Seventh FRAMEWORK PROGRAMME
FP7-ICT-2009-5 - ICT-2007-1.6
New Paradigms and Experimental Facilities

SPECIFIC TARGETED RESEARCH OR INNOVATION PROJECT

Deliverable D3.4

"Measurement data analysis/mining"

Project description
Project acronym: **EULER** Project full title: **Experimental UpdateLess Evolutive Routing** Grant Agreement no.: **258307**
Document Properties
Number: **FP7-ICT-2009-5_ICT-2009-1.6_(258307)_D3.4** Title: **Measurement data analysis/mining** Responsible: **UPMC** Editor(s): **Matthieu Latapy, Fabien Tarissan** Contributor(s) listed in alphabetical order: **Albert Cabellos (UPC/CAT), Davide Careglio (UPC/CAT) Jean-Charles Delvenne (UCL), Julien Hendrickx (UCL), Matthieu Latapy (UPMC), Albert Lopez (UPC/CAT), Bivas Mitra (UCL), Dimitri Papadimitriou (ALB), Fabien Tarissan (UPMC)** Dissemination level: **Public (PU)** Date of preparation: **Sep.2012** **Version: 1.1**

Contents

Chapter 1

Introduction

Our understanding of the structure of the Internet topology and its dynamics is extremely important, as it has much impact on our ability to extend and manage the network, to improve its reliability and efficiency, and to design appropriate protocols for its various applications. Indeed, such works rely on theoretical studies and simulations conducted on artificial graphs obtained from models of the Internet topology.

However, due to its decentralized nature and fast evolution for several decades, a global view of the Internet is not directly available. Instead, one relies on maps of its topology obtained through intricate and expensive measurement procedures. These measurements typically rely on the traceroute tool, which basically gives a path from a given node to another node. One then builds maps of the Internet by running traceroute from several nodes to many others, and then merging the obtained paths. Much effort is devoted to making these maps as complete and reliable as possible, and one then generally makes the assumption that they are representative of the true network. Indeed, once such a map is obtained, one generally considers that properties of the map are properties of the true Internet. As a consequence, models try capturing properties of the maps, and generate graphs similar to these maps.

This general approach may seem reasonable but it faces extremely challenging issues. The most important one certainly is that measurements give very partial views of the network, which are in addition biased by the measurement process. For instance [75, 5] show both experimentally and formally that the degree distribution (fraction of nodes with k links, for all k) observed on measurements may differ significantly from the actual one. Similar problems occur for other properties and other networks [61, 79, 112, 111]. In addition, measurement tools themselves are imperfect and prone to errors.

These issues have crucial consequences for the field, as we do not know whether observed properties should be trusted or are only properties of partial, biased and erroneous maps currently available. As a consequence, current knowledge of key properties of the actual Internet topology, even the most basic ones like its degree distribution, remains very limited. This is subject to controversy [120] with a strong impact on applications. For instance, famous results claiming that the Internet is very resilient to failures but sensitive to attacks [11, 28, 36, 37] rely on the assumption that the Internet has a power-law degree distribution with a given exponent, observed from measurements. The fact that the true network may actually have a totally different degree distribution makes the relevance of such results unclear. This leads to difficult discussions and analyses of the extent to which the observed degree distribution may be trusted [44, 61].

Of course, the degree is only one of the many properties useful to describe and model a complex object like the Internet topology. Problems encountered during the measurement and study of these other properties are in general just as challenging, or even more. In particular, if studying the structure of the Internet topology at a given moment is difficult, studying its dynamics is even harder. Indeed, trying to cope with this dynamics leads to new obstacles raised both during the measurement and during the analysis. First, there is no measurement tool enabling to directly grab information on the dynamics and studies often rely on comparing static views of the Internet acquired at different times. Second, when comparing two collections of routes it is difficult to distinguish between modifications of the routing and modifications of the underlying topology. As a consequence, current knowledge of the dynamics of the Internet topology is even more limited than knowledge of its static properties.

The goal of Task 3.2 has been to change this situation by designing new tools, methods and approaches

to accurately and reliably measure key properties of the Internet topology and its dynamics. Consistently with these measurements, we designed models able to capture observed properties, thus leading to artificial graphs more realistic than any map currently available. Indeed, these graphs will have the same key properties as the true topology, while maps are partial, biased and erroneous. In addition to the fundamental need for a true understanding of the Internet, such artificial graphs are crucial for formal studies and simulations of protocols.

Outline of the document

In Deliverable D3.2 we already reported the measurement tools and the dataset provided during the first year of the project. The purpose of the present deliverable is to rely on those dataset to extract relevant information.

As outlined above, one of the key characteristic of the Internet properties is the degree distribution of the routers. This is why we dedicated effort to develop a method providing key insight on this property. While Deliverable D3.2 we already presented the tools developed in order to implement the method, we focus here on how one could exploit the data obtained with the measurement tools and apply it on the dataset obtained during the first year of the project. More precisely, in Chapter 2, we explore all details of this approach, including its expected outcome through simulations, technical difficulties, and assessment of results. In Chapter 3, we turn to the determination of the nature of the distribution obtained with this method.

A second contribution of Task 3.2 has been to address the dynamics on the Internet. During the first year of the project, we presented the data obtained from the Radar tool that enables to study partially the routing topology dynamics in Internet and we proposed a first model able to account for the observed properties. Two main contributions emerged from this work. First, we investigated in depth the differences between different types of random graphs in terms of forwarding paths. Second, we tried to confront those observation to the one performed on BGP routing data. All those aspects are reported Chapter 4.

Besides, as a complementary approach to the detection of invariant in the routing topology dynamics documented in Chapter 4, we have also designed different methods for detecting events in times series data and applied them using the tool obtained from the Radar measurement performed in EULER. Description of the methods, their application, and obtained results are described in Chapter 5.

Finally, in ordre to help the preparation of realistic scenarios of trafic demands over the Internet, we also focused on p2p systems which have driven a lot of attention in the past decade as they have become a major source of internet traffic. Indeed, this development has crucial implications for traffic engineering and information diffusion at the same time. Thus, models able to generate synthetic traffic and diffusion data which mimic accurately real p2p activity. Chapter 6 addresses this question, by analysing the p2p traffic dataset presented in Deliverable D3.2 that records p2p activity at a remarkable scale. We show how relevant properties extracted from the dataset may bring insights of how to model relevant diffusion and traffic demands.

Chapter 2

Inference of the degree distribution of core routers

Active measurements to study the Internet topology have been led and extensively studied since the seminal papers of Pansiot *et al.* [95] and of Faloutsos [53]. The classic method, to build maps by merging *traceroute*-like paths from a limited number of monitors to many targets [95, 53, 87] has been shown to lead to intrinsically biased results [120, 76, 5, 44, 61, 62, 79]. The hope that increasing the size and the quality of the measurement data set would overcome this bias has led to multiple other studies [18]. But the work of Lakhina *et al.*, first experimentally [76], and then formally in the case of a single source [5], revealed that the Power law (and even the heterogeneous) shape of the degree distribution found by the classic method may be *only* a measurement bias, since actual homogeneous graphs could display heterogeneous degree distribution with the paths-merging map method, even with a very large dataset.

We present in this chapter the advantage of a new approach proposed in the EULER project, which makes it possible to accurately infer the actual degree distribution of the core Internet topology. In order to make it effective in practice, we explore all details of this approach, including its expected outcome through simulations, technical difficulties, and assessment of results. We implement it using PlanetLab and finally obtain the first reliable estimate of the degree distribution of routers in the core Internet topology.

2.1 The approach

Both RFC 1122 and 1812 state that when a monitor m sends an UDP packet to a target IP address t on an unallocated port, then the target t should answer with an ICMP Destination Unreachable (Code 3/Port unreachable) packet to m. An important detail is that the source of this ICMP packet is in principle the IP

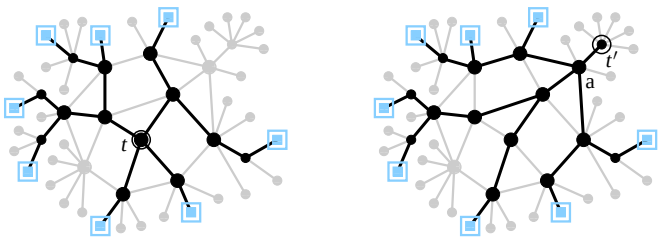


Figure 2.1: Left: an example of target t in the core. The routes coming from the set of monitors go through all of its neighbors. Right: an example of target t' in the border. Only the neighbor a of t' is accessible from the core. In the two figures, the light colored squares stand for the monitors, the big nodes denote a node in the core while the small ones represents the nodes in the border and the black links belong to a route from a monitor to the target.

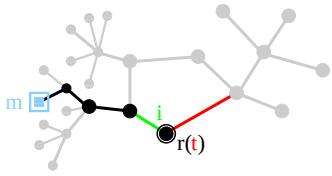


Figure 2.2: UDP Ping send an UDP message from m towards a router r designed by one of its addresses t towards an unallocated port. The router $r(t)$ replies using its interface i.

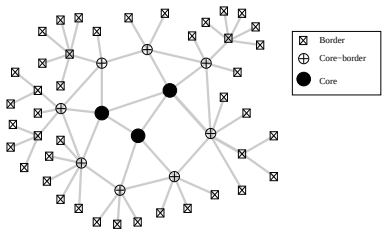


Figure 2.3: A graph can be divided into three parts: the core, the border, and the core-border. Border nodes are the ones removed when we iteratively remove nodes of degree one. Core node are the one remaining after this process. Core-border are core nodes directly connected to border nodes.

address of the interface by which t sends the packet. As a consequence, if a set M of monitors succeeds in sending such UDP probes in a way such that for each interface i of t there is in M a monitor m_i to which t answers through its interface i, the set M of monitors is able to discover *all* interfaces of t. We will use this as our basic measurement primitive, see Figure 2.2.

Now let us divide the Internet topology into three parts: the border, the core, and the core-border (included in the core), see Figure 2.3. Border nodes are the ones removed when we iteratively remove nodes of degree one, i.e. the nodes of initial degree one, the nodes which have degree one when these have been removed, and so on. The core nodes are the remaining nodes. Among them, some are linked to a border node, and we call them core-border nodes. Notice that for each border and core-border router r there is a set of nodes which necessarily go through r to reach the core (these are trees routed at r). We denote this set B_r for all r.

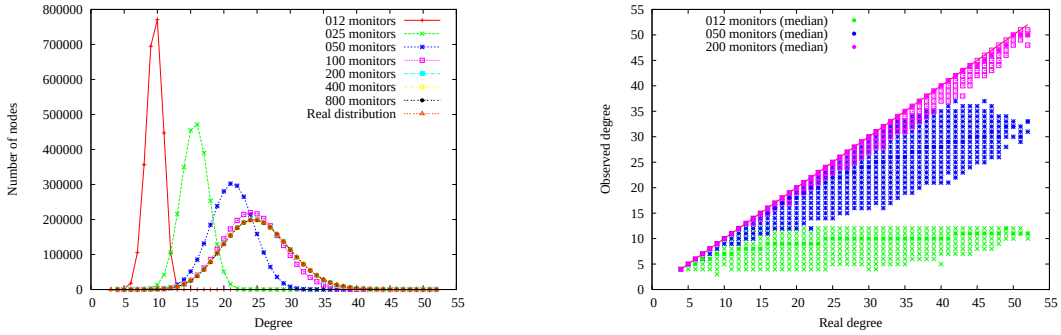
If we target a core router r with a reasonably low degree, then it seems reasonably easy to build a set of monitors M able to measure it. For instance, if r has degree 3 we only require that the monitors in M are distributed enough to get answers from all 3 interfaces of t. However, if r has degree k then getting answers from all its interfaces requires at least k monitors, and certainly significantly more as several monitors may get answers from a same interface. If the set M is not large enough, or poorly distributed, then one may underestimate the degree of r.

If we target a border router r, then appropriately determining its degree requires that we have in M at least as many monitors in B_r as r has border interfaces, which is not feasible in general. This is why our method is not suited for degree estimation of border routers, and we aim at estimating the degree distribution of core routers only.

The particular case of border-core routers deserves special attention, see Figure 2.3. In such cases, one may easily estimate the core-degree of r (i.e. its number of interfaces connected to core nodes) but not its border-degree (i.e. its number of interfaces connected to nodes in B_r).

Finally, our method consists in the following steps:

1. Build a set M of monitors as large and distributed in the Internet as possible;
2. Build a large set T of target core routers chosen uniformly at random;
3. Estimate the degree of each target t in T using distributed measurement from M;
4. Assess the quality of the obtained estimate.



(a) Observed degree distributions

(b) Comparison between observed and real degrees for all nodes

Figure 2.4: Simulations on a Poisson graph with 2.5 million nodes and mean degree 25

The degree distribution of the nodes in T then gives an estimate of the actual degree distribution of all core routers. The lager T is, the more accurate this estimation will be. In addition, the degree of some targets may be underestimated, depending on M, and so we must carefully assess the quality of obtained estimates.

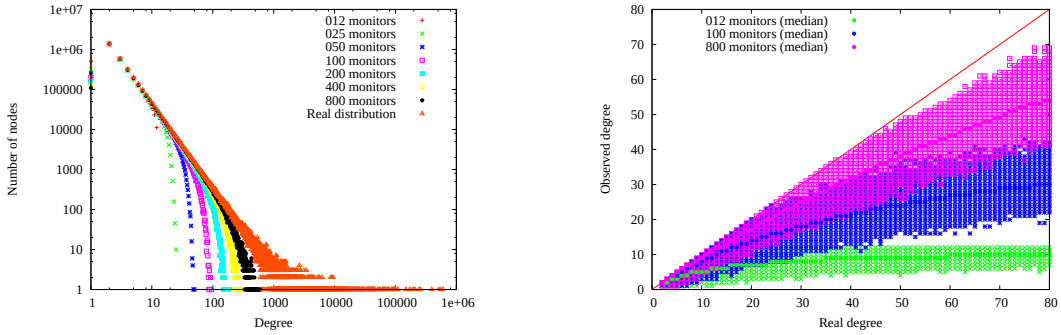
2.2 Proof of concept

The goal of this section is to estimate the theoretical relevance of our approach by means of simulations. Assuming that we are able to build an appropriate set of monitors and targets, the key questions we want to answer are: what is the risk that our estimate of the degree of a given node is different from its actual degree, and how many monitors do we need to have an accurate estimate of the degree distribution?

For this purpose, we conduct simulations as follows (see [41] for more details): we consider different kinds of artificial graphs to model the topology; we use as monitors random nodes with degree one (representing end-hosts); and we use *all* core targets (i.e. nodes in the graph obtained by iteratively removing degree one nodes). We then assume that each target answers to probes from each monitor randomly using one of its interface which starts a shortest path from the target to the monitor. We use two different kind of topology: one with a Poisson degree distribution, which is a typical homogeneous distribution, and one with a Power law degree distribution, which is a typical heterogeneous distribution.

Figure 2.4(a) shows the observed degree distributions for a Poisson graph of 2.5 million nodes and mean degree 25, using respectively 12, 25, 50, 100, 200, 400 and 800 monitors. As one could expect, with 12 monitors (which is less than the average degree) the degree distribution is poorly estimated. Nevertheless, it is remarkable that, even with this poor level of quality, the nature of the distribution is properly discovered: though its parameter are modified, the observed degree distribution is still Poisson-like. When the number of monitors increases, so does the quality of the observed degree distribution, and the observed distribution and the real distribution becomes visually indistinguishable with 200 monitors. This is strengthened by the plot on Figure 2.4(b) which shows the scatter plot of real degree (on the x-axis) and observed degree (on the y-axis) for all targets in the core. We can see that using 200 monitors, the estimate degree of all nodes is quite close to its real degree. Thus, we can conclude that our method performs very well on Poisson graphs.

The situation with Power law distribution is similar. Figure 2.5(a) shows the results of the simulation we conduct on a Power law graph with 10 million nodes and exponent 2.1. Again, we can see that the quality of the observed distribution, which is poor using 12 monitors, increases with the number of monitors. When this number is sufficient, the distribution is properly observed, except for very large degrees where we observe a cut-off in the distribution, close to the number of monitors we use. As we mentioned previously, this comes from the limitation of our method that we identified a priori: the observed degree cannot exceed the number of monitors, and more generally, the observation becomes inaccurate for targets whose degree is too closed to the number of monitors. On the other hand, for reasonably low-degree targets, let say up to 20, the observed distribution and the real one are visually indistinguishable when using 200 monitors. This fact is again reinforced by the scatter plot of real degree



(a) Observed degree distributions

(b) Comparison between observed and real degrees for all nodes

Figure 2.5: Simulations on a power-law graph with 10 million nodes and an exponent of 2.1

and observed degree for all targets in the core (Figures 2.5(b)). We can see that using 200 monitors, the estimate degree of low-degree nodes is quite close to the actual one, showing that for this type of nodes, our method performs very well for Power law graphs as well.

One may wonder if these results still hold for graphs of different size and with different parameters, average degree for Poisson graphs and exponent for Power law graphs. These questions were investigating in [41], as well as the influence of some other parameters of the simulations. It turns out that the conclusions we derive here are still valid for different size and parameters. In particular, [41] shows that the size has very little effect, if any, on the quality of the observation, both for Poisson graphs and Power law graphs. Then, the conclusion obtained by simulations on graphs of some millions of nodes must still hold for graphs of the size of the Internet, with hundreds of millions of nodes.

To summarize, we observe that both for Poisson graphs and Power law graphs, the degree distribution is correctly observed by our method as the observed distribution fastly tends to the actual one when the number of monitors augments. This conclusions have to be tempered with for the high-degree nodes in Power law graphs, for which we obtain a cut-off in the observed distribution. This is not a real obstruction since we do not intend to observe high degree nodes with our method, and the part of the distribution corresponding to low-degree nodes is quite accurately observed. Moreover, the simulations show that not only the degree distribution is properly estimated, but also the degree of each node as far as this degree remains reasonably low, which is precisely the range of degree we are interested in observing in the Internet.

2.3 Monitors

Our method relies on the use of a large set M of monitors distributed in the Internet. It is crucial that this set is large because the observed degree of targets is bounded by $|M|$ (each monitor observes only one interface, with several monitors observing the same one in general). It is also crucial that these monitors are well distributed in the Internet, because the observed interfaces are the ones used by the target to answer probe packets; monitors located at a same place probably lead to the observation of a same interface.

However, it is not easy to have a monitor set that is both large and well distributed: for example, it is straightforward to add monitors by multiplying machines on a given observation site, but this will not increase the global quality of the measurement. Our monitor set could be biased by construction: it may be easier to obtain monitors in the same region than distributed over the whole Internet. Therefore, we need tools to assess the quality of a set of monitors, and identify which monitors are actually co-located. This also plays a key role in our target selection method described in Section 2.4.

2.3.1 Similarity

Intuitively, two close monitors m and m' in monitor set M lead to similar observations in our measurements: most targets will answer to their probes using the same interface. Conversely, if most targets

answer with the exact same interface to two given monitors, then they can be considered redundant regarding the quality of the monitor set.

However, given that a target can (and often will) have less interfaces than the size of the monitor set, two monitors may observe the same interface *by chance*, while not being co-located at all. And the lower the degree of the target, the higher the chances that two, non-colocated monitors, observe the same interface. For example, if a target t has exactly two interfaces i and i', and uses them exactly as much in general, then one given monitor m has a prior probability of observing each interface of exactly $P(m(t) = i) = P(m'(t) = i') = \frac{1}{2}$. More importantly, two non-colocated (or independent) monitors m and m' have a significant probability of both observing one given interface of t, equal to $\frac{1}{4}$.

In general, for a target t with exactly d interfaces equally used, two independent monitors have a probability of observing one given interface of t equal to $\frac{1}{d^2}$. Therefore, the higher the degree of the target, the less likely it is for two monitors to observe the very same given interface for this particular target.

To capture the similarity between observations of the core by two monitors, we compute what we call their *similarity* as the expression of the number of times they observe the same interfaces for the targets over the course of the measurement, weighted with a factor accounting for the unlikeliness to observe the same interface without being colocated.

$$\sigma(m,m') = \sum_{t \in T, m(t)=m'(t)} deg(t)^2$$

To enhance this key element of the monitor set evaluation, we further extend this definition to iterated measurements using the same sets of monitors and targets. We name $\hat{m}(t)$ the union of the interfaces observed by m over the multiple iterations of the measurement (which can be different from $\{m(t)\}$), and $\hat{deg}(t) = |\bigcup_{m \in M} \hat{m}(t)|$. And we extend the definition of similarity between monitors as :

$$\hat{\sigma}(m,m') = \sum_{t \in T} \frac{|\hat{m}(t) \cap \hat{m'}(t)|}{|\hat{m}(t) \cup \hat{m'}(t)|} \hat{deg}(t)^2$$

Notice that two co-located monitors *must* have a very high similarity; however, it is possible that two non-colocated monitors have a very high similarity. Our expectation, confirmed in Section 2.6 is that it is necessary to strengthen the very high similarity criterion with an *a priori* knowledge of the monitor set to clearly draw classes of colocated monitors, that observe the core very similarly.

In order to identify colocated monitors, and to check that they indeed lead to very similar observations, we use their IP address prefixes. We denote B the length of the longest common prefix of their IP addresses (expressed in their 32 bits form) :

$$B(m,m') = LCP(IP_{binary}(m), IP_{binary}(m'))$$

We state that m and m' are colocated if $B(m,m') > 24$, as measurement in Section 2.6 shows that ensures very high similarity, and very high confidence that m and m' are actually colocated. Since the relation defined by having a longest common prefix higher than 24 is an equivalence relation, , we compute the equivalence class of each monitor: $\bar{M} = M/\ = \{\bar{m} | m \in M\}$.

$|\bar{M}|$ is the number of non-colocated monitors of the monitor set, and is therefore a more precise indication of the quality of the monitor set than $|M|$.

Finally, notice that some monitors in M may be deficient or may experience problems like failures or network shutdowns during the measurement. We identify these monitors by computing for each monitor m the number of targets from which m received an answer. If this number is too low (which we decide by observing the distribution of obtained values, see Section 2.6), we discard m. In this way, we ensure that all remaining monitors significantly contribute to the measurement.

2.3.2 Colocation

Besides observed, *a posteriori* similarity, monitors may or may not be actually colocated *a priori*, i.e., be connected to the Internet, and to the Internet core in particular, in a very similar way. Formally, m and m' are *colocated* iff they are leaves of the same tree on the IP graph. This can be expressed in terms of routes

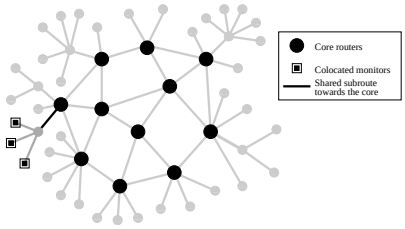


Figure 2.6: If multiple monitors are colocated, then they will share a common subroute towards reaching the core. If one is able to detect such subroutes for a given subset of monitors, then one can assume this subset is a colocation class.

: m and m' are colocated iff all the routes from m and m' truncated when they reach the Internet core have a non-empty common suffix.

To decide whether two monitors are colocated, we use an active probing method resembling UDP ping (and UDP traceroute), which we call UDP explore. UDP Explore floods the IP neighborhood of a monitor with increasing TTLs probes targeted at random IP addresses, in order to reach *all* the routers between a monitor and the Internet core. The flooding stops whenever more than exactly one router replies to a given TTL, meaning that the monitor is a that distance (in terms of IP hops) to the core. The output of UDP explore is a chain of hosts (either a star ("*") or an IP address) $(u_n(m))$ such that all the probes sent at TTL n replied with a star ("*", ie timeout) or with u_n.

Measuring these chains for all the monitors allows us to identify precisely which monitors are colocated, since then two monitors m and m' iff $(u_n(m))$ and $(u_n(m')$ have a common suffix. See Figure 2.6.

2.4 Targets

The method described above relies on our ability to select uniformly at random a core router in the Internet[1]. There is no direct way to do so, though. On the contrary, it is easy to select uniformly at random IP addresses, as they are nothing but 32 bit integers. Of course, sampling such a random integer does not necessarily lead to a *valid* IP address: the address may belong to unallocated or private ranges, and more generally the address may not be associated to any machine in the Internet. Going further, the address may belong to a machine which does not give any answer to our probes (UDP packets) and/or does not belong to the core of the Internet (it may for instance belong to an end-host).

More precisely, given any IP address i, the possible situations at a given moment in time are as follows:

- i is not a routable address (private address, address in an unallocated range, ...),
- i is a routable address but is not associated to a machine in the Internet,
- i is a routable address associated to a machine in the Internet which does not answer to our probes,
- i is a routable address associated to a core Internet router which does answer to our probes.
- i is a routable address associated to another machine (end-host or border router) in the Internet which does answer to our probes.

The first situation may be identified easily using known classes of reserved addresses. The second and third situations are identified by sending one probe and getting no answer for it. We will see in the following how to distinguish between the two last cases, the ones where we receive an answer from a core router or another machine in the Internet (end-host or border router).

Before this, let us notice that when a given core router does not answer to our probes, the absence of answer is *a priori* not related to its degree. As a consequence, the degrees of such core routers are representative of the ones of all core routers, and in particular they have the same distribution. We may therefore obtain the degree distribution of core routers by measuring the number of core routers which answer to our queries.

[1] *Uniformly at random* means that all routers are selected with the same probability.

To do so, we still have to distinguish between answers from core routers and answers from other machines in the Internet (end-hosts or border routers). First notice that a core router necessarily has more than just one interface or else it cannot *route*. Moreover, from the definition of a core router, at least two of its interfaces are connected to core routers and our measurement method will see at least two of them if the monitors are reasonably well distributed. Instead, an end-host has only one interface, which will be discovered by our probes.

Let us now consider a router r which is either a border-core router or a border router. Remember (Section 2.1) that this means that router r belongs to a tree between a border-core router (root of the tree) and end-hosts (its leaves). When we measure the degree of such a router, several situations may occur. In the vast majority of cases, as long as our monitor set is a small fraction of all possible ones, none of the machines in set B_r is a monitor. Then, our measurement method is able to observe only the interfaces of r that are *directed toward the core*, that is those interfaces used to send packets to core routers. If r is a border router then it has only one interface directed toward the core (r_0 on the figure), which is the only interface of r we discover with our probes. Thus, we may make no difference between r and an end-host, which is quite satisfying for us as we intend to discard both end-hosts and border routers from our data set. In the other case, where r is not a border router but a border-core router, we may discover several interfaces of r. But, since none of our monitors belongs to set B_r, then we can only discover the core interfaces of r, which is again precisely our goal.

The difficult case to be carefully examined is the one where some of our monitors belong to set B_r: as our monitor set is large, and as the number of targets increases, the probability that we have in our target set a router r with one or several monitors in B_r increases. In this case, an interface of r, connected towards B_r, is observed by the monitors in B_r.

Fortunately, in this case we are able to detect when an interface is actually not directed towards the core. While we use UDP Explore mainly to detect colocated monitors, we can also use its output to list all such interfaces. Indeed, the output of UDP Explore from a monitor is exactly the list of all the interfaces oriented towards the monitor that are not in the core, that is, for a given monitor, all the interfaces not oriented towards the core that may be observed by Distributed UDP Ping. We use this observation to build a blacklist, which is the reunion of all the interfaces observed by UDP Explore. Whenever an interface is observed that is on the blacklist, it is discarded for not being an interface in the core.

Fortunately, in this case we are able to distinguish between the interfaces of r that are directed towards the core and those that are not. Indeed, consider the case where several machines in B_r are in our monitor set. Then they clearly are what we have called *colocated monitors* in Section 2.3. Thanks to the previously described method, we are able to identify these colocated monitors, and we are able to detect when an interface is observed only by monitors that are colocated (in the same colocation equivalence class). Since all the other interfaces of r are observed by monitors that are not in B_r, we only have to discard interfaces that are seen by at most 1 colocation class of monitors. The remaining interfaces of r after removing such interfaces are called the *core* interfaces of r.

Finally, we build our target list as follows. We sample random 32 bit integers and discard the corresponding IP address i if one of the following is true:

- i is in an unallocated or private range,
- a probe sent to i does not lead to a valid answer,
- our method observes only one interface for i (it is an end-host or a border router),
- our method observes more than one interface for i, but only 0 or 1 of these interfaces remains after removing the non-core interfaces.

Additionally, after the measurement, interfaces from the blacklist are discarded.

2.5 Bias correction

The procedure above (Section 2.4) selects uniformly at random IP addresses of core routers (among the ones answering to our probes, representative of all core routers). It however does not select uniformly at random *core routers* themselves: a core router with k interfaces has k possibility to be chosen, so there is a bias due to degree. More precisely, if p_k is the fraction of core routers with degree k, the probability $\tilde{p}_k$

to choose a core router of degree k is proportional to kp_k. Thus, if we knew the number of interfaces k of the core router to which corresponds each of the IP addresses we selected, then we could easily infer the true degree distribution by applying the correction formula $p_k = \tilde{p}_k/(k.\Sigma_{i\geq 2}\tilde{p}_i/i)$, where p_k is the real proportion of routers of degree k in the core of Internet and $\tilde{p}_k$ is the observed proportion in our sample.

The obstruction to proceed tis way is that we do not know the number of interfaces k of the core routers in our sample since we cannot discover all of their non-core interfaces. On the other hand, we hopefully observe all of their core interfaces and, as explained earlier, we are able to decide for each of the observed interface whether it is a core interface (observed by several monitors) or a non-core interface (not observed or observed only once after choosing a unique representative for each set of colocated monitors, which we call quotienting). Then, we artificially change our selection method so that the probability of selecting a core router is proportional to its number of interfaces in the core instead of its total number of interfaces : we simply decide to discard all routers that have been selected by an interface that we do not observe or that we observe with only one monitor after choosing a representative for sets of colocated monitors. In this way, we obtain the desire property and the correction formula given above becomes valid, up to substituting the degree k of routers by their core degree k'.

Proceeding this way even offers a subtle but very interesting advantage: it improves the accuracy of our sampling for the high-degree routers, which are the most difficult to obtain since they are less numerous. Indeed, the statistical relevance of a sample to evaluate the rate of a given category of elements is directly related to the number of such elements n_i contained in the sample. If $n_i >> 1$ then the accuracy of the observation is good, while on the opposite, if $n_i \sim 1$ one can have only low confidence in the observed proportion of the referred category of elements. It turns out that, with our sampling method, the presence of degrees in our sample is biased toward the highest degree, since the probability to choose a core interface of a given core router is proportional to $k'.p_{k'}$, where k' is the core degree. This augments the accuracy of observation of the proportion of these nodes, whose number is very low in the Internet[2], while the accuracy of observation of the proportion of low-degree nodes is guaranteed by their high number.

2.6 Measurement

To confirm the feasibility of our method and to get first results, we have conducted actual, real-world measurement.

2.6.1 Data collection

The monitor set was composed of $\sim$700 machines from the PlanetLab platform, capable of sending, receiving and processing UDP and ICMP packets towards the large number of targets, as required by our method (2.3). The target set was composed of $\sim$ 3 millions UDP ping responding, uniformly randomly chosen among routable IP addresses, as described previously (2.4).

The most lengthy part was the creation of the target set : since relatively few routable IP addresses respond to UDP ping, a large number of addresses must be probed to construct a suitable target set, and this part took about 10 hours, while each single measurement pass took no more than 4 hours. The whole measurement (creation of the list, three rounds of measurement) lasted less than 24h hours. Each pass is considered an independent data set for the first step of data post-processing, and the three rounds are merged for the last step of data post-processing.

2.6.2 Data cleaning and filtering

To meet the requirements described in the previous sections (See 2.3 and 2.4), we had to apply both data pre- and post-processing.

Invalid IP addresses, or IP addresses filtering UDP or ICMP were immediately discarded upon the target list creation. The distributed measurement was then executed and the data retrieved for further post-processing. Monitors and targets producing a suspiciously low amount of data (monitors observing less than 80% of the targets, and targets being observed by less than 80% of the monitors) were purged for

[2]This is a widely acknowledged fact, which we confirm in the result section 2.7.

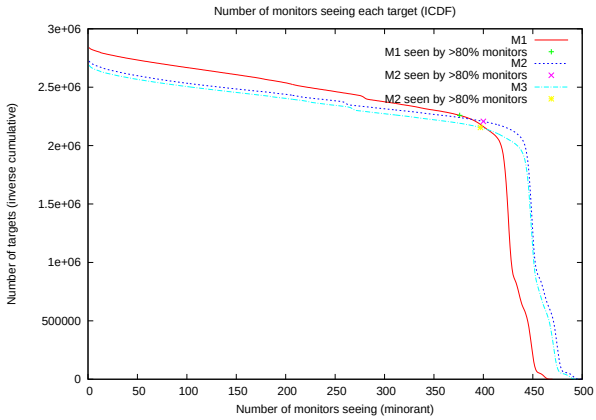


Figure 2.7: Inverse cumulative function of the distribution of the number of monitors that get responded at least once, per target. We keep 80% of the targets from each data set, cuttin g at a point where the number of monitors that get responded is high enough (respectively 2025391, 1928313 and 1901725). All other targets are considered suspicious and are discarded.

being unreliable. Targets generating multiple responses from single probes, or not answering at least once with the interface they are designated by (their *nominal* interface), or not answering at least once with an interface different from the one they are designated by (and hence being of degree 1, and therefore not in the core), are all removed from the data set for the reasons explained in 2.4.

Dataset	M1	M2	M3
Initial number of monitors	619	625	622
Initial number of targets	2849740	2734548	2699642
Total dataset size (monitor/interface/target triplets)	1082691302	1100694241	1077678410
Targets generation multiple responses for single probes	10150	9842	11048
Monitors observing too few targets	65	73	72
Targets being observed atleast once by their nominal interface	2641485	2526446	2491990
Targets being observed atleast once by an other interface	215514	215228	215159
Targets being observed by atleast two different interfaces	7259	7126	7507
Remaining targets after target filtering	5593	5623	5619
Remaining monitors after monitor filtering	421	442	442
UDP Explore blacklist length	440	440	440
Blacklisted interfaces in the unfiltered data	1040	1107	1097
Blacklisted interfaces in the filtered data	0	0	0

Some targets generated more than one response for at least one monitor. We consider this as suspicious, probably being due to packet duplication or improper configuration. Since the number of such targets is low, we deny them the benefit of the doubt and remove them from our data set. About 10^4 such targets are removed from each data set.

Targets that respond significantly less than the other, and monitor that get responded significantly less than the others, are also considered suspicious and discarded. About 5×10^5 targets (see Fig 2.7) and 70 monitors (see Fig 2.8) are removed from the data set.

Finally, targets that have only one observed interface at this point, or that never use the interface by which they have been selected (to prevent the selection bias) are discarded, for a total of about 2.8×10^6) for each data set.

After this post-processing, the data is assumed to be clean, and valid for the representative remaining $\sim$ 5600 targets.

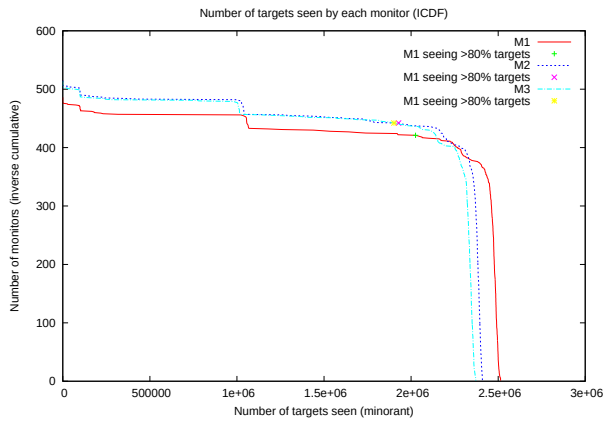


Figure 2.8: Inverse cumulative function of the distribution of the number of targets that respond at least once per monitor. We keep 80% of the monitors from each data set, cutting at a point where the number of responding targets is high enough (respectively 376, 400 and 397). All other monitors are considered suspicious and are discarded.

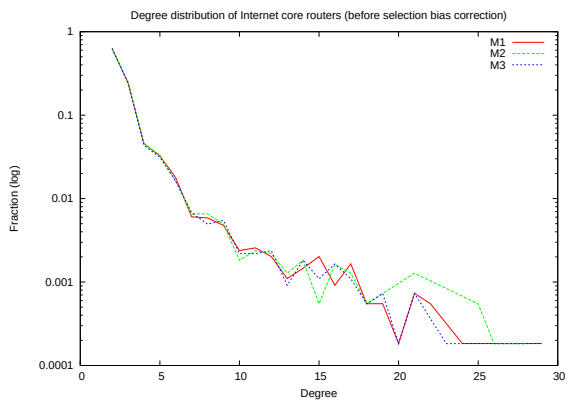


Figure 2.9: Observed degree distribution for the three measurements, before selection bias correction (lin-log scale).

2.7 First results

2.7.1 Observed degree distribution

The computation of the number of different observed interfaces of each of the targets allows us to draw the number of interfaces distribution of our target set 2.9.

However, this distribution does not account for the selection bias described previously (2.5). To derive a proper estimation of the distribution of the number of interfaces of the Internet core, we apply the distribution transformation $(k, p_k) \rightarrow (k, p_k/k)$. 2.10

2.7.2 Assessment of results

We explore here several approaches to assess the quality of obtained results.

Colocation Using the UDP Explore method presented in 2.3.2, we were able to identify classes of colocated monitors, as defined previously, that is the number of actual different vantage points in the monitor set. From the total number of monitors, 203 actual classes were extracted, of average size 2.11. Most

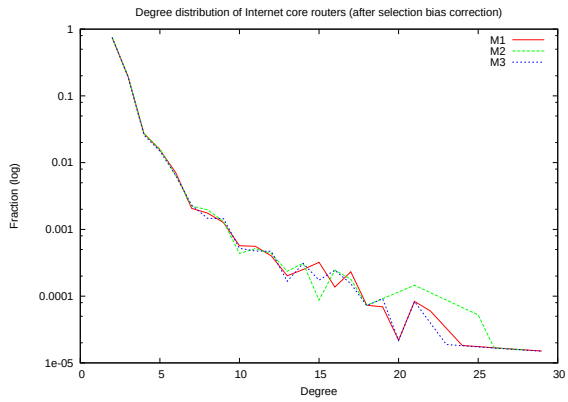


Figure 2.10: Observed degree distribution for the three measurements, after selection bias correction (lin-log scale).

