Internet Core Topology Mapping and Analysis

Damien Magoni, Mickaël Hoerdt

Internet Core Topology Mapping and Analysis

Damien Magoni
Université Louis Pasteur – LSIIT
67400 Illkirch, France
magoni@dpt-info.u-strasbg.fr

Mickaël Hoerdt
Université Louis Pasteur – LSIIT
67400 Illkirch, France
hoerdt@lsiit.u-strasbg.fr

***Abstract*—The study of the Internet topology allows us to discover properties that can be exploited in order to improve the performance of network protocols and applications. Thus, Internet mapping is very useful to simulation software focusing on network protocols. Unfortunately Internet mapping has two major drawbacks. First it takes a lot of time to be carried out and secondly it is often incomplete. To solve these two problems, we have developed a fast mapping software whose aim is to map the heart of the Internet as fast as possible with the highest possible accuracy. In this paper, we describe our software and evaluate its performance compared to an existing and freely available mapping software. We assess the completeness of the router level maps by creating overlays with an autonomous system topology. We also study the absolute efficiency of our software by making an extensive campaign. We show that limitations on the amount of information do appear when trying to carry out an exhaustive mapping. Finally we study the path inflation phenomenon caused by the routing protocols.**

I. Introduction

In this paper, we describe our Internet mapping software and we analyze the maps that it produces. The section II describes earlier work concerning router level Internet mapping as well as the software used for comparison with our software. Section III describes in detail how our mapping software works. Section IV presents the collected maps from each software and the autonomous system map used to create the overlays. It also contains the results of the analysis of these maps as well as an evaluation of their accuracy which can assess the efficiency of each mapping software. In section V we describe our efforts to create an exhaustive Internet map as well as the resulting data and the problems that we encountered. Finally in the section VI we study the path inflation phenomenon at the router level and we show that it is a common and non-negligible aspect of Internet routing.

II. Previous work

The Internet mapping research effort began a few years ago. In 1995, Pansiot *et al.* created a router level map from a single host toward about 5000 destinations [1]. They used source routing to discover traversal links. In 1999, Burch *et al.* used BGP tables to find the destination prefixes [2]. They also used traceroutes from a single host but they used tunnels toward other hosts to obtain a similar effect to source routing. Also in 1999, Govindan *et al.* wrote a mapping software called Mercator. It uses a random address selection for the traceroute destinations [3]. We use this software for comparison with *nec* because it is the only free available mapping software for the research community. It runs from a single host, sending UDP packet with a increasing TTL (similar to traceroute). It also uses source routing to discover traversal links. In 2002, Spring *et al.* wrote the Rocketfuel software [4]. This software has a similar approach than *nec* but it only enables to map a single ISP at a time. In addition, this software is not freely distributed in contrary to the alias resolution software used in it. This alias resolution software is named Ally and it can be freely downloaded. We used Ally with *nec*. Router level Internet mapping has received much attention very recently. Aside of our project, we can also cite Scriptroute [5] which is the successor of Rocketfuel (and the code is now available) and the Opte project [6] for which no code is currently available. However we could not compare *nec* to these last two because they were done nearly at the same time.

Concerning our overlay creation method, it is different and new compared to the methods used in [7] and in [8] because we compute a correspondence between discovered routers from Mercator or *nec* and the ASes found in the *route-views* BGP tables using the IP interface and BGP prefixes. As a consequence, we do not need to generate the AS graph with a grouping algorithm as the one presented in [7] and we avoid the possible

introduction of errors in the cases where many disjoints router groups belong to the same AS and thus have to be reassigned. Moreover, our creation method, unless other approaches, has the advantage to give an completeness rate evaluation (For the AS and BGP connections) of the AS level map over the router level map. This allows us to estimate the completeness quality of our router level maps. Our overlay creation method has been described in [9].

III. NETWORK CARTOGRAPHER

Our cartographer software is named *network cartographer* or *nec* in short. *nec* is launched from one given point in the Internet (typically on a host computer) and queries in parallel a set of traceroute web servers to send requests towards a set of targets (i.e. destination addresses) locally stored. This technique is illustrated in figure 1. As the figure shows, the computer running *nec* sends requests to various traceroute web servers in parallel (sets a and b of messages). However the process of querying a traceroute web server, the latter doing the trace and sending the result back to *nec* is sequential (ordered from 1 to 3 in the figure). The number of items in the sets of sources and destinations can be potentially high, that's why we implement a local and distant resource saving mechanism. At the startup of *nec*, we can fix a limit to the local number of simultaneous requests (typically between 10 and 100). The number of distant simultaneous requests coming from the same host where *nec* is running is limited to one in order to avoid overloading a server more than a normal manual web request and being kicked out for overuse.

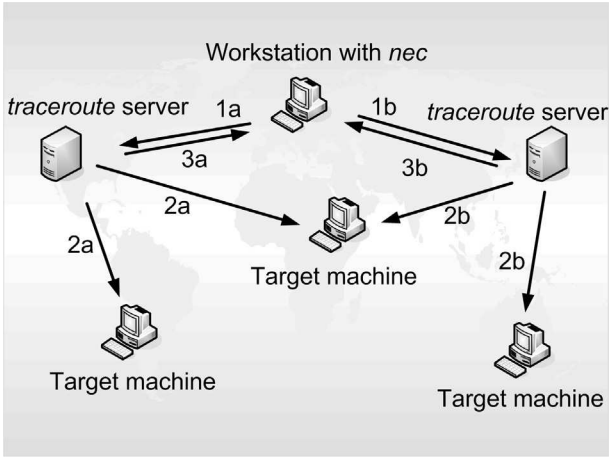

Fig. 1. *nec* functioning principle.

In the following text, we will call x the local limit. On initialization, we open x HTTP connections toward x distinct traceroute web servers for which we submit a request towards the same destination D. We wait for the results, knowing that we can't predict the arrival order of the answers and that some answers can be incomplete given the inherent properties of the Internet (i.e. ICMP packets filtered on some routers).

Upon the reception of the answers, we store them for later offline computation. If a request is complete, we close the connection to open another one and send another request toward D, using the next unsolicited traceroute web server. If the latter has already sent a request for D, it will send a request to the next destination address after D. If the answer is not complete, we store the already sent data and wait again, trusting TCP to achieve the completion of the information recovery. If a traceroute request is filtered, we close the connection (i.e. if we receive a succession of * symbols). Before each new connection initialization, we make sure that our resource saving criteria are still true: that there is not another request running on the traceroute server that we would like to query and that the number of locally running request is less or equal to x. *nec* keeps some state information in order to be able to restart an interrupted mapping without having to carry out all the already completed requests.

The IP destination addresses (i.e. traceroute targets) selection is a crucial aspect or our mapping system. In our first campaign, we have selected targets based on their owning autonomous system and known from previous mapping campaigns (see section IV). The destination addresses all belong to distinct autonomous systems in order to obtain an optimal topological distribution of the targets and as a consequence the more accurate map possible using the least possible amount of requests. In our second campaign, we have built the target addresses from a list of the network prefixes contained in a BGP dump from *route-views* (see section V in order to achieve a more complete list of targets.

nec is written in C and C++. Its source code is freely available for non-commercial use. It can be downloaded at the address [10]. But it is still not finished and we would like to implement additional functionalities. Thus it is currently available to the network research community only for functional evaluation and testing purposes.

