase, decrease, or stay constant. We will call these behaviors the *hop direction*, considered with respect to either the destination or the source. In principle, a hop should always increase the distance from the source and decrease the distance to the destination; in such cases, the route is a shortest path. Notice that hop directions in the IP graph correspond to the ones in the router graph, since distances are preserved between the two graphs.

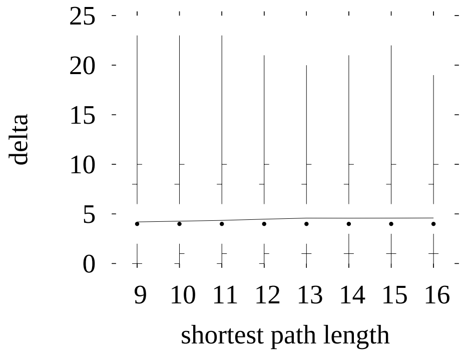

Fig. 4. *Quantile and average plots for δ for various shortest path lengths. Dots show the medians. Lines above and below show the top and bottom quartiles. Tick marks show the 95th, 90th, 10th, and 5th quantiles. The horizontal line shows the average value.*

Hop directions with respect to a given source may be computed for all the routes starting at this source using a breadth-first search rooted at it. This has a cost linear with respect to the size of the graph. Likewise, it is possible to study hop directions with respect to the destinations using a breadth-first search rooted at each destination. In our case, we have many destinations but only a few sources. Therefore, only hop directions with respect to sources can be observed in a reasonable complexity. We will restrict ourselves to this case in the following [6].

Examining the route traces, we found that 87.3% of hops go forwards, 4.6% go backwards, and 8.1% remain at the same distance from the source (we call these *stable* hops). More precisely, Fig. 5 shows the portion of forward, backward, and stable hops as a function of the hop distance for routes of 15 hops [7].

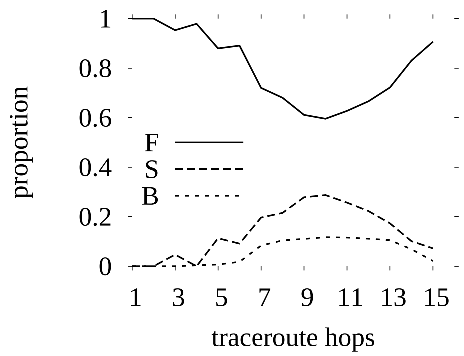

Fig. 5. *Hop directions along 15-hop routes (F: Forward, S: Stable, B: Backward).*

As one would expect, the first and last few hops are generally forward because there are few alternatives, if any. On the contrary, in the core of the network a significant proportion of the hops (more than one third) do not go further from the source. This type of behavior has already been described in the literature as a consequence of policy-based routing in the core of the internet. As Tangmunarunkit et al [6], [5] note, such behaviors may be induced by load balancing, commercial considerations, etc.

C. Degree evolution along a route

Recent work has shown that many real-world complex networks tend to have very heterogeneous degrees, well fitted by power laws. This is in particular true for the internet, as observed by Faloutsos et al. [27] and others. Moreover, most of the short paths between pairs of nodes in these networks tend

[6] Hop directions with respect to the destinations may be studied using only a part of all the destinations. But, since the number of sources is small, the approximation would be poor in this case.

[7] We choosed this length because it correspond to the most numerous routes, roughly $140{,}000$. The obtained plot is typical of what we obtained for any length. This will be true everywhere we will choose to focus on routes of a given length in the following.

to pass through the highest degree nodes. Actually, almost all paths (not only short ones) tend to pass through these nodes, which make them essential for network connectivity [35], [36], [37], [38], [39], [40].

These observations lead us to ask how the node degree evolves along a route. If routes tend to pass through high degree nodes, where do they do so, and what degree nodes do they encounter? Furthermore, does this tendency to pass through high degree nodes imply that, when a choice exists between next hops, the next hop that leads to the highest degree node is generally chosen?

Fig. 6 shows how node degrees evolve along routes of length 15 (notice the logarithmic vertical scale for the quantile plot). There is a significant increase in the degrees at the very begining of the plot, as well as a significant decrease at the end. In between, the plot is quite flat. This leads us to the following interpretation: the hosts have low degree, they are connected at their first hop router to relatively high degree nodes which play the role of access points, and then packets are routed in a core network where the degree (10 on average) does not depend much on the distance from the source or from the destination. Notice that the flatness in the middle of the plot does not mean that all the nodes in the core have a similar degree (the degrees in the core follow a power law). But, once a packet has entered this core, there is no correlation any longer between the degree of the node and the distance from the source or from the destination.

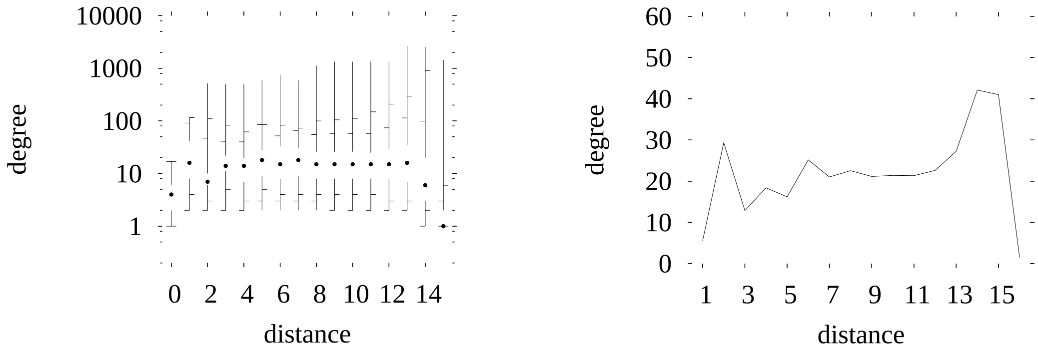

Fig. 6. Degree evolution along routes of length 15. Left: quantile plots (dots indicate the median, vertical lines run from the min to the first quantile, and from the third to the max, tick marks indicate the 5^{th}, 10^{th}, 90^{th} and 95^{th} percentiles). The median and lower quantiles do not appear on this plot for hop 15, as the median at that hop count is zero and the vertical scale is logarithmic. Right: the average value.

One may wonder if there is a simple local rule that can be observed for the degree evolution along a route. In particular, when there is a choice of next hop along a route, is there a correlation between the degree rank of the neighbors and their probability of being chosen? For instance, are highest degree nodes chosen preferentially over lower degree ones? Note that such a rule could be perfectly compatible with the observed flat degree evolution in the middle of routes.

Fig. 7 (left) plots the probability that a packet goes to a node's i-th ranked neighbor, where the neighbors are ranked from highest degree to lowest. We show the plots obtained for degrees 4 to 10, which are the cases where both the degrees and the number of nodes are non-trivial.

There is no apparent correlation in this plot, which seems to invalidate our hypothesis. However, if one considers only the neighbors of a node to which it sometimes sends a packet (in other words, we consider the *skitter* directed according to the ways the collected routes are traveled), then one obtains the plot on the right of Fig. 7. One can then see a clear bias towards highest degree nodes, though this bias is rather small.

IV. Route models

The previous section provides a set of statistical tools to capture some non-trivial properties of routes in the internet. We now propose three simple models (only two of which we eventually retain) designed to capture these features.

