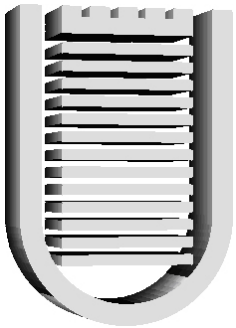
UNIVERSITÀ DI ROMA TOR VERGATA

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria delle Telecomunicazioni



SIMULAZIONE DI TRAFFICO IN UNA RETE DI TELECOMUNICAZIONI A TOPOLOGIA COMPLESSA: ANALISI DEGLI STATI DI CONGESTIONE E DELLA VULNERABILITÀ

Relatore: Prof. Nicola Blefari Melazzi

Co–Relatore: Dott. Vittorio Rosato

Tesi di Laurea Specialistica di:

Fabio Tiriticco

Luglio 2006

Università degli studi di
Roma "Tor Vergata"

SIMULAZIONE DI TRAFFICO IN UNA RETE DI TELECOMUNICAZIONI A TOPOLOGIA COMPLESSA: ANALISI DEGLI STATI DI CONGESTIONE E DELLA VULNERABILITÀ

Contents

List of Figures

Chapter 1

Introduction

The last decade has witnessed the birth of a new research domain related to the study of objects, arising in diverse scientific disciplines (from sociology to medicine, from biology to technology), generically defined as "complex systems".

A "complex system" is a set of physically (or, even, only logically) interconnected elements whose collective behaviour cannot be predicted on the basis of the knowledge of the properties of its constitutive elements. All these systems can be mapped onto complex graphs which have been observed to share peculiar topological properties. Two seminal papers, that by Watts and Strogatz on small-world networks, appeared on *Nature* in 1998 [1], and that by Barabási and Albert on scale-free networks, appeared one year later on *Science* [2], have triggered a flurry of research activities which have seen the physicists community in the front line. Research activities have benefited of the present computational power, which have allowed to produce numerical results on models based on the available data of many "real–world" complex networks [5]: transportation networks, phone calls networks, the Internet and the World Wide Web, the actors' collaboration network in movie databases,

scientific coauthorship and citation networks from the Science Citation Index, but also systems of interest in biology and medicine, as neural networks or genetic, metabolic and protein networks in living organisms.

Complex systems appear as the result of an *unsupervised* aggregation of elements. If this definition is evident when dealing with social networks, it becomes less evident in technological cases such as the Internet or the network formed by the WWW pages. However, also in the latter cases, those networks are the result of `independent, local` growth mechanisms and not of some coherent, large scale, supervised design. They should be considered, after all, as originating by an effective driving force, induced by some selective pressure, aimed at leading those structures to be globally $efficient$ under some point of view. Which is this driving force and how it acts on the systems?

A major result in this domain can be stated as follows. Most of the graphs representing complex networks share an important topological feature, that of being "scale free" (i.e. the distribution of node's degree follows a power law). This means that, independently on the type of system that the graphs represent, their structures have a common property which is probably the first (and more evident) effect that the driving force is able to produce. A number of hypotheses have been raised to explain which are the overall benefits that a scale–free topology is able to introduce. These have allowed to define specific growth mechanisms able to design networks with the same properties or "real" ones.

The effective driving force which, by trials and errors (very much like the genetic selection for living beings) realizes the complex system's net-

works, acts for the fulfillment of two major goals: producing a `robust` and an `efficient` system. Robustness is the property providing the system an high resilience against structural *faults* (i.e. distruction of its components). Efficiency enables, in turn, the system to perform (i.e. to produce a given result) with the least possible workload for all its components. These properties should be also intimately related: it has been demonstrated that protein–protein interaction networks in living unicellular organisms, resulting from hundred of millions years of genetic selection, display a large robustness which manifests in the fact that a *random* removal of nodes (i.e. the elimination of proteins of the network through gene knock–out) does not imply the death of the organism.

The Internet, which is the object of the present study, is a technological system which shares many relevant properties with living systems. It is certainly a growing, unsupervised system whose continuous structural changes are "self–selected" to enhance its effectiveness. Changes are performed locally, without the knowledge of the general plan (or the complete structure) of the system. Internet can be also considered as a part of the "nervous system" of the current civilization and it is a vital and strategic part of it. There are thus many reasons which produced, in the last years, a number of relevant research efforts for the study of the Internet [3]. We will report on these, and on the obtained results, all along this work.

The aim of our effort is related to an even more ambitious goal: that of studying the system generated by the interaction of two or more complex technological systems. It is nowadays evident that technological systems are mutually interdependent; a fault on one systems inevitably affects the oth-

ers, generating the onset of *emerging* and *cascade* phenomena. The large electrical blackouts experienced by Italy and US in 2003 have pointed on the existence and the extent of these interdependencies and have dramatically push forward the need of increasing the knowledge and the control of interdependency–based effects.

The science of complex systems's inderdependency is at its infancy. Complex systems, due to their extreme complexity, are usually modelled through simplified models, attempting to capture their essential features and to reproduce, at least qualitatively, their behavior [5]. The modelling of interacting sets of complex networks does, *a fortiori*, compels the use of simplified models; this helps in attempting to provide some initial insights on the emerging behavior of technoclogical interdependent systems.

This work is a part of a larger work aimed at simulating the behavior of a system composed by interconnecting the electrical and the communication networks. Both these systems have been modelled as `stand alone` systems, by using "realistic", though simplified, models of their functioning. This work reports on the results obtained in modelling one of them, the Internet, while the study of a simplified model of the high-voltage electrical transmission network is an ongoing activity. When the two models will be ready, they will be, somehow empirically, interconnected and the joint behavior analysed.

During this work, however, we had the opportunity of making some reflections on the nature of the Internet, on its growing mechanisms and on the driving forces, finalized to increase its robustness and efficiency. We have noticed that the Internet seems to be subjected to different stimuli: on one side, those which build the structure up (the network) and those which must ar-

range its "intelligence". It seems (this is known) that robustness is very much related to the network's topology which comes out from the growing mechanisms. We collected further evidences, that we will try to present in this work, which point on the inadequacy of the resulting "topological" structure to efficiently support data traffic. For this reason, the Internet, differently from other complex systems where the topological structure `also` guarantees efficiency, must "correct", by other means, the substantial inefficiency generated by its growth process to which it is subjected. In other words, the growth mechanisms which governs the Internet topology produces, *per se*, a structure which is not totally efficient. Man-made intelligence should be introduced to correct this (partial) inadequacy of the growth process to produce an highly efficient test–bed for the flow of the traffic of data.

The plan of this work is thus the following.

In Chapter 2, we define the basic features of graphs and we introduce those complex structure that are later used to represent the Internet network, providing a definitions of the two most important network classes, Random and Scale Free.

In Chapter 3 we analyze "experimental" data, relative to the results of an EU-funded project (DIMES), aimed at defining the worldwide map of the Internet. These data allowed to study its large scale structure and to evaluate its main topological properties.

Chapter 4 is fully devoted to the definition of the "dynamical model" that we have set up to mimic the dynamics and the traffic of packets of data flowing on the network.

Chapter 5 is devoted to summarize the simulation results obtained on the

Internet dynamical model, particularly with respect to the occurrence of a *congestion* phase transition. We also investigated the effects of structural perturbations (in terms of removal of arcs and nodes) on the traffic flow. We have also investigated the effect of strongly localized communications in triggering the onset of the congested phase at different traffic levels with respect to those triggering the congestion in non–localized communications.

Chapter 6 contains the conclusions and an overview of the future perspective in this field of research.

In Appendix A we discuss the technological aspects of the Internet, seen as a network composed by Autonomous System (AS) level routers. These informations will be used to design a basic features of the model of this system in terms of simple interconnected elements and their basic operations.

Chapter 2

The structure of complex networks

2.1 Definitions and notations

Graph theory is the natural framework for the exact mathematical treatment of networks. Graphs are able to map the structure of a raw model of a complex network which, at the lowest level, is described as a set of elements (nodes) phyisically or logically interconnected. The physical or logical interconnections between nodes are the arcs (or links).

The graph we consider to map the Internet network is an *undirected* graph $G = (N, L)$ defined by two sets of elements, N the nodes and L the arcs. A graph must have $N \neq 0$ and L a set of unordered pairs of elements of N. The elements of $N \equiv \{n_1, n_2, ..., n_N\}$ are the nodes (or vertices, or points) of the graph G, while the elements of $L \equiv \{l_1, l_2, ..., l_K\}$ are its links (or edges, or lines).

A node is usually referred to by its order i in the set N. In a undirected graph, each link is defined by a pair of nodes i and j, and is denoted as (i, j) or l_{ij}. The link is said to be *incident* in nodes i and j, or to join the two

nodes; the two nodes i and j are referred to as the $end-nodes$ of link (i, j). Two nodes joined by a link are referred to as *adjacent* or *neighboring*. In a directed graph, the order of the two nodes is important: l_{ij} stands for a link from i to j, and $l_{ij} \neq l_{ji}$. The usual way to picture a graph is by drawing a dot for each node and joining two dots by a line if the two corresponding nodes are connected by a link (see figure 2.1).

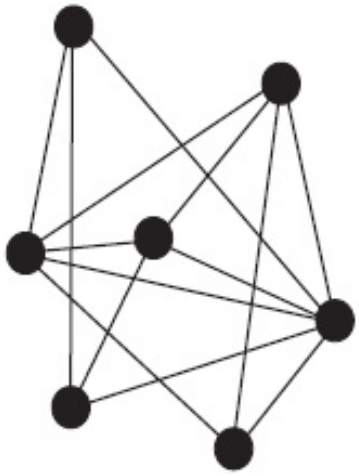


Figure 2.1: an undirected graph example

For a graph G of size N, the number of edges L is at least 0 and at most $N(N-1)/2$ (when all the nodes are pairwise adjacent). A central concept in graph theory is that of *reachability* of two different nodes. In fact, two nodes that are not adjacent may nevertheless be reachable from one to the other. A *walk* from node i to node j is an alternating sequence of nodes and edges (a sequence of adjacent nodes) that begins with i and ends with j. The length of the walk is defined as the number of edges in the sequence. The walk of minimal length between two nodes is known as *shortest path*.

It is often useful to consider a matricial representation of a graph. A graph $G = (N, L)$ can be completely described by giving its *adjacency* (or *connectivity*) *matrix* A, a $N \times N$ square matrix whose entry $a_{ij}(i, j = 1, ..., N)$ is equal to 1 when the link l_{ij} exists, and zero otherwise. The diagonal of the adjacency matrix contains zeros. This is thus a symmetric matrix for undirected graphs. For a comprehensive treatment of these topics, we refer to [6, 7].

2.2 Node degree and degree distributions

The degree k_i of a node i is the number of edges incident with the node, and is defined in terms of the adjacency matrix A as:

$$k_i = \sum_{j}^{N} a_{ij}. \tag{2.1}$$

The basic topological characterization of a graph G can be obtained in terms of the degree distribution $P(k)$,defined as the probability that a node has degree k or, equivalently, as the fraction of nodes in the graph having degree k. Information on how the degree is distributed among the nodes of a undirected network can be obtained either by a plot of $P(k)$ and by the evaluation of the moments of the distribution. The n-moment of $P(k)$ is defined as:

$$\langle k^n \rangle = \sum_{k} k^n P(k). \tag{2.2}$$

The first moment $\langle k \rangle$ is the mean degree of G. The second moment measures the fluctuations of the connectivity distribution, and, as we shall see in further chapters, the divergence of $\langle k^2 \rangle$ in the limit of infinite graph size, radically changes the behavior of dynamical processes that take place over the graph.

According to the probability distribution of the degree, graphs can be classified in several standard models. Among the most important are the Random and Scale Free networks.

2.2.1 Random networks

The systematic study of random graphs was initiated by Erdos and Rényi in 1959 [6] with the original purpose of studying, by means of probabilistic methods, the properties of graphs as a function of the increasing number of random connections. The term "random graph" refers to the disordered nature of the arrangement of links between different nodes.

A random graph can be simply built by adding new nodes linking them randomly to m_0 randomly chosen pre-existing nodes. The resulting degree distribution displays that the nodes degrees are concentrated around a mean value, with little tails on both side. The function that approximates the degree distribution is a decaying negative poissonian exponential,

$$P(k) \sim e^{-\alpha k} \tag{2.3}$$

as shown in figure 2.2.

2.2.2 Scale Free networks

Random graphs were thought to be the underlying structure of almost every unsupervised–grown network in nature. Networks mapping most of ordinary complex systems have been thought, for a long time, to be homogeneous, leading to its classification into the class of random graphs [2]. In fact, in random graphs, each of the $N(N-1)/2$ possible links is present with an equal probability, and thus the degree distribution is binomial or Poisson

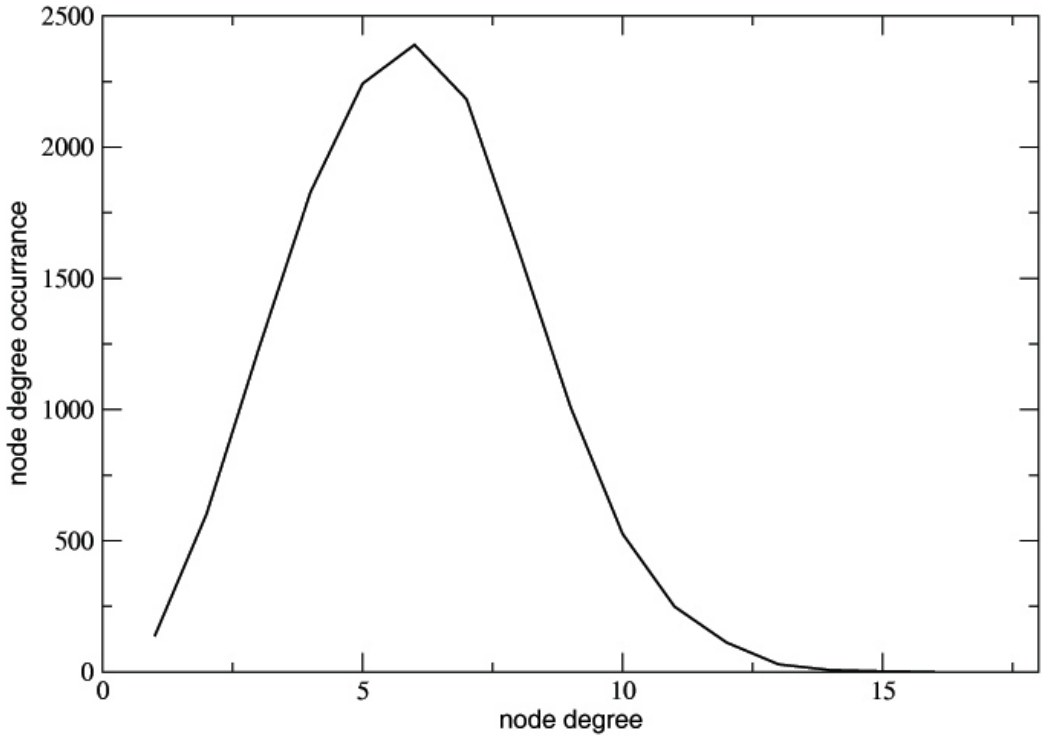


Figure 2.2: degree distribution for a random graph, $N = 14154$, $m_0 = 6$.

in the limit of large graph size. Contrarily to what previously thought, the analysis of the structure of many "real world" networks revealed a much complex degree distribution which cannot be classified at all as random. The degree distribution of most real networks [2, 7] displays a power–law shaped distribution of the type

$$P(k) \sim k^{-\gamma}, \tag{2.4}$$

with exponents varying in the range $2 < \gamma < 3$ depending on the nature of the considered network [2, 7]. The average degree $\langle k \rangle$ in such networks is therefore well defined and bounded, while the variance $\sigma^2 = \langle k^2 \rangle - \langle k \rangle^2$ is dominated by the second moment of the distribution that diverges with the upper integration limit k_{max} as

$$\langle k^2 \rangle = \int_{k_{min}}^{k_{max}} k^2 P(k) \sim k_{max}^{3-\gamma} \tag{2.5}$$

the value of eq.(3.4) diverges in the case $\gamma < 3$. Such networks have been named scale-free networks because power-law has the property of having the same functional form at all scales. In fact, power-law is the only functional form $f(x)$ that remains unchanged, apart from a multiplicative factor, under a rescaling of the independent variable x, being the only solution to the equation $f(\alpha x) = \beta f(x)$. These networks, having a highly inhomogeneous degree distribution, result in the simultaneous presence of a few nodes (the *hubs*) linked to many other nodes, and a large number of poorly connected elements (the *leaves*).

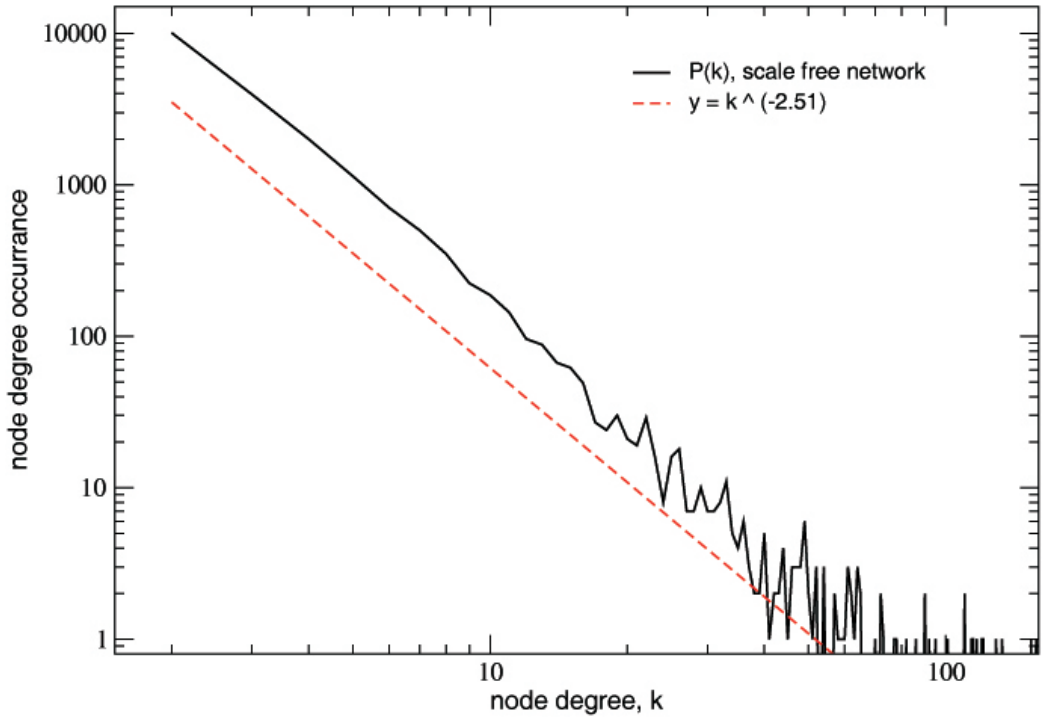


Figure 2.3: degree distribution for a scale free network, $N = 20000$, $m_0 = 2$

From the topological point of view, there are a number of properties which can be simply evaluated from the knowledge of A: these quantities can be used to extract relevant informations on the network and, even more importantly, allow to design growth mechanism able to build up network with topological properties to those of real networks. In the next sections, we will present some of these similarities.

2.3 Clustering

The graph *clustering coefficient* C is a measure firstly introduced in social networks [1]. Clustering, also known as "transitivity", is indeed a typical property of acquaintance networks, where two individuals with a common friend are likely to know each other [8]. It's defined as follows: the quantity c_i (the local clustering coefficient of node i) is first introduced, expressing how likely $a_{jm} = 1$ for two neighbors j and m of node i. Its value is obtained by counting the actual number of edges e_i in G_i (the subgraph of neighbors of i) and then by evaluating the ratio between e_i and $k_i(k_i - 1)/2$, i.e. the maximum possible number of edges in G_i [1, 10]:

$$c_i = \frac{2e_i}{k_i(k_i - 1)} = \frac{\sum_{j,m} a_{ij} a_{jm} a_{mi}}{k_i(k_i - 1)} \tag{2.6}$$

Considering all the nodes in the network, the clustering coefficent of the graph is given by the average of c_i over all the nodes in G:

$$C = \langle c \rangle = \frac{1}{N} \sum_{i=1}^{N} c_i. \tag{2.7}$$

By definition, $0 \leq c_i \leq 1$ and $0 \leq C \leq 1$. Of course, a node i with $k_i = 1$ will have $c_i = 0$.

As showed further in this work, clustering is really an important charachteristic of the network for what concerns robustness and perturbations tolerance.

2.4 Shortest path lenghts, network's diameter, arc's and node's betweenness centrality

Shortest paths play an important role in the transport and communication mechanisms within a network. In our case, suppose one needs to send a data packet from one computer to another through the Internet: the shortest path provides an optimal path way, since one would achieve a fast transfer and save system resources [3]. For such a reason, shortest paths have also played an important role in the characterization of the internal structure of a graph [8, 9]. It is useful to represent all the shortest path lengths of a graph G as a matrix D in which the entry d_{ij} is the length of the geodesic from node i to node j . The maximum value of d_{ij} is called the diameter of the graph (hereafter indicated as d). A measure of the typical separation between two nodes in the graph is given by the average shortest path length, also known as "characteristic path length" $\langle d \rangle$, defined as the mean of shortest path lengths over all couples of nodes [1, 7, 10]:

$$\langle d \rangle = \frac{1}{N(N-1)} \sum_{i,j \in N, i \neq j} d_{ij}. \tag{2.8}$$

A problem with this definition is that $\langle d \rangle$ diverges if there are disconnected components in the graph. One possibility to avoid the divergence is to limit the summation in eq.(3.9) only to couples of nodes belonging to the largest

connected component. An alternative approach, that is useful in many cases, is to consider the harmonic mean of shortest path lengths, and to define the so-called *efficiency* of G as [11, 12]:

$$E = \frac{1}{N(N-1)} \sum_{i,j \in N, i \neq j} \frac{1}{d_{ij}}. \tag{2.9}$$

Such a quantity is an indicator of the traffic capacity of a network, and avoids the divergence of $\langle d \rangle$ since any couple of nodes belonging to disconnected components of the graph yields a contribution equal to zero. However, we will not work with disconnected networks as we want to reproduce a scenario where any node must be able to send packet through the network to any other node.

The communication of two non-adjacent nodes, say j and k, depends on the nodes belonging to the paths connecting j and k. Consequently, a measure of the "relevance" of a given node in the network can be obtained by counting the number of shortest paths going through it, and defining the so-called *node betweenness*, or *node centrality*. The centrality b_i of a node i is defined as [8, 9, 13, 14]:

$$b_i = \sum_{j,k \in N, j \neq k} \frac{n_{jk}(i)}{n_{jk}}, \tag{2.10}$$

where n_{jk} is the number of shortest paths connecting j and k, while $n_{jk}(i)$ is the number of shortest paths connecting j and k and passing through i.

The most common algorithm used to find shortest paths is the Dijkstra algorithm [15], also used in the present work. The concept of *centrality* can be extended also to the edges. The edge centrality is defined as the number of shortest paths between pairs of nodes that run through that edge [7].

These latter quantities will be used extensively in further traffic simulation and analysis, as the edge (node) centrality somehow defines the importance of that edge (node) in the network behaviour. Therefore an edge (node) with a high centrality is more likely to have, in case of fault, a major impact on the network functionality.

2.5 Graph Spectra

The spectrum of a graph is the set of eigenvalues of its adjacency matrix A [16]. A graph $G(N, L)$ has N eigenvalues $\mu_i (i = 1, 2, ..., N)$, and N associated eigenvectors $\mathbf{v}_i (i = 1, 2, ..., N)$. When G is undirected, without loops or multiple edges, A is real and symmetric; the graph has thus real eigenvalues $\mu_1 \leq \mu_2 \leq ... \leq \mu_N$, and the eigenvectors corresponding to distinct eigenvalues are orthogonal.

A further matrix which can be associated to the network is the so-called *Laplacian matrix* L, defined as $L = D - A$ (where D is the diagonal matrix having $D_{ii} = k_i$, with k_i defined as the degree of the i-th node).

Further insights on the structure and the properties of the network can be gained by the spectrum analysis of the A and L matrices [2, 13]. There is a wide literature on the spectral analysis of the A and L matrices [18, 19, 20, 21, 22]. Several authors have pointed out the scaling properties of the eigenvalues [18], while others were interested to extract information concerning the structure of the graph [19].

We focus, in turn, on a specific result derived by the spectral analysis of the L-matrix and the so-called "min-cut" theorem [23, 24, 25]. This can be stated as follows. The lowest eigenvalue μ_1 of the L-matrix is always vanish-

ing ($\mu_1 = 0$) and the orthonormalized components of its associated eigenvector $\mathbf{v}_1^L$ are all equal to $1/\sqrt{n}$ (n being the number of nodes). The components of the second eigenvector $\mathbf{v}_2^L$ of L, associated to the second eigenvalue μ_2 of the Laplacian ($\mu_2 \neq 0$) have, in turn, different signs. The eigenvector $\mathbf{v}_2^L$ of L provides a recipe allowing the partition of the network into two nearly equal sub-networks: the first formed by nodes with a positive component, the second with those with a negative component. The "min-cut" theorem ensures that these two sets of nodes are connected via the minimum number of connections n_l, i.e. the cut has a minimum "weight". In other words, the "min-cut" theorem allows to bisecate the graph into two connected components. Larger cuts (i.e. the removal of a larger number of links) produces the formation of more than two, connected, subgraphs.

We have thus defined n_l as the number of links joining nodes belonging to the different subnetworks (i.e. from the total number of links m, we count only those joining nodes belonging to different sub-networks). The links defined by the "min-cut" algorithm are those whose failure would induce the "most effective" perturbation to the network by producing the maximum number of "effective broken links". This result can also be used for heavy computational problems, as it helps in dividing a problem into two subprocesses with the minum number of interconnection between them [lavoro nostro]. We have introduced n_l as a pretty relevant quantity.