A. Interface disambiguation

The collected traces contain IP interface addresses of Internet routers. Thus alias resolution (i.e. the identification of interfaces belonging to the same router) is an important mapping step which is called interface disambiguation. In *nec*, alias resolution is done only when the collect of the traces is completed. We first build a map of the interfaces (i.e. not the routers) from the traces by using our *nem* software [11]. Then we create a list of pair of interfaces that may be potential aliases. Indeed, the basic method of checking all the pairs for aliasing would be impossible as it is proportional to the square of the number of interfaces (i.e. $n \times (n-1)/2$). For a 50k interfaces' map this yields 1.25G checks! Thus we first use techniques to reduce the number of pairs to check for aliasing (i.e. test if they belong to the same router or not). Two IP addresses (i.e. interfaces) are suspected to belong to the same router if they comply with *all* the following rules:

- they have the same DNS suffix (i.e. the suffix is the part after the first dot of the FQDN when the latter is defined) and,
- they have been discovered by traceroute requests launched by different traceroute web servers and,
- they belong to the same AS and,
- they have a common neighbor in the interface map (but they are not neighbors themselves).

The last condition is illustrated in figure 2. If two traces coming from 2 different sources and targeted towards the same destination do merge at a given point (IP address 2 in the figure), then there is a probability that the 2 previous interfaces are aliases (IP addresses 1 and 4 in the figure). If they do not comply with the last condition, it is not necessary to test them as they can't be aliases. Indeed, suppose that there is a router with IP address 0 between S1 and R1 in figure 2. Testing IP address 0 with IP address 4 is useless because if they were aliases, the next hop (obeying the shortest path property) for IP address 0 would then be IP address 2 (the same as the next hop for IP address 4) and not IP address 1 as it would not make a shortest path to D. Thus IP address 0 and IP address 4 can't be aliases and in general, all pairs that do not comply with the fourth condition can't be. Of course this is true only if the routing system do provide to the traceroute packets the shortest path to their destinations. This may not be true in cases of transient routing failures which may induce loops or re-routing. However we do not currently have enough information to take these cases into account. These techniques to reduce the list of interface pairs to check brought the number of checks needed for disambiguating the 50k interfaces' map to less than 100k (instead of 1.25G).

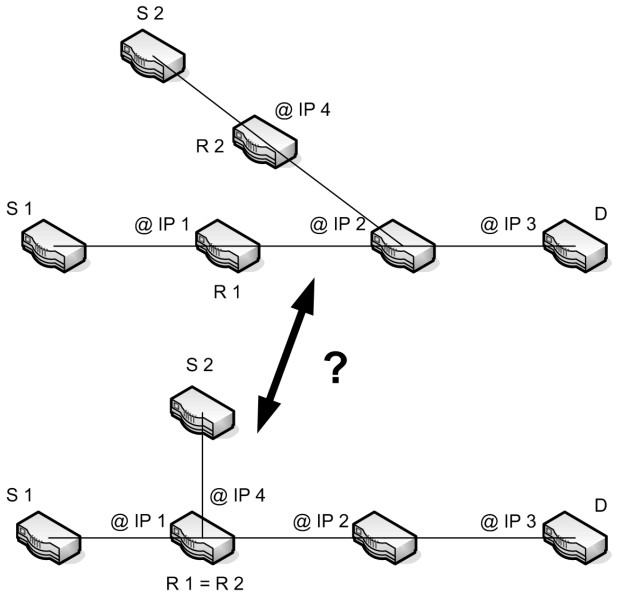

Fig. 2. Reducing disambiguation cases by using the map of interfaces.

In Mercator, disambiguation is achieved at the same time than the IP interfaces discovery by using a method initially proposed by Pansiot *et al.* in [1]. The software sends an UDP packet toward a closed port to each suspected address belonging to the same router interfaces. If the source addresses of the ICMP packets are identical, then there is a high probability that the two interfaces belong to the same IP node. This method is illustrated in figure 3.

In *nec*, the alias resolution for the suspected interface pairs is done by using a free software named Ally and created by Neil Spring. Ally is fully described in [4]. It sends an UDP packet toward a closed port to two suspected addresses belonging to the same router interfaces. If the values of the IP identifier field of the ICMP packets are almost the same, it then sends a third packet to the first interface. It the third value is close to the two previous ones, and that the three values respect an increasing order then there is a good chance that the two interfaces belong to the same router (i.e. the IP identifier is computed with the same counter and as a consequence from the same router). This technique is illustrated in figure 4.

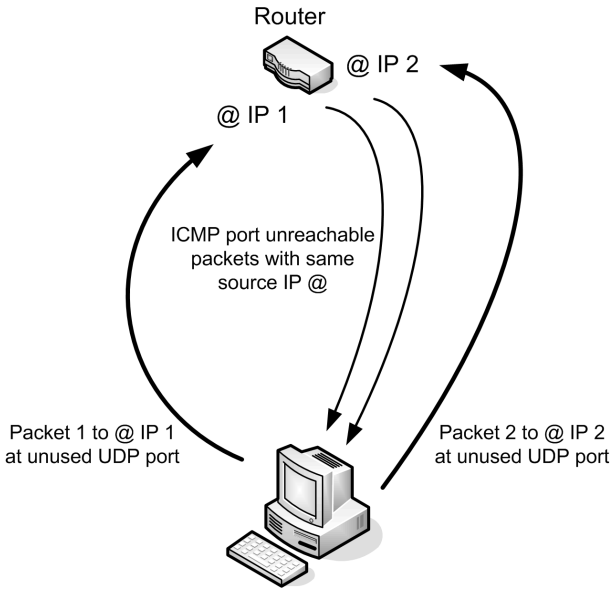

Fig. 3. Disambiguation with the source address of the reply.

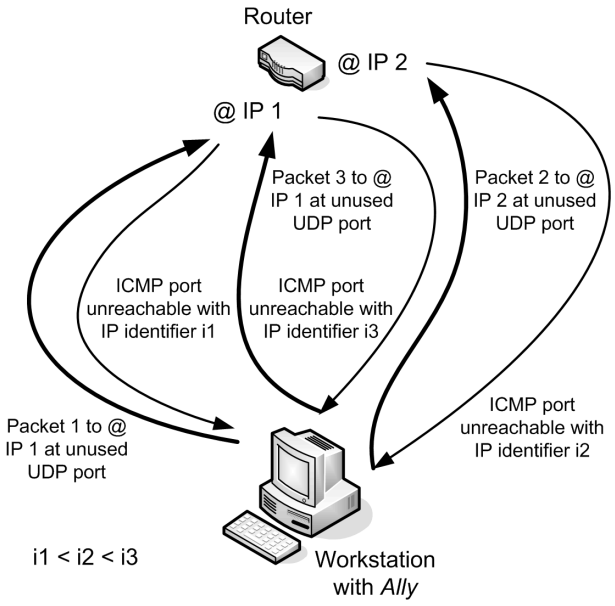

Fig. 4. Disambiguation with the IP identifier value of the reply.

