Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Mestrado em Engenharia Informática

Scale-free Networks and Scalable Interdomain Routing

Pedro Miguel Fonseca Rodrigues
(aluno nº28083)

2º Semestre de 2009/10

Setembro de 2010

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Scale-free Networks and Scalable Interdomain Routing

Pedro Miguel Fonseca Rodrigues
(aluno nº28083)

Orientador: Prof. Doutor José Legatheaux Martins

Trabalho apresentado no âmbito do Mestrado em Engenharia Informática, como requisito parcial para obtenção do grau de Mestre em Engenharia Informática.

2º Semestre de 2009/10
Setembro de 2010

Acknowledgements

This page is intentionally written in portuguese.

Agradecimentos são devidos a várias pessoas que possiblitaram realizar esta dissertação. Em primeiro lugar gostaria de agradecer ao Prof. José Legatheaux Martins pelo seu apoio e orientação durante toda a dissertação, e pela forma como me motivou para chegar o mais longe possível. Gostaria de destacar o seu entusiasmo relativamente ao trabalho de investigação que me cativou para prosseguir uma carreira de investigação científica.

Gostaria de agradecer à minha família, em especial aos meus pais e à minha irmã, que sempre me apoiaram incondicionalmente e que sem o seu esforço e ensinamentos não conseguiria ter atingido os meus objectivos. Adicionalmente, gostaria de agradecer à minha namorada Catarina que ao longo dos anos tem sido um apoio importante para que continue motivado e empenhado no caminho que tracei, e pela sua paciência para ouvir-me falar sobre assuntos fora da sua área de estudo.

Por fim, gostaria de agradecer ao Departamento de Informática pelo apoio através de Bolsas de Estudo que me permitiram ter a experiência de leccionar aulas práticas, bem como pelo apoio financeiro ao longo do Mestrado.

Resumo

O crescimento exponencial da Internet, devido ao seu enorme sucesso, evidenciou várias limitações do desenho actual ao nível da arquitectura e do encaminhamento, tais como escalabilidade e convergência, falta de suporte a mecanismos de *traffic engineering*, mobilidade, diferenciação de rotas e segurança.

Alguns destes problemas surgem das opções no desenho da actual arquitectura, enquanto outros são causados pelo esquema de encaminhamento interdomínio - BGP. Como seria extremamente difícil solucionar os problemas enunciados anteriormente, tanto ao nível da arquitectura da Internet como do esquema de encaminhamento, vários investigadores afirmam que uma solução só será alcançada através de uma nova arquitectura e de um novo esquema de encaminhamento.

Uma nova estratégia de encaminhamento emergiu a partir de estudos sobre redes de larga escala, adequada a um tipo especial dessas redes cujas características são o independentes da sua escala: redes *scale-free*. Utilizando a estratégia *greedy routing* um nó encaminha uma mensagem para um dado nó destino somente utilizando informação relativa ao mesmo e aos seus vizinhos, escolhendo aquele que se encontra mais *próximo* do destino. A estratégia garante as seguintes propriedades notáveis: estado mantido da ordem do número de vizinhos; não requer que os nós troquem mensagens para efectuarem encaminhamento; os caminhos escolhidos são os mais curtos.

Esta dissertação tem como objectivos: aprofundar a problemática acima apresentada, estudar a configuração da Internet enquanto rede scale-free e propor uma definição preliminar de um esquema de encaminhamento utilizando a estratégia de *greedy routing* a fim de ser utilizado em encaminhamento interdomínio.

Palavras-chave: Arquitectura da Internet, BGP, Esquemas de Encaminhamento, Redes *Scale-free*, *Greedy Routing*

Abstract

The exponential growth of the Internet, due to its tremendous success, has brought to light some limitations of the current design at the routing and arquitectural level, such as scalability and convergence as well as the lack of support for traffic engineering, mobility, route differentiation and security.

Some of these issues arise from the design of the current architecture, while others are caused by the interdomain routing scheme - BGP. Since it would be quite difficult to add support for the aforementioned issues, both in the interdomain architecture and in the in the routing scheme, various researchers believe that a solution can only achieved via a new architecture and (possibly) a new routing scheme.

A new routing strategy has emerged from the studies regarding large-scale networks, which is suitable for a special type of large-scale networks which characteristics are independent of network size: scale-free networks. Using the *greedy routing* strategy a node routes a message to a given destination using only the information regarding the destination and its neighbours, choosing the one which is *closest* to the destination. This routing strategy ensures the following remarkable properties: routing state in the order of the number of neighbours; no requirements on nodes to exchange messages in order to perform routing; chosen paths are the shortest ones.

This dissertation aims at: studying the aforementioned problems, studying the Internet configuration as a scale-free network, and defining a preliminary path onto the definition of a greedy routing scheme for interdomain routing.

Keywords: Internet architecture, BGP, Routing Schemes, Scale-free Networks, Greedy Routing

Contents

1. Introduction

The ARPANET was designed in the mid 60's to connect military bases and research departments of the USA government. Since its early stages, it was based on radical new architectural principles: packet switching, stateless core, minimal structural bindings as well as complexity sent to the edge. A decade after its design, more universities and other institutions joined the ARPANET and it grew to about 100 nodes. In the late 80's, the NSFNET was created as a centralised backbone to which regional and academic networks would connect. A few years posterior to 1990, the NFSNET was replaced by commercial provider networks as a result of the emerging of public-accessible Internet. Due to its major commercial success, the Internet has grown exponentially until 2001 and has been growing super linearly since then [17].

One of the consequences of the transition from the NSFNET backbone to the public Internet was the design of a new architecture, where Internet Service Providers (ISPs) exchange packets directly to reach every network connected to the Internet, instead of relying on the governmentally funded and centralised NSFNET backbone. The Internet is now composed by a set of autonomously administrated networks, known as Autonomous Systems (ASes). An AS is an institution that manages its private network(s) and provides Internet access to its hosts by being connected through other AS(es) to the rest of the Internet. Additionally, each AS has one or more unique IP prefixes, *i.e.*, a range of IP addresses, from which it assigns IP addresses to its hosts. ASes run an interdomain routing protocol called Border Gateway Routing Protocol (BGP) to exchange reachability and dynamic information, *i.e.*, link failures and availability, regarding their prefixes.

BGP was initially defined when the Internet had a few hundred ASes and was fairly limited to academic usage. Over the years, BGP has been updated to follow the evolution of the Internet. Although BGP continues to support the current functioning of interdomain routing, there are some critical aspects that it does not manage quite well. Some of them are caused by the continually increasing number of ASes, which are more than 30 000 nowadays. Other issues arise from the current Internet design.