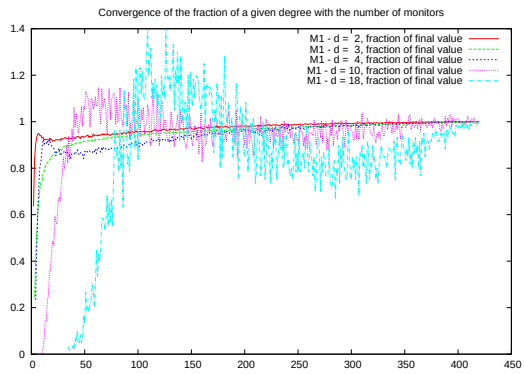


Figure 2.11: Convergence of the fraction of nodes of degree k with the number of monitors

classes regroup monitors obviously from the same Planetlab registered institution, typically matching a *.domain.ext-like pattern. We have used these classes to further analyze the quality of the monitor set.

Impact of the number of monitors A crucial issue with the method we have presented is to use a monitor set that is large enough and distributed enough to actually observe *all* the interfaces of each given target. We do not have a direct method to support this assertion, but we can at least check that our monitor set is *convergent*. More specifically, we check that the marginal addition of new monitors from our monitor pool does not affect a lot the shape of the measured distribution. To do so, at post-processing, we re-sample monitors from our monitor pool and extract the fraction of nodes of every degree observed in average for every number of monitors between 1 and the maximum number of monitors in our pool. We then plot the evolution of the average fraction (over its final value, observed when all the monitors are included in the monitor set) and verify that this fraction converge. (Figures 2.11, 2.12)

Although it provides merely an indication of local optimality and not a global validation, the convergence of the fraction of nodes of a given degree (and to an extend, of the degree distribution itself) with the number of monitors proves the stability of the method with the addition of marginal monitors to the monitor set.

Balance between interfaces As a validation of our method to decide whether an observed interface is in the core or not, we compare the number of the interfaces classified as non-core by our method for monitors observed with degree two, with a rough estimation of the expected number of such interfaces in our sample. Let us suppose that the routes from every representative monitor to the core of Internet are distinct and of average length 5. Since there are about 400 representative monitor in our measurement, there are

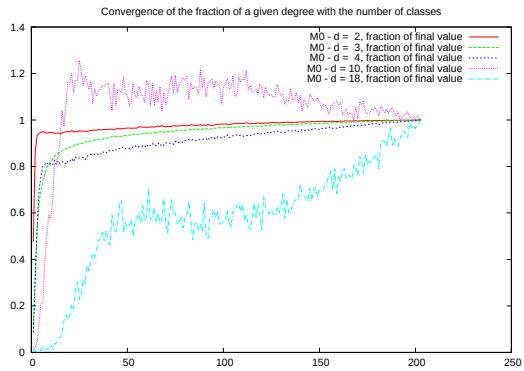


Figure 2.12: Convergence of the fraction of nodes of degree k with the number of classes

about 2 000 routers for which we can observe a non-core interface. Moreover, since we discard the targets that do not respond with the interface we selected, there are 4 000 interfaces that lead to the presence of these 2 000 routers in our sampling : each of them can be selected by its unique core interface (we consider only monitors of observed degree two in the estimation) or by its unique interface directed toward a representative monitor. This gives a number of $4\,000 \sim 2^{12}$ IP addresses out of 2^{32} IP addresses. Our sample of IP addresses for a 3 000 000-targets measurement was made by randomly generating roughly $80\,000\,000 \sim 80.2^{20}$ 32-bit integers. Our sampling should then contain about $80.2^{20}.2^{12}/2^{32} = 80$ routers of degree two having one non-core interface. This number, obtained by a rough estimation, is to be compared with the number of interfaces of routers of observed degree two that were classified as non-core interfaces by our selection method.

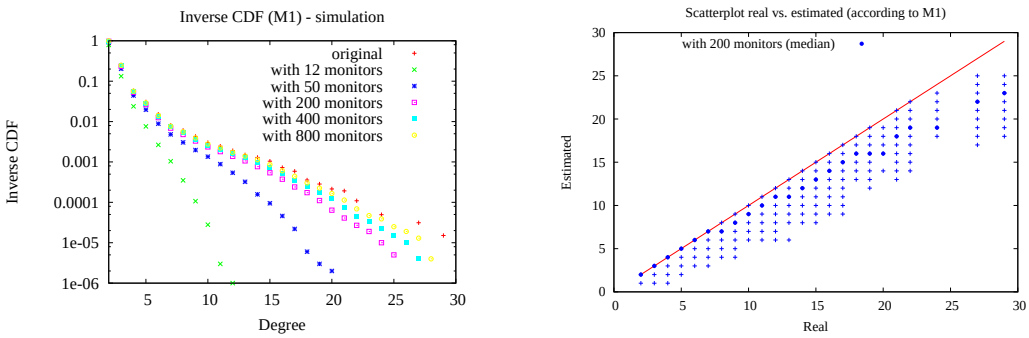
Iterated measurements In order to corroborate the former results, we propose in this section another validation based on the simulation framework discussed in Section 2.2. The purpose is to answer the following question: let us suppose that the real degree distribution is exactly the one measured above, then how our measurement techniques would perform? The simulation enables to investigate in particular whether it respects the shape of the distribution or not and how accurate is the method according to the real degree of the nodes.

Note that the validation presented in this section differs from the one proposed in Section 2.2 since the graphs used previously belonged to two very specific classes, following either a Powerlaw or a Poisson distribution. In this section, the distribution cannot be formally characterized as one of those two possible types and it seems then natural to assess the relevance of the method by means of the same simulation framework.

Simulations The simulation setup consists in the following: we generated 5 different graphs of 1 million nodes according to each of the 3 measured distributions; for each of the graphs, we chose 5 different sets of nodes defined as the virtual monitors and simulated the measurement process from those monitors towards all the other nodes. This represents 75 different simulations for which we tested sets of 12, 25, 50, 100, 200, 400 and 800 monitors.

Note that this process avoids the issue of co-located monitors since each virtual monitor in the simulation is chosen as a node of the graph and then would rather stand here as the entry point of a set of co-located monitors in the measurement framework. As such, the number of monitors in the simulation that would be similar to the one used in the PlanetLab measurement is around 200.

We first present in Figure 2.13 the simulation results regarding the assessment of the degree distribution. In Figure 2.13(a), one can see the inverse cumulative degree distribution observed with the different sets of monitors. It shows that with a number of monitors higher than 200, the shape the original distribution seems preserved, even if the proportion of large degree nodes tends to be less accurately estimated. This results is coherent with previous studies. Besides, the plot tends to show that the proportion of small degree nodes seems to be particularly well estimated. Manual investigations confirmed this statement for nodes with degree less or equal to 10 and for sets of monitors higher than 200.



(a) Inverse cumulative degree distribution

(b) Real vs. estimated degree

Figure 2.13: Qualitative assessment of the observed distribution

We now analyze the accuracy of the method and in particular to assess how close the measurement of a degree is to the real value. Indeed, while the number of correctly estimated nodes is very high in the simulations (96.3% of the nodes have been correctly estimated over the 75 simulations), which explains why the distribution is globally well estimated, one may wonder how the mis-estimated nodes impact the quality of the estimation.

Figure 2.13(b) presents the correlation between the real and the estimated degree for all the nodes (we only plot the 200 monitors case for readability sake). An emphasis is made on the plot to outline the median value for each degree. This figure shows that the method is also efficient on measuring the degree of specific nodes. In particular, one can see that the median value remains close to the real one, even for the highest degrees. Moreover, even for the highest degrees, the estimation value is never far from the real one: for instance, 18 has been the worst estimation made for a 29-degree node; 17 for a 27-degree one and 14 for a 22-degree one.

These analyses confirm that the degree estimation provided by the proposed the method is reliable whether we are interested in the global distribution or in the degree of a specific node in the network.

Chapter 3

Determining the nature of the degree distribution

The degree distribution of a network carries a significant information regarding the structural and evolutionary properties of a network. The knowledge of the degree distribution will be helpful to allow us to discriminate between various random graph models that have been, could be or will be proposed to model the Internet. Indeed, one expects that a valid model of the Internet topology should at least approximately reproduce its degree distribution, among other properties. Consequently, we can accept or reject a proposed hypothetical model based on its ability to mimic the degree distribution of Internet.

Our goal is to select or eliminate explicative and generative models based on their consistency with the degree distribution measured in Chapter 2. In this purpose, for different classes of distributions Γ, we identify the particular distribution $D \in \Gamma$ that is the most consistent with the experimental data. We pre-assume different random graph models which generates a variety of hypothetical distributions Γ such as power law, exponential and lognormal, and apply a set of rigorous methodologies to test each of them to accept or reject these candidate conjectures. To be more precise, we parameterize a given class of random graphs by a vector Φ, and want to check whether such a random graph for one value of the vector Φ could represent the Internet. We first estimate the vector Φ of parameters for which the corresponding theoretical degree distribution D_Φ matches best our measured degree distribution $D_{Measured}$, and then test the hypothesis "could $D_{Measured}$ have been obtained by randomly selecting degrees according to D_Φ". We use different standard methodologies such as maximum likelihood estimator [90], least square estimator [103] etc to estimate the parameters Φ for different hypothetical distributions D_Φ. Next we use different statistical hypothesis test methodologies (Monte Carlo-based technique [35] and χ^2 test [102] etc) to determine if the measured degree distribution $D_{Measured}$ is not consistent with the hypothetical distributions D_Φ; based on the outcome, we reject some of the hypothetical distributions.

Classes of distributions considered

Our first and main objective is to determine whether the measured degree distribution follows a power law, which is probably the most popular model that is used to explain the evolution of a network. Nevertheless, recently few other distributions came up as the possible candidate to model different kinds of networks. Exponential degree distributions have for example been found in many real world complex networks [45]. Inspired from the non-equilibrium network theory, Deng et al. [45] constructed the network according to two mechanisms: growing and adjacent random attachment. Their results showed that many empirical datasets, such as the Worldwide Marine Transportation Network (WMTN), the Email Network of University at Rovira i Virgili (ENURV) in Spain and the North American Power Grid Network (NAPGN) closely matches with the exponential degree distribution. In [106], Sala et al. also showed that modeling large scale online social networks using power law produces a significant fitting errors, hence they proposed a more accurate node degree distribution model based on the log-normal distribution.

We will therefore test three classes of distributions, Power laws in Section 3.2, Exponential distributions in Section 3.3 and Lognormal distributions in Section 3.4.

3.1 Challenges

In this section, we focus on the challenges that we face to model Internet with different hypothetical distributions.

3.1.1 Independent samples and post-processing

Parameter identification methods and statistical tests are normally designed for experimental data sets consisting a certain number of *independent samples* obtained from some (a priori unknown) distribution. This assumption does not hold in our case, as several post-processing steps have been performed in order to clean the data and to correct biases of the real experiments. In particular, for reasons explained in Chapter 2, the degree distribution p_k resulting from the direct measurements was biased, and needed to be corrected by computing $q_k = p_k/k$ (and re-normalizing). The distribution obtained is unbiased, but does not correspond anymore to a number N of values for independent samples. In particular, one cannot identify Nq_k nodes with degree k. To avoid this issue, we will work directly on the biased data, and transform our hypotheses and distributions accordingly.

Besides, node degrees in a network are never entirely independent: Their sum over the whole network needs for example to be an even number, and we suspect that there might be stronger dependency issues. The precise influence of such dependency remains unknown, but it should be absent if the random selection of samples on which measurements are made is made with repetitions, which is always approximately the case when the proportion of selected samples is small, as in Chapter 2. In any case, we believe that the effect should be negligible with respect to other possible artifacts of the measurements and the analysis method.

3.1.2 Parameter estimation

As explained in the introduction of this section, we consider different classes of distributions Γ which can be proposed as a hypothetical model for the Internet. We would like to stress on this point that unlike ad hoc and 'a priori' hypothesis, these hypothetical models are suitably parameterized based on the experimental dataset. Moreover, in one of the predominant candidates called power law distribution, the power law behavior is usually not observed for very small degrees, but emerges smoothly for the higher degrees (see Figure 3.1).

Deciding where the power law behavior starts and how to parametrize the first part of the distribution without over-parametrizing it are challenging issues for which there is no universally agreed upon answer. The number of parameters needed to describe the first part of the distribution is in particular often not clear, and may depend on where the actual power law behavior starts. As a result, the class of models considered does not have a fixed number of parameters.

Most standard techniques for parameter estimation and hypotheses testing are designed for classes of models with fixed number parameters, and can therefore not be directly applied here.

3.2 Hypothesis: Power law distribution

In this section, we consider the power law distribution as a candidate hypothetical distribution for the measured Internet. The power law distribution is defined by

$$p_k = Ck^{-\alpha} \tag{3.1}$$

where $\alpha > 1$ is the scaling exponent, C is the normalizing constant and k is the discrete variable representing individual degrees. The exponent α can be observed as the slope in the log-log scale of the distribution. As represented in Fig. 3.1, the slope of the curve does often not follow $\alpha = 2.5$ for the few initial degrees ($k < 8$), however, for the higher degrees ($k \geq 8$) the linear slope becomes quite evident. In such a case, we say that the distribution follows power law with a slope α starting from a minimum degree $k \geq x_{\min}$. Hence we redefine the discrete power law distribution Eq. (3.1) for $x \geq x_{\min}$ with proper normalization as follows

$$p_k = \frac{k^{-\alpha}}{\zeta(\alpha, x_{\min})} \tag{3.2}$$

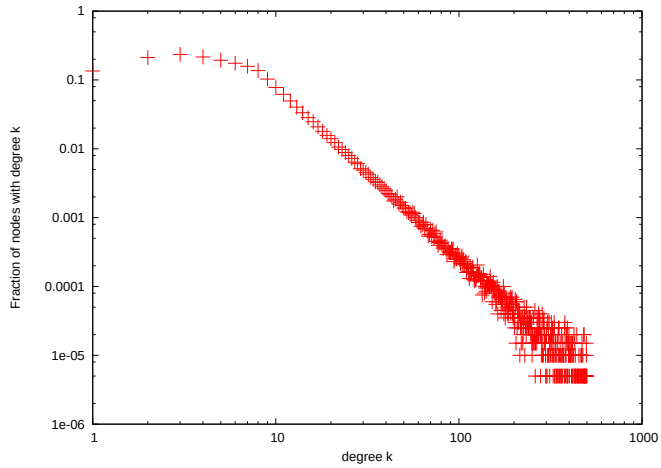


Figure 3.1: A power law distribution with $x_{\min} = 8$

where α is the scaling parameter and $x_{\min}$ is the minimum degree at which power-law behavior holds and $\zeta(\alpha, x_{\min}) = \sum_{m=0}^{\infty} (m + x_{\min})^{-\alpha}$. Our first task lies in correctly fitting this power law distribution with the experimental distribution. In order to do that, we first estimate the parameter set $\Phi = \{x_{\min}, \alpha\}$ using various statistical estimation techniques. Then we test the corresponding power law hypothesis D_Φ using different hypothesis testing methodologies.

3.2.1 Parameter estimation

In this section, we use different standard statistical estimation methodologies to compute the parameters of the power law distribution. Precisely, we focus on two popularly used methods such as the maximum likelihood estimator [90] and least square estimator [103] and in the following, we explain them one after another.

Maximum likelihood based estimation In statistics, maximum-likelihood estimation (MLE) is a popular method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model's parameters [90].

For the particular case of the power law, let us first assume that the lower bound $x_{\min}$ is known, and estimate the slope α. We use maximum likelihood estimation to compute the estimator $\hat{\alpha}$ of the exponent from a discrete dataset containing N observations. Suppose that the N observed degrees $x_1, x_2, \ldots, x_N$ are assumed to be independent and identically distributed according to p_k. Then, the likelihood function can be expressed as

$$L(\alpha | x_1, x_2, x_3, \ldots, x_N) = p_{x_1} \times p_{x_2} \times p_{x_3} \cdots \times p_{x_N} \tag{3.3}$$

$$= \prod_{i=1}^{N} \frac{x_i^{-\hat{\alpha}}}{\zeta(\alpha, x_{\min})} \tag{3.4}$$

Taking the logarithm, we obtain

$$\ln L = -N \ln \zeta(\alpha, x_{\min}) - \alpha \sum_{i=1}^{N} \ln x_i$$

Taking as estimator $\hat{\alpha}$ the value α that maximizes the log-likelihood function by setting $\frac{\partial \ln L}{\partial \alpha}|_{\alpha=\hat{\alpha}} = 0$, we obtain

$$\frac{\zeta'(\hat{\alpha}, x_{\min})}{\zeta(\hat{\alpha}, x_{\min})} = -\frac{1}{N} \sum_{i=1}^{N} \ln x_i$$

The solution of this equation admits the following closed form approximation (see Appendix B.3 in [35])

$$\hat{\alpha} = 1 + N \left[\sum_{i=1}^{N} \ln \frac{x_i}{x_{\min} - 0.5} \right]^{-1}$$

where $x_i, i = 1 \ldots N$ are the observed values of x such that $x_i \geq x_{\min}$. We can also transform the above equation in terms of the measured frequencies p_k^{Meas}

$$\hat{\alpha} = 1 + \frac{1}{\sum_{k=x_{\min}}^{k_{\max}} p_k^{Meas} \left(\ln \frac{k}{x_{\min} - 0.5} \right)} \tag{3.5}$$

We now turn to the problem of estimating the lower limit $x_{\min}$ from the data, for which we follow the approach of Clauset et al. [35]. The basic assumption is that, if we choose too low a value for $x_{\min}$ we try to fit a region which does not fall under the power law behavior (in Fig. 3.1, the region is $k < 8$). On the other hand, if we choose too high a value for $x_{\min}$, we are effectively throwing away legitimate data points in the region $k < x_{\min}$, which increases both the statistical error on the scaling parameter and the bias from finite size effects. Our goal is to find a good compromise between these cases. The fundamental idea behind the method is very simple: we choose the value $\hat{x}_{\min}$ that makes the probability distributions of the measured data and the best-fit power-law model as similar as possible above $\hat{x}_{\min}$. In general, if we choose $\hat{x}_{\min}$ higher than the true value $x_{\min}$, then we are effectively reducing the size of our data set, which will make the probability distributions a poorer match because of statistical fluctuation. Conversely, if we choose $\hat{x}_{\min}$ smaller than the true $x_{\min}$, the distributions will differ because of the fundamental difference between the data and model by which we are describing it. In between lies our ideal value.

We use Kolmogorov-Smirnov (KS) statistic for quantifying the distance between the data and the fitted model. Let $P(k)$ be the cumulative density function of the hypothetical power-law distribution for $k \geq x_{\min}$. We compute the CDF as

$$P(x) = \frac{\zeta(\hat{\alpha}, k)}{\zeta(\hat{\alpha}, x_{\min})}$$

which has the nice normalization property as $P(x) = 1$ at $x = x_{\min}$. On the other hand, we suitably re-normalize the experimental distribution to obtain cumulative density function $S(k)$ for $k \geq x_{\min}$. Here also we observe $S(k) = 1$ at $k = x_{\min}$. Finally the KS-statistic can be computed as

$$d = \max_{k \geq x_{\min}} |S(k) - P(k)|$$

Our estimated $\hat{x}_{\min}$ is then the value of $x_{\min}$ that minimizes d. A brief outline of the methodology is provided as Algorithm 1.

Algorithm 1: Parameter estimation using MLE

Input: Degree distribution p_k
Output: Estimated parameters $\hat{x}_{\min}, \hat{\alpha}$
Two temporary vectors $\alpha[k_{\min} \ldots k_{\max}], d[k_{\min} \ldots k_{\max}]$ are used in this algorithm
foreach $x_{\min}$ *such that* $k_{\min} \leq x_{\min} \leq k_{\max}$ **do**
  Estimate $\hat{\alpha} = 1 + \frac{1}{\sum_{k=x_{\min}}^{k_{\max}} p_k \left(\ln \frac{k}{x_{min} - 0.5} \right)}$
  foreach k *such that* $x_{\min} \leq k \leq k_{\max}$ **do**
    $P(k) = \frac{\zeta(\hat{\alpha}, k)}{\zeta(\hat{\alpha}, x_{min})}$
    $S(k) \leftarrow$ Re-normalized p_k in $x_{\min}$ to $k_{\max}$
  $d[x_{\min}] = \max_{k' \geq x_{\min}} |S(k') - P(k')|$
  $\alpha[x_{\min}] = \hat{\alpha}$
$\hat{x}_{\min} = i$, for the minimum value in the vector $d[k_{\min} \leq i \leq k_{\max}]$
$\hat{\alpha} = \alpha[\hat{x}_{\min}]$

Least square fit method The method of least square assumes that the best-fit curve of a given dataset is the curve that has the minimal sum of the deviations squared (least square error) from a given set of data [103]. In our case, the measured degree distribution of Internet of size n can be described as $(1, p_1)$, $(2, p_2)$, ..., (n, p_n), where p_k is here the measured proportion of nodes with degree k. Using least square method, we aim to propose a mathematical model $p_k^{Th} = f(k, \Phi) = Ck^{-\alpha}$ (parameter set $\Phi = (C, \alpha)$) that fits with this experimental distribution. This is important to note that the normalizing constant C and the exponent α are mutually dependent parameters. However using least square method, we primarily focus to estimate the exponent α only, giving less attention to C. Instead of working directly on the original function, we prefer to linearize it to reduce the weight given to the high degree nodes. The function can be linearized by taking logarithms

$$\ln p_k^{Th} = \ln C - \alpha \ln k$$

The fitting function $f(k, \Phi)$ has the deviation (error) r from each data point, i.e., $r_i = \ln p_{k_i} - f(k_i, \Phi)$. According to the method of least squares, the best fitting curve has the property that the sum S, of squared deviations

$$S = \sum_{i=0}^{n} r^2$$

is minimum. To find the values of the parameters C and α which minimizes S, we set the partial derivatives $\frac{\partial S}{\partial \ln C} = 0$ and $\frac{\partial S}{\partial \alpha} = 0$. Solving the equations, we find the parameters

$$\ln \hat{C} = \frac{(\sum_{i=1}^{n} \ln p_k)(\sum_{i=1}^{n} (\ln k)^2) - (\sum_{i=1}^{n} \ln k)(\sum_{i=1}^{n} \ln k \ln p_k)}{n \sum_{i=1}^{n} (\ln k)^2 - (\sum_{i=1}^{n} \ln k)^2}$$

$$\hat{\alpha} = -\frac{(n \sum_{i=1}^{n} \ln k \ln p_k) - (\sum_{i=1}^{n} \ln k)(\sum_{i=1}^{n} \ln p_k)}{n \sum_{i=1}^{n} (\ln k)^2 - (\sum_{i=1}^{n} \ln k)^2}$$

The quality of the least square fit is measured with the help of coefficient of determination R^2 which quantifies how well the proposed model is able to fit the experimental data points. R^2 is computed as the square of the sample correlation coefficient between experimental and predicted values [104]. This coefficient can be expressed as

$$R^2 = 1 - \frac{S_{err}}{S_{tot}}$$

where

$$S_{tot} = \sum_k (p_k - \langle p_k \rangle)^2, \qquad S_{err} = \sum_k (p_k - Ck^{-\alpha})^2$$

In the above, $\langle p_k \rangle$ is the mean of the observed data

$$\langle p_k \rangle = \frac{1}{n} \sum_{i=1}^{n} p_k$$

3.2.2 Hypothesis testing

The tools described in the previous sections allow us to fit a power-law distribution to the measured Internet dataset and provide good estimates of the parameters. However, they hardly tell us anything about whether the data are well fitted by the power law. In particular, data that are actually generated from a different distribution can always be fit to a power-law model, but the fit may be very poor. In practice, therefore when considering the measured degree distribution of Internet, our challenge is to decide not only what the best parameter choices are but also whether the power-law distribution is even a reasonable hypothesis to model Internet. In this section, we use different classical hypothesis testing methodologies such as Monte Carlo, χ^2 test etc to test the validity of the power law models. We first describe a hypothesis testing methodology based on Monte Carlo technique proposed in [35] by Clauset et al., and then a more standard χ^2 test.

Monte Carlo method Given an experimental degree distribution $D_{Measured}$ and the corresponding fitted hypothetical power-law distribution D_Φ, we want to know whether "that hypothetical distribution is a likely model given the experimental data". We answer this question by computing a p-value. By definition [107], a p-value quantifies the probability that our data were drawn from the hypothesized distribution, based on the observed goodness of fit.

If the p-value is close to 0, then it is unlikely that the data are drawn from a power law. If it is closer to 1 then the data is more consistent with the hypothesis of a random generation using a power law, but the latter hypothesis is of course not guaranteed. Note that when the data are indeed generated according to the hypothesis, the expected p-value is 0.5.

We numerically compute a p-value following the Monte Carlo based based methodology proposed by Clauset et al. in [35]. We generate a large number of synthetic data sets Syn_Φ drawn from the power-law distribution D_Φ that best fits the observed data $D_{Measured}$, fit each data set individually to its own power-law distribution, calculate the KS statistic for each one relative to its own best-fit model, and then simply count what fraction of the time the resulting KS statistic is larger than the value d observed for the experimental data. This fraction is our p-value. This is important to note that we create synthetic datasets Syn_Φ that have a distribution similar to the experimental data $D_{Measured}$ below $x_{\min}$ but that follow the power law distribution D_Φ above $x_{\min}$. The steps to compute p-values are as follows:

1. Determine the best fit of the power law D_{Φ_M} to the experimental Internet degree distribution $D_{Measured}$. Estimate the parameter set $\Phi_M = \{x_{\min}, \alpha\}$.

2. Calculate the KS statistic of the best-fit power law D_{Φ_M} to the data $D_{Measured}$.

3. Generate a large number of synthetic data sets Syn_Φ (of same size as experimental dataset) from the parameters Φ_M, estimate the parameters $\Phi_S = \{x_{\min}, \alpha\}$ to fit each of them with a hypothetical power law distribution D_{Φ_S}, and then calculate the KS statistic for each fit. Note crucially that for each synthetic distribution Syn_Φ, we compute the KS statistic relative to its own best-fit power law distribution D_{Φ_S}, not relative to the original distribution D_{Φ_M}. In this way we ensure that we are performing for each synthetic data set the same calculation that we performed for the experimental data set.

4. Calculate the p-value as the fraction of the KS statistics for the synthetic data sets whose value exceeds the KS statistics for the real data.

5. If the p-value is sufficiently small, the power-law distribution can be ruled out.

χ^2 test In addition to the Monte Carlo method, we also use χ^2 test [102] to examine whether "The hypothetical power law distribution is a likely model the measured Internet degree distribution". If $p_k^{Measured}$ for $1 \leq k \leq n$ is the measured degree distribution of the Internet (assuming $x_{\min}$ fixed) and $p_k^{Th} = f(k, \Phi) = \frac{k^{-\alpha}}{\zeta(\alpha, x_{\min})}$ is the hypothetical distribution with parameter $\Phi = \{x_{\min}, \alpha\}$, then we compute the χ^2 statistic as

$$\chi^2 = \sum_{k=x_{\min}}^{n} \frac{(np_k^{Measured} - np_k^{Th})^2}{np_k^{Th}} \tag{3.6}$$

This is crucial to note that, since the hypothetical distribution p_k^{Th} is defined within the interval $x_{\min} \leq k \leq n$, we need to re-normalize the empirical degree distribution $p_k^{Measured}$ within the same interval.

Like Monte Carlo method, here also we assess the confidence of the hypothesis using p-value. The p-value is calculated by comparing the value of the χ^2 statistic to a χ^2 distribution keeping in mind the number of "degrees of freedom" which is again calculated by the number n of values of degree for which we have an estimated probability , minus the reduction in degrees of freedom m. The reduction in the degrees of freedom m is calculated following $m = s + 1$ where s is the number of parameters for the hypothetical distribution.

Note that the number n of values of degrees for which we have an estimated probability is unclear: Since there is a priori no maximal degree in a power-law, this n could be infinite, or very large. The χ^2 test is however only valid asymptotically, for number N of samples that are sufficiently large with respect to n and the distributions considered. It does thus not provide trustful results if we chose a too large n, and we must therefore arbitrary fix a sufficiently small maximal degree.

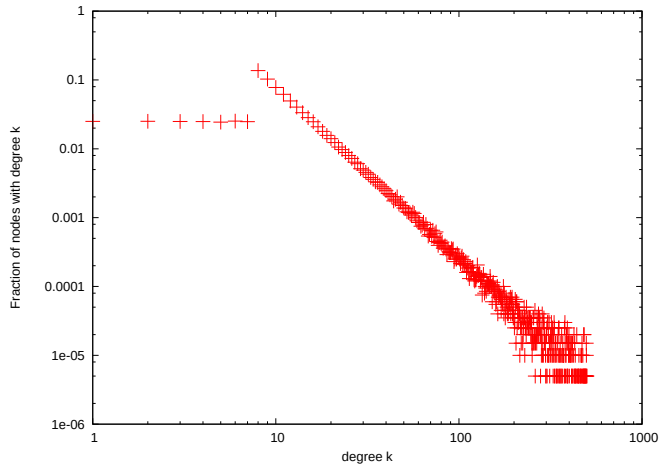


Figure 3.2: Example of artificially generated dataset following a power-law, with $N = 10^4$, $x_{\min} = 8$ and $\alpha_0 = 2.5$.

These ambiguities are one of the reasons for which we use the Monte-Carlo based approach of Section 3.2.2.

3.2.3 Validation

Before applying the estimation and testing the relatively novel methodologies described in Section 3.2.1 and in 3.2.2 on the experimental data, we first validate them on the artificially generated datasets. Next we illustrate the generation of the synthetic datasets and show how accurately we can recompute the a priori parameters and calculate p-value.

Artificial datasets We fix values of $x_{\min,0}$ and α_0, and generate a discrete synthetic dataset Syn_{NU} containing N individual degrees x_i in the following way. $N - N_{tail}$) are taken randomly uniformly between 1 and $x_{\min,0} - 1$), and 'N_{tail}' of them are selected according to a strict a power law distribution with slope α_0 starting at $x_{\min}$. This is done using the following expression

$$x = \lfloor (x_{\min,0} - 0.5)(1-r)^{-1/(\alpha_0 - 1)}) + 0.5 \rfloor,$$

where r is a random number uniformly selected between 0 and 1.

Remark: For practical implementation reasons, it can be convenient to impose a maximal degree $k_{\max}$ in the synthetical datasets and to discard all the generated degrees larger than that $k_{\max}$. When this option is selected, one should be very cautious to specify a sufficiently large $k_{\max}$, especially for small values of α, for otherwise one could discard a significant number of data points, resulting in erroneous estimations and validations. Hence, while generating synthetic datasets for the computation of p-value, we dynamically set the $k_{\max}$ based on the estimated exponent α such that the fraction of discarded data points remains less than a predefined fraction (say 10^{-4}) following the expression $k_{\max} = \left(\frac{x(\alpha-1)}{c}\right)^{1/(1-\alpha)}$.We observe that for low α, the $k_{\max}$ gets a quite high value so that only a few data points are discarded.

An example of resulting distribution for $N = 10^4$, $x_{\min,0} = 8$ and $\alpha_0 = 2.5$ is presented in Fig. 3.2.

Validation results Applying the MLE-based approach on the synthetic dataset Syn_{NU} as represented in (Fig. 3.2), we calculate $\hat{\alpha} = 2.51$ and $\hat{x}_{\min} = 8$. More generally, we have generated datasets for various values of α_0 between 2 and 4 and re-estimated the parameters using the MLE-based approach. The results, presented in Fig. 3.3, show that the estimated slopes $\hat{\alpha}$ match the actual values very closely (usually with a difference smaller than 0.02).

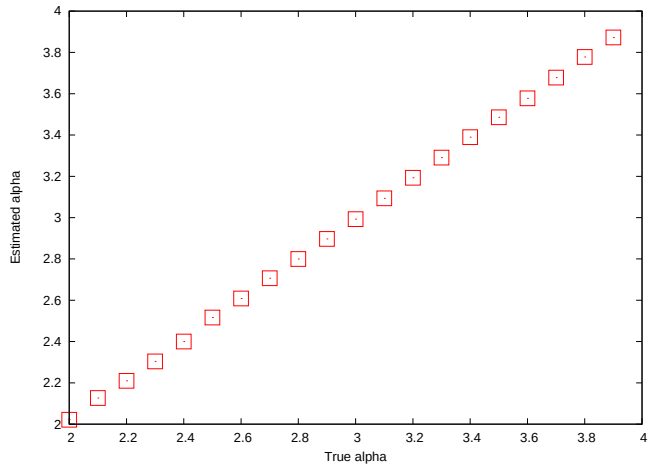


Figure 3.3: Comparison between the estimated slope $\hat{\alpha}$ and the actual slope that was used to generate the synthetical datasets, for $N = 10^4$ nodes, and $x_{\min,0} = 8$. The figure shows that the estimated and actual slopes match very closely.

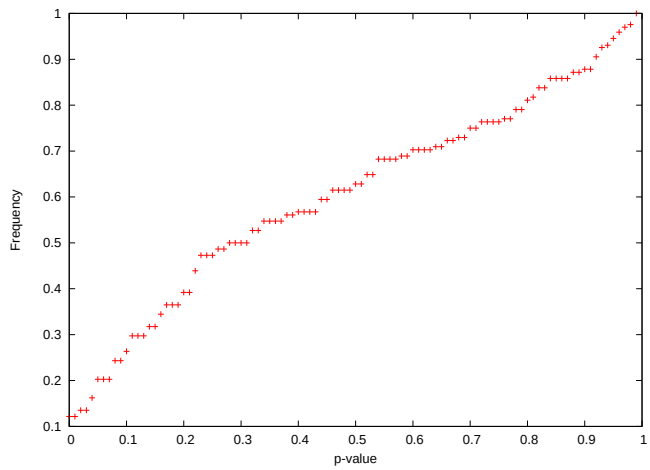


Figure 3.4: Cumulative distribution funcitn of the computed p-value for 120 different artificial datasets with parameters $\alpha_0 = 2.8$, $x_{\min,0} = 10$ and $N = 10^5$.

Let us now analyze the Monte-Carlo based p-value evaluation method. By definition of the p-value, when the artificial datasets are indeed generated by taking N i.i.d. realizations of a random variable following a power law, the p-value should behave as a uniform random variable. Several reasons related to the heuristic nature of the method applied, approximations in the estimation methods, and the possible distortion of KS-distances for different values of α could however have prevented this from happening.

To validate the method used, we have generated 120 different artificial datasets for same parameters $\alpha_0 = 2.8$, $x_{\min,0} = 10$ and $N = 10^5$. For each of them, we have estimated the parameters $\alpha_0, x_{\min,0}$, and computed the p-value. The results, presented in Fig. 3.4, show that these p-value do indeed behave approximately as a uniform random variable.

3.2.4 Measured degree distribution

The detailed measurement methodology and tools used to measure the distribution of degrees on the Internet have been explained in Chapter 2. Three experiments were performed at different times, yielding 5453, 5482 and 5478 valid measurements respectively. In addition, we will consider a larger dataset $(1+2+3)$ resulting from the union of all these valid measurements. Note that this combinations of the

Experiment	$\hat{\beta}$	$\hat{x}_{\min}$	p-value (%)
1	3.245	5	2.0
2	3.331	5	8.8
3	3.276	5	1.8
1+2+3	3.230	5	4.6

Table 3.1: Values of the power-law parameters obtained using the MLE-based estimator, and corresponding p-values obtained the Monte-Carlo approach.

Experiment	$\hat{\beta}$	$\hat{x}_{\min}$	p-value (%)
1	3.386	5	12
2	3.386	5	16
3	3.385	5	10
1+2+3	3.392	5	7

Table 3.2: Values of the power-law parameters obtained using the MLE-based estimator, and corresponding p-values obtained the Monte-Carlo approach.

three experiments should be considered with appropriate care, as these there were conducted at different times, possibly in slightly different conditions.

As already explained in Section 3.1.1 we need to work on the biased initial measurements, because our methodology assumes that we have access to a certain number of independent measurements randomly drawn from the same distribution, and this does not hold true once the distribution has been corrected for bias.

We use the accurately measured biased degree distribution to *indirectly* estimate parameters of Internet. Lets assume that the hypothetical degree distribution of the Internet follows power law degree distribution $p_k^{Th\_unbiased} = \frac{k^{-\alpha}}{\zeta(\alpha, x_{\min})}$. Subsequently, the biased hypothetical degree distribution becomes

$$p_k^{Th\_biased} = k \times \frac{k^{-\alpha}}{\zeta(\alpha, x_{\min})} = \frac{k^{-\beta}}{\zeta(\beta, x_{\min})}$$

where $\beta = \alpha - 1$. Since the empirical distribution $p_k^{Emp\_biased}$ is essentially a biased distribution, we need to estimate the parameter β of the hypothesis $p_k^{Th\_biased}$ from these experimentally measured distribution and then compute the power law exponent α of the unbiased hypothetical degree distribution $p_k^{Th\_unbiased}$.

MLE and Monte-Carlo based tests Applying the Maximum likelihood-based estimator of Section 3.2.1 and computing the p-values for the results obtained using the Monte-Carlo based method described in Section 3.2.2 yields the results presented in Table 3.1. One can see that while the power-law hypothesis is not strongly supported, it can certainly not be rejected outright. Besides, all results predict a $x_{\min} = 4$, and a value of β around 3.25 corresponding to a slope α of the degree distribution close to 4.25. A comparison between the experimental data and the estimated power-law distributions are presented in Fig. 3.5 for the three experiments.

Since we have very few nodes with high degrees, we have try applying our methodology to truncated distributions, in which we remove every node with degree higher than 20. As presented in Table 3.2, the values of the slope obtained are higher and much more uniform. A higher slope is naturally explained by the need to account for the absence of nodes at a degree above 20. On the other hand, the uniformity of the result suggests that the difference between the values $\hat{\beta}$ obtained in Table 3.1 for different experiments could be caused by discrepancies at high degrees. This hypothesis should however be considered with necessary caution, as the need to account for the absence of nodes with degree higher than 20 may ”compress” the differences between the different experiments.