When the list of suspected interface pairs has been produced by *nem* and resolved by Ally, we finally build the router map by merging the interfaces that belong to the same router. We keep the lowest IP address of all the IP aliases of a router as the canonical IP address for this router. We have implemented this procedure in our *nem* software.

B. Internet mapping

We want to fast map the Internet core. Previous work on Internet mapping [1], [2], [3] campaign have typical duration of several weeks or even many months. Here we make a campaign where the measurement duration is 10 days. This may be seen as a long time too, but one should not forget that a fastest campaign could damage the service provided by the traceroute servers (when using *nec*) and/or could be seen as a distributed denial of service attack toward the destinations addresses (Mercator and *nec*) or as a network spying tentative (Mercator). In addition, we have only requested IP addresses in the public range, not considering private addressing behind NAT boxes. We are limited by the ICMP processing of the different routers. Some of them do not respond with an ICMP message and others filter all ICMP packets. This last behavior is very common within large Internet providers but not within the large network operators. This explains why we are mostly mapping the Internet core rather than the Internet leaves (i.e. traceroute requests do not always reach their destinations). We ran Mercator and *nec* during 10 days and we compared the collected router maps. To create the overlays required for completeness analysis, we need router maps and an autonomous system map. We collected the two router level maps from the same host, located in our laboratory (LSIIT, at Illkirch, France). We used the Mercator software written by Govindan *et al.* and documented in [3] for the first map and our software for the other. The next section presents the collected maps as well as their analysis.

IV. Accuracy comparison of *nec vs* Mercator

In this section, we study the map properties collected with Mercator and *nec* in order to evaluate their quality. We also gathered an AS level map, computed from the routing table of the BGP router observer *route-views* dated from April 2003. The table I contains general information about these maps. We notice that Mercator detected 4.6% of aliases and Ally 6.1%. Without any hypothesis about the efficiency of these alias resolution methods, we note that the alias detection rate is about 5%, which is one third of the value observed in [12].

A. General properties

The table II shows the main properties of the maps. We recall that these properties have been collected in 10 days. First, *nec* discovered 15% router less than Mercator. An increase in the request frequency and/or

TABLE I. INTERNET MAPS USED FOR ANALYSIS

Source	Interface or prefix number	router or AS number	link count	Date
Mercator	58119	55425	65237	7-17/4/2003
nec	50123	47055	119909	19-29/4/2003
route-views	130815	15129	31378	1/4/2003

TABLE II. COLLECTED MAP PROPERTIES

Property	Mercator map	*nec* map	Deviation
Routers	55425	47055	-15.1%
BGP routers	12.0%	59.8%	+398.3%
IP links	65237	119909	+83.8%
Mean degree	2.35	5.09	+116.6%
Mesh part	24.8%	85.7%	+245.5%
AS completeness	11.7%	35.0 %	+199.1%
BGP connection completeness	6.1%	27.8%	+355.7%

the number of traceroute servers would permit us to make up with this difference. After the overlay construction, we found that 12% of the routers present in the Mercator IP topology can be defined as BGP routers according to our resolution method explained in [9]. With *nec* this rate is almost 60%. We do not possess enough information which could permit us to assess this number with its real value in the Internet but this shows that our target distribution method (i.e. the traceroute destination address selection) based on taking one address in each AS discovers a lot of inter-domain IP links. Second, and this is the most important point, *nec* discovered almost 84% more links than Mercator! This result is very encouraging. It clearly shows the advantage of using multiple distributed vantage points over the source-routing techniques which are not fast enough to discover a similar number of traversal links. As a result, the mean degree of the *nec* map is twice more as in Mercator (And this means having a higher density of redundant links). The meshed part (i.e. the graph without its trees) of the maps is important. The Internet is far more meshed than other previous work showed (85% of the *nec* map routers are in the mesh). This shows that the meshed value of Internet graphs must not be considered as a good indicator (i.e. as a stable graph property). Always with the AS overlay map, we can compute the completeness of the router maps. The AS completeness give the percentage of AS containing at least 1 router (ideally this should be 100%). These indicators are relatively inaccurate but sufficient enough to show the limits of our router maps. We can see in these values the high bias of Mercator which only fill 12% of AS and 6% of connections. *nec* reach very superior values with a respective filling rate of 35% and 28%. Although these values are still not sufficient, the progress offered by *nec* compared to the other mapping methods is real.

B. Power laws and distances

Topological indicators are interesting tools to quantify the similitude of a graph with the Internet topology.

TABLE III. MAIN INDICATOR VALUES

Indicator	Mercator map	*nec* map
Degree CC	0.98	0.96
Degree exponent	-2.80	-2.36
Rank CC	0.96	0.95
Rank exponent	-0.65	-0.73
Tree size CC	0.98	0.97
Tree size exponent	-2.12	-3.92
Tree rank CC	0.98	0.96
Tree rank exponent	-0.83	-0.35
Number of shortest paths CC	0.67	0.95
Number of shortest paths exponent	-1.38	-1.51
Pair rank CC	0.99	0.99
Pair rank exponent	-0.58	-0.58
Mean distance	16.3	9.0
Eccentricity	30.9	18.9
Radius	23	15
Diameter	44	28

These indicators are defined in our previous work [13]. The table III contains the main indicator values for our two router maps. In general, they are very similar (more important variations for exponents are usual), except for those concerning the number of shortest paths and those concerning the distances.

The correlation coefficient (CC) of the number of shortest paths (which illustrates the presence of the power law number 5 described in [14]) is not verified with the Mercator map (i.e. it is less than the threshold value of 0.95). This property is linked to the redundant link quantity and we can illustrate by this default the inaccuracy of Mercator. In opposition, the *nec* map verifies this crucial property thanks to the important number of discovered links. The figure 5 shows the percentage distribution of the pair of routers having between 1 to 100 shortest paths. The dots must be aligned. We notice the strong dissemination of Mercator. This is not the case for *nec*. Concerning the distances (measured in number of hops, the Mercator map has very different property values compared to the ones present in big router maps. In the opposite, *nec* possesses almost the same property values than those measured in the maps bigger than 200k routers [15]. Missing links in the Mercator map stretch the distances between the routers. This increases

the distance values by up to 50% as compared to the usual distance indicators. The figure 6 shows the router percentage distribution having a given mean distance with the others (rounded to the tenth). The *nec* dots make a clear and well centered around 9 hops normal distribution while the Mercator dots are disseminated around the value of 17 hops.