2.6 Growth mechanism for Random and Scale Free networks

A random network can be simply built starting from an initial set of nodes connected between them; each new node will then estabilish m_0 new links with randomly choosen previously existing nodes, prohibiting multiple connections between two nodes [26]. It's been showed that artificially grown random networks are not able to reproduce the topological characteristic of several classes of real networks.

We discuss a class of growth mechanism to reproduce the growth processes taking place in real networks. The rationale is that, by mimicking the dynamical mechanisms that assembled the network, one will be able to reproduce the topological properties of the system as we see them today. We concentrate primarily on the model of network growth proposed by Barabási and Albert in 1999, and on its different variations.

The Barabasi-Albert (BA) model is a model of network growth inspired to the formation of the World Wide Web and is based on two basic ingredients: growth and preferential attachment [2]. The basic idea is that in the World Wide Web, sites with high degrees acquire new links at higher rates than low-degree nodes. In graph terms, a node with a higher degree tends to have a higher probability to receive new connections: "rich gets richer". Starting from a set of m_0 nodes where each $0, 1, ..., m_0 - 1$ node is connected to the m_0^{th}, at each $1, 2, ..., N - m_0$ time step a new node is added to the network and m_0 links are estabilished between the new node ad m_0 pre-existing ones (double connection are prevented). Each pre-existing node has a probability

p_i to receive a new link that depends on its degree and it's calculated as follow:

$$p_i = \frac{k_i}{k_{tot}}, \tag{2.11}$$

where k_i is the node's degree and $k_{tot} = \sum_i k_i$ is the sum of all nodes degrees.

This mechanisms has been used to build the network in figure 2.3; it leads to a network with a degree distribution as in equation 2.4. A sequence of the growth is sketched in figure 2.4. In the limit $N \to \infty$ the model produces a power law degree distribution with an exponent $\gamma = 3$.

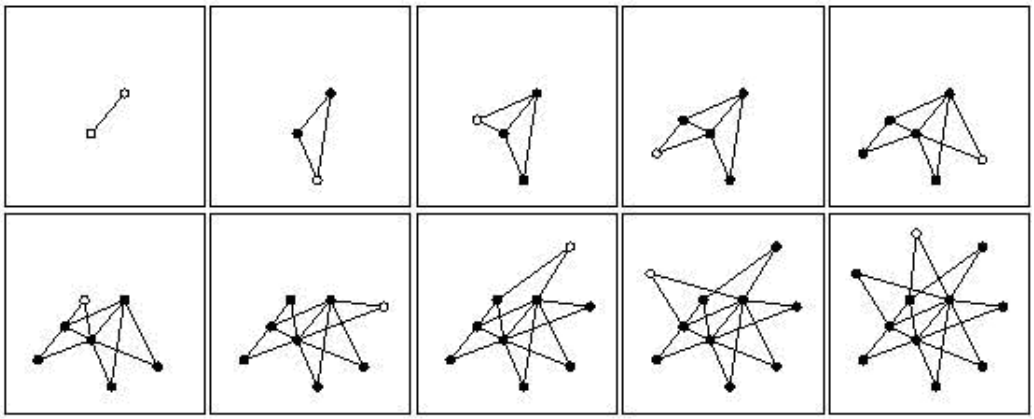


Figure 2.4: sequential growing process of a scale free network, $m_0 = 2$.

We can thus use the random growth mechanism and the BA preferential attachment mechanism to grow Random and Scale Free networks. Table 2.1 reports all the relevant quantity that have been extracted. Relevant differences are evident.

	n	l	m_0	$\langle k \rangle$	$\langle k^2 \rangle$	σ^2	k_{max}
Random	14154	84924	6	12	150.044	6.044	24
Scale Free	14154	84988	6	11.9949	399.787	257.008399	490

	C	d	d_o	$\langle d \rangle$	n_l	γ
Random	$7.91994 \cdot 10^{-4}$	6	0.0024	4.13729	25337	-
Scale Free	$6.25568 \cdot 10^{-3}$	5	1.46969	3.64731	28515	2.51

Table 2.1: Relevant properties of the networks: N is the number of nodes; L the number of links; m_0 the initial node set and new connection per new node; $\langle k \rangle$ is the network's mean node degree (or the degree distribution's first moment); $\langle k^2 \rangle$ is the degree distribution's second moment; σ^2 is the variance of the degree distribution; k_{max} the degree of the network's hub; C is the clustering coefficient of the network; d is the network diameter; d_o is the occurrance of the network diameter between two nodes expressed in percentage over all the node distances; $\langle d \rangle$ is the mean node distance of the network; n_l is the number of links being part of the min-cut set; γ is the scale coefficent as expressed in equation 2.4.

2.6.1 Characteristic topological properties of Random and Scale Free networks

Degree's distribution of the two networks are shown in figures 2.2 and 2.3. The SF network's largest hub's degree is about 20 times bigger than that of the RAN network, but what's most important about the degree is the difference between the respective variances: this parameter is largely influent on dynamical processes that take place over the graph, such as cascade failures [28].

The distances data are also fairly different: in a SF networks, the diameter and the mean distance value between nodes are always lower than in a Random network, leading to shorter path between nodes, as shown in figure 2.5.

The clustering coefficients also show radical difference of nearly a dimen-

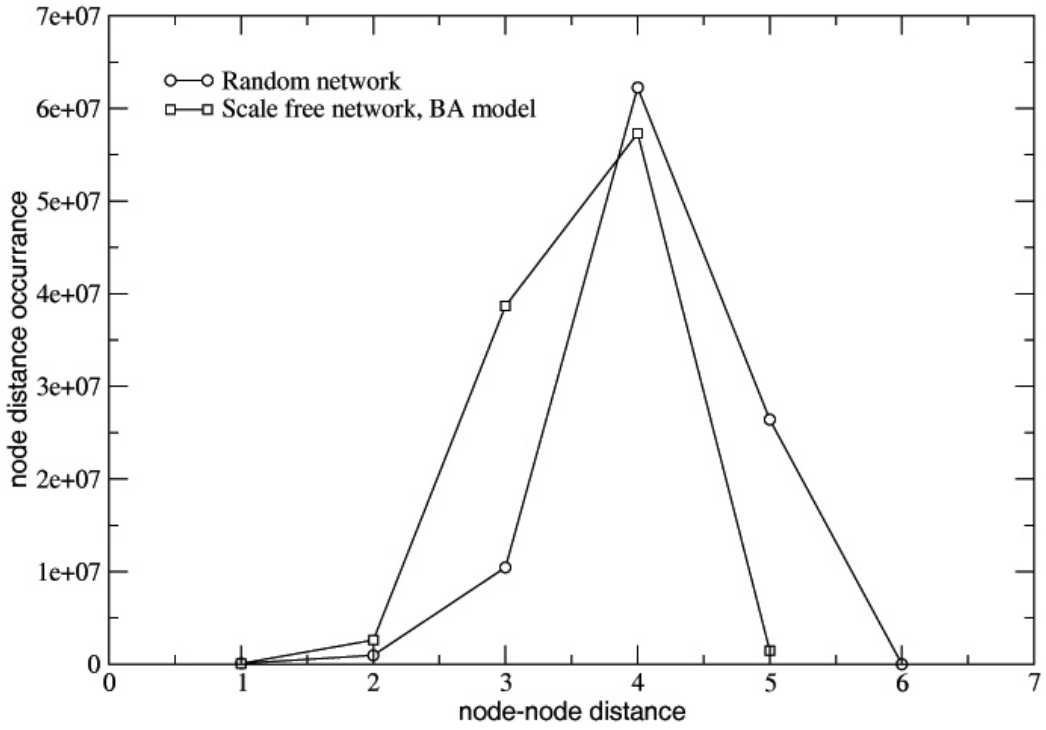


Figure 2.5: distances distributions in random and scale free networks. $N = 14154$ in both networks.

sional order. Clustering has been pointed out by lots of scientific papers to be a key feature in the network robustness (i.e. fault toholerance). Therefore, SF networks appear to be natively more toholerant and robust. Clustering really represents an important graph's charachterization and its value heavily affects the graph response to faults and perturbances; this aspect is crucial in telecommunications networks, as shown further on.

The BA model has attracted an exceptional amount of attention in the literature. In addition to analytic and numerical studies of the model itself, many authors have proposed modifications and generalizations to make the

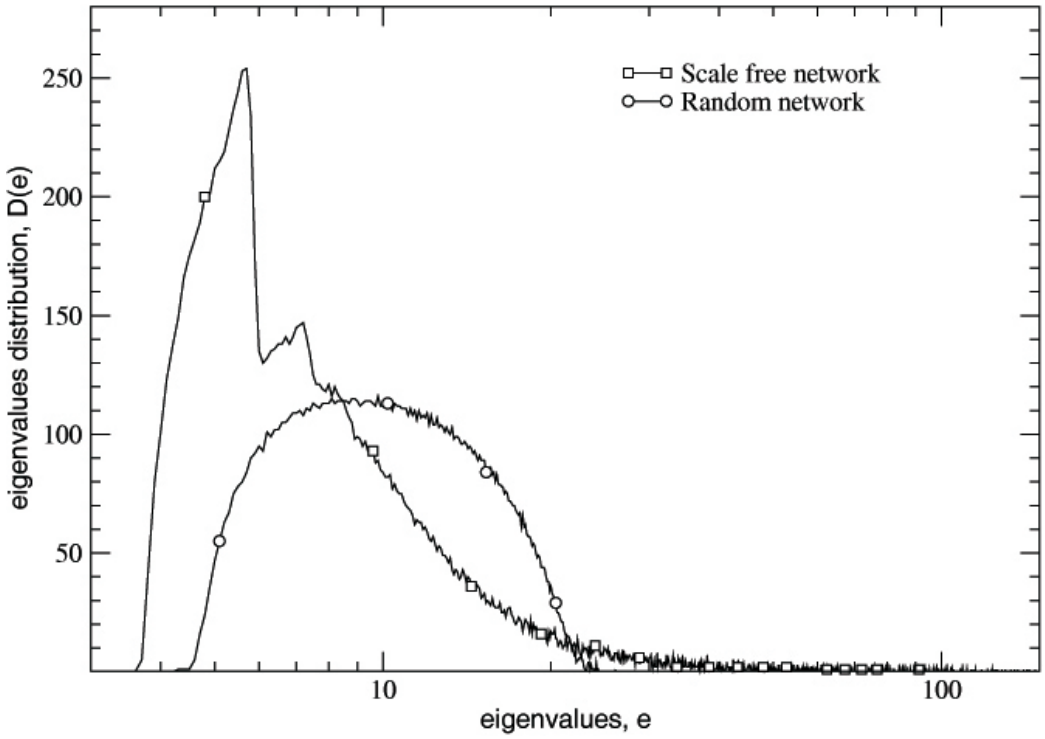


Figure 2.6: laplacian eigenvalues distribution, referred to a scale free networks and a random networks both of 14154 nodes.

model a more realistic representation of real networks [6]. Various generalizations, such as models with nonlinear preferential attachment, with dynamic edge rewiring, fitness models and hierarchically and deterministically growing models, can be found in the literature. Such models yield a more flexible value of the exponent γ. Furthermore, modifications to reinforce the clustering property, which the BA model lacks, have also been considered. The *Triad Formation* mechanism (TF) is an alternative way to preferencial attachment to estabilish a new connection of a new node [27]. Its aim is to increase the number of triangles (triads), thus raising the local clustering coefficent and the average clustering coefficent of the network. Within this mechanism, the first link of a new node is estabilished by the preferential

attachment mechanism with a previously existing node i; further links are ther drawn with nodes that are randomly selected among i's neighbors. An example is shown in figure 2.7.

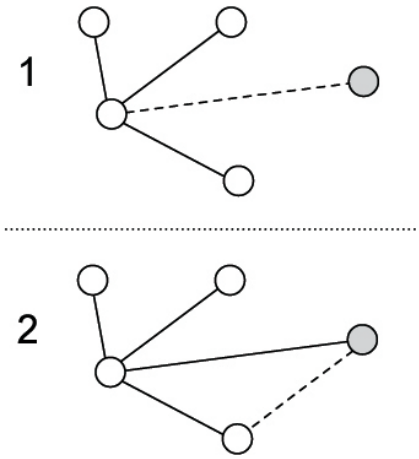


Figure 2.7: example of triad formation mechanism.

Chapter 3

The DIMES network: introduction and topological analysis

As the Internet evolved rapidly in the last decade, so has the interest in measuring and studying its structure. A number of research projects have ventured to capture the Internet's growing topology [29, 30, 31, 32, 33, 34, 35, 36, 37, 38], the delay and bandwidth distributions, with varying levels of success. As the Internet continues to grow, especially far from its North American based core, measurement discrepancies are growing as well. A main handicap of current measurement projects is their rather limited number of measurement nodes, usually a few dozens up to a few hundreds of BGP routers from which explore the "rest of the world".

DIMES [39, 40] is a highly distributed, global Internet measurement infrastructure, with the aim of measuring the structure and evolution of the Internet using a large set of interacting measurement agents. The key shift suggested in DIMES is the move from a small set of dedicated nodes, with measurements as their virtually sole objective, to a large community of host

nodes, running light weight low signature measurement agents as a background process.

3.1 Goals and guidelines

The main reason that lead to this project instead of the usual BGP routers exploration was the BGP feature that allows ISP to communicate only certain paths, hiding others at the same time, due to political or financial strategies (e.g. an ISP may not want to serve as transit AS because there's no financial incentive for this). There could be a link between two ISP that's hided to the rest of the world by omitting it in the path routing information that are forwarded by BGP, maybe because both of them don't want any additional traffic on it. As a result, a researcher collecting BGP announcements from a point outside of the two local ISPs cannot learn about the existence of the local connection. An attempt to learn about this local peer-to-peer connection using traceroute from few measurements points will fail as well from the same reason, and only a presence in, at least, one of the two local ISPs will reveal the peer-to-peer link existence.

Additionally, it's shown [33, 34] that by adding more vantage points, new links are revealed, and that the marginal utility of adding new vantage points decreases fairly fast. What escape these findings is the fact that while the marginal utility decreases, the mass of the tail is very significant (especially in a supposed scale free network topology), thus if one is using a few vantage points, say up to a few tens, there is a small advantage to add a few more, but there is a significant advantage to add additional thousands of points as they will add a significant percentage of new links. Ideally, one would wish

to have a DIMES agent in all ASes, and most of the routable IP prefixes.

The effectiveness of the distributed approach, where lightweight measurement software is hosted by volunteers on computers all over the globe, has been demonstrated by several projects [41, 42, 43, 44] in various contexts, mostly related to computation intensive tasks. For Internet measurements, the contribution of a distributed approach is in the location heterogeneity.

In order to establish a sustainable large community of users the DIMES architecture must follow several guidelines:

- *Security*: being a platform with high degree of flexibility and remote programming abilities poses several serious security risks,such as the potential of hijacking the platform to perform DDoS attacks. Thus, it is of outmost importance to guarantee that the DIMES infrastructure is secured. To do this we do not keep the agent database on our web server, thus even a successful penetration into our web server will not provide the infiltrator with data about our agents. The agents do not expose the host machine to attacks since all the communication between the agent and the server are initiated by the agent.

- *Constrain network resource usage*: as a guest in someone's machine, the DIMES agent must be polite in the way it uses network resources. The network resources usage should follow the well established strategy of distributed computing projects, that is giving the agent the lowest priorities and freeing resources whenever other processes need them.

- *Incentives*: as a system dependent on the good will of people, it is crucial for the success of DIMES to establish incentives which will generate

enough interest to achieve sustainability.

- *Transparency*: to ease privacy concerns, it is important to be as transparent as possible. Thus, the DIMES platform is poised as an open-source platform.

There are several level of granularity at which the Internet topology can be investigated and mapped. At the coarse level each node is an AS, while at the finest level each node represent a router. AS is too coarse a measure, where a node can represent a network that spans a continent or a small metropolitan ISP, while the router level is too fine to achieve a reasonable accuracy. The DIMES goal is to generate a mid-level granularity map where each node represents a group of routers working together, such as a small AS or a PoP of a large or medium size AS, like was suggested in the RocketFuel project [35].

3.2 Architecture

DIMES is mostly written in Java. The two main reasons to choose Java are its natural sandboxing and security mechanisms, and the ease of portability for different operating systems. In addiction, the scripting language PENny has been deployed: it's the language used to create scripts that will be interpreted by the agents. It allows multiple flexible experiment all over the world with the possibility of both local and global synchronization.

Data are stored in MySQL Relational Databases (RDB), as once in an RDB, it is very easy to export the data in any format, even if there are tens of terabytes of data to handle.

DIMES is using http and https as the communication protocol for data and control, respectively. In today's Internet, there are many networks where all other TCP based traffic is being blocked, making other options impossible.

DIMES addresses additional issues of security and privacy by using only secured HTTPs communication between agents and server and by being an open-source platform, thus gaining user confidence and at the same time opening up its capabilities to a larger community. Additional modules are indeed under developement by other groups.

3.3 Data collection

Up to June 1st 2005 about seventy six million measurements were collected, consisting of about sixty million traceroutes and sixteen million pings, from over 3000 agents, spread in more than 350 AS. The first step to be done is to infer IP-AS relationships. In order to translate IP level paths provided from the traceroutes to an AS level topology, one needs to associate IP addresses to ASes. Our current approach for the association process is to mimic a router's decision making process using a longest prefix matching algorithm, which looks for the longest prefix in our database that matches the IP in question. The prefix database, in turn, is built from prefix announcements in BGP data. The resolution process is augmented with a second tier consisting of whois data resolution, which is performed for IP addresses for which the main process has failed. Typically about 2-3% of the IPs fail the longest prefix matching and are resolved using whois. Currently, between 1-1.5% of the IPs fail AS resolution entirely. The translation process is somewhat challenging due to several issues surveyed in [45, 46].

The IP graph from which the Internet topology will be inferred is constructed directly from the traceroutes, where an edge is added for every pair of IPs which are adjacent in a traceroute path. It is important to note that there are many cases where routers do not send ICMP packets when they drop packets or do not answer ping echo requests. In this case an alternative way of mapping is available: the router is identified not by its address (which is unknown in this case) but rather by a combination of his closest neighboring responding nodes. Specifically, it is defined by a triplet of two IP addresses and an index, where the index indicates the number of hops of the unknown router from the former closest neighbor which is responding. However, in the analysis done in this present work nodes (routers) are only associated to responding IP address.

The resulting router graph is finally based on mapping one or more IP addresses (aliases) to a single router and then merging multiple edges between two routers. The current methodology of alias resolution is based on performing a large scale UDP ping survey of all identified interfaces in our IP graph. When an UDP ping probe is sent to IP address **A** from an agent **a**, the router will answer from an interface **A'** which is not necessarily equal to **A**, but in many cases is just associated with the router's interface which is closest to agent **a** or is the default responding interface of the router. This procedure is reproduced from many other agents distributed all over the world, a fact that increases substantially the possibility that no interface of the router will be left unresolved.

3.4 Topologic analysis

The input network has been taken from the DIMES project data repository. They refer to the snapshot of the Internet taken on July 2005. The network's graph contains $N = 14154$ nodes and $E = 38928$ arcs. Available raw data consists in the structure of the network in terms of a graph $G = G(N, E)$, with N nodes and E arcs connecting the nodes, represented by an Adjacency matrix A ($A_{ij} = 1$ if nodes i and j are linked, 0 otherwise). The graph represents the connections presents between nodes (AS-level routers). Arcs represent the physical connections between routers (optical fibers, cables etc.). They are bi-directional, as they allow the flux of data in both directions. All of the topological analysis described in the previous chapter have been applied to the DIMES network, and relevant parameters have been extracted. These are showed in table 3.1, together with the result of an artificially generated scale free networks using the BA model, choosing $m_0 = 3$ which leads to the closest "basic" scale free network, in terms of first moment, to the DIMES network.

Differencies are evident. We will try to capture a general view, considering the different average parameters.

The first unexpected property is the degree distribution $P(k)$ (see section 2.2), that is shown in figure 3.1. The network displays clearly a scale free nature, but the graph also highlights some relevant aspects.

Concerning hubs, the DIMES largest one's degree is about 4 times the degree of an artificially generated scale free network. Hubs are bigger in size: the largest one has a degree of 1977 and the second one is close to the first

	N	L	$\langle k \rangle$	$\langle k^2 \rangle$	σ^2	k_{max}
DIMES	14154	38928	5.50064	1418.24	1393,1759	1977
Scale Free	14154	42453	5.99873	108.742	72.7572	302

	C	d	d_o	$\langle d \rangle$	γ
DIMES	0.412464	9	$9.98394 \cdot 10^{-7}$	3.3426	2.35
Scale Free	$3.44032 \cdot 10^{-3}$	7	0.01030	4.43217	2.6

	n_l	p_l	N_1	N_2
DIMES	675	$1,73\%$	13534	620
Scale Free	10861	$25,58\%$	9010	5144

Table 3.1: Summary of the most relevant topological properties of the DIMES network, compared to those evaluated on a SF network generated by the BA mechanism: N is the number of nodes; N the number of links; $\langle k \rangle$ is the network's mean node degree (or the degree distribution's first moment); $\langle k^2 \rangle$ is the degree distribution's second moment; σ^2 is the variance of the degree distribution; k_{max} the degree of the network's hub; C is the average clustering coefficient; d is the network diameter; d_o is the occurrance of the network diameter between two nodes expressed in percentage over all the node distances; $\langle d \rangle$ is the mean node distance of the network; γ is the scale coefficient as expressed in equation 2.4; n_l is the number of links being part of the min-cut set; p_l is the value of n_l expressed as the fraction over the total number of links; N_1 and N_2 are the number of nodes in the two subsets after the bisection.

($k_{2^{nd}hub} = 1854$); there are four nodes with a degree larger than 1000. At the same time, there's a huge amount of low degree nodes, with 4124 leaves (nodes with degree = 1) and 5020 nodes with degree = 2, which represent nearly two thirds of the network, the 65 % of the nodes.

This has a major influence especially on distance distribution. The DIMES diameter is quite larger than in a scale free network, but at the same time the mean node distance is lower. The two distributions are compared in figure 3.2. How can we explain the coexistence of a larger diameter and a

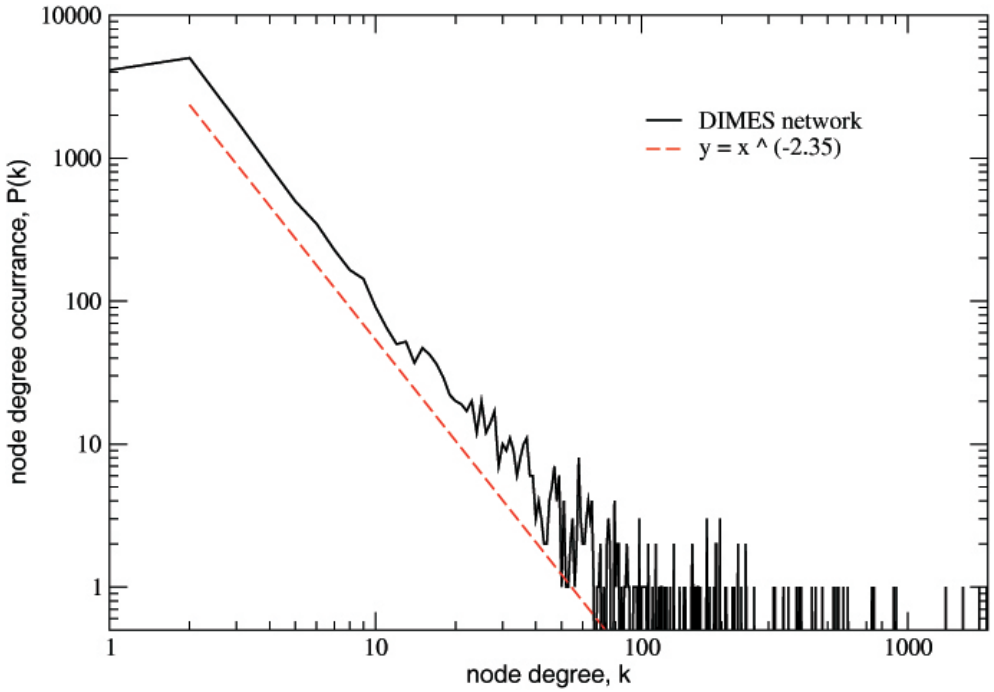


Figure 3.1: DIMES network's degree distribution.

lower mean node distance? The answer is probably in the extremely high cluster coefficient C. As explained in section 2.3, node's clustering is a local property that can be also seen as the indicator of the presence of "triangles". If a node i displays a high clustering, it's likely that its neighbors j and k are connected. Therefore they can directly communicate without a role of i in transferring informations; communications can take place even after the eventual i removal, so the clustering coefficient is related both to network efficiency (lower inter-node distance) and to network robustness. Figure 3.3 shows the relation between the node's degree and the node's clustering, allowing a comparison between the DIMES network and a scale free network. In both cases, high–degree nodes show a low clustering coefficient but, in the DIMES case, the majority of the low–degree nodes show a really high

clustering value, while in the scale free network mid-high degree nodes shows a clustering coefficent even smaller than that of the higher degree nodes. For instance, in the DIMES network there are 3337 over 5020 2-degree nodes (66,5 %) that display $c = 1$; of course the clustering coefficent of a 2-degree node can either be 0 or 1, so in this network there are a huge number of triangles. Considering that the 4124 network's leaves have clustering 0 (by definition) and that they all partecipate in the evaluation of the average clustering coefficient value, the rest of the nodes exhibits a mean clustering coefficient tremendously high.