In the last years, a significant part of the companies connected to the Internet have chosen to be connected through more than one ISP in order to improve connectivity reliability and traffic distribution. Thereby, such companies are no longer part of the private network of their only ISP and have to participate in interdomain routing, so that they can be reached through the different ISPs to which they are connected. In addition, it is also necessary that these multi-homed ASes obtain their own IP prefix. This trend has significantly increased the number of ASes, and more specifically the number of different IP prefixes, and, as a consequence, the number of routing entries and routing events has *exploded*. At the time of writing, each border BGP router knows more than 300 000 different destinations. Traffic engineering techniques to balance traffic also contribute to those increases. Consequently, these issues pose a threat to the scalability of BGP.

A mechanism which delays the propagation of routing updates is present in BGP in order to control the amount of updates exchanged per unit of time. Although it accomplishes its purpose,

it also increases the convergence time of BGP after a failure, reaching up to tens of minutes. Some alternatives to this delay mechanism, that reduce the number of messages exchanged as well as the convergence time, have been proposed in the literature. However, BGP and the current architecture have other critical issues: lack of support for traffic engineering, for security, for quality of service and for mobility.

Since most of these critical issues arise from the current Internet design, and it would be utterly difficult to support it in the current architecture, some researchers believe that time has come to define a new one for the *future* Internet. Moreover, in the current architecture an IP address does not only uniquely identifies a host but it also identifies its location. One common factor of the new architectures that have been proposed in the literature is the separation of the current IP address scheme into two address spaces: one to represent host location and other for host identification.

The routing algorithm used by a network is somehow orthogonal of its architecture. This claim is not always true in what concerns the network scale. Therefore, one cannot completely avoid reassessing the routing scheme if the Internet is redesigned. If the new architecture does not solve most scalability issues, in what concerns routing, some form of scalable routing must be considered in parallel with the Internet architecture redefinition. Thus, forms of scalable routing are a relevant contribution for this discussion.

In parallel with the definition of proposals to the current and future Internet, there has been studies regarding the structure and topological characteristics of the Internet graph. Surprisingly, most of these are common to other large-scale networks, opening the possibility of applying mechanisms from these networks to the Internet.

The study of large-scale networks has opened a new exciting field known as Networking Science. Researchers acting on this area have proposed a new routing strategy known as *greedy routing* [34]. According to this routing strategy, each node only knows its characteristics and the characteristics of its neighbours. With this information and the characteristics of a given destination, a node is able to *greedily* route a message by selecting the direct neighbour closest to the destination.

1.1 Goals

A greedy routing scheme needs small routing state, since each node only needs information regarding its directly connected neighbours instead of all possible destinations as in traditional routing schemes, *e.g.*, BGP. Additionally, if successful, a greedy routing scheme ensures that the majority of chosen paths are the shortest ones. It can also be very robust, since changes to node population and connectivity are locally handled.

The success of a greedy routing scheme depends on the type of network to which it is applied. On paper [6] it is studied the suitability of a special type of networks: scale-free networks. These networks have thousands of nodes and are characterised by presenting properties which do not depend on their size.

Taking in consideration the state of affairs, the goal of this work is twofold: *a*) perform a study on topological characteristics of the Internet AS graph to confirm it is a scale-free network; *b*) devise a preliminary greedy routing scheme for interdomain routing. The application of such scheme to the Internet AS graph must comprise several main components:

- a mapping of the AS topology into a coordinate space, *i.e.*, defining an embedding for the AS graph;
- a distance function;
- a greedy routing strategy/algorithm guided by the above components.

1.2 Contributions

The contributions of this work is an updated study regarding the topological characteristics of the Internet AS graph as well as the preliminary definition of a scalable routing scheme.

The updated study relies upon a recent snapshot of the Internet AS graph [9] to analyse the following distinguishable properties of scale-free networks:

- assortativity;
- clustering;
- *small-world* property;
- network navigability restrictions;
- network growth and construction algorithm.

On top of that, we presented a taxonomy for rescalled AS graphs [58] that is supported by the scale-free topological properties of the Internet AS graph. This taxonomy was used to evaluate assortativity and clustering on the snapshot of the Internet AS graph [9].

Furthermore, the greedy routing scheme has the following characteristics:

Small routing state: To route packets to any destination in the network, a node needs little routing information, in the order of the number of its neighbours. As opposed to BGP that requires each router to maintain routing information proportional to the number of possible destinations. Only configuration messages are rarely exchanged between pair of ASes, hence the number of updates that routers have to process is almost negligible. Thus, greedy routing has low maintenance requirements of the routing table.

Small routing stretch: Most of the paths chosen by the greedy forwarding mechanism are the shortest paths to a given destination. Since the routing stretch is small, the greedy routing scheme ensures low global resource consumption as well as low end-to-end latency. Therefore, it provides good end-to-end message delivery.

Expressiveness: It is possible to define preference mechanisms similar to Local Preference and Multi-Exit Descriminator ones in BGP. An AS can also express to which customers (destinations) it allows the usage of a peering link. The configuration of these mechanisms rely upon messages that are only exchanged between directly connected pairs of ASes.

In addition, we present an initial discussion on the components of an architecture for interdomain routing that uses our greedy routing scheme.

1.3 Document Structure

This document is structured as follows. Chapter 2 introduces the current state-of-the-art of interdomain routing. It begins with a description of the currently used protocol for interdomain routing (Border Gateway Protocol) followed by a discussion of its current problems and short-term fixes proposed in the literature.

The requirements of a future internet architecture for interdomain routing are discussed in chapter 3. Subsequently, some new architectural proposals for interdomain routing are presented along with alternative routing schemes to BGP.

Chapter 4 first describes the most important topological properties of scale-free networks. Then the underlying concepts regarding greedy routing are detailed as well as some routing proposals using the greedy routing strategy.

In chapter 5 we present a study concerning the topological properties of a snapshot of the Internet AS graph made available by CAIDA [9]. In addition, we illustrate how the topological properties of the Internet AS graph can be used to annotate rescaled AS graphs, *i.e.*, graphs which maintain (most of) the main topological characteristics, though having a different size.

Chapter 6 presents in detail a greedy routing scheme for the Internet AS graph relying upon an euclidean metric space. A comparison between our greedy routing scheme and BGP is presented, followed by a discussion on the design foundations of a new architecture for interdomain routing. Lastly, the closing remarks and future work are presented in chapter 7.

2. Current State-of-the-Art of Interdomain Routing

In this chapter we present a brief overview of the current interdomain routing protocol - BGP, its main characteristics, advantages and drawbacks. We also briefly review some of the main proposals put forward to deal with the main problems already identified.