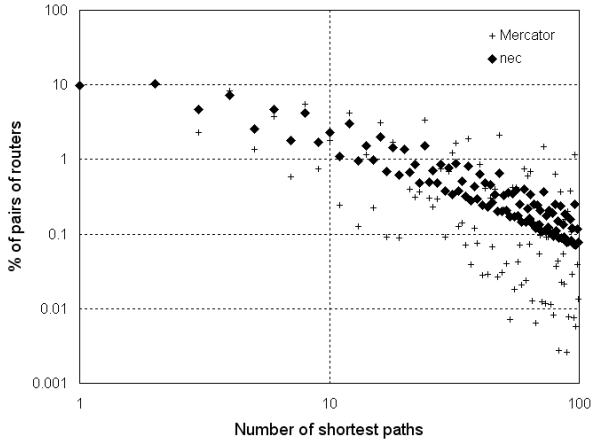

Fig. 5. Number of shortest paths distribution.

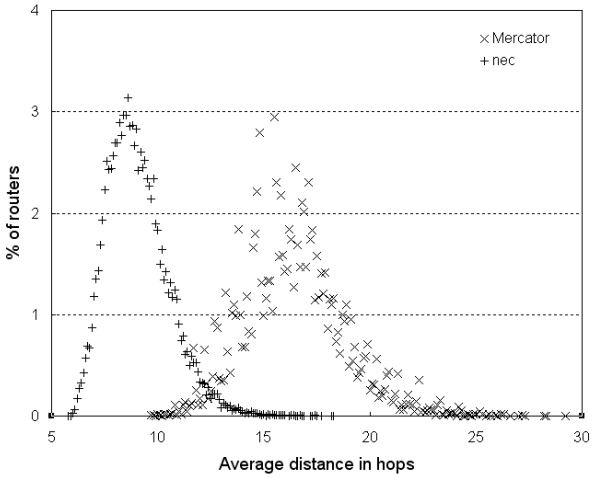

Fig. 6. Mean distance distribution.

C. Overlay quality analysis

To verify the accuracy of our overlays, we study the distribution of the AS sizes. The size of an AS is equal to the number of routers inside this AS. We showed before that the completeness rate is equal to 12% for Mercator and 35% for *nec*. This clearly shows that our router level map is far from having a total completeness. We look in our collected maps for the presence of heavy tailed distributions found in the Internet overlays created by Tangmunarunkit *et al.* in [16]. An efficient method used in [16] in order to achieve this consists in plotting the cumulative complementary distribution function *CCDF* of the AS size on a log–log scale. Figures 7 and 8 show the CCDF of the AS size distribution from Mercator and *nec* and they clearly show the characteristics of a heavy tailed distribution. We can notice that the Mercator CCDF is slightly curved contrary to the *nec* CCDF which is perfectly straight and shows a perfect correlation. However the heavy tailed distribution from *nec* is broken after a size of 1000. This is probably a consequence of the missing data due to the short measurement period. This phenomena appears for Mercator too but it's not clearly visible because of the curved line.

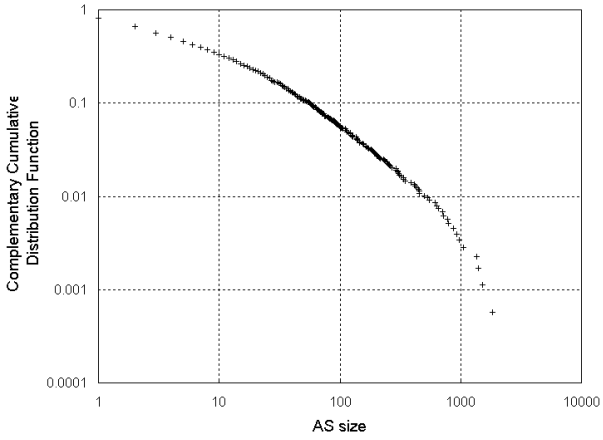

Fig. 7. Cum. comp. distr. of the AS size seen by Mercator.

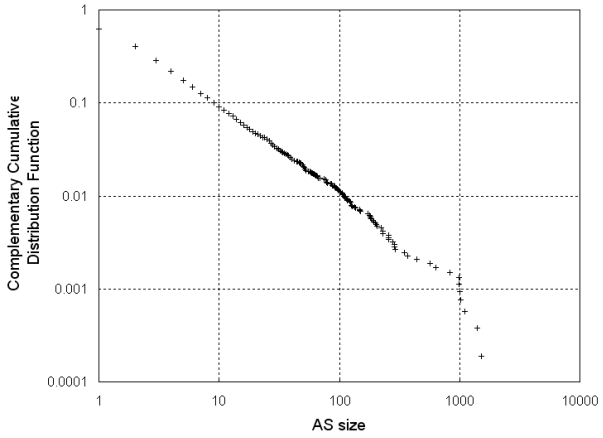

Fig. 8. Cum. comp. distr. of the AS size seen by nec.

We trace on figures 9 and 10 the correlation between the size and the degree of the 100 most important AS.

The correlation for these AS is strong and very similar to the one found in [16] especially in the *nec* map where the linearity is better than in the Mercator map.

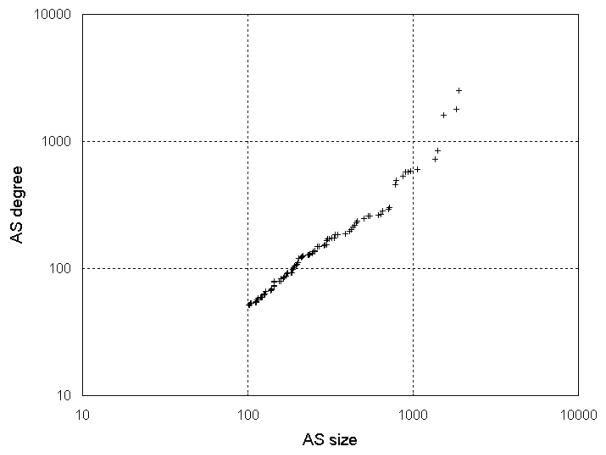

Fig. 9. Correlation between the AS sizes and their degrees seen by Mercator.

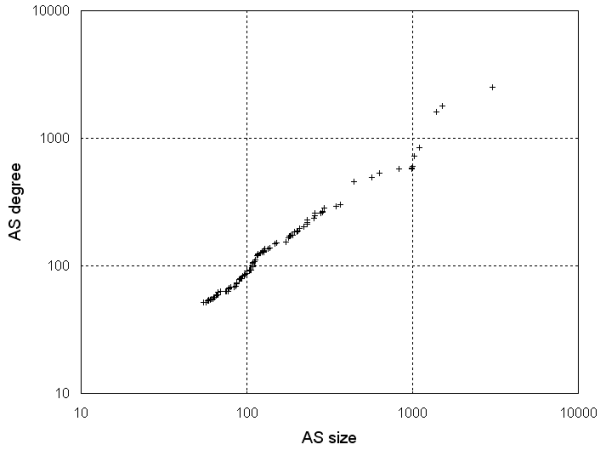

Fig. 10. Correlation between the AS sizes and their degrees seen by nec.