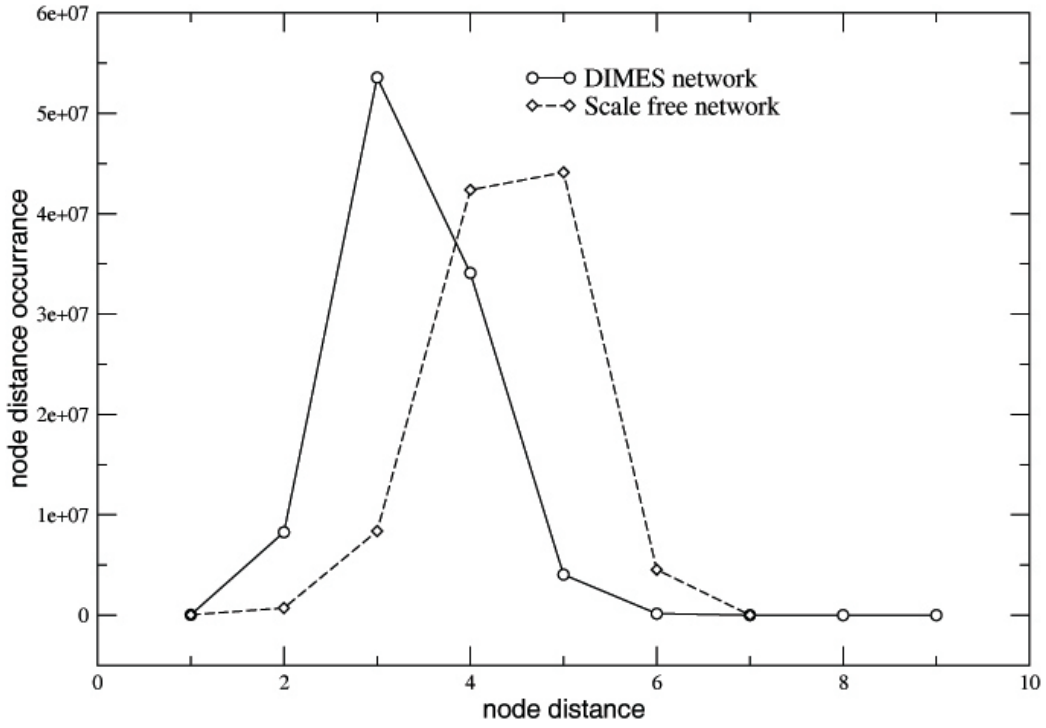


Figure 3.2: distances distribution in DIMES network.

The picture that comes out from the topological analysis of the DIMES network points to its "extreme" scale free character; with respect to the pure scale free networks, generated by the BA model, the DIMES network shows

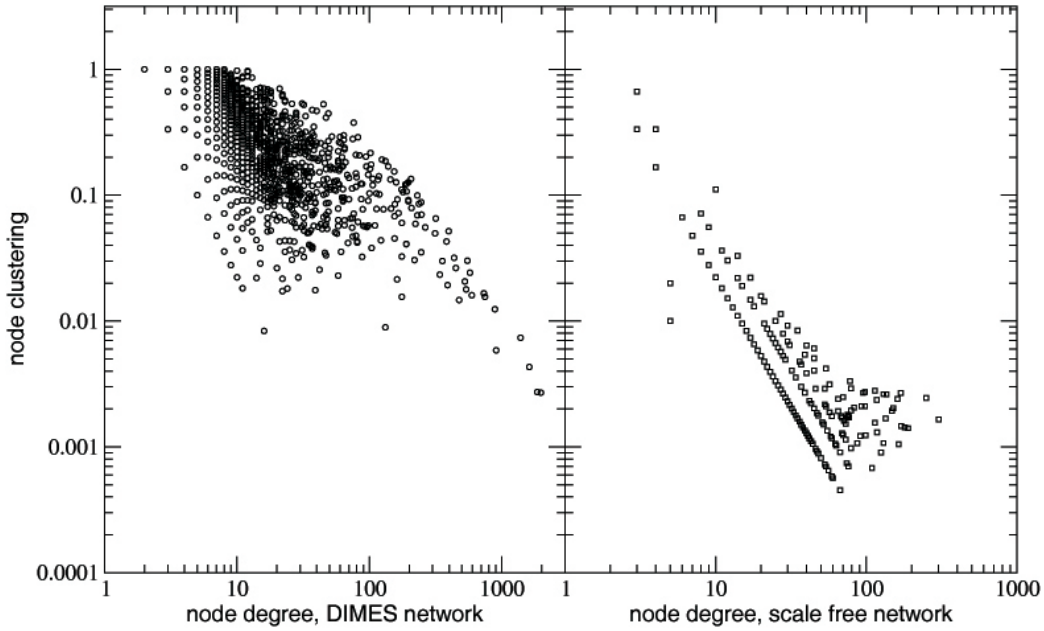


Figure 3.3: nodes degree associated to nodes clustering.

a more "stretched" degree distribution, including both lare hubs and a larger fraction of low–degree nodes. The clustering coefficent C is very high, while diameter d and mean node distance $\langle d \rangle$ are lower.

An important concept that can provide us with further insights for a deeper understanding of the topological nature of the DIMES network is the bisection analysis, as introduced in section 2.5. Aside "global" values of the main topological quantities, showing both the SF character and the large clustering coefficient, our attention has been attracted by three major results of the spectral analysis:

1. the spectrum of the Laplacian eigenvalues of the DIMES network does not look like what expected for a "pure" SF network with low or large clustering coefficient (see fig. 2.6 and fig 3.7)

2. the analysis of the graph bisection (made by the mincut theorem), has shown that the largest possible cut (in terms of links to be removed) allowing a separation of the network into two connected subgraphs is as small as $n_l = 675$, this value is surprisingly low with respect to the total number of links (1.73 %) and to the same quantity referred to the scale free network (10861, about the 26 % of the total number of links).

3. the two subsets are dimensionally very different, while one would expect a roughly similar number of nodes.

Generally, the bisection process produces the division of a network into two, nearly equal, *connected* subnetworks. The number of links to be removed to attain the bisection is the maximum number of links which can be removed to produce the largest possible bisection of the graph. If a further link was removed, the graph would result splitted in more than two subcomponents. In the case, for instance, of a "star" graph (figure 3.4) the solution of the bisection problem is the creation of two subnetworks, one constituted by a simple node; the maximum number of links that can be removed to bisecate the graph is one. If we cut more than one link, the graph will be split into three (or more) subgraphs.

In the DIMES case, the mincut bisecates the graphs into two subcomponents, one much smaller than the other. The analogous procedure applied to SF or Random networks (artificially produced by using the BA and the Random growth mechanisms, respectively) produces cuts separating the networks into two (almost equal) halves, with a very large number of links to be removed.

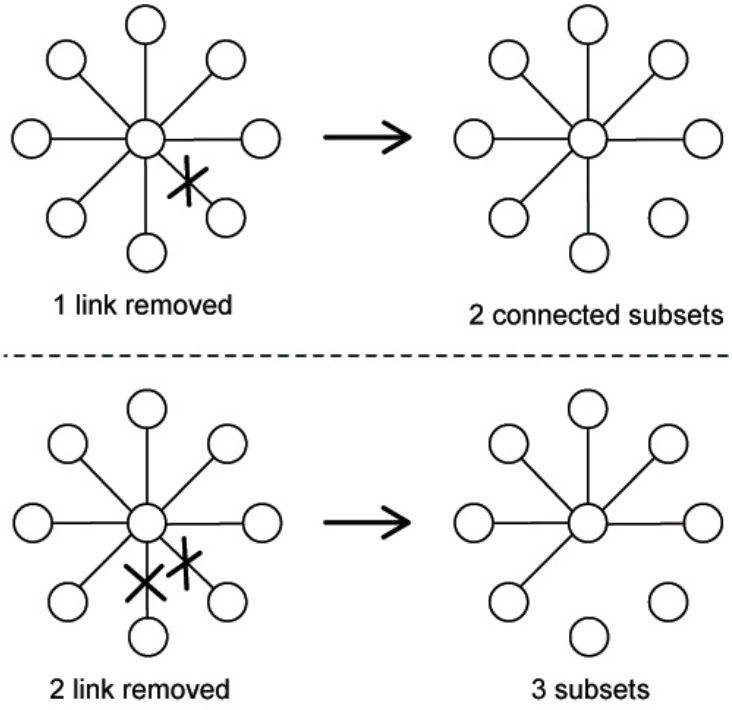


Figure 3.4: solution to the min-cut problem for a star topology small network.

In largest and complex networks, such as those we are analysing, a visual, direct inspection of the graph cannot be used. This makes more complex the the task of producing a clearcut understanding of their structure.

Recalling our previous results of the spectal analysis of the DIMES network, there is a unifying picture able to rationalize them. Let us assume that the synthetic networks (formed by some growth mechanism) are "bulk" objects, in the sense that they have been built by using a single accretion seed (as it really happens). The DIMES network, in turn, might be seen as formed by a lot of "islands" (subnetworks) which are sewed together, mostly because the specific method of data acquisition. If we accept this view, graph's bisection can only take away the largest tile present in the network; a larger number of removed link would imply the separation of the network in more

than two connected components. A further verification of this view is provided by the result of a successive graph's bisection applied to the largest subnetwork resulting from the first bisection. Given the DIMES network, the first bisection produced two subnetworks: the largest one, with $N_1 = 13534$ and the second with $N_2 = 620$. If we repeat the graph bisection on the largest one, we obtain a further bisection which provides two other graphs with $N_{11} = 13030$ and $N_{12} = 504$. This somehow confirms the hypothesis of DIMES as a "tiled" structure.

However, due to the method used to collect data, the picture which emerges from the analysis of the DIMES network puts some limit on the degree of trust that can be accorded to the similarities between the DIMES network structure and the real Internet structure. Despite of the advanced features described in chapter 3, we cannot completely rule out the hypothesis that the Internet is *really* structured as a mosaic made of many different tiles; we just say that the specific mechanism used to collect the data *compulsorily* provides a "tiled", rather than a "bulk", structure. The data acquisition "affects", in some sense, the resulting topology. In fact, DIMES agents act from a given point in the network (i.e. from the inside of a given AS subnetwork) and explore the network "radially" (by expanding its exploration in all directions towards targeted IP). At the end, collected data are sent to the project site, where they are "sewed together" to form the global map. This process allows to progressively form a map, by a continuous insertion of new small tiles which stick, somewhere, to previously inserted tiles. This process, however, tends to form a mosaic, made by small pieces linked all together. What we can aspect is that network sites with a more significant DIMES agents

presence are more finely mapped then those with fewer agents. This issue is indeed highlighted by the DIMES staff [39], and in case this phenomenon is highly effective this can largely reflects on the map's topology.

3.5 Proposed growth mechanism to reproduce the DIMES network

We have ascertained some significant difference between the DIMES network and those generated by the BA growth mechanisms. In the specific case of Internet, the BA model is not able to fully reproduce all the properties of the real network. We thus propose a growth mechanism able to generate "Internet–like" networks with topological properties much more similar to those measured on the DIMES network. We will then use the network produced by this model in dynamic simulations.

The DIMES charachteristics that we have attempted to reproduce in the growth mechanism are

- the higher hubs'degree
- the higher clustering coefficient;
- the larger presence of low degree nodes;
- the lower diameter and mean node distance.

3.5.1 Preferential attachement & Triad Formation

The proposed growth mechanism is based on a suitable combination of the *Preferential Attachment* (PA) and the *Trial Formation* (TF) mechanisms (see section 2.6).

It starts with an initial seed made by m_0 interconnected nodes. When connecting a new node, the first new link is drawn with a suitable modification of the PA scheme: although keeping the proportionality to the degree of the joined node (larger the degree, higher the probability to be joined), the proposed scheme tends to favour the creation of large hubs, as those observed in the DIMES network (table 3.1). The modified PA scheme is such that the probability of the node j to be joined by a new node is

$$p_i = \frac{k_i^\alpha}{\sum_{j=1}^{P} k_j^\alpha} \tag{3.1}$$

where P are the nodes already attached. The remaining links of the new node will be choosen either by this modified PA mechanism or with the TF mechanism, needed to achieve the high average clustering C characterizing the DIMES network. The proposed mechanism prescribes the use of a PA mechanism to select the first node where a new node must be added: then, further links of the new node are connected either with the PA mechanism or to nearest neighbors of the first node, in a way to form "triangles". This growth mechanism allows the choice among the two options with a given probability, which is an adjustable parameter of the model. So, once drawn the first link of a new node, the choice of the TF or the PA mechanism for connecting further links is triggered by a probability value $q(0 < q < 1)$, in a way to compose a growth mechanism G of the following type:

$$G(1) = PA, \tag{3.2}$$

$$G(2, ..., m_0) = (1 - q)PA + qTF \tag{3.3}$$

where the index $1, 2, ..., m_0$ refers to the number of connections of the generic node added to the net. We found that suitable values to reproduce a network

with the DIMES characteristics are

$$\alpha = 1.44,$$

$$q = 0.93.$$

It must be pointed out that these two mechanisms are strictly bounded in their behaviour, and overall non-linear dynamics take place when applied together. The proposed values for the α and q parameters have been discovered by an "empirical" optimization made by "trials and errors".

3.5.2 Variable m_0

DIMES' mean degree is 5.50064; so our model's m_0 can't be expressed by an integer value anymore - as seen in table 3.1, the previous model is only able to reproduce $\langle k \rangle$ values close to integer numbers. At the same time, we needed to reproduce the strong presence of low degree nodes.

We thus introduced a stochastical way to choose the m_i value (i.e. the number of links to estabilish for a new node i); $1 \leq m_i \leq m_0$. m_0 is set to the constant value 6. The following algorithm is iterated over all the n nodes in the network.

1. m_i is randomly choosen between 1 and m_0 with equal probability.

    - if $m_i = 1$, the node will estabilish only a new link and the next 3 (probability 50 %) or 4 (probabilty 50 %) nodes will be added with $m_i = 1$.
    - if $m_i = 3$, 4 or 5, m_i for this given node will be set to 2 with probability of 95 %, otherwise the original 3, 4 or 5 value is used.

- if $m_i = 2$ or 6, the node is attached with this m_i and no constraints are set on next nodes.

2. the next node is considered vith eventual constraints coming from the previous extraction.

The model is quite efficient in reproducing some of the most relevant DIMES topological properties.

3.5.3 Comparisons

Comparison is done between the DIMES network, a pure BA scale free network and a network generated with the model described, which will be reffered to as Scale Free DIMES replica (SFD). Results are shown in table 3.2.

Concerning the first topological paramenters, the clustering coefficient, the mean node degree and the largest hub of the SFD network are quite close to those of the DIMES network. The presence of leaves is quite similar as well (4044) but when it comes to immediately higher degree nodes, we weren't able to reproduce the exact distribution of k, as there's a lack of mid-low degree nodes (from 2 to 8), as shown in figure 3.5.

The node distance distribution reflects the higher weight of hubs and the consistent presence of low degree nodes, thus resulting much closer to the original one. The mean inter-node distance is closer to the original as well. The comparison between the three network is shown in figure 3.6.

The spectral analysis offers some more insights. The eigenvalue distributions of the three examined networks are shown in figure 3.7. The SFD distribution is more similar to the DIMES one than the pure scale free.

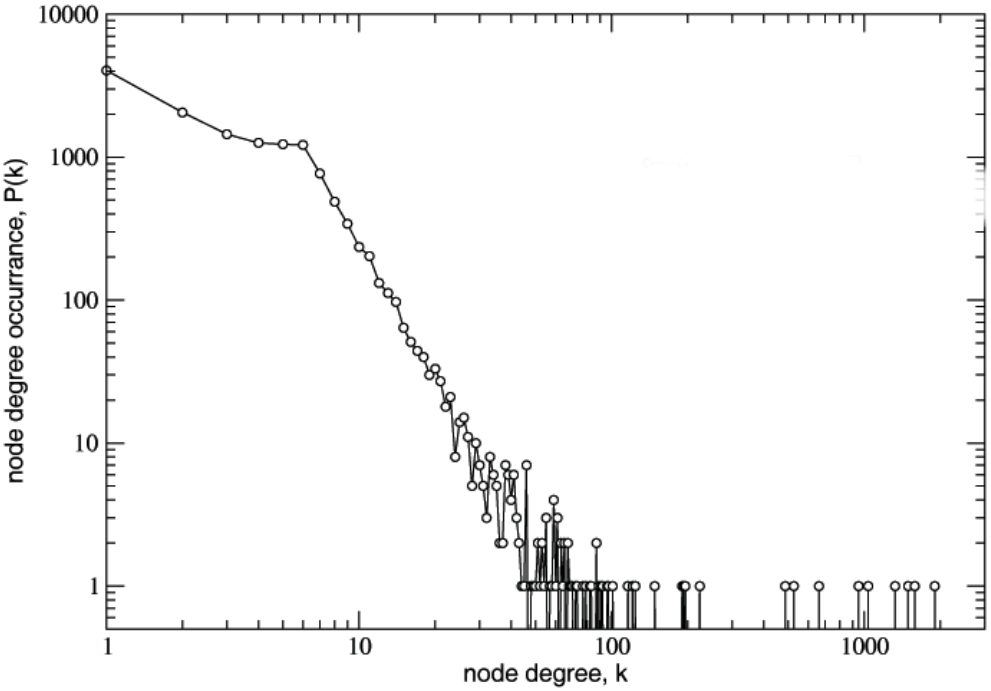


Figure 3.5: node degree distribution for the modified scale free network

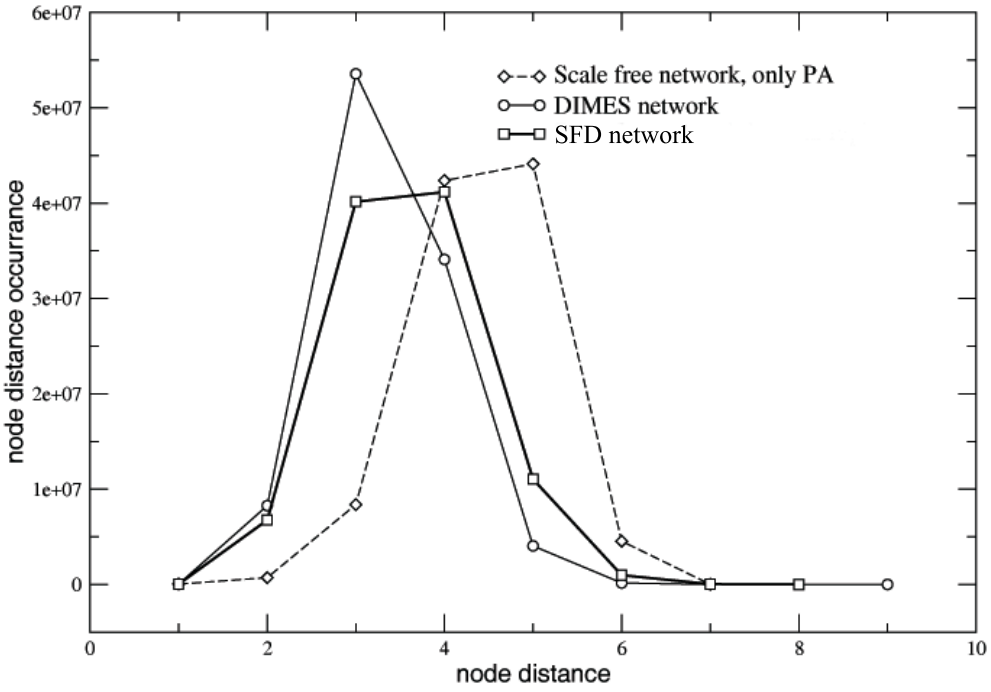


Figure 3.6: distance distribution in the three networks being compared.

	n	l	$\langle k \rangle$	$\langle k^2 \rangle$	σ^2	k_{max}
DIMES	14154	38928	5.50064	1418.24	1393,1759	1977
Scale Free BA	14154	42453	5.99873	108.742	72.7572	302
SFD	14154	38474	5.43564	966.311	936.764	1848

	C	d	d_o	$\langle d \rangle$	γ
DIMES	0.412464	9	$9.98394 \cdot 10^{-7}$	3.3426	2.35
Scale Free BA	$3.44032 \cdot 10^{-3}$	7	0.01030	4.43217	2.6
SFD	0.422705	8	$7.03981 \cdot 10^{-4}$	3.594433	2.95

	n_l	p_l	N_1	N_2
DIMES	675	1.73%	13534	620
Scale Free BA	10861	25.58 %	9010	5144
SFD	3095	8.04%	11815	2339

Table 3.2: Summary of the most relevant topological properties of the DIMES network, compared to those evaluated on a Scale Free network generated by the BA mechanism and to our proposed SFD mechanism: N is the number of nodes; N the number of links; $\langle k \rangle$ is the network's mean node degree (or the degree distribution's first moment); $\langle k^2 \rangle$ is the degree distribution's second moment; σ^2 is the variance of the degree distribution; k_{max} the degree of the network's hub; C is the average clustering coefficient; d is the network diameter; d_o is the occurrance of the network diameter between two nodes expressed in percentage over all the node distances; $\langle d \rangle$ is the mean node distance of the network; γ is the scale coefficient as expressed in equation 2.4; n_l is the number of links being part of the min-cut set; p_l is the value of n_l expressed as the fraction over the total number of links; N_1 and N_2 are the number of nodes in the two subsets after the bisection.

The bisection process of the SFD netork divides it into two rather different subsets, with a lower number of links to be cut with respect to the pure BA scale free network, but still the two subnets are not so different and n_l is not so low as in the DIMES network. As seen in previous works, a high clustering coefficient always tends to decrease n_l: being the network more locally tight, there are less links to be cut in order to separate two internally tight cluster. In this case, though, we must assume that the two high clusterings are of

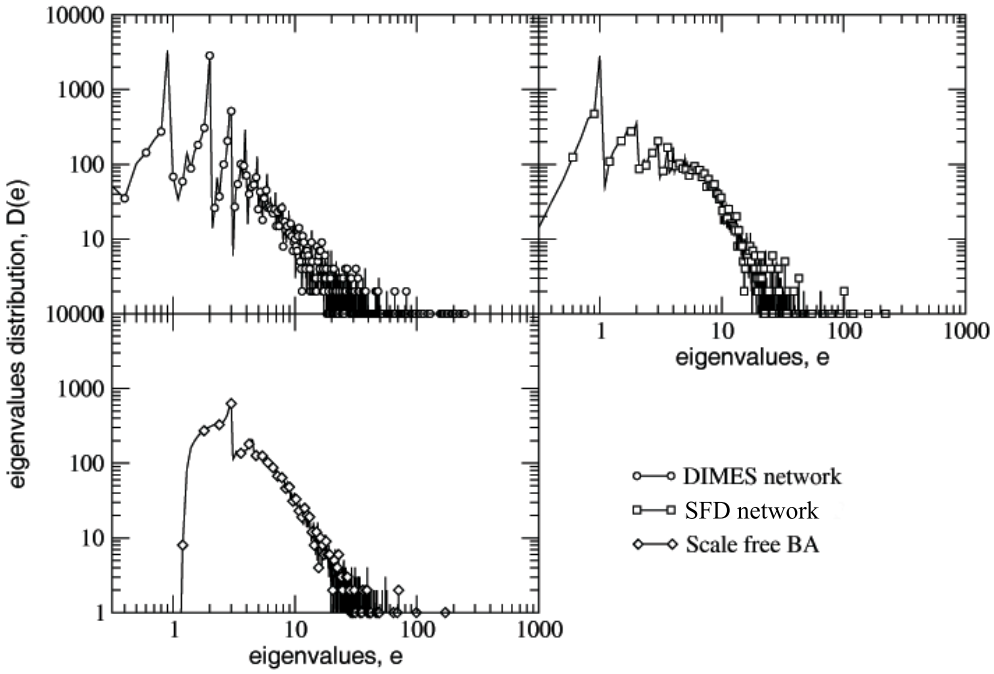


Figure 3.7: eigenvalues distribution in the three analyzed netwoks. SFD refers to the Scale free network - DIMES Replica, the model we introduced.

"different nature", as shown in figure 3.8. In the SFD network, the high clustering isn't due to the low degree nodes as it is in the DIMES one, but rather to the mid-high degree nodes. This implies a fairly higher "general tightness", instead of the small size local tightness that charachterizes the DIMES network with its "islands".

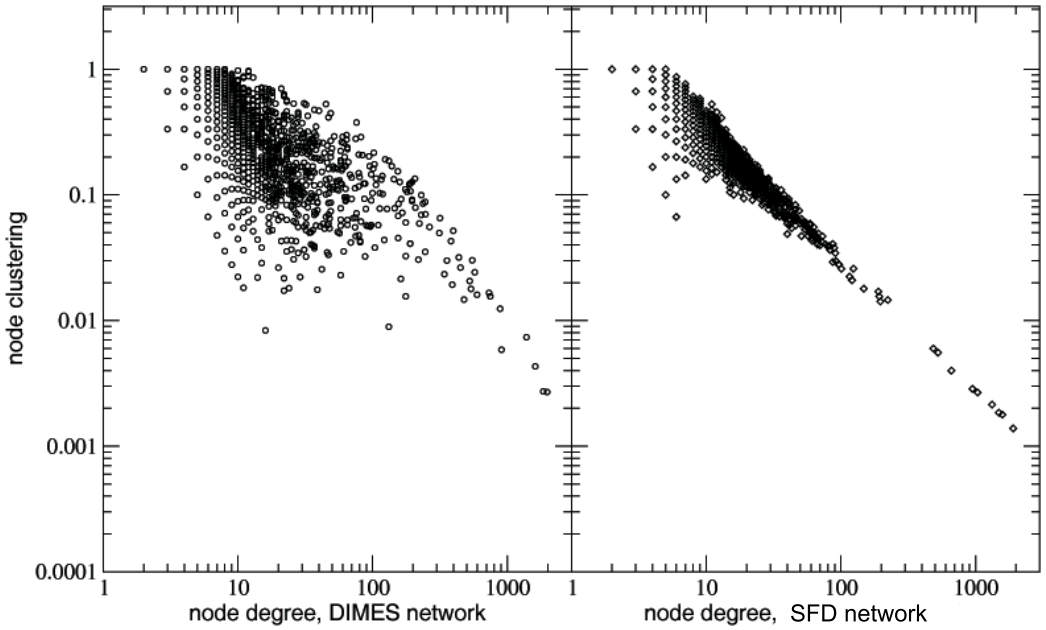


Figure 3.8: relation between node degree and node clustering: comparison between the DIMES network and the SFD network.