Maximization of p-values It is interesting to compute the p-values using the approach of Section 3.2.2 for different values of the parameters. As can be seen in Fig.3.6, there are values of α leading to much higher p-values than those obtained by the MLE. It is therefore tempting to take as estimators of the

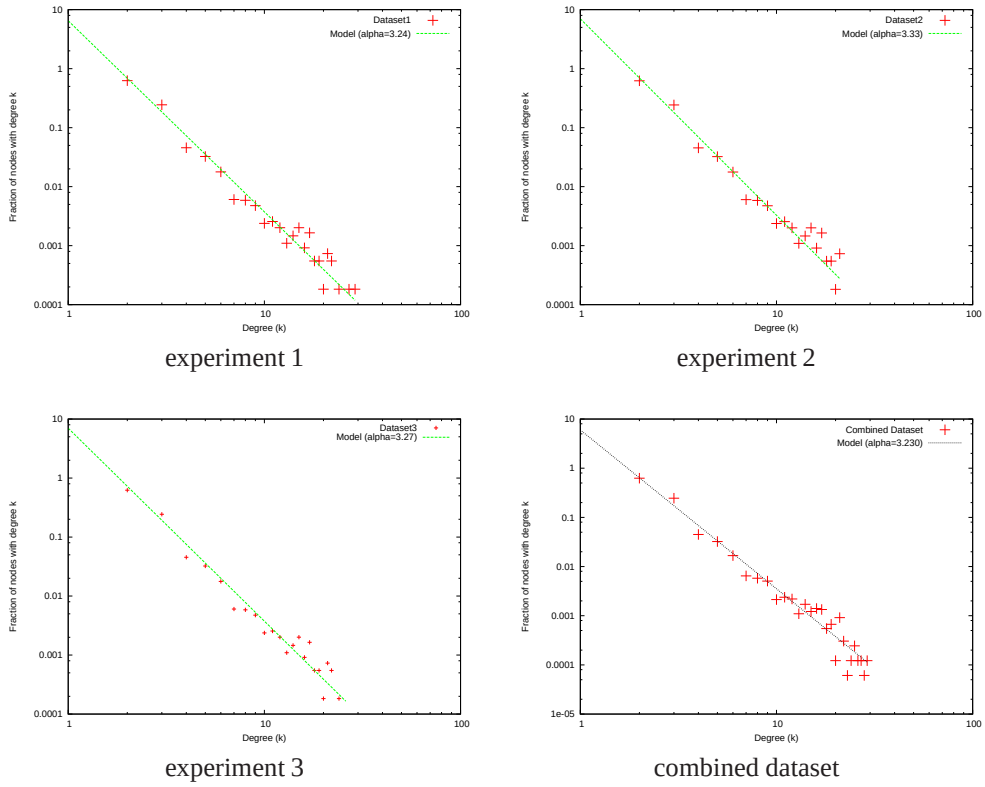


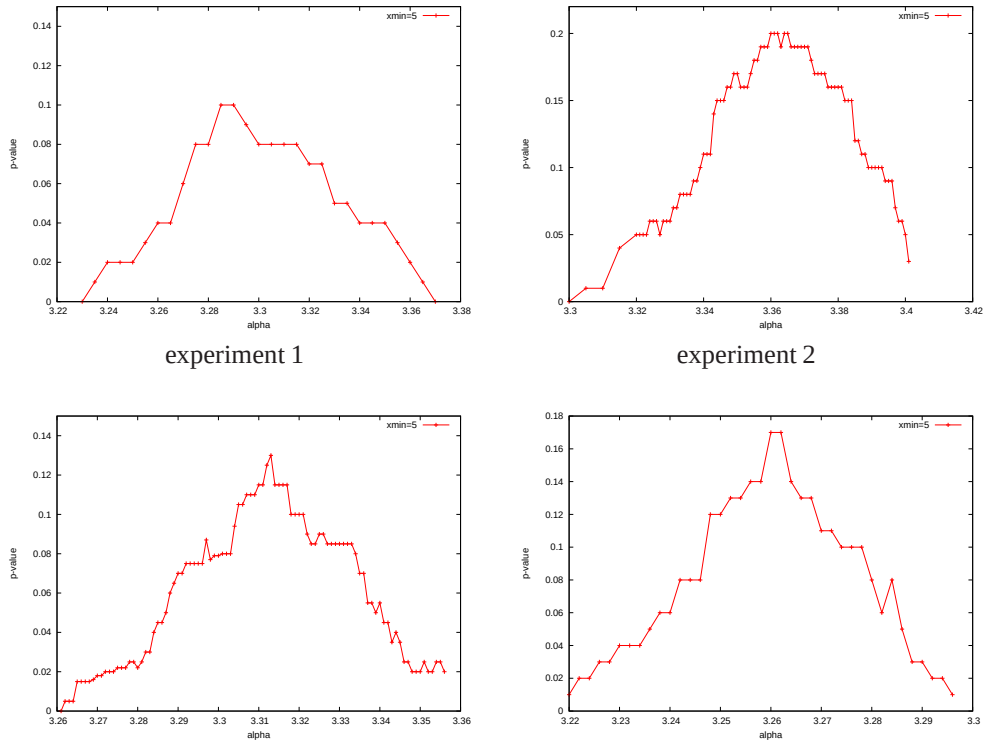
Figure 3.5: Comparison between measured data and the distribution estimated using the MLE-based method.

Experiment	$\hat{\beta}$	$\hat{x}_{\min}$	p-value (%)
1	3.29	5	10
2	3.36	5	20
3	3.31	5	13
1+2+3	3.26	5	17

Table 3.3: Values of the power-law parameters obtained by maximizing the p-values, as computed by the Monte-Carlo approach of Section 3.2.2; and corresponding "p-values".

power-law parameters the $x_{\min}$ and β that maximize these p-values. The results of this approach are presented in Table 3.3. One can see that the estimated slopes are close to but slightly higher (typically by 0.05) than those estimated in Table 3.1. The p-values are however significantly higher, lying between 10% and 20%. One should however note that these percentages cannot be formally interpreted as p-values, as the methodology used to compute them relies on the assumption that α and $x_{\min}$ were obtained by the MLE-based approach.

We have also tried to identify the best parameters for the power-law using a direct p-value approach that does not assume that we use the MLE to identify parameters. More precisely, for given parameters $\Phi_H = (x_{\min}, \beta)$, we compute the KS statistic for the difference between the measured degree distribution $D_{Measured}$ and the power-law distribution D_{Φ_H}. We then compute the KS statistic for the difference between the power law D_{Φ_H} and 100 synthetically generated datasets (on the same number of nodes as $D_{Measured}$) and take as $p-$value the percentage of those synthetic datasets for which the KS-distance is larger than that of the Measured data. We repeat this operation for all relevant values of $x_{\min}$ and β. Figure 3.9 shows the values obtained for $x_{\min} = 5$, and Table 3.4 shows the values β (for $x_{\min} = 5$) for which the $p-$values obtained are the largest. One can see that the optimal values of β are very similar to those obtained in Table 3.3 where the MLE approach was re-used in the computation of the $p-$values. But, the $p-$values themselves are much larger, being as high as 50%. Remember though that these values can again not be strictly interpreted as p-values once one select the parameters by optimizing these values.



experiment 1 experiment 2

experiment 3 combined dataset

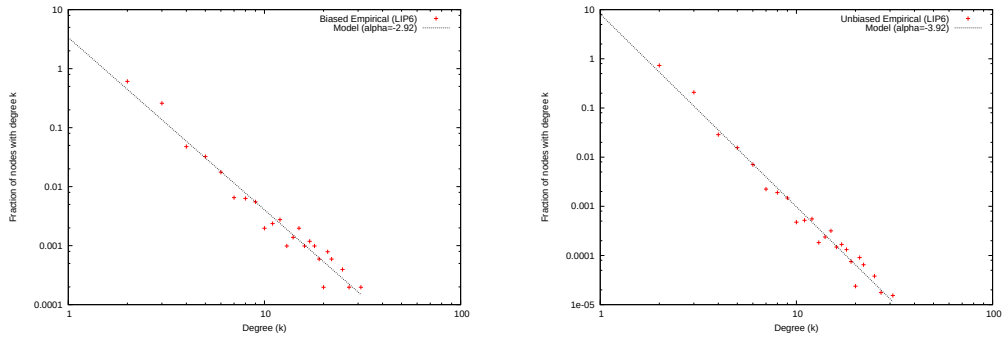
Figure 3.6: Evolution of p-values as computed by the Monte-Carlo based method described in Section 3.2.2 with α, for $x_{\min} = 5$.

Experiment	$\hat{\beta}$	$\hat{x}_{\min}$	p-value (%)
1	3.29	5	44
2	3.36	5	46
3	3.31	5	40
1+2+3	3.26	5	51

Table 3.4: Values of the power-law parameters obtained by maximizing the p-values computed by a Monte-Carlo based method using the KS-statistic; and corresponding "p-values".

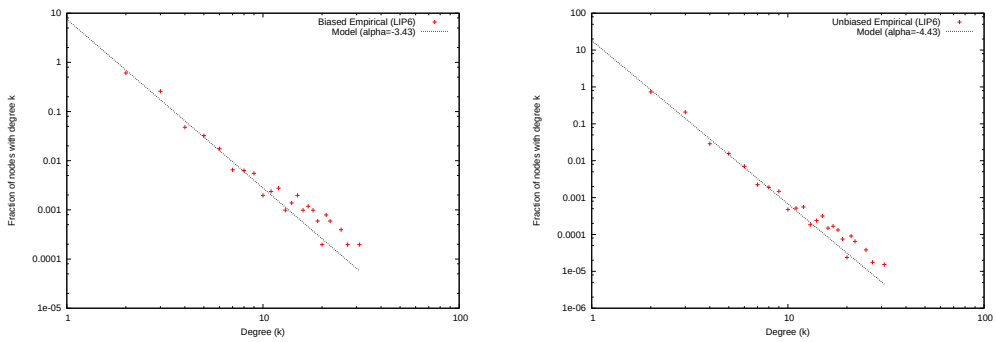
Concerning the methodology, it is interesting to note that the highest "p-values" are obtained when no maximum likelihood estimator is used at around $40\% - 50\%$. Selecting the parameters by optimizing "p-values" computed using the maximum likelihood estimator as explained in Section 3.2.2 leads to smaller values, of the order of $10\% - 20\%$, and computing the p value for the parameters directly obtained using the maximum likelihood estimator leads to even smaller values. This could indicate that the maximum likelihood is efficient to recover parameters corresponding to a small KS-statistic when the data are artificially generated, but not on the measured dataset.

χ^2 test In addition to the Monte Carlo method, we also apply χ^2 test (explained in Section 3.2.2) to examine the goodness of fit of the hypothetical model. For biased distribution with estimated $\hat{x}_{\min} = 5$ and $\hat{\beta} = 3.21$, we evaluate the χ^2 statistic as 1760.22 (using the Eq. (3.6)). The total number of data points $n = 31 - 5 = 26$ and reduction in the degrees of freedom $m = 2 + 1 = 3$. Hence the number of the degrees of freedom $= 26 - 3 = 23$. Using the χ^2 statistic, χ^2 distribution and number of the degrees of freedom, we compute a very low p-value$= 0.001$. As explained in Section 3.2.2), it is however not so clear that the hypotheses under which the χ^2 test is valid are satisfied here.



(a) Biased degree distribution (b) Unbiased degree distribution

Figure 3.7: Comparison between the measured and hypothetical degree distributions using Least square method considering all the data points



(a) Biased degree distribution (b) Unbiased degree distribution

Figure 3.8: Comparison between measured and hypothetical degree distributions using Least square method considering only initial 10 data points.

Least Square Estimator In addition to the maximum likelihood estimator, we also estimate the parameters from the empirical distributions $p_k^{Emp\_biased}$ and $p_k^{Emp\_unbiased}$ using the least square method explained in section 3.2.1. We observe that the value of the estimated parameter following the least square method heavily depends on the number of datapoint (i.e. total number of degrees) taken into consideration. For instance, in the combined $(1+2+3)$ dataset, if we consider all 31 different degrees, we compute the power law exponent as $\beta = 2.92$ with coefficient of determination $R^2 = 0.96$. When working directly on the corrected unbiased distribution becomes $\alpha = 3.92$ with $R^2 = 0.97$. In Fig. 3.7 we show a comparison between the model and measured data for both biased and unbiased distributions. However, if we consider only the initial 10 degrees, the least square method computes the exponent β as 3.43 with $R^2 = 0.97$ for the biased distribution, and $\alpha = 4.43$ with $R^2 = 0.98$ for the corrected unbiased one (see Fig. 3.8).

Obtaining much higher values of $\hat{\alpha}$ or $\hat{\beta}$ when considering only the first degrees should not be very surprising. Indeed, we know that the power-law behavior was not very pronounced for degrees smaller than 5, and smaller slopes are needed to account for this flatter start of the curve. So, to avoid this artifact, and the effect of noisy data at high degrees, we have also applied the least square estimator to versions of our data truncated on both sides. More specifically, we have removed all nodes with degree smaller than 5 or higher than 20. For the three experiments and the combined datasets, we obtained estimated of $\hat{\beta}$ between 2.802 and 2.805 (corresponding to $\hat{\alpha} \simeq 3.805$), with values of R^2 around 0.89. This suggests again that the difference between the coefficients estimated using the different experiments are mostly caused by the noisy data for high degrees.

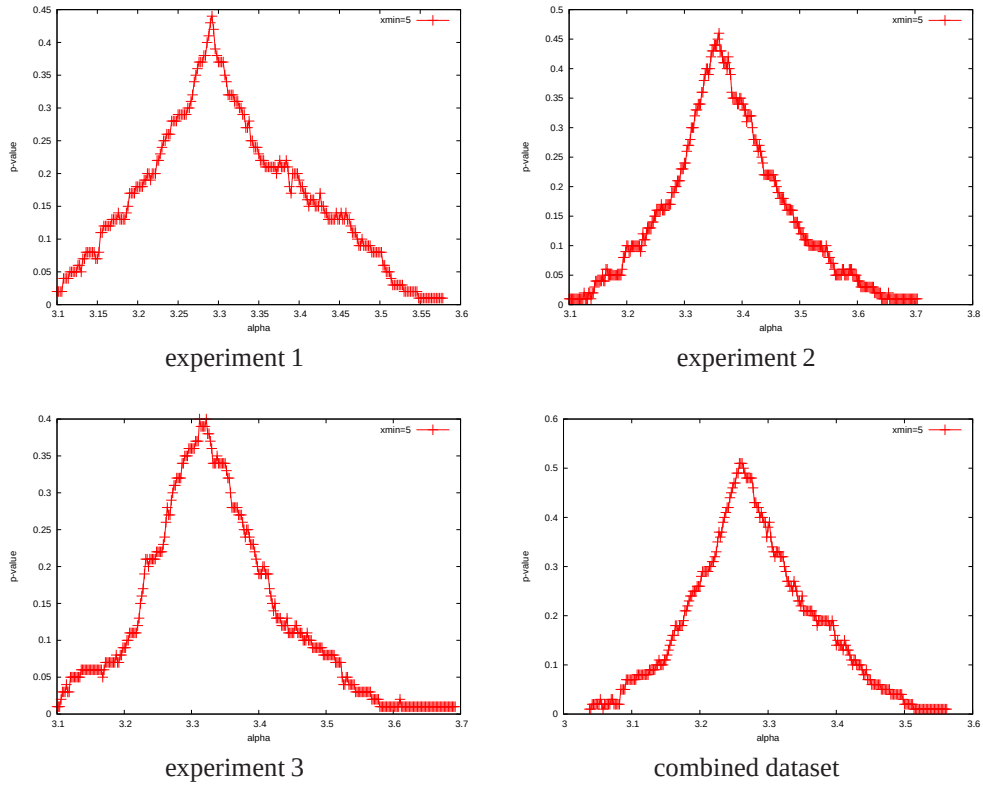


Figure 3.9: Evolution of p-values computed by a Monte-Carlo based method using the KS-statistic with α, for $x_{\min} = 5$.

3.3 Hypothesis: Exponential distribution

In this section, we consider the exponential distribution as the candidate hypothetical distribution to model Internet. We chose to work with an expression of probabilities similar to the exponential probability density function for continuous variable:

$$p_k = \lambda \exp(-\lambda k)$$

where λ is the parameter. In the previous section, we have already illustrated the fact that the actually measured degree distribution of Internet is essentially a biased distribution. Hence the hypothetical biased exponential distribution can be expressed as

$$q_k = kp_k = k\lambda \exp(\lambda k),$$

with appropriate normalization.

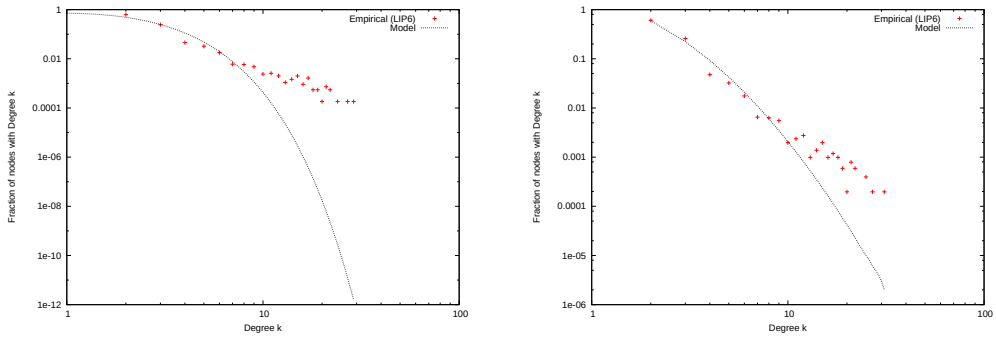
Similar to what was done for power laws, our first task is to correctly fit this biased exponential distribution with the measured degree distribution (We work here only with a combination of the results of the three experiments). In order to do that, we use MLE to estimate the exponent λ from the experimental distribution.

3.3.1 Parameter estimation

We approximate the discrete exponential distribution from the continuous function by suitable normalization as follows

$$h_k = \frac{q_k}{\sum_j q_j}$$

So we assume that the N measured degrees x_i, $1 \leq i \leq N$ are independent and identically distributed observations, coming from a discrete exponential distribution p_k with scaling parameter λ. Similar to



(a) Exponential hypothesis

(b) Lognormal hypothesis

Figure 3.10: Comparison between the measured degree distribution and the hypothetical models

Eq. (3.3), the likelihood function becomes

$$\begin{aligned} L(\lambda|x_1,x_2,\ldots,x_N) &= \prod_{i=1}^{N} h_{x_i} \\ &= \prod_{i=1}^{N} (x_i\lambda\exp(\lambda x_i))\frac{1}{\sum_j q_j} \end{aligned}$$

Taking the logarithm of the likelihood, we obtain

$$\ln L = \ln\left(\sum_j q_j\right)^{-N} + \ln\left(\prod_{i=1}^{N}(x_i)\lambda^N\exp(-\lambda\sum_i x_i)\right).$$

Taking as estimator $\hat{\lambda}$ the value λ that maximizes the log-likelihood function by setting $\frac{\partial \ln L}{\partial \lambda}|_{\lambda=\hat{\lambda}} = 0$, we obtain

$$\frac{N}{\hat{\lambda}} - \sum_{i=1}^{N} x_i - \frac{N\sum_j \exp(-\hat{\lambda}j)(1-\hat{\lambda}j)}{\sum_j j\hat{\lambda}\exp(-\hat{\lambda}j)} = 0 \tag{3.7}$$

We numerically solve the above expression to estimate the exponent $\hat{\lambda}$ for the experimental dataset.

3.3.2 Validation and parameter estimation

In order to validate the correctness of the estimated $\hat{\lambda}$, we generate a synthetic network of size N with exponential distribution with parameter λ_0. Precisely, we numerically generate the degree sequences $x_i, 1 \leq i \leq n$ following the exponential distribution with $\lambda_0 = 0.5, 1.0, 1.5$ respectively. From these degree sequences, we compute the synthetic degree distributions p_k^{EX}. We apply Eq. (3.7) on p_k^{EX} to estimate the parameters $\hat{\lambda}$ as $0.498, 1.008$ and 1.487 respectively which proves that MLE is correctly able to evaluate the true parameters.

Having validated the method, we apply Eq. (3.7) to the (biased) measured degree distribution from Chapter 2 $p_k^{Emp\_biased}$ and estimate the exponent as $\hat{\lambda} = 1.08$. Hence, we propose the null hypothesis that the best fit exponential distribution for the Internet can be expressed as

$$p_k = 1.08\exp(-1.08k)$$

Fig. 3.10(a) shows the comparison between the biased experimental distribution $p_k^{Emp\_biased}$ and the corresponding hypothetical distribution.

3.3.3 Hypothesis test

Similar to power law distribution, we use the Monte Carlo-based method to test the exponential hypothesis of the Internet degree distribution. The procedure is exactly same as described in section 3.2.2,

nevertheless for the moment we ignore the impact of $x_{\min}$ and only put our attention to λ. One word of caution regarding the computation of the KS distance between the best fit exponential distribution and the experimental distribution, which requires the normalization of the cumulative biased exponential distribution. The re-normalized cumulative distribution of the biased discrete exponential distribution can be expressed as

$$P(k) = 1 - \frac{\exp(-\lambda k(k+\frac{1}{\lambda}))}{\exp(-\lambda x_{\min}(x_{\min}+\frac{1}{\lambda}))})$$

contrary to the normalized cumulative distribution of the classical exponential distribution $P(k) = (1 - \exp(-\lambda k))$. Similar to power law distribution, first we generate a synthetic dataset following discrete exponential distribution and compute the p-value using the Monte Carlo-based method. As expected, the p-value follows random distribution. After this validation step, we compute the p-value for the (biased) measured degree distribution and obtain 0. This clearly rejects our null hypothesis of modeling Internet with exponential distribution.

3.4 Hypothesis: Lognormal distribution

In this section, we consider the lognormal distribution as the candidate hypothetical distribution for the Internet. Lognormal distributions are normally used for continuous variables. To treat discrete distribution, we chose to assign to each degree k a probability proportional to the value of the (continuous) probability density function at that k, with a proper renormalization. This gives us probabilities of the form

$$p_k = \frac{1}{k\sigma\sqrt{(2\pi)}} \exp\left(-\frac{(\ln k - \mu)^2}{2\sigma^2}\right)$$

where μ is the mean and σ is the standard deviation. Earlier, we have already illustrated the fact that the actually measured degree distribution of Internet is essentially a biased distribution. Hence the hypothetical biased lognormal distribution can be expressed as

$$q_k = kp_k = \frac{1}{\sigma\sqrt{(2\pi)}} \exp\left(-\frac{(\ln k - \mu)^2}{2\sigma^2}\right)$$

We first try to correctly fit this biased lognormal distribution with the experimental distribution. In order to do that, we use MLE to estimate the mean μ and standard deviation σ from the experimental distribution.

3.4.1 Parameter estimation

We approximate the discrete lognormal distribution from the continuous function by suitable normalization as follows

$$h_k = \frac{q_k}{\sum_j q_j}$$

We assume again that the measured degrees x_i, $1 \le i \le N$ are independent and identically distributed observations, coming from a discrete lognormal distribution p_k with mean μ and standard deviation σ. We aim to estimate μ and σ. Similar to Eq. (3.3), the likelihood function becomes

$$L(\mu,\sigma|x_1,x_2,\ldots,x_N) = \prod_{i=1}^{N} h_{x_i} \tag{3.8}$$

$$= \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{(2\pi)}} \exp\left(-\frac{(\ln k - \mu)^2}{2\sigma^2}\right) \frac{1}{\sum_j q_j}, \tag{3.9}$$

whose logarithm is

$$\ln L = \ln\left(\sum_{k=\min}^{\max} q_k\right)^{-N} + \ln\left(\frac{1}{\sigma\sqrt{(2\pi)}} - \exp\left[\frac{1}{2}\sum_{k=k_{\min}}^{k_{\max}}\left(\frac{(\ln k - \mu)}{\sigma}\right)^2\right]\right)$$

Again, we take as estimator the maximizers of log-likelihood function. We first have $\frac{\partial \ln L}{\partial \mu}|_{\mu=\hat{\mu},\sigma=\hat{\sigma}} = 0$, which becomes

$$\sum_{i=1}^{N}\left(\frac{\ln x_i - \hat{\mu}}{\hat{\sigma}^2}\right) - \left(\frac{N}{\sum_{k=k_{\min}}^{k_{\max}} \frac{1}{\hat{\sigma}\sqrt{(2\pi)}} \exp\left(-\frac{(\ln k-\hat{\mu})^2}{2\hat{\sigma}^2}\right)} \times P\right) = 0 \tag{3.10}$$

where

$$P = \sum_{k=k_{\min}}^{k_{\max}} \frac{1}{\hat{\sigma}\sqrt{(2\pi)}} \exp\left(-\frac{(\ln k-\hat{\mu})^2}{2\hat{\sigma}^2}\right) \times (\frac{(\ln k-\hat{\mu})}{\hat{\sigma}^2}. \tag{3.11}$$

Similarly setting $\frac{\partial \ln L}{\partial \sigma}|_{\mu=\hat{\mu},\sigma=\hat{\sigma}} = 0$ yields

$$\frac{N}{\hat{\sigma}} - \sum_{i=1}^{N}\left(\frac{(\ln x_i - \hat{\mu})^2}{\hat{\sigma}^3}\right) + \left(\frac{N}{\sum_{k=k_{\min}}^{k_{\max}} \frac{1}{\hat{\sigma}\sqrt{(2\pi)}} \exp\left(-\frac{(\ln k-\hat{\mu})^2}{2\hat{\sigma}^2}\right)} \times Q\right) = 0$$

where

$$Q = \sum_{k=k_{\min}}^{k_{\max}} \exp\left(-\frac{(\ln k-\hat{\mu})^2}{2\hat{\sigma}^2}\right) \frac{1}{\hat{\sigma}\sqrt{(2\pi)}} \times \left[\frac{(\ln k-\hat{\mu})^2}{\hat{\sigma}^2} - 1\right]$$

We numerically solve the simultaneous equations Eq. (3.10) and Eq. (3.12) to estimate the mean μ and standard deviation σ from the experimental dataset. Similar to exponential distribution, we validate the correctness of our estimation by numerically generate the synthetic dataset. Precisely, we numerically generate the degree sequence $x_i, 1 \leq i \leq N$ following the lognormal distribution with $\{\mu_0 = 1, \sigma_0 = .5\}$, $\{\mu_0 = 1.5, \sigma_0 = 2\}$ and $\{\mu_0 = 0.5, \sigma_0 = 0.8\}$ respectively. From these degree sequences, we compute the synthetic degree distributions p_k^{Log}. We apply Eq. (3.10) and Eq. (3.12) on p_k^{Log} to estimate the parameters $\{\hat{\mu} = 1.01, \hat{\sigma} = 0.49\}$, $\{\hat{\mu} = 1.5, \hat{\sigma} = 1.97\}$ and $\{\hat{\mu} = 0.51, \hat{\sigma} = 0.81\}$ respectively which proves that MLE is correctly able to evaluate the true parameters.

After this validation step, we apply Eq. (3.10) and Eq. (3.12) on the (biased) measurements from Chapter 2 $p_k^{Emp\_biased}$ and estimate the parameters for Internet as $\{\hat{\mu} = 0.12, \hat{\sigma} = 0.74\}$ (We work again on a combination of the three experiments described there). Hence, we propose the null hypothesis that the best fit lognormal distribution for the Internet can be expressed as

$$p_k = \frac{1}{0.74k\sqrt{(2\pi)}} \exp\left(-\frac{(\ln k - 0.12)^2}{1.09}\right)$$

Fig. 3.10(b) shows the comparison between the biased experimental distribution $p_k^{Emp\_biased}$ and the corresponding hypothetical distribution.

3.4.2 Hypothesis test

We use Monte Carlo method to compute the p-value of the lognormal hypothesis of the experimental degree distribution. The procedure is exactly same as described in section 3.2.2, nevertheless for the moment we ignore the impact of $x_{\min}$ and only put our attention to estimate μ and σ. Similar to exponential distribution, here also we raise the issue regarding the computation of the KS distance between the best fit lognormal distribution and the experimental distribution, which requires a properly normalized cumulative biased lognormal distribution. However, unlike power law and exponential distribution, the re-normalization of the biased cumulative distribution has to be performed numerically, due to the absence of analytical expression.

Computing the p-values on the (biased) degree distribution measured in Chapter 2, we obtain again 0, which rejects our null hypothesis of a lognormal distribution.

3.5 Results summary

To the best of our knowledge, this work represents the first rigorous statistical analysis of hypothesis about the degree distribution of the Internet relying on real accurate measurements. Our analysis has shown that one can definitively reject the hypotheses of exponential and lognormal distributions. Similar results not presented here apply for Poisson distributions.

The case of power-law distribution is much more nuanced. We have seen indeed that, while the p-values computed are not very high ($2-8\%$ with the standard Monte-Carlo based methodology), they are way to high to justify an outright rejection of the power-law hypothesis, especially since much higher values are obtained when the less formal maximization of the p-value approach is used ($10-20\%$ and even up to 50%). Besides, we have seen that our results are sensitive to small variations of the data or methodology (identification via regression, modified maximum likelihood, maximization of the p-value, etc.). Nevertheless, our results confirm the qualitative impression that the Internet degree distribution is ”close” to a power law for degrees higher than 4, and identify a slope close to 4.25.

More precise results would need further experiments, or the development and application of new approaches in parameter identification, allowing taking the specificities of power-law distribution into account.

Taking a step back and considering the complexity of the phenomena at stakes, one can however suspect that sufficiently long further experiment would eventually reject the power-law hypothesis, and that a modification of paradigm would be needed to make more precise statements. Indeed, our approach here has relied on the assumption that all discrepancies between the theoretical and experimental distributions are due to finite-size effects. It is however unlikely that the degree distribution was generated exactly by any nice law, that would eventually be matched by the experimental data if the size of the Internet and the number of measurements were sufficiently large. A refinement of our approach should take into account the possible presence of second order unmodeled phenomena, in order to test hypotheses such as ”The degree distribution could have been generated by a law that differs from a power law by less than a certain measure.” In the meantime, the safest practical option is to consider that the internet degree distribution is well approximated by a power-law with slope 4.1 for degrees above 4.

Chapter 4

Analysing the routing topology dynamic

4.1 Study of the IP-level routing dynamics over Radar data

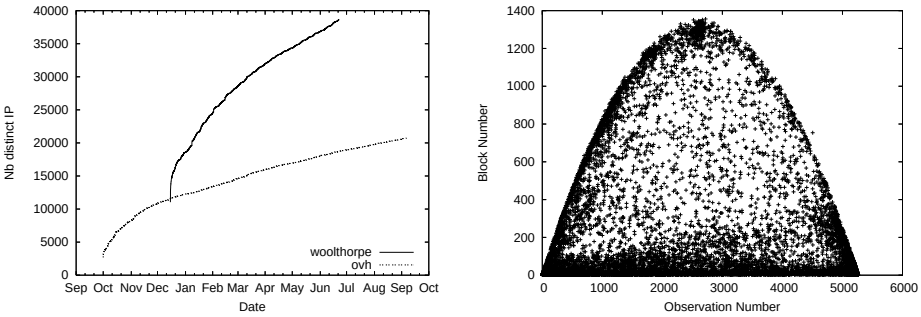
The Internet is a living system that evolves over time. Everyday, many nodes and links are added or removed, during planned maintenance or because of unexpected network failures. It is important to map the Internet topology, in particular when designing future network protocols which can be hard to test on the real Internet. It is equally or even more important to understand its dynamics. This can be very helpful for future protocols or new types of applications to make use of its evolving nature.

Study of the dynamics of the Internet topology has been tackled both by analyzing the dynamics of individual routes [100, 108, 74, 73] and from a more global perspective, mainly at the AS- or IP-level [57, 32, 92, 98, 94]. In addition, routing changes that happen at the IP-level topology does not necessarily imply changes at the physical level and *vice-versa*. This work focuses on the IP-level routing topology and asks the question of how it evolves over time. Instead of individual routes, we study a *tree* of IP-level routes from one monitor to a fixed set of destinations in the Internet.

In our previous work reported in Deliverable D3.2, we already analyzed the dynamic of the IP-level routing topology discovered around a single node [85]. Using a traceroute-like measurement tool, we periodically probed the route to several destinations from a single monitor in the Internet. This results in a series of routing trees which represent different *ego-centered* views of the routing topology around the monitor. Analyzing these trees, two dynamic behaviors were apparent. In particular, we observed that we never stop discovering new IP addresses over time. Understanding the observed dynamics without knowing the properties of the real Internet topology is complex. Therefore, we relied on simulations to identify the factors behind these behaviors and to study their influence. We proposed a model whose main goal was explanatory. This model represents the Internet IP-level routing topology as an Erdös-Rényi random graph $G = (V, E)$ where vertices correspond to IP addresses and edges correspond to the IP-level connectivity or links between two IP addresses. Then, it incorporates on G well known apparent dynamic factors: load-balancing and route evolution. Finally, it simulates Internet measurements on G to create a routing tree. This process is repeated many times to create several routing trees that we use to analyze the dynamics. From this work, we learn that it is possible to reproduce on Erdös-Rényi graphs the dynamic behaviors observed on the Internet.

This work goes further and studies the dynamic behaviors by using power-law random graphs to model the routing topology. With Erdös-Rényi random graphs, we made no assumption on the underlying topology. Here, we use a graph with a power-law degree distribution. Indeed, Faloutsos and al. [54] have shown that power-law graphs may be close to the Internet topology in term of their degree sequence, so they may well approximate its structure. We first ask the same questions as in the analysis with Erdös-Rényi graphs: (1) can we reproduce the dynamics behaviors on power-law graphs ? and (2) how does the dynamic behavior depend on various simulation parameters ? Then, we investigate the differences of results that appear for Erdös-Rényi and power-law graphs.

The rest of the section is organized as follows. In Section 4.1.1, we describe two characteristics of the dynamics of the IP-level routing topology around a single monitor. Section 4.1.2 presents the simulation



(a) Number of distinct IP addresses observed since measurement beginning.

(b) Observation number vs. block number.

Figure 4.1: Properties of the observed dynamics.

model. In Section 4.1.3, we analyze the results of our experiments. Finally, Section 4.1.5 discusses some related works, and Section 4.1.6 presents our conclusions.

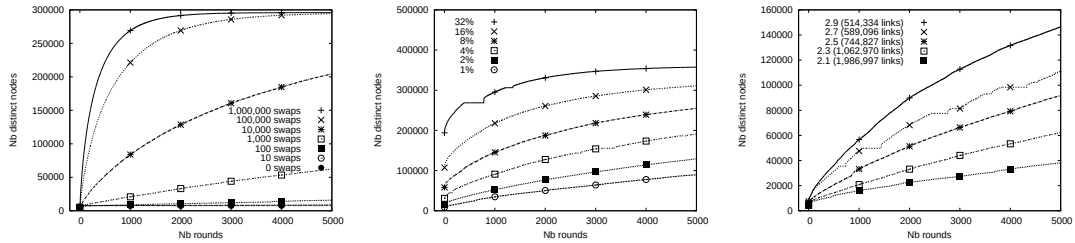
4.1.1 IP-level routing topology dynamics

In our previous work [85], we presented two main characteristics of the dynamics of the IP-level routing topology around a given monitor. To perform this study, we needed several snapshots of the IP routing topology between the monitor and a given set of destinations. We use tracetree [78] which is a traceroute-like measurement tool that aims at discovering a tree of routes or routing paths with the monitor as the root and the destinations as the leaves. The intermediary nodes of the tree are the IP addresses found on the routing path for each pair (monitor, destination). The size of a tree is then the number of all its nodes (intermediary nodes plus the monitor and destination nodes). The link between two nodes represents a hop at the IP level. One routing tree represents a subset of the IP-level routing topology between the monitor and the destinations. It is called an *ego-centered* view of this topology. Repeating many tracetree measurements periodically gives a set of routing trees that one can use to analyze the dynamics. Each measurement round lasts about 4 *min* and the frequency between a pair of rounds is about 15 *min*. Different datasets were collected from many monitors around the world (almost 150 monitors, mostly on PlanetLab) and are publicly available [1].

Analyzing these datasets, two main dynamic characteristics came out: (1) new IP addresses are persistently discovered around the monitor, (2) the pattern of occurrence of IP addresses (number of occurrence/observation, and numbers of blocks of consecutive observation) follows a parabolic shape. Here, we present these characteristics for two of our monitors which are `woolthorpe` and `ovh` [85]. All other monitors exhibit similar results. The collection on `woolthorpe` started in December, 2010 and ended in June, 2011 and $3,000$ destinations were used. The monitor `ovh` only used 500 destinations with a higher measurement frequency. It was collected from October, 2010 to September 2011.

New IP addresses are persistently discovered around the monitor Given a set of routing trees T_1, T_2, ..., T_r, we computed the cumulative union $C_i = \cup T_k, 1 \leq k \leq i$. Fig.4.1(a) plots the size of all sets C_i as a function of time for `woolthorpe` and `ovh`. We observe that new IP addresses are discovered at a fast rate. In other terms, we never stop discovering new IP addresses between the monitor and the destinations over time.

This plot presents the number of distinct IP addresses observed, and not the number of distinct routers, as in general several IP addresses, or interfaces, correspond to a same router. Detecting which interfaces correspond to which routers is a difficult task. Though several methods exist, none is 100% accurate. We used the MIDAR tool developed by CAIDA [27], and studied the number of discovered *routers* observed since measurement beginning. The results were consistent with the plot presented in Figure 4.1(a). Moreover, previous work has studied the number of distinct ASes discovered by such measurements, and showed that it also increases significantly [86]. All in all, there is a good evidence that new routers are actually discovered at a significant rate, even if part of the observed growth may be caused by discovering new interfaces for already observed routers. As there is no method that allows to know with certainty



(a) Impact of swaps ($\alpha = 2.3$, $d = 3,000$) (b) Impact of destinations ($\alpha = 2.3$, $s = 1,000$) (c) Impact of links ($d = 3,000$, $s = 1,000$)

Figure 4.2: Analyzing the impact of simulation parameters on *PL* graphs ($n = 500,000$).

which interfaces correspond to a same router, we limit ourselves to the study of interfaces in the rest of the paper.

The pattern of occurrence of IP addresses follows a parabolic shape We defined two values that quantify the occurrence of IP addresses around a monitor. First, the *observation number* of an IP address represents the total of distinct rounds in which it occurs. Secondly, the *block number* of an IP address is the number of groups of consecutive rounds in which it is observed. As an example, an IP address which was observed on rounds $1,2,3,5,6,8,9$ and 11 has an observation number of 8 and a block number of 4. Fig. 4.1(b) presents the correlation between these two quantities for the monitor `woolthorpe`. The plot exhibits a clear parabolic shape with a large number of points close to the x-axis and to the line $y = x/2$.

This can be explained in the following way. The presence a large number of IP addresses close to the parabola can be explained by load-balancing routers. If a load-balancing router randomly spreads traffic among k paths [1], each router belonging to any of these paths has a probability $p = 1/k$ of being observed at each round, leading to an observation number equal to rp approximately.