Figures 11 and 12 show on a log–log scale the CCDF distribution of the number of BGP routers per AS. These are very similar to the AS size and can be commented in the same way. The higher number of BGP routers in the *nec* map issue an almost perfect linearity, except for high abscissas. It's clear with these measures that these distributions are strongly correlated with the AS size.

Thanks to our overlay creation method, we can study the BGP connection sizes. The BGP connection size is equal to the number of IP links coming from any router in an AS toward any router in a given AS different from

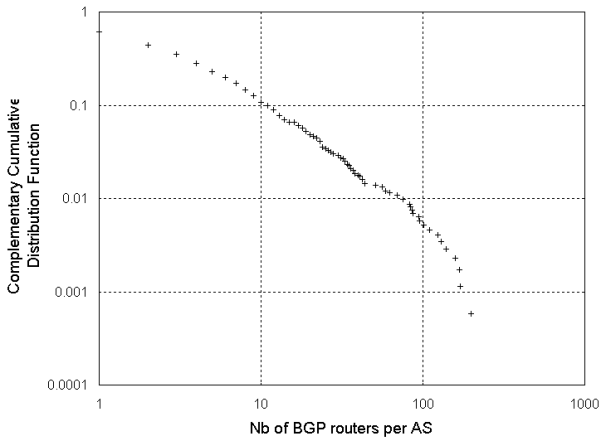

Fig. 11. Cum. comp. distr. of the BGP routers per AS seen by Mercator.

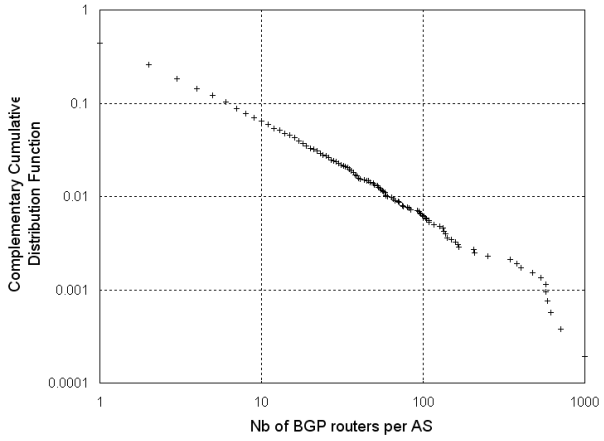

Fig. 12. Cum. comp. distr. of the BGP routers per AS seen by nec.

the first one. In a similar way to the AS, only a part of all connections contains one or more IP links. This gives us a ratio of the number of filled connections (i.e. having a size ≥ 1) *vs* the total number of connections which equals 6% for Mercator and 28% for *nec* which is lesser than the filling AS ratio. Once again this tends to show that the Mercator map misses a significant amount of IP links. Figures 13 and 14 show on a log–log scale the CCDF distribution of the BGP connection sizes and it is heavy tailed. To our knowledge, we are the first to show these kind of results (although they were intuitively predictable).

We can conclude that despite our router–AS overlays are incomplete, they contain a lot of heavy tailed distributions typically found at the macroscopic level of the Internet topology [16]. On this point, the Mercator and

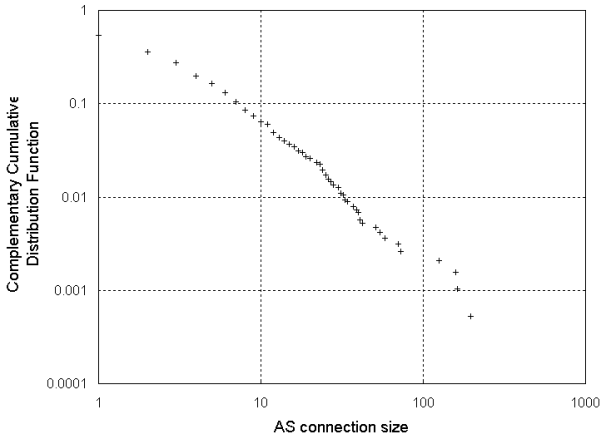

Fig. 13. Cum. comp. distr. of the connection sizes seen by Mercator.

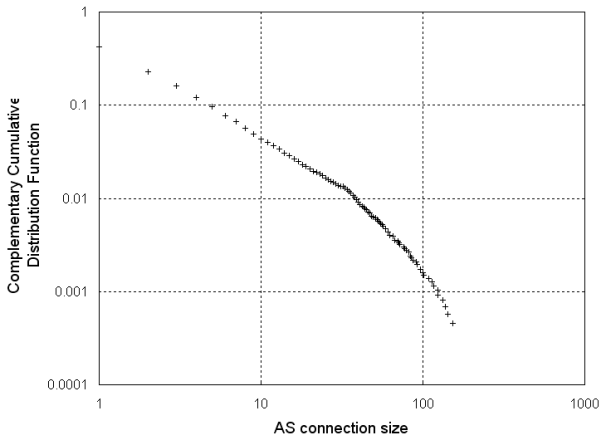

Fig. 14. Cum. comp. distr. of the connection sizes seen by nec.

nec maps show similar distributions, although those from *nec* are more accurate with regard to the linear regression correlation coefficients.

V. EXHAUSTIVE MAPPING

Our first mapping campaign was mainly aimed at investigating how fast and accurate we could map the Internet compared to other mapping efforts and especially Mercator. However with respect to the number of routers (i.e. 47k) the collected map was rather small. For our second mapping campaign we wanted to achieve a much more complete map. We wanted to keep the benefits of our previous mapping (i.e. discovery of traversal links) while collecting a much bigger map. We used *nec* the same way as for the first campaign but we changed parameters and data sets. For the target addresses we took random addresses created from every prefixes of a *route-views* BGP dump with prefixes above /16 being reduced to /16. This yielded nearly 14k destination addresses. We incorporated many new tracers in *nec* (i.e. traceroute web servers). The campaign required 40 days to be carried out which is 4 times longer than the first campaign. The data comparison of the two campaigns are shown in table IV.

TABLE IV. CAMPAIGNS COMPARISON

Parameter	First campaign	Second campaign	Ratio
End date	4/29/2003	9/15/2003	-
Duration (days)	10	40	4.00
Tracers	76	228	3.00
Targets	6013	13828	2.30
Routers	47055	75885	1.61
Links	119909	357317	2.98
Traces	456988	3152784	6.90
Traffic (traces/day)	45698.8	78819.6	1.72
Efficiency (routers/trace)	0.103	0.024	0.23
Efficiency (links/trace)	0.262	0.113	0.43

If the mapping produced by *nec* was linear and given the figures in table IV, we should have collected a nearly 7 times bigger map than the first one. That is a 325k router map with nearly 827k links! However, the values in table IV clearly show that this was not the case. The mapping efficiency was clearly lower in the second campaign and this illustrates the limits of the ”more is better” approach. A partial explanation of this limitation is that big parts of the public Internet are not mappable because of ICMP filtering (for security considerations) and we won’t be able to map these areas no matter how many tracers and targets we use. Another explanation is that our target address set may not be optimum (i.e. we may not be looking ”everywhere” in the addressing space). Thus we need to define our intended traces with great care in order to avoid redundant traces that reduce the overall efficiency of *nec* and to increase the amount of interesting traces. We should do our best to follow these guidelines in order to keep collecting the highest amount of information (i.e. new routers and links). We are currently studying this issue. Between the two campaigns, the amount of routers found per trace fell by 77% while the amount of links found per trace fell by 57%. Incremental traceroute (i.e. not starting at the source but at the middle of the path) would greatly help reducing useless traces but this would require a special traceroute version. That is we would have to deploy our own tracers and we would loose the benefit of using all the public traceroute servers as well as looking glasses. Of course we could associate both methods.