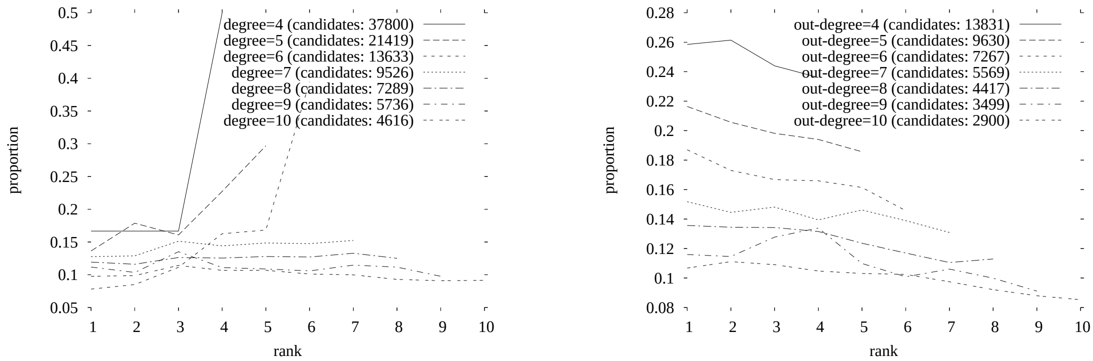

Fig. 7. Choice of next hop node as a function of this node's degree ranking. Left: on the (undirected) *skitter* graph. Right: with the directed vision.

Our approach is as follows: we design a model as simple as possible which focuses on one of the properties of interest, and then we use the other statistics to evaluate the model (in the next section). This ensures that the models stay very simple, and this makes it possible to study the relations between the observed properties (are they independant or on the contrary can some of them be seen as consequences of others?).

A. *Path length model*

The path length model is the simplest and the most obvious one conceptually, but it proves to be unusable in practice. The model aims at producing routes of the same lengths as real ones. As discussed in Sec. III, a real route length typically exceeds that of the shortest known path by some small integer value $\delta \geqslant 0$.

In order to construct a route from a source s to a destination d, the path length model first computes the length ℓ of a shortest path from s to d. Then it samples a deviation δ from a distribution such as the one shown in Fig. 3, and a route is generated by choosing a path at random from s to d among the ones which are loop-free and have length $\ell + \delta$. This ensures that the difference between shortest path lengths and actual route lengths will be captured by the model.

To choose such a path at random implies however that one must construct all the loop-free paths of length $\ell + \delta$ from s to d. In practice, the computation required to generate this number of paths may be prohibitive, since even in simple cases it is exponential in $\ell + \delta$. For example, in trying to generate all paths of length 21 between a pair of nodes in the *skitter* graph, we enumerated 1,206,525 possible paths. Therefore, despite its conceptual simplicity, we will not consider this model further.

B. *Random deviation model*

The random deviation model is based upon the idea that a route usually follows a shortest path, but might occasionally deviate from it. We modeled this using one single parameter, p, the probability at any point of deviating from the current shortest path to the destination, if such a deviation is possible.

A random deviation route from source s to destination d is therefore based upon a shortest path u from s to d. At each hop, with probability $1 - p$, the route continues along u. But with probability p it will, if possible, deviate off u to another path. A deviation from current node x to a neighboring node y is deemed possible only if there is a shortest path w from y to d that does not pass through x. Should there be a deviation, the route continues along w to d (unless another deviation should occur). The model is precisely described by Algorithm 1.

Fig. 8 shows an example of how a route can be generated using the random deviation model. In this graph, there is a five hop shortest path from source s to destination d. The route follows this path for three hops and then deviates at v. This deviation is possible because the shortest path from v' to d does not contain v. The resulting route is seven hops long.

Algorithm 1: rand_dev_route (G,s,d,p)

```
Input  : A network G, a source s, a destination d, a deviation probability p.
Output : An artificial route r from s to d in G, following the random deviation model with parameter p.
begin
    u ← a shortest path from s to d chosen at random;
    r ← empty list;
    v ← first element of u;
    copy v to the end of r;
    remove it from u;
    while v ≠ d do
        if rand[0,1] ⩽ p then
            C ← set of all the shortest paths from any neighbor of v to d;
            Remove from C the paths containing v;
            if C ≠ ∅ then
                u ← random element of C;
        v ← first element of u;
        copy v to the end of r;
        remove it from u;
    return r;
end
```

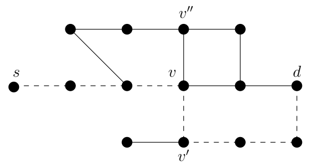

Fig. 8. A route (dashed lines) obtained using the random deviation model.

We can use Fig. 8 to illustrate some details of the random deviation model. It shows instances in which no deviation is possible. For example, there can be no deviation at the first node of the shortest path from s to d, since it has no neighbor that is not already on the shortest path being followed. Also, there can be no deviation at the second node, even though there is a neighbor that is not on the shortest path. The reason for this is that the only shortest path from this neighbor to d passes through the node we come from. The figure also shows an instance where two deviations are possible: at node v, deviations to v' and v'' are both possible. The choice of which to take (if any) is random.

Finally, notice that large numbers of routes to a given destination d can be efficiently generated with the random deviation model once a shortest path tree rooted at d has been computed.

C. Node degree model

Several previous authors, including [41], [36], [42], have tried to use the heterogeneity of node degrees to compute short paths in complex networks. The basic idea is that a path which goes preferentially towards high degree nodes tends to see most nodes very rapidly (a node is considered to be seen when the path passes through one of its neighbors).

The node degree model is based upon a similar approach, as follows. Two paths are computed, one starting from the source and the other from the destination. The next node on the path is always the highest degree neighbor of the current node. The computation terminates when we reach a situation where a node is the highest degree neighbor of its own highest degree neighbor. One can show that only this kind of loop can occur. Then, one of two cases applies: either the two paths have met at a node, or they have not. In the first case, the route produced by the model is the discovered path (both paths are truncated at the meet up node, and are merged). In the second case, we compute a shortest path between the two loops, and then obtain the route by merging the two paths and this shortest path, removing any loops. The overall model is precisely described in Algorithm 2.

Fig. 9 shows an example. There are three tree-like structures (the shaded areas). The source s belongs to the leftmost one, which is rooted at r_s, and the destination d to the rightmost one, with root at r_d. Each

Algorithm 2: node_deg_route (G,s,d)

Input : A network G, a source s, a destination d.
Output : An artificial route v from s to d in G, following the node degree route model.
Functions: reverse(p): returns the path obtained by reading p from the end to the beginning.
climb_degrees(G,v): returns the path in G obtained from v by going to the highest degree neighbor at each hop, until it loops.

begin
  $p_s \leftarrow$ climb_degrees (G,s);
  $p_d \leftarrow$ climb_degrees (G,d);
  if *p_s and p_d meet up* **then**
    let u be the first node they have in common;
    remove from p_s all the nodes after u;
    remove from p_d all the nodes after u;
    $p \leftarrow (p_s$,reverse$(p_d))$;
    return p;
  $q \leftarrow$ random shortest path from the last node of p_s to the one of p_d;
  $p \leftarrow (p_s,q,$reverse$(p_d))$;
  remove loops from p;
  return p;
end

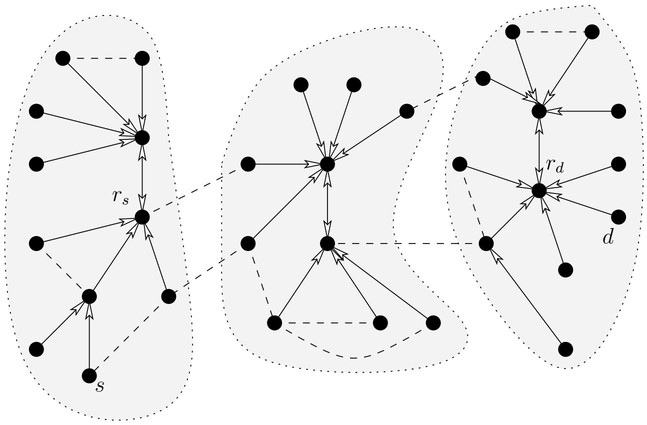

Fig. 9. The node degree model: an example.

directed link goes from one node to its highest degree neighbor (the dotted lines are links which do not satisfy this). When one wants to build a route from s to d according to the node degree model, one first finds the path from s to r_s, and the one from d to r_d. One then has to compute a shortest path from r_s to r_d, which has length 5 in this example. The final route is obtained by merging these paths, and then removing the loops (which leads to the removal of a link, in our example). It has length 7 (while the shortest path has length 6).