A given round is then the first of a consecutive block of observations for one of these routers with the probability p that this router was observed in this round, multiplied by the probability $1-p$ that it was not observed in the previous round. Multiplying this probability by r gives the expected block number, which is then equal to $rp(1-p)$ and is the equation of the parabola. This is a simplification of the real case in which a router may belong to paths used by several load balancers, themselves belonging to paths used by other load balancers.

In practice, an IP address belonging to load-balanced paths can have any probability p, $0 < p < 1$, of being observed. The set of IP addresses closed to the x-axis are often observed on consecutive rounds. Finally, points on the line $y = x/2$ correspond to IP addresses that are observed only during a finite part of the measurement and have a probability of $p = 1/2$ of being observed during that time, due to load balancing.

4.1.2 Model

Our purpose here is to propose relevant and simple mechanisms that reproduce the observations made in Section 4.1.1. For that, we use the same simulation model we have already proposed in [85] and presented in Deliverable D3.2. Note that we do not aim at proposing a realistic model, but rather at providing a first and significant step towards understanding the impact of simple mechanisms on the observed dynamics. This model incorporates four ingredients: the routing topology, the routes from the monitor to the destinations in this topology, load balancing, and routing changes. For modeling each ingredient, we try to make the simplest choice possible, our goal being to obtain a baseline model which makes it possible to investigate the role of each component, and to which future and more realistic models should be compared. In that sens, this work follows a *top down* approach, starting from properties observed on real data and proposing a model able to reproduce the. The philosophy behind stands in the fact that if the model succeed in reproducing them, then it naturally captures the processes that play a fondamental role in the observed properties. As we will see, this approach enables to have significant insights on the relation between the different parameters and thus to extrapolate with the measured data.

[1] It has been shown [14] that per-packet or per-flow load-balancing routers spread –r–pobes equally among all paths to the destination, which is roughly equivalent to randomly choosing a path.

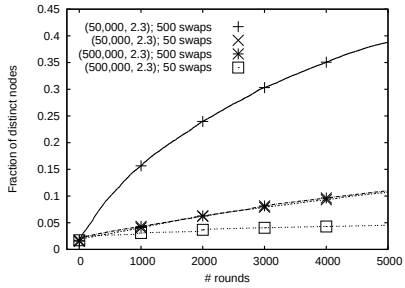


Figure 4.3: Relation between size and swaps ($m/n = 2$, $d = 1\%$).

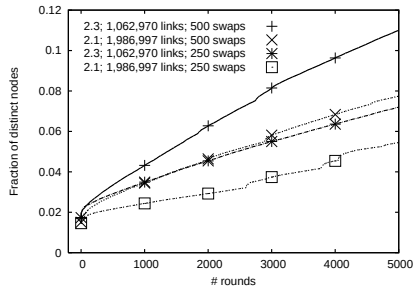


Figure 4.4: Relation between links and swaps ($n = 500,000$, $d = 1\%$).

This work would of course benefit from the opposite *bottom up* approach that would focuse more precisely on the technical mechanisms ruling the situation.

We represent the IP-level routing topology of the Internet by an undirected graph $G = (V,E)$. Each vertex in V corresponds to an IP address and each edge in E corresponds to the connectivity between two IP addresses [2]. Then, we simulate tracetree measurements in G. As a preliminary step, we randomly choose one node as the monitor and d nodes as the destinations. Then from G, we inferred a routing tree T with the monitor as the root, and the destinations as leaves. In practice, we simply perform a *breadth-first search* (*BFS*) starting from the monitor, and then discard all branches that do not lead to the destinations.

At this point, we have a routing tree of *shortest paths* from the monitor to the destinations. The next step is to repeat this procedure r times to simulate r measurement rounds. We simulate load-balancing by a *random BFS*. We model route evolution using link rewiring, or *swap*. This consists in choosing uniformly at random two links (u,v) and (x,y) [3] and swap their extremities, *i.e.* replace them by (u,y) and (x,v).

Our previous work [85] used the Erdös-Rényi random graph model [51] to produce G. Here, we use a random graph with a power-law degree distribution [23]. For power-law graph generation, we use the following procedure: (1) given an exponent α, randomly generate a list of degrees that respects the following power-law [9] (d is a degree value, $f(d)$ the frequency of d):

$$f(d) = d^{-\alpha}, \alpha > 0, \tag{4.1}$$

(2) for each node, create as many half links as the value of its degree, (3) randomly sort the previous list and, (4) connect consecutive half links to form links.

4.1.3 Simulation results

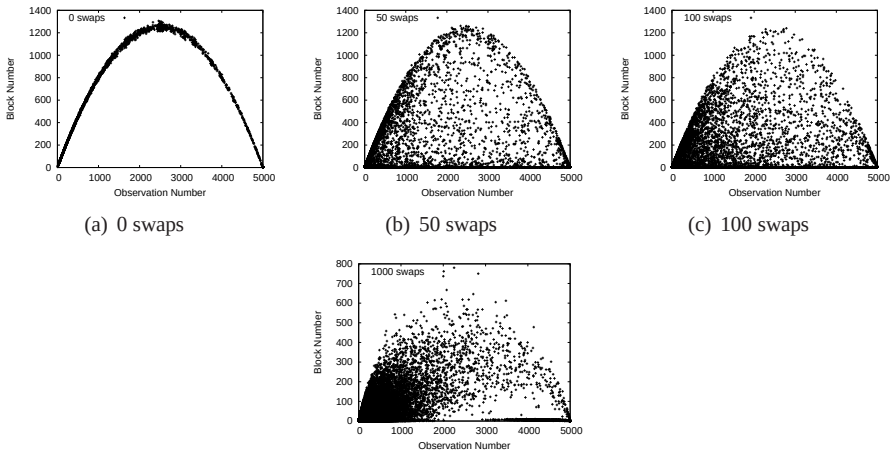
This section investigates whether it is possible to reproduce on power-law (*PL*) graphs the dynamic behaviors observed on the Internet or not. We perform several simulations with different values of the parameters of the model which are: the number n of nodes, the exponent α, the numbers d of destinations, s of swaps per round and m of links for Erdös-Rényi (*ER*) random graphs. We further look for relations between the simulation parameters that may lead to invariants of the dynamic behaviors, and explore the differences in the simulation results with *PL* and *ER* graphs.

4.1.3.1 Reproducing the evolution of IP addresses discovery

Let us first focus on the evolution of the discovery of new IP addresses over time (meaning in the context of this study, new vertex of the graph). As a preliminary step, we vary the number s of edge swaps. Fig. 4.2(a) presents the simulation results on a *PL* graph with $n = 500,000$ and $\alpha = 2.3$, for varying values of s. For this first step, we adopt the same number of destinations as in our measurements ($d = 3,000$).

[2] Note that we do not address here the problem of aliases and we identify a vertex to the set of all IP addresse of a given router. For sake of readability, we will omit this formal precision in the following.

[3] We choose them such that the four nodes are distinct.



(a) 0 swaps (b) 50 swaps (c) 100 swaps

(d) $1,000$ swaps

Figure 4.5: Observation numbers vs. block numbers for various values of s ($n = 500,000$, $m = 2.3$, $d = 3,000$).

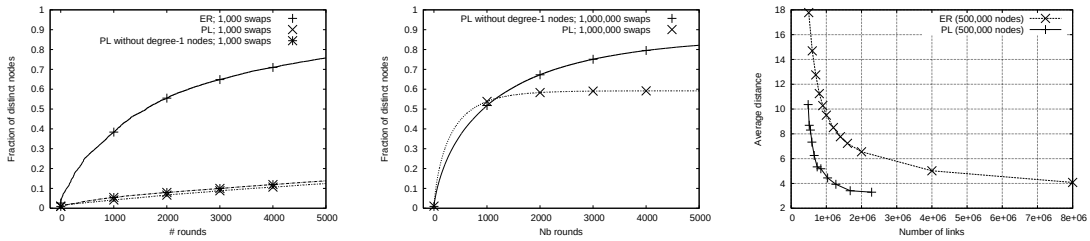
Three main observations appear from Fig. 4.2(a). First, it is possible to reproduce on *PL* graphs the fast discovery of IP addresses observed on our tracetree data, in particular for $s = 1,000$ or $s = 10,000$ swaps. Second, the larger the number of swaps, the faster new nodes are discovered. In fact, routing paths change more quickly with a larger number of swaps (*e.g.*, for $s = 100,000$), than with a lower number of swaps (*e.g.*, for $s = 10$). Recall that we use edge swaps to simulate link changes due to route evolution. Therefore, increasing the number of swaps also naturally increases the probability for routes between the source and the destinations to change, which leads to the fast discovery of new nodes. Third, the first point of all curves are very close. This means that the number of swaps have no influence on the size of routing trees, which was expected.

For $s = 0$ swaps, the curve has a fast initial growth, and then it remains flat until the last round. In the absence of swaps, the only dynamic observed comes from *load balancing* and not from route evolution. All nodes that belong to load-balancing routing paths are quickly discovered at the beginning.

We never succeed in discovering all nodes for *PL* graphs, even when we swap almost all their links at each round. For instance, using $s = 1,000,000$ swaps leads to the quick revealing of only 60% of the nodes in the graph in less than $1,000$ rounds. In the next section, we explore in depth the reasons behind this.

We also test the impact of the number of destinations on the dynamic behaviors. Fig. 4.2(b) shows the results on a graph with $n = 500,000$, $\alpha = 2.3$ and $s = 1,000$ swaps. We observe that the number of destinations clearly has an influence on the height of the first point of the curves, which represents the size of the first routing tree. The larger the number of destinations, the higher the first point of the curves, *i.e.* the larger routing tree. Also, for different numbers of destinations, the slope of all curves are very similar, but not exactly identical. For instance, one may assume that doubling the number of destinations (*e.g.*, from 1% to 2%), we also double the size of the resulting routing trees. However, for this to be true, two conditions need to be verified: (1) all destinations should be on strictly different routing paths from the monitor, (2) and all destinations should be chosen at the same distance from the monitor. This is not the case in our experiments.

Finally, we vary the exponent α of *PL* random graphs. Fig. 4.2(c) presents the simulation results on a graph with $n = 500,000$, $s = 1,000$ and $d = 3,000$, for various values of the parameter α. Note that, the value of the exponent α determines the number m of links for *PL* graphs. From Fig. 4.2(c), it appears that the lower the value of α (or the higher the number m of links), the slower the rate of discovering new nodes over time. Indeed, the proportion of links affected by swaps are negligible on graphs with a high number of links. In addition, distances between pairs of nodes are highly reduced on graphs with a high number of links. Indeed, the closer the destinations are to the source, the shortest are the paths between the source and the destinations. Therefore, less new nodes will be discovered over time.



(a) Evolution of the number of distinct IP addresses observed

(b) Comparison between a reduced *PL* graph where all degree-1 nodes are removed and its original

(c) Number of links vs. the average distance

Figure 4.6: Comparison between ER ($m = 1,000,000$) and PL ($\alpha = 2.3$) graphs of the same size ($n = 500,000$)

4.1.3.2 Finding relations between simulation parameters

To analyze the interaction between the simulation parameters, we vary several of them at the same time. The goal here is to find invariants which can be very helpful to understand our model in depth.

The first relation we explore is between the size of the graph and the number of swaps. We set $\alpha = 2.3$ and $d = \frac{n}{100}$. The value of α are chosen so that the ratio $\frac{m}{n} = 2$ is verified. Fig. 4.3 presents the simulation results on two graphs of different sizes $n = 50,000$ and $500,000$, for different values of swaps $s = 500$ and 50. The y-axis on Fig. 4.3 represents the fraction of discovered nodes over the total of nodes, and the x-axis, the number of rounds. We observe that the two middle curves are very close and almost follow the same slope. It appears that these curves correspond to graphs with a similar ratio $\frac{s}{m}$ of the number of swaps over the number of links.

The second relation concerns the number of links and the number of swaps. Here, we fix n, and set the proportion of destinations to 1%. We vary α and s. Fig. 4.4 presents the results of the simulation on graphs of $n = 500,000$ nodes, with $\alpha = 2.1$ and 2.3, for $s = 500$ and 250 swaps. It seems that when the number of links doubles, the number of swaps also needs to be doubled as well in order to obtain curves with similar slopes. This result also tends to confirm our previous observation that simulations with the same ratio $\frac{s}{m}$ may follow a similar slope for node discovery. Sometimes, some abrupt increases may deviate a curve from its initial growth rate (*e.g.*, the case $\alpha = 2.1$ and $s = 500$ for $d = 1\%$). We find that these events are caused by swaps that happen close to the monitor and therefore may affect a high number of paths to destinations. However, these events usually do not change the slope of curves.

These results are interesting because they imply that knowing the trend of the evolution curve for a given graph and for a given value of swaps, it can be possible to infer the slope for a range of other graphs. During our experiments, we have tested the previous two relations for other sizes of graphs and obtained the same conclusions. These analyses are at a very preliminary stage. We visually observe the similarities between different curves for various parameters of *PL* graphs. Later, we may need some statistical analysis to confirm our conclusions.

4.1.3.3 Reproducing the parabolic shape on the occurrence of IP addresses

We now turn to the correlation between the observation numbers and the block numbers. Fig. 4.5 illustrates our results on a *PL* graph with $n = 500,000$, $\alpha = 2.3$ and $d = 3,000$, for various values of s. We are clearly able to reproduce the parabola observed on our tracetree data (for instance, for $s = 50$ and 100). In that particular case, we also observe that a large number of points are close to the x-axis. For $s = 0$ swaps, all points strictly appear on the parabola. We already know that without swaps, the only dynamic factor in our model is load balancing. This means that nodes are revealed by load-balancing related dynamics. Increasing the number of swaps, the parabola tends to vanish. For instance, for $s = 1,000$ the parabola has already started to vanish. For higher values of s, it completely loses its shape. This can be explained by the fact that with a higher number of swaps, the effects of load balancing becomes negligible. In practice, we find that if we increase the exponent of a *PL* graph, which also increases its number of links, we maintain the parabola if we increase the number of swaps as well.

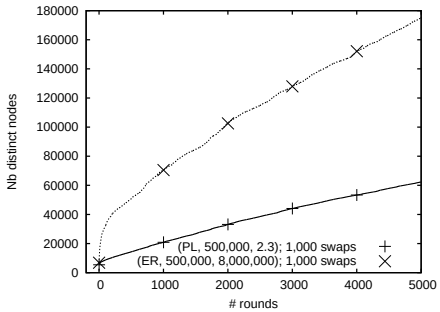


Figure 4.7: Comparison between ER and PL graphs with the same average distance ($n = 500,000$, $d = 3,000$).

4.1.4 Exploring the differences between PL and ER graphs

The evolution of node discovery on PL and ER graphs are very different. Fig. 4.6(a) shows that the majority of nodes of an ER graph, with $n = 500,000$, $m = 1,000,000$ and $d = 3,000$, are discovered within $r = 5,000$ rounds with only $s = 1,000$ swaps. Using the same amount of swaps on a PL graph with approximately the same size and the same number of destinations, we end up discovering only 12% of all its nodes. Clearly, nodes are discovered more slowly over time on PL graphs than on ER graphs. In this last set of experiments, we investigate the reasons that explain this difference.

Our first intuition concerns nodes of degree one. They represent the largest fraction of nodes on PL graphs and, unless they are chosen as destinations, it is impossible for them to be discovered during simulations since they are not router nodes. From Fig. 4.2(a), we have seen that even when we swap almost all links on a PL graph with $n = 500,000$ and $\alpha = 2.3$ at each round, we never succeed in discovering all its nodes. Indeed, with $s = 1,000,000$ swaps the curve tends to flatten out close to the value $y = 295,877$. Examining the remaining nodes, we find that 99.9% of them represent the degree-1 nodes. At the end, we almost discover no nodes of degree 1 on PL graphs.

We now ask the question whether a reduced PL graph in which we have removed every nodes of degree 1 will follow the same evolution of node discovery as an ER graph with the same dimensions; if this is true, then degree-1 nodes may be the only reason of the difference of results observed for PL and ER graphs. Fig. 4.6(a) shows that this is not the case. The reduced PL graph has only $n = 293,328$ nodes and $m = 841,326$ links. Therefore, we plot its curves with $d = 1,760$ to maintain the same ratio of the number of destinations over the total of nodes as in the original PL graph. With $s = 1,000$ swaps, the curve of the reduced PL graph grows more slowly than the curve of the ER graph, but similarly as the one of the original PL graph. This means that nodes of degree 1 are not the reasons behind the slow evolution. Increasing the number of swaps until we reach the deprecated case where a maximum is reached for the original PL graph in Fig. 4.6(b), we end up discovering the majority of nodes of the reduced PL graph. This confirms the fact that the flat phase on PL graphs are due to the non-discovery of degree-1 nodes.

Our second intuition concerns the difference in the average distance that exists between PL and ER graphs. It has been proven that the average distance is in the order of $log\ log\ n$ on PL graphs [33], while it is of $log\ n$ on ER graphs [26]. We explore this result in Fig. 4.6(c) which plots the average distance as a function of the number of links for both PL and ER graphs with $n = 500,000$ nodes. We use the approximation proposed in [77] to compute the average distance. It appears that average distances are effectively much smaller in PL than in ER graphs. This implies that the destination nodes on PL graphs are closer to the monitor; therefore, the resulting routing trees on PL graphs will have fewer nodes. To confirm this result, we examine the size of the trees produced on ER and PL graphs with $n = 500,000$ and the same number of links. We uses $d = 3,000$ destinations and $s = 0$ swaps. We find that the average size on $r = 5,000$ trees is $5,363$ and $12,868$ for PL and ER, respectively. We then study in Fig. 4.7 the evolution of node discovery on PL and ER graphs with the same average distance. We find that they still do not follow the same slope. Finally, this shows that apart from the degree-1 nodes and the average distance, there are still other factors behind the difference between PL and ER graphs.

4.1.5 Related work

Study of the dynamics of the Internet topology has been tackled both by analyzing the dynamics of individual routes [100, 108, 74, 73, 118] and from a more global perspective, mainly at the AS- or IP-level [57, 32, 83, 92, 98, 94, 63, 110, 47, 46]. Load balancing has also been acknowledged for playing an important role in the dynamics of routes as measured with traceroute-like tools [14]. Cunha et *al.* [43] used a method for measuring load-balanced routes, i.e. routes containing one or more load-balancing routers, and study their dynamics.

Some works [64, 98] argue that the topology dynamics should be taken into account in order to produce realistic models for the Internet topology. Work on modeling this topology and its dynamics can be roughly divided between approaches aiming at realistically mimicking the evolution mechanisms of the topology, e.g. reproducing the criteria taken into account by ASes for creating peering or customer-provider links, see for instance [52, 31, 117, 99, 84], and approaches aiming at reproducing global network characteristics through simple mechanisms, thus exhibiting simple causes for more complex observations. This paper belongs to this second approach. Tangmunarunkit *et al.* [114] showed that this approach is relevant by establishing that network generators based on local properties, such as the degree distributions of nodes, can capture global properties of the topology, such as its hierarchical structure. Most related to our characterization and modeling of the evolution of the Internet topology is the work by Oliveira et *al.* [92], which analyzes the AS topology and shows that real topology dynamics can be modeled as constant-rate births and deaths of links and nodes. In a similar spirit, Valler *et al.* model BGP routing churn by a process similar to an epidemic spreading on a network [115]. Park *et al.* [99] studied several growing models for the Internet topology, i.e. models in which nodes and links are progressively added over time. They compared the evolution of these induced networks with the evolution of the real topology, and use this to distinguish between the quality of the different models.

Whereas most existing works focus on the long-term evolution (e.g. from the Internet birth to current times) of the *physical* AS topology, we are concerned here with the short- to medium-term evolution of the *routing* topology at the IP-level. The routing and physical topology are closely linked but not identical objects. In particular, routing changes can occur in the absence of physical changes. They are closely linked to BGP dynamics which have been studied for instance in [74, 83, 115, 118]. Finally, our model does not take into account node appearance and disappearance, which would be necessary for modeling the long-term topology evolution.

4.1.6 Results summary and Next steps

This work focuses of the dynamics behaviors observed at the Internet IP-level routing topology. We use an existing model that incorporates a routing topology, its dynamics and traceroute-like measurements to explain the observed dynamics. Our former work detailed in Deliverable D3.2 represented the routing topology by an Erdös-Rényi random graph. Here, we use a power-law random graph and investigate the effect of its degree distribution on the dynamics. As in Erdös-Rényi graphs, we are able on power-law graphs to reproduce the dynamic behaviors observed on the Internet. However, we find that the results between the two types of graphs are quantitatively different.

Two main reasons for this difference appear: (1) it is difficult to discovered the degree-1 nodes, which represent the largest fraction of nodes on power-law graphs, (2) the average distance on power-law graphs are much smaller than for Erdös-Rényi graphs. Therefore, traceroute-like measurements on power-law graphs produce smaller routing trees, which leads to a slower discovery of new nodes over time.

Future work lies in two main directions. First, we strongly believe that this model can be used to estimate some properties of the actual IP-level routing topology that are not directly available through measurements. For instance, performing more extensive studies of the relations between the model's parameters and the observed behavior, would allow to infer the parameters from the observed behavior. Applying this knowledge to real-world data would allow to estimate the real-world values corresponding to these parameters, such as for instance the frequency of link changes in the whole topology. Moreover, since our model is based on random graphs and simple mechanics for load balancing and routing dynamics, it lends itself well to formal analysis. This would allow to obtain formal proofs for such results.

Second, the field of Internet topology modeling is very active, and models far more realistic than random graphs are available. One should explore the combination of our routing mechanisms principles with these topology models, to investigate the role played by the topology structure on the observed

dynamics. In particular, our model does not take into account the long term topology evolution, since it does not model node birth or death. Coupling the ingredients of our routing dynamics with, e.g., a growing model for the Internet topology which would reflect its long term dynamics would surely lead to insightful results.

4.2 Routing stability analysis

4.2.1 Introduction

In Deliverable D3.2, we investigated the problem of path-vector routing stability by means of:

1. the development of a method to process and interpret the data part of Border Gateway Protocol (BGP) routing information bases in order to identify and characterize occurrences of BGP routing system instability; such characterization can be used as a comparison point for the stability of the newly developed schemes (candidate to replace BGP) and characterize instability phenomena any routing system would have to cope with;

2. the definition of a set of stability metrics and develop methods for using them in order to provide a better understanding of the BGP routing system's stability;

3. the investigation how path-vector routing behavior and network dynamics mutually influence each other.

The experimental results show that the proposed method enables detecting instability events affecting routing tables, and deriving the local impact on the stability of the routing system (local stability). We have also determined a differential stability-based decision criterion that can be taken into account as part of the BGP route selection process. A significant fraction of the routes (90%) of the routes selected by means of this process is not stretch increasing. Moreover, if one would admit an increase of one AS-hop, only a minor fraction of the routes (about 2%) would be penalized by a higher stretch increase (two AS-hops and above).

The complete this stability analysis, we investigate the correlation between forwarding path and BGP routing path instability. The objective is to relate the measurements carried in subtask 2 on ego-centered visions of the topology (along its forwarding paths) and subtask 3 dedicated to the stability of routing paths (as locally determined by individual routers).

The measurements carried in subtask 3 are based on processing BGP update messages collected by the RouteViews project (www.routeviews.org), and processing them by applying the algorithm described in [40] [96]. The proposed approach to relate such measurements with the ones obtained in Subtask 2 relies on the observation that RouteViews monitors a rather large set of Internet eXchange (IX) points [4]. The data obtained from these routers part comprise the full list of Routing Information Base (RIB) entries (updated each two hours) and the received updates from the peer ASs separated in files every 15 minutes. The format used of this files is MRT [25].

The measurements carried by RADAR are traceroute-like probes running from a set of monitoring nodes. Such probes target a large set of prefixes and end-hosts distributed in the Internet. Based on these measures, radar builds ego-centered views of the forwarding topology. A subset of the forwarding paths followed by the radar probes will, expectedly, use one of the IX points monitored by RouteViews and consequently, a subset of the AS-path monitored by the tool developed in subtask 3 are also monitored by radar. Based on that both tools can complement each other.

It is worth noticing that the tool developed by subtask 3 cannot process the BGP data collected by all the RouteViews monitoring points because of the very high volume of information (remember that the tool developed in the context of subtask 3 associates processing of BGP routes to a single BGP speaker). The selected router to be monitored by subtask 3 is route-views.wide.routeviews.org.

[4] a complete list can be found in http://www.routeviews.org/peers/

Destination prefix	Forwarding path	Routing path
71.240.192.0/19	2500 701 19262	2500 701 19262
71.126.64.0/18	2500 701 19262	2500 701 19262
193.138.220.0/22	2500 2914 174 16243 174 34305	2500 174 34305
75.89.192.0/21	2500 2914 3356 7029	2500 7018 7029
211.60.0.0/16	2500 2516 3786	2500 2516 3786
124.139.60.0/22	2500 2516 3786 38409	2500 10026 18302 38409

Table 4.1: Example of different paths following by RADAR and routeviews.

4.2.2 Alignment of the data

The first step to perform the correlation between forwarding path and routing path stability is to align the data obtained in subtask 2 and subtask 3. For this purpose, the following points have been addressed.

On the one hand, the sampling intervals of the measurements obtained through Radar and the MRAI time interval as used by the tool to process the BGP routes are different (the latter is shorter than the former). In Radar, each round of measurement takes approximately 4 minutes and 10 minutes elapses between the end of a given round and the beginning of the next one. We thus run two different tests with the routes obtained from the routeviews data. The first run uses the real BGP update time interval and the routing path stability is computed according to real period between updates as dictated by the MRAI (without having the forwarding path stability at this granularity but assuming this value remains constant over that period). The second run only tests the stability of a routing path at each RADAR iteration (around each 10 minutes).

On the other hand, the data provided by RADAR are IP addresses while the stability analysis is performed in terms of AS path. Finding association (or matching) between IP forwarding paths to AS routing paths is thus required. This matching can been obtained executing the Whois tool. Whois is a query and response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address prefix, or an autonomous system. In particular, we used the Whois-based web tool made available by the Team Cymru (http://www.team-cymru.org/). This tool takes as input a file containing quite large numbers of IP addresses and translates them into AS numbers as output.

4.2.3 Matching IP address to AS Number

The matching between IP and AS Number (ASN) presents several shortcomings for the analysis of the forwarding path stability. These are described in the following paragraphs.

4.2.3.1 IP address prefix owner vs. IP address allocation

The mapping provided by Whois is not able to discriminate between the current owner of an IP prefix and the actual user of an IP subnet part of this prefix (thus, in particular, the IP prefix itself). Indeed, a customer AS may use IP addresses that actually belong to the (transit) provider AS to which the customer AS is attached to. In the mapping, these addresses appear as belonging to the transit provider AS instead of designating network attachments located at the customer AS (or eventually to the customer AS router connected to the transit provider AS). This explains much of the differences observed in AS sequences between the routing path and the forwarding path. An example is shown in Table 4.1.

4.2.3.2 Concatenation of non-routable IP addresses and filtered IP addresses

In some cases, the forwarding path trace as provided by RADAR may consist of a sequence of non-routable IP addresses such as 10/8 (commonly used to number internal interfaces of edge routers) like in the following example:

- Source interface: 203.178.141.138

- Intermediate interfaces: 203.178.135.1, 203.178.138.243, 61.213.145.93, 61.213.162.87, 61.213.162.97, 129.250.2.141, 129.250.4.118, 129.250.4.231, 62.156.138.197, 62.154.5.5, 193.159.224.118, **10.232.3.66, 192.168.196.17, 10.233.16.4, 10.232.1.2, 10.232.0.9, 10.233.66.130, 10.233.66.135**
- Destination interface: 84.241.194.22
- Destination prefix: 84.241.192.0/18

According to the IP to ASN mapping, the IP address 193.159.224.118 belongs to the AS 3320 while 84.241.194.22 to AS 31615. The Whois tool provides the information that the AS 3320 is peering AS 31615, thus we can derive that the sequence of non-routable IP addresses belongs to a single AS (either the AS 3320 or the AS 31615). In order thus to map the AS path routing sequence to the forwarding path requires thus to capture such sequence.

In other cases, the filtering procedure applied by some routers may hide some IP addresses along the forwarding path. The resulting information is incomplete.

These two cases are the most common. For example, in one RADAR iteration, the paths to reach around 1400 of the overall 3000 destinations contain either non-routable IP addresses or the IP addresses are filtered and thus unknown.

4.2.3.3 Paths with AS loop

It may happen that the resulting AS path presents a loop. An example is shown below. The forwarding path provided by RADAR consists of the following IP address sequence:

- Source interface: 203.178.141.138
- Intermediate interfaces: 203.178.135.1, 203.178.138.243, 61.213.145.93, 61.120.144.87, 61.213.162.229, 129.250.2.34, 129.250.8.182, 4.68.111.249, 4.69.132.78, 4.69.134.22, 4.69.132.38, 4.69.132.42, 24.164.96.140, 64.156.66.50, 65.25.128.254, 24.29.164.132, 24.29.161.118
- Destination interface: 71.67.90.93
- Destination prefix: 71.67.0.0/17

Applying the IP address to ASN map shown in 4.2, we obtain the AS path **2500 2914 3356 11060 3356 10796** which contains the loop **3356 11060 3356**.

The observation of AS loop (from the IP address to the AS number mapping) could be due to the following :

- Misconfiguration of a transit router, policy-induced oscillation or the use of redundant protocol like VRRP (the traceroute message passing twice by the AS interconnecting the two routers giving both access to the destination prefix);
- The user of the IP address is not the actual AS owning the corresponding IP address prefix itself as already discussed in Section 4.2.3.1.

4.2.3.4 Paths to the same destination prefix but with different AS sequence

Another problem that makes such analysis difficult is that during the same time interval, forwarding paths directed to IP destination address part of the same destination prefix can be associated to different routing paths directed to that destination prefix. As such it is thus not possible to consider that all forwarding paths directed collectively to a given destination prefix follow the same AS path.

For example, the previous AS sequence is obtained at the same time of the following one

- Source interface: 203.178.141.138
- Intermediate interfaces: 203.178.141.138, 203.178.135.1, 203.178.138.243, 61.213.145.93, 61.120.144.87, 61.213.162.229, 129.250.2.34, 129.250.8.182, 4.68.111.249, 4.69.132.78, 4.69.134.22, 4.69.132.38, 4.69.132.42, 4.69.132.205, 4.78.216.14, 65.25.128.158, 71.67.121.69

Mapping IP	Prefix	AS
203.178.141.138	203.178.128.0/17	2500
203.178.135.1	203.178.128.0/17	2500
203.178.138.243	203.178.128.0/17	2500
61.213.145.93	61.213.144.0/20	2914
61.120.144.87	61.120.144.0/20	2914
61.213.162.229	61.213.160.0/19	2914
129.250.2.34	129.250.0.0/16	2914
129.250.8.182	129.250.0.0/16	2914
4.68.111.249	4.0.0.0/9	3356
4.69.132.78	4.0.0.0/9	3356
4.69.134.22	4.0.0.0/9	3356
4.69.132.38	4.0.0.0/9	3356
4.69.132.42	4.0.0.0/9	3356
24.164.96.140	24.164.96.0/19	11060
64.156.66.50	64.152.0.0/13	3356
65.25.128.254	65.25.128.0/19	10796
24.29.164.132	24.29.160.0/20	10796
24.29.161.118	24.29.160.0/20	10796
71.67.90.93	71.67.0.0/17	10796

Table 4.2: Example of an AS loop in the forwarding path

- Destination interface: 71.67.121.69
- Destination prefix: 71.67.0.0/17

Applying the IP to ASN map shown in 4.3, we obtain the AS sequence of the routing path **2500 2914 3356 10796** which is different than the AS sequence obtained in Section 4.2.3.3.

One possible explanation is the application of load balancing techniques that may split the announcement of a given prefix into two (or more) routes (prefix de-aggregation), each one advertising a sub-prefix towards a different interface whereas closer to the source both are aggregated (prefix aggregation along the common segment).

4.2.4 Data Processing

In order to process the data, we use the following procedure:

- Association between forwarding and routing paths (on per destination basis) does not require full identification before association but only that a given forwarding path can be associated unambiguously with a given routing path
- Based on this pair of data sequence, characterization of instability can be limited to a first level analysis that would consist in determining whether an instability the forwarding and/or the routing path is detected or not.
- These pairs can be then grouped/classified into 4 classes/subsets:
  - RP Stable - FP Stable: does not require any further analysis
  - RP Unstable - FP Stable: translate routing instability without forwarding instability (or forwarding path stability)
  - RP Stable - FP Unstable: translate routing stability without forwarding stability (or forwarding path instability)
  - RP Unstable - FP Unstable: requires identification if a common segment is at the origin of the instability (thus AS-IP address mapping is required to determine whether there is a common origin to instability)

Mapping IP	Prefix	AS
203.178.141.138	203.178.128.0/17	2500
203.178.135.1	203.178.128.0/17	2500
203.178.138.243	203.178.128.0/17	2500
61.213.145.93	61.213.144.0/20	2914
61.120.144.87	61.120.144.0/20	2914
61.213.162.229	61.213.160.0/19	2914
129.250.2.34	129.250.0.0/16	2914
129.250.8.182	129.250.0.0/16	2914
4.68.111.249	4.0.0.0/9	3356
4.69.132.78	4.0.0.0/9	3356
4.69.134.22	4.0.0.0/9	3356
4.69.132.38	4.0.0.0/9	3356
4.69.132.42	4.0.0.0/9	3356
4.69.132.205	4.0.0.0/9	3356
4.78.216.14	4.0.0.0/9	3356
65.25.128.158	65.25.128.0/19	10796
71.67.121.69	71.67.0.0/17	10796

Table 4.3: Example of a forwarding path crossing different ASes to reach the same prefix as the case above

- For each of the four classes, we record the min, the max stability metric value in addition to the computation of the average and variance of the stability metric (over all pairs associated to each class)

Cases 2) and 3) are the interesting cases because they translate routing instability without forwarding instability and vice-versa, characterizing (in particular, identifying) the origin for these subsets provides a location analysis which can be performed in a second step of analysis. Hence, the identification problem is to be considered as part of the analysis/step and is limited to a specific subsets (at least for a first level analysis). Moreover, this approach (by association) is much simpler compared to the approach that requires full mapping of data sequences (IP addresses (forwarding) and AS (routing paths)) before analysis.

Over the whole duration of the measurement and processing period, it has become clear that it was not possible to classify all routing-forwarding path by means of a single discriminant (as some of these pairs can exhibit multiple patterns during the measurement period). Instead, we count the number of matches labeled as (1 for RP Stable - FP Stable, 2 for RP Unstable - FP stable, 3 for RP Stable - FP Unstable, and 4 for RP Unstable - FP Unstable) for each routing-forwarding path pair. We then derive a dominant behavior/main trend (corresponding to the label with the maximum number of counts) and sub-trend would be more appropriate.

4.2.5 Results and Analysis

The table 4.4 presents a summary of the classification of the routing - forwarding paths pairs using the following classes (unstable,unstable), (unstable,stable), (stable,unstable). The second column indicates the number and the percentage of paths per class for which at least one instability event has been observed. The third colum provides the maximum number of measurement intervals over which the corresponding behaviour has been observed together with the median value.

Label	Number and Percent of Pairs	Max.Count - Median
FP unstable - RP unstable	517 - 54%	40 - 2
FP unstable - RP stable	915 - 96%	223 - 47
FP stable - RP unstable	182 - 19%	117 - 2

Table 4.4: Pair classification

The table 4.5 accounts for the dominant behaviour/trend and the sub-trend. In case of a tie during classification, we counted only 1/2 (two conditions tie) or 1/3 (all conditions tie). The majority of the pairs falls in the (FP unstable,RP stable) class. The latter determines the dominant behaviour, i.e., the most representative behaviour: the instability observed for about 95% of the pairs results from forwarding path instability. The second trend indicates that for about 50% of the pairs the observed instability result from both forwarding and routing path instability.

Main trend	Number of Pairs	Score
FP unstable - RP unstable	36	32
FP unstable - RP stable	912	906
FP stable - RP unstable	15	12
Second trend	Number of Pairs	Score
FP unstable - RP unstable	474	444
FP unstable - RP stable	3	3
FP stable - RP unstable	58	12

Table 4.5: Trend analysis

Figure 4.8 plots the percentage of the pairs labeled (stable,stable) over the entire measurement period in the form of a cumulative distribution function (CDF). The following observations can be drawn from this figure:

- Few pairs (less than 4%) are constantly unstable, meaning that only a small fraction of the observed routing path instability correspond to a forwarding path instability.
- The majority of the pairs (around 60%) are labeled as (stable,stable) more than 90% of time, around 75% show the same behavior more than 80% of the time.

These observations combined with the fact that main cause of instability results from the forwarding plane corroborates the assumption that the dynamic properties of the forwarding and the routing system are different. Henceforth, it is impossible to derive the one from the other. Moreover, it can also be observed that a second order effect correlates forwarding and routing path instability for about 50% of the observed instability.

Figure 4.8: Cumulative Distribution Function (CDF)

Chapter 5

Detecting events in samples and time series

Faced with complex networks that evolve over time, a frequent need is to monitor their evolution and automatically raise alerts on abnormal behaviors of the system, i.e., events which are statistically different from most others. This challenging task is generally called *outlier detection*. In spite of many works addressing this question for decades in various fields, the diversity of cases leading to different outlier definitions makes it hard to create a single universal method. Here we consider the case of a property measured on an evolving network (Figure 5.1). How can we automatically and reliably identify outliers in it? It is challenging because these data contain both regime changes (i.e. sudden changes of the mean of the time series) due to the evolution of the normal behavior, and outlying values that deviate globally or locally from the main trend. Moreover, we have no prior knowledge on the data; events may occur at different time scales; we want an on-line method for real-time analysis. These settings are known to pose a difficult problem. This work introduces a new method to automatically detect outliers in sets of numbers and in time series. We also show its relevance for detecting abnormal events in computer and social networks. The source code is available[2].

5.1 Introduction

Related Work

Given a data set, outlier detection aims at finding data points which are very different from the remainder. This field has received a large attention in the last decades because outliers often represent critical information about an abnormal behavior of the system described by the data. It covers a broad spectrum of applications such as the identification of mechanical faults, changes in system behavior, human and instrument errors, natural deviations in a population, or data cleaning prior to modeling. Outliers are also called: event, novelty, anomaly, noise, deviation or exception [66].