We have analyzed the main topological properties of our newly collected 75k router Internet map. First of all we have calculated the clustering coefficient of our map as defined in [17] and we have found 0.278. Furthermore the median path length of our map is equal to 7.46. Besides, we have been able to generate a random graph of 76000 nodes having a median distance of 7.65 which is within 2.5% of the one of our map. This random graph has a clustering coefficient of 0.073 which is much smaller than the one of our map. Thus our 75k Internet router level map is a *small world* graph according to [17]. This result is consistent with previous work on the Internet topology [18], [17]. In order to analyze both degree and rank distributions as one, we plot the empirical cumulative degree distribution as defined and used in [17]. This plot is illustrated in figure 15 and it shows that this distribution does not obey a power law especially when looking at high degree nodes. Bu *et al.* have shown in [17] that the degree empirical cumulative distribution (ECD) of the AS topology does comply with a power law and it is expected that the router topology should exhibit a similar property typical of heavy tailed distributions. However it is easier to have a full knowledge of the AS topology than of the router topology. This may be an indication that our map is incomplete due to a lack of data and/or a mapping bias (i.e. no full exploration of the addressing space).

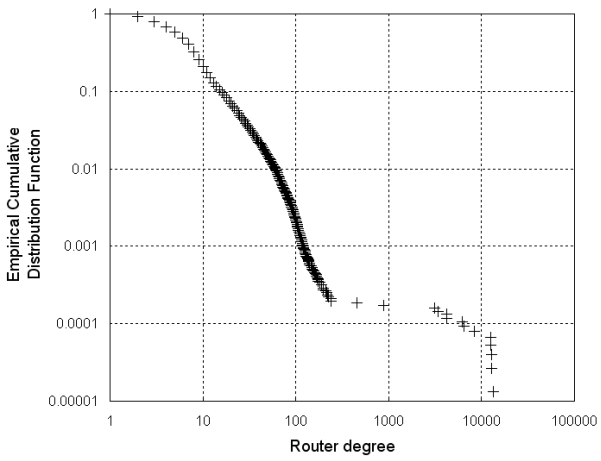

Fig. 15. Empirical cumulative degree distribution in the 9/15 map.

In the same way we plot the empirical cumulative tree size distribution in figure 16 to account for both the tree size and the tree rank power laws defined in [14]. We can see on the plot that the empirical cumulative tree size distribution does comply with a power law despite the fact that only 7.8% of the routers are in trees.

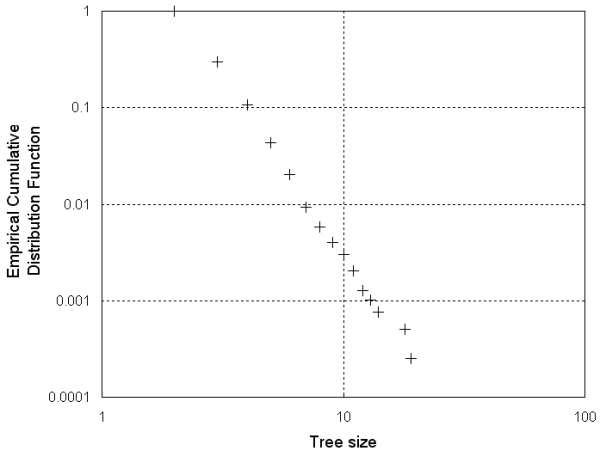

Fig. 16. Empirical cumulative tree size distribution in the 9/15 map.

Figure 17 shows the empirical cumulative distribution of the number of shortest paths between pairs of routers. We notice that this distribution does not obey a power law. The causes may be the same as for the degree ECD, that is mostly a lack of data. Indeed, the accuracy of the number of shortest paths property is closely dependent on the amount of traversal/redundant links found during the collect and this is one of the most difficult aspects to catch in a router level mapping.

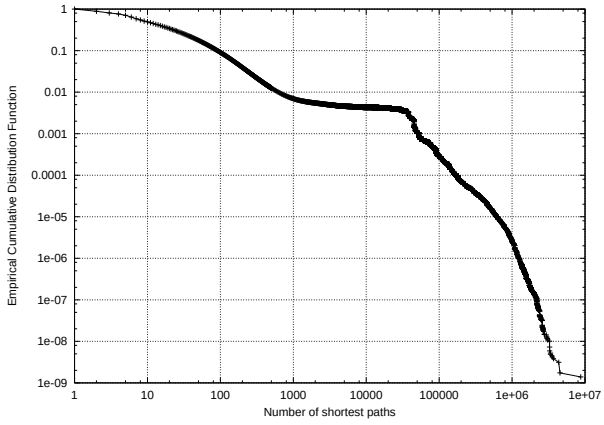

Fig. 17. Empirical cumulative distribution of the number of shortest paths in the 9/15 map.

The average path length (measured in hops) distribution is shown in figure 18. The plot exhibits a strange behavior compared to the one of the 4/29 map shown in section IV. Six points of the distribution with values around 4.5 have an unusually high frequency. All the other points form a normal distribution shape centered around 5.8 although it does not clearly appear in the

figure because the plot is shrinked by the presence of the few unusual values. As this phenomenon was not seen in the data of the first campaign, we think that it may come from our target selection method (which was different in the second campaign). In the second campaign, we have randomly built the target addresses from /16 limited network prefixes. By doing so, we may have generated a lot of non-attributed network addresses, that is IP addresses where even the network part is not valid. Thus the traceroute packets targeted to these bad addresses would reach the Internet core (i.e. the center composed of major operators) and then the routing would fail when trying to find the specified ”badly generated” network prefix. The 4.5 average distance would represent the path from the source to the average location in the Internet where the routing no longer finds the path.

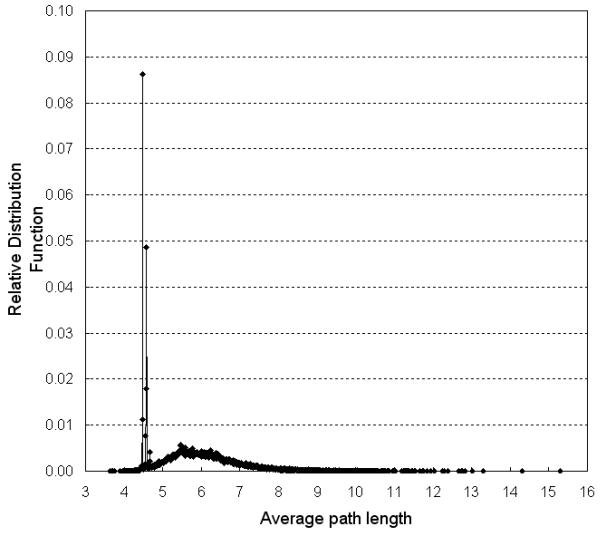

Fig. 18. Average path length distribution in the 9/15 map.