This method has already been proposed [42] as an efficient way to compute short paths in complex networks in practice: the obtained paths are very close to shortest ones. Moreover, the computation of the tree-like structure where each node points to its highest degree neighbor is very simple and only has to be processed once. Likewise, the shortest paths between a small number of loops are computed only once.

V. Evaluation

This section is devoted to the evaluation of the models we have just proposed, and to the discussion of their possible use. Our basic methodology to achieve this will be to compare the properties of the obtained artificial routes to the ones of the original routes. One therefore has to choose a graph on which the routes will be constructed, and then choose sources and destinations. This can be done in various contexts, from the ones closest to the original conditions to ones significantly different.

We will first generate routes on the *skitter* graph using the same sources and destinations as in the original data, and then using random sources and destinations. After this, we will use other maps of the internet with random sources and destinations, and finally we will run our models on the most widely used graph models of the internet. All of these experiments give some information on the behavior of our models, as well as on the relevance of the underlying graph.

In each case, we will compute a large number of artificial routes and study the same properties as the ones we studied on real routes. Therefore, the evaluation of each results is done by comparing the obtained plots to the ones in Sec. III, which we recall in Fig. 10.

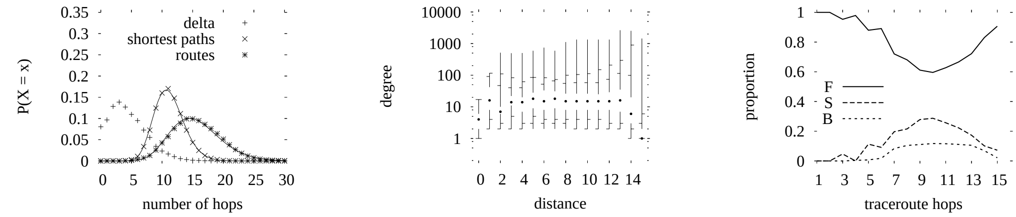

Fig. 10. Original *skitter* routes. From left to right: length distributions, degree evolution along routes, and hop directions.

Finally, the evaluation of the random deviation model depends on a parameter, namely the deviation probability p. We took the same value for all the experiments, $p = 0.2$, which was chosen to give the best fits when the random deviation model is compared to the original *skitter* routes on the *skitter* graph with the same sources and destinations. Tuning its value to the best fits in the other cases too would also be relevant, but we observed that the results do not vary significantly as long as the value is not too different. We therefore kept always the same value in order to make the presentation and the interpretation easier.

A. On the skitter *graph*

Fig. 11 and 12 show the obtained results with both models on the *skitter* graph, when one takes the very same sources and destinations as in the original data, and when one chooses sources and destinations at random, respectively.

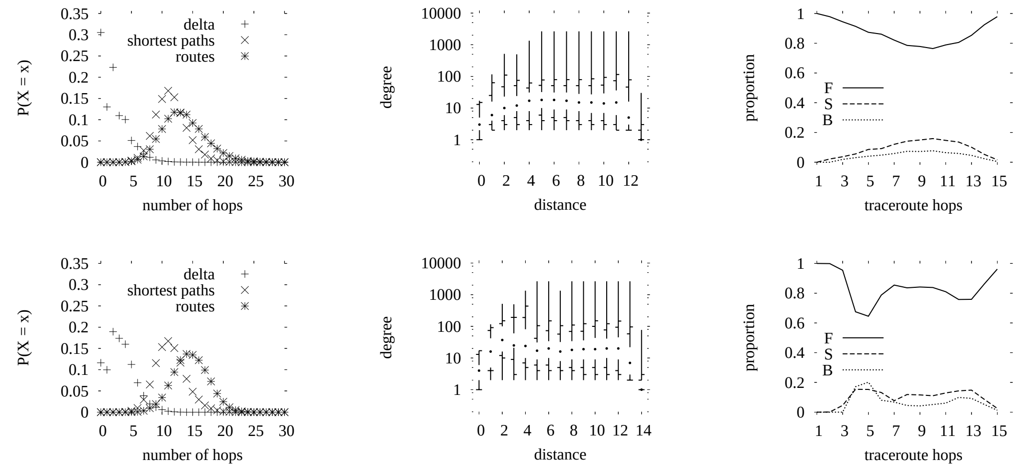

Fig. 11. Models on the *skitter* graph with the same sources and destinations as in the original measurement. Top: random deviation model. Bottom: node degree model. From left to right: length distributions, degree evolution along routes, and hop directions.

Before entering in more details, let us notice that the results seem very good: the global shape of all the plots fit quite well the original ones for both models, even when sources and destinations are taken at random.

The average route lengths are 13.6 with the random deviation model and 14.7 with the node degree model, when the sources and destinations are the original ones. They are 15.1 and 14.9 when sources and destinations are random. This is to be compared to the average shortest path length in this graph, 11.4,

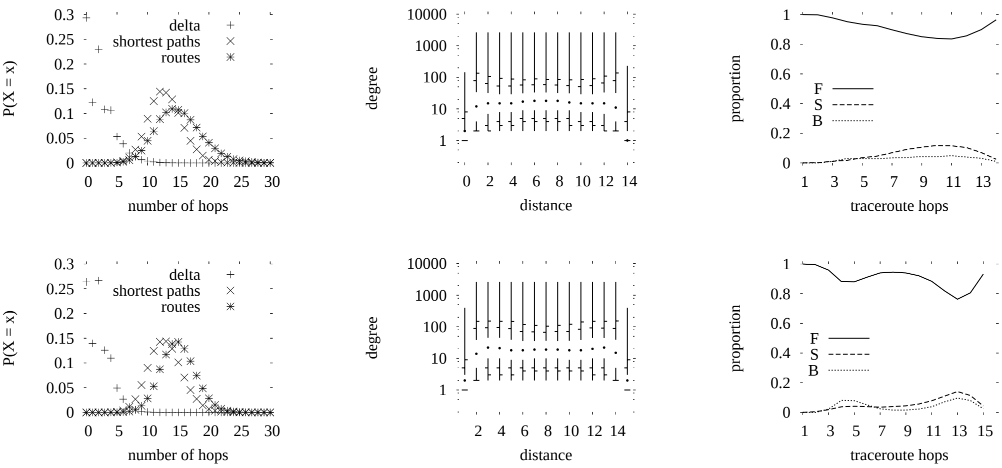

Fig. 12. Models on the *skitter* graph with random sources and destinations. Top: random deviation model. Bottom: node degree model. From left to right: length distributions, degree evolution along routes, and hop directions.

and to the average length of real routes, 15.6. We may conclude that the average route length is quite well captured, though not exactly.