2.1 BGP

The Internet is structured as a partition of several sub-networks that are designated as Autonomous Systems (ASes). Border Gateway Protocol (BGP) is the routing protocol used for interdomain routing, *i.e.*, routing among ASes on the Internet. It uses a path-vector distance algorithm to distribute reachability information regarding IP prefixes among ASes.

However, while in Interior Gateway Protocols (IGPs) each node sends all information present in the routing table to its neighbours, BGP firstly applies filters before sending reachability information to a neighbour. A filter consists of a set of policies that are specific for a particular neighbour. For instance, a customer AS does not advertise routes learned from one of its providers to the others. This is done so that the customer AS is not used to route traffic among its providers.

The first version of BGP was defined in 1989 (RFC 1105) to substitute Exterior Gateway Protocol (EGP). EGP needed a tree-structured network in order to exchange reachability information among ASes. By using a path-vector distance algorithm, BGP-1 could construct a graph of connectivity without loops. However, in BGP-1 routers could not automatically find neighbours, it required them to be configured manually. Additionally, in the first version it was also possible to apply policy decisions which may influence the referred graph.

In BGP-2 (RFC 1163) the limitation of manual configuration of neighbours was abandoned along with considerable changes on the messages formats. BGP-3 (RFC 1267) introduced a mechanism to solve connection collision, *i.e.*, when two BGP neighbours initiate a TCP connection at the same time, among other changes. In 1994, BGP-4 (RFC 1654, 1771, 4271) introduced Classless Interdomain Routing (CIDR) along with aggregation support. These are the main mechanisms that have been making BGP capable of managing interdomain routing despite Internet growth. Before CIDR was defined, the IP address space was divided in IP prefixes of four sizes (8, 16, 24, 32 bits) corresponding to respectively four classes (A, B, C, D). An AS which needed more than 256 IP addresses (IP prefix of class C) had to request a IP prefix of class B that comprised 65536 different IP addresses. CIDR allows the construction of IP prefixes having different sizes, *e.g.*, 12 bits. With the aggregation mechanism, updates regarding sub-prefixes derived from a given prefix can be aggregated into one update regarding that prefix.

BGP routers establish a session to exchange messages containing reachability information. Update messages, which contain reachability information, are sent to advertise or withdraw a

given destination. Each destination is advertised as a route to it, which contains an AS path from the origin to the destination and a list of attributes that characterise the route. The list of attributes is detailed in subsection 2.1.3. As mentioned above, the path is used to avoid loop advertisement and can be used to perform policy based decisions.

Each router maintains two collections of routes: a Forwarding Information Base (FIB) and a Routing Information Base (RIB). The RIB consists of all the advertised routes from the neighbour routers, whereas the FIB contains the best route to each received destination, computed by the route-selection algorithm from the RIB. Since BGP does not support multi-path routing, if multiple routes are considered as the best ones, a tie-break rule is applied to select the one to be put in the FIB. Moreover, when a router receives an update message, if the update is not invalidated by incoming filters, it executes a set of actions depending on the type of update message, *i.e.*, a message to advertise or to withdraw routes:

- Advertisement
    - the received route replaces the one for the same destination in the FIB if, according to the route attributes, the received route is better than the current one;
    - otherwise, the received route is only added to the RIB;
- Withdrawal
    - the route is withdrawn from the RIB and, if it is in the FIB, the route is removed and the route-selection algorithm picks a new best route to that destination from the available ones in the RIB.

After being processed, an update message is propagated under the following conditions:

- route-selection algorithm considers the received route better than previously existing one in the routing table for the same prefix;
- the new route is not invalidated by the outbound filter, specific for each neighbour.

In accordance to the policy fillters set up by its administrators, a router does not propagate updates received from a provider to another provider, or from a peer to another peer as peering relationships are not transitive, so that it does not provide transit to non-customer ASes. Only own prefixes and the ones received from clients are propagated to non-customer ASes.

The following sub-sections detail the main aspects concerning BGP.

2.1.1 iBGP - eBGP

Exterior BGP (eBGP) is used to exchange reachability among ASes. The routes learned from neighbours have to be injected in Interior Gateway Protocols (IGPs) such as OSPF and RIP. This is done by a set of routers, internal to an AS, using interior BGP (iBGP). The most common iBGP architectures are fully-mesh and route reflection. Confederations are another approach which consists of dividing an AS in sub-ASes, each one with a private AS number, and using

eBGP among sub-ASes. iBGP is used in sub-ASes. This is only used when an AS has a big infrastructure where common iBGP architectures do not scale.

In the following sub-sections the main aspects of e-BGP are specified, which will be mentioned only as BGP.

2.1.2 BGP Session and Messages

BGP routers establish a session using TCP in order to exchange reachability information. The main messages exchanged by BGP routers are:

- `open`: after establishing a TCP connection both routers firstly send an open message to initiate a BGP connection, then they exchange parts of their routing table (FIB) that are validated by policy filters;
- `update`: an update message is sent when a route is no longer reachable, when a non-reachable route becomes active or when a new route is discovered;
- `keep-alive`: in order to maintain a session both routers have to periodically send keep-alive messages;
- `notification`: reports errors or closes the BGP connection.

When a router does not send keep-alive messages for a period higher than the `hold-down timer` it is considered to be inoperative. Hence, the router that detects the failure sends an update message withdrawing routes received from the failed router. When a failed router recovers, it restarts the previous BGP sessions with its neighbours. In that case the neighbours send an update message informing the new routes received from the recovered router, according with the update message propagation conditions previously described.

In order to control the rate of update messages, each advertisement regarding a prefix has to be separated by a Minimum Route Advertisement Interval (MRAI). The current default values of MRAI are 30 seconds on eBGP sessions and 5 seconds on iBGP sessions, as defined in the RCF 4271. Recently, it has been suggested using MRAI values of 5 seconds or less on eBGP sessions and 1 second or less on iBGP sessions. For more details, one can see the discussion on this subject in section 2.3.

2.1.3 Route-Selection Algorithm

When multiple routes to a given destination are available, the route-selection algorithm is used to pick the best one. The most important attributes used to select a route are:

Local Preference
: Local Preference is a local attribute to an AS that is used to define which route is preferred when multiple routes to the same destination are available, *e.g.*, set a Local Preference value to the routes learned from a customer higher than the Local Preference value to the routes learned from a provider.

AS Path
: The AS Path contains all the AS numbers of the ASes which compose the path between the source and the destination. It is used to prevent routing loops and can be used to apply policy decisions based on the presence of certain AS(es).

Next Hop
: IP address of the entry router for the next AS in the AS Path.

Multi Exit Discriminator
: This attribute is used by an AS to give a hint to a neighbour on which router it prefers as the entry point, when multiple connections points between two ASes are available.