The eccentricity (measured in hops) distribution is shown in figure 19. Here the plot is similar to the ones found in the 4/29 map (not shown in section IV). We do not find here unusually high values such as in the average path length distribution. As the eccentricity is defined as a ”maximum” function and not an ”average” one, it has discrete values whose spectrum is much narrower than for the average path length. This may hide the routers that have unusual average path lengthes. However we do have a steep slope between routers having an eccentricity of 13 and those with 14. Thus routers having unusual average path lengthes most probably have an eccentricity of 14.

We have built an overlay on the 9/15 map in the same way as for the 4/29 map. Figure 20 shows the AS size

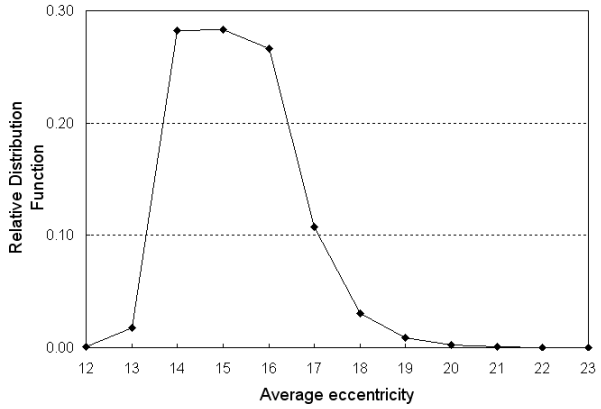

Fig. 19. Eccentricity distribution in the 9/15 map.

complementary cumulative distribution (CCD) plot. It is defined and obtained as explained in section IV-C. This plot is very similar with the one of the 4/29 *nec* map. We do not show the plots of the other overlay properties (e.g. connection size CCD) such as in section IV-C because they are all similar. We can deduce that the overlay CCD related properties are relatively independent of the router and AS map sizes. In the 9/15 map, the AS completeness (defined in section IV) equals 34.3% which is nearly equal to the one of the 4/29 map. However the BGP connection completeness equals 20.2% whereas it equals 27.8% in the 4/29 map. We can conclude that the qualitative overlay properties found in the CCD plots remain the same in the 9/15 map while the quantitative properties such as the overlay completeness do get a bit worse because of the increasing impact of the mapping limitations.

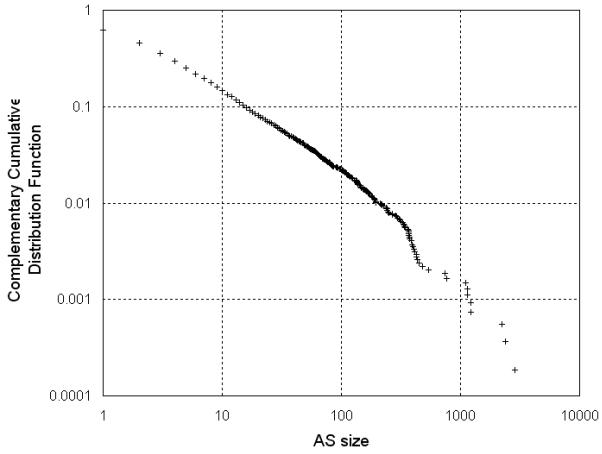

Fig. 20. Complementary cumulative AS size distribution in the 9/15 map.

We have seen in this section that extending the amount of mapping as much as possible is quite difficult to carry out efficiently with just the use of the traceroute servers. It is hard to substantially increase the number of routers and links found, and some of the distribution plots (e.g. degree and average path length) do not match the expected behavior. This leads us to think that the mapping remains incomplete. We need to add other mapping techniques to *nec* in order to overcome the limitations that appear when we try to make an exhaustive mapping of the public Internet.

VI. PATH LENGTH INFLATION

With our second bigger map collected in September, we have studied the inflation of the Internet paths. Any traceroute is defining a path really taken by a packet from the traceroute server (source) to the target address (destination). After having created the map, we can calculate the all-pairs shortest paths by using the Dijkstra algorithm for every router of the map. We obtain among all the shortest paths those between the sources and destinations of our traceroute paths. We measure the length of the traceroutes and the shortest paths by using the hop count metric. The path inflation is the traceroute length of a source-destination pair divided by the shortest path length of the same pair. The division result is a real number equal or greater than 1. We call it the *path length ratio* or the *path inflation*. With the traces collected for the built of the September map, we have been able to calculate the ratio of 2196960 paths (i.e. source-destination pairs). The distribution of the path inflation of the 2M pairs are plotted in figure 21.

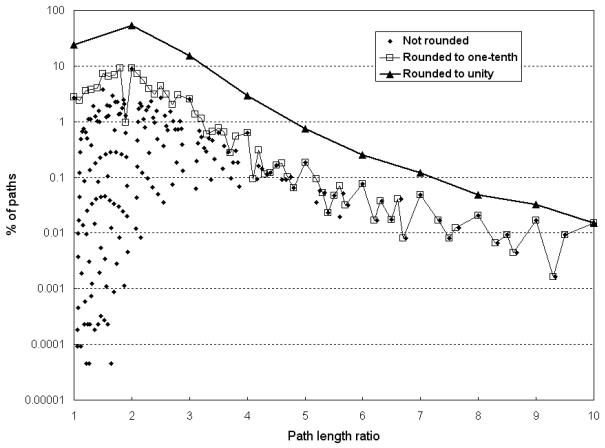

Fig. 21. Path inflation with different granularities.

The ratio ranges from 1 to 10. We chose 10 as an arbitrary upper limit. We had a few results above 10 that we consider abnormal. This can be readily explained by the fact that the traceroute paths come from the raw data while the shortest paths come from the router map obtained after desaliasing the interfaces and making the map connected. Thus some IP interfaces and links have disappeared from the map and they can be the cause of strange paths leading to high path inflation. To better visualize the distribution of the path length ratios, we have rounded the values to 1/10th and to unity. Figure 21 show all distributions on a y-log plot. Because path lengthes are natural numbers, the path length ratios have not a smooth distribution when they are not rounded to unity (i.e. sawtooth plot). This is an expectable behavior. We can see that the distributions are spread out mainly between 1 and 5 but we are far from the ideal case where most of the paths would have a ratio of 1. These results confirm that path length inflation is definitely not a marginal phenomenon but is globally impacting most of the paths with a high effect. The causes can be multifold:

- Mostly this is due to the routing policy in the Internet especially in inter-domain. Links and routes can have administratively attributed costs (especially in inter-domain) and thus they may not be optimum. Furthermore in inter-domain routing, it is by design impossible to determine the shortest *router-level* path from any node to any other one located in a different AS because only a subset of the topology information is advertised by intra-domain protocols to the inter-domain one (i.e. BGP).
- Many paths are aborted in the traces (i.e. they stop before reaching the desired destination) because of ICMP filtering. We do use the real destination (i.e. last responding router) and not the desired one to determine ratios but still if we had the full paths maybe the ratios would be lower especially if there is no path overhead at the last hops (the ones we do not have).
- It can also be caused by our selection of destinations as traceroute targets. We may choose paths that do not reflect typical source-destination pairs (i.e. that have uncommon sources and/or destinations).