In all the cases the route length distributions are symmetric, average somewhat higher than the shortest path distribution, and have tails similar to the actual route length distribution shown in Fig. 10. Lengths of paths generated with the node degree model tail off somewhat quicker than in reality (approaching zero closer to length 20 than length 25), but the degree of fidelity is nonetheless remarkable given that the length distributions are not explicitly part of the model. The random deviation model generates more routes that are shortest paths than in reality (roughly 30% compared to roughly 20%), whereas the node degree model generates somewhat fewer (roughly 12%).

As one may have expected, the node degree model performs better than the random deviation model in capturing the evolution of the degree along routes, especially close to the source. This is particularily true when using the same sources and destinations as in the original measurement. The difference is less significant with random ones. This indicates that there are more possible choices for routing close to the source, which is probably a bias due to the measurement itself (the map is more precise close to the sources than close to the destinations). The fact that the random deviation performs well on average (random sources and destinations) indicates that the shortest path to the destination generally goes to the highest degree neighbor. If the source is an original one, however, this is not true anymore and the node degree model performs better.

Now focusing on the hop directions in 15-hop routes, it appears clearly that the random deviation model behaves much better than the node degree model. Both capture qualitatively the properties of real routes, but the behavior of the random deviation model is very similar to the original one. Overall proportions of forward, stable, and backward hops closely match reality in both cases: 88%, 8% and 4%, and 84%, 9% and 7% for the random deviation model and for the node degree model when we take the original sources and destinations, and 89%, 7% and 4%, and 82%, 11% and 7% for the random deviation model and for the node degree model when we take random sources and destinations. The proportions for the original routes were 87% forward, 8% stable, and 5% backward.

Finally, we conclude that both the random deviation and node degree models do a reasonable job of emulating real routes, though each model captures some aspects better than others, and their strengths are different. The fits with original data are however surprisingly good, which makes them relevant in this context.

B. On other internet graphs

Another context in which our models may be used is when one has a map of the internet, collected in a previous work or provided by other researchers, and wants to use it for a simulation where routes are needed. One may then use our models to generate routes which will be more realistic than shortest paths usually performed in such cases.

We therefore ran our models on two internet maps provided by other researchers. For one of this map, both the router and the IP levels were provided. We considered this as an occasion to test the robustness of our models to a change from the IP level to the router one. Moreover, still for this dataset, the routes were also provided. Therefore, we computed the statistics on them. The results are presented and discussed in this section.

1) On the Mercator *graph:* The first case we will consider is the *Mercator* graph studied by Govindan et al. in [19] and provided at [43]. This graph was obtained in 1999 using `traceroute` massively from one source only but with *source routing*. Some antialiasing has been done in order to make it closer to the router graph. See [19] for more details. This data corresponds to the very beginning of the research on large scale internet topology; it may contain significant bias and errors, but it is still one of the very few maps publicly available, and it is widely used.

We ran our models on it and obtained the results in Fig. 13. Since the real routes used to construct this map are not available, we could not compare the artificial routes to them.

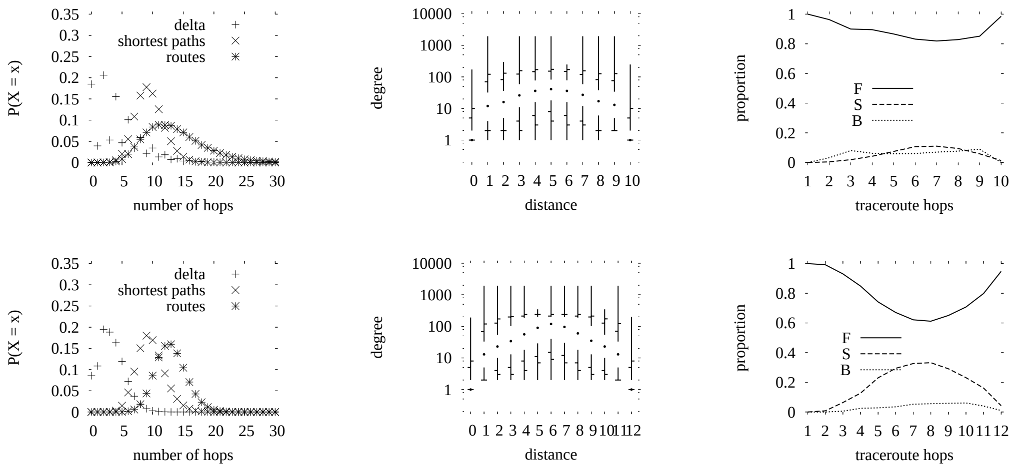

Fig. 13. Models on the *Mercator* graph with random sources and destinations. Top: random deviation model. Bottom: node degree model. From left to right: length distributions, degree evolution along routes, and hop directions.

The obtained results are in accordance with the properties of *skitter* routes concerning path lengths and hop directions. However, the degree evolution along routes is significantly different. We believe that this is due to the fact that, using only one source (and despite *source routing*), the graph has a tree-like structure with high degree nodes close to the root (*i.e.* the source of all `traceroute`). The routes therefore go up this tree, encountering nodes with higher and higher degree, and then go down to the destination. The non-trivial behaviors of route lengths and hop directions would then be a consequence of the links which prevent the map to be exactly a tree.

2) On the nec *graph:* Despite it is also obtained using massively `traceroute`, the measurement method is quite different for this graph provided by Magoni [44] and studied in [45]: it is based on the use of looking-glasses, which makes it possible to use several hundreds of sources. However, to avoid an overload of these sources, the number of destinations also has been reduced to a few hundreds. Moreover, many destinations are routers, whereas in the other maps they generally are hosts. As we will see, this has important consequences on route properties.

This dataset however has the important advantage of being available both at the router level and at the IP one [44]. Moreover, Magoni provided us with the actual routes he used to contruct it. This gives us the opportunity to study the statistical properties of these routes, just like we did with the *skitter* ones. It also makes it possible to compare the properties of interest at the IP and router levels.

We plot the properties of these real routes at IP level in Fig. 14, and at router level in Fig. 15.

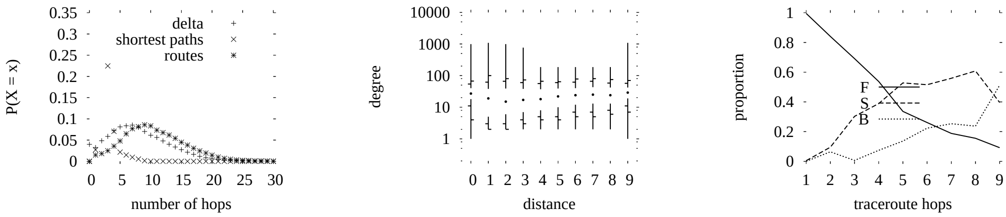

Fig. 14. Original *nec* routes at IP level. From left to right: length distributions, degree evolution along routes, and hop directions.

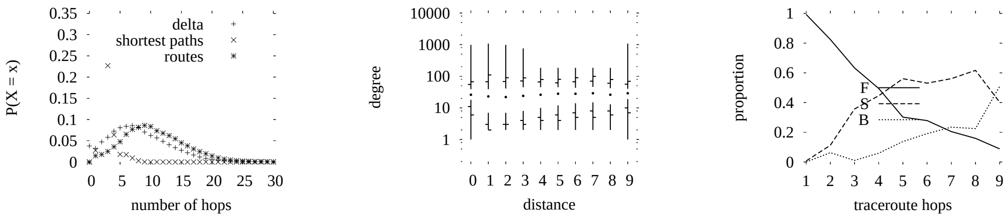

Fig. 15. Original *nec* routes at router level. From left to right: length distributions, degree evolution along routes, and hop directions.