However there is no formal definition of an outlier because this intuitive notion varies with the context and the desired characteristics of outliers. In a statistical perspective, Grubbs [59] defined that “an outlying observation, or outlier, is one that deviates markedly from other members of the sample in which

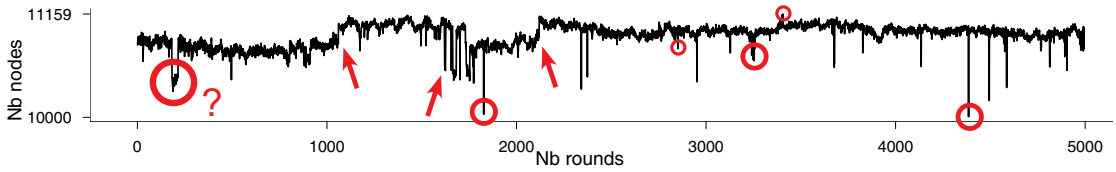


Figure 5.1: Evolution of a property measured on a network during time. Some outliers are circled. Regime changes are pointed by arrows.

it occurs". Hawkins [65] defines an outlier as "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism", while Barnett and Lewis [19] call an outlier "an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data". The diversity of applications has led to the introduction of various techniques for outlier detection [30]. Areas of research such as statistics, data mining, information theory and process control theory have produced various methods for spotting outliers in stochastic processes. Specific researches also address the question of detecting anomalies in the Internet traffic [56].

Existing methods may be divided between *univariate methods* (i.e. considering one variable), proposed in earlier works in statistics, and *multivariate methods* (i.e. considering multiple variables) which form the main part of the current body of research. Although univariate methods have been studied during a long time, and despite recent focus on multivariate methods due to the increase of computational power, univariate methods remain important to study.

We also distinguish parametric and non-parametric (model-free) procedures [22]. Parametric procedures assume the values to be identically and independently distributed following a known probability distribution (generally a normal distribution), or at least a statistical estimate of the distribution parameters to fit the data. They flag as outliers the values that deviate from the model hypotheses. They are often unsuitable for data sets without prior knowledge of the underlying distribution [97] because the hypotheses (e.g. the independence of values) are not satisfied, and because the statistical models are not reliable for real data and are hard to validate since many data sets do not fit one particular model.

Non-parametric procedures do not assume knowledge of the data distribution, and learn to detect outliers. In some cases (*supervised learning*) labeled data sets are available, from which the program builds a model of normal behavior (and sometimes also a model of outlying behavior). Otherwise (*unsupervised learning*), the procedure builds a probabilistic model of the data set, and updates this model as new points appear. These procedures classify as outliers the data points that deviate significantly from the model. These approaches are based on histogram analysis, kernel density, distance measures or clustering analysis.

The output of an outlier detector is a score of "outlierness" assigned to each data point, which represents its probability to be an outlier, or the distance from normal points. Data points are ultimately classified as outliers when their score is above a given threshold which is a parameter of the method.

The detection of outliers in temporal data relies mainly on two approaches. In the first one, points which deviate from a temporal model like the autoregressive integrated moving average (ARMA) model [3] or a finite-state automaton model [71] are marked as outliers. In the second one, points very different from other points within a sliding window are marked as outliers. Regime changes (i.e. change points in time series that are observed by sudden changes of the mean) may be considered as anomalies as well [29, 113].

Finally, recent papers address the issue of outlier detection in networks and graph streams [7] by finding surprising motifs [10][72].

Our Contribution

We propose here a new unsupervised non-parametric univariate method that reliably detects multiple outliers on either static or temporal data sets given the following setting, which is known to be hard: values may not be independent and identically distributed; we have no prior knowledge of the underlying process which generated the data, or of the probability distribution; in time series, regime changes may exist due to the evolution of the normal behavior (non-stationarity), and also outlying values which deviate globally or locally from the main trend. We finally want an on-line method for real-time monitoring. In this context, our method has the following advantages: (a) it uses a novel approach based on the study of the skewness of distributions, and is easy to interpret; (b) it looks for outliers only when the notion of outlier is relevant in the considered data set; (c) it is easy to use, as the only parameter is the size of the time window for time series, and (d) it may be used on-line.

We describe our method in Section 5.2, validate it in Section 5.3, and apply it on real-world data in Section 5.4; we conclude in Section 5.5.

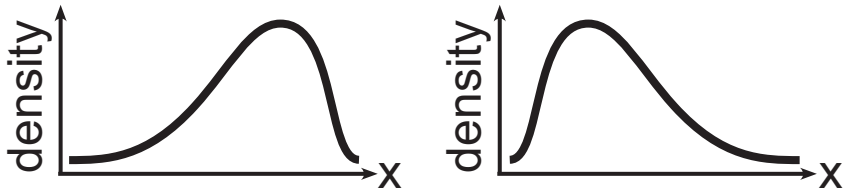


Figure 5.2: Example of negative (left) and positive (right) skewed distributions.

5.2 The Outskewer Method

Our method relies on the notions of *skewness* of distributions and its evolution when extremal values are removed, which we call *skewness signature*; we use this to detect outliers in multisets of numbers and in time series.

5.2.1 Skewness

We consider a multiset (i.e. a set in which members may appear more than once) X of n values. The distribution of these values is the fraction P_x, for each x, of values in X which are equal to x. Such distribution samples are basically described by their mean $\bar{x} = \sum_{x \in X}(x/n)$ and standard deviation $\sigma = \sqrt{1/(n-1) \cdot \sum_{x \in X}(x-\bar{x})^2}$. Going further, the sample skewness is a measure of distribution asymmetry, and can be estimated by:

$$\gamma(X) = \frac{n}{(n-1)(n-2)} \sum_{x \in X} \left(\frac{x-\bar{x}}{\sigma} \right)^3 .$$

Intuitively a negative skewness indicates a tail on the left of the distribution more pronounced than the one on the right, while a positive skewness means the converse, see Figure 5.2. If no tail exists, i.e. all values are equal, $\gamma(X)$ is undefined because $\sigma = 0$. If both tails exist on each side and are equal, $\gamma(X) = 0$. For normal distributions ($P_x = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$), $\gamma(X) = 0$, while for Pareto distributions ($P_x = \frac{ab^a}{x^a+1}$ where $0 < b \leq a$), $\gamma(X) > 0$. Examples of unimodal skewed distributions are shown on Figure 5.2.

The skewness has the interesting feature to be influenced by values which are far from other values, because it is based on the cubed distance from values to the mean. Hence its value changes a lot if they are removed. We show now how to use this feature for outlier detection.

5.2.2 Skewness Signature

We consider the evolution of the skewness of a distribution of values in a multiset X while extremal values are removed one by one from X, which we call the *skewness signature* of X. The extremal value of X, denoted by $e(X)$, is:

$$\mathrm{e}(X) = \begin{cases} \max(X) & \text{if } \gamma(X) > 0, \text{and} \\ \min(X) & \text{otherwise.} \end{cases}$$

In practice, the skewness is almost never equal to zero, hence always choosing $\min(X)$ in the case where $\gamma(X) = 0$ induces a negligible bias.

We define a series of multisets as follows: $X_0 = X$, $X_i = X_{i-1} \backslash \{\mathrm{e}(X_{i-1})\}$, for all $i > 0$. In other words, X_i is the multiset obtained by removing one occurrence of the largest (resp. smallest) value of X_{i-1} if the distribution of values in X_{i-1} has a positive (resp. negative or zero) skewness. Finally, we define the skewness signature as the function $s(p, X) = \gamma(X_{\lfloor p \cdot n \rfloor})$, where n is the size of X and $X_{\lfloor p \cdot n \rfloor}$ is the multiset obtained from X by removing $\lfloor p \cdot n \rfloor$ extremal values, i.e. a fraction p of extremal values.

For example, if $X = \{$-3, -2, -1, -1, 0, 1, 2, 3, 7$\}$, values 7, 3, 2, -3, 1, -2, 0 are removed in this order[1], and the values of the skewness signature are 1.09, 0.22, 0.17, 0, 0.40, 0, 1.73.

The skewness signature may be used to find outliers in unimodal distributions because outliers lie at their extremities, and because skewness is sensitive to the removal of outliers.

[1] Values are removed until $\gamma(X)$ is not computable: our skewness estimator is only relevant for data sets with at least 3 values.

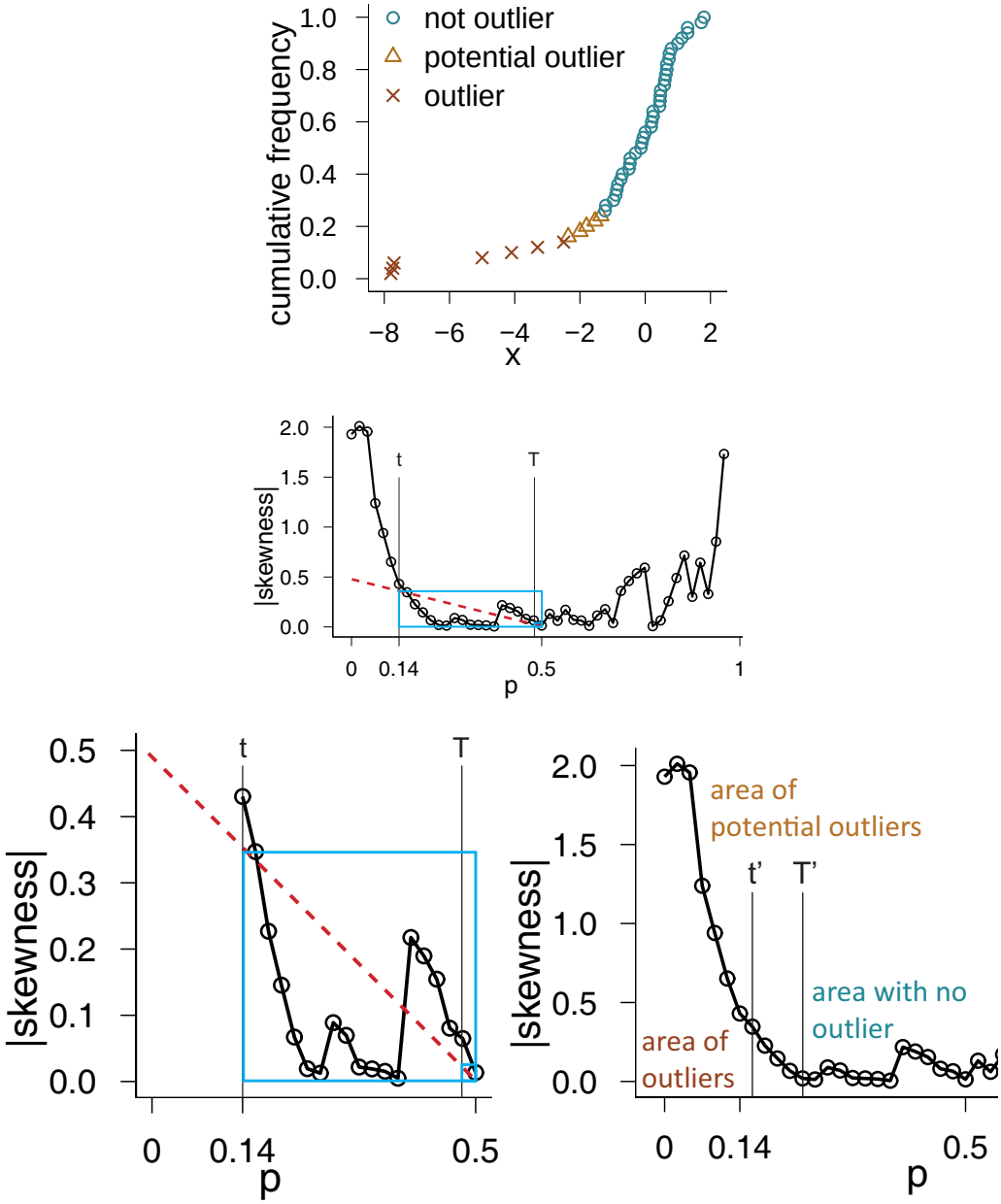


Figure 5.3: Example of 50 values with 7 outliers and 5 potential outliers (from left to right): cumulative distribution (top); absolute values of the skewness signature (middle); zoom on it (bottom left); absolute values of skewness for which outliers and potential outliers are detected (bottom right). We obtain $t = 0.14$, $T = 0.48$, $T' = 0.16$, $t' = 0.24$.

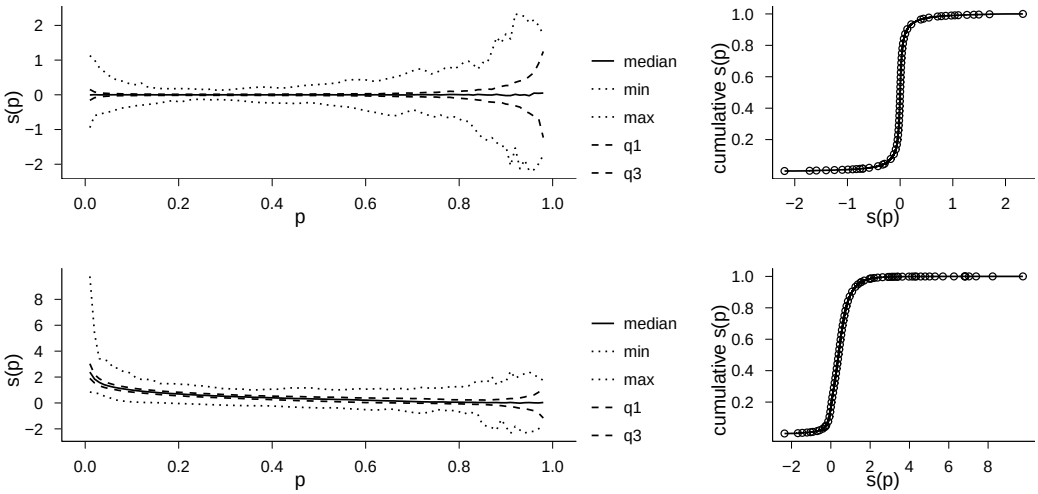


Figure 5.4: Quartiles, min and max of $s(p,X)$ on 1,000 normal (top) and Pareto (bottom) samples with the cumulative frequency distributions of $s(p,X)$.

5.2.3 Outliers Detection

Our method relies on the following hypotheses: outliers are extremal values which cause the skewness to be far from zero; the skewness signature converges to zero (i.e. the distribution becomes more symmetric) when outliers are removed one by one. Therefore, the distance of the skewness to zero can be used to identify outliers. Extremal values which cause this distance to be too large should be classified as outliers. But how is it possible to determine that the distance is too large without making any hypothesis on the data set?

We propose to consider the distance relatively to the proportion of extremal values removed: the more extremal values removed, the closer to zero the skewness is expected to be. For any $p \in [0;0.5]$ we say that s is p-**stable** if and only if $|s(p',X)| \leq 0.5-p$, for all $p' \in [p,0.5]$. We do not consider values of p larger than 0.5 because this corresponds to a removal of more than half of all values; in such situations, the skewness has little to do with the original data, and it may vary much if too many values are removed.

Let t be the smallest value such that s is t-stable, and T be the largest value such that s is T-stable. When s is never p-stable for any p, t and T do not exist. This case indicates that it is irrelevant to look for outliers in the given data set, according to our notion of outlier; in this case our method classifies all values in the data set as **unknown**. Otherwise we find outliers as follows.

We denote the smallest and largest numbers in X_i by $min_i = \min(X_i)$ and $max_i = \max(X_i)$. Then, $min_{\lfloor p \cdot n \rfloor}$ (resp. $max_{\lfloor p \cdot n \rfloor}$) is the smallest (resp. largest) remaining value when a fraction p of all values has been removed. Let t' (resp. T') be the smallest value of p such that $|\gamma(X_{\lfloor t' \cdot n \rfloor})| \leq 0.5-t$ (resp. $|\gamma(X_{\lfloor T' \cdot n \rfloor})| \leq 0.5-T$). Our method concludes as follows: below $min_{\lfloor t' \cdot n \rfloor}$ and above $max_{\lfloor t' \cdot n \rfloor}$, values are **outliers**; between $min_{\lfloor t' \cdot n \rfloor}$ and $min_{\lfloor T' \cdot n \rfloor}$ included (resp. $max_{\lfloor t' \cdot n \rfloor}$ and $max_{\lfloor T' \cdot n \rfloor}$), values are **potential outliers**; values are **not outliers** otherwise. Notice that when $t' = T'$, $min_{\lfloor t' \cdot n \rfloor} = min_{\lfloor T' \cdot n \rfloor}$ (resp. $max_{\lfloor t' \cdot n \rfloor} = max_{\lfloor T' \cdot n \rfloor}$). In this case, values equal to $min_{\lfloor t' \cdot n \rfloor}$ (resp. $max_{\lfloor t' \cdot n \rfloor}$) are potential outliers. Figure 5.3 illustrates our method on an example.

5.2.4 Dynamic Extension

Our method may be used on time series representing the evolution of a system's property. Let $\{x_0, x_1, ..., x_n\}$ be a time series. We consider the multisets which contain w values: $X^i = \{x_{i-w+1}, ..., x_i\}$. Any value x_i of the series belongs to $X^i, X^{i+1}, ..., X^{i+w-1}$. We use our method on all these w multisets, and consider the final class of x_i to be the one which occurs most often among these w classifications. In case of equality, we give priority of *outlier* upon *potential outlier* upon *not outlier*, because we prefer to detect too much outliers than too few.

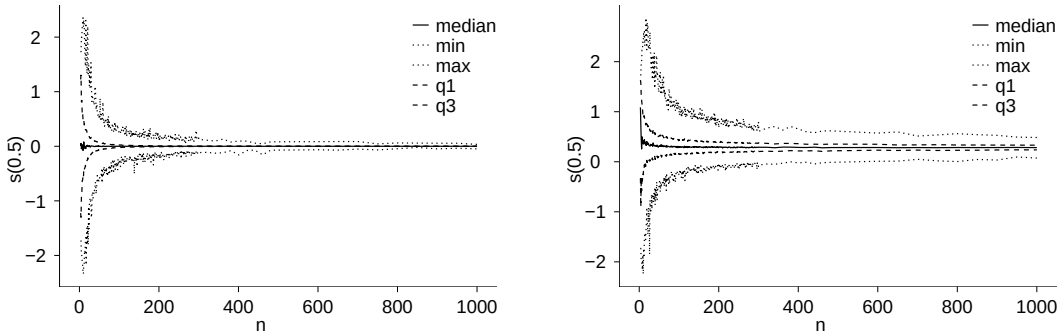


Figure 5.5: Quartiles, min and max of $s(0.5, X)$ as a function of n for normal (left) and Pareto (right) distributions.

5.3 Experimental Validation

The validation of outlier detection methods is difficult because of the various outlier definitions, hypotheses and use cases [22]. Labeled data sets raise also the issue of prior criteria to label the data. We consider that our method should detect outliers, if any, if the notion of outlier is relevant for the given data set. In particular, we consider for our experimental validation the following cases: (a) distributions like Power laws (e.g. Pareto and Zipf's law) commonly contain extremal values far from the mean (i.e. *heterogeneous*), so it is erroneous to consider them as outliers, moreover Power law distributions are asymmetrical so our method should conclude that looking for outliers in them is irrelevant; (b) normal distributions are symmetrical and extremal values far from the mean are uncommon (i.e. *homogeneous*), so no outlier should be detected but these extremal values when they occur; (c) half-normal distributions ($P_x = \frac{2a}{\pi} e^{-x^2 a/\pi}$, where $a > 0$), which are basically the absolute of normal distributions with mean equal to 0, are asymmetrical but homogeneous, so this case is ambiguous and should be unclear for our method as well; (d) symmetric Pareto distributions ($P_x = \frac{ab^a}{2} |x|^{-1-a} 1_{|x|>b}$, where $0 < a < 2$ and $b > 0$), which are basically the mirror symmetric of Pareto distributions about the vertical axis, are symmetrical but heterogeneous, so we study the behavior of our method in this case.

We first study the relevance of our method on these four distributions, and we study the effect of the sample size (III.A). Then we study the performance of our method to detect outliers, and evaluate the rate of true outliers and false outliers detected (III.B). We finally study the behavior of our method when regime changes occur in temporal data (III.C).

5.3.1 Relevance

Our method is applicable if and only if the given data set is p-stable for at least one value of p between 0 and 0.5. A necessary condition for this is that $|s(0.5, X)| < 0.5$. We show in this section that this is true for normal distributions (even with a few outliers) and false for Pareto distributions, which is the expected behavior: normal distributions are symmetrical and homogeneous and Pareto distributions are asymmetrical and heterogeneous.

We study the behavior of s on normal $\mathcal{N}(0,1)$ and Pareto (shape=6, location=2) probability distributions[2]. For each one, we randomly generate 1,000 samples of 100 numbers to obtain skewness signatures; we compute and plot the skewness signature of each sample in Figure reffig:skewness-signatures. We observe that the values of normal signatures oscillate around zero, whereas the values of Pareto signatures globally decrease and are above zero until $p \approx 0.5$. The cumulative frequency distributions of $s(p, X)$ on Figure 5.4 confirm these observations. We also computed the skewness signatures of normal and Pareto distributions with various parameters, and also various symmetrical distributions[3] which we do not present here due to space constraints. All of them exhibit patterns similar to normal signatures.

It is clear that the probability for Pareto skewness to be within $[-0.5; 0.5]$ increases with p. We estimate $\mathbb{P}(|s(0.5, X)| < 0.5)$ on 1,000 Pareto and 1,000 normal samples. We obtain that this probability

[2]Other parameters lead to similar results.
[3]Cauchy, Laplace, some Gamma and Weibull distributions.

is equal to zero for Pareto samples, and is greater than 0.95 for normal samples. We conclude that our method is able to characterize symmetrical and homogeneous versus asymmetrical and heterogeneous distributions at a confidence level of 0.95. Moreover, the addition of some outliers in these distributions produces almost the same signatures than without outliers, because extremal values are firstly removed. Therefore existing outliers do not notably change the characterization.

Let us study the evolution of $s(0.5, X)$ when the sample size n varies. We generate 1,000 normal and Pareto samples for each value of n between 3 and 1,000, then compute $s(0.5, X)$ for each sample, and we finally obtain the quartiles, min and max of the values of $s(0.5, X)$ at each n. We observe in Figure 5.5 that the results converge to zero for the normal distribution, and to ≈ 0.3 for the Pareto distribution. Thus, increasing n should lead to a better characterization.

We verify this hypothesis by evaluating the rate of samples where s is never p-stable, for 1,000 normal and Pareto samples for each size n. We observe in Figure 5.6 that it seems to follow a fast decrease for normal samples. For $n \geq 37$, less than 5% of normal samples are incorrectly characterized, and less than 5‰ for $n \geq 55$. We also observe that it increases with n for $n > 50$ on Pareto samples. The minimum is 79% at $n = 52$, is around 85% at $n = 100$, around 95% at $n = 240$, and above 99.5% for $n > 500$.

We also evaluate this rate for half-normal and symmetric Pareto samples. We observe in Figure 5.6 that it seems to follow a fast decrease for symmetric Pareto samples, but a slow decrease for half-normal samples. This result is not surprising because the theoretical skewness of half-normal distributions[4] is ≈ 1, and the skewness decreases slowly when extremal values are removed one by one. As expected, our method has unclear results in this case.

We conclude that our methods characterizes samples with size 100 very well, and is excellent on samples of size 1,000. Our method also considers that the symmetric Pareto distribution should contain no outlier.

In addition, we study the skewness range where our method considers s to be p-stable at least once. We vary the shape parameter of a Gamma distribution (thus its skewness) to incrementally generate 1,000 samples of 100 numbers for each skewness value, from Pareto-like samples to normal-like samples, and compute the rate of s that are p-stable at least once for each skewness. We remind that s is p-stable if and only if $|s(p', X)| \leq 0.5 - p$, for all $p' \in [p; 0.5]$. The result in Figure 5.7 shows that s is always p-stable at least once for samples of skewness below 1.5, and never p-stable for samples of skewness above.

5.3.2 Performance

We study the effect of the sample size on outlier detection in normal, Pareto, half-normal and symmetric Pareto distributions. We generate 1,000 samples for each distribution and size n, then we detect outliers on each sample. Normal and Pareto samples contain no outlier by definition, so no outlier should be detected; they are called *false outliers*.

We observe in Figure 5.8 that the rate of false outliers is low, with at most 3% for the normal distribution and at most 5% for Pareto. This rate decreases when n increases to be less than 1‰ above $n \approx 100$ for the normal distribution, and above $n \approx 500$ for the Pareto distribution. We also evaluate the rate of outliers detected for the symmetric Pareto distribution: reaching 5% at most, it seems to follow a fast decrease when n increases, to reach 1‰ at $n \approx 1000$. For the half-normal distribution, this rate is between 8% and 12% for $n > 100$, and is consistent with the fraction of samples for which s is never p-stable. We conclude that our method detects few false outliers on samples of size 100, and almost none on samples of size 1,000, which is an excellent performance; it rarely detects outliers on symmetric Pareto samples, which is the expected behavior regarding the characterization.

Now we estimate the ability to detect true outliers by generating a sample of size 1,000 composed of a normal sample of variance equal to 1 and a uniform sample (called the *noise*) of size varying from 0.2% to 50% of the total number of values. We then count the number of noise points which are classified as outliers and potential outliers. It is the worst case because the initial skewness is close to zero and outliers are uniformly distributed around the mean with no gap between them and the rest of the distribution. This is also a way to evaluate the robustness of our method against a problem known as the *masking* effect [24], occurring when some outliers are not detected because of the presence of other outliers close to them.

We generate uniform samples of various ranges (i.e. largest minus smallest value). The range of normal samples of size 1,000 is roughly 6 and the range of samples of size 10^6 is roughly 10, so we select

[4] $\gamma = (\sqrt{2} \cdot (4 - \pi))/(\pi - 2)^{3/2}$

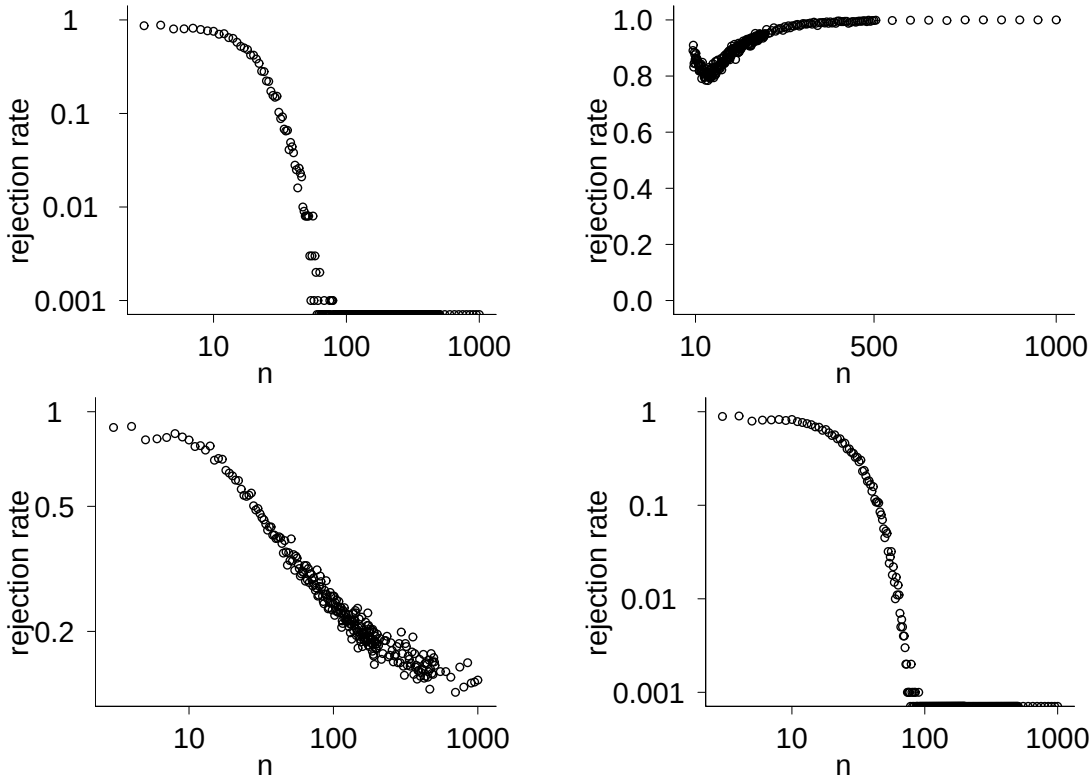


Figure 5.6: Fraction of samples for which s is never p-stable as a function of the sample size n, for normal (top left), Pareto (top right), half-normal (bottom left), and symmetric Pareto (bottom right) distributions.

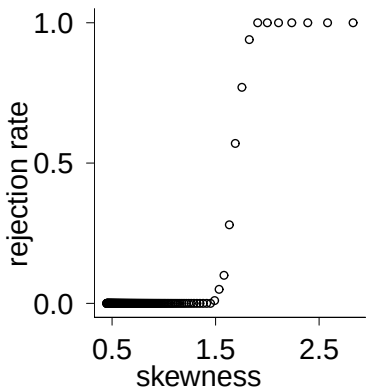


Figure 5.7: Fraction of samples for which s is never p-stable as a function of the skewness for Gamma samples (shape varying from 0.3 to 20).

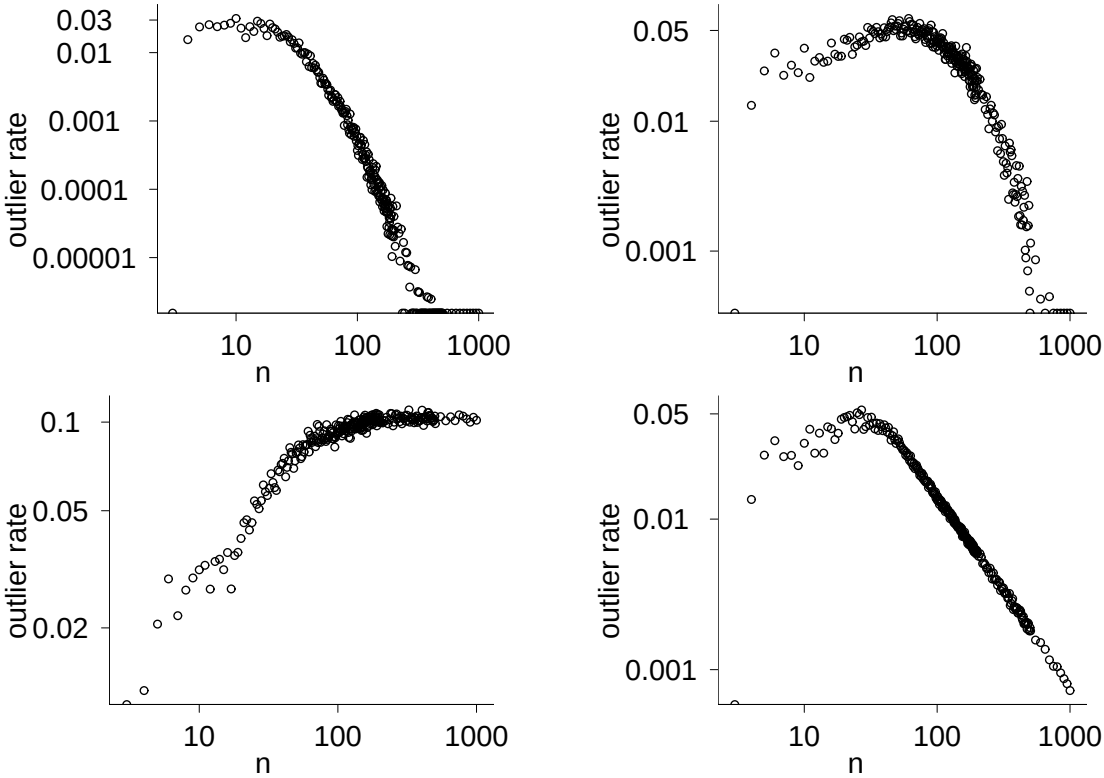


Figure 5.8: Fraction of sample points classified as outlier as a function of n for normal (top left), Pareto (top right), half-normal (bottom left), and symmetric Pareto (bottom right) distributions.

noise ranges larger than this: 10, 50 and 100. We observe in Figure 5.9 that noise points very close to the signal points (range 10) are classified as potential outlier. Larger ranges increase the number of detected outliers. We also see that the less noise, the higher the power to detect true outliers. However almost no outlier can be detected with more than 10% of uniform noise.

5.3.3 Regime Changes

Regime changes are change points in time series that are observed by sudden changes of the mean. When they occur we are faced with non-trivial distributions. We study now how our method deals with them. We simulate a stream of values by generating two normal samples of size 110 with mean equal to 0 and 3 respectively. t indicates the order of appearance of the values. Figure 5.10 shows our method applied dynamically with a sliding window of size $w = 100$. The outlier status of values is unknown at the beginning. At the end, none of them are outliers but one potential outlier. Our method is hence robust against regime changes. Notice that 72 values are classified as potential outliers when our method is applied on the whole data set at once.

5.4 Real-World Applications

5.4.1 Dynamics of Internet Topology

We applied our method to data collected with the radar for the Internet [81], which makes possible to observe the dynamics of the Internet's topology at the scale of a few minutes. It consists in focusing on the part of the Internet's topology viewed from a single computer called the *monitor*. Periodical measurements of this map, called *ego-centered view*, were performed every 15 minutes during several months, leading to a series of graphs.

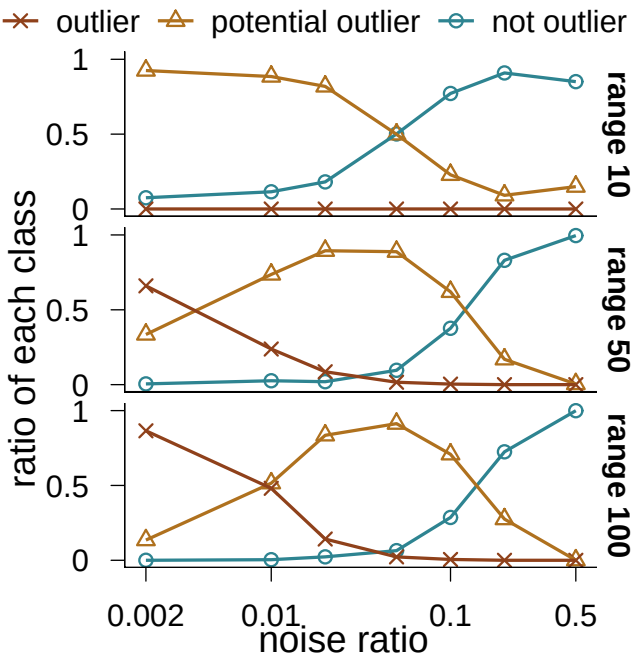


Figure 5.9: Ratio of noise points detected as outliers, potential outliers and not outliers as a function of the proportion of noise, for different noise ranges.

The most natural idea to detect events in the dynamics captured by a radar measurement from a given monitor certainly is to study the number N_i of nodes observed at each round i. We plot it for a typical case in Figure 5.1. Clear outliers appear under the form of sharp decreases of N_i for some values of i, but this brings little information because they may be due to losses of connectivity by the monitor. Except from these statistical outliers, which are detected by our method, the number N_i of nodes observed at each round i in Figure 5.11 is very stable.

We thus compute the number of distinct nodes seen in five consecutive rounds to avoid the outliers which only reveal losses of connectivity in one round of measurement. We observe events in the dynamics shown in Figure 5.12, where many decreases existing in Figure 5.11 have disappeared. Figure 5.12 is well centered around a typical value, but still exhibits sharp increases and decreases. This means that these outliers, which were also detected by our method, may reveal real events in the dynamics of this network. Outliers above the typical values indicate a sudden appearance of many new nodes in the network, while outliers below the typical values may indicate longer losses of connectivity or a sudden disappearance of many nodes.

Our approach is hence relevant for studying the evolution of ego-centered views of the Internet topology, and for raising automatic alerts in real-time when significant changes of connectivity occur.

5.4.2 Search Engine Queries

We applied our method on the data set of search queries captured from a eDonkey server [80]. It consists in textual queries made by users for lists of files matching certain keywords. The measurement lasted for 28 weeks. The data set contains 205,228,820 queries entered from 24,413,195 IP addresses. Samples and procedure descriptions are publicly available [80].

In order to study the number of queries related to the film *Harry Potter and the half blood prince*, we filtered the queries to get only those which contain the words ”half blood prince”. Then for every 10 minutes we counted the number of queries made during the last hour of measurement. Outliers were finally detected using a sliding window of size $w = 1,008$ (7 days) to capture meaningful events at the scale of one week. We plot in Figure 5.13 the number of outliers and potential outliers observed each day and each week. The scale of a day seems better for observing fast increases of user queries.

We identify three main events: we observe many values marked as potential outliers during the week

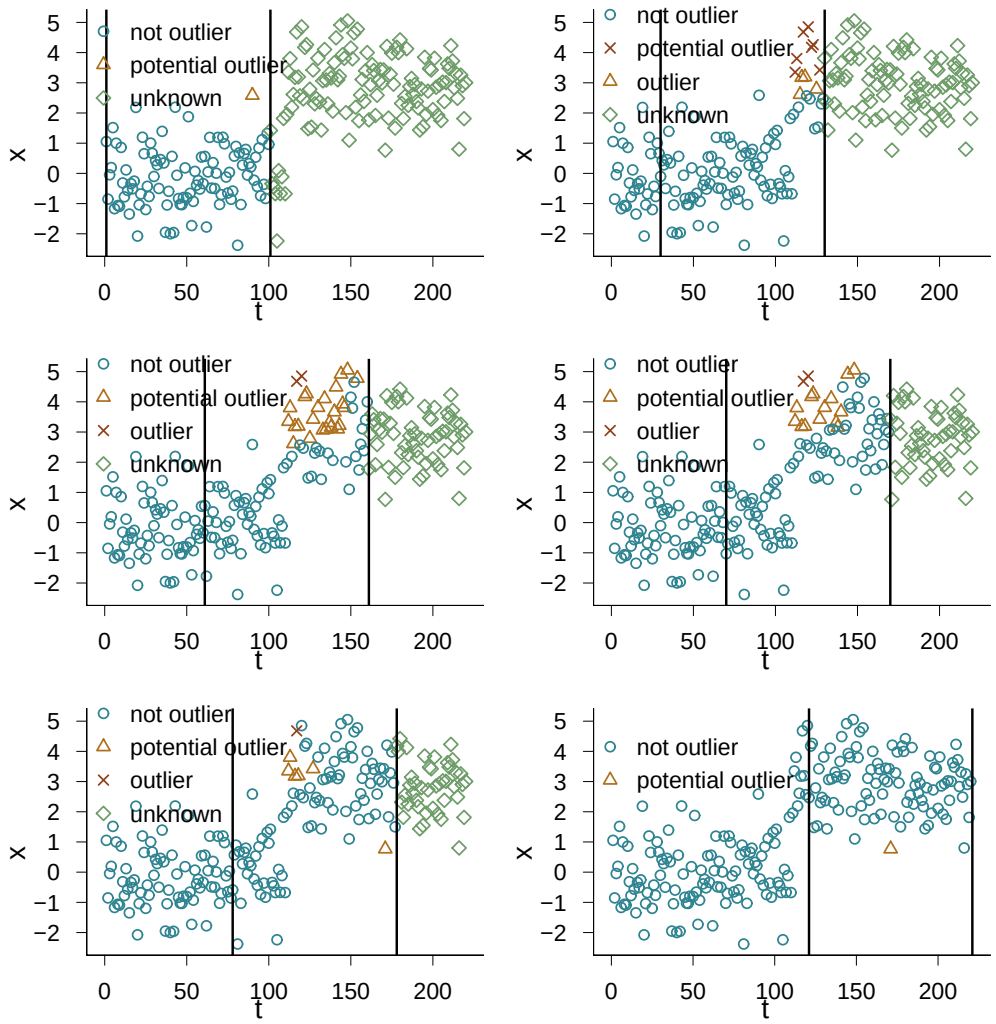


Figure 5.10: From top-left to bottom-right, evolution of the outlier status of values in a time series of size $n = 220$, and having one regime change (mean value changing from 0 to 3). Vertical lines indicate the time window boundaries between what outliers are detected.

after July 15, 2009, when the film was out in theaters. Then an unknown event appears from August 23 to 25, when almost all values are outliers. The last automatically detected event, from October 10 to 12, coincides with the release of a pirated version of the film on October 10 on BitTorrent, another P2P network, as discovered by searching on `https://thepiratebay.se`. We suppose that this release was made from a promotional DVD, because the commercial DVD was released on December 7 only; we observe no noticeable event on this day.