Origin
: The origin attribute indicates how a router learned a particular route. The attribute can have one of the following three values:

  - `IGP` - the route is local to the originating AS;
  - `EGP` - the route was learned from eBGP;
  - `Incomplete` - the origin of the route is unknown or the route was learned from other mean.

Communities
: A route can contain one or more community values which are represented as a 4-bytes value that is structured as follows: the first two bytes represent an AS number and the two last bytes represent the semantic of the community. Therefore, each AS can define 2^{16} different communities. Each community allows a group of ASes, or a single AS, to express a given action to be performed automatically. Communities are divided in two types: well-known communities and private communities. The most common well-known communities are:

  - `No_Export`: do not advertise the route to eBGP peers;
  - `Local_AS`: the route should not be advertised outside the AS, but can be advertised to sub-ASes in confederation architectures;
  - `No_Advertise`: do not advertise the route to any eBGP and iBGP peers;
  - `Internet`: the route can be advertised to any peer.

  Private communities are specific for a given AS, *i.e.*, two ASes can have a community with the same semantic value that is used for different actions. For instance, AS286 has defined the community `286:1n` to specify how many times its AS number is prepended in advertised routes to its neighbours, *cf.* sub-section 2.2.3. Another example of the usage of this attribute is tagging a route received from a neighbour according to a given metric, *e.g.*, type of peer.

The general decision process, for choosing the best route to a given prefix, is a set of steps applied sequentially until one of the routes is preferred:

1. select the route with the highest Local Preference value;
2. select the route with the shortest AS Path;
3. select the route with the lowest MED attribute, if the routes were received from the same AS;
4. select the route with the lowest IGP metric, *i.e.*, the closest egress point (used for hot-potato routing);
5. apply other tie-breaking rules, *e.g.*, IP address of next hop.

2.2 Critical Issues

Although Internet has grown exponentially until 2001 and linearly since then [17], BGP has managed to do interdomain routing on the Internet. Notwithstanding the fact that despite Internet growth, BGP continues to support the current functioning of the Internet at the interdomain routing level, there are some critical aspects which it does not manage quite well. In the following sub-sections the main critical issues of BGP are analysed.

2.2.1 Scalability and Convergence

Scalability is often a major issue in every distributed protocol intended for a system of the size of the Internet. BGP has remained scalable mainly due to prefix aggregation introduced in its 4th version. This slowed the growth of the routing table size as well as the number of updates exchanged in BGP sessions. However, with the increasing use of multi-homing that led to an explosion of IP prefixes, it is not possibly to always apply prefix aggregation. This leads to increasing size of the routing table and of the number of updates exchanged [13]. While the former problem has been solved by incrementing the computing power and the size of memory in routers, the latter is a considerable threat to BGP's scalability. The steady evolution of the size of BGP routing tables is shown in figure 2.1.

BGP does not rapidly react to a failure. In fact, convergence studies show that it is rather slow, from tens of seconds to tens of minutes. A single link failure can make most of the BGP routers to exchange a considerable number of updates when exploring alternative paths. MRAI defines the interval by which updates from a prefix have to be separated, but may cause delaying on important BGP updates [54]. Additionally, the route flap damping mechanism is crucial to avoid large amounts of updates from a flapping router but it also increases the BGP convergence time [43].

Due to the policy-driven nature of BGP reachability updates, it is not possible to prove that routing converges in all situations [28]. Such situations occur when there is no equilibrium on the path choices of ASes, *i.e.*, there is at least one route that is not preferred by all ASes. For

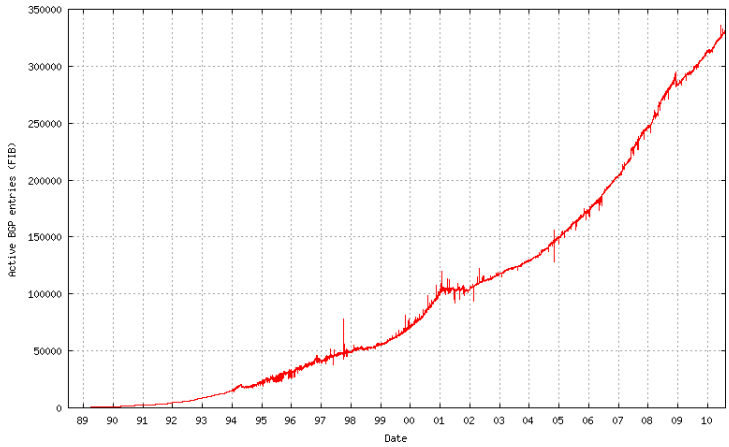


Figure 2.1 Evolution of number of entries in FIB routers [1]

instance, a group of ASes can continually switch the route to an AS X since for the announced best route of an AS Y to X, there is at least one AS Z which prefers other route to X [24].

Multi-homed ASes have been splitting its prefixes to achieve incoming load balancing which increases the size of the routing table and the number of update messages exchanged. Thus augmenting identified scalability and convergence problems.

Convergence and Scalability are two issues that cannot be considered individually due to its interdependency. It is important to consider what implications on scalability a solution to diminish convergence time has, and vice-versa.

2.2.2 Routes Quality

When a BGP router receives advertisements from its neighbours, it applies an inbound filter in order to consider only the information relevant to it. Thereat the router calculates its routing table using specific policies. However, as each AS has specific policies which can be different from policies of other ASes, the latter mechanism limits the amount of detail sent in each advertisement.

In addition, BGP does not have a cost function as all ASes are equally seen in an AS path, without distinguishing ASes with a big infrastructure from ASes with a small one. Therefore the shortest AS path may not be the one that provides best end-to-end performance, *i.e.*, having the smallest total number of hops. Moreover, some route decisions are also influenced by policy constraints. Since each AS applies its own metrics and policies, this leads to situations where the preferred route from one AS can be considered worse to another AS.

Furthermore, due to scalability issues, it is not possible to send multi optimal paths for each prefix thus reducing the quality of available routes.

2.2.3 Load Balancing

Load Balancing is divided in two categories: inbound load balancing and outbound load balancing. When an AS receives the same prefix from its providers, a possible method to balance outbound traffic between the providers is the following: dividing the prefix in sub-prefixes and assigning each sub-prefix to a provider in the routing table. Another possible approach is to use the Local Preference attribute for the routes received from more than one provider to control outbound traffic. Both approaches need to be dynamically tuned in order to improve outbound load balancing.