One may be surprised by the fact that the properties of these real routes differ significantly from the ones of *skitter*: the lengths are smaller, the degree does not grow rapidly at the beginning of the route and does not decrease rapidly at the end, and, even more strikingly, many (and even most) of the hops are not *forward* at the end of the route. This can however be explained simply by two complementary facts. First, the destinations of these routes often are routers (not hosts), which is equivalent to say that these routes are only the beginning of host-to-host routes (unlike the *skitter* ones). Moreover, the neighborhood of the destinations is much better explored than in the *skitter* graph because of the large number of sources. Therefore, it is more dense, and this makes the number of *forward* hops decrease. These plots are therefore in reasonable accordance with the ones we obtained for *skitter* routes.

Moreover, they allow us to check an important assumption we have made at the beginning of the paper: the fact that the plots at IP and at router levels are very similar tends to confirm that our choice to stay at the IP level has little influence on our results, which may also be valid at router level.

In order to push further the evaluation of our models, let us now study the properties of artificial routes generated using them, from random sources and destinations (the use of the same sources and destinations as in the original data give very similar results, therefore we do not present them here), both in the IP and in the router *nec* graphs, see Fig.16 and 17.

Again, these plots confirm that, as long as one is concerned with the simple statistics and models we propose here, there is no significant difference between the IP and the router levels. Moreover, one can see that the models do their best to simulate host-to-host routes, and therefore produce routes which are much more similar to the *skitter* routes than the original *nec* routes. This may be considered as a good point for our models, which may be applied on other graphs than the *skitter* one and which are able to use the properties of the underlying graph to produce *realistic* routes.

C. On synthetic graphs

Until now, we only tested our models on real-world mappings of the internet topology. They proved to be usable in this context. They would however be even more interesting if one could use them on synthetic graphs obtained with models of the internet topology. This subsection is devoted to this case.

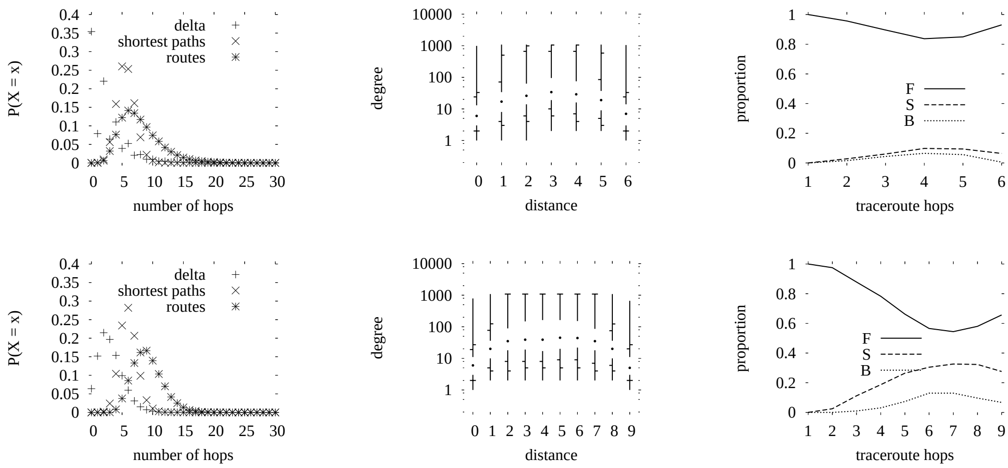

Fig. 16. Models on the *nec* graph at IP level with random sources and destinations. Top: random deviation model. Bottom: node degree model. From left to right: length distributions, degree evolution along routes, and hop directions.

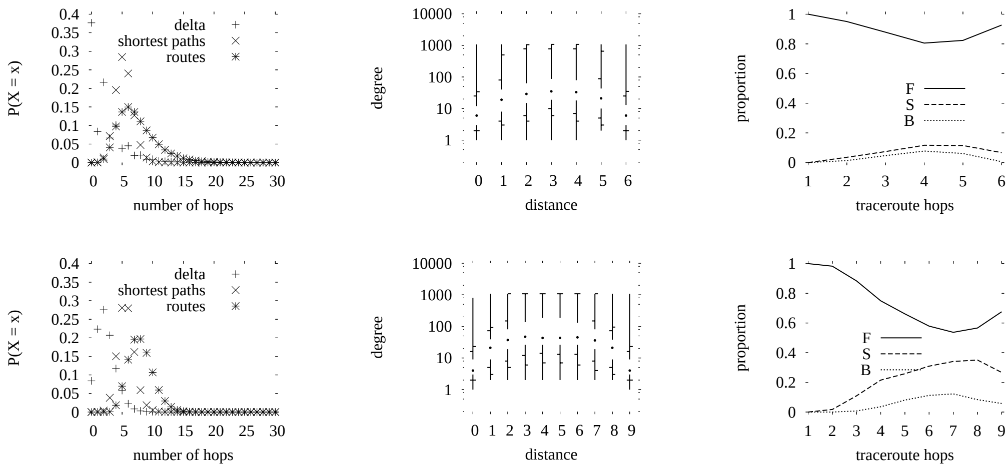

Fig. 17. Models on the *nec* graph at router level with random sources and destinations. Top: random deviation model. Bottom: node degree model. From left to right: length distributions, degree evolution along routes, and hop directions.

Since a few years, much effort has been done to propose accurate models of the internet as a graph. However, the problem is challenging, and there is nowadays no consensus on such a model, or even on a family of such models. A few basic approaches however appear as building blocks for more intricate models, and are now widely used. We will focus on them, as they are sufficiently simple to understand the behavior of our route models. We will point out some difficulties which will occur with any model based on these approaches, as we will see with the case of BRITE.

1) On purely random graphs: We begin the evaluation of our route models with the most simple model, the classical random graphs from Erdös and Rényi [46], [47]. Such a graph is constructed from n disconnected nodes by adding links between m randomly chosen pairs of nodes. Here, we took for n and m the same values as in the original *skitter* graph, in order to have a random graph comparable to this original one.

It is well known that the internet is significantly different from a random graph, in particular concerning its degree distribution (see for instance [27]). We consider this model as an interesting case however because it is the simpliest and it is often used as a building block of more intricate models.

Fig. 18 shows the results obtained with our models on such a graph (they are representative of all the experiments we ran on such graphs). The sources and destinations are chosen at random.