Chapter 4

A dynamical traffic model for topology analysis

Aside to the assessment of the properties of the graph representing the topology of the Internet, we have been interested to define a dynamical model of the network, able to reproduce the main features of the network seen as a route where packets of data can be exchanged (sent/received) among the different nodes. This for, at least, two reasons: the first is to evaluate to what extent the topology which the network issues upon its "inevitable" growth mechanisms (producing robustness) is also fit to efficiently sustain the main action of the Internet (that is to host a traffic of packets of data). The second is to see if the topological robustness keeps alive also in presence of the traffic of data. This will be done by perturbing the network (by removing nodes or arcs) and to evaluate the change of the traffic behavior.

The dynamical model we have designed assumes that the network's nodes represent AS-level routers (see appendix A), which are thus characterized by properties and functions typical of those realized by routers. The dynamical model should thus be able to reproduce the traffic of packets of data, gener-

ated, with a given frequency, by a node and directed to an other node of the network.

In order to check the very influence of the network's topology on traffic, we have designed a dynamic model for the traffic flow enough *simple* to avoid a deep superposition between network's intelligence and network's structure. To this aim, we have not introduced in the model all the "actions" aimed at mitigating the effects of the network's topology. These actions will be briefly reviewed in Section 4.2.

Our model is thus composed by two parts: the topological network, represented by a graph of N nodes and L links, and the traffic simulation model that runs on the network, which is presented in this chapter. For the first part, we chose the DIMES structure as reference, that probably represents the most reliable map of the Internet available today (see chapter 3). Moreover, we have developed a growth mechanism able to capture the most relevant topological properties of DIMES (see subsection 3.5). This model has been used to generate networks, topologically similar to DIMES but with a smaller number of nodes. These networks have been used in some cases where the use of large networks would have produced an untractable computational complexity.

4.1 Description of the model of traffic dynamics

Each node of the network represents an AS–level router; a link represents a connection between two routers, along which data can be exchanged in data unit referenced to as *packet*. Routers are identified by an *identification*

number (ID) i, $(0 \leq i \leq (N-1))$, where N is the total number of routers in the network.

The dynamical evolution in time is discretized in *Time Step(s)* (TS). At each TS, a router can send a packet to another router under a certain probability. A node cannot send a packet to itself. The amount of traffic present in the network is measured by the variable $\lambda(0 \leq \lambda \leq 1)$ which measures the frequency with which a node emits a packet (or, equivalently, the fraction of nodes that, at a given TS, emits a packet). The packet is generated by a randomly chosen "emitting" node and directed to a randomly chosen "destination" node. With this definition, for instance, $\lambda = 0.1$ represents a level of traffic where, at each time step, 10% of the N nodes of the network generates a packet of data directed toward an equal number of destination nodes. The two sets of nodes could be similar or different; the only forbidden action is that of a node sending a packet to itself.

Each router hosts a routing table (RT) where information about the next hop to reach each other router in the network is stored. The RTs present on each node are evaluated, once forever, via the *Dijkstra* algorithm [15]. The strategy which rules packet dispatching is based, in fact, on the shortest path between sender and receiver: each node's pair (origin-destination) is associated to a routing path which is formed by the nodes joining the minimal–distance path between the two nodes.

In the present model, at a given TS, a node can send *only* a data packet but, in turn, can receive as many packets as needed (i.e. if more than one neighbor sent it a packet at the previous TS). In order to host multiple packets arriving simoultaneously and to dispatching them at later times, each node

has an infinite–size buffer, where packets are stored. When a packet arrives to a router, two cases are taken into account:

- if the router is the packet's destination, the packet is eliminated from the network - it is supposed to be forwarded by an *intra-AS* routing to the destination host within the AS represented by that router.
- otherwise, the packet is stored in the last position of the router's buffer where it waits to be delivered.

The buffer management complies with the *First In - First Out* (FIFO) policy: at each TS, each router picks the first packet (in order of arrival) from its queue and forwards it according to its routing table. All the remaining packets get ahead in the queue. Several aspects need to be pointed out:

- two or more packets can reach the router at the same TS; in case one or more have not reached the destination, they are all stored into the buffer and they will be ordered according with the provenience router (lowest ID, higher yield).
- each link between two routers is traversed in a TS: hence, every packet hops from a router to the next in the same TS.
- a packet cannot traverse more than one link in a TS.
- a router can forward only a packet from its queue in a TS.

Once a packet is emitted, it is immediately put into its emitter's buffer; in case this buffer is empty, the packet is forwarded during the same TS it has been generated.

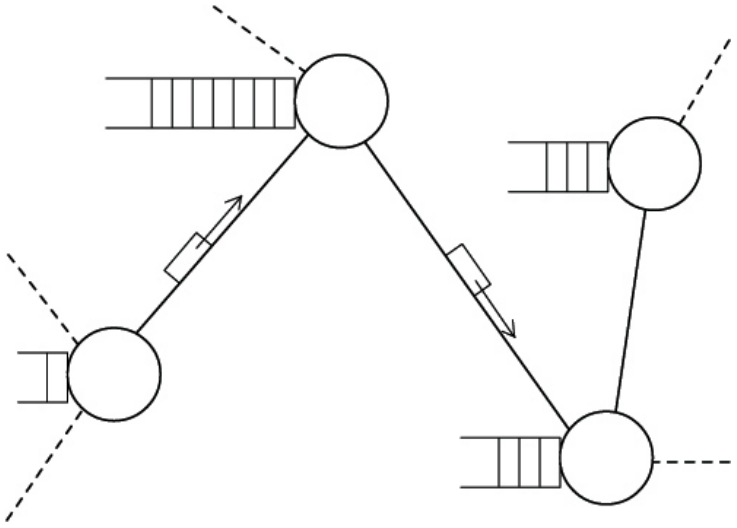


Figure 4.1: visualization of routers with their respective buffers and packet being sent.

Simulation starts with an "empty" network, i.e. with no traffic and all buffers empty. As soon as simulation progresses, buffers start filling and emitted packets take a certain time (i.e. a certain number of simulation TS) to reach the destination node. In the best possible case, the packet is re-emitted by a node the TS after its arrival to that node. However, if the buffer of that node is already filled by packets arrived at earlier times, the packet must remain, on that node, a number of TS equal to the number of packets waiting in the buffer, according to the FIFO policy.

4.1.1 Routing

When a packet must be forwarded, the next router is selected so that the packet is delivered to its destination along the shortest path. If more than one candidate to the next hop exists, a strategy is needed to select the recipient. In our simulations we consider three routing strategies: the `fixed routing`,

the `deterministic routing` and the `probabilistic routing`.

The difference is based on the number of paths that routers take into account and on the way they choose on which one they will forward a packet.

In the `fixed routing`, each router has only one possible neighbor node associated to a given destination node. In other words, the RT of node i associates, to each destination node j, one and only one node, say k belonging its neighbors.

In the other two strategies, more than a single shortest path can be selected. In case more than one shortest path exist for a couple of nodes (i, j), the routing table of i will have two choices (i.e. two links) associated to the node j. Note that more than two choices could exist: we found that for an artificially generated 3000 nodes Scale Free network - DIMES replica (see section 3.5) the average number of possible choices for each couple of nodes is 1.58721, while for a Random network of the same size is 1.89529.

How does the router choose the node where the packet must be forwarded?

In order to introduce the `deterministic routing` and the `probabilistic routing`, we must introduce a "routing probability function" [47]. When a router has to forward a packet choosing between two routes A and B based on the destination address, we assign the probability to choose A and B by the following equation:

$$P(A) = \frac{e^{-\beta X_A}}{e^{-\beta X_A} + e^{-\beta X_B}}, \tag{4.1}$$

$$P(B) = \frac{e^{-\beta X_B}}{e^{-\beta X_A} + e^{-\beta X_B}}, \tag{4.2}$$

$$P(A) + P(B) = 1, \tag{4.3}$$

where β is an adjustable parameter, X_A and X_B are the number of packets that have already traversed the routes A and B, representing thus the usage of the link. In our mechanism, we evaluate these probabilities by evaluating firstly $P(A)$ and then $P(B) = 1 - P(A)$. We can use the parameter β to drive the probabilities.

- if $\beta \to 0$, both of the exponential terms in the equations 4.1 and 4.2 tends to 1 and both $P(A)$ and $P(B)$ tends to 0.5. In such a case the route is randomly choosen with no relevance accorded to the usage of the two links. This is what we call `probabilistic routing`.

- If $\beta \to 1$, the usage of the routes is more important in the evaluation of the two probabilities, making bigger the probability of the less used link. This is what we call `deterministic routing`.

- If $\beta \to \infty$, $P(A)$ always tends to 0, thus recalling the `fixed routing` by always choosing the route B.

The variation of β between 0 and 1 determines the degree of randomness of the routing, and even if $X_A > X_B$ there's a probability of routing a packet through B; this probability decreases as the difference of usage between A and B increases. Figure 4.2 shows an example of routing where a decision must be taken.

4.2 Neglected mechanisms of real communication inter-networks

Throughout this work, as stated in the introduction of this chapter, all efforts have been finalized to the assessment of the impact that the topology of

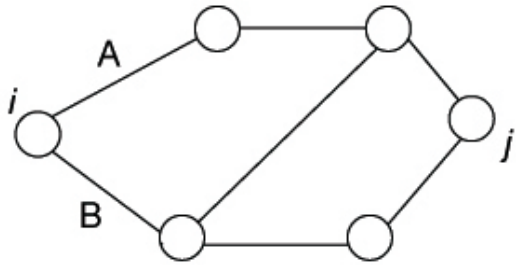


Figure 4.2: example of possibility of different routing: three different shortest paths exist between nodes i and j.

the Internet network has on its functionality: the proposed dynamic model complies with this idea and has been designed to contain only *basic* routing mechanisms androuter features. In the following section, however, we provide a list of the main differences between our model and the real Internet intelligence, discussing the Internet's features which have been neglected in the design of the present model. They are often key features of huge relevance in the Internet functioning behaviour and in its evolution process over time. See appendix A for all the definitions.

4.2.1 Links

Referring to the topology scheme, we do not consider differences between links: an arc of the graph here represents only a "connection" between two routers, regardless of its nature (coaxial cable, optic fiber, radio channel...) that in real world implies strong consequencies. For instance, the bandwith is extremely variable, implying that packets will take different "link traversing time" on account of that. In this model packet are of "infinitesimal" size as they only and always take one discrete TS to being transferred from a router to another. The *error ratio* is also a crucial point that depends on the physical

medium data are transferred on. In this simulation we do not take errors into account: a packet is assumed to always hop from router to router without risking of being lost or corrupted. Moreover, arcs are always considered bi-directional, so communication in both directions is allowed on the same link, whereas in the real Internet this assumption cannot be considered true.

4.2.2 Routers

A router is here intended as an inter–exchange node, and no differences between routers are considered: each one is considered to be connected to an AS which, in turn, is only connected to that router, thus there are no routers that only forward packets as in figure A.3. These "core routers" in real world are usually more powerful in terms of packet handling and forwarding than a general subnetwork router.

4.2.3 Packets

A packet represents an ideal data unit that is exchanged between two nodes in the network. There is no difference between them, so each packet is treated in the same way: no mechanism of *Quality of Service* (QoS) is implemented and the type of service that packets carry is not relevant at all. In real Internet this is not a strong issue as well, but in the telecommunication field differences can be set between packets that need different responses from the network, especially in terms of delay.

4.2.4 Routing protocol

Out routing table is evaluated once forever via the Dijkstra algorithm, where the complete knowledge of the network topology is assumed; it is a sort of OSPF routing. The most diverging point, with respect to the real Internet, is that we use a slightly adaptative routing to select the route (in the case of `deterministic` and `deterministic` routing). Routing Tables are static, in the sense that they are never updated. On one hand there is not any strict necessity for doing this as new links are never introduced nor removed. On the other hand there is not an alternative metric to the strict topological distance, as it could be for instance a metric based on *round trip times* (RTT) that could examine different paths in order to choose the less traffic loaded.

4.2.5 Traffic limiting controls & packet management

This is probably the most relevant issue that marks the difference between the real Internet traffic flow and our model. As seen in chapter A, the Internet relies on several mechanisms that prevent traffic from overloading all resources. The most important ones are briefly summarized, with respect to the difference in our model.

- **Router buffers size:**

  our routers'buffer size is unlimited: each packet that reaches a router is stored in the router's queue and no packet is ever discarded. This leads to diverging lifetimes, where a packet spends nearly the 100% of its lifetime in a router's queue, even if its destination is topologically very close, due to the strict FIFO policy with no QoS classes.

- **TCP congestion control:**

  Traffic congestions in the Internet are prevented by the TCP strict *congestion avoidance* mechanism (see A.2.5). All this structure has been omitted in our model. The acknowledgements system is left aside: our nodes have no memory of packets that have been sent and no timeouts are implemented. That, as we shall see in results, leads to heavy congestions on the network.

- **IP's packet *Time-to-Live*:**

  as seen in section A.2.3, IP packets carry the information "time-to-live" (TTL): it's basically a counter that is decreased of one unit each time the packet is forwarded by a router, thus avoiding infinte loops. Actually, in our model this feature wouldn't affect traffic behaviour anyway, as our routing mechanism prevents loops from happening, being based on the shortest distance path with no possibility of error or malfunctioning.

- **ICMP protocol:**

  Internet Control Message Protocol (ICMP) is used to send error/control messages over the Internet, to eventually signal an error to another Internet component, such as "TTL exceeded in transit" that is sent to the source of a datagram by the router that bring to zero the TTL parameter of a packet, thus discarding it. The sender is now aware of the problem and can take its countermeasure. Many network utilities are based on ICMP, such as *Ping* or *Traceroute*. Our model doesn't implement such a control protocol - errors never occur.

In the next chapter, we analyse in details the properties of different types of networks when they are used as a bed for the traffic flow.

4.3 Traffic properties evaluated during the simulations

Different kind of technological observables are measured on the network during the simulations. These are carried out for a large number of TS (hereafter referred to as τ). τ is chosen to be enough large to allow the "exploration" of all the network, by all the generated packets, to allow a complete sampling of the network's topology.

The main quantities which have been evaluated during the simulation time are:

- the fraction p of delivered packets during the duration of the simulation;
- the average delivery time $\langle T \rangle$ (packet *delivery time*) which is the time distance between the TS when the packet is produced and the TS when the packet arrives to its destination;
- the $\langle T \rangle$ standard deviation, σ_T;
- the length of the router's buffers, at different times of the simulation;
- the usage of a link, measured as the fraction of the generated packets passing through it;

The most important quantity, which somehow summarizes the quality of functioning of the network and its efficiency is the packet's average delivery

time $\langle T \rangle$, which has been evaluated as a function of the traffic level λ. Each generated packet, at the end of its path, arrives to the destination node. The measure of the overall time spent by the packet to perform its path from the emitting node i to the destination node j,provides a significant estimate of the capability of the network to sustain the traffic level. After all, the performance of a network is measured in terms of the time a data packet spent to be delivered. If t_k is the time needed to the packet k to travel from its emitting node to the destination node, then the required value of $\langle T \rangle$ is

$$\langle T \rangle = \frac{1}{M} \sum_{k=1}^{M} t_k \tag{4.4}$$

where M is the number of packets which have been $effectively$ delivered within the simulation time τ. This definition accounts for the fact that, as we assume a finite simulation time, at its completion, only a fraction of emitted packets will have been effectively *delivered* to the destination node; the remaining fraction will still be in travel toward the destination node. Therefore the average of eq.4.4 is evaluated only on packets which have been completed their route up to the destination node during τ. The others will not be taken into account in the average process, still they affect the network's behaviour draining time and resources to be forwarded.

The standard deviation σ_T is also evaluated as

$$\sigma_T = \sqrt{\langle T^2 \rangle - \langle T \rangle^2} \tag{4.5}$$

where

$$\left\langle T^2 \right\rangle = \frac{1}{M} \sum_{k=1}^{M} t_k^2. \tag{4.6}$$

Chapter 5

Results

In this chapter, we will firstly show the results of the simulation of the dynamical model on different network structures. The second part reports the results obtained in the simulations aimed at testing the effects of perturbations on the traffic behavior. This chapter is organized as follows. In the first section, we report the general features of the traffic which develops on these types of networks. According to that, we have settle a number of parameters (i.e. the simulation length) which have been used throughout all the simulations. In the second section we have investigated the differences in the traffic dynamics generated by the network's topology. In that, we have compared the results coming from Scale Free and Random networks. The third section will be devoted to show the differences introduced by the routing strategy. The last section will be devoted to summarize the results obtained after the introduction of "faults" into the network, by recording the variation of the traffic properties associated with them. In particular, we will describe the network's behavior after link's and node's removal (we have adopted either a random removal of nodes and links, and a "deliberate" removal of the most central links and nodes, to simulate an attack). We have also introduced a

further type of perturbation consisting in the localization of traffic. With this we mean a type of traffic which, instead of being directed everywhere in the network, is localized in a specific region. This type of fault has reference to the type of communication which establishes in a network upon some natural calamity or event which induces a strong localization of the destination nodes.

5.1 Results of the traffic dynamics

5.1.1 General behavior and the DIMES network

As we have started to deal with the complete DIMES network ($N = 14154$ nodes), we introduce the results on the traffic dynamics which we have been able to obtain on such a huge network. The first result concerns with the behavior of the traffic on DIMES, in terms of the value of $\langle T \rangle$ as a function of the traffic level λ, in the case of `fixed` routing (fig.5.1).

The two "main" values are the network traffic intensity, λ, on the X axis, and the average packet delivery time $\langle T \rangle$ on the Y axis. It is worth noticing that the most relevant feature of $\langle T \rangle$ is the presence of a two–phase behavior: a low–traffic, linear behavior (we will indicate hereafter as *equilibrium* phase), characterized by values of $\langle T \rangle$ of the same order of magnitude of the average node's distance and a high–traffic, highly non–linear region (we will indicate hereafter as *congested* phase) where the value of $\langle T \rangle$ grows rapidly up to very large values (much larger than the internode distances). From these results we can evaluate the *transition phase point*, λ_c, that is the traffic value where the network switches to the congested phase.

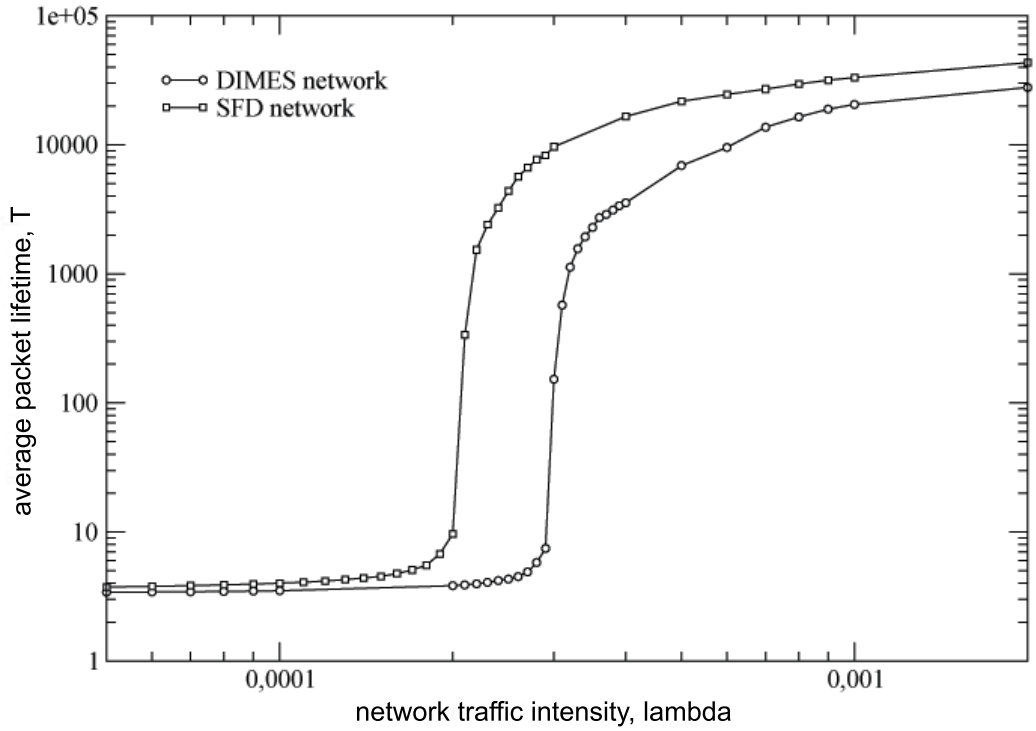


Figure 5.1: comparison between DIMES and SFD networks in a simulation with fixed routing. $N = 14154$ for both networks.

The λ_c value in fig.5.1 is as low as $3 \cdot 10^{-4}$.

The first questions which arise are the following:

1. which is the key factor of the dynamic model which determines the onset of the phase transition?

2. is the phase transition somehow related to the simulation parameters (i.e. the simulation length, the size of the network etc.)?

Before going further, we have make use of the results obtained in the definition of a growth model able to produce a network with the same topological properties of the DIMES network. The DIMES network with $N = 14154$ nodes required a very large amount of computational power to be analyzed,

and, due to technical limits, only the fixed routing (see subsection 4.1.1) could be applied to it. Therefore we have decided to make all our simulations on a smaller–scale network with the same topological properties of the DIMES network. A Scale Free network has been thus generated with $N = 3000$ nodes, by using hte model proposed in section 3.5. This network, which will be used throuhought this work, represents a smaller "replica" of the DIMES network. As such, it will be indicated as SFD. Figure 5.1 shows a comparison between the behavior of the two networks: λ_c is slightly different, but the "shape" of the behavior is clearly coherent, letting us assume that we managed to reproduce a good model, at least at this approximation level. The difference in λ_c are due to the difference between the DIMES network and the SFD model that we already pointed out in subsection 3.5.3, with special relevance to the lack of low–med degree nodes in our network.

Concerning question (1) above, the most relevant role in this context is played by the length of the node's buffer. If we display (fig.5.2) the buffer's lengths in all the nodes (IDs) of the SFD network before and after the transition point ($\lambda_c = 0.0016$), we see that the buffer's lengths undergo to a rapid increase. The figure clearly shows which is the key for the understanding the congestion of the network. Hubs play a central role in Scale Free networks. These nodes display the higher centality values and most of the shortest paths pass through them. So the global performance of the network highly depends on their performance. In figure 5.2, hubs have a low node ID (they likely belong to the first set of nodes of the network by definition of the growth model), thus making the different behavior easy to spot on the log–log plot. While the lower–degree nodes (the tail of the figure) show a

little increase in the average queue–lenght, hubs buffer size raise from less than 0.1 up to 10000 and more, thus playing the role of `bottlenecks`. Of course, also in the non-congested phase ($\lambda < \lambda_c$) hubs are more "busy" than other routers in receiving and forwarding packets; this is shown by the proportions between sizes in different times of the simulation. Looking at the average router's buffer size throughout the simulation (small inset of fig.5.2), we indeed note that it's closely bound to the average packet lifetime and this figure diverges at the same point.

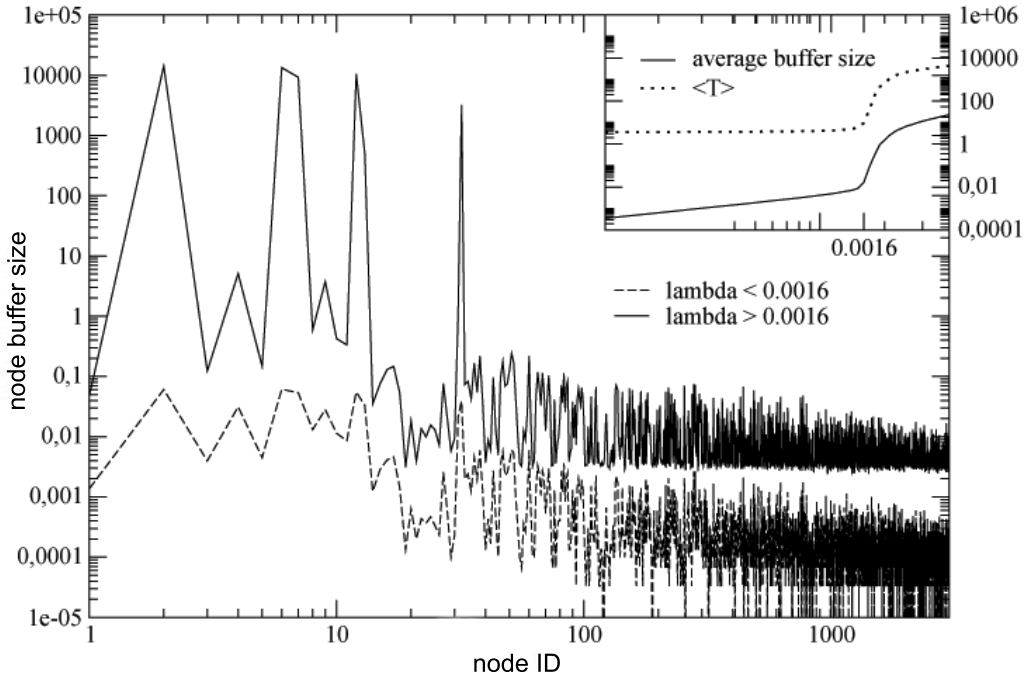


Figure 5.2: buffer sizes in a SFD network with $N = 3000$, photographed with two different λ values, before and after λ_c.

Concerning question (2) above, we have performed the several simulations to ascertain the effects of both the simulation duration τ and the size of the network.

The duration of the simulation time τ doesn't affect the overall results.

What changes upon τ is the "size" of the congested tail, while λ_c is not affected at all. τ must not be too short as it would prevent the network from showing its behavior, or making it difficult to spot. It can't be too long neither, as the computational time increases with τ. We found that a good value of τ to abtain clear results is an oreder of magnitude bigger than N. Figure 5.3 shows the behavior of different networks with different routing policies.