Our approach is hence relevant for studying logs of search queries, and for detecting bursts of queries related to a same topic.

5.5 Results summary and Next steps

The proposed *Outskewer* method to detect statistically significant outliers in samples and time series relies on the study of the distribution skewness. This method is easy to interpret because values are classified as *outliers*, *potential outliers* or *not outliers*. The class of all values is unknown when the notion of outlier is not relevant in the considered data set. Our method is also easy to use because it requires no prior knowledge on the data, and the only parameter is the size of the time window for time series. Moreover, it may be used on-line.

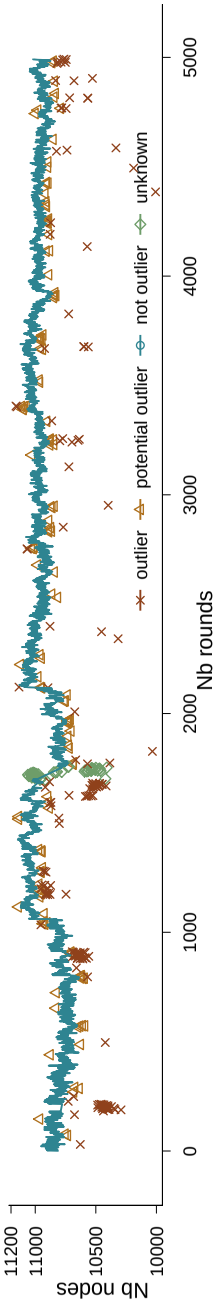


Figure 5.11: Number of nodes at each round of radar measurement; outliers are detected using a sliding window of 100 rounds (25 hours).

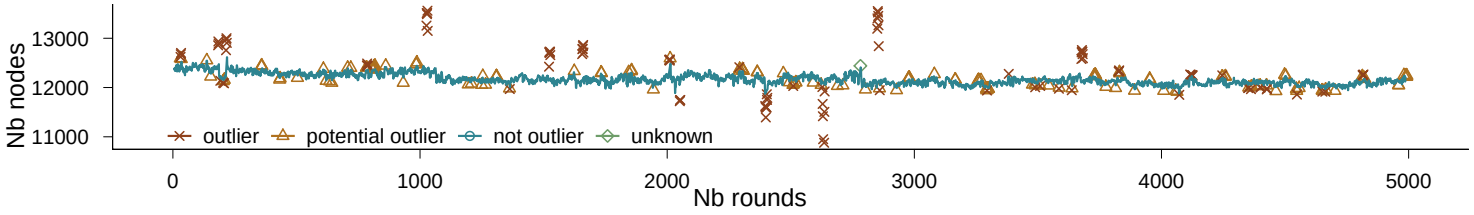


Figure 5.12: Number of nodes in the union of 5 consecutive rounds of radar measurement; outliers are detected using a sliding window of 100 rounds.

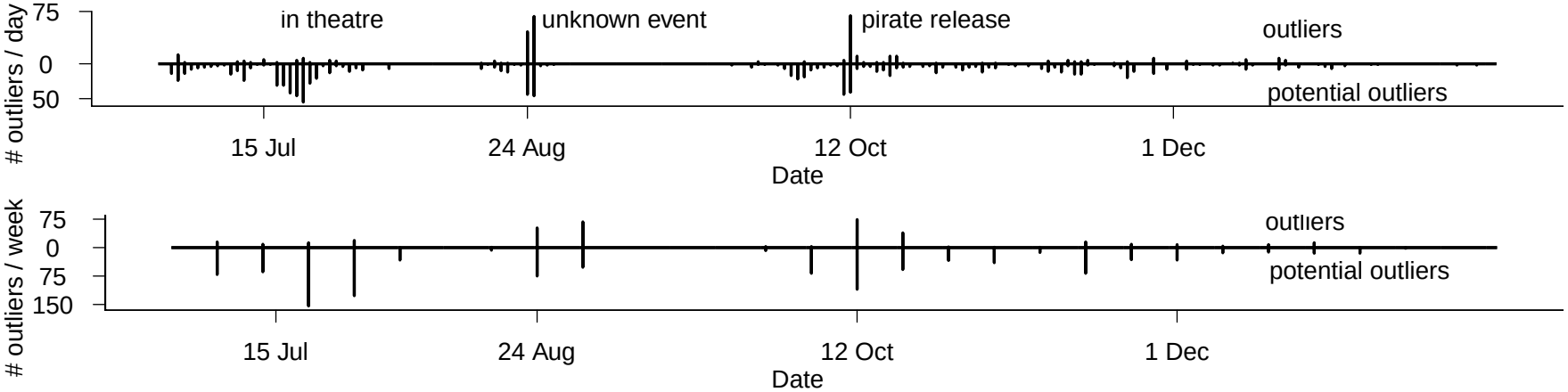


Figure 5.13: Number of outliers and potential outliers each day (top) and each week (bottom) in the number of search queries containing "half blood prince".

We applied this method on two data sets representative of many use-cases: evolution of ego-centered views of the Internet topology, and logs of queries entered into a search engine. We clearly identify events in the evolution of ego-centered views of the Internet topology as shown in Figure 5.11 and Figure 5.12. We also automatically detect the release of a pirated version of a film in a P2P system, through the queries entered by users in the search engine, as show in Figure 5.13.

This work opens the way to further investigation of the use of the skewness to detect multiple outliers in samples, and to detect events at different time scales in time series. Further studies may also extend our method to detect regime changes.

Chapter 6

Anaysis of P2P data

Peer-to-peer (p2p) file sharing applications have evolved into a major traffic source in the Internet [16, 4, 15, 69, 109]. This development has crucial implications for traffic engineering and information diffusion at the same time, since p2p networks constitute a remarkable case of interaction between a *technological* layer (network of computers) where the traffic occurs upon which we have a *social* layer (overlay network of peers, structured by related interests), where the file spreading occurs. Indeed, understanding p2p activity and its dynamics is critical to assure a good quality of service, enhance network architecture, forecast long-term provisioning and design better protocols upon it. Such tasks naturally rely on measurements of real-world traffic and models that can generate synthetic traffic. Consequently, identifying key properties of p2p traffic and conceiving models capable of reproducing them is decisive for the domain.

In the chapter, we present the analyses performed on a 48 hour record of the file sharing activity in an eDonkey server (akin to [8], see Deliverable D3.2 for furthur delails), suitably anonymized for privacy protection purposes. We show that our dataset presents non trivial properties, both in terms of download requests and file exchanges, that we study from the point of view of traffic dynamics and diffusion. We also propose different model in order to reproduce those observations. This work has to be seen in direct relation with WP2 and WP4 related to the elaboration of realistic scenario of traffic demand that can be used to test routing algorithm and policies proposed in the EULER project.

6.1 Cascade properties

In this section, we explain how traces of peer-to-peer file sharing may be used to this goal. We also perform simulations to assess the relevance of the standard SIR model to mimic key properties of spreading cascade. We examine the impact of the network topology on observed properties and finally turn to the evaluation of two heterogeneous versions of the SIR model. We conclude that all the models tested failed to reproduce key properties of such cascades: typically real spreading cascades are relatively “elongated” compared to simulated ones. We have also observed some interesting similarities common to all SIR models tested.

6.1.1 Introduction

Diffusion phenomena in complex networks – such as the spread of virus on contact networks, gossip on social networks and files in peer-to-peer (P2P) networks – have spawned an increasing interest in recent years. The boost of computer networks and online social network platforms offers data and new insights on these phenomena in large scale networks; the possibility to validate and refine current models might lead to breakthroughs in the field.

Although large scale diffusion phenomena have always attracted considerable interest, it has been historically challenging to obtain open, extensive and detailed real-world data at this level. Despite this obstacle, diffusion models emerged, notably in epidemiology. The early models, both discrete and continuous (see [13, 12] for a survey), focused primarily on *macroscopic* aspects of diffusion – such as the evolution of the number of infected individuals in a population – overlooking the *microscopic* dynamic of

the epidemic – i.e., how (by whom) individuals become infected. The advent of network analysis in various contexts has pushed for a more detailed description of the diffusion process. Indeed, models based on the detailed interactions of agents on a network have blossomed in sociology [58], computer science [49] and economics [68], among others. New epidemic models inspired by the classical approaches featuring a detailed dynamic description in the context of networks also appeared (see [20, 48] for a survey). In particular the network version of the SIR family of models has established itself as reference model in the study of information diffusion [38, 91, 39, 82, 93].

In this context, assessing the pertinence of such models to describe real-world data is critical. In order to validate this model a comprehensive empirical spreading trace, consisting of (1) detailed chronological data of who transmitted the information to whom and (2) data describing the underlying network on which the diffusion process takes place. Indeed, the network version of the SIR model (henceforth called simply SIR model) is based on local rules of transmission which take into account the network topology. In large epidemic bursts the available data often provides the evolution of large aggregate quantity, such as the number of touched individuals, but rarely uncover the local trail of the epidemic. Conversely, other empirical studies feature transmission events, but lack complete information of the underlying network structure on which the diffusion takes place [6, 67]. This work analyses the relevance of the SIR model for real-world diffusions, using data obtained measuring the activity on a peer-to-peer file sharing network. This rich dataset allows one to reconstruct both the underlying network and the detailed diffusion trail at a remarkable scale.

We begin with a description of our dataset and framework in section 6.1.2. In section 6.1.3 we define the spreading cascade. Next, in section 6.1.4, we simulate the spreading of files as a standard SIR process and confront it with the observed spreading; we also investigate the interplay between this process and structural properties of the underlying network where the spreading takes place. In section 6.1.5 we examine the spreading pattern when we modify the SIR model to account for heterogeneity in the behavior of the peers and in the popularity of files. We conclude with future work perspectives.

6.1.2 Dataset and framework

The data used in this study comes from file sharing in an eDonkey server, obtained from a measurement of six hours of activity (akin to [8]). In this setting, peers query the eDonkey server indexing files and for each file they get a list of available peers in the network possessing the requested file. Next, peers contact potential providers directly and transmission between them ensues. This dataset is a collection of answers to these queries, encoded as 4-tuples of integers in the following format: (t,P,C,F), where capital letters represent unique ids (e.g. in Figure 6.2). Each tuple accounts for a query made at time t of the file F by the peer C, satisfied by the peer P – that is, P has provided F to the peer C at time t. Let $\mathbf{D}$ be the set of all recorded tuples, $\mathscr{P}$ the set of all peers in these tuples and $\mathscr{F}$ the set of all files exchanged. In our dataset we have $|\mathscr{P}| = 1\,908\,500$ peers, $|\mathscr{F}| = 801\,280$ files and $|\mathbf{D}| = 22\,944\,800$ file transfers.

6.1.2.1 Underlying network

The trace $\mathbf{D}$ naturally induces a relationship between files and peers (who request or provide them), which we encode in a bipartite graph $\mathscr{B} = (\mathscr{P},\mathscr{F},\mathscr{A})$ on the disjoint sets of peers $\mathscr{P}$ and $\mathscr{F}$ files respectively. Let $(t,P,X,F) \in \mathbf{D}$ be a recorded transmission of the file F by the peer P to some peer X at some time t, which we denote simply by $(\cdot,P,\cdot,F)$. Likewise, let $(\cdot,\cdot,P,F) \in \mathbf{D}$ be a recorded transmission of the file F to the peer P, provided by some peer at some time instant. Hence:

$$\mathscr{A} = \{(P,F) \in \mathscr{P} \times \mathscr{F} : (\cdot,P,\cdot,F) \in \mathbf{D} \vee (\cdot,\cdot,P,F) \in \mathbf{D}\}$$

To study the diffusion, it is necessary to define the underlying network on which spreading takes place. Focusing on information content diffusion among peers, it is natural to consider the *interest graph* in which each node represents a peer and each edge joining two peers stand for common interest. Interests connecting peers may include broad subjects such as open source software, folk rock or French literature or narrow ones such as movies by Quentin Tarantino, a particular computer game or pictures

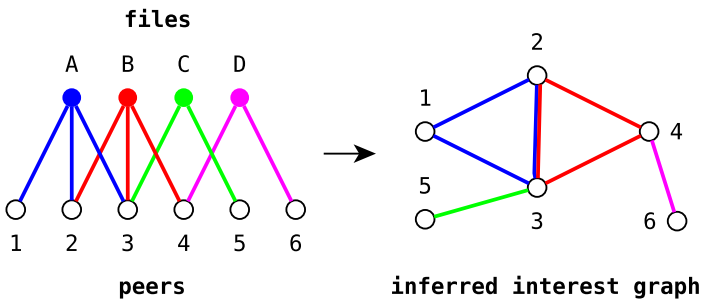


Figure 6.1: Interest graph as a projection of the bipartite graph of peers and files contructed from the trace **D**.

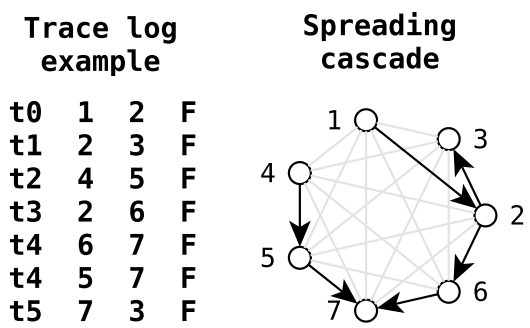


Figure 6.2: Trace log example with corresponding spreading cascade in black and underlying network in light gray.

of Beijing. It is reasonable to suppose that peers store and share content related to their interests and, likewise, peers will search for content matching their interests. Hence the diffusion of files among peers takes place on the interest graph and occurs from neighbor to neighbor. Indeed, if a peer P provides a file F (corresponding to a music album for example) to another peer P' then there is link between them in the interest graph, since both are interested in the same content, namely F.

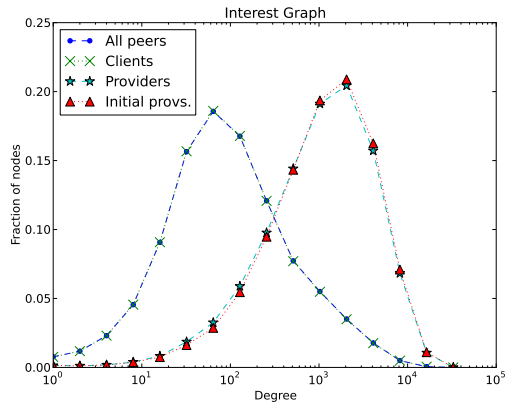
It is beyond doubt extremely difficult in a large scale interaction network to know precisely whether any two individuals have a common interest. Nonetheless, it is possible to approximate this graph using the data in **D**: the inferred interest graph is given by the projection $\mathscr{G} = (\mathscr{P}, \mathscr{E})$ of $\mathscr{B}$ on $\mathscr{P}$, connecting the peers who belong to the neighborhood of a common file in the bipartite graph, for each file:

$$\mathscr{E} = \{(P, P') \in \mathscr{P} \times \mathscr{P} : \exists F \in \mathscr{F}, (P, F) \in \mathscr{A} \ \wedge \ (P', F) \in \mathscr{A}\}$$

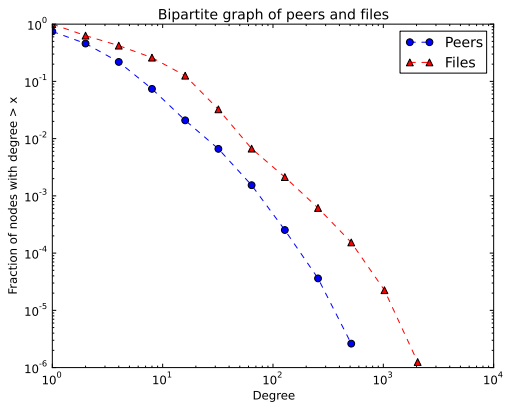
See example in Figure 6.1. For the sake of readability the inferred interest graph will be henceforth called simply *interest graph*.

6.1.2.2 Observed network structure

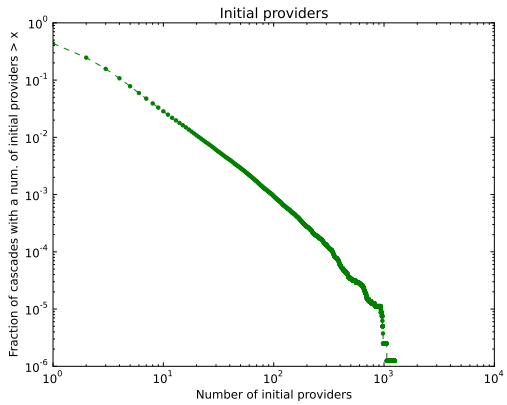
We begin examining properties of the bipartite graph $\mathscr{B}$ constructed from the P2P diffusion trace. In order to estimate the typical number of interested peers per file we have calculated the median degree of the files in the bipartite graph, 5, and the average degree, 14.73, with standard deviation of 34.74. Likewise, we have calculated the same statistics for the peers, to estimate the number of files commonly shared by peers: its median degree in the bipartite graph is 3 and the average degree is 6.19, with corresponding



(a) Degree distributions on the interest graph. Superposed curves: all peers and clients, providers and initial providers



(b) Complementary cumulative degree distributions of peers and files on the bipartite graph $\mathscr{B}$



(c) Complementary cumulative distribution of the number of initial providers in the spreading cascades

Figure 6.3: Properties of the underlying network and observed spreading cascades

standard deviation of 12.66. The degree distribution of both peers and files is however heterogeneous (Figure 6.3(b)) and mostly concentrated on small values; all degree values for peers and files remain below 10^4.

The interest graph obtained from the observed bipartite graph (as explained above and in Figure 6.1) has a single giant component containing almost all nodes (99.99%) and density 2.62×10^{-4}. In Figure 6.3(a) we have plotted the degree distribution for the peers: considering the set of all peers, the median degree is 118 and the mean value is 500.11, with corresponding standard deviation of 1271.42. We proceed to a finer analysis of the degree distribution, grouping peers in categories (Figure 6.3(a)). Let us consider first the set of *clients* $C \in \mathscr{P}$ such that $(\cdot,\cdot,C,\cdot) \in \mathbf{D}$: i.e., peers having requested files during our measurements. Their degree distribution superposes the degree distribution of all nodes. This is due to the fact that 99.63% of peers in our observations have requested at least one file, so the clients degree distribution is essentially the global degree distribution. A much more restrictive category is the set of *providers* P such that $(\cdot,P,\cdot,\cdot) \in \mathbf{D}$, i.e., peers having supplied files during our measurements. They account for 4.33% of the peers in $\mathscr{P}$. Their degree distribution has a similar shape, but it is concentrated on larger values, indicated by a median of 1821 and an average degree of 2906.54 – with corresponding standard deviation of 3471.80. The last curve, superposing the curve corresponding to the providers, represents the degree distribution of a particular subset of providers called *initial providers*, which will be detailed in the next section.

We close this section with a brief summary of our dataset. Using the framework introduced, we were

able to reconstruct the interest graph of the peers, where the spreading of files takes place. This graph connects essentially all peers, which can be grouped in two categories: providers and clients. Most peers in our observations are clients, but only a small fraction supply files: this revels a high proportion of *free-riders* (peers obtain files and do not share back) in the P2P-network observed. Furthermore, there is a sharp distinction between clients and providers in terms of their degree distribution.

6.1.3 Spreading in our data

In this work we analyze the *spreading cascade* representing the diffusion of each file in the P2P network. For a file F, the spreading cascade is a directed graph featuring the set $\mathscr{P}_F$ of peers who have participated in the spread of F (as clients and/or providers) and links $P \to C$, connecting each client C with the first peer(s) who provided F to it. More formally, let $\tau_F(C) = \inf\{t : (t,\cdot,C,F) \in \mathbf{D}\}$ be the first instant C obtained F and let the directed graph $\mathscr{K}_F = (\mathscr{P}_F, \mathscr{L}_F)$ be the spreading cascade of F, with

$$\mathscr{P}_F = \{P \in \mathscr{P} : (P,F) \in \mathscr{A}\}$$

$$\mathscr{L}_F = \cup_{C \in \mathscr{P}_F} \{(P,C) \in \mathscr{P}_F \times \mathscr{P}_F : (\tau_F(C), P, C, F) \in \mathbf{D}\}$$

A client requesting a file may receive a response from potentially several providers simultaneously, which implies that nodes in the cascade graph not only have multiple outgoing links, but also multiple incoming links in general. The causality induced by the fact that we only consider the links corresponding to the first time a node received F prevents the appearance of cycles. Hence the cascade is in fact a directed acyclic graph (DAG).

The first key property encoded in the spreading cascade of a given file F is the number of nodes who possess it at the end of the observed period, which is given by the *size* of the cascade $|\mathscr{P}_F|$. We also explore two other key topological properties of the cascade, namely its *depth* and *number of links*. The former is defined as the length of the longest path on the cascade and captures the maximum number of hops from peer to peer that the file has undergo before it was relayed from a provider to a client. The number of links, given by $|\mathscr{L}_F|$, combined with the size of the cascade gives information on the sharing pattern of the network. An example of observed trace and constructed spreading cascade is given in Figure 6.2: the spreading cascade has size 7, depth 3 and 6 links.

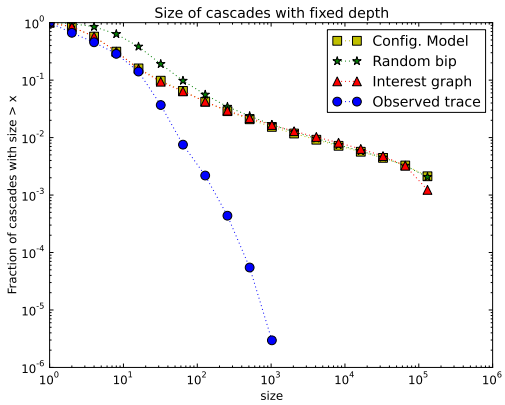
Another relevant spreading data concerns the *initial providers* for each file F, namely the set of peers that possessed it prior to any transfer activity on the observed trace. These nodes are the origin of the spreading cascade, triggering the diffusion of the file F. This information can also be inferred from the request log and be determined in the following way. Let $\mathscr{C}_F(t) = \{C \in \mathscr{P} : (t',\cdot,C,F) \in \mathbf{D}, t' < t\}$ be the set of peers who requested F prior to t. We define the set of initial providers of F as the set of peers P who have provided F at some time t, without having obtained it before t from another peer in the network:

$$\mathscr{I}_F = \{P \in \mathscr{P} : (t,P,\cdot,F) \in \mathbf{D}, P \notin \mathscr{C}_F(t)\}$$

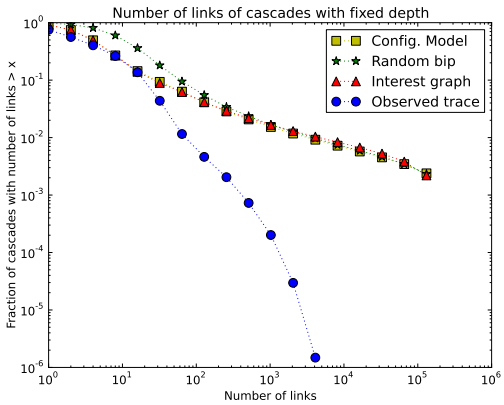
Plotting the complementary cumulative distribution of the number of initial providers for the spreading cascades (Figure 6.3(c)) we obtain an interesting curve, revealing a scale-free distribution. This means that although most spreading cascades in our observation have few initial providers, there is a non negligible fraction of cascades with a large number of initial providers.

6.1.4 Simple SIR model

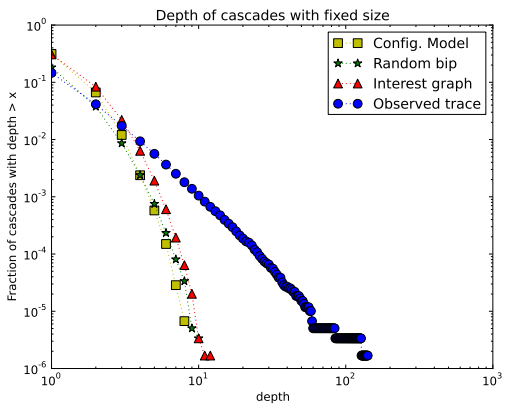
As mentioned in the introduction, we have decided to investigate the file spreading in the light of the simple SIR model. In our setting, each file spreading corresponds to an independent epidemic in the interest graph, in which each node is in one of the following states: *susceptible*, *infected* or *non-interacting* (sometimes denoted *removed*, hence the acronym SIR). Susceptible nodes do not possess the file and may receive it from an infected node, thus becoming infected. Infected nodes, in turn, spread the file to each of its neighbors, independently, with probability p and become promptly non-interacting thereafter.



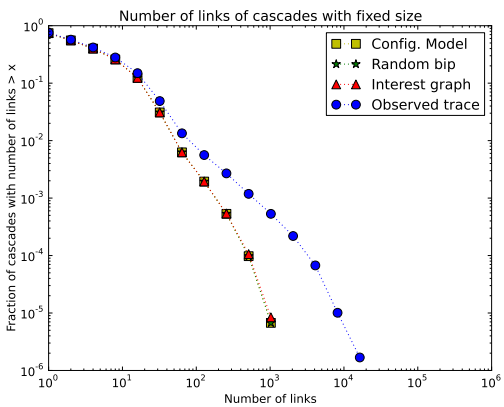
(a) Size of cascades with fixed depth. Curves corresponding to the interest graph and CM superposed.



(b) Number of links of cascades with fixed depth. Curves corresponding to the interest graph and CM superposed.



(c) Depth of cascades with fixed size.



(d) Number of links of cascades with fixed size. Curves corresponding to the interest graph, RB and CM superposed.

Figure 6.4: Simulation of file spreading on different underlying networks: complementary cumulative distribution of cascade properties

Although non-interacting nodes remain in this state, infected nodes may unsuccessfully try to infect them sending the file.

Supposing the observed diffusion trace was the result of such a simple SIR epidemic we may estimate the spreading parameter p. Each neighbor-to-neighbor transmission trial can be seen as a Bernoulli random variable, whose value is 1 in case of success and 0 otherwise and whose expected value is p. Assuming each trial is independent and the parameter p is homogeneous for each P and F, we may estimate it by the empirical proportion of successes over all trials. Since each tuple in $\mathbf{D}$ accounts for a successful neighbor-to-neighbor transmission, $|\mathbf{D}|$ is the number of successful trials for all diffusion cascades. The total number of trials, in turn, is given by the sum of the degrees of all nodes involved in the spreading of each file. Hence, we obtain the following estimate, with a 95% confidence interval $\hat{p} \pm 10^{-6}$:

$$\hat{p} = |\mathbf{D}| \,/\, \sum_{F \in \mathscr{F}} \sum_{P \in \mathscr{P}_F} d(P) = 1.063 \times 10^{-3}$$

Since the simple SIR model depends upon a single parameter, namely the spreading probability p, we have fully characterized it with the preceding estimation.

6.1.4.1 The underlying network influence

The goal of simulating the standard SIR model and comparing the simulated cascades with the observed ones is primarily to assess how realistic this model would perform on the interest graph, in terms of size, depth and number of links of the spreading cascades. Secondly, we wish to compare the results with simulations on random networks to understand the role of the network topological structure on the shape of the spreading cascades generated with the SIR model. With this aim, we have considered the spreading of files in a sequence of random networks derived from the interest graph, with increasing topological complexity. More precisely we begin considering an Erds-Rnyi (ER) random graph with the same density of our interest graph, the simplest random graph in our sequence. Then we have chosen a random graph with the same density and degree distribution using the Configuration Model (CM) approach [91]. Next we have generated a Random Bipartite (RB) graph, with the same density and degree distribution as our original bipartite graph $\mathscr{B}$ of peers and files [60]. Compared to the interest graph, the projection of this random bipartite graph has similar density, degree distribution and clustering coefficient. In sum, for each new element of this sequence of (uniformly chosen) random graphs we introduce a new constraint to make it more realistic – in the sense that its topological properties will be closer to the interest graph.

6.1.4.2 File spreading simulation

Combining the network topology, the initial condition information (the list of initial providers $\mathscr{I}_F$ calculated for each file F) and the calibrated spreading parameter $\hat{p}$ we can proceed to the simulations for each underlying network: for each F, we begin with the initial providers in an infected state and the other nodes in a susceptible state. At each step, infected nodes will infect each of its neighbors with probability $\hat{p}$, becoming non-interacting afterwards. The epidemic continues as long as there are active infected nodes.

The first observation concerning the model simulation is that the observed time (measured in seconds) has no direct relation with the simulation time (number of steps). Furthermore, our dataset corresponds to an observation in a bounded window of time of six hours, so that we have no reason to suppose that the file spreading cascades we observe correspond to the whole spreading cascade of a file. In other words, if we had measured a longer time window we would likely observe bigger cascades (in terms of size and depth) for the same files – due to, among other reasons, new users who could eventually request the same files. This is also true for our SIR model: we observe increasingly bigger cascades as time increases. In fact performing unconstrained simulations we have obtained a distribution of significantly bigger cascades than the ones we have observed in the real trace. Thus, in order to perform a suitable comparison with the observed cascades, we have decided to hold one property fixed and compare the other properties. More precisely for each file we generate a simulated cascade with the same size (resp. depth) as the corresponding observed cascade and compare the depth (resp. size) and number of links. In practice, for each file we simulate the SIR epidemic as described earlier and halt it when it reaches the size (resp. depth) of the corresponding observed cascade.

We have generated populations of simulated cascades for each underlying network and constraint (on depth and size). We have performed 801 280 file spreading simulations (one for each file in $\mathscr{F}$) for each network and have selected every simulated file spreading cascade which attained the depth (resp. size) of the real spreading cascade for the same file – and have rejected the others for purpose of comparison. With this procedure, each underlying network yields a different population of file spreading cascades, since the rejected cascades may be different in each case. However 93.80% of the files have generated simulated cascades with the same depth as the corresponding real cascades, for all networks. Similarly, 85.64% of the files have generated simulated cascades with the same size as the corresponding real cascades, for all networks – except the ER network. Indeed, only 21.76% of the files have generated the contemplated size in the ER graph. Furthermore the properties of these simulated cascades on the ER graph deviated significantly from the properties of the cascades on the other graphs. Hence, in the following analysis we do not include the simulations for the ER graph. Rather, we focus on the properties of the of files with comparable spreading cascade depth (resp. size) on all networks but ER.

In Figure 6.4(a) we plotted the complementary cumulative distribution of the size of cascades with

comparable depth. We observe a divergence of the cascade size from the observed cascades: simulated cascades are typically much bigger in size for a given depth compared to real cascades. The range of values in both categories is also striking: the biggest real cascade is at least two orders of magnitude smaller than the biggest simulated ones. Among the simulated cascades, there is a remarkable matching in size values for the simulation on the CM and the interest graph (curves are superposed). In Figure 6.4(c) we plot the complementary cumulative distribution of the depth of cascades with fixed size. Real cascades feature a much higher depth compared to simulations, holding cascade size constant. In particular there is a cutoff on the cascade depth for the simulations: we do not observe any cascade depth bigger than 11 in the simulations. As for the number of links, we have two interesting situations. If we fix the depth (Figure 6.4(b)) the number of links distribution resembles closely the size distribution (Figure 6.4(a)). This is not completely surprising, since the two quantities are related. In this case we observe a larger number of links for all simulations compared to the number of links in the real cascades since the simulated cascades itselves are bigger. If, in constrast, we fix the cascade size to fit the observed cascades size (Figure 6.4(d)), we observe a typically smaller number of links. Combining these observations on both plots we conclude that real spreading cascades are denser than simulated ones, a clear qualitative feature not captured by the simple SIR model. Finally we note that most cascades are simple, featuring depth equal to one and correspondingly small size.

To sum up, we have compared simple topological properties of real spreading cascades and simulated cascades from a calibrated SIR model, with comparable depth and size. We have observed that simulated cascades are relatively "wider" whereas real cascades are relatively "elongated", that is, real cascades have a smaller size per depth ratio. Moreover, real cascades are typically denser than simulated ones. In terms of interplay between underlying network structure and the simple SIR spreading cascades, we have observed that respecting the interest graph degree distribution was the only property that caused a striking change in simulations behavior on the considered random networks. Indeed we have observed sharp qualitative dissimilarities between the simulations on the ER graph (different degree distribution) and no sensible dissimilarities between the simulations on the CM, RB and the interest graphs.

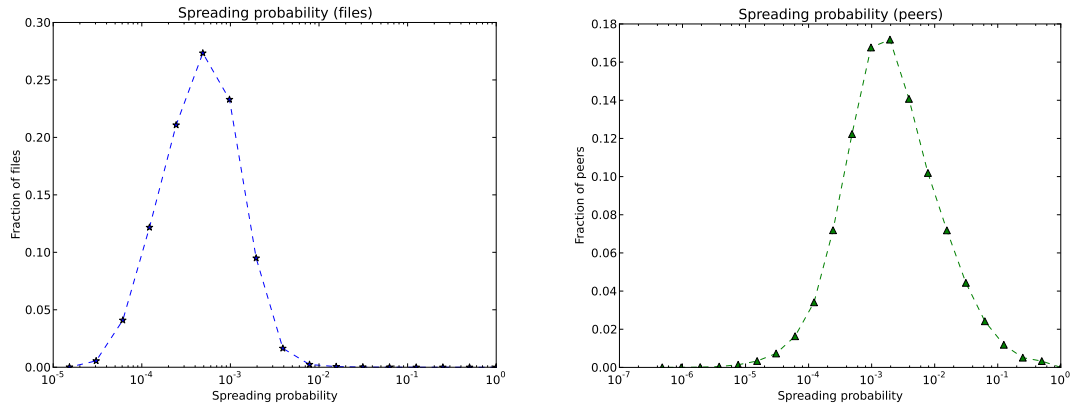
6.1.5 Heterogeneous SIR models

In the previous section we have examined the adequacy of the simple SIR model to generate verisimilar file spreading cascades. We have also inspected the interplay between the underlying network and the model simulating file spreading in different networks. In this section we perform a complementary analysis, focusing on a single underlying network and examining different extensions of the SIR model considered previously. In particular we consider two heterogeneous versions of the SIR model, characterized by a distribution of spreading probabilities, instead of a single homogeneous parameter. The natural choice in this case for the underlying network is the interest graph, which is the most complete and realistic graph among the ones tested in the previous section.

6.1.5.1 File popularity

A first refinement of the simple SIR model consists in introducing different spreading probabilities according to the file being spread. The rationale in this case is to account for different levels of popularity depending on the file. Exogenous reasons – such as a movie release or the death of an an artist – can change the supply and demand of a given file and consequently alter its spreading probability. The knowledge of the actual reasons that explain the heterogeneity in file popularity are irrelevant to the characterization of this model, if we know the spreading probabilities for each file, i.e., $\{p(F) : F \in \mathscr{F}\}$. An estimate of these probabilities, in turn, can be obtained from the trace $\mathbf{D}$ if we suppose it was generated by a process following this extended SIR model. Indeed, since each file spreading is independent of the others, it is possible to estimate $p(F)$ for each F separately, with the same method used to derive the homogeneous parameter. Restricting the calculations to the spreading cascade of F, $\hat{p}(F)$ will be given by the empirical proportion of successful transmissions of F over all possible transmissions of F:

$$\hat{p}(F) = |\{(\cdot,\cdot,\cdot,F) \in \mathbf{D}\}| \,/\, \sum_{P \in \mathscr{P}_F} d(P)$$



(a) Depending on the files

(b) Depending on the peers

Figure 6.5: Heterogeneous spreading parameter distributions

In Figure 6.5(a) we plot the distribution of the heterogeneous spreading parameters depending on the files. The values of $\hat{p}$ are concentrated on the range 10^{-5} to 10^{-2}, indicating that there is a considerable fraction of cascades with a significantly different spreading regime (bigger than one order of magnitude). This distribution characterizes the extended SIR model we use in the following simulations.