Both the degrees and the shortest path lengths in a random graph are very homogeneous [47], [48]: all the nodes have a degree close to the average value, and all the pairs of nodes are at a distance close to

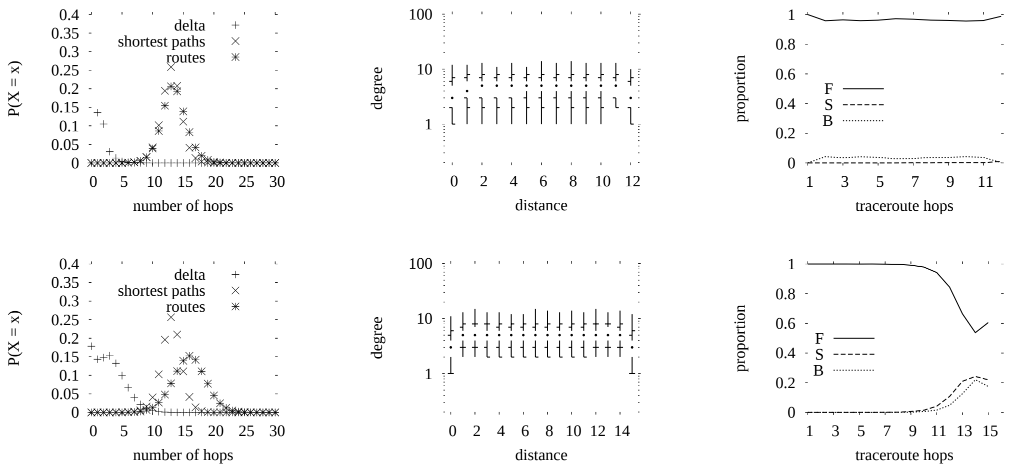

Fig. 18. Models on a purely random graph with random sources and destinations. Top: random deviation model. Bottom: node degree model. From left to right: length distributions, degree evolution along routes, and hop directions.

the average distance. This is confirmed by the plot of the shortest path length distribution. Moreover, with each model, the degree along a route is very stable due to the low variability of degrees in the graph: the first and last nodes have the average degree since they are chosen at random, and all the nodes in between are chosen with a probability proportional to their degree, which explains that their degree is larger than the average degree but quite stable.

The random deviation model produce routes with very rare deviations, since most of the time no deviation at all is possible because of the low average degree of nodes (no deviation at all is possible if the degree of a node is lower than 3, which is often the case as one can check on the plot of the degree along the routes). Therefore the routes produced by this model are mostly shortest paths, which explains the statistics.

The node degree model produces routes with properties closer of the ones of real routes: the length distribution is different from shortest paths, and not all the hops are forward. One can have quite a precise idea of the structure of the produced routes by noticing that, since all the degrees are close to the mean value, the route rapidly reaches the place where it becomes a shortest path. Therefore, a route produced by this model is nothing but very few hops towards higher degree nodes, then a shortest path, and again a few hops to degrees in decreasing order to the destination. This explains the fact that the length distribution of these routes is close to the one of shortest paths, it describes precisely the degree evolution along routes, and finally it explains the observed hop directions.

In both case, the produced routes are quite different from real ones. Since the underlying graph has properties qualitatively different from the ones of the IP graph, this can not be seen as surprising.

2) On scale-free graphs: We now examine how the models behave on scale-free graphs, *i.e.* graphs with a power law degree distribution as obtained using the Albert and Barabási model, see [49], [32]. Such a graph is constructed by adding nodes one by one until we have the wanted number of nodes, each new nodes being linked at random to k pre-existing nodes with a probability proportional to their degree. The value of k is chosen in order to induce the wanted number of links at the end of the construction (it is half the average degree).

Tangmunarunkit et al. found [7] that power law based generators create topologies that better match the internet's topology than do other common sorts of graphs, such as those produced by explicitly hierarchical topology generators. Despite this model remains very simple, it may therefore capture important features of the internet topology, and the models we proposed may be relevant on it. Morever, it is very often used to model the internet, see for instance [34], [37], and as a building block for more accurate models (see below).

Again, we choosed the parameters to fit the number of nodes and links of the original *skitter* graph

($k = 1.4$), in order to obtain a comparable graph. We choosed sources and destinations at random, and obtained the results plotted in Fig. 19. They are representative of all the experiments we ran on such graphs.

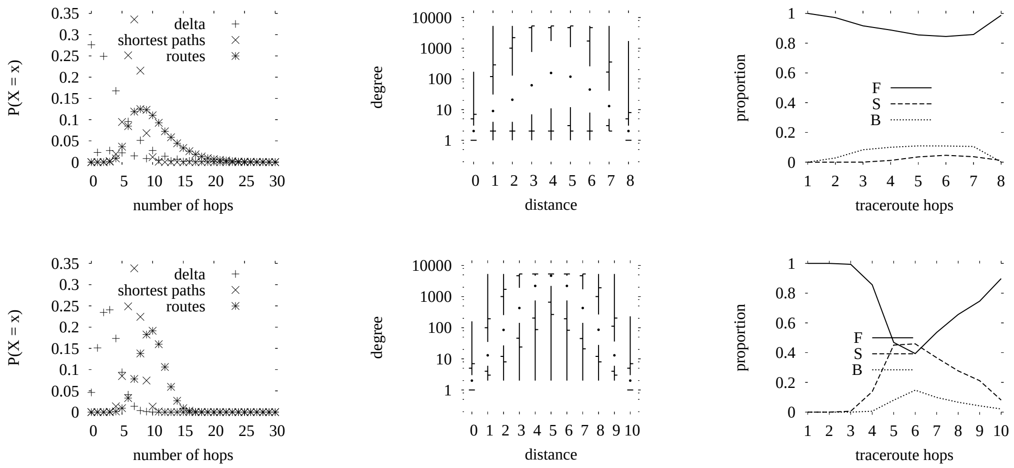

Fig. 19. Models on a scale-free graph with random sources and destinations. Top: random deviation model. Bottom: node degree model. From left to right: length distributions, degree evolution along routes, and hop directions.

First notice that scale-free graphs have a very low average shortest path length in general, see for instance [50], [51], [52], here 7.7, as can be seen in the length distributions. The models produce longer routes, but they remain quite short. This leads us to consider statistics on routes of length 8 or 10 depending on the model, which are the most numerous.

Both models clearly fail in capturing the degree evolution along routes in such a graph. The highest degree nodes are always reachable, as can be seen on the plot of the degree evolution along routes from the node degree model. This induces a regular increase in the degrees along such routes until a very high degree node, and then a decrease until it reaches the destination. Notice also that a random deviation tends to go towards high degree nodes (they have more links and thus a randomly chosen links has a high probability to be connected to such a node), which explains the degree evolution along routes from the random deviation model.

Finally, let us observe that the random deviation model captures surprisingly well (compared to the other statistics and to the other model) the hop directions. It might be seen as a consequence of the fact that at the beginning and at the end of routes one has very few choices for the next hop (low degree nodes) while in the middle there are many choices. The hop directions in the node degree model show that going to the highest degree neighbor at each hop may be a very bad strategy in the core of the graph.

3) On BRITE *graphs:* BRITE [53], [54] is one of the most widely used models in network simulation, in particular in internet simulation. We therefore used it to generate a variety of graphs supposed to be good approximations of the *skitter* graph (in terms of size and degree distribution at least), and ran our models on them.

Two cases should be considered:

- *a flat topology*, which is nothing but a scale-free graph as described above. However, since BRITE needs an integer value for the number k of links added at each step (the original definition of the model did not specify what to do when k is non-integer), we had the choice between $k = 1$ and $k = 2$. In the first case, one obtains a tree, in which each model produces nothing but shortest paths. We therefore obtain trivial statistics (length distributions are the same, degree evolutions grow to a maximal and then decrease, and there are only forward hops). We therefore took $k = 2$ and then obtained results very similar to the ones described above for $k = 1.4$. Therefore, we do not detail expermients on flat topologies here.