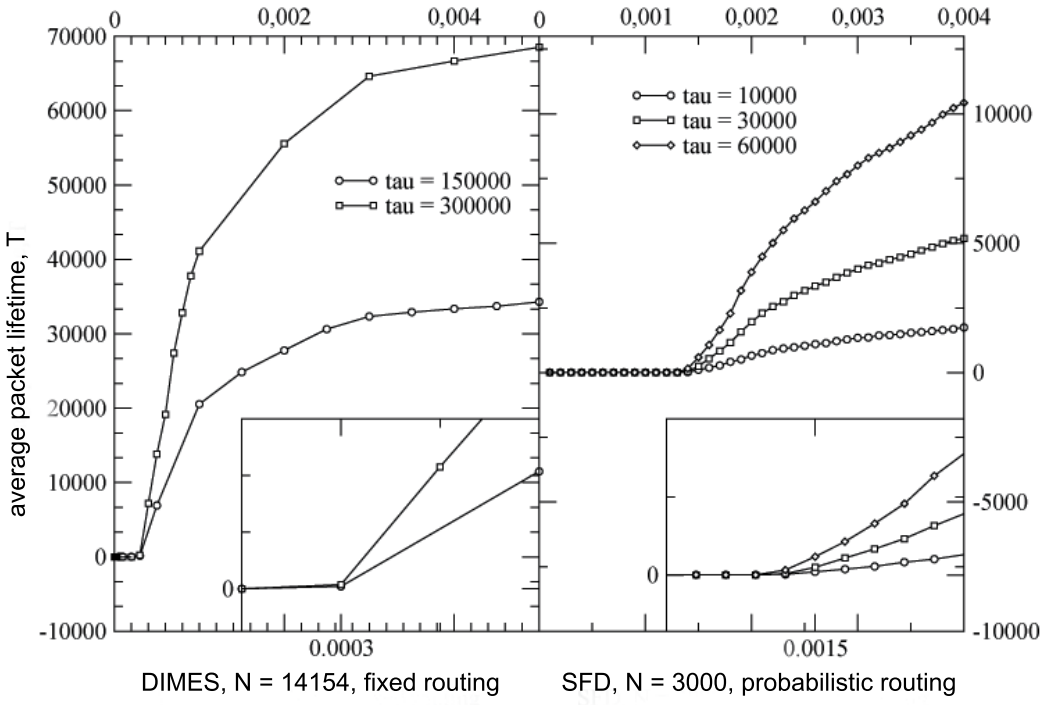


Figure 5.3: average packet lifetime measured over different networks, routing strategies and simulation duration time.

In both cases, λ_c does not change. In the SFD case with $\tau = 10000$ it's slightly less clear where λ_c occurs. The fact that our simulations runs on a finite time implies that when the simulation ends there are packets that are still in queue, waiting to be delivered, and this is a first important

motivation in checking the percentage p of delivered packets over the total of emitted packets: we found that when p goes under 50 or 40 % results are not significative anymore, as $\langle T \rangle$ is evaluated on less than half of the total packets, leaving aside all of the packets that are still waiting. Therefore we have set the total duration of our simulations, τ to a value of $\tau = 30000$ for the SFD and Random systems with $N = 3000$ nodes.

Concerning the size of the network, we evaluated λ_c in both Random and SFD networks, varying the the number of nodes N. This is the parameter that mostly influences λ_c, which decreases when N increases ($\lambda_c \sim \frac{1}{N^\alpha}$, with $\alpha = 1$ for SFD network and $\alpha = 0.33$ for Random newtork, as shown by the log-log plot of figure 5.4.

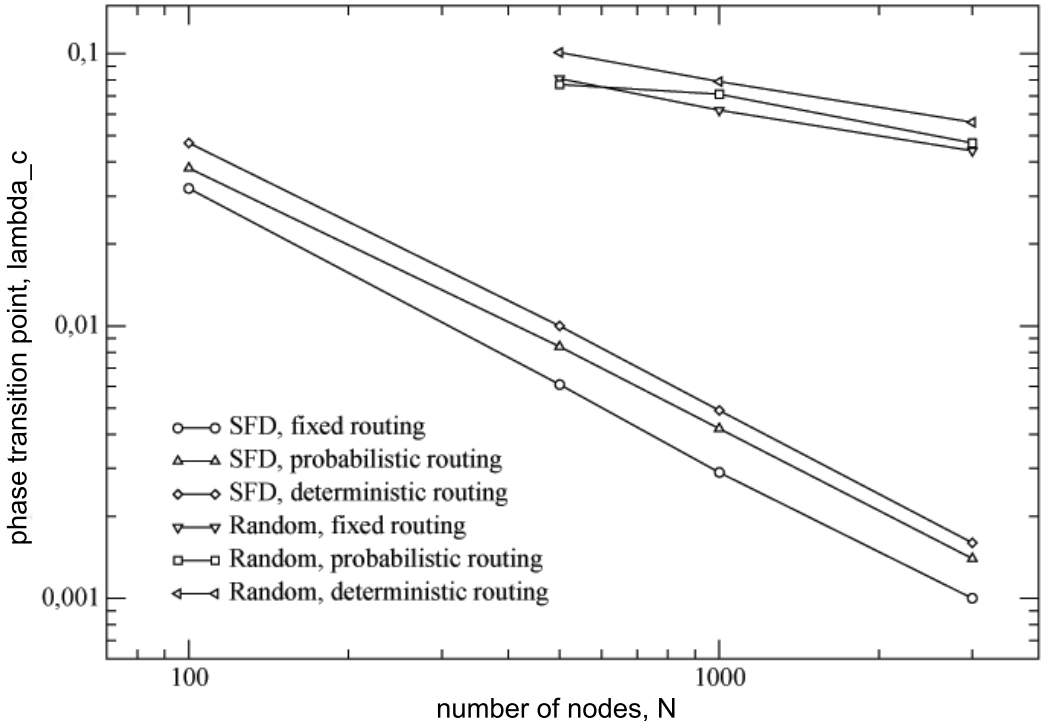


Figure 5.4: behaviour of λ_c for different networks and routing strategies.

The transition to the congestion phase occurs at different simulation stage according to N; simulations and comparison must therefore be referred to the same network size.

Figure 5.4 also highlights the importance of routing policies in the simulation. Once we have investigated the influences of N and τ on results, we can consider the different behaviors depending on network topologies and routing policies, respectively addressed in subsections 5.1.2 and 5.1.3.

5.1.2 Relation between traffic dynamics and network's topology

Figure 5.5 compares results from simulations run over Random and SFD networks, toghether with some quantities that can be extracted from the simulation.

In addition to the two "main" values, λ and $\langle T \rangle$, the two segmented lines represent the interval $[\langle T \rangle - \sigma_T, \langle T \rangle + \sigma_T]$, where σ_T is the standard deviation of the measure of $\langle T \rangle$ (see section 4.3). The two little graphs below show the percentage p of *delivered* packets over *emitted* packets; it's evident that the number of packet effectively delivered decreases as the network goes deeper into the congestion phase.

There's a fundamental shape difference between SFD and Random networks. Those are both based on $N = 3000$ nodes, and we let the simulation run till $\langle T \rangle$ and p reached approximately the same values. The Random network shows a much smoother transition into the congested phase, and it continues to linearly grow. At the same time it takes slightly more time - in terms of λ - to reach the same $\langle T \rangle$ and p values (about the 2.1% over λ_c

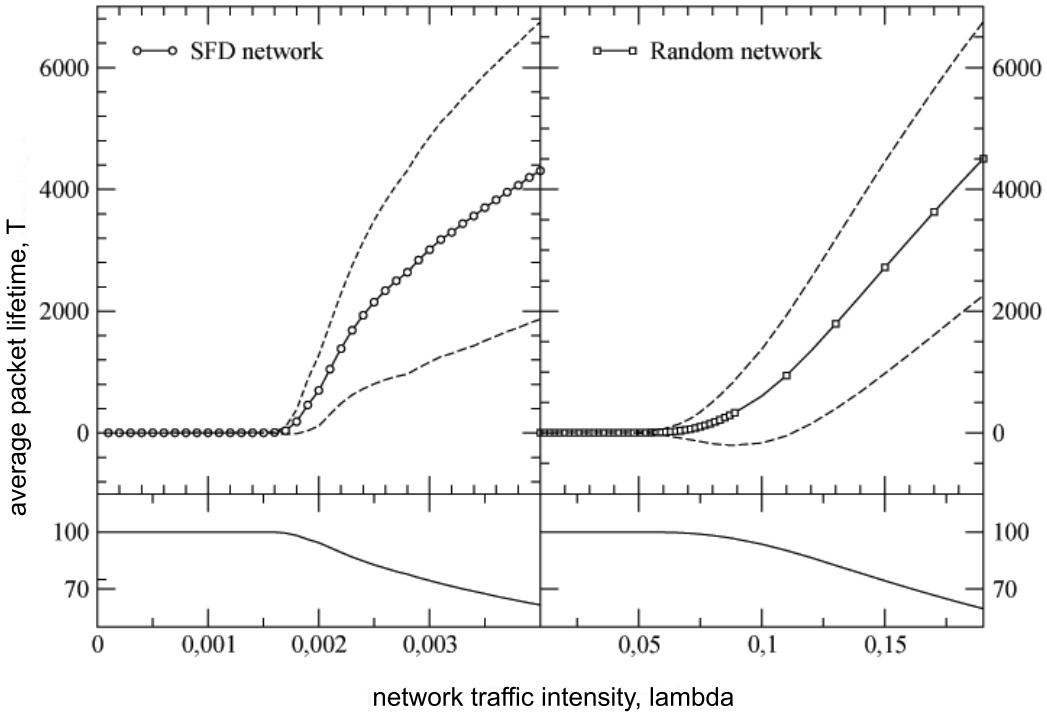


Figure 5.5: example of transition phase points in two networks, a SFD network (left) and a Random network (right). $N = 3000$ for both networks, and deterministic routing is used. Dashed lines indicate the limits of the quantity $\langle T \rangle \pm \sigma$. The two little subgraphs below show the percentage of packets delivered over the total of emitted packets.

against the 1.5% displayed by the SFD network). This means that not only the Random network better handles higher quantity of traffic, but also that the congestion phase is slightly better contermeasured as the traffic increases. On the other hand, the values of $\langle T \rangle$ in the equilibrium phase ($\lambda < \lambda_c$) are lower in the SFD network, an average of 4 against an average of 5 of the Random network. This two values are closely bound to the average networks inter–node distance, as stated in subsection 5.1.1. Indeed, $\langle d \rangle$ is 3.58321 for the SFD network and 4.69546 for the Random network.

The two parameters σ_T and p will not be shown anymore in forthcoming

graphs, so it's worth noting a few points about them. When it comes to higher $\langle T \rangle$, the SFD network displays a higher standard deviation: this means that packets delivery times vary a lot between different packets. At the same time, Random networks show σ_T values much higher than SFD networks when $\lambda \approx \lambda_c$, therefore the uncertainty grade associated to the results is quite high. A good contermeasure is the average process; simulations are run more than once (generally four times) and results are averaged over all the simulations. For very little Random networks, however, simulations are very loosely bounded to the network characteristics, so we will not take into account Random networks with $N < 500$. The p value somehow reflects the behavior of $\langle T \rangle$ with respect to λ: in the Random network it decreases more smoothly that in the SFD network.

In figure 5.6, the main reason why Random networks behave "better" than SFD networks in terms of transition phase point is highlighted. This issue is closely related to the underlying network structure.

As seen in figure 5.2, SFD network's hubs are of crucial importance in turning the network into the congested phase. While hubs play the role of bottlenecks in SFD networks, in Random networks the degree distribution is quite uniform (see subsection 2.2.1), as is the probability that a shortest path passes through a given node. Figure 5.6 shows how routers'buffer sizes are more homogeneous than in a SFD network, especially in the equilibrium phase. Shortest paths are more distributed over the networks; still there are several more central routers that receive more packets than others, but the impact of this phenomenon is much less significative than on the SFD structure.

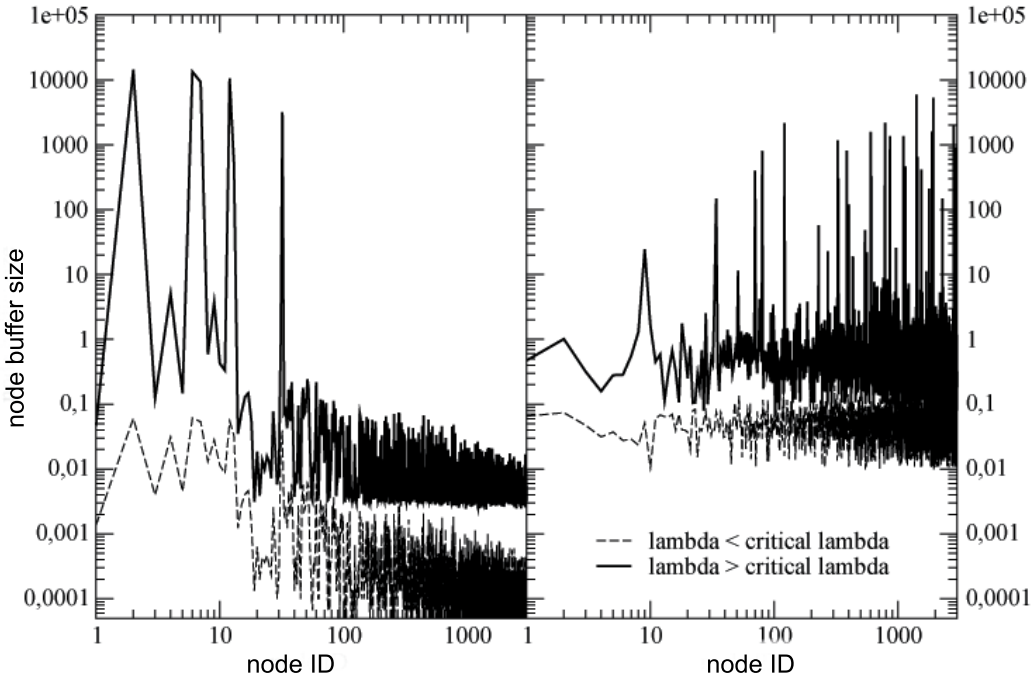


Figure 5.6: comparison between Random (right) and SFD (left) networks in terms of router's buffer size with respect to varying λ. The highest peaks of the Random network are about 6000, while the SFD displays sizes of 10000 and more.

It must be pointed out that, in addition to this issue, a technical limit could also occur in the process: as described in 4.1.1, we only consider *two* out of all the possibile links that a node could choose as next hop to forward a packet. More than in SFD networks, where the problem of bottlenecks is structurally unavoidable at a certain degree, in Random networks the bottleneck could be introduced with this technical issue, whereas exploiting all of the possibile choices could lead to a even more homogeneous usage of links and nodes.

5.1.3 Relation between traffic dynamics and routing strategies

We have already seen (subsection 5.1.1) that, in the transition from equilibrium to congested phase, the main problem to cope with is the presence of bottlenecks and that the choice of routing strategy really has crucial relevance in both cases of Random and SFD network; with respect to routing, the behavior of these two network topology is the same. As showed in figure 5.4, fixed routing is the worst routing strategy, while it is clear that, in these kind of networks, the deterministic routing is more effective than the probabilistic, at the opposite of results obtained in other works with different network topologies such as lattices [47].

In a fixed routing context, routers can forward packets only along a given route depending on the packet destination, thus often concentrating traffic always on the same way and bringing the bottleneck problem to arise in the most rapid way possible. The probabilistic and deterministic strategies, in turn, allow routers to choose between two forwarding routes, each one complaining with the shortest path conditions; this is based on the existence of more than one shortest path between two nodes. See subsection 4.1.1 for the definition of the three strategies.

The possibility of routing packets along more than one link is an important enhanchement for avoiding bottlenecks, and both probabilistic and deterministic routing bring the network to a "better" behavior than the fixed routing. Then, within these two policies, the deterministic routing is the one that better helps to mitigate bottleneck issues by raising the probability of forwarding packets along less used links. Comparisons of routing strategies

effects are shown in figure 5.11 in the following section, that is devoted to simulation results under induced network perturbations. For instance, a node might be constrained to send packets on a given links because there are no other links that respect the shortest path condition towards a given destination. The node at the other side of the link notices the high usage of that links and will likely choose a valid alternative route, thus avoiding putting even more traffic on the same link.

5.2 Influence of network's faults on traffic dynamics

The second part of this chapter is devoted to the analysis of different networks under the stress induced by a given perturbation, in order to explore the response of the different network's topology. We have focussed on several types of perturbations acting on the network:

- links removal, both randomly and "by usage";
- nodes removal, both randomly and "by degree";
- localized traffic.

The key feature that we used to measure the variation V_T induced by the perturbation is $\langle T \rangle$. The quantity that we evaluate is the relative variation of $\langle T \rangle$, given by

$$V_T = \frac{\langle T \rangle_p - \langle T \rangle_{np}}{\langle T \rangle_{np}}, \tag{5.1}$$

where $\langle T \rangle_p$ is the average packet lifetime for a given λ in the perturbed simulation, and $\langle T \rangle_{np}$ is the average packet lifetime for the same value of λ,

evaluated in a non–perturbed context. Simulations have been performed over connected networks: if the network resulted disconnected upon some kind of perturbation, simulation is not carried out . Of course, routing strategies affect the network in perturbed situations as well: the proportions between fixed, probabilistic and deterministic routing are the same, though, so exhaustive results with respect to that will not be showed, unless it's important to obtain a good sketch. In case of two different values of λ_c, we will address the phase transition point before the perturbation as λ_c^1 and the phase transition point after the perturbation as λ_c^2.

5.2.1 Links removal

Links have been removed with two different mechanisms of choice. The simpler one consists in randomly removing links, while the other one removes links on the basis of their usage in a non–perturbed simulation.

Random removal

The two types of network we examinated behaved very differently. Figure 5.7 shows the first big difference between Random networks and SFD networks. The teo little graphs below show V_T over λ as the simulation goes on.

The removal of randomly choosen links nearly doesn't affect the SFD network, whose behavior over λ is approximately the same; the variation induced by the perturbation is very small and it's concentrated around the transition phase point λ_c. In turn, variation is much more relevant in a Random network, being V_T much more significant and not limited only to the λ_c region. As we can see, the perturbation even improves the Random

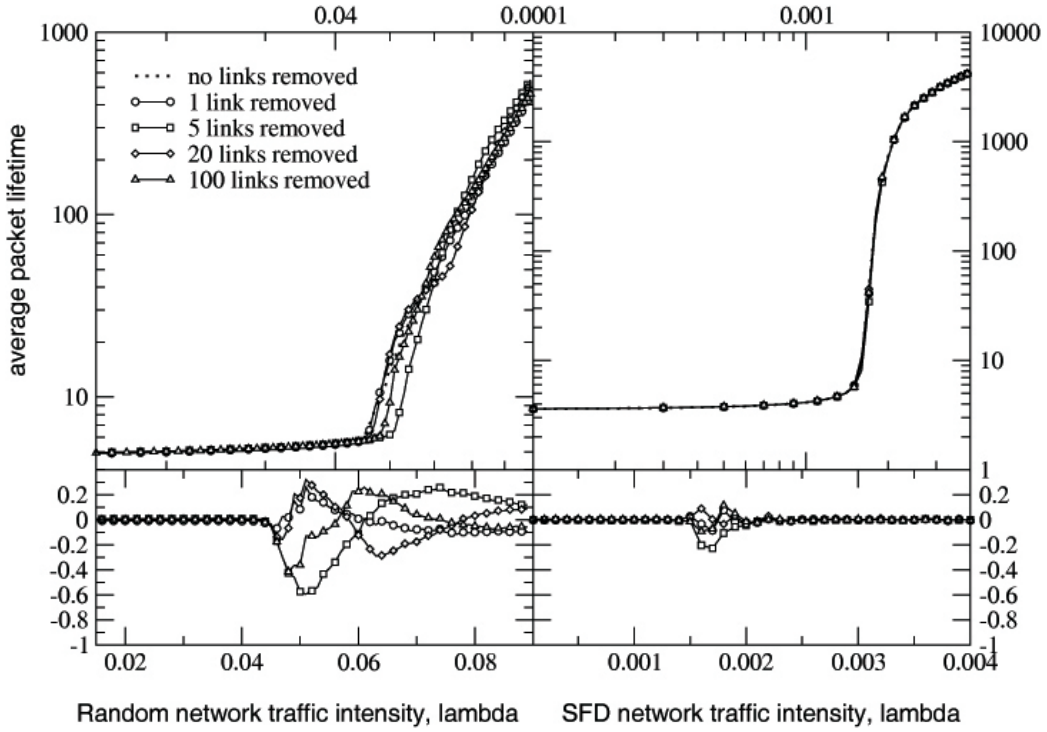


Figure 5.7: behavior of Random (left) and SFD (right) network with respect to the removal of randomly choosen links.

network behavior, where λ_c^2 moves towards a higher value with respect to λ_c^1. This fact will be discussed in the following subsection.

Usage removal

By running a non–perturbed simulation, we can see the different usage level of links. This perturbation removes links starting from the most used, thus representing an extremely targeted attack. Figure 5.8 reports the different networks behaviors.

The SFD network is more sensible to targeted perturbation; the variation induced is very high, especially with respect to λ_c. The same phenomenon of improvement of traffic handling after the removal of links occurs here:

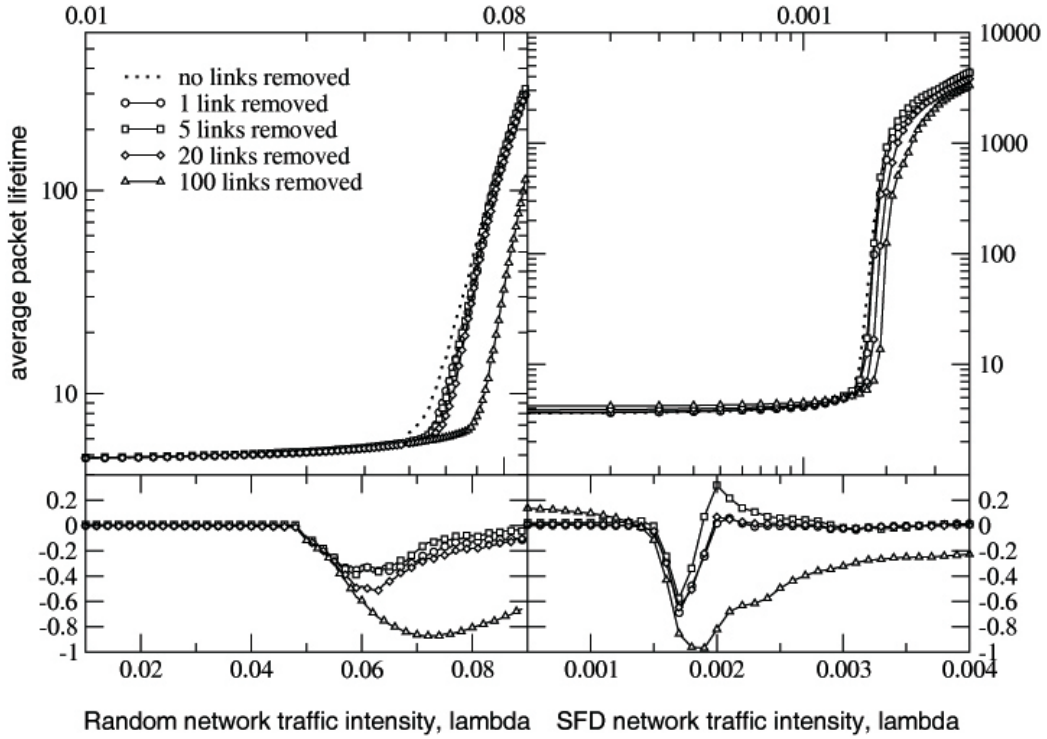


Figure 5.8: comparison between Random (left) and SFD (right) network response upon removal of links choosen by their usage.

the SFD V_T behavior shows clearly, especially in the 100 links removal case, that the network performance in handling traffic radically changes. The new phase transition point λ_c^2 is always moved towards right, but at the same time the values of $\langle T \rangle$ *before* λ_c^2 are higher than in the non–perturbed network. This is clearly shown in figure 5.9.

This fact occurs in every experiment we carried on, rather than being strictly connected to a particular topology or a particular routing, but in Random network it's less striking than in SFD network. As stated before, routing performances can vary significantly the results, but the basical behavior is independent from it and proportions are preserved.

An explanation for this behavior can be inferred by a deeper look to

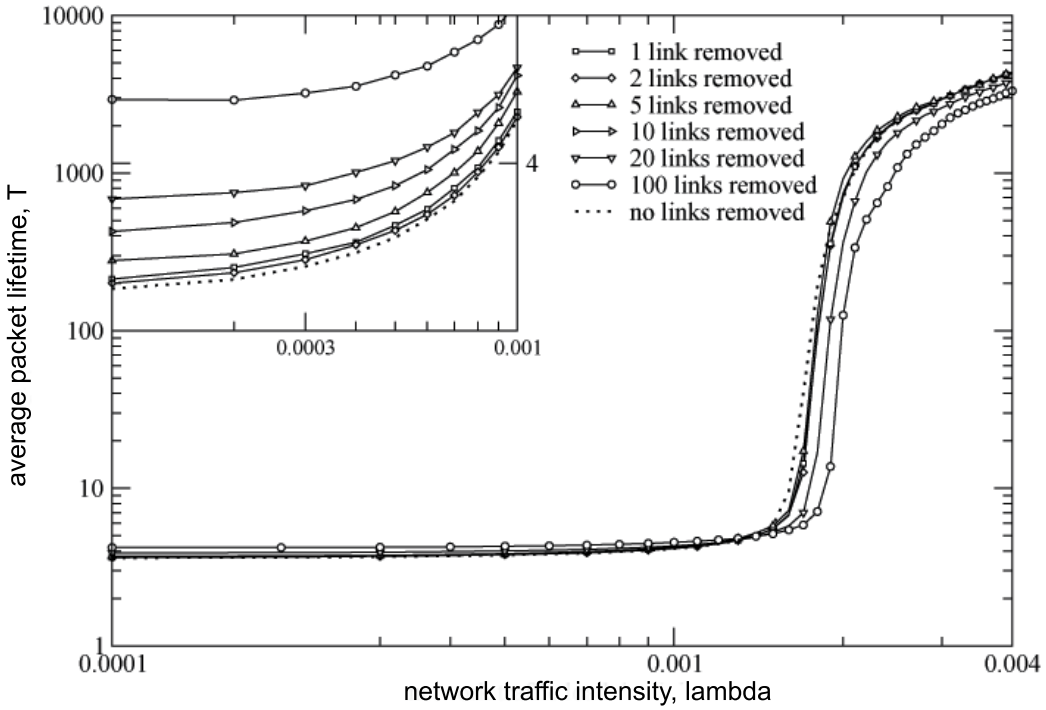


Figure 5.9: behavior of a 3000 nodes SFD network upon removal of targeted links. The graph in the top left corner shows a "zoom" on the equilibrium phase of the simulation.

the state of buffers. The removal of the most used links raises the phase transition points ($\lambda_c^2 > \lambda_c^1$) because it forces the network to find alternative shortest paths and to make a larger usage of different links. The most used links are indeed those who connect hubs with other nodes and between them: the network exploits them in order to keep distances short. By eliminating those links, the new shortest paths between nodes must pass elsewhere, thus reducing the bottleneck issue seen in section 5.1. At the same time, distances between nodes increase, and the average node distance raises; in a 3000 nodes SFD network, it goes from 3.58321 to 3.90273 after removal. Figure 5.10 shows the relation between $\langle T \rangle$ and the average network buffer size for a given value of λ (the average buffer size is averaged over all the time steps in

a simulation for each value that λ assumes). It has to be stressed, however, that in the equilibrium phase ($\lambda < \lambda_c$), link's removal *decreases* the network's efficiency, particularly in the SFD network. When the traffic is sufficiently low and the routers can work properly, the hubs play a relevant role for traffic dispatching and the elimination of elements of high centrality reduces (even if slightly) the network's efficiency (see lower right curve in fig.5.8 related to the performance variation in SFD networks and inset graph in fig.5.9).