In what concerns inbound load balancing, a widely used technique is prepending the own AS number in advertisements in order to increase the AS-path length of specific prefixes so that it can influence the selection of the best route. Unfortunately, the behaviour of BGP when prepending the AS number is not well-defined. Additionally, aggregation mechanisms applied by transit providers can discard specific routes sent from their customers, or even eliminate the redundant numbers in the AS path. In those cases AS number prepending is useless. Nonetheless, as mentioned above, if an AS has its own prefix space, its providers are not able to easily perform prefix aggregation. In this case an AS can divide its own prefix in sub-prefixes and advertise each one to a different provider. However, some ASes may apply more restrict rules, such as filtering small prefixes, *e.g.*, $< /24$ prefixes.

Additionally, the lack of multipath advertisement support in BGP makes impossible to perform load balancing in a set of best routes, *e.g.*, in a round-robin manner.

2.2.4 Quality of Service

BGP has not built-in Quality of Service (QoS) capacities. It was designed to mainly distribute reachability information. The lack of QoS agreements among ASes contributes to the non-definition of QoS mechamisms at the inter-AS level.

2.2.5 Security

There are no mechanisms in BGP that prevent an AS from advertising arbitrary prefixes, *i.e.*, BGP does not support prefix authentication. Without manually configured filters in the neighbours of a given AS, it can advertise several popular prefixes and move a substantial amount of that prefixes traffic to it [48].

Secure-BGP (S-BGP) [33] proposes using certificates to bind prefixes to ASes, however this solution requires a Public-Key Infrastructure which introduces high overhead on messages. An alternative, which do not require a greater modification like S-BGP, is to use an external repository with registries of AS-prefix bindings.

Additionally, regardless keep-alive mechanism, a BGP session stays up as long as BGP messages can be exchanged over a TCP connection. A possible attack is to send a TCP reset segment to interrupt the TCP connection between two BGP routers, therefore leading to a BGP session to fail and causing a cascade of BGP updates, which consists in a Denial of Service

attack. A possible solution is to authenticate the BGP messages exchanged in a session, forcing BGP neighbours to share a different secret per pair of neighbours (RFC 2385). However, since the number of neighbours in transit ASes is often in the order of hundreds, the usage of RFC 2385 is optional.

In chapter 3 we will return to the discussion of some other BGP limitations since future Internet requirements, at the architectural level, are not fulfilled by the current framework where BGP *resides*.

Before that, in the following section some patches to BGP are presented. They are dubbed short-term fixes since they do not solve all problems of BGP, but may solve some for some next years. Due to the sensitivity of BGP to change, along with its fundamental role in the current Internet, defining a solution that can greatly improve BGP performance is an extremely challenging task.

2.3 Short-term fixes

When a router detects a failure it sends a withdrawal regarding the prefixes that became unreachable due to that failure. The reception of a withdrawal leads to the exploration of alternative paths. A router that does not know the best alternative route to the withdrawn prefixes will first consider a worse route. On the reception of a better route, the router will change to that route and announce it. This exploration of alternative routes, until the best route is received, is known as path exploration. The MRAI mechanism was introduced to limit the churn of BGP, specially during the path exploration phase. Although it has reduced the number of messages exchanged per unit of time, it also has increased convergence time since two messages for the same prefix are delayed regardless their type and what triggered them. It has been proved [39] that the convergence time of BGP after a failure is $n \times MRAI$, where n is the length of the longest path in the network *i.e.*, the network diameter. Adding the path exploration problem, this brings the average convergence time to many minutes.

The main goal of short-term fixes proposals is to reduce the time of the path exploration phase. The most common techniques are: preventing withdrawals, fast removing of obsolete paths and differentiating advertisements. Moreover, the latter technique results in an alternative timer mechanism to MRAI timers.

2.3.1 Preventing withdrawals

As the full-mesh iBGP architecture does not scale, other alternatives are needed, leading to a partial knowledge of the routes received in eBGP sessions. For instance, with Route Reflectors (RRs) only the best route for a RR is propagated to iBGP clients. Several ASes have multiple connections to each neighbour and other ASes are multi-homed with more than one provider. Therefore, although internal routers only receive one route to a given destination, there are alternative paths that can be used when the primary route fails. A mechanism [59] to prevent the

propagation of unnecessary withdrawals when alternative routes are known has been presented
.

When propagating in iBGP an update regarding the primary route for a destination, if multiple paths are available for that destination, it is attached to the update a community value `PATH_DIVERSITY`. When the primary path is withdrawn, instead of propagating the withdrawal message, routers start a timer and re-advertise that route with a local-preference value of 0. Allowing the withdrawn route to temporarily stay in the FIB blocks withdrawal messages, but does not prevent traffic loss. The local-preference value is 0 so that alternative routes can be preferred over the withdrawn one. When the timer expires the withdrawn message is propagated, meaning that the failure has also affected the alternative route(s).

If the alternative routes are from another neighbour, it may not be possible to advertise it to all other neighbours due to export-policy constraints. For those neighbours it is sent a withdrawal message. Although the proposed mechanism prevents unnecessary propagation of withdrawal messages, the authors have not specified the value of the timer and have not evaluated the impact on convergence time.

2.3.2 *Flushing* obsolete paths

While the above mechanism is aimed at preventing unnecessary propagation of withdrawals, the Root-cause Notification (RCN) mechanism [52] discards alternative routes in the RIB that are affected by the same failure of a received withdrawal message. Each update piggybacks its root cause to inform affected nodes, its direct neighbours and so on. Since updates triggered by the same root cause can be propagated along different multiple paths, routers must differentiate which update is *fresher* and if an update has already been received.

In order to detect invalid routes, each router maintains a table with the highest sequence number received from an AS r. In the AS path of a routing table entry, each AS is associated with its last received sequence number. After receiving an advertisement, a router updates the sequence number of the root cause AS r, not only in sequence numbers table, but also in the routing entries with routes traversing AS r. When a router receives a withdrawal regarding a route to a destination firstly sent by AS r, before computing a new alternative path, the router removes all outdated paths in the RIB, *i.e.*, all paths which contain a sequence number of AS r lower than the one received in the withdrawal. By doing this, a router can safely remove obsolete paths before computing a new alternative one and advertise it to its neighbours, thus reducing convergence time after a failure. Moreover, when a router receives an outdated update, according to its sequence numbers table, instead of propagating it, the router sends its last received update concerning the same destination. Thus helping to quickly remove obsolete updates from the network.

As the MRAI timer delays withdrawal messages, routers exchange invalid routes until the timer expires, which increases convergence time after a failure. With the Ghost Flushing proposal [2], withdrawals are sent as soon as possible in order to *flush ghost information*, *i.e.*, invalid routes, without changing the message format like in RCN.

When the route to a given destination is updated to a worse AS path and a MRAI timer have not expired since the last advertisement, a router sends a withdrawal message to all of its neighbours. If the MRAI timer has elapsed then the new path is advertised. Therefore, the withdrawal message is only sent in the situation where MRAI prevents the router of sending the new AS path in order to rapidly $flush$ the current AS path from neighbouring routers, and, consequently, other routers that are interested in that route.