- *a hierachical topology* with nodes distributed in Autonomous Systems. BRITE first generates the AS topology with the scale-free model already described, and then the topology inside each AS is generated using this model again. The obtained degree distribution is a truncated power law, meaning that the degree are heterogeneous but there is no node with very high degree. We generated such a topology with $n = 900{,}000$ nodes distributed in $9{,}000$ Autonomous Systems (100 routers per AS). At the AS level we choosed $k = 10$ and inside each AS $k = 1$. This leads to an average degree 2.2 approximately. One may also use the purely random model at a level or the other, or both. We present here the parameters which gave the better results, plotted in Fig. 20.

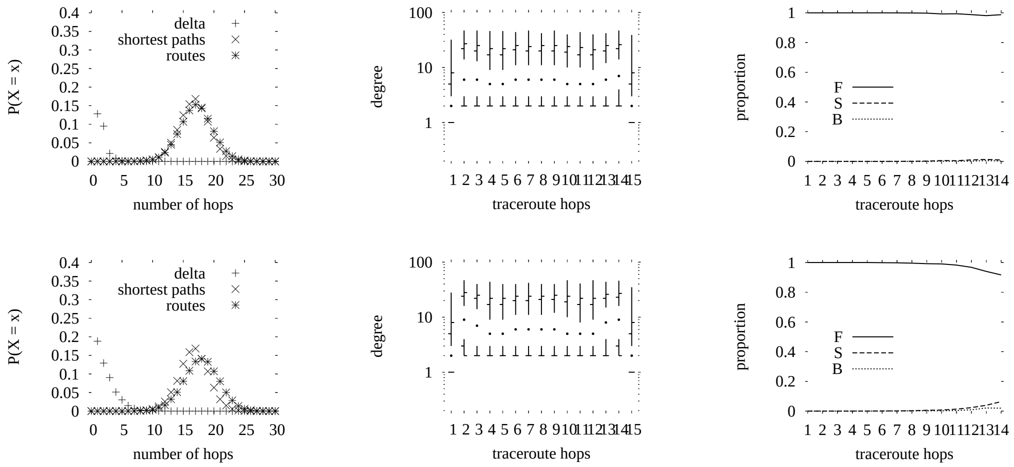

Fig. 20. Models on a scale-free graph with random sources and destinations. Top: random deviation model. Bottom: node degree model. From left to right: length distributions, degree evolution along routes, and hop directions.

The performances obtained in these experiments are very poor. The fact that $k = 1$ inside each AS induce that these graphs are trees. Therefore, most routes actually are shortest paths, which explains the statistics. Larger values of k should be considered, but BRITE forces them to be integers, and $k = 2$ gives an average degree too large for internet modeling. Moreover, one can clearly see on the plot of the degree evolution that the two-level structure induce a quasi-periodic variations on the nodes degrees which does not fit the properties met in practice.

VI. Conclusion and Discussion

The first contribution of this paper is to provide a vocabulary for describing routes in the internet, and to use that vocabulary to describe routes in one of the largest and most complete data sets currently available.

The characteristics we have used to describe routes are: their lengths, and the differences between those lengths and the lengths of corresponding shortest paths; the direction of hops along a route; and the evolution of the degree of nodes along a route. We have chosen these characteristics based upon graph theoretic knowledge of the typical properties of real-world complex networks graphs, of which the internet is an example. Let us notice that these characteristics are very general and may be used (and extended) with benefit in other complex network studies: until now, no statistical tool had been proposed to describe large sets of paths in such networks.

Other graph theoretic characteristics may also be studied in the manner we have done here. The evolution of the clustering coefficient of nodes along a route would be a natural candidate, for instance. One may also study the link clustering coefficient:

$$\mathrm{cc}(u,v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}.$$

Other interesting perspectives are to consider the routes as *directed* (from sources to destinations), the links as *weighted* (by the measured delay), and to take into account the dynamics of the internet and its routes. Paxson [2], [3] and, more recently, Amini et al. [55] have characterized the asymmetry of routes in the internet. Likewise, we have focussed on the topological characteristics of internet routes. Could we tie this in to the considerable body of knowledge concerning the delay characteristics of routes? Savage et al. [56] and Spring et al. [4], for instance, have characterized round-trip time (RTT) inflation. These works need to be continued, and describing these important characteristics in a way similar to what we have done here for static unweighted undirected routes would certainly make sense.

The other main contribution of this paper is to propose simple models which make it possible (and easy) to generate large amounts of artificial routes similar to real ones (in the sense of the statistical properties we have observed). These models may be used in particular for simulation purpose.

We have shown both that these models capture non-trivial features (the obtained routes are not shortest paths) and that they fit well real-world data. This last point however depends on the underlying graph and its properties. If we consider the original graph, then the results are in very good accordance with the real-world data. If we take other internet maps, then the results remain very good. If we turn to graph models, however, the results are very poor. This indicates that the performances of our models rely on some properties which are not captured by these graph models, thus confirming that there is still much to do for the accurate modeling of internet topology.

Just like we have done here, it would also probably make sense to model the fact that routes are *directed*, *dynamic* and *weighted*. The node degree model is static and undirected by nature: it always produce the same route from a given node to another (except if there is a choice between several shortest paths in the middle). The random deviation, on the countrary, already contains an amount of dynamics and even of direction. The obtained route may vary from one time to another. However, most remains to be done to model these characteristics.

We have also shown that the properties of the graph used to model the internet has a crucial impact on the performance of our models. We explained most of the influence of the graphs on the models, which leads us to conclude that any model would perform poorly because of the fact that graph models are still not accurate enough to actually contain routes with the properties we captured. This is an important point which argues for the three following points:

- first, it would make sense to conduct experiments on more intricate models in order to confirm our conclusion that current models are not accurate enough;
- second, the most relevant *models* of the internet topology seem to be the real-world maps obtained by actual measurement. Simulation should therefore be ran on such graphs, but also on models which have the advantage of being well understood, which in turn makes it possible to interpret the observed phenomena;
- and finally there is still much to do for the accurate modeling of the internet topology, and much to understand concerning its precise features.

Finally, this study has restricted itself to the IP graph (though we have made a comparison with the router graph in the case of the *nec* graph). As we mention in the Introduction, measurements of the AS graph are also available, and it is well known that much of path inflation can be explained by decisions taken at the inter autonomous system level. Making the same kinds of analysis and modeling as we have done, but at the AS level, would certainly be interesting. Moreover, relating the results at one level to the other would improve significantly our understanding of internet routes, and of the internet in general.

Acknowledgments. We thank kc claffy and the staff at CAIDA for making the *skitter* data available to us. We thank Ramesh Govindan and Damien Magoni for providing useful data. No such study would be possible without the real-world data these colleagues collect and make available. We also thank Mark Crovella and Miriam Sofronia for their helpful comments. This work is supported in part by the RNRT's *Metropolis* and the ACI *Sécurité et Informatique*'s *MetroSec* projects.