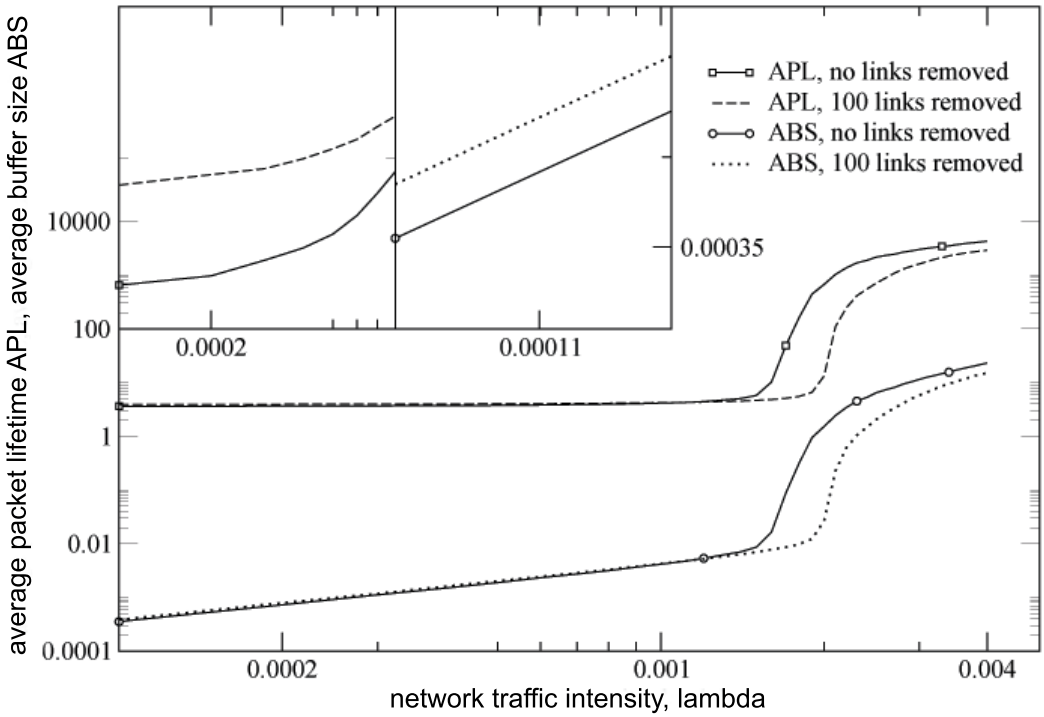


Figure 5.10: relation between average packet's lifetime and average buffer size over all the simulation. Insets show a magnification of the equilibrium regions, in both cases.

The average buffer size and the average packet delivery time are closely related. After the link's removal, throughout the equilibrium phase, distances are bigger and packets takes more time to be delivered, so routers are

constantly slightly more occupied. On the other hand, being the traffic load more distributed, the network can handle a larger amount of traffic before the onset of the congested phase.

Routing policies bring some differencies into the network's response to this perturbation, with effects that are proportional to those induced in non–perturbed networks (see subsection 5.1.3). Figure 5.11 shows routers buffer size in three simulations run on the same SFD network, using each time a different routing strategy and removing the most 100 used links.

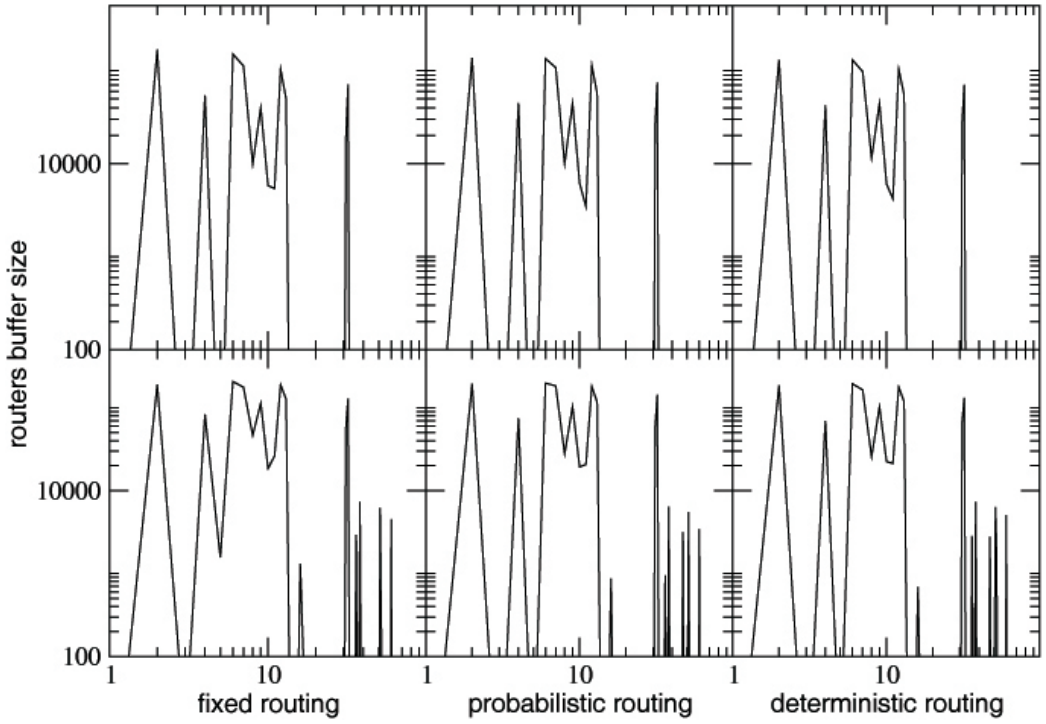


Figure 5.11: routers buffer size in a 3000 nodes SFD network before and after the removal of the 100 most used links, with the three different routing strategies.

In absence of perturbation, buffer size peaks are slightly higher in the fixed routing case, which is the one that most "benefits" of the link's removal

in terms of improving the λ_c value. The link's loss forces the network to redistribute the load over a larger number of routers than under normal circumstances, and the combination of link removal and deterministic routing brings the most effective improvement.

5.2.2 Node removal

The process of node removal can be ascribed to that of link removal, as removing a node is equivalent to cutting its links with the neighbor nodes. The targeted node removal didn't bring in any interesting results: in case of an SFD model network, due to its "extreme" Scale Free nature in terms of hubs and leaves (see section 3.4, where the DIMES network is described) the removal of a high degree node always brings to the disconnection of the network, thus conferming the little robustness to targeted attacks (*single point failure*) that is a characteristic of Scale Free networks. In a Random network there's no real meaning in "targeted node removal", as each node displays about the same degree as others, thus having the same "weight" in network structure and dynamic processes. Moreover, Random networks tend to disconnect even with a small set of randomly removed nodes (even 1-2 nodes), so the only result that is worth showing is the SFD network behaviour upon removal of randomly choosen nodes (figure 5.12).

The behavior is roughly the same as for the removal of randomly choosen links (figure 5.7), thus conferming also the property of Scale Free network to mantain their topology structure also if random nodes are removed, highlighting the strong robustness to random failures.

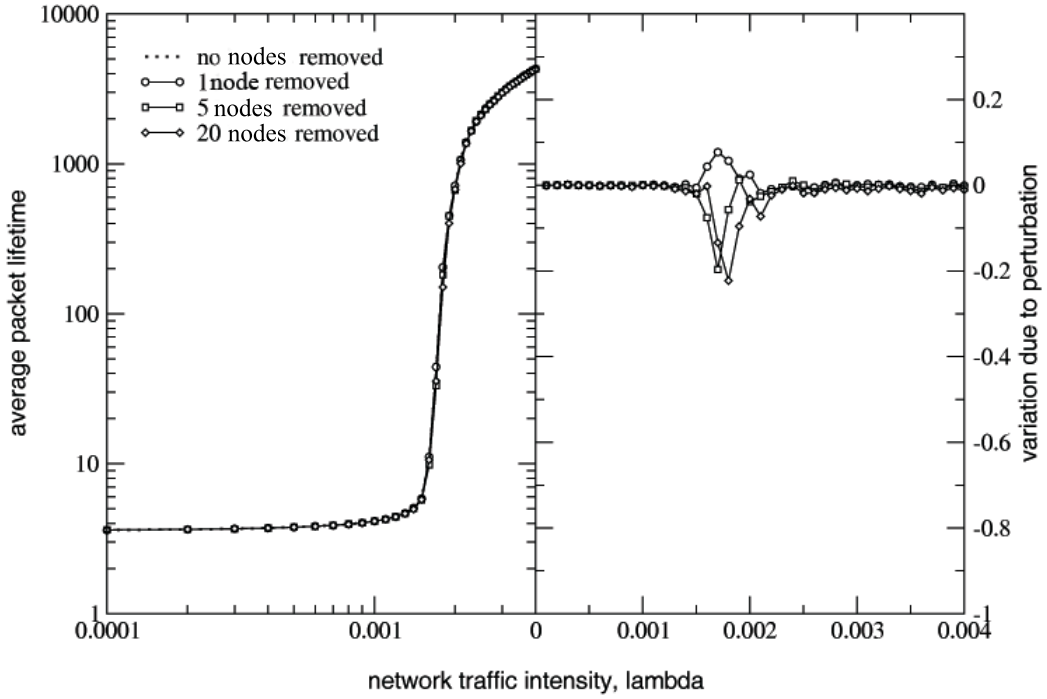


Figure 5.12: behavior of a 3000 nodes SFD network upon removal of randomly choosen nodes.

5.2.3 Localized traffic

This perturbation aims to simulate a real situation where most of traffic is addressed towards a specific (i.e. geographically limited, that is "localized") region of the network; this could happen upon a disaster as an earthquake, when people tries to contact the site where it happened.

The simulation of a similar scenario has been attacked by firstly choosing a "region", represented by an (intermediate–size) hub, and sending, with equal probability, packets from all of the other nodes to the selected hub and to its neighbors. Figure 5.13 shows some results on a 3000 nodes SFD network, when the 7^{th} and the 10^{th} largest hubs are targeted.

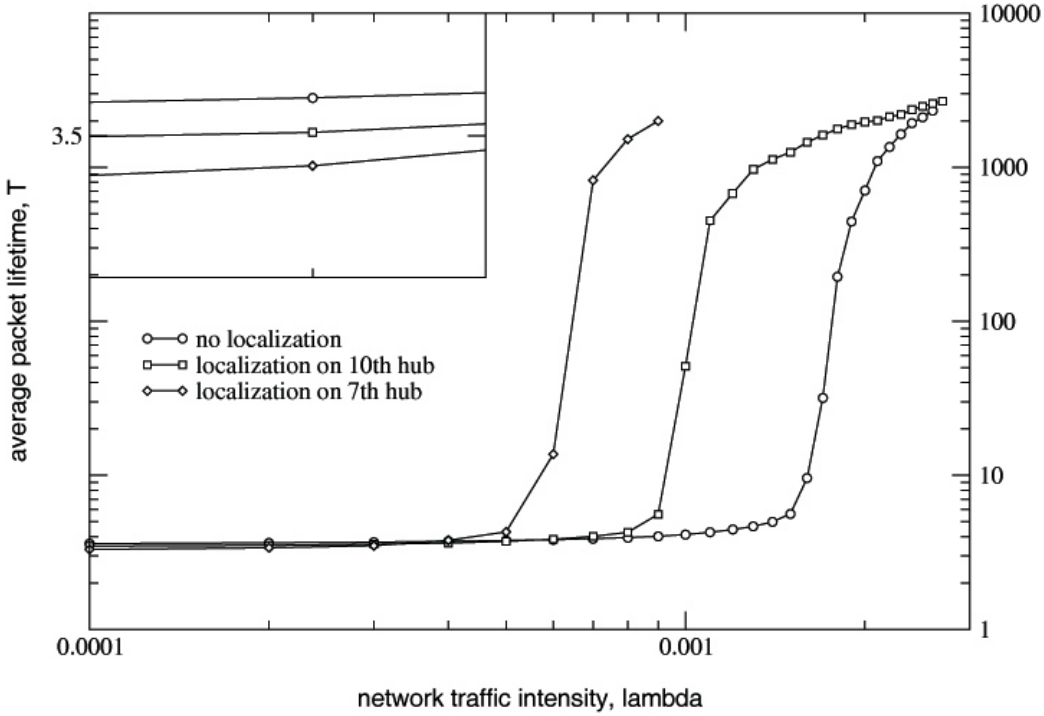


Figure 5.13: behavior of a 3000 nodes SFD network with respect to a localized traffic.

The 10^{th} largest hub has degree 79, while the 7^{th} has degree 193. The

perturbation induces shorter lifetimes for packets travelling in the equilibrium phase: that's due to the destination nodes being all neighbors of a large hub (if not the hub itself), so that the distances that separate emitting nodes and receiving nodes (that can emit packets as well) are shorter. However, the phase transition point λ_c occurs earlier, reproducing on a small size scale the bottleneck process that's been discussed in section 5.1. In figure 5.13 we showed only the 7^{th} and 10^{th} largest hubs because effects are proportional to the degree; choosing a higher degree hub, $\langle T \rangle$ is lower in the equilibrium phase and λ_c occurs earlier.

Random network response to this perturbation is similar, but variance is very high. The effects of this kind of perturbation are less evident than in SFD networks, mostly because of the structure: the small set of nodes that receives packets is not actually bound to a hub, as nodes have roughly the same number of connections.

Chapter 6

Conclusions and perspectives

This work has addressed the problem of understanding the relation between the *topology* of a communication network and its function and efficiency.

Recent works have highlighted the impact that the structure of a network (formed by self–assembly process in an unsupervised regime) might have on its functional properties (the way it works, its efficiency etc.). In many of them (in cellular protein interaction networks, in social networks etc.) the network function seems to be *supported* by the structure. In other words, the network's topology helps the system to behave in an highly efficient way. The topological structures of these networks, moreover, provide the achievement of an high *robustness* (small propensity to being damaged by random faults).

We have investigated, in some details, these issues on a model system representing the Internet.

The Internet has been studied, in recent years, as being one of the most relevant prototypes of "complex systems", which grows under the effect of some "phenomenological" driving force which, in the last ten years, for instance, has determined its impressive growth–rate towards the actual highly–complex structure. The relevance of the Internet as prototype of complex

system has stimulated a great deal of efforts to produce a complete and reliable map, in terms of a graph reproducing its world–wide structure, at the level of AS–level routers.

In this work, we have made use of the results of the DIMES project, one of the latest efforts, funded by EU, to map the global structure of the Internet.

In the first part of the present work, we have analysed the Internet map under the topological point of view. This means that we have been interested in evaluating the quantities characterizing the Internet topology, in order to extract relevant insights on its structure.

This analysis has allowed to underline two main facts:

1. the "extreme" Scale Free character of its structure (coexistence of large hubs and many loosely connected nodes called *leaves*), including a high clustering (presence of a large number of three-vertices structures), leads the Internet network to display a low average inter–node distances (functional to transmission of data), providing at the same time a good resilience to random (i.e. un–targeted) faults. Such an extreme character has compelled the use of a modified growth mechanisms, with respect to the simple Preferential Attacchment (PA) mechanisms, composed of a modified PA growth to enhance the hub's sizes and the inclusion of the Triad Formation for allowing the reproduction of large clustering coefficients. Despite these efforts, the DIMES replica shows still few differences with respect to the real topology. Many differences arise for the reasons explained in the following point.

2. the actual map of the Internet, issued from the DIMES project, has a

tiled structure. The spectral analysis and the min–cut theorem have, in fact, allowed to demonstrate that the DIMES network is not "homogeneous" but it is constituted by large, highly clustered regions which merge one into the other through weak boundaries (i.e. with a low number of interconnecting links). This fact, which should be carefully investigated, can be either a specific tract of the Internet structure or be a consequence of the method employed, in the DIMES project, to create the map. Further measurements performed on a different world–wide Internet map, provided by the US–funded Routeviews project, have confirmed the possibility of a "tiled" structure.

Aside to the analysis of the Internet graph topology, we have also realized a model able to describe the traffic flow, whose action has been simulated when acting on the Internet network. This model attempts to reproduce a number of technological mechanisms which are effectively used in the "true" Internet network, allowing it to efficiently sustain the data flow. Many other features, which constitute a further "intelligence" of the software layers running on the Internet, have not been included in the model, as our main concern was to investigate the role of the network structure to sustain data traffic. The main actions implemented in the traffic model are:

- each node represents a different router; each link represents a physical connection between two routers. A link is bi–directional and has an infinite bandwidth (or, analogously, data packets running on the links have an infinitesimal size, to avoid the relevance of the communication bandwidth);

- at a given time–step, each node can send only one packet and can receive as many packets it has to;
- each router has a infinite–size buffer which can be filled by all the data packets waiting to be dispatched. Data in the buffer are treated on the basis of a FIFO policy;
- each router hosts a Routing Table which associates, to each destination node, the address of one of its neighboring node from where the minimal path to the destination node starts. According to the number of addresses associated to each destination node and with the policy for the choice of such address, we have considered three different routing strategies: `fixed`, `deterministic` and `probabilistic`.

We have built up several networks, of two different topological classes, in order to observe their differences in sustain data traffic; these structures were a DIMES–like Internet structure (though of a size smaller than the DIMES network) and an equivalently large *Random* network. The behavior of both these structures have been investigated by using the dynamic model, with `fixed`, `deterministic` and `probabilistic` routing strategies.

All types of networks, independently on the adopted routing strategy, display a two–phase regime: a low–traffic regime, characterized by a linear increase of the average time for packet delivery $\langle T \rangle$ (equilibrium phase) and an high–traffic regime, characterized by an highly non–linear behavior of $\langle T \rangle$ (congested phase) which attains values of several orders of magnitude larger that in the equilibrium phase. The critical traffic value at which transition occurs, λ_c decreases with the increase of the network's size, for both scale–free

and random networks.

Whereas the Internet's topology ensures robustness at the structural level (there is a low probability that a random fault produces an high structural damage to the network), it does not allow the reach of an equal efficiency for the traffic flow. A major result of this work is the fact that, under the simple flow model implemented to reproduce data traffic, the critical traffic value λ_c for an Internet–like network results to be *lower* than that a random network of the same size. It means that an Internet–like network reaches the congested phase with lower traffic levels with respect to a random network. This has been ascribed to the specific Internet topology: the large hubs, on one side, are responsible of the high robustness; on the other side, however, constitute a bottleneck for communications. Nodes and links, in scale–free type networks such as the Internet, are highly non–homogeneous (in terms of centrality, for instance), while in random networks all the nodes are practically equivalent over all points of view. This asymmetry is the principal responsible of the fact that, under the hypothesis of equal technological power of all the nodes independently on their degree, the routers of the most central nodes start filling at a rate higher than that with which they can unload.

The picture which emerges from our results can be summarized as follows.

The Internet, as many of the complex systems which self-assembly in an unsupervised–growth condition, *is compelled* to adopt a growth mechanism which inevitably produce a scale–free type network. In the Internet case, it is indeed a combination of several mechanisms (the basic one being the Preferential attacchment), whose global result is the realization of a "extreme" scale–free network. This specific growth mechanism depends on the type of

forces which produce the growth: the creation of communities (large hubs), the presence of triangles which settle down the communities etc. This structure, however, although being robust, is not appropriate for traffic flow. This probably represents one of the major driving forces for the development of efficient intelligence strategies, which are thus needed to overcome the functional limitations introduced by the topological structure. This is one of the strongest issue that we intend to put forward with this work.

Intelligence strategies are indeed widely used in the Internet to reduce congestion. Among them, the TCP's traffic congestion mechanism is one of the best contermeasure the network's intelligence offers against bottlenecks problem. If we imagine the protocol being applied to our networks, the strict control imposed by TCP provides a "global" threshold of traffic. This countermeasure is equivalent to the "perception" of an unsustainable traffic level (which is indeed similar to the phase transition point that has been discussed in this work); when traffic intensity goes behind that, TCP reduces the network emission of packets, thus restoring the traffic level under the threshold, where data can be delivered with the best possibile performance.

The limitations introduced by the network's topology are, indeed, partially overcome by introducing more efficient routing strategies. The "central–links" bottleneck (most minimal paths pass through the same nodes and links, which unavoidably collapse under high traffic conditions) is partially removed by wiser routing strategies, such as that allowing to choose between several paths for forwarding a packet and by relating this choice to the traffic previously delivered along those paths (higher the previous traffic, lower the probability of delivering new traffic on that path).

We have used the traffic model to evaluate the "functional vulnerability" of the network. With "functional vulnerability" we mean the proneness of the network to decrease its functionality under the effects of some structural faults, such as node's or link's removal. Whereas the scale–free character introduces a structural robustness, does it produce the same effect when a traffic flow takes place on it?

Results, in this case, are quite astonishing but, for what previously remarked, largely comprehensible. In an Internet–like network, the removal of highly central links (i.e. those where most of the traffic of data flows) produces a sizeable increase of the average time for packet delivery $\langle T \rangle$ at low traffic values (which is expected as, in equilibrium conditions, hubs and central links do not constitute a bottleneck but a structural advantage). For high traffic values, in turn, the elimination of central links provides a relief to the network which can route the traffic differently, by using more many links for the communications. This allows to clearly shift the value of λ_c to higher values. This would represent a sort of increase of efficiency after a fault! Although this could be counter–intuitive, it is the logical conclusion on the basis of what has been previously remarked concerning the bottleneck effect produced by high central nodes and links. Moreover, this effect is hardly detected in random networks, where the absence of central nodes and links prevents the occurrence of bottlenecks.

The last simulations have been dedicated to describe the behavior of the network to sustain a specific type of traffic: the localized traffic. In normal conditions, traffic establishes homogeneously, i.e. emitting nodes have the same probability to send a message to any of the remaining $N - 1$ nodes

of the network. When transmission *localizes*, communications are directed toward the same region of the network. This effect has been simulated by forcing all the destination nodes to belong to a small–area region, typically that represented by an intermediate–size hub and its neighbors. This is a truly devastating case for the functioning of the Internet network. In this case, in fact, the value of the critical traffic λ_c shifts dramatically toward lower values: it means that the network cannot sustain localized traffic levels which could be, in turn, easily sustained if not localized. Localized communications mimic a typical consequence of natural calamities or disaster: in those events, all communications tend to localize in a specific area and many users instantiate queries to routers which are "geographically" close and thus probably physically connected.

The present work has allowed to make some interesting observations on the large–scale structure of the Internet. More importantly, it has allowed the realization of a dynamic model able to reproduce the main effects of the traffic flowing in a communication network. Based on this approach, the simulation model can be enriched by introducing further mechanisms (i.e. the non–homogeneity of the routers, by letting more actions to be performed by hugh–degree routers) and to introduce different routing strategies.

A dedicated effort will be produced in discovering `ipremonitory` effects which anticipate the transition to the congested phase and the development of new routing strategies to displace, as much as possible, the critical traffic λ_c to higher values. In particular, several bio–mimetic strategies (i.e. strategies which are used by natural systems, such as bacterial colonies, swarms

etc.) will be "transposed" to the Internet case. A major goal is that of providing the network of a "global awareness" of its state (against the actual "local" awareness given by the recording of the local traffic flow). Router's intelligence, for instance, could be enhanced with a strategy able to spread information about the global network usage. The evaluation of the router's buffer sizes could be related to the congestion threshold (recall the shown correlation between the average buffer size and the phase transition point) and provide a way to anticipate its occurrence.

A further use which will be done of the present model concerns with the study of network's *inter–dependencies* As recalled in the introduction, this work is part of a larger project focused on the study of system's inter–dependencies, i.e. the study of the effects induced on one network by a fault which is produced on another network with which it is functionally coupled. It is under study the interconnection between the electrical power distribution grid and a telecommunication network. Once defined the mutual connection, one could evaluate the relative resiliences to perturbations of different types.

Appendix A

The world we model: an introduction to the Internet

This chapter aims to describe the real object that is modelled in this present work, the Internet. Of course, the model that we will introduce will take into account only a limited number of properties and functions of the Internet. Therefore this chapter, although doesn't pretend to provide an exhaustive description of the Internet, will firstly introduce the general context and than focus more on specific aspects related to our data source, the DIMES project, that works mostly on lower level functions. All informations provided in this chapter have been extracted by [48] and [49].