Additionally, advertisement of better routes to a destination are guaranteed to be delayed. A router announces a better AS path only if it has received the announcement about this AS path at least δ seconds before, otherwise it delays the announcement δ seconds. Even new advertisements regarding recovered paths are delayed. This mechanism enforces that $flush$ withdrawals are propagated during the δ interval, thus reducing the number of advertisements containing invalid routes.

2.3.3 Advertisement differentiation

One of the causes of message churn in the path exploration phase is the reception of worse routes before the reception of the best one for a given destination. If a router firstly received the best route, it would not advertise the worse routes after receiving them. Thus, other routers would not explore routes to a destination which would not be chosen. While MRAI timers blindly delay each advertisement, MRPC timers [40] are a generic delay mechanism which enforces an ordering between advertisements according to a route metric as well as routing policies.

Instead of defining a fixed value for every update like MRAI timers, the MRPC timer value of each update relies upon the shortest path metric and the import-export policies defined in each router, while being set independently without sharing information among neighbours. The value of the MRPC timer for a given advertisement can be represented by $s \times n + f(inbound_r, outbound_r)$ - *c.f.* table 2.1, being s a common scaling factor, n the number of ASes in the path and f a function that adds a factor related to neighbour classification.

Table 2.1 Neighbour class preference function : $f(inbound_r, outbound_r)$ - inbound relationship (line) and outbound relationship (column)

f	$c2p$	$peer$	$p2c$
$c2p$	0	k	$2.k$
$peer$	$+\infty$	$+\infty$	k
$p2c$	$+\infty$	$+\infty$	0

The first factor guarantees that routes with shortest paths are received first. Whereas the second factor ensures that routes from a preferred neighbour class are received before routes from a less preferred one, regardless the AS path length. f values are coherent to the import-export policies and preference order of BGP[1]. The value of k is such that routes received from

[1] customer $\gg$ peer $\gg$ provider

customers are received before those of peers, which themselves are received before those from providers. As long as the AS path length of routes is shorter than k, it is ensured that this mechanism avoids path exploration. Finally, withdrawal messages are not delayed so that obsolete paths are purged from the network.

While MRPC timers define three preference classes and distinguishe routes in each one by the length of its AS path, in Differentiate Update Processing (DUP) [63] updates are only classified in two classes, high and low priority. The timer's value of low priority updates is equal to the default MRAI timer, whereas high priority ones are delayed with half MRAI timer's value.

The classification method is as follows. Updates regarding new destinations belong to high priority class. Advertisements to a peer or a customer have low priority. Advertisements to a provider have high priority if they contain a better route than the previously known one, otherwise they have low priority.

The above classification assumes that updates contain valid routes. However, after a failure, updates with invalid routes may be propagated before valid routes since the new best one may not be shorter than some invalid routes. After a network failure, the withdrawn prefixes enter transient state and an interim route that has the minimum similarity to the withdrawn one is used, in order to prevent the latter situation. A timer is started after selecting the interim route, during which updates are only added to/removed from the RIB to allow the propagation of valid updates and flushing of invalid ones. When the timer expires a new route is computed using BGP route selection algorithm and the prefix enters stable state. If the interim route is withdrawn during transient state, a new one is selected and the timer is restarted. In addition, in the transient state an update has high priority if the lcs(update_route, susceptible_path) is lesser than the lcs(iterim_route, susceptible_path) [2]. Otherwise it has low priority.

Although the interim route may not be the best one, as long as it is valid, it guarantees the reachability of the withdrawn destination. Additionally, by selecting the shortest path with the largest difference from the withdrawn route, it is more likely to have a valid one during transient state. Avoiding exploration of possibly invalid paths helps routers to converge faster to a valid route.

2.4 Summary

The main goal of all the presented proposals is to reduce message churn in the path exploration phase, in an attempt to reduce the convergence time of BGP. Table 2.2 summarises the characteristics of all discussed proposals and presents the convergence time of each one, as defined in the correspondent article. When a proposal does not comprise a characteristic, the $\times$ symbol is placed in the correspondent place. By contrast, the $\surd$ symbol is placed when a proposal comprises a characteristics.

Some of the presented proposals achieve a convergence time of $O(d)$, where d is the network's diameter. Additionally, h represents the average delay between two neighbouring ASes,

[2] $lcs(x,y)$: the least common sequence between x and y.

Table 2.2 Comparison between the presented short-term fixes proposals

	Prev. [59]	*RCN* [52]	*MRPC* [40]	*DUP* [63]	*Ghost* [2]
Advertisements differentiation	×	×	√	√	√
Advertisements filtering	×	√	×	×	×
Withdrawals filtering	√	×	×	×	×
Invalid advert. prevention	×	√	×	√	√
Modification of message format	×	√	×	×	×
Convergence time	N/A	$h.d$	(1)	(2)	$h.d\frac{K}{K-1}$

which includes message processing and propagation delay. Specifically to Ghost Flushing, K is the ratio between the speed at which withdrawals propagate and the speed at which announcements propagate, and it is equal to $K = \frac{\delta+h}{h}$ [2]. Even though MRPC and DUP proposals do not define analytically the convergence time achieved, like RCN and Ghost Flushing proposals, their proponents have compared the simulation results obtained for BGP and their proposed mechanism. Thereby, the average convergence time (1) and (2) is around 40% of the convergence time of BGP.

Although convergence time and message churn are reduced with the presented proposals, the other critical issues continue unsolved. Mixing a set of (independent) singular fixes and add-ons to the current specification of BGP, would probably increase its complexity and make even arduous to control and understand its behaviour.

BGP has been more or less able to cope with the current scalability requirements of inter-domain routing, though at the price of some drawbacks:

- poor route selection capabilites;
- multi-homing, load balancing and traffic engineering limitations;
- no security mechanisms;
- slow convergence time.

The aforementioned short-time fixes mainly try to deal with the last issue, being the others largely untouched.

In the next chapters we will return to some of them.

3. Routing Schemes and the Future Internet Requirements

There are two major directions that researchers have been taking in order to overcome the identified issues of BGP. On the one hand, short-term fixes may delay the decline of the current architecture or even solve its main problems for some next years. These proposals need to be backward compatible, *i.e.*, the changes they present have to be compatible with (mostly of) the BGP current specification. On the other hand, to solve scalability, security, quality of routes problems and adding mobility support to the Internet, a totally new approach is required. Therefore, a new architecture for interdomain routing is needed. However, any clean-slate approach will only succeed if the current interdomain architecture can be (progressively) migrated to a new one, which may use a different routing scheme.