6.1.5.2 Peer behavior

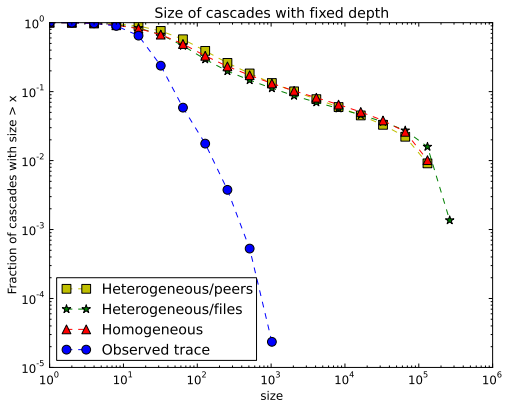
A second possible refinement is motivated by the fact that peers might have intrinsically distinct levels of "generosity" regarding file sharing. Under this hypothesis we extend the standard SIR model assigning an heterogeneous spreading probability to each peer, regardless of which file it is sharing. Thus, we do not need any other information but the spreading probability distribution to characterize the model. In this context altruistic peers, who typically spread files to a large proportion of their neighbors, would feature a bigger spreading probability compared to the homogeneous spreading probability corresponding to the diffusion aggregates of all peers. By the same token, the extreme case of free-riders would have their spreading probability assigned to zero. Again we can study transmissions as outcomes of Bernoulli trials to estimate the spreading probabilities. Let $\mathscr{F}_P = \{F \in \mathscr{F} : (P,F) \in \mathscr{A}\}$ be the files carried by the peer P; for each such file the number of transmission trials P could perform corresponds to its degree in the interest graph, namely $d(P)$. Hence, to obtain $\hat{p}(P)$ for each peer P we divide the number of successful transmissions of P to other peers (of any file carried by P) over the total number of potential trials:

$$\hat{p}(P) = \frac{|\{(\cdot,P,\cdot,\cdot) \in \mathbf{D}\}|}{|\mathscr{F}_P| \times d(P)}$$

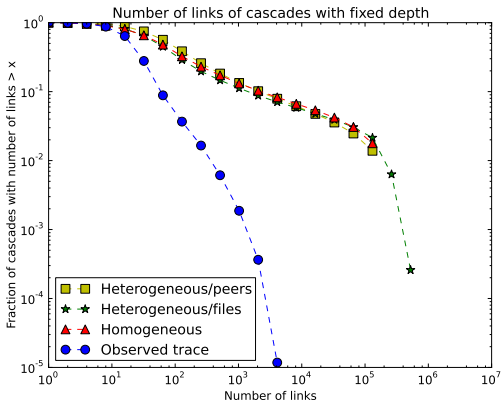
We have plotted the distribution of the positive spreading probabilities estimates in this case (Figure 6.5(b)). They account for small fraction of all the peers, since the only peers who have a positive spreading probability are those who provided a file at least once – 4.33% cf. observations made in section 6.1.2. Conversely, a large fraction of the peers do not share the file in this model. We observe a marked range of values, which is significantly greater than the one calculated for the homogeneous SIR.

6.1.5.3 File spreading simulation

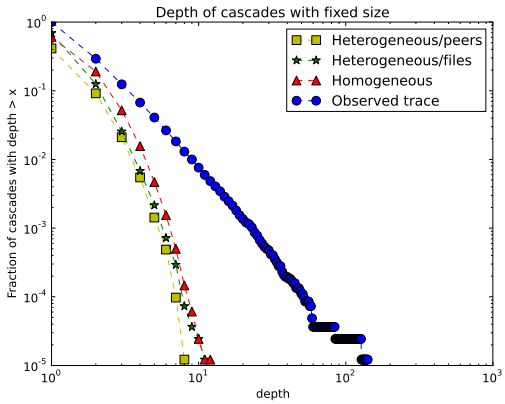
Our aim is to generate simulated cascades following both extensions of the SIR model presented – with heterogeneous spreading probability depending on the files and on the peers – and compare their properties with the simulated cascade of the simple SIR model and the real observed cascades. In this sense, we apply the same methodology of the previous simulations: we fix the depth (resp. size) for the simulated cascades and examine the other two properties – the idea is to compare similar spreading cascades in terms of the chosen property. As discussed previously, the great majority of the cascades is simple, with depth equal to one and a small size. Hence the simulated cascades corresponding to the simple observed cascades will likely correspond in terms of depth, size and number of links. For this reason, we have



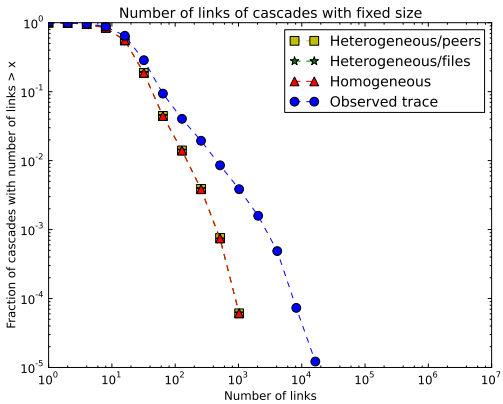
(a) Size of cascades with fixed depth. Curves corresponding to the simulations are superposed.



(b) Number of links of cascades with fixed depth. Curves corresponding to the simulations are superposed.



(c) Depth of cascades with fixed size.



(d) Number of links of cascades with fixed size. Curves corresponding to the simulations are superposed.

Figure 6.6: Simulation of file spreading on the interest graph with different SIR processes: complementary cumulative distribution of cascade properties

decided in this section to focus on the spreading cascades with depth greater than one.

The simulation results are plotted in Figure 6.6: we have plotted the complementary cumulative distributions of the spreading cascade depth, size and number of links. Imposing a constrain on the depth for the simulated cascades and comparing their size (Figure 6.6(a)) we observe the contrast between the simulated and the real observed cascades with the same depth: the former have a typically bigger size compared to latter. What is remarkable, however, is the agreement among all the simulated cascade distributions – curves superposed in Figure 6.6(a). Next, if we fix the size for the simulated cascades and examine their depth, we are faced with the same qualitative similarity among simulated curves. Indeed, the curves corresponding to the heterogeneous SIR models also feature a cutoff in depth, failing to reproduce the scale-free curve representing the depth of the observed real cascades. Finally, the cascade links distribution plotted in Figure 6.6(b) and Figure 6.6(d) reveals the pattern observed previously, namely that the observed spreading cascades are typically denser than corresponding simulated cascades.

Inspite of the improvements in the SIR model, introducing an heterogeneous spreading parameter to account for different profile of files (respectively peers), the simulations indicate that this refinement does not change qualitatively the basic properties of the simulated spreading cascades. Indeed we observe a surprising agreement between the three SIR models compared, notwithstanding the particularities of each model.

6.1.6 Conclusion and perspectives

We have presented a large-scale dataset from a real-world peer-to-peer network, featuring diffusion of files among peers. We have proposed a framework to study this dataset which allows us to obtain, simultaneously, the interest graph of peers – where the diffusion of content takes place – and the spreading cascade. Guided by simulations we have examined spreading cascades generated by the simple SIR model and have analyzed the interplay between this model and the network topology. We concluded that simulated file spreadings do not capture key qualitative properties of the observed spreading cascades. Furthermore, in terms of the studied properties, the simple SIR model generates similar cascades on random networks having the same degree distribution as the interest graph. Next we have focused on the spreading of files on the interest graph and studied extended versions of the SIR model featuring an heterogeneous spreading parameter. Surprisingly enough, simulated cascades using both extensions of the SIR model show similar properties as the simple homogeneous SIR model – and thus, fail to reproduce qualitative features of the observed cascades.

The SIR model is an attractive choice to model the information spreading in complex networks: it is based on classical epidemiological models, it is based upon few assumptions and can be characterized with one parameter. However, the results suggest that this model might not be suited to describe file spreading in our data. Furthermore, extensions of this epidemic model to make it more realistic, featuring heterogeneous spreading probabilities do not offer a better alternative in terms of the properties we observed. At this point, we consider two main exploration tracks. The first possibility consists in constructing a weighted interest graph, which takes into account the number of interactions (file exchanges) between peers. In this case the same analysis may be performed and a comparison with the results presented here would be pertinent. The second possibility is to contrast epidemiological models to adoption/threshold models [38, 58].

6.2 Statistical properties of the file exchange times series

A key step to characterise information supply and demand on an overlay network such as the P2P network considered here is the study of statistical properties of file request times from peers.

As a preliminary illustration, Figure 6.7 shows the evolution of cumulated number of requests for a few selected files. One can see regular progression, indicating a constant rate of requests, interspaced in some cases with short or long plateaux, suggesting that no provider (by provider, we mean a peer owning the file and ready to share it) is online at that time interval ; the unsuccessful requests are unrecorded in the data.

In a first part we notice that peers show a wide range of activity, as well as files show a wide range of popularity. We then show the time evolution of the intensity of requests, suggesting a marked circadian rhythm. Then we study the burstiness of the data, showing a slight burstiness in the requests emitted by a given typical peer, and no evident burstiness for the requests of a given typical file. Therefore it suggests that the requests of peers for files can be correctly approximated by Poisson processes with arrival rates that follow a circadian rhythm. Finally, we estimate from simple assumptions the number of peers that own a file and the number of peers that offer the file for sharing. In this way we can estimate the number of 'good peers', who transmit back the files they download, as opposed to the 'free riders' who do not contribute widely to the distribution of files in the system.

6.2.1 Activity and popularity distributions

By activity of a peer, we mean the total number of different requested files over the observation period. In other words several requests by the peer for a same file are counted as one. By popularity of a file we mean the total number of distinct peers that have requested the file. Here again multiple requests from one peer are counted as one. Note that activity and popularity are precisely the degree of nodes in the bipartite graph defined in Section **??**.

We show in Figure 6.8 the distribution of activity among users, and the distribution of popularity

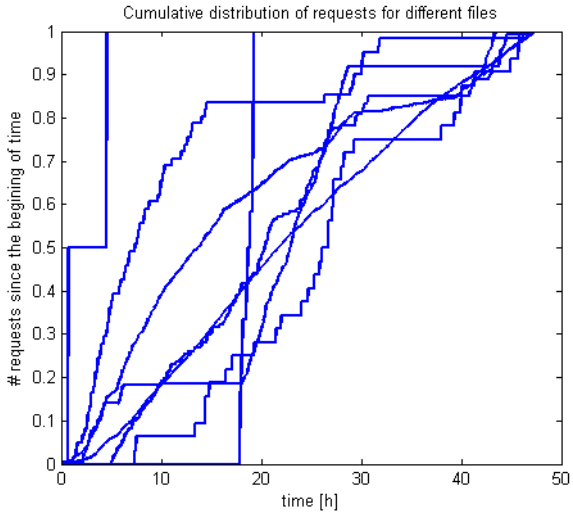


Figure 6.7: Cumulative request curves for several files, each normalized by the total number of arrivals in the 2-days time-window.

among files. It can be observed that those distributions, if not properly scale-free, seem to be heavy-tailed.

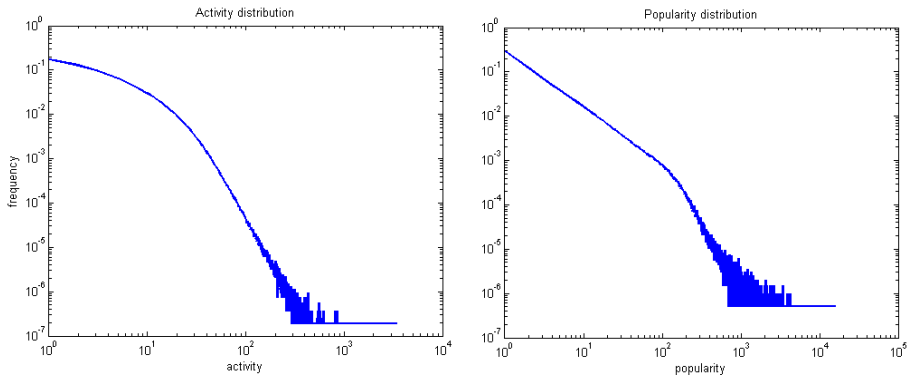


Figure 6.8: The distribution of activity (left) and popularity (right) in loglog scale.

It should be noted that the data for small values of popularity and activity can be corrected in the following way. Suppose in first approximation that the arrival of requests for a given file is a Poisson process with an average of λ arrivals in the observation period. Then the popularity actually observed for this file may be more or less than λ, and those fluctuations around the expected value will be averaged out if a sufficient number of files are present with the same parameter λ (which is certainly the case for not-too-popular files). However there is also a probability $e^{-\lambda}$ that the file will not be observed at all, thus will not be represented in the Figure, introducing a bias that tends to underestimate the number of unpopular files. Therefore, each point of this curve corresponding to popularity λ should have its number of occurences multiplied by $(1-e^{-\lambda})^{-1}$. A similar effect takes place for the activity distribution among users, of course. However it would lead to only a small correction of the curves, especially in the log-log scale used here.

Note that alternative measure for activity (resp. popularity) would be to normalize the above-defined measure by the total online time (resp. availability time) of the considered client (resp. file). These normalized measure would give a better indicator of how a client is active when connected or how a file is popular when available. However they are less accessible as the evaluation the online time (availability time) cannot made reliably.

6.2.2 Circadian rhythm

Counting the number of requests of all peers for all files altogether at every time instant (Figure 6.9) shows the existence of a rhythm that probably corresponds to a circadian rhythm of the peers using the observed eDonkey server. This shows that the random process meant to be a model of the time series of requests must be non stationary, with a cyclic variation of the parameter across the day. The data we have, covering only 48 hours, does not allow us to extrapolate to a weekly cycle.

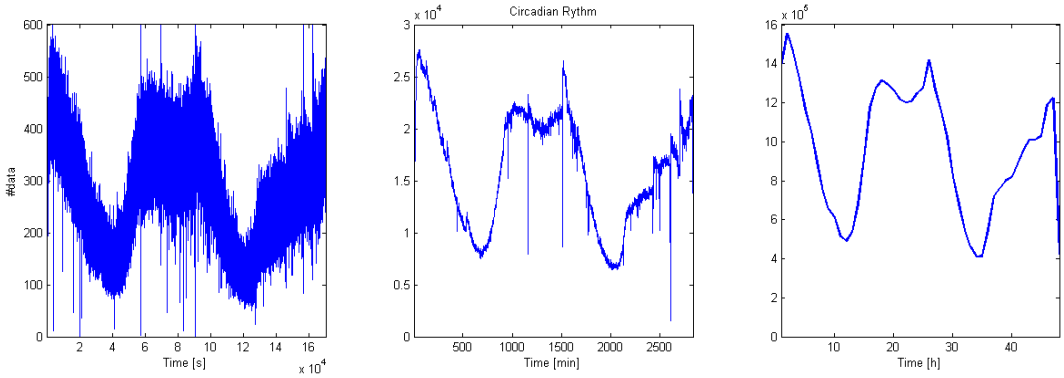


Figure 6.9: The circadian rhythm averaged for different time scales (seconds, minutes and hours). Nights and days are clearly visible.

6.2.3 Burstiness

6.2.3.1 Measure of burstiness

In general, burstiness is meant as a particular deviation from Poisson behaviour in a series of arrivals or 'events' (note that the word 'event' is overloaded in probability theory so we avoid to use it in the following).

Suppose that an arrival occurred at time zero. For a Poisson arrival process, the probability of an arrival in a short time interval $[t, t+\Delta t]$ (conditionnally on the fact that no arrival has occurred in $[0,t]$) is $\lambda\Delta t + o(\Delta t)$, not function of the time t elapsed since the last arrival. This leads to an exponential probability $e^{-\lambda\Delta t}\lambda\Delta t$ of an unconditionnal first arrival occurring in the interval $[t, t+\Delta t]$, an expected time $\tau = \lambda^{-1}$ for the first arrival, and a number of arrivals in $]0,t]$ following a Poisson law (with a probability $\frac{(\lambda t)^k}{k!}e^{-\lambda t}$ of k arrivals in that interval).

The assumption that the probability rate for arrivals λ is a constant function of t is not always verified for various kinds of arrival processes associated to various social contact networks, see for instance phone networks, e-mail networks or sexual contact networks [67, 70, 116, 105]. Note that those contact networks can be directed, in the sense that one node takes the initiative of a contact (e.g. phone call) with another node, which receives the contact. The directedness can be preserved or ignored following the kind of process that is studied on the network (e.g. the spreading of information or of a disease).

6.2.3.2 Burstiness analysis on peer-to-peer networks

We want to test the existence of burstiness on the peer-to-peer data in order to shed some light on the behaviour of users of the Internet, at least in the P2P application. One may think that the activity of users will happen in bursts: a user who decides to request a file is likely to download another one shortly afterwards. Similarly, a small number of requests for an initially unpopular file may unlock other requests thus creating a burst of interest for that file.

For every node, or at least the most active/popular among them, we can study the inter-request time through its average and its variance, and compute its burstiness as defined above. Since we cannot expect to have an homogeneous Poisson process on the 2 days time interval, we limit ourself to time windows of limited length (e.g., 1 hour). This also has the advantage that it smoothes the effect of the circadian

rythm. For each time window, we compare the average inter-event time with its standard deviation : if both measures are close, then it means that the burstiness is low on that window.

On Figure 6.10 we represented the mean and variance of the arrival rate of requests from a given peer, on a moving window of one hour-length. We show a remarkable correspondance for all the files that we have looked at.

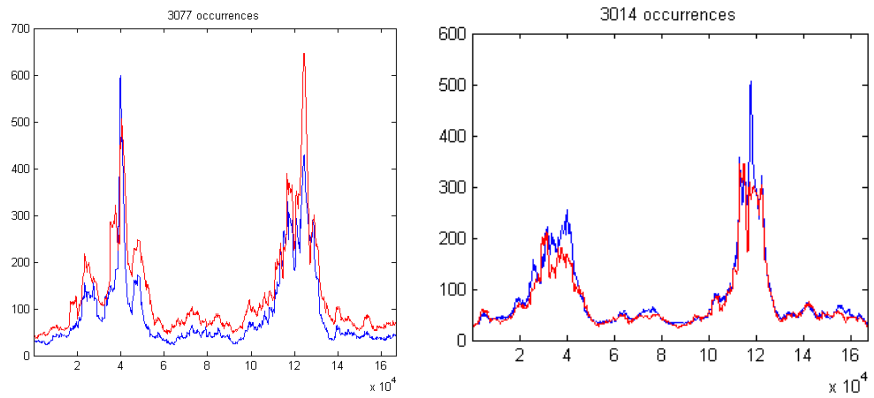


Figure 6.10: Left : we compare the average inter-request time of a typical active client (blue) with its standard deviation (red). Each point corresponds to a measure estimation on a 1-hour time-window. The chosen client appears 3077 times in the data. Right : the same for a typical popular file that was requested 3014 times in the data. Clearly, blue and red curves seem to be similar (especially for files) which suggests local non-burstiness.

The burstiness of the number of requests for a given peer is however a little more pronouced (Figure 6.10). Nevertheless it seems fair to conclude that burstiness is not a dominant property for those time series, which can therefore be represented by a Poisson process whose value is proportional to the activity of the given peer/the popularity of the given file, modulated by the time of the day according to the circadian rhythm.

6.2.3.3 Burstiness of traffic demand

The models we draw on P2P modelling can be used as generators of end-to-end traffic demand taking place on a physical model of the Internet. The conclusions above show that it is not unreasonable to model those with Poisson processes whose parameters vary in time according to the time zone of the user. Of course this requires an extrapolation from the traffic characteristics of P2P application to any kind of traffic, but again this extrapolation does not seem unreasonable.

The modelling of activity of users by a Poissonian process instead of a bursty process calls for some comments here. It has been showed, in the last two decades more particularly, that many probability distributions characterising human activities are sometimes heavy tailed, e.g. described by a power law, also called Pareto distribution. However later on it has been noticed that some of those distributions were only apparently heavy tailed, and that a proper statistical hypothesis testing was not conclusive [34, 101, 50].

This debate has found a particular resonance in the case of human-generated stochastic processes. E-mail activity data has been found and explained to have time-invariant Pareto-like inter-arrival times [17]. It was later argued that the data was better fitted and more simply explained by a time-varying Poisson process, with a parameter following weekly and daily patterns [88]. Albeit illustrated by e-mail data, the discussion of these two papers were of a general nature. This shows the difficulty of distinguishing between a time-invariant bursty model and a time-varying Poisson process.

Peer-to-peer inter-packet time has been found to fit acurately a heavy tailed hybrid Weibul-Pareto distribution e.g. in [21], while the statistics we measured are fully compatible with a time-varying Poisson process for the inter-request time. This apparent contradiction may be seen as a further instance of the dichotomy above, but also of the fact that a Poissonian flow of requests may lead to a non-Poissonian flow of packets, due to the fact that files sizes may themselves be approximated by a Pareto distribution

[42, 89]. A model of requests, together with a distribution of file sizes and the assumption that every request is followed by a file download, therefore induce a model of Internet traffic.

6.2.4 Good peers and free riders

The presence of relatively long plateaux in Figure 6.7 in the cumulated request curve even for not-so-unpopular files suggests that only a fraction of requests translate into sources for the further propagation of the file. This may be explained by the fact that not all requests for a file are followed by a successful download, or that the peer having proceeded to the download does not offer the file for further sharing on the network (thus called 'free riders'). It is hard to distinguish between those two phenomena, but by simplicity we call free riders all peers that for one reason or another do not share their file.

To estimate the number $p(0)$ of peers owning a file and ready to share it at time zero, whether or not they are online, we count the number of peers appearing as providers at least once in the time interval, that never appear as requesters for this file. Then we increment this number of potentially available providers $p(t)$ with time t, as requesters are subsequently observed as a provider for the first time before time t.

We then estimate the total number $d(t)$ of peers owning the file (whether they are willing to share it or not) at time t as the sum of $s(0)$ and the number of peers having requested the file for the first time before time t.

The comparison of the two curves for several files (Figure 6.11) shows very similar profiles for the evolution of $p(t)$ and $d(t)$, with a ratio of the slopes of approximately 8. This suggests, according to the above assumptions, seventy-eight percent of free riders for only twelve percent of 'good peers'.

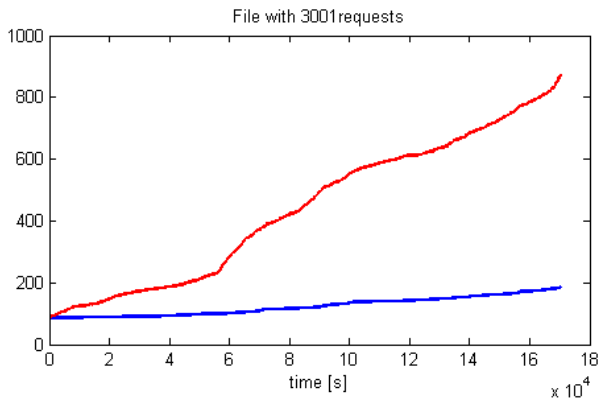


Figure 6.11: The blue curve is an estimation of the evolution of the number of providers while the red curve is the evolution of the number of clients. By looking at the slopes, there seems to be 8 times more new clients than new providers.

6.3 Traffic simulation on a P2P network

6.3.1 Introduction

In this section, we describe two different models which aim to generate synthetic traffic in the peer-to-peer networks. These models take a set of peers N and a set of files F as the input and generate traffic for any desirable time window of length T. First we propose our model which is based on the Markovian transitions and illustrate the outline of the implementation. Next, we describe an agent based model which is adapted from an existing framework proposed in [55]. Finally we show the results of both of these models and highlight the improvements of the proposed model over the agent based model.

6.3.2 Markovian transition based model

This model is based on the following statistics which can be obtained from the empirical dataset.

1. Popularity distribution of files
2. Activity distribution of the peers
3. Login-logout frequency distribution of the peers
4. Proportion of free riders

Ideally, these statistics should be obtained from the empirical dataset. However, first we synthetically generate these statistics based on some realistic assumptions in order to simulate our model. The details of this synthetic statistics generation methodology is illustrated in the next section.

Our model relies on the following assumptions

1. We ignore the client preferences towards specific files and subsequent community effects
2. We assume that downloading a file is instantaneous
3. Each peer can request/download one file at a time. However multiple peers can download a given file simultaneously.

Assumption 2 may be easily relaxed later by using the size of files as an indicator of the download time. In that case, it could be useful to assume that the file size to be independent from the file popularity. Note that assumption 1 can also be handled by tuning the preference vector of peers.

We split the rest of the section in two parts; in the first segment, we describe the synthetic generation of the statistics (such as login-logout frequency, peer activity distribution, file popularity distributions, etc.) and the second segment illustrates the traffic model and its implementation.

6.3.3 Synthetic statistics generation

In this section, we aim at synthetically computing the following statistical parameters.

a. Login-logout frequency of peers: We assume that the online and off-line slots of a peer node follow exponential distribution (see Fig. 6.12). Hence, the durations of the online slots On_i of peer i can be assigned based on the exponential distribution with parameter λ_{on_i} (λ_{on_i} is a random number < 0.2). Similarly, we generate offline slots Off_i following another exponential distribution with parameter λ_{off_i}. We continue to generate these slots (periodically online and offline) until we reach the end of the timespan T. From this statistics, we compute the average online (and offline) time of a peer i and subsequently compute their login and logout transition probabilities.

b. Peer activity distribution: For each peer i, we assign the total number of queries generated by that peer following a power law distribution (with exponent α) and suitably normalize it by the total online time of that peer. This eventually provides us the 'true' or 'a priori' peer activity distribution.

c. File popularity distribution: For each file f, we assign the popularity of that file (i.e. total number of queries for that file) following power law distribution (with exponent β). We normalize this quantity by the total number of queries made for all the files.

6.3.3.1 P2P Traffic model

We propose a Markovian transition based traffic model (see Fig. 6.13) where each peer node at any point of time can be in one the three states (such as 'Online', 'Off-line' and 'File download'). At each timestep, a peer node can switch from one state to another (or remain at its current state) based on the input statistics (login-logout frequency, peer activity distribution, file popularity etc). All the peers are independent in nature, hence they are allowed to act simultaneously. At time t, a peer node i may be in 'Offline' state Off_i or in 'Online' state On_i and the transition probabilities to switch from one state to another (and also to remain at its current state) in the next timestep $t+1$ is directed by the login-logout frequency of peer i. Moreover, while in the online state On_i, a peer i may initiate a file download based on its peer activity profile (obtained from peer activity distribution). The selection of the file is based on the **(a.)** popularity of that file and **(b.)** availability of the providers for that file. While downloading, the peer node switches from the 'Online' state to 'File download' state and it remains in the download state for one time unit (for simplicity, we assume that it takes one unit of time to download a file and this is fixed for all the

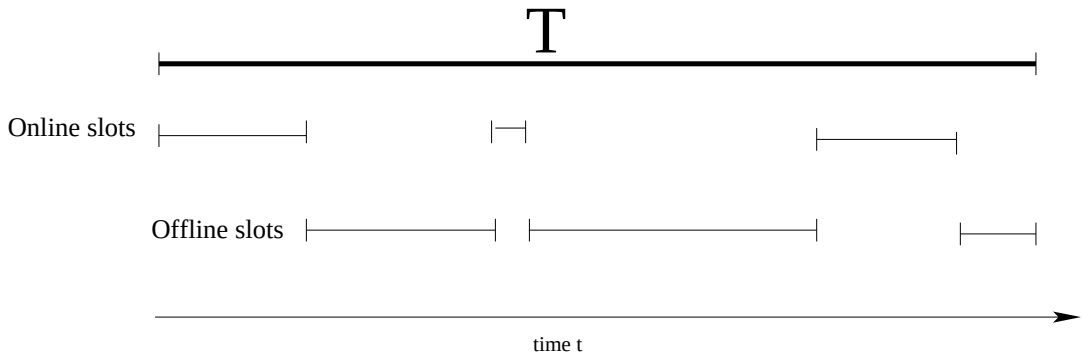


Figure 6.12: Online-offline slots of peer i

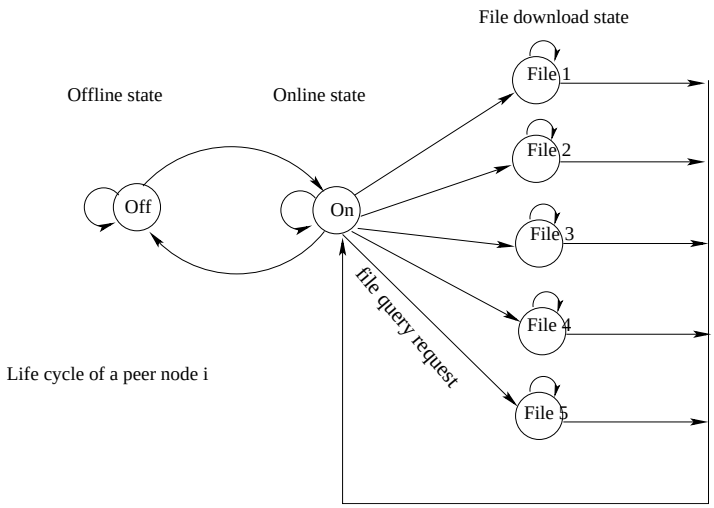


Figure 6.13: The life cycle of a peer node i. All the peers in the network follow similar state transition protocol.

files). After download, the peer again moves to the online state. This model also takes into account the cooperative peers and the free riders. The model assumes that the cooperative peers share their files after logging in whereas free rides do not share them (except when they are downloading a file).

The information about the shared files and their providers are maintained in a $m \times n$ state matrix where m is the number of peers ($|N|$) and n is the number of files ($|F|$). At each timestep t, we update the state matrix $A(t)$ such that it reflects the information of files shared by different peers, current state of a peer (online/offline), availability of providers for a file etc. We assume that $A_{ij}(t) = 0$ if peer i doesn't possess file j at time t. $A_{ij}(t) = 1$ if peer i possesses file j and online at time t. $A_{ij}(t) = -1$ if peer i possesses file j and offline at time t.

After each timestep, we update the state matrix A based on the activities of the peers.

- 1. If a peer i launches a download for a file j at time t, $A_{ij}(t) = 1$. Additionally, if i is a bad peer, assign $A_{ij}(t) = |A_{ij}(t-1)|$ to ensure that its files are shared at time t
- 2. If a bad peer i finishes downloading file j at time t, $A_{ij}(t) = -A_{ij}(t-1), \forall j$
- 3. If a good peer i disconnects, $A_{ij}(t) = -A_{ij}(t-1), \forall j$
- 4. If a good peer i connects, $A_{ij}(t) = |A_{ij}(t-1)|, \forall j$
- 5. $A_{ij}(t) = A_{ij}(t-1)$ for all other cases

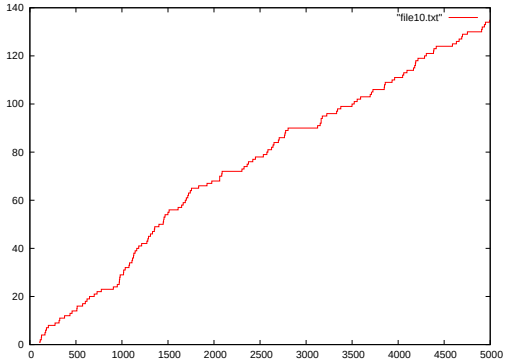


Figure 6.14: Cumulative number of downloads of a popular file in Markovian transition based model.

The number of providers $p_j(t)$ for a file j at time t can be computed from the state matrix $A(t)$. Lets assume $a_{ij}(t) = \max(A_{ij}(t), 0)$. Therefore the number of providers $p_j(t)$ for file j becomes

$$p_j(t) = \sum_i a_{ij}(t) \tag{6.1}$$

6.3.3.2 Simulation results

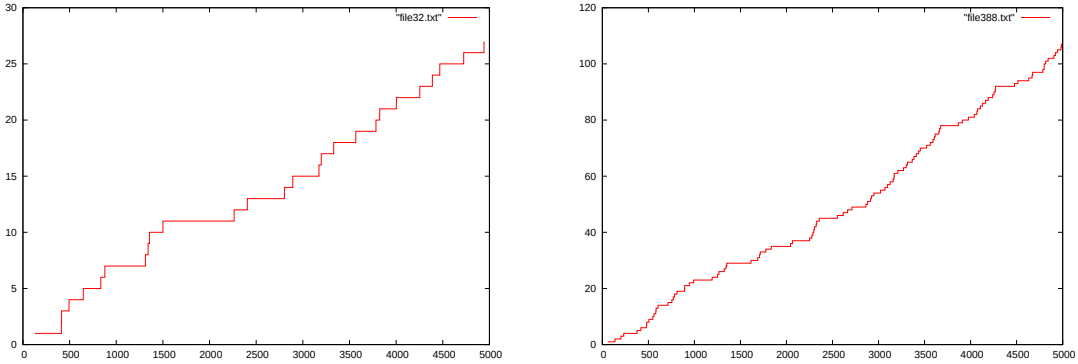
We implement the traffic model with the help of a discrete event simulator. We simulate the model with $|F| = 500$ files and $|N| = 500$ peers, of which 50% are free riders. We generate the synthetic statistics such as login-logout frequency of peer nodes, file popularity distribution and peer activity distribution using the methodology described in section 6.3.3. Fig. 6.14 shows the cumulative number of queries generated for a popular file ($file_{10}$) after $T = 5000$ timesteps. This is important to note that the slope of the curve is in general quite steep, which is an evidence of the popularity of that file. However, few plateaus can be observed possibly due to the unavailability of the online providers for that file. The similar kind of behavior can be observed for the moderately popular and less popular files also (Fig. 6.15(a), 6.15(b)). In Fig. 6.16(a), we illustrate the popularity distribution of files, both for the a priori power law distribution and the distribution obtained from the simulation. This is important to note that although there is an agreement between the true and observed distributions for the moderate to highly popular files, a strong discrepancy can be observed for the unpopular files. Since the a priori file popularity distribution follows power law, we have a large fraction of unpopular files in the network. However, only a few of these unpopular files gets downloaded by the peers and as a consequence, most of these unpopular files hardly find any provider for them. This essentially reduces the presence of unpopular files across the peer nodes in the network. Hence the frequency of unpopular files, although being high in the a priori file popularity distribution, becomes low across the peers in the network. On the other hand, the frequency of highly popular files, being low in the a priori file popularity distribution, remains low across the peers also. Nevertheless, Fig 6.16(b) shows that the observed peer activity distribution from the simulation has a nice agreement with the true peer activity.

6.3.4 Agent based model

In this section, we propose an agent based traffic model which is adapted from the framework proposed in [55]. This model is an alternative to the Markovian transition based traffic model explained above, and although the latter seems to outperform the former as we shall see, we explain it as a means of comparison.

The model is based on three basic components or blocks, each of which can be modeled with $M/M/\infty$ queues (see Fig. 6.17). The most simple model consists of 1. Offline block 2. Query generation block. 3. File download block (see Fig. (6.17)).

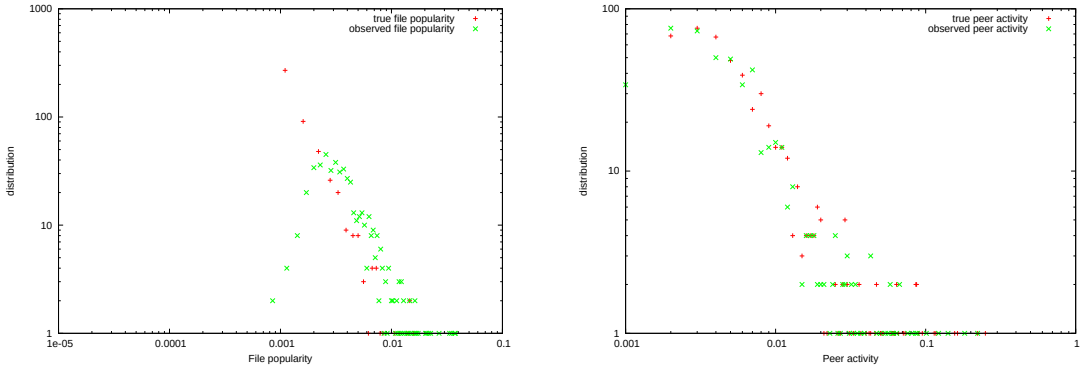
At any point of time, peer nodes in the system remain in one of these three blocks. Since this is a closed network system, the total number of peers in the network remain constant. Peer nodes move from



(a) file 32 (less popular file)

(b) file 388 (moderately popular file)

Figure 6.15: Cumulative number of downloads for Markovian transition based model



(a) File popularity distribution (a priori and observed)

(b) Peer activity distribution (a priori and observed)

Figure 6.16: Comparative study between the observed and true (a priori) distributions

one block to another based on the arrival and service rate of the respective queues. The basic assumption is that, a peer node either remains offline, or in online state. While online, a peer may be in idle state or it generates query and downloads files. In the idle state, a peer can work as a provider. The details of different blocks and their functionalities are explained next.

1. Offline block: At any point of time, this block stores all the peers which are in the offline state. We assign one dedicated queue-server for each individual peer node, which essentially captures the intrinsic properties of that individual peer (for example, frequency of login). The arrival rate at queue i is the rate at which peer i goes offline. We assign a service rate to the peer's server following a power law distribution. The service rate of a queue regulates the login rate of node i. This power law "service rate" emulates the fact that most of the peers in the Offline block 'slowly' move to the online block whereas only a few peers quickly switch their state from the offline to online.

2. Online-idle and query generation block: This block stores all the peers which are in online state; either they generate queries or remain idle (i.e., not generating queries, but may work as a provider). A dedicated queue-server is assigned for each individual peer node, which characterizes the query generation behavior of that peer (referred as peer activity). The (scale free) service rate assigned to the individual server results in a heterogeneous peer activity distribution. Arrival rate at queues depends on the login rate of peers (service rate of offline block). Service rate of the server depends on the rate at which peer node generates query. The peers waiting inside the queue remain idle and may work as a provider.