REFERENCES

[1] V. Jacobsen, "traceroute," 1989, ftp://ftp.ee.lbl.gov/traceroute.tar.gz. Also see "NANOG traceroute," ftp://ftp.login.com/pub/software/traceroute/.
[2] V. Paxson, "End-to-end routing behavior in the Internet," in *Proc. ACM SIGCOMM*, 1996.
[3] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. on Networking*, vol. 5, no. 5, pp. 601–615, Oct. 1997, See also *Proc. ACM SIGCOMM* 1996.
[4] N. Spring, R. Mahajan, and T. Anderson, "Quantifying the causes of path inflation," in *Proc. ACM SIGCOMM*, 2003.
[5] H. Tangmunarunkit, R. Govindan, and S. Shenker, "Internet path inflation due to policy routing," in *Proc. SPIE ITCom*, 2001.
[6] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin, "The impact of routing policy on internet paths," in *Proc. IEEE Infocom*, 2001.
[7] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network topology generators: Degree-based vs. structural," in *Proc. ACM SIGCOMM*, 2002.
[8] P. Barford, A. Bestavros, J. Byers, and M. Crovella, "On the marginal utility of network topology measurements," in *Proc. Internet Measurement Workshop (IMW)*, 2001.
[9] L. Li, D. Alderson, W. Willinger, and J. Doyle, "A first-principles approach to understanding the internet's router-level topology," in *Proc. ACM SIGCOMM*, 2004.
[10] K. Fall and K. Varadhan, "The ns manual," 12 2003.
[11] B. Huffaker, D. Plummer, D. Moore, and k claffy, "Topology discovery by active probing," in *Proc. Symposium on Applications and the Internet (SAINT)*, Jan. 2002.
[12] CAIDA, "skitter," http://www.caida.org/tools/measurement/skitter/.
[13] A. Lakhina, J. Byers, M. Crovella, and P. Xie, "Sampling biases in IP topology measurements," in *Proc. IEEE Infocom*, 2003.
[14] G. F. Riley, M. H. Ammar, and R. Fujimoto, "Stateless routing in network simulations," in *MASCOTS*, 2000, pp. 524–531.
[15] J.-L. Guillaume and M. Latapy, "Relevance of massively distributed explorations of the Internet: Simulation results," in *Proc. IEEE Infocom*, 2005.
[16] Aaron Clauset and Cristopher Moore, "Traceroute sampling makes random graphs appear to have power law degree distributions," in *Proceedings of the International Symposium on Theory of Computing (STOC)*, 2005.
[17] D. Meyer, "University of Oregon Route Views Project," http://www.antc.uoregon.edu/route-views/.
[18] J. J. Pansiot and D. Grad, "On routes and multicast trees in the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 1, pp. 41–50, Jan. 1998.
[19] R. Govindan and H. Tangmunarunkit, "Heuristics for internet map discovery," in *Proc. IEEE Infocom*, 2000.
[20] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," in *Proc. ACM SIGCOMM*, 2002.
[21] N. Spring, M. Dontcheva, M. Rodrig, and D. Wetherall., "How to resolve ip aliases," Tech. Report 04-05-04, Washington Univ. Computer Sci., 5 2004.
[22] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker, "In search of path diversity in ISP networks," in *Proc. Internet Measurement Conference (IMC)*, 2003.
[23] K. Keys, "iffinder," a tool for mapping interfaces to routers, http://www.caida.org/tools/measurement/iffinder/(restricted access).
[24] A. Broido and kc Claffy, "Internet topology: Connectivity of IP graphs," in *Proc. SPIE International Symposium on Convergence of IT and Communication*, 2001.
[25] IANA, "Special-use IPv4 addresses," RFC 3330, Internet Engineering Task Force, Sept. 2002.
[26] S. Jin and A. Bestavros, "Small-world internet topologies," Tech. Report BUCS-TR-2002-004, Boston U. Computer Sci., 2002.
[27] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *Proc. ACM SIGCOMM*, 1999.
[28] A. Vazquez, R. Pastor-Satorras, and A. Vespignani, "Internet topology at the router and autonomous system level," [cond-mat/0206084].
[29] A. Vazquez, R. Pastor-Satorras, and A. Vespignani, "Large-scale topological and dynamical properties of the Internet," *Phys. Rev. E 65, 066130*, 2002.
[30] L.A. Adamic, "The small world web," in *Proc. 3rd European Conf. Research and Advanced Technology for Digital Libraries, ECDL*, S. Abiteboul and A.-M. Vercoustre, Eds. 1999, number 1696, Springer-Verlag.
[31] A.Z. Broder, S.R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J.-L. Wiener, "Graph structure in the web," *Computer Networks*, vol. 33, no. 1–6, pp. 309–320, 2000.
[32] R. Albert and A.-L. Barabasi, "Statistical mechanics of complex networks," *Reviews of Modern Physics 74, 47*, 2002.
[33] M.E.J. Newman, "The structure and function of complex networks," *SIAM Review 45, 2*, pp. 167–256, 2003.
[34] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vasquez, and A. Vespignani, "A statistical approach to the traceroute-like exploration of networks: theory and simulations," in *Proc. 1st Int'l. Conf. on Combinatorial and Algorithmic Aspects of Networks (CAAN)*, 2004, full version to appear in Theoretical Computer Science (TCS).
[35] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance in complex networks," *Nature*, vol. 406, pp. 378–382, 2000.
[36] B. J. Kim, C. N. Yoon, S. K. Han, and H. Jeong, "Path finding strategies in scale-free networks," *Phys. Rev. E 65, 027103*, 2002.
[37] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, "Breakdown of the internet under intentional attack," *Phys. Rev. Lett.*, vol. 86, pp. 3682–3685, 2001.
[38] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, "Resilience of the internet to random breakdown," *Phys. Rev. Lett.*, vol. 85, pp. 4626, 2000.
[39] D.S. Callaway, M.E.J. Newman, S.H. Strogatz, and D.J. Watts, "Network robustness and fragility: Percolation on random graphs," *Phys. Rev. Lett.*, vol. 85, pp. 5468–5471, 2000.
[40] J.-L. Guillaume, M. Latapy, and C. Magnien, "Comparison of failures and attacks on random and scale-free networks.," in *Proceedings of the 8th International Conference on Principles of Distributed Systems (OPODIS)*, 2004, http://www.liafa.jussieu.fr/˜latapy/Publis/.
[41] T. Walsh, "Search in a small world," in *Proc. IJCAI*, 1999.

[42] E. Bampis, , M. Latapy, and F. Pascual, "Computing short paths in scale-free networks," 2004, preprint.

[43] Ramesh Govindan and Hongsuda Tangmunarunkit, "Mercator," http://www.isi.edu/scan/mercator/mercator.html.

[44] Damien Magoni, "nec (*ne*twork *c*artographer)," https://dpt-info.u-strasbg.fr/~magoni/nec/.

[45] Damien Magoni and Mickaël Hoerdt, "Internet core topology mapping and analysis," *Computer Communications*, vol. 28, pp. 494–506, 2005.

[46] P. Erdos and A. Renyi, "On random graphs i," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[47] B. Bollobas, *Random Graphs*, Academic Press, 1985.

[48] S.N. Dorogovtsev and J.F.F. Mendes, "Evolution of networks," *Adv. Phys. 51, 1079-1187*, 2002.

[49] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[50] Linyuan Lu, "The diameter of random massive graphs," in *12th Ann. Symp. on Discrete Algorithms (SODA)*, ACM-SIAM, Ed., 2001, pp. 912–921.

[51] M.E.J. Newman, D.J. Watts, and S.H. Strogatz, "Random graphs with arbitrary degree distributions and their applications," *Phys. Rev. E*, 2001.

[52] R. Cohen and S. Havlin, "Scale free networks are ultrasmall," *Phys. Rev. Lett.*, vol. 90, 2003.

[53] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE," Boston University Representative Internet Topology Generator, http://www.cs.bu.edu/brite/.

[54] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proc. MASCOTS*, 2001.

[55] L. D. Amini, A. Shaikh, and H. G. Schulzrinne, "Issues with inferring Internet topological attributes," in *Proc. SPIE ITCom*, 2002.

[56] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of Internet path selection," in *Proc. ACM SIGCOMM*, 1999.