3.1 Design Requirements

BGP has some inherent limitations such as the lack of support for traffic engineering, multihoming, mobility and security. Adding support for these issues in the current Internet framework is, probably, an impossible task due to the restrictions related to the current interdomain architecture. Many researchers consider that it is impossible to solve all the identified limitations with more incremental patches, as it has been done till today. Thereby, researchers consider that time has come to define a new interdomain architecture. Before discussing the design requirements of a new architecture, it is important to review the design principles of the current Internet and the way they conflict with the (future) Internet requirements, so that we can better understand today's challenges.

The current Internet architecture was design considering the following design goals [14], in order of importance:

1. to connect existing networks;
2. survivability;
3. to support multiple types of services;
4. to accommodate a variety of physical networks;
5. to allow distributed management;
6. to be cost effective;
7. to allow host attachment with a low level of effort;
8. to allow resource accountability.

To accomplish the above design goals, the following design principles have been used:

- A network of collaborating networks;
- Packet switching;
- Strict layering;

- Intelligent end-systems;
- End-to-end argument.

Before defining a new Internet architecture it is important to not only list today's requirements but also anticipate additional requirements based on the evolution of the Internet [17]. An architecture that contemplates by design (most of) all those requirements would not need, in the near future, patches or fixes to accomplish the upcoming challenges. Some of the most important challenges are listed below:

Scalability: At the time of writing, routers in the core have more than 300 000 prefixes in their routing tables and peak rates of 1 000 prefix updates per second occasionally occur. A new architecture must isolate topologic details, such as routing updates and prefixes, in order to reduce the number of globally visible updates and routing table entries.

Multi-homing and traffic engineering: Recently, the number of multi-homed ASes has been substantially increasing. A new architecture must provide means for inbound-outbound traffic engineering without increasing routing table size within the core of the Internet.

Mobility: When the first design of the Internet was made, host's mobility was not a concern. The lack of mobility arises from the design decision of merging the identifier and locator functionality into IP addresses. A new architecture has to inherently support mobility and consider mobile hosts as *first class citizens*, as more and more mobile hosts are connecting to the Internet nowadays.

Security: Internet was not planned to be used as a mainstream form of communication, as such only well-behaved users were taken into account during its early stages. Security is a major challenge of today's Internet. Identifier source address authentication as well as authentication of ASes advertisements should be supported by a new architecture.

Less burden in the core: One of the current problems is the tremendous burden in routers in the core. Support for multi-homing, traffic engineering and mobility, as well as new components and mechanisms, should be placed as far as possible from the core. However, the core should provide route diversity as well as give network status feedback.

Route Differentiation: In the current interdomain architecture only one path is announced for each destination. If the new architecture supports multiple paths for each destination, it should support path choice based on different user demands such as reliability, low latency, *etc.*.

Migration: If a new architecture design demands a D-day, *i.e.*, it is completely disruptive with the current specification, its deployment will be very difficult. Therefore, a new architecture should have a progressive migration plan which allows ASes to change a minimal number of devices while supporting legacy routers and hosts.

Economics: While (re)designing the above technical aspects, one must not forget the implications of each mechanism/component on the business model of the Internet, which should

preserve the AS-relationship-based model. Additionally, it would be convenient to increase competition among ASes while still giving them conditions to make revenue.

A new architecture should address as many of the above requirements as possible. Moreover, researchers in the Routing Research Group (RPG) in Internet Research Task Force (IRTF) have agreed that the separation of IP prefixes into the end systems' addressing space and the routing locators' space will lighten the routing management as well as enable mobility support.

Proposals for a new interdomain architecture are described in the next section.

3.2 New architecture Proposals

3.2.1 LISP

LISP [29] (Locator-ID separation protocol) is an approach that has emerged from discussions in the IETF-RPG, where end hosts continue to use IP addresses (End-point identifiers-EID) to communicate with each other. However, routing to a EID is only possible in the domain (AS) where it resides. Moreover, EID addresses are bounded to the host and not to a location. In order to reach a host, its address must be mapped to a routing locator (RLOC) address, *i.e.*, IP addresses hierarchically organised bounded to a domain (AS). By applying strict RLOC prefix aggregation the size of the routing table is greatly reduced.

There are several proposed designs to map EIDs to RLOCs. The most recent one relies on a modified version of Chord DHT [44]. While classical DHTs tend to randomise which node is responsible for a key-value pair, in LISP-Chord the mapping is always stored on the AS in control of the mapping, thus preserving the locality of the mapping. Each domain (AS) controls the mapping of its hosts relying on one or more mapping servers in order to guarantee the reachability of its hosts. In addition, the mapping system supports one-to-many EID-RLOC mapping to support multi-homing.

End-to-end routing is based on a simple IP-over-UDP tunnelling approach, though end-hosts send packets as in the current Internet. When a packet arrives at a border router of an AS, acting as the Ingress Tunnel Router (ITR), the router enquires the mapping service for the correspondent RLOC to the destination EID, referring to the Egress Tunnel Router (ETR) of EID's AS. In the arrival at the destination AS, the ETR decapsulates the packet and forwards it to the destination host. Furthermore, each RLOC has two fields: weight and priority. The priority field represents the class of the RLOC, *e.g.*, primary or back-up, whereas the weight field specifies the amount of traffic regarding the RLOC class that should be sent to it. Thereat inbound load balancing can be achieved for RLOCs with the same priority value. Moreover, in order to reduce the number of requests to the mapping service, each border router maintains a local cache of EID-RLOCs mappings.

3.2.2 HAIR

HAIR [25] (Hierarchical Architecture for Internet Routing) combines a similar hierarchical routing approach with locator-identifier separation. The HAIR architecture is divided in three layers:

- *LAN*: Access networks that connect end nodes to the Internet, *e.g.*, Ethernet LAN;
- *MAN*: A single entity which manages various LANs. In the actual infrastructure it corresponds to a small ISP that does not provide transit to other ASes;
- *WAN*: The backbone network which routes packets between MANs. It may be formed by transit providers and tier-1 ASes in the current infrastructure.

The three layers are connected through attachments points:

- *MAP-MAN Attachment Point*: routers in a MAN that are connected to one or multiple LANs;
- *WAP-WAN Attachment Point*: routers in a WAN that provide access to the backbone to MANs. Each WAP can have multiple MANs connected.

The mapping service is divided in two layers: a global DHT is used to obtain the correspondent mapping server of a MAN. Each MAN manages its own mapping service, which mappings between identifiers and locators are done in the same way as in LISP. Moreover, the number of updates to the global mapping service is considerable small since changes among MANs are infrequent. In addition, changes within a MAN only trigger updates in the local mapping server.