3. File download block: In this block, we assign one queue-server for each file in the system (all of them works in parallel). All the peers requesting a specific file arrive at the corresponding queue. The arrival rate at i^{th} queue depends on the popularity of file i in the network. The service rate of a queue

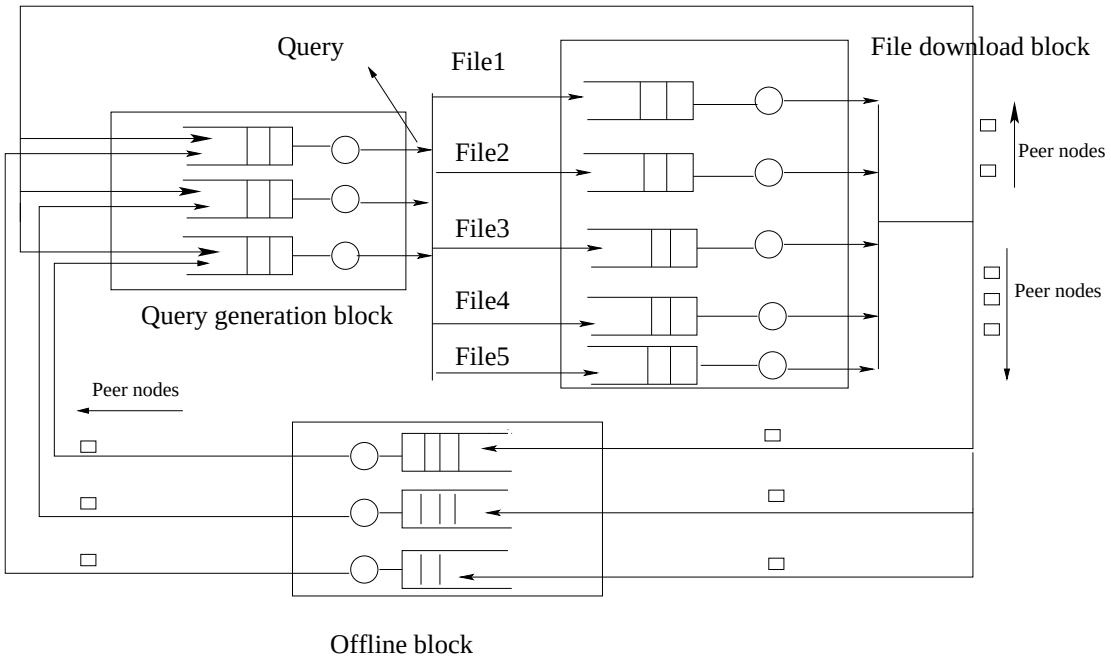
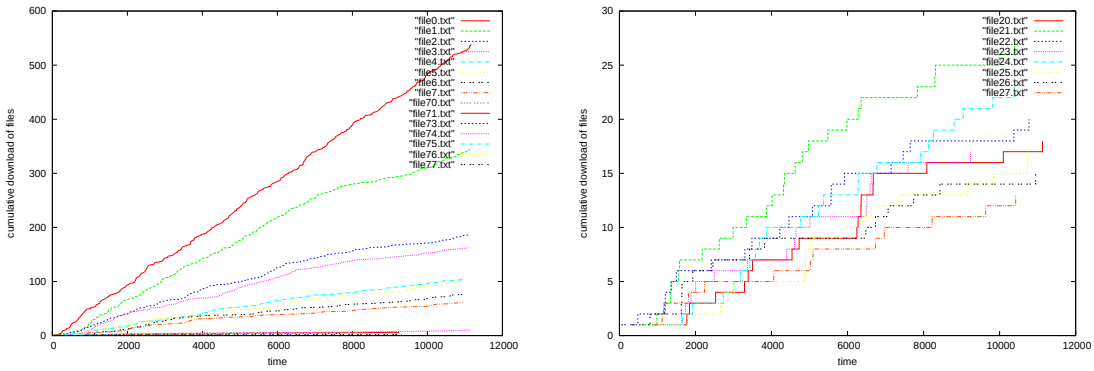


Figure 6.17: Agent based model for p2p traffic

depends on the number of providers available for that specific file (in edonkey, providers send file chunks in parallel to the client peer).

Let us show some simulation model. We implement the agent based traffic generation model with the help of a discrete event simulator. We fix the service rates of the servers in the offline and online blocks following power law distribution with exponent α and β respectively. Fig. 6.18 shows the cumulative number of downloads for the highly popular, moderately popular and highly unpopular files. Fig. 6.19 shows the observed file popularity and peer activity distribution after $T = 5000$ timesteps.



(a) Popular and unpopular files

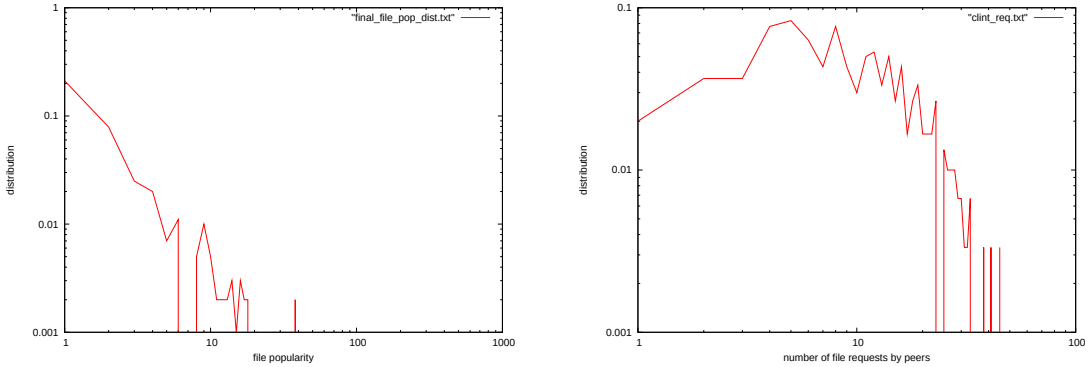
(b) Moderately popular file

Figure 6.18: Cumulative number of downloads for agent based model

6.3.5 Advantages of the Markovian model over agent based model

There are at least two advantages for which the Markovian model must be prefered.

- A significant improvement of the Markovian model over the agent based model is that this model allows all the peers to work (say file download) in parallel rather than in sequential manner. If



(a) File popularity

(b) peer activity distribution

Figure 6.19: Observed file popularity and peer activity distributions

we carefully observe Fig. 6.14 (cumulative download frequency of a popular file $file_{10}$), we can identify some special timesteps, when the $file_{10}$ is downloaded by a group of peers simultaneously. This phenomenon cannot be observed in the agent based model (such as in Fig. 6.18).

- In Markovian model, we can observe plateaus even for the popular files (such as Fig. 6.14) due to the unavailability of the providers. This is interesting since we noticed similar behavior also in the empirical dataset. However, in the agent based model, the plateaus can only be observed for the moderately popular and unpopular files.

Chapter 7

Conclusion

In this deliverable, we presented the results achieved during Task 3.2 as regard the analyses and the mining made on the measurement data. Following Deliverable D3.2, in which we presented both the data obtained through intensive measurement campaigns and the models one could design in order to capture the observed properties, we focused here on extracting as much information as possible in order to validate and provide good comprehension of the different mechanisms observed on Internet.

In particular, compared to the results presented in Deliverable D3.2:

1. We investigated in depth the data obtained with UDP ping in order infer the degree distribution of core routers in order to propose clean samples and validate the method.
2. We performed a rigorous statistical analysis in order to assess the nature of the degree distribution as observed via UDP Ping.
3. We provided evidences of the impact of the underlying topology on the Internet dynamics as observed by the Radar tool.
4. We captured that the dynamics of the Internet routing and forwarding system (through the analysis of routing and forwarding path instability) show different properties. Our analysis shows that the main cause of instability results from the forwarding plane; this corroborates the assumption that the dynamic properties of the forwarding and the routing system are different. Hence, it is impossible to simply derive the one from the other. However, it can also be observed that a second order effect correlates forwarding and routing path instability.
5. We developed a new method (completing the one presented in Deliverable D3.2) for detecting events in time series and we applied it on different dataset in order to validate the approach.
6. We perdomed a careful analyses of a dataset presenting traffic demands and diffusion phenomena at a very large scale and we propose different models able to reproduce the main observed properties.

Although not completely unified, the different models of the Internet and its dynamics along with the different analyses we have performed have proven to bring new perspectives in the domain. They constitute a real progress over the state-of-the-art. Indeed, for the first time, an unbiased measured distribution of degrees of the core Internet at the physical level has been proposed and this measurement campaign has been strengthen by extensive analyses detailed in the present document. The same statement is also true as regard the model of the dynamics of the IP routing topology.

These results will naturally be used in the other tasks of EULER. The new insights on the properties of the Internet topology and its dynamics will be used as input for generating synthetic graphs mimicking the real network structure. This would allow to conduct realistic simulations as foreseen in WP4. Besides, the work made on the dynamics will also feed both WP2 and WP4 work as they will guide the choice of relevant scenarios involving dynamic routing topology and policy and help the consortium as regard the emulation perspective that is planned to be done before the end of the project. This last point will in particular benefit from the work performed on p2p activity. This work will indeed help the definition of realistic scenarios of traffic demand as an alternative to the natural one-to-any and any-to-any scenarios.

Bibliography

[1] A Radar for the Internet – Publicly available datasets. `http://data.complexnetworks.fr/Radar/.10August2011`.

[2] Outskewer source code.

[3] *Time series analysis: Forecasting and control*. Holden-Day, 1976.

[4] Cisco expects p2p traffic to double by 2014. *http://torrentfreak.com/cisco-expects-p2p-traffic-to-double-by-2014-100611/*, 2010.

[5] Dimitris Achlioptas, Aaron Clauset, David Kempe, and Cristopher Moore. On the bias of traceroute sampling: or, power-law degree distributions in regular graphs. *J. ACM*, 56(4), 2009.

[6] Eytan Adar, Li Zhang, Lada A. Adamic, and Rajan M. Lukose. Implicit structure and the dynamics of blogspace. In *World Wide Web Conference Series*, 2004.

[7] Charu C. Aggarwal, Yuchen Zhao, and Philip S. Yu. Outlier detection in graph streams. In *Proc. IEEE 27th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, 2011.

[8] Frederic Aidouni, Matthieu Latapy, and Clémence Magnien. Ten weeks in the life of an edonkey server. In *23rd IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2009, Rome, Italy, May 23-29, 2009*, pages 1–5, 2009.

[9] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *Proc. of STOC '00*, pages 171–180, NY, USA, 2000. ACM.

[10] Leman Akoglu, Mary Mcglohon, and Christos Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD10)*, 2010.

[11] R. Albert, H. Jeong, and A.L. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[12] R.M. Anderson and R.M. May. *Infectious Diseases of Humans: Dynamics and Control*. Science Publications, Oxford, 1991.

[13] Hakan Andersson and Tom Britton. *Stochastic Epidemic Models and Their Statistical Analysis (Lecture Notes in Statistics) (v. 151)*. Springer, 1 edition, July 2000.

[14] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Traceroute Anomalies: Detection and Prevention in Internet Graphs. *Computer Networks*, 52:998–1018, 2008.

[15] N.B. Azzouna and F. Guillemin. Analysis of adsl traffic on an ip backbone link. In *IEEE GLOBECOM 2003*, volume 1, pages 3742–3746. IEEE, 2003.

[16] Tao Ban, Shanqing Guo, Zonghua Zhang, R Ando, and Y Kadobayashi. Practical network traffic analysis in p2p environment. In *Proceedings of the 7th International Conference on Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1801–1807. IEEE, 2011.

[17] A.L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[18] Paul Barford, Azer Bestavros, John W. Byers, and Mark Crovella. On the marginal utility of network topology measurements. In *Internet Measurement Workshop*, pages 5–17, 2001.

[19] Vic Barnett and Toby Lewis. *Outliers in Statistical Data*. John Wiley & Sons Ltd., third edition, 1994.

[20] Alain Barrat, Marc Barthlemy, and Alessandro Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, New York, NY, USA, 2008.

[21] N. Basher, A. Mahanti, A. Mahanti, C. Williamson, and M. Arlitt. A comparative analysis of web and peer-to-peer traffic. In *Proceedings of the 17th international conference on World Wide Web*, pages 287–296. ACM, 2008.

[22] Irad Ben-gal. Outlier detection. In O Maimon and L Rockach, editors, *The Data Mining and Knowledge Discovery Handbook: A Complete Guide for Researchers and Practitioners*. Kluwer Academic Publishers, 2005.

[23] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory (A)*, 24:357–367, 1978.

[24] S. M. Bendre and B. K. Kale. Masking effect on tests for outliers in exponential models. *Journal of the American Statistical Association*, 80(392), 1985.

[25] L. Blunk, M. Karir, and C. Labovitz. Multi-threaded routing toolkit (mrt) routing information export format. *IETF RFC 6396*, October 2011.

[26] B. Bollobas. *Random Graphs*. Cambridge University Press, 2001.

[27] CAIDA. MIDAR antialiasing tool. `http://www.caida.org/tools/measurement/midar/`.

[28] Duncan S. Callaway, M. E. J. Newman, Steven H. Strogatz, and Duncan J. Watts. Network robustness and fragility: Percolation on random graphs. *Phys. Rev. Lett.*, 85:5468–5471, Dec 2000.

[29] E Carlstein. Non-parametric change point estimation. *Annals of Statistics*, pages 188–197, 1988.

[30] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection : A survey. *ACM Computing Surveys (CSUR)*, 41(3), July 2009.

[31] H. Chang, S. Jamin, and W. Willinger. To peer or not to peer: modeling the evolution of the internet's AS-level topology. In *Proc. of IEEE INFOCOM*, 2006.

[32] Qian Chen, Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. The Origin of Power-Laws in Internet Topologies Revisited. In *Proc. of IEEE Infocom*. IEEE, 2002.

[33] Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, December 2002.

[34] A. Clauset, C.R. Shalizi, and M.E.J. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[35] Aaron Clauset, Cosma R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Reviews*, June 2007.

[36] Reuven Cohen, Keren Erez, Daniel ben Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Phys. Rev. Lett.*, 85:4626–4628, Nov 2000.

[37] Reuven Cohen, Keren Erez, Daniel ben Avraham, and Shlomo Havlin. Breakdown of the internet under intentional attack. *PHYS.REV.LETT*, 86:3682, 2001.

[38] Jean-Philippe Cointet and Camille Roth. How realistic should knowledge diffusion models be? *Journal of Artificial Societies and Social Simulation*, 10(3):5, 2007.

[39] Vittoria Colizza, Alain Barrat, Marc Barthlemy, and Alessandro Vespignani. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Sciences of the United States of America*, 103(7):2015–2020, 2006.

[40] EULER Consortium. Measurement-based topology modelling. *IETF Working Document*, December 2011.

[41] Christophe Crespelle and Fabien Tarissan. Evaluation of a new method for measuring the internet degree distribution: Simulation results. *Computer Communications*, 34(5):635–648, 2011.

[42] M.E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *Networking, IEEE/ACM Transactions on*, 5(6):835–846, 1997.

[43] Italo Cunha, Renata Teixeira, and Christophe Diot. Measuring and Characterizing End-to-End Route Dynamics in the Presence of Load Balancing. In *Passive and Active Measurement Conference*, 2011.

[44] Luca Dall'Asta, J. Ignacio Alvarez-Hamelin, Alain Barrat, Alexei Vázquez, and Alessandro Vespignani. Exploring networks with traceroute-like probes: Theory and simulations. *Theor. Comput. Sci.*, 355(1):6–24, 2006.

[45] Weibing Deng, Wei Li, Xu Cai, and Qiuping A Wang. The exponential degree distribution in complex networks: Non-equilibrium network theory, numerical simulation and empirical data. *Physica A: Statistical Mechanics and its Applications*, 390(8):1481–1485, 2011.

[46] Amogh Dhamdhere, Himalatha Cherukuru, Constantine Dovrolis, and Kc Claffy. Measuring the evolution of internet peering agreements. In *Proceedings of IFIP Networking*, 2012.

[47] Amogh Dhamdhere and Constantine Dovrolis. Twelve years in the evolution of the internet ecosystem. *IEEE/ACM Transactions on Networking*, 19(5):1420–1433, 2011.

[48] Moez Draief and Laurent Massouli. *Epidemics and rumours in complex networks*. Number 369 in London Mathematical Society lecture note series. Cambridge University Press, 2010.

[49] David A. Easley and Jon M. Kleinberg. *Networks, Crowds, and Markets - Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[50] A.M. Edwards, R.A. Phillips, N.W. Watkins, M.P. Freeman, E.J. Murphy, V. Afanasyev, S.V. Buldyrev, M.G.E. da Luz, E.P. Raposo, H.E. Stanley, et al. Revisiting lévy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature*, 449(7165):1044–1048, 2007.

[51] P. Erdös and A. Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[52] Alex Fabrikant, Elias Koutsoupias, and Christos H. Papadimitriou. Heuristically optimized tradeoffs: A new paradigm for power laws in the internet. In *Proceedings of ICALP*, 2002.

[53] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251–262, 1999.

[54] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *Proc. of SIGCOMM '99*, pages 251–262, NY, USA, 1999. ACM.

[55] Z. Ge, D.R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley. Modeling peer-peer file sharing systems. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 2188–2198. IEEE, 2003.

[56] Prasanta Gogoi, D K Bhattacharyya, B Borah, and Jugal K Kalita. A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, 54(4), 2011.

[57] R. Govindan and A. Reddy. An Analysis of Internet Inter-Domain Topology and Route Stability. In *Proc. of INFOCOM*. IEEE, 1997.

[58] Mark Granovetter. Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[59] Frank E Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1), 1969.

[60] Jean-Loup Guillaume and Matthieu Latapy. Bipartite structure of all complex networks. *Inf. Process. Lett.*, 90(5):215–221, 2004.

[61] Jean-Loup Guillaume and Matthieu Latapy. Relevance of massively distributed explorations of the internet topology: simulation results. In *INFOCOM*, pages 1084–1094, 2005.

[62] Jean-Loup Guillaume, Matthieu Latapy, and Damien Magoni. Relevance of massively distributed explorations of the internet topology: Qualitative results. *Computer Networks*, 50(16):3197–3224, 2006.

[63] H. Haddadi, D. Fay, S. Uhlig, A. Moore, R. Mortier, and A. Jamakovic. Mixing biases: Structural changes in the AS topology evolution. In *Proceedings of the second international workshop on Traffic and Measurements Analysis (COST-TMA 2010)*, 2010.

[64] Hamed Haddadi, Steve Uhlig, Andrew Andrew Moore, Richard Mortier, and Miguel Rio. Modeling internet topology dynamics. *ACM SIGCOMM Computer Communication Review*, 38(2):65, March 2008.

[65] Douglas M Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.

[66] Victoria J. Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2), 2004.

[67] J. L. Iribarren and E. Moro. Impact of Human Activity Patterns on the Dynamics of Information Diffusion. *Physical Review Letters*, 103(3):038702–+, July 2009.

[68] Matthew O. Jackson. *Social and Economic Networks*. Princeton University Press, Princeton, NJ, USA, 2008.

[69] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc claffy. Transport layer identification of p2p traffic. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 121–134, New York, NY, USA, 2004. ACM.

[70] M. Kivelä, R.K. Pan, K. Kaski, J. Kertész, J. Saramäki, and M. Karsai. Multiscale analysis of spreading in a large communication network. *Arxiv preprint arXiv:1112.4312*, 2011.

[71] Jon Kleinberg. Bursty and hierarchical structure in streams. In *Proc. 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 91–101. ACM, 2002.

[72] Lauri Kovanen, Mrton Karsai, Kimmo Kaski, Jnos Kertsz, and Jari Saramki. Temporal motifs in time-dependent networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(11), 2011.

[73] Amelie Medem Kuatse, Renata Teixeira, and Michael Meulle. Characterizing network events and their impact on routing. In *Proc. of ACM CoNEXT 2007, student workshop*, pages 1–2. ACM, 2007.

[74] C. Labovitz, G.R. Malan, and F. Jahanian. Origins of Internet routing instability. In *Proc. of IEEE INFOCOM '99*, pages 218–226 vol.1, 1999.

[75] A. Lakhina, J.W. Byers, M. Crovella, and P. Xie. Sampling biases in IP topology measurements. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 332 – 341 vol.1, 2003.

[76] Anukool Lakhina, John W. Byers, Mark Crovella, and Peng Xie. Sampling biases in ip topology measurements. In *INFOCOM*, pages 332–341, 2003.

[77] M. Latapy and C. Magnien. Complex network measurements: Estimating the relevance of observed properties. In *Proc. of INFOCOM 2008*, pages 1660–1668, 2008.

[78] M. Latapy, C. Magnien, and F. Ouédraogo. A radar for the internet. In *Proc. of the 2008 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 901–908, Washington, USA, 2008. IEEE Computer Society.

[79] Matthieu Latapy and Clémence Magnien. Complex network measurements: Estimating the relevance of observed properties. In *INFOCOM*, pages 1660–1668, 2008.

[80] Matthieu Latapy, Clémence Magnien, and Raphaël Fournier. Quantifying paedophile activity in a large P2P system. *Information Processing and Management*. to appear.

[81] Matthieu Latapy, Clemence Magnien, and Frederic Ouédraogo. A radar for the internet. *Complex Systems*, 20(1), 2011.

[82] Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie Glance, and Matthew Hurst. Cascading Behavior in Large Blog Graphs. April 2007.

[83] Jun Li, Michael Guidero, Zhen Wu, Eric Purpus, and Toby Ehrenkranz. BGP routing dynamics revisited. *SIGCOMM Comput. Commun. Rev.*, 37(2):5–16, 2007.

[84] Aemen Lodhi, Amogh Dhamdhere, and Constantine Dovrolis. Genesis: An agent-based model of interdomain network formation, traffic flow and economics. In *Proceedings of the IEEE INFOCOM conference*, 2012.

[85] Clemence Magnien, Amelie Medem, and Fabien Tarissan. Towards realistic modeling of IP-level routing topology dynamics. In *http://arxiv.org/abs/1112.4645v1*. arXiv:1112.4645v1, April 2011.

[86] Clemence Magnien, Frederic Ouedraogo, Guillaume Valadon, and Matthieu Latapy. Fast dynamics in internet topology: Observations and first explanations. In *Proc. of the Fourth International Conference on Internet Monitoring and Protection*, pages 137–142. IEEE Computer Society, 2009.

[87] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Xenofontas Dimitropoulos, kc claffy, and Amin Vahdat. The internet as-level topology: Three data sources and one definitive metric. *ACM SIGCOMM COMPUTER COMMUNICATION REVIEW*, 36:17, 2006.

[88] R.D. Malmgren, D.B. Stouffer, A.E. Motter, and L.A.N. Amaral. A poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences*, 105(47):18153–18158, 2008.

[89] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251, 2004.

[90] I. J. Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100, 2003.

[91] M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.

[92] Ricardo Oliveira, Beichuan Zhang, and Lixia Zhang. Observing the Evolution of Internet AS Topology. In *Proc. of ACM SIGCOMM*, 2007.

[93] J.-P. Onnela, J. Saramki, J. Hyvnen, G. Szab, D. Lazer, K. Kaski, J. Kertsz, and A.-L. Barabsi. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, 104(18):7332–7336, 2007.

[94] Jean-Jacques Pansiot. Local and Dynamic Analysis of Internet Multicast Router Topology. *Annales des télécommunications*, 62:408–425, 2007.

[95] Jean-Jacques Pansiot and Dominique Grad. On routes and multicast trees in the internet. *SIGCOMM Comput. Commun. Rev.*, 28(1):41–50, 1998.

[96] Dimitri Papadimitriou, Florin Coras, and Albert Cabellos. Path-vector routing stability analysis. In *13th Workshop on MAthematical Perforformance Modeling and Analysis*, San Jose (CA), USA, June 2011.

[97] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. LOCI: fast outlier detection using the local correlation integral. In *Proc. 19th International Conference on Data Engineering (ICDE)*. IEEE, 2003.

[98] S.-T. Park, A. Khrabrov, D.M. Pennock, S. Lawrence, C.L. Giles, and L.H. Ungar. Static and dynamic analysis of the Internet's susceptibility to faults and attacks. *Proc. of IEEE INFOCOM*, pages 2144–2154, 2003.

[99] S.-T. Park, D. M. Pennock, and C. L. Giles. Comparing static and dynamic measurements and models of the Internet's AS topology. In *Proc. of IEEE Infocom*. IEEE, 2004.

[100] V. Paxson. End-to-end routing behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, 1997.

[101] R. Perline. Strong, weak and false inverse power laws. *Statistical Science*, pages 68–88, 2005.

[102] R. L. Plackett. Karl Pearson and the Chi-Squared Test. *International Statistical Review / Revue Internationale de Statistique*, 51(1):59–72, 1983.

[103] M Raphan and E P Simoncelli. Empirical Bayes least squares estimation without an explicit prior. In *SIAM Conf. on Imaging Science*, Minneapolis, MN, May 2006.

[104] Olivier Renaud and Maria-Pia Victoria-Feser. A robust coefficient of determination for regression. *Journal of Statistical Planning and Inference*, 140(7):1852 – 1862, 2010.

[105] L.E.C. Rocha, F. Liljeros, and P. Holme. Information dynamics shape the sexual networks of internet-mediated prostitution. *Proceedings of the National Academy of Sciences*, 107(13):5706, 2010.

[106] Alessandra Sala, Haitao Zheng, Ben Y. Zhao, Sabrina Gaito, and Gian Paolo Rossi. Brief announcement: revisiting the power-law degree distribution for social graph analysis. In *Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, PODC '10, pages 400–401, New York, NY, USA, 2010. ACM.

[107] Mark J. Schervish. P Values: What They Are and What They Are Not. *The American Statistician*, 50(3):203–206, 1996.

[108] Yaron Schwartz, Yuval Shavitt, and Udi Weinsberg. On the Diversity, Stability and Symmetry of End-to-End Internet Routes. In *Global Internet*, 2010.

[109] Subhabrata Sen and Jia Wang. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Trans. Netw.*, 12(2), April 2004.

[110] Yuval Shavitt and Udi Weinsberg. Topological trends of internet content providers. In *Proceedings of the Fourth Annual Workshop on Simplifying Complex Networks for Practitioners (SIMPLEX)*, 2012.

[111] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger. Sampling techniques for large, dynamic graphs. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–6. IEEE, 2006.

[112] Daniel Stutzbach, Reza Rejaie, Nick G. Duffield, Subhabrata Sen, and Walter Willinger. On unbiased sampling for unstructured peer-to-peer networks. *IEEE/ACM Trans. Netw.*, 17(2):377–390, 2009.

[113] Jun-ichi Takeuchi and Kenji Yamanishi. A unifying framework for detecting outliers and change points from time series. *IEEE Trans. on Knowl. and Data Eng.*, 18(4):482–492, April 2006.

[114] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. Network topology generators: degree-based vs. structural. In *Proc. of ACM SIGCOMM*, volume 32, October 2002.

[115] Nicholas Valler, Michael Butkiewicz, B. Aditya Prakash, Michalis Faloutsos, and Christos Faloutsos. Non-binary information propagation: Modeling BGP routing churn. In *IEEE INFOCOM Workshop on Network Science for Communication Networks (NetSciCom)*, 2011.

[116] A. Vazquez, B. Racz, A. Lukacs, and A.L. Barabasi. Impact of non-poissonian activity patterns on spreading processes. *Physical review letters*, 98(15):158702, 2007.

[117] X. Wang and D. Loguinov. Wealth-based evolution model for the internet AS-level topology. In *Proc. of IEEE INFOCOM*, 2006.

[118] Masufumi Watari, Atsuo Tachibana, and Shigehiro Ano. Inferring the origin of routing changes based on preferred path changes. In *Passive and Active Measurement Conference*, 2011.

[119] Chris Wild and George Seber. The wilcoxon test. `http://www.stat.auckland.ac.nz/~wild/ChanceEnc/Ch10.wilcoxon.pdf`.

[120] Walter Willinger, David Alderson, and John C. Doyle. Mathematics and the internet: A source of enormous confusion and great potential. *Notices of the AMS*, 56(5):586–599, May 2009.

Appendix A

Supplementary tests for degree distribution of Internet

This is an addendum to Deliverable D34 of the Euler project. In response to the recommandation R2.3 of the Technical Review Report for the second period of the projet, we use the (non-parametric) Mann-Whitney-Wilcoxon tests to perform further study on the data collected by LIP6 (UPMC partner) regarding the degree distribution of the Internet. They support the conclusions that obtained from the Clauset-Shalizi-Newman test and described in the main document of D34, and indicate that the data is compatible with a power law of exponent around 4.25.

A.1 Context

In the main document of Deliverable D34 (Section 3.2), we seek to test whether the data collected by LIP6 supports the fact that the degree distribution of Internet at the router level is power law, and if so, what is the exponent of the power law.

Remember that if the degree distribution is a power law proportional to $k^{-\alpha}$ for degree k and exponent (or 'slope') α. The measurement according to LIP6 methodology will produce an observed degree distribution proportional to $k^{-\beta}$, where $\beta = \alpha - 1$. This is due to a bias that high-degree nodes are more easily observed (hence observed more frequently) than low degree nodes. As in Section 3.2.4, we therefore test the hypothesis that *observed* degree distribution is a power law, and we seek to estimate β. (See Section 3.2.4 of the main document for more details.)

In the main document, we test this hypothesis following two methods:

- Pearson's Chi squared tests which tests whether an observed frequency differs from a theoretical distribution (in our case, a power law whose exponent was identified using a least-square method). It is designed for random variables taking their values in a finite set, and the size of this set plays an explicit role in the test. Applying it to the case of a degree distribution following a power law requires therefore truncating the distribution by introducing a maximal degree. The value of this parameter is somewhat arbitrary, and affects the results. Besides, other technical difficulties imply that the conditions of applicability of this test may not really be satisfied in our case.

- The Clauset-Shalizi-Newman test [35], that is specifically designed to test the power law assumption on collected data, along with computation of the correct exponent.

The Clauset-Shalizi-Newman is non parametric (in the evaluation of the quality of fit ; it outputs a p-value for the assumption that the data follows a power law at all) and has become a standard in the complex network literature to test power law assumption, despite its relatively high comutational cost. We have applied it to the measured degree distribution, and we have concluded that, while the p-values are not extremely high, they are sufficiently high the forbid the rejection of the power law hypothesis. Besides, the exponent α of the power law would be between 4.2 and 4.4. We had also used the validation part of the Clauset-Shalizi-Newman method to obtain new estimators of the coefficient: we compute the

p-value obtained by that method for a large number of different values of the coefficient, and select that with the largest p-values. The results obtained were also in the range 4.2-4.4.

Nevertheless, since the p-values obtained were not entirely conclusive, it is worthwhile to apply other non-parametric tests from the statistical toolbox to support that conclusion more thoroughly.

Therefore the reviewer's suggestion is excellent, and we investigate a method based on the non-parametric Mann-Whitney-Wilcoxon test as suggested by the reviewer.

A.2 Methodology

The test that we use goes by the name of Mann-Whitney-Wilcoxon, Wilcoxon rank-sum test, or Wilcoxon-Mann-Whitney test. See for instance [119]. It is a version of Kruskal-Wallis test for two samples. It is non-parametric.

Given two samples X and Y drawn from continuous distributions, the null hypothesis being tested is the following: $P(x > y) + \frac{P(x=y)}{2} = \frac{1}{2}$. If we reject the null hypothesis, we therefore reject the possibility that X and Y come from the same distribution. However if we accept the hypothesis, we cannot deduce that they come from the same distribution, unless we have more information on the shapes of the distributions (for instance that the two distributions are the same but for a shift, which can be reasonable in some circumstances, and is still much weaker that Gaussianity), even though it is an argument in that direction.

We use this test in two ways.

A.2.1 Hypothesis testing

We run the Mann-Whitney-Wilcoxon test on the empirical data X against a sample Y that follows a power-law distribution with exponent β, where β is taken from the estimations computed in the main document of the deliverable (Tables 3.1 to 3.4 of Section 3.2.4). The test outputs a p-value that, if high enough, allows to conclude that we cannot reject the assumption that X is indeed distributed according to a β-exponent power law. As said above, accepting the null assumption in Mann-Whitney-Wilcoxon does not formally allow to accept the assumption that X and Y are drawn from the same distribution.

A.2.2 New estimator of the slope β

Our goal here is to find a new way of estimating β. For this purpose, we run for every β in a certain range the Mann-Whitney-Wilcoxon test between the empirical data X and a sample Y hat follows a power-law distribution with exponent β, and compute the corresponding p-value. We then select as estimator the value β yielding the highest p-value.

We should immediately underline a formal limitation of this new estimator. The p-value should be understood with caution, as the concept of p-value is only valid if the two distributions are independent from each other, while here the sample Y is chosen among a family of samples for its highest resemblance with X, which destroys the independence assumption. The 'p-values' should thus here just be considered as a metric of the distance between the sample and the theoretical distribution. This is already commented upon in Section 3.2.4 (page 28) of the main document.

A.3 Validation

In view of the limitations of the methods, we have performed experiments to test their efficiency on toy examples. We choose X as an artificial data set, drawn from a power law distribution of exponent β_0. We then perform the test as described in this section, and we recover the right β_0 with good precision. The results are better and better as we choose a larger and larger sample Y. Nevertheless we see that from a size of 10^4, the improvement is marginal. Indeed the p-value obtained for size 10^5 differs by at most one or two percent.

We also performed the same procedure with exponential X, i.e. distributed as γ_0^k, and test against samples Y drawn from γ-exponential laws, for various values of γ. There again we recover the right value of γ, with good precision.

We also test an exponential law X against power laws for Y. In that case, we still find values of β for which the p-value is quite high. This confirms the limitation above that the p-value only have a relative value: they can only say that one value of β is more suitable than another value, and is not very reliable as a test of the power law assumption itself. In other words, our new estimator based on Mann-Whitney-Wilcoxon test can be seen as essentially a parametric method, where the power law assumption is not reliably tested per se, but only the value of the exponent. Therefore it comes only as complement to the other non-parametric methods (Clauset-Shalizi-Newman in the main document and hypothesis testing in Section A.2.1).

Finally, another limitation of these methods based on Mann-Whitney-Wilcoxon test is that they apply a method for continuous variables on discrete data (degrees). Nevertheless the large range of the discrete variable ensures that the approximation by a continuous variable is acceptable. Applying Mann-Whitney-Wilcoxon tests on basic discrete distributions showed that the results are the one we expect, as soon as the number of samples is reasonably high.

A.4 Results

A.4.1 Consistency of data sets

The data collected by LIP6 is composed of three samples X_1, X_2 and X_3.

As a first run, we run the Mann-Whitney-Wilcoxon test of those samples one against another, and we find that we cannot reject the hypothesis that they come from the same distribution. This is of course good news as they indicate consistency between the three entire data sets:

- for X_1 vs X_2 we obtain a p-value of 0.7181;
- for X_1 vs X_3 we obtain a p-value of 0.4637;
- for X_2 vs X_3 we obtain a p-value of 0.2723;

Remember that in the main document we only test power law hypothesis for degrees higher than five. Indeed it seems that the power law property is not accurate for the low degree nodes. It is customary in the literature to asses the power law property for an interval of values, as exreme values may often show erratic behaviour.

If we repeat the test for the data sets restricted to degrees higher than five, we obtain the following results, which confirm the validity of the tresholding:

- for X_1 vs X_2 we obtain a p-value of 0.7590;
- for X_1 vs X_3 we obtain a p-value of 0.6713;
- for X_2 vs X_3 we obtain a p-value of 0.9050;

A.4.2 Hypothesis testing

Let us now apply our method for testing hypothesis described in Section A.2.1, which uses the Mann-Whitney-Wilcoxon test to assess the quality of the results obtained from the parametric Pearson Chi squared method and the non-parametric Clauset-Shalizi-Newman method in the main document. Remember that those methods were applied on nodes of degrees higher than five.

The test applied to results of Table 3.1 (Section 3.2.4 of the main document) gives:

Experiment	$\hat{\beta}$	p-value
1	3.245	0.8937
2	3.331	0.7832
3	3.276	0.6178
$1+2+3$	3.230	0.7968

Applied to the results of Table 3.2 (Section 3.2.4 of the main document) we see:

Experiment	$\hat{\beta}$	p-value
1	3.386	0.4095
2	3.386	0.5221
3	3.385	0.2219
$1+2+3$	3.392	0.1214

Applied to Table 3.3 Section 3.2.4 of the main document) we see:

Experiment	$\hat{\beta}$	p-value
1	3.29	0.8583
2	3.36	0.6386
3	3.31	0.4686
$1+2+3$	3.26	0.9296

Therefore the these tests tend to support the conclusions of Section 3.5 of the main section of the deliverable

A.4.3 Estimation of β

Let us now apply the the estimator Mann-Whitney-Wilcoxon-based estimator as explained in Section A.2.2, consisting in comparing every X_i against power laws of various exponents β. We test 1000 values of β between 2.8 and 3.8.

Here are the results:

Experiment	$\hat{\beta}$	'p-value'
1	3.265	0.9986
2	3.2906	0.9982
3	3.202	0.9978
$1+2+3$	3.252	0.9977

As an illustration, the p-value obtained as a function of β is plotted on Fig. A.4.3 for $X_1 \cup X_2 \cup X_3$. It should be recalled here that the very high p-values obtained here should not be intepreted as usual p-value, for the reason explained above (we selected the best sample among many samples, which destroys an independence assumption). But it is still interesting in relative value, as indicating quite sharply the most reasonable exponent for a power-law, provided that we believe that the data is indeed distributed according to a power law. The fact that we recover values for β that are very consistent with those obtained with other methods in the main document is one more confirmation of the validity of the conclusions.

A.5 Conclusions

The Mann-Whitney-Wilcoxon-based tests that we performed on the data collected by LIP6 allows to strengthen the conclusions of Section 3.5 of the main document of Deliverable 3.4.

In particular, we could show that

- we cannot reject the hypothesis that the three samples follow the same distribution (Consistency Tests);
- we cannot reject the hypothesis that the the three samples follow a power law with exponent β computed by the methods (Pearson's Chi-squared tests, Clauset-Shalizi-Newman) of the main document (test of Section A.2.1)
- those values of β are in the same range of the β values found by testing many power laws against the samples (estimator of Section A.2.2).

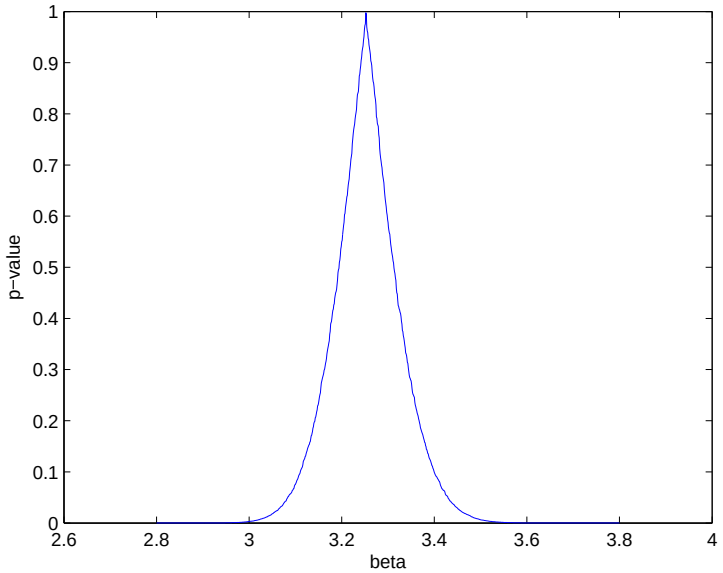


Figure A.1: The p-valued obtained for every β on the union of the three samples $X_1 \cup X_2 \cup X_3$ in the Second Method.

We therefore feel justified to claim that the degree distribution of the nodes appears to follow a power law with exponent $\alpha = \beta + 1$ between 4.1 and 4.4, with 4.25 being the most likely value.

The Mann-Whitney-Wilcoxon-based tests therefore offer a supplementary support to the previous conclusions to Section of the main document of the Deliverable, which were deduced from the non-parametric (but computationnaly expensive) Clauset-Shalizi-Newman method.