Figure 22 shows the cumulative distribution of the path length ratios for an easier classification of the paths having a ratio above a given threshold. The most interesting facts of these data is that only 2.72% of the paths

have a ratio of 1 (i.e. cases where the traceroute path has the same length as the corresponding shortest path). Moreover, 43.4% of the paths have a ratio above 2 which is clearly not negligible! (i.e. a path length increased by at least 100% compared to the corresponding shortest path)

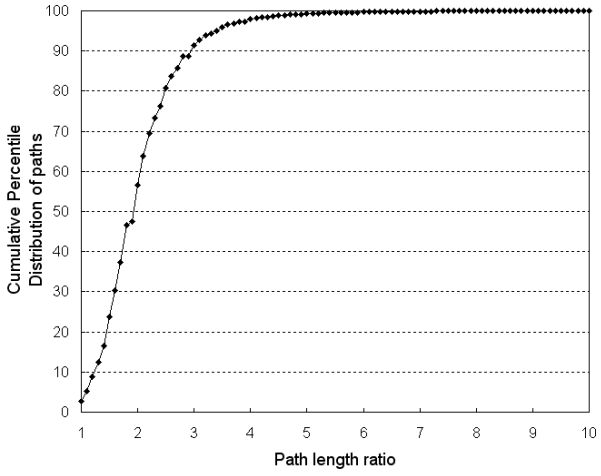

Fig. 22. Path inflation cumulative distribution.

Path inflation has been studied by Tangmunarunkit *et al.* in [7] during year 2000. They have found that more than 95% of their paths hade a path length ratio equal or less than 2 while we only found 56.6% of our paths verifying this property. However their study involved only 61k traces (while we had 2M) and they used a map generated by Mercator (i.e. thus a map missing a large number of traversal links, see section IV). Therefore we are confident that our results are more accurate. We have shown in this section that it is clear, especially on figure 22, that the vast majority of the paths have a non negligible path inflation. However the metric that we used for path inflation calculation was the hop count. A heavily increased hop count caused by path inflation between two routers may not automatically translate in a heavy delay penalty. It would be interesting to study the path inflation with regard to the delay metric in order to see if the phenomenon still holds. Very recently, Spring *et al.* carried out a closely related study in [19] and they found that the mean increase in path delay for an inter-domain path (the worse case scenario) is less than 14ms. However they did not correlate the delay inflation to the hop count inflation.

VII. Conclusion

Internet mapping is a complex operation. In this paper we described our *nec* fast mapping software and showed that it is far more efficient than Mercator, the only free mapping software at the time of our study (a new free software called Scriptroute is now available). In 10 days, *nec* discovered almost the same number of routers and nearly twice as much links as Mercator. The resulting map finished in April 2003 possesses a relative amount of links even higher than a typical AS map and higher than any previously collected router level map of the public Internet core. Moreover the overlays show that its completeness is higher than the Mercator map collected in the same duration. However our second more exhaustive campaign showed that the performances of *nec* were far from linear. We showed that trying to exhaustively map the Internet requires other techniques to be able to cope with the limitations incurred by bad target selection and traceroute web servers. We are currently studying these issues. Although the maps collected by *nec* are still far from complete, they are among the most detailed Internet maps available when considering the amount of link information. *nec*is freely available mainly for code examination at [10]. Furthermore, people who would like to use the maps collected by *nec* (and containing their corresponding overlays) in order to run protocol simulations can find them on the web [10] where *nec* is available.

References

[1] J.-J. Pansiot and D. Grad, "On routes and multicast trees in the internet," *ACM Computer Communication Review*, vol. 28, no. 1, pp. 41–50, January 1998.

[2] H. Burch and B. Cheswick, "Mapping the internet," *IEEE Computer*, vol. 32, no. 4, pp. 97–98, 1999.

[3] R. Govindan and H. Tangmunarunkit, "Heuristics for internet map discovery," in *Proceedings of IEEE INFOCOM'00*, Tel Aviv, Israël, March 2000.

[4] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," in *Proceedings of ACM SIGCOMM'02*, Pittsburgh, PA, USA, August 2002.

[5] N. Spring, D. Wetherall, and T. Anderson, "Scriptroute: A public internet measurement facility," in *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS)*, March 2003.

[6] B. Lyon, *The Opte Project*, http://www.opte.org/, 2003.

[7] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin, "The impact of routing policy on internet paths," in *Proceedings of IEEE INFOCOM'01*, Anchorage, Alaska, USA, 2001.

[8] H. Chang, S. Jamin, and W. Willinger, "Inferring as-level internet topology from router-level path traces," in *Proceedings of SPIE ITCom'01*, Denver, CO, USA, August 2001.

[9] D. Magoni, “Tearing down the internet,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 949–960, August 2003.

[10] ——, *network cartographer (nec)*, Université Louis Pasteur, http://www-r2.u-strasbg.fr/˜magoni/nec/.

[11] ——, *network manipulator (nem)*, Université Louis Pasteur, http://www-r2.u-strasbg.fr/nem/.

[12] P. Barford, A. Bestavros, J. Byers, and M. Crovella, “On the marginal utility of network topology measurements,” in *Proceedings of ACM SIGCOMM Internet Measurement Workshop’01*, November 2001, pp. 5–17.

[13] D. Magoni and J.-J. Pansiot, “Evaluation of internet topology generators by power law and distance indicators,” in *Proceedings of the 10th IEEE International Conference On Networks*, Singapore, August 2002, pp. 401–406.

[14] ——, “Analysis of the autonomous system network topology,” *ACM Computer Communication Review*, vol. 31, no. 3, pp. 26–37, July 2001.

[15] ——, “Internet topology modeler based on map sampling,” in *Proceedings of the 7th IEEE Symposium on Computers and Communications*, Giardini Naxos, Sicily, Italy, July 2002, pp. 1021–1027.

[16] H. Tangmunarunkit, J. Doyle, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, “Does as size determine degree in as topology?” *ACM Computer Communication Review*, vol. 31, 2001.

[17] T. Bu and D. Towsley, “On distinguishing between internet power law topology generators,” in *Proceedings of IEEE INFOCOM’02*, New York City, NY, USA, June 2002.

[18] R. Albert and A.-L. Barabási, “Topology of evolving networks: local events and universality,” *Physical Review Letters*, no. 85, p. 5234, 2000.

[19] N. Spring, R. Mahajan, and T. Anderson, “Quantifying the causes of path inflation,” in *Proceedings of the ACM SIGCOMM’03*, August 2003.