A.1 Global description of the Internet

The Internet is the worldwide, publicly accessible system of interconnected computer networks that transmit data by packet switching using the standard Internet Protocol (IP). It consists of millions of smaller business, academic, domestic, and government networks, which carry various information and services, such as electronic mail, online chat, and the interlinked Web pages

and other documents of the World Wide Web.

The main goal of the Internet is to provide interoperability between different kind of data network. Those networks were (and still are) based on different architectures and protocols, different physical medium and, consequently, different bare services. The power of the Internet's protocol and interconnection systems was to make all of them able to interoperate. The concept that's necessary to stress is that Internet is not a data network itself but the union of more dishomogenous data networks through some interconnection devices. Today's Internet can reach any kind of computer (personal home computers, scientific workstations, mobile laptops and so on), mobile phones and even some household electrical appliance. Its purposes are not specific: it simply provides a way to exchange data and informations.

Historically, the Internet grew from the public size of a project carried by the *United States Departement of Defense*, that gave birth in the early seventies to *ARPANET*, a packet switched data network that, after having grown and improved its mechanisms and protocols, allowed several universities and research centers to be interconnected through a long distance backbone transport network. That happened between the mid seventies and the mid eighties. From this initial core, Internet expanded till today's classless and purposeless geographical extension.

Throughout the seventies and eighties, the standardization organizations ISO and ITU-T were also working on the definition of a set of communication protocols based on the *Open Systems Interconnection* (OSI) model. It was common belief that an evolution of OSI-based protocols would have been the future of telecommunications. The aim of these organizations was to trans-

form the present analog telephonic network into a general, multi-purpose digital network, merging both circuit-switched and packet-switched technologies. Each network would have converged towards this architecture that would have provided all kind of services, the B-ISDN (Broadband-Integrated Services Digital Network).

The Internet approach to the interoperability issue was completely different: instead of unifying services and technologies, the principle was to take into account the existing diversities and to make them interact, without changing all the different substructures. That's were the concept of *inter-network* emerges: an number of *sub-networks*, containing each a certain number of *hosts*, can communicate through some *interconnecting devices*. Any host in any sub-network must be able to "talk" to any other host in any other sub-network by the same set of *protocols*, which are the real distinctive element of an inter-network.

Internet "won" over the B-ISDN project: it is today the most significant example of world–wide inter-network. Its characteristic properties are at the bases of its success:

- the Internet technology doesn't need a central architecture that controls everything and to which every new node must be connected. There's no specific fixed topology that a new sub-network must be complained with: it's enough to connect it to any other already connected sub-network to gain the possibility to exchange information with all the other sub-networks in the inter-network. Thus, this leads to an un-driven growing process that has been object of many studies [ ].

- the user-network interface is independent on the access sub-network the user is connected to: internet protocols are simply *added* to the specific ones of the given sub-network, that don't need to be modified. Some functionalities might be duplicated, with consequent loss of efficiency, but, on the other hand, this is a simpler way to implement the inter-network functionality in an almost universal sense.

- aside to these properties, the management system is highly *distributed*, making it "light" and robust.

A unified broadband technology would have brought significant implementational and economical advantages, as well as a degree of efficiency and performance that today's Internet is far from. However the concretization of such a monolithical architecture was too difficult to get and the Internet conception succedeed.

Some additional reasons helped this architecture to rapidly wide spread. Among the most important, the costs to the final user have always been accessible; the software that realizes the Internet protocols has always been freely available and distributed with all of the most important operative systems as the Open Source UNIX (and his "son" Linux) and Microsoft Windows. Also the most used application layer software is often free (web browsers, recent VoIP - Voice over IP - software, instant messaging products and so on). The Internet architecture itself is a sort of "open source" technology: detailed informations about the underlying protocols have always been available and everyone can submit his ideas to improve the system. This meant faster procedures than usual, official standardization processes as ISO's.

In the next section we will introduce the Internet components, the logical way they are organized and the protocollar architecture with the two main actors, TCP and IP, and the principal routing protocol BGP, as it's the one mostly used by the DIMES project to build up its database.

A.2 The Internet components and protocols

A.2.1 Entities organization

As stated, Internet is a global merge of different sub-networks. Each sub-network uses protocol that may differ from those of the Internet and from those of other sub-networks; hence, hosts in a particular sub-networks can exchange informations only with hosts connected to sub-networks that use the same protocol set, the same "language". All sub-networks are linked by network layer interconnection devices, *routers* hereafter. Routers consider every sub-network as a single entity, without knowing its internal structure nor how it works. Their task is to route an informative unit to te destination sub-network; once there, the sub-network protocols will fulfil the task of sending the packet to the right host. This way, the load that must be processed by the interconnection devices is proportional to the number of sub-networks and not to the number of hosts, thus leading to a much lighter required processing power.

As shown in figure A.1, routers usually don't provide direct connection between each element of the inter-network: the data flow, that must be transferred from a sub-network to another, should (and usually do) pass through more sub-networks; each of them carry the information till the next

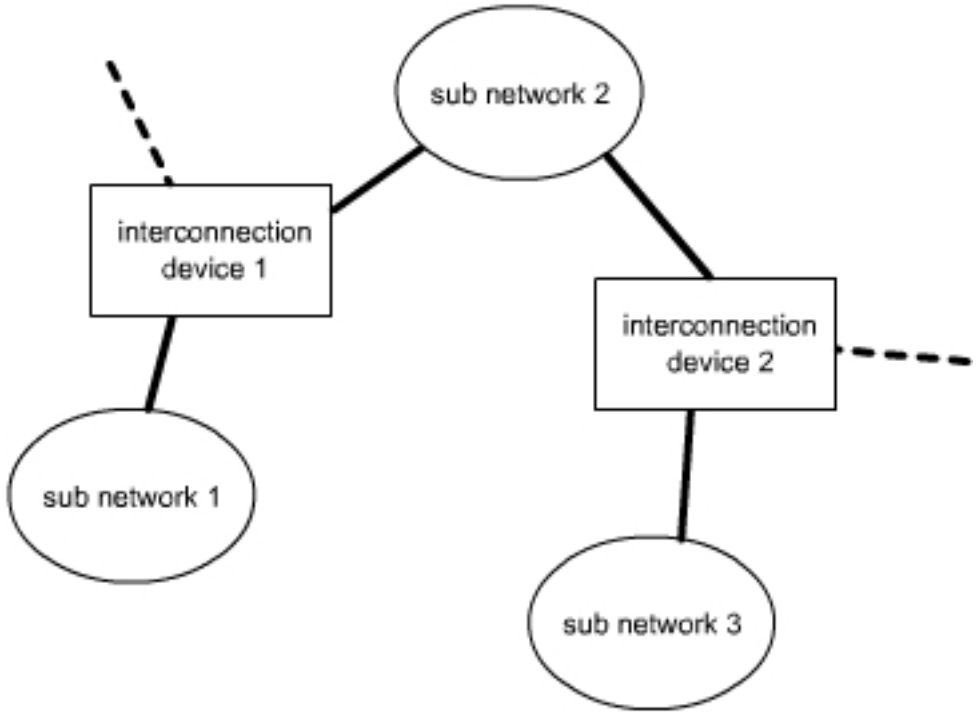


Figure A.1: three sub-networks interconnection example

one along the path and so on, till the destination sub-network is reached. In this context, every sub-network contributes to the information transfer. As introduced before, the Internet management process is extremely distributed, and routing protocols assume a very important role. Each sub-network of any kind (mobile phone network, a wireless LAN, an intercontinental satellite connection...) appear the same.

That's where a new concept is necessary. An *Autonomous System* (AS) is a collection of IP networks and routers under the control of one entity (or sometimes more) that presents a common routing policy to the Internet. Therefore any sub-network appears as an AS, and the important difference between *Intra-AS* routing and *Inter-AS* routing must be introduced. The entity that controls an AS can choose the routing protocol to be used inside

it, so, in general, different AS can use different routing protocols. In order to make possible interconnectivity between different ASs, each AS must employ one or more router to interface with the "outer world", informing it of the AS presence and topology. Usually there are specific routers to accomplish this task, the so called *Border Routers*. Clearly these routers must adhere to the Internet rules and protocol set (explained further on).

The AS notion leads to the definition of two routing protocol classes:

- *Interior Gateway Protocols* (IGP), used inside AS (the term "gateway" comes from a deprecated way to indicate nowadays router).

- *Exterior Gateway Protocols* (EGP), used for communication between different ASs routers. The DIMES project uses one of these protocols to collect data, so it will be explained in more details.

A unique AS number (or ASN) is allocated to each AS for use in BGP routing. BGP is the most used protocol to build up and mantain information about Inter-AS routing. With BGP, AS numbers are important because the ASN uniquely identifies each network on the internet.

Originally, the definition of AS required the control by a single entity, typically an Internet service provider or a very large organization with independent connections to multiple networks, that adheres to a single and clearly defined routing policy. The newer definition came into use because multiple organizations can run BGP using private AS numbers to an ISP that connects all those organizations to the Internet. Even though there are multiple AS supported by the ISP, the Internet only sees the routing policy of the ISP. That ISP must have a public, registered ASN.

AS numbers are assigned by the IANA (*Internet Assigned Numbers Authority*) to regional internet registries (RIRs) in blocks. The local RIR then assigns an AS number to an entity from the block assigned by the IANA. Entities wishing to receive an ASN must complete the application process of their local RIR and be approved before being assigned an ASN. AS numbers are currently 16-bit integers, which allows for a maximum of 65536 assignments. AS numbers are divided into two ranges: the first is that of public AS numbers, which may be used on the internet and range from 1 to 64511. The second range, from 64512 to 65535, is known as that of private numbers, and can only be used internally within an organization. The RIRs plan to issue 32-bit AS numbers, starting in 2007. These numbers will be written using a number format of *<upper16 bits>.<lower 16 bits>*.

A.2.2 Protocollar architecture

The Internet protocollar architecture or (Internet Protocol suite) is organized on four layers, that can be roughly fitted in the seven layers of the OSI protocol stack. As in this standard model, the IP suite uses encapsulation to provide abstraction of protocols and services: a protocol at a higher level uses a protocol at a lower level to help accomplish its aims, offering a *service* to the higher level protocol.

With respect to the seven layers OSI model, only four layers can be identified in the IP suite. From the higher to the lower:

- *Application layer.* This is where all of the higher level application process and services are listed. It spans the top three layer of the OSI model (Session, Presentation and Application).

- *Transport layer*. It corresponds to the fourth OSI layer, and provides the higher layer applications a realiable, error protected, connected service.

- *Network* or *IP layer*. The name comes from the protocol that rule this level, IP (*Internet Protocol*). It can be seen as a "high Network layer" in the OSI model, and it provides a connectionless, best-effort packet routing & delivering service.

- *Link* or *Network Access Layer*, that allows the utilization of infrastructural non-homogenous resources and technologies. This layer is roughly equivalent to the Physical, Link and partly Network layers in the OSI model.

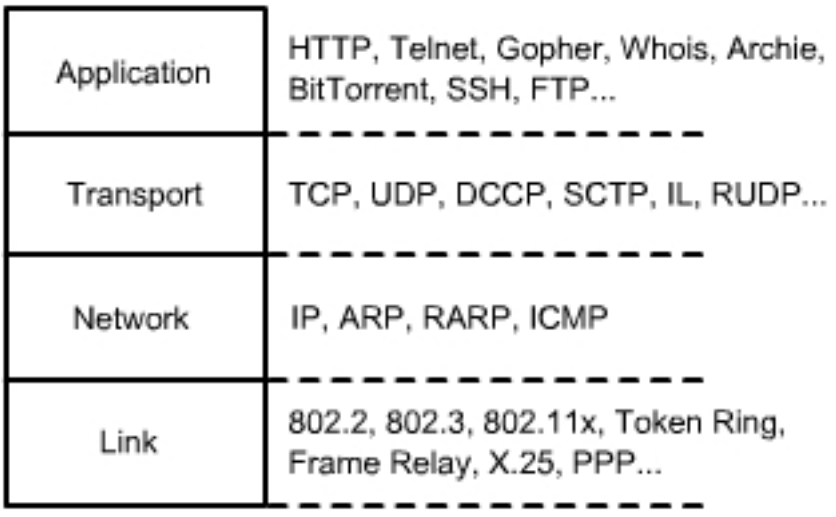


Figure A.2: protocollar architecture with associated example protocols

The principal reason in the low efficiency of Internet is that the IP stack implements many of the typical 2-3-4 OSI levels functions, as routing, error

checking, segmenting and reassembling of data, no matter whether the underlying sub-network already does that or does not. Consequently, in most cases there are duplicate functionalities, but this way the Internet platform can guarantee the same functionalities to everyone, without setting any constraints to the interconnected sub-networks.

A.2.3 The IP protocol

The *Internet Protocol* (IP) is a data-oriented protocol used for communicating data across a packet-switched internetwork.

The principal functionalities that IP realizes are:

- it defines a global addressing scheme: every host belonging to the Internet must be reached through one or more IP addresses (due to several mechanisms (as subnet masking or Mobile IP) that we're not going to explain in details).
- it define the base data unit that will carry information through the Internet.
- it finds the path that a data unit (packet) has to follow to reach its destination.
- it defines the rules for segmenting a packet into *fragments* that will be reassembled once the destination is reached.

As stated before, IP can be used to communicate data across any packet-switched internetwork. This means that, despite its name, the Internet is not the only context this protocol can apply to. Being a network layer protocol, it must be encapsulated in a data link layer protocol. Because of the

abstraction provided by encapsulation, IP can be used over a heterogenous network (i.e., a network connecting two computers can be any mix of ethernet, ATM, FDDI, Wi-fi, Token ring, etc.) and it makes no difference to the upper layer protocols. In case the underlying network doesn't use IP itself, the lower layers have to resolve IP addresses to data link addresses. This resolving is addressed by the *Address Resolution Protocol* (ARP). Indeed, following the Internet success, lots of networks began to use IP as a native addressing scheme and routing protocol, thus eliminating the redundancy of functionalities due to the replication of layers introduced before, as well as simplyfing or eliminating several procedures, i.e. the IP address resolution.

IP provides an unreliable service (i.e. best effort delivery). This means that the network makes no guarantees about the packet and none, some, or all of the following may apply:

- data corruption
- out of order (packet A may be sent before packet B, but B can arrive before A)
- duplicate arrival
- lost or dropped/discarded

In terms of reliability, the only thing IP does is to ensure the IP packet's *header* is error-free through the use of a checksum. This has the side-effect of discarding packets with bad headers on the spot, and with no required notification to either end; IP *can* try to address the problem by sending *Internet Control Message Protocol* (ICMP) message. To address any of these

reliability issues, an upper layer protocol must handle it (in the Internet this role is assumed by IP's alter-ego TCP, as we shall see in subsection A.2.5). For example, to ensure in-order delivery the upper layer may have to cache data until it can be passed up in order. The primary reason for the lack of reliability is to reduce the complexity of routers. While this does give routers the possibility to do as they please with packets, anything less than best effort yields a poorer experience for the user. So, even though no guarantees are made, the better the effort made by the network, the better the experience for the user. The most widely used version of IP is IPv4, the fourth one. IPv6 introduced lots of improvements (that we're not going to explain here) but its diffusion is really slow and therefore it can't be widely used due to compatibility issues.

IP addressing

As with any other network-layer protocol, the IP addressing scheme is integral to the process of routing IP datagrams through an internetwork. Each IP address has specific components and follows a basic format. Each host on a IP network is assigned a unique 32-bit logical address that is divided into two main parts: the network number and the host number. The binary notation is usually translated into decimal notation in order to be more easily managed by human beings, so a typical IP address is usually seen as 192.105.76.218, where each of the four decimal number could vary from 0 $(2^0 - 1)$ to 255 $(2^8 - 1)$. What is really important about routing is that an IP address is composed by two logically separated parts: the *Net_ID* part and the *Host_ID* part. As the names say, the first identifies a subnetwork, while

the second identifies the proper host (within the given subnetwork). The lenght of the two is variable; the first *IP address classes* were defined on account of that, whereas a subnetwork identified by a Net_ID of 8 bit can have up to 2^{24} hosts (because that Net_ID leaves 24 bits for the Host_ID) while a 24 bits Net_ID can only discriminate between 256 hosts (because there are only 8 bits available for the Host_ID part in that subnework). Originally there were three main classes publicy available (A, B & C) but, as the Internet exponentially grew, problems arose due to the rapid consumption of B classes (that could address up to 65536 (2^{16}) hosts). Few new addressing scheme were introduced, especially the "subnetting" mechanism that will be briefly introduced further on. []

The Net_ID identifies a network and must be assigned by the *Internet Network Information Center* (InterNIC) if the network is to be part of the Internet. An ISP can obtain blocks of network addresses from the InterNIC and can itself assign address space as necessary, but an arbitrary set of IP addresses can be used in a network only if it isn't going to be connected to the Internet, or, eventually, addresses used in the network must be "translated" into Internet-compatible addresses: systems apt to do that are known as *Network Address Translation* (NAT). Some IP addresses sets are reserved and therefore must not be used without permission.

The IP address must identify unequivocally a *system*: considering that a system can be shared by more users (eventually, each user uses a different host behind the same IP address) and a user can generally run more network application, the single process is addressed by a composition of addressing scheme, of whom the IP address is only a part, that implies also addresses

managed by upper layers protocols, as TCP and UDP.

IP routing

The goal of routing operations is to "drive" the data packet through the network to its destination. The path can be choosed following several different strategies, based on different criteria: shortest physic distance, resources availability, packet lenght and more. The algorithm can be static (if it doesn't change over time) or dynamic; for instance it could react to network perturbations or changes. In the Internet, the most used routing rules are simply based on shortest path finding algorithms.

Each packet usually follows a path composed by both routers and subnetworks. When a router sends a packet into a subnetwork, it must be encapsulated into the subnetwork data unit format, and eventually its IP address must be resolved into a subnetwork address. Basically, routers only address packets to subnetworks, instead of addressing them directly to hosts. Once reached, the destination subnetwork is able to map its hosts on an IP-address basis and to forward the packet within its mechanisms. That's why IP routers only look to the Net_ID part of the destination address; the Host_ID is only considered when the destination subnetwork has been reached. Two types of routing are then considered, as introduced before: *direct* and *indirect*. The direct routing is the routing that takes place within an AS, whitout needing to step through other external routers, and the task is accomplished by the *Interior Gateway Protocols* (IGP) such as *Routing Information Protocol* (RIP) and *Open Shortest Path First* (OSPF). On the contrary, indirect routing involves more routers and the protocols used for the indirect routing

are the *Exterior Gateway Protocols* (EGP), such as *Border Gateway Protocol* (BGP), that will be explained more in details later.

The main concept behind IP routing is the *Routing Table*. Routing tables are resident in both router and hosts to direct the forwarding of a packet by matching destination addresses to the network paths used to reach them. The construction routing tables is the primary goal of routing protocols, along with the maintenance of consistency (i.e. the informations stored in routing tables must be kept valid).

In the simplest model, hop-by-hop routing, each routing table lists, for all reachable destinations, the address of the next device along the *next hop* to that destination. Each router's routing table provides a list of entries *(R, I)*, where R is the Net_ID of the packet destination and I is the complete IP address of the next router the packet must be forwarded to: every time that a packet must be forwarded, the router simply compare his Net_ID with those in the table, and when it finds correspondence, it forwards the packet towards the right router on the right port (that indeed is listed with the entry above). Assuming that the routing tables are consistent, the simple algorithm of relaying packets to their destination's next hop thus suffices to deliver data anywhere in a network. In case the packet destination's Net_ID is not listed in the routing table (i.e. the router is not aware of the next router it should forward the packet to) there's always a routing table's line that indicates a *default* router, to whom each packet with no other currespondances in the table is sent. This default router should be able to forward the packet towards the destination, but it can eventually forward the packet to its default router: with such a recursive behaviour, the destination should be finally reached.

Closed loops should be always avoided, but, in case, this problem is prevented by a packet "time-to-live" parameter that deletes it after a certain number of hops.

The key concept of IP routing is that each router only knows the *next* router along the path that connects itself to the packet destination. Moreover, routing tables only have informations about the destination subnetworks (represented by the routers that interconnect them) and not about *every single host* in the Internet. These two key features are really important in Internet's dynamics as they let the routing tables being small and easy to check, thus higly affecting routers performances especially in terms of memory needs and access time, being that quite critical in the first Internet period when technology wasn't as developed as today.

Once the packet has reached the destination subnetwork, it will be forwarded directly to the destination host. In a more general sense, "going from a router to the next one" often means to go through a subnetwork to which the two routers are connected, so we can say that indirect routing is a chain of direct routings.

A.2.4 Determining and maintaining a routing table: the BGP protocol

Routing tables are dynamic objects. When a system is switched on, the routing table must be initialized, by the interaction with a fixed server, by consulting a local database, even manually if the system is "small". Then, tipically, a routing table must be updated through the informations exchanged by a routing protocol. In this present work we're interested in inter-AS dynamics

more than in intra-AS's so we'll refer to exterior gateway protocols.

The need for a dynamic update process is evident in the Internet: new hosts and new subnetworks are very frequently connected to or disconnected from the whole inter-network, so the entry in the table must be removed. Links can vary dynamically: if a link becomes unavailable, the path to reach a certain subnetwork might change as packets that were sent over that link can't be routed that way anymore. More, if optimization strategies are done in the network, a certain path may result too busy to be convenient, or a router's buffer (the memory stack where it stores packets waiting to be forwarded) can be full, leading that router to discard packets, that shouldn't be sent there anymore.

In all these cases routing protocols are in charge of maintaining fresh and valid the informations on the actual state of the network, as their goal is to provide routers with the best solutions in forwarding packets. Within those, the *Border Gateway Protocol* (BGP) is the core routing protocol of the Internet. It works by maintaining a table of IP networks or 'prefixes' which designate network reachability between ASs. It is described as a path vector protocol, that means that each AS announces to its neighbors not only the cost ("distance" in a metric dependant sense) of its path to every destination but also the path itself. BGP is designed to enable Internet Service Providers (ISPs) to control the flow of data, thus an AS may choose not to announce some paths it knows due to *policies*, usually determined by financial consideration. The knowledge of the whole network *isn't* needed, as it is in link-state protocol (as *Open Shortest Path Firs* (OSPF)). BGP was developed to replace its predecessor, the now obsolete Exterior Gateway

Protocol (EGP), as the standard exterior gateway-routing protocol used in the global Internet. BGP solves serious problems with EGP and scales to Internet growth more efficiently. As of January 2006, the current version of BGP, version 4, is codified in RFC 4271.

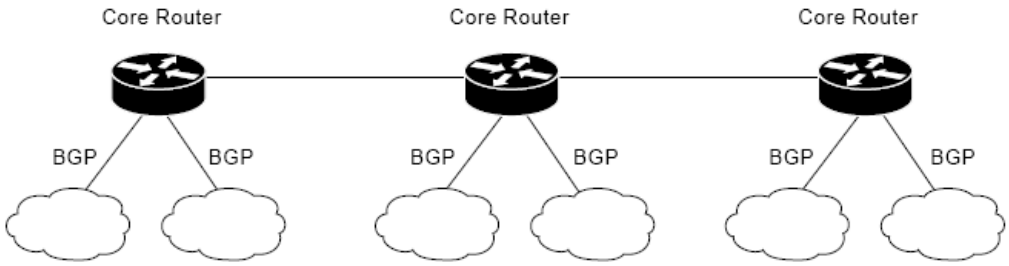


Figure A.3: core routers use BGP to route traffic between AS

Classless Inter-Domain Routing

The Internet running version of BGP supports *Classless Inter-Domain Routing* (CIDR), an evolution of the original classfull addressing scheme of IP, which allows increased flexibility when dividing ranges of IP addresses into separate networks and thereby promotes more efficient use of increasingly scarce IPv4 addresses as well as greater use of hierarchy in address assignments (prefix aggregation), lowering the overhead of the Internet-wide routing. CIDR is principally a bitwise, prefix-based standard for the interpretation of IP addresses. It facilitates routing by allowing blocks of addresses to be grouped together into single routing table entries. These groups, commonly called CIDR blocks, share an initial sequence of bits in the binary representation of their IP addresses. Basically, the Net_ID is not committed to a fixed number of bits anymore, but it can vary its lenght (*prefix lenght*) that in a routing table is specified with a number preceeded by a slash before

the IP address.

$$10.10.1.32 \longrightarrow 10.10.1.32/27$$

For example, a classfull B subnetwork can now be divided into more logically separated network, so the *logic subnetwork* definition arises. The association with the right subnetwork is then done by a bitwise comparison between the packet destination's IP address and the entry in the routing table. An IP address is part of a CIDR block, and is said to match the CIDR prefix if the initial N bits of the address and the CIDR prefix are the same. Thus, understanding CIDR requires that IP address be visualized in binary. Since the length of an IPv4 address is fixed at 32 bits, an N-bit CIDR prefix leaves $32 - N$ bits unmatched, and there are $2^{(32-N)}$ possible combinations of these bits, meaning that $2^{(32-N)}$ IPv4 addresses match a given N-bit CIDR prefix. Shorter CIDR prefixes match more addresses, while longer CIDR prefixes match fewer. A *subnet mask* is a bitmask that encodes the prefix length in a form similar to an IP address - 32 bits, starting with a number of 1 bits equal to the prefix length, ending with 0 bits, and encoded in four-part dotted-decimal format. A subnet mask encodes the same information as a prefix length, but predates the advent of CIDR. CIDR uses *variable length subnet masks* (VLSM) to allocate IP addresses to subnets according to individual need, rather than some general network-wide rule. Thus the network/host division can occur at any bit boundary in the address. The process can be recursive, with a portion of the address space being further divided into even smaller portions, through the use of masks which cover more bits. An address can match multiple CIDR/VLSM prefixes of different lengths, as shown in A.4: the longest matching prefix is chosen as the destination's subnetwork.

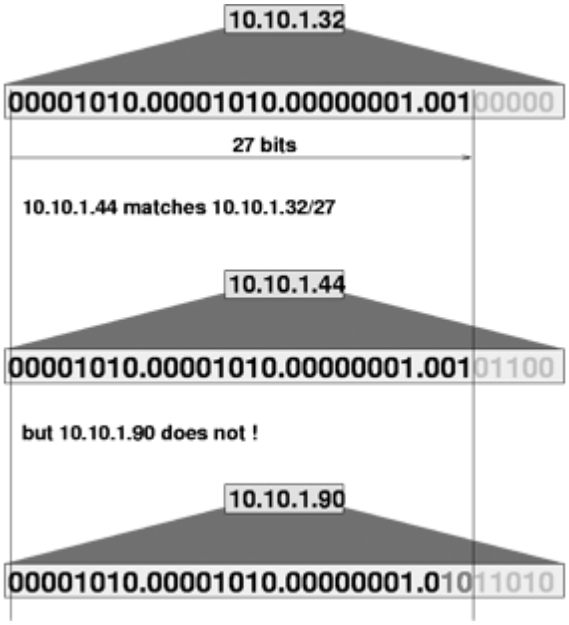


Figure A.4: matching of an IP address to a prefix. The two matching address are assumed being part of the same logic subnetwork.

BGP operations and routing

BGP performs three types of routing: inter-AS routing, intra-AS routing, and "pass-through AS routing".

Inter-AS routing occurs between two or more BGP routers in different AS. Peer routers in these systems use BGP to maintain a consistent view of the internetwork topology.

Intra-AS routing occurs between two or more BGP routers located within the same AS. Peer routers within the same autonomous system use BGP to maintain a consistent view of the system topology. BGP also is used to determine which router will serve as the connection point for specific external autonomous systems. The fact that the BGP protocol can provide both inter- and intra-AS routing services makes it really important for the Internet, that

is indeed an inter-network of AS.

Pass-through AS routing occurs between two or more BGP peer routers that exchange traffic across an AS that does not run BGP. In this case, the BGP traffic did not originate within the autonomous system in question and is not destined for a node in the autonomous system. BGP must interact with whatever intra-AS routing protocol is being used to successfully transport BGP traffic through that autonomous system. Figure A.5 shows an example.

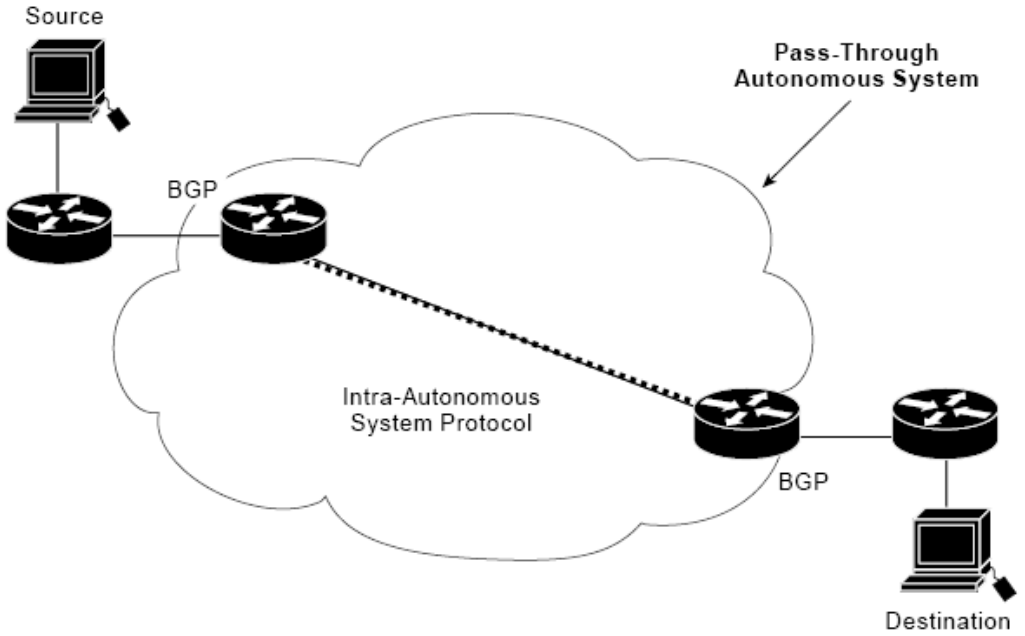


Figure A.5: traffic being routed across a non-BGP AS.

BGP maintains routing tables, transmits routing updates, and bases routing decisions on routing metrics. The primary function of a BGP system is to exchange network-reachability information, including information about the list of AS paths in between, with other BGP systems. That's a key point: a "BGP neighbor"' can even be physically separated by one or more AS that

provide traffic transport, but it's still a logic neighbor. This information can be used to construct a graph of AS connectivity from which routing loops can be pruned and with which AS-level policy decisions can be enforced.

Each BGP router maintains a routing table that lists all feasible paths to a particular network. The router does not refresh the routing table, however. Instead, routing information received from peer routers is retained until an incremental update is received. BGP devices exchange routing information upon initial data exchange and after incremental updates. When a router first connects to the network, it must individuate his neighbors and contact them to make them aware of its presence. Then those others BGP routers exchange their entire BGP routing tables. As time goes on, when the routing table changes, routers send the portion of their routing table that has changed, but not the whole. BGP routers do not send regularly scheduled routing updates, but they verify the connection state by regularly sending "keep-alive" messages (an usual time interval value is 30 seconds).

BGP uses a single routing metric to determine the best path to a given network. This metric consists of an arbitrary unit number that specifies the degree of preference of a particular link. The BGP metric typically is assigned to each link by the network administrator. The value assigned to a link can be based on any number of criteria, including the number of AS through which the path passes, stability, speed, delay, or cost. BGP routing updates between routers advertise only the optimal path to a network, not the metric distance.

A.2.5 The TCP protocol

The *Transmission Control Protocol*(TCP) is one of the core protocols of the Internet: we can undoubtelty state that without it the Internet couldn't have reached today's dimension. Using TCP, applications on networked hosts can create connections to one another, over which they can exchange data or packets. This is the strong TCP's feature that compensates IP's connectionless routing: TCP guarantees reliable (error free) and in-order delivery of sender to receiver data. It also distinguishes data for multiple, concurrent applications (e.g. Web server and e-mail server) running on the same host: TCP connections are estabilished through *ports*, so that a specific application can be reached only through that port, and at the same time, more connections can be accepted on the same port, all related to the same application. The complete address of an application running on a specific host is now

$$port@IP\_address$$

4622@10.87.3.21

and a connection is therefore univocally identified by two paired sockets, as

("2031@10.10.1.31","20@10.10.1.32")

TCP supports many of the Internet's most popular application protocols and resulting applications, including the World Wide Web, e-mail and Secure Shell; in the Internet protocol suite, TCP is the intermediate layer between the Internet Protocol below it and an application above it, thus representing the fourth layer of the OSI model in the simplified Internet layers stack.

The protocol scheme is the following: applications send streams of octets to TCP for delivery through the network, and TCP divides the byte stream into appropriately sized *segments*. TCP then passes the resulting packets to the Internet Protocol, for delivery through a network to the TCP module of the entity at the other end. Figure A.6 shows the complete recursive incapsulation of data from the application layer to the physical layer. TCP checks to

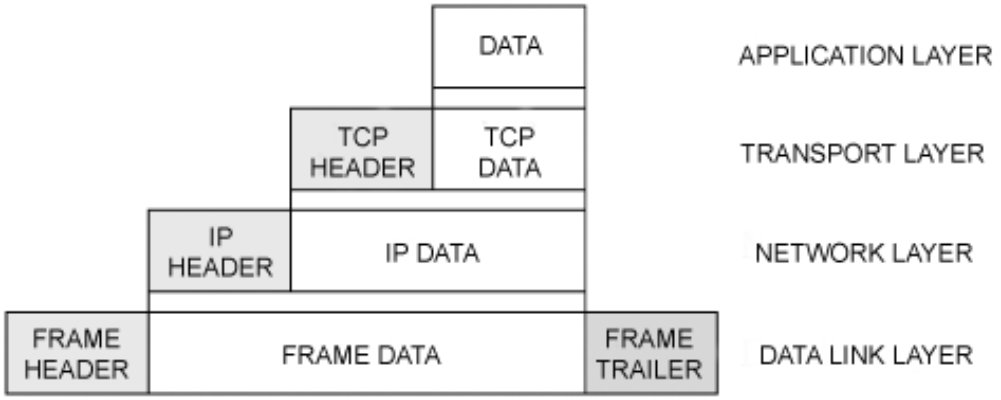


Figure A.6: incapsulation of application data

make sure that no packets are lost by giving each packet a *sequence number*, which is also used to make sure that data are delivered to the entity at the other end in the *correct* order. The TCP module at the far end sends back an *acknowledgement* (ack) for packets which have been successfully received; a timer at the sending TCP will cause a timeout if an acknowledgement is not received within a reasonable round-trip time (or RTT), and the (presumably lost) data will then be re-transmitted. TCP checks that no bytes are damaged by using a checksum; one is computed at the sender for each block of data before it is sent, and checked at the receiver.

TCP also had a very important role in preventing the Internet from col-

lapsing due to unsustainable traffic.

Basically, it embeds a congestion control mechanism that allows a host to send only a certain number of packet without having received an ack relative to the first packet sent. This number (*congestion window*, hereafter) is variable in time, increasing as acks are received without troubles, till the maximum upper bound is reached. As soon as a sent packet timeout runs out without having received his relative ack, the window shrinks to one, dramatically slowing down the transmission rate. The process to reach a large window again takes time (assuming that no other errors occur) during which the host sees a low performance. This mechanism was introduced to reduce the traffic amount on the web in case of a congestion that eventually leads to long packet delivery time, and it really helped the Internet to offer a certain stability, especially in the beginning. But at the same time it's a waste of time if all the mechanism is activated when a congestion didn't occur. That's way several working schemes have been developed and added to this basical structure, such as *fast recovery* or *fast retransmit*.

The congestion window constrain can be used also as a flux control mechanism: in a two way communication, if a host is in lack of resources (e.g. its memory buffer is full and it can't handle any more packets) it can slow down the transmission rate by simply avoiding sending acknowledgements thus reducing the other partecipant's congestion window dimension.

The UDP protocol

User Diagram Protocol (UDP) is another transport protocol, but it differs from TCP in being way more simple and faster. It only provides the same

application addressing through ports and an optional checksum over the user load. Apart from that is closer to IP than to TCP, as it's connectionless and it doesn't offer control on congestion or right order of arrival of packets.

Nevertheless, without the overhead of checking if every packet actually arrived, UDP is faster and more efficient for many lightweight or time-sensitive purposes. Also its stateless nature is useful for servers that answer small queries from huge numbers of clients, being sensitively faster than TCP (basically because no connection must be set up nor acknowledge be sent). Lacking reliability, UDP applications must generally be willing to accept some loss, errors or duplication. Most often, UDP applications do not require reliability mechanisms: streaming media, real-time multiplayer games and voice over IP (VoIP) are examples of applications that often use UDP.

Lacking any congestion avoidance and control mechanisms, network based mechanisms are required to minimize potential congestion collapse effects of uncontrolled, high rate UDP traffic loads. In other words, since UDP senders cannot detect congestion, network-based elements such as routers using packet queueing and dropping techniques will often be the only tool available to slow down excessive UDP traffic.

While the total amount of UDP traffic found on a typical network is often on the order of only a few percent, numerous key applications use UDP, such as *Domain Name System* (DNS), the *Routing Informtation Protocol* (RIP), or *Dynamic Host Configuation Protocol* (DHCP).

Appendix B

Ringraziamenti e momento di digressione intellettuale

Non capita tutti i giorni di scrivere qualcosa che si sa essere destinato ad una stampa semi–professionale e ad una copertina rigida, che per eleganza ed importanza emotiva (ma soprattutto per il *peso*) ben si presta al ruolo di pietra miliare tra un prima ed un dopo, di chiusura di uno dei capitoli del grande libro che chiamiamo vita[1]. Il "momento di digressione intellettuale" del titolo è una citazione del grande Ivo Leonard Furano.

Questa tesi è stata preparata in tempo per la discussione estiva anche e soprattutto grazie al fatto che sono riuscito a rispettare una tabella di marcia serratissima che "spannava" su un anno e mezzo di tempo: il mio periodo Erasmus in Francia ha costituito un rallentamento dal punto di vista accademico, quindi da febbraio 2005 a luglio 2006 non c'è stato un attimo di tregua, riuscendo anche a confezionare una media niente male. Ma da dove ho attinto motivazione e forza di volontà?

L'Erasmus è stato, senza dubbio, la *più importante* esperienza della mia vita. I cambiamenti che ha introdotto nel mio modo di vedere le cose, nel mio modo di giudicare, nel mio modo di comportarmi, nel mio modo di pensare e di sognare sono stati così grandi e forti che ancora adesso non sono sicuro di averli capiti ed assorbiti appieno. Ma uno dei propositi era già chiaro al momento del rientro: finire gli studi nel più breve tempo possibile e nella maniera più seria possibile, per ritrovarmi in mano qualche strumento

[1] citazione di Eddy Carrillo, compagno erasmus nato e vissuto su un'isoletta nel Borneo da madre francese e padre venezuelano (cuoco); adesso vive in Olanda da qualche anno e parla correntemente 5 lingue inclusa quella nativa dell'isoletta del Borneo.

e conoscenza in più per potermi tuffare nuovamente alla scoperta di nuove vie, nuove strade, le opportunità all'estero... è sicuramente questo che mi ha più aiutato nell'impegno!

Passando un attimo ai ringraziamenti veri e propri, volevo anzitutto menzionare tutti (o quasi) i compagni di università propriamente detti con i quali ho condiviso questi sei anni accademici. Chi prima, chi dopo, chi a lungo, chi solo per qualcosa; chi non sento più già da tanto, chi spero di continuare a sentire ancora a lungo, sono tutti in figura B.1!

Luigi Salamandra, Stefano Tauriello, Davide Senna, Marco Violano, Florinda Giorgio, Hermes D'Ottavio, Tommasello, Federica Teodori, Daniele Romano, Alessandro Lollobattista, Massimo "Benson", Luca Migliori, Danilo Zaccariello, Dario Formichetti, Ivo Leonard Furano, Andrea Varamo, Gianni Zitelli, Enrico Zanza, Carlo Muti, Mario Nicolini, Andrea "Nonno", Valentina Facco, Yuppi Passamonti, Marco Bonola, Elisa Serrecchia, Clara Abate, Federico Saccone, Massimo di Santo, Andrea Mancinelli, Gianluca "Artale" Vastaroli, Luigi Vesce, Francesco Vestri Boncore, Trilly, Alessandra Universitor, Yuri Faenza, Roberto Tachis, Federica Pazzola, Luca Barbaro, Stefano Cassella, Luca Ninivaggi, Giacomo Ulisse, Simone Placidi, Paolo Isidori, Cristina Bandel, Fabrizio Storch, Raffaele Papitto, Ruben Faiola, Luigi Pagnozzi, Mattia Musella, Lucia Demofonte, Saverio Proto, Lorenzo Bracciale, Luca Fiengo, Valerio Marcelli, Pamela Rammauro, Giulia Rossi, Dimitri Scarana, Raffaele Bianchini.

Figure B.1: short list of university mates; most of them are male names.

Per questa tesi in particolare, volevo ringraziare anzitutto lo staff di Ylichron e Limor per tutti i momenti passati insieme, per i taralli, per fluent, per lo yoga, per i caffé, per i buoni mensa (ah no, quelli erano i miei), per non avermi fatto perdere la navetta delle 18:08 (17:56), per non avere ucciso tutti i miei processi spalmati su trudy, per la grandissima disponibilità, per le passeggiate avanti e indietro dalla mensa ("..e dai, prendiamo la navetta... dai, su..."). Una menzione particolare va a Rocco Casilli, che oltre ad essere invischiato in losche vicende politiche è anche l'amministratore di sistema più competente e disponibile, un aiuto essenziale in tanti piccoli e grandi intoppi [`if (problema) then rocco(problema);`]. Dal momento che ho già occupato troppo spazio su trudy, quest'ultima dedica ve la faccio in binario nella tabella B. (suggerito l'uso di hexedit e la lettura in una stringa)

```
01100001  01101100  01101100  01100001  00100000  01100110
01100001  01100011  01100011  01101001  01100001  00100000
01110110  01101111  01110011  01110100  01110010  01100001
00100001
```

Sandro Meloni! Pigiaminux! La miglior compagnia che potessi desiderare per la mia stanza! Grazie per aiutarmi a far sopravvivere le piante, per lavorare insieme con i Metallica a cannone (grazie anche a loro), per cercare di occupare meno memoria possibile con le nostre tabelle `n` $\times$ `n`, per le risate che ci siamo fatti con gli altri compagni di stanza, l'indiano Rashdip, il cinese Xuan Jin e Simone. Grazie infine per l'aiuto nelle fasi finali del codice (ora è in mano tua, amigo!). Yes, thanks to you too, Xuan and Rashdip, room–mates from the outer world, for having endlessy endured our voices (Sandro's and mine), probably without catching a word. "He edited it!"

Restando in ENEA, ma coprendo anche tutto il percorso casa–ENEA e quasi tutto il percorso parcheggio–casa (!!!), da ringraziare c'è il mio co–relatore Vittorio Rosato. Le cose si fanno complicate qui... oltre a chiamarmi ancora una volta in ENEA, Vittorio si è anche occupato degli spostamenti mattutini e pomeridiani, conditi dalle più svariate discussioni! Nonostante ci prenda in giro, sappiamo che ha un debole per gli "spazzoloni" e sono sicuro che ci rincontreremo in fricchettonia!! Grande Vitt, grazie per tutto. Un esempio da seguire, magari un po'meno workaholic!!

Ci sono tante persone che mi sono state vicine, prima ra tutte Alutxa, che ha sopportato di buon grado tutti i miei momenti di "assenza dal real world" dovuti allo stress e che ho avuto accanto.. anche in orma di statuina! Tutti gli amici che sento vicino, malgrado molti siano lontani, e penso ad Alex, ad Albert, a Maurits e a tanti altri... tra i vari amici qui a Roma posso sempre contare sul Paoletto, e sono così contento di aver condiviso anche quest'anno post–Erasmus con la Fra, l'amicizia con lei è sicuramente la cosa più bella che l'Erasmus mi ha regalato! Una menzione speciale la volevo dedicare al Dodo. Qualche anno fa mi hai mostrato cosa volesse dire avere un amico che si interessa davvero a te, insegnandomi tante cose che sono rimaste con me. Di persone così non ce ne sono tante nella mia vita. Volevo dedicare due righe anche alle ragazze che hanno condiviso con me qualche passo sul cammino, ognuna a modo suo mi ha supportato e incoraggiato, e mi pareva giusto ringraziarle qui, anche se probabilmente non lo sapranno mai. Voglio ringraziare anche Dana, in Arizona: l'idea di partire dopo la laurea è stata un primo, concreto passetto nel sogno di viaggiare e scoprire dopo l'università!

Ma su tutti ringrazio la mia famiglia. I miei genitori, anche non partecipando in maniera diretta a tutte le cose di cui sopra, sono coloro che lo hanno reso possibile, dalle amicizie, all'università, ai viaggi ed ai sogni. Grazie per tutte le libertà che ho avuto, per le disponibilità e gli incoraggiamenti. Gra-

zie a mia sorella, con la quale abbiamo cercato di stare vicino il più possibile, e solo il pensiero di allontanarsi mi fa star male.

Volevo scegliere il testo di una canzone da inserire: non può che essere una canzone dei Litfiba, per tutto ciò che han rappresentato nella mia vita. In particolare si tratta di "ElettroMacumba", title–track dell'album del periodo nel quale cominciai a stringere i rapporti, lavorando anche con loro. Anche questa è stata una grande esperienza, e questa canzone, che ha anche ben descritto alcuni episodi nella mia vita, ha Internet come suo tema. Cosa di più adatto per la chiusura di questa tesi?

Elettro macumba elettro
Elettro macumba elettro
Adesso c'è il nuovo cyber re
Sorveglierà ogni tua connessione
Non ti fidare mai, cliccando dici chi sei
Forse non sai che tu sei nel suo file

ritornello:
Elettro macumba elettro
Elettro macumba elettro
Scatena il voodoo digitale
Nel bunker di chi sa tacere
Elettro macumba elettro
Auuuuuu, auuuuuu, auuuuuu

Gira con te il nuovo cyber re
Ti scoverà nel database mondiale
Rimpiangerai la penna, l'uomo e l'errore
Resisterai con tutto quello che hai

rit.

Ti ascolterà un entità virtuale
Registrerà la tua confessione
Ti pentirai persino del cyber sex
Ti assolverà l'elettrotrinità
Elettro macumba elettro
Elettro macumba elettro

Bibliography

[1] D.J. Watts, S.H. Strogatz, Nature **393** (1998) 440.

[2] A.–L. Barabàsi, R, Albert, Science **286** (1999) 509.

[3] R. Pastor-Satorras, A. Vespignani, Evolution and Structure of the Internet: A Statistical Physics Approach, Cambridge University Press, Cambridge, 2004.

[4] M.L. Sachtjen, B.A. Carreras, V.E. Lynch, Phys. Rev. E **61** (2000) 4877.

[5] S.H. Strogatz, Nature **410** (2001) 28.

[6] A.–L. Barabási, R. Albert, Rev. Mod. Ohys. **74** (2002) 47.

[7] S.Boccaletti et al., Phys. Reports **424** (2006). 175.

[8] S. Wasserman, K. Faust, Social Networks Analysis, Cambridge University Press, Cambridge, 1994.

[9] J. Scott, Social Network Analysis: A Handbook, 2nd ed., Sage Publications, London, 2000.

[10] D.J. Watts, Small Worlds: The Dynamics of Networks between Order and Randomness, Princeton University Press, Princeton, NJ, 1999.

[11] V. Latora, M. Marchiori, Phys. Rev. Lett. **87** (2001) 198701.

[12] V. Latora, M. Marchiori, Eur. Phys. J. B **32** (2003) 249.

[13] C.L. Freeman, Sociometry **40** (1977) 35.

[14] L.C. Freeman, Social Networks **1** (1979) 215.

[15] E. W. Dijkstra, Numerische Math **1** (1959) 269.

[16] D.B. West, Introduction to Graph Theory, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[17] D.J. Watts, Six Degrees: The Science of a Connected Age, Norton, New York, 2003.

[18] I. Farkas, I. Derényi, A.-L. Barabási, T. Vicsek, Phys. Rev E **64** (2001) 26704.

[19] K.-I. Goh, B. Kahng, D. Kim, Phys. Rev. E **64** (2001) 051903.

[20] D. Vukadinovic, P. Huang, T. Erlebach T. ETH Zurich, TIK Report **118**, July 2001.

[21] V. Rosato, F. Tiriticco, Europhys. Lett. **66** (2004) 471.

[22] M. Mehta, Random Matrices, Academic Press, New York, 1995.

[23] Mohar B., Discrete Math., **109** (1992) 171.

[24] Pothen A., Simon H. D. and Liou K. P., SIAM J. Matrix Anal. Appl., **11** (1990) 430.

[25] Seary A. J. and Richards W. D., in Proceedings of the International Conference on Social Networks, edited by Everett M. G. and Rennolls K., Vol. **1**: Methodology (1996) 47.

[26] P. Erdos, A. Rényi, Publ. Math. Debrecen **6** (1959) 290.

[27] Holme P. and Kim B. J., Phys. Rev. E, **65** (2002) 026107.

[28] V.M. Eguíluz, K. Klemm, Phys. Rev. Lett. **89** (2002) 108701.

[29] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in ACM SIGCOMM 1999, Boston, MA, USA, Aug./Sept. 1999.

[30] R. Govindan and H. Tangmunarunki, "Heuristics for internet map discovery," in IEEE Infocom 2000, Tel-Aviv, Israel, Mar. 2000, 1371.

[31] L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos, "A simple conceptual model for the internet topology," in Global Internet, Nov. 2001.

[32] P. Barford, A. Bestavros, J. Byers, and M. Crovella, "On the marginal utility of network topology measurements," in ACM SIGCOMM IMW '01, San Francisco, CA, USA, Nov. 2001.

[33] A. Broido and K. Claffy, "Internet topology: connectivity of IP graphs," in SPIE International symposium on Convergence of IT and Communication '01, Denver, CO, USA, Aug. 2001.

[34] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "The origin of power-laws in internet topologies revisited," in IEEE Infocom 2002, New-York, NY, USA, Apr. 2002.

[35] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," in ACM SIGCOMM '02, Pittsburgh, PA, USA, Aug. 2002.

[36] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie, "Sampling biases in ip topology measurements," in IEEE INFOCOM '03, San Francisco, CA, USA, Apr. 2003.

[37] S. Bar, M. Gonen, and A. Wool, "An incremental super-linear preferential internet topology model," in PAM '04, Antibes Juan-les-Pins, France, Apr. 2004.

[38] "University of Oregon Route Views Project," http://www.antc.uoregon.edu/route-views/.

[39] see `www.netdimes.org` and references therein.

[40] Y. Shavitt, E. Shir, "Dimes: let the internet measure itself", June 2005.

[41] "SETI@Home," http://setiathome.berkeley.edu/.

[42] "Distributed.net," http://www.distributed.net/.

[43] M. Dharsee and C. Hogue, "Mobidick: A tool for distributed computing on the internet", in Heterogeneous Computing Workshop '00, Cancun, Mexico, May 2000.

[44] J. Charles Robert Simpson and G. F. Riley, "Neti@home: A distributed approach to collecting end-to-end network performance measurements," in PAM '04, Antibes Juan-les-Pins, France, Apr. 2004.

[45] A. Broido and k. claffy, "Internet topology: connectivity of ip graphs," in Proceedings of SPIE, 2003.

[46] Z. Mao, D. Johnson, J. Rexford, and R. K. J Wang, "Scalable and accurate identification of as-level forwarding paths," in INFOCOM, 2004.

[47] T. Ohira, R. Sawatari, Phase transition in a computer network traffic model, Phisical Review E. **58** (1998) 193

[48] N. Blefari Melazzi, Internet: architettura, principali protocolli e linee evolutive, McGraw–Hill 2006.

[49] Wikipedia, the free encyclopedia, http://www.wikipedia.org/.