In order to reduce routing table size, only attachment points are used for routing, *i.e.*, each locator represents a *loose source route* composed by a sequence of attachment points. When a packet arrives at an attachment point, only information about a single MAN or the attachment points of the WAN is needed to route the packet towards the next hop. Therefore, individual MANs can run separate routing protocols. However, ASes in the WAN need to use the same routing protocol where WAPs are used as locators. Since this scheme limits the scope of updates to a single MAN or to the WAN, it greatly reduces message churn.

If multiple paths are available to a given identifier, its MAN mapping server can return more than one locator. The sender can choose to use multiple locators in a single connection or use each locator for a different connection. Alternatively, the MAN mapping server can return a different locator for each request in order to achieve inbound load balancing.

Mobile hosts move to geographically close locations when moving from a LAN to another, thus it is highly probable that mobile hosts stay in the same MAN. In this situation only the MAN mapping service is changed. In addition, close MANs can support *foreign* identifiers to allow mobility between different MANs. Since packets contain the locator and identifier of hosts, the new locator of the mobile host can be inferred as soon as the first end-to-end packet carrying the new locator is received. However, new connections need to wait for the mapping system to be updated. Furthermore, new components of HAIR are placed as closed as possible

to the edge of the network, *i.e.*, MANs. For example, the actual mapping service is stored in MANs, thus not interfering with routers in the WAN.

In the next section some alternative routing schemes for interdomain routing are presented.

3.3 Alternative Routing Schemes to BGP

The routing scheme used is somehow orthogonal to the architecture, provided that it is a form of scalable routing. Scalable routing schemes aim at reducing the routing state needed to guarantee reachability and improve convergence. Hierarchical-based routing schemes hierarchically distribute nodes, *e.g.*, on a tree, and rely upon clusters of different levels that group a set of nodes. Geographical routing is a routing scheme where nodes route messages relying on geographic information of the destination, instead of using the network addresses. In Landmark hierarchical routing nodes are organised in multi-level hierarchy of landmarks, where the landmark hierarchy determines node addresses and routing tables. An hybrid routing scheme uses more than one routing scheme, *e.g.*, HLP uses a link-state algorithm in provider-customer hierarchies and a path-vector algorithm among provider-customer hierarchies.

Some alternative routing schemes to BGP that were found in the literature are next presented.

3.3.1 NIRA

Yang *et al.* propose a new interdomain routing architecture (NIRA) [66] which gives a user the ability to choose the route of its packets. In NIRA the core is composed by tier-1 ASes, *i.e.*, which do not purchase transit from other ASes. Each tier-1 AS has a globally unique address prefix and allocates non-overlapping subdivisions of the prefix to each of its customers. A customer can recursively allocate non-overlapping subdivisions of its sub-prefix to its customer ASes. Each end-user has an address from some or all of the sub-prefixes of its providers.

A provider-rooted hierarchical address scheme is used to encode a route part that connects a user to a core provider. An end-to-end route is represented by a sender part and a receiver part, which can be obtained from a name-to-route lookup service (NRLS). The sender part is used to reach the core and the receiver part is used to reach the receiver host. Therefore, choosing different source-destination address pairs of hosts allows the selection of alternative routes through different providers. This is an essential role of the provider-rooted routing scheme. Moreover, this scheme limits source address spoofing since both the source and destination addresses are used for forwarding. Additionally, for peering links outside the core, an address prefix is allocated and divided by the two peers, of which sub-prefixes are allocated in the same way as global prefixes. Peering address prefixes are not propagated into the core.

A topology information propagation protocol (TIPP) is used to inform users of their *up-graph*, *i.e.*, the routes a user has to reach the core from its providers. Nonetheless, ASes in the core run an interdomain routing protocol to set up their forwarding tables. TIPP has two

components: a path-vector component to distribute the set of provider-level routes in a user's up-graph without selecting paths, and a policy-based link state component to propagate dynamic information concerning a user's up-graph.

Although the default usage of NIRA is to let users select routes based on their preference, route choice is not required to be performed by end users. Access routers of a domain can perform route choice and isolate users from the rest of the Internet, *i.e.*, do not let users to be aware of the multiple routes available. Nevertheless, giving users the possibility of choosing domain level routes enhances competition among providers to offer better end-to-end performance and reliability.

3.3.2 HLP

HLP [62] is a hybrid link-state path-vector protocol. Its routing structure relies upon a hierarchical organisation of the AS topology based upon provider-customer relationships. Each provider-customer hierarchy is composed by a root AS, which is not customer of any other AS, all its customers ASes and their customers, and so on, until customer ASes that do not provide transit to other ones. A multi-homed AS can be part of more than one provider-customer hierarchy, similar to the hierarchy of NIRA [66]. The roots of provider-customer hierarchies are connected through peering links, which use a fragment path vector (FPV) similar to BGP where updates only contain the AS path through different hierarchies. The part of the AS path local to a hierarchy is omitted. Communication within each hierarchy is made using a link-state protocol. When a link failure occurs within a provider-customer hierarchy, if an alternative path exists with a comparable cost *i.e.*, that it is not higher then a threshold Δ defined in each AS, an FPV advertise is not propagated to other hierarchies. Additionally, when an AS receives a FPV withdrawal, if it has an alternative path through other peer it does not advertise the withdrawal to its customers.

In what concerns traffic engineering, inbound load balancing can be improved by two ways: (i) if a root AS has more than one route for a given destination with comparable costs, it can distribute the packets for that destination through those routes, without needing to send any message to its customers; (ii) an AS can manipulate the cost attribute of its FPV paths.

On one hand, root-ASes also execute the link-state part within their hierarchy, which incurs some additional workload. On the other hand, since routing is based on AS numbers, the routing table size is reduced. Additionally, the isolation of routing updates in each hierarchy greatly reduces global message churn, which relieves routers in the core of the network.

3.3.3 Feedback Based Routing

A source-routing approach dubbed feedback-based routing has been proposed in [68], where structural information regarding the existence of links is separated from dynamic information related to the quality of routes. In order to reduce the burden in transit providers, transit

routers periodically propagate structural information about its direct links. Short-term link failure events are not propagated. They also maintain a forwarding table for its direct neighbours to forward packets based on the route carried by each packet. Additionally, each transit router has an access control list (ACL) to filter packets according to its policy rules.

Access routers, *i.e.*, routers from the edge of the network, compute a graph representation of the Internet with the structural information received from transit routers. Dynamic information is discovered by access routers based on feedback and round trip time (RTT) probes. Each link is associated with a timestamp and an expiration timer, after which the link is removed. For each destination an access router computes a primary and a backup route which should differ as much as possible. Initially, the primary route is the one with the shortest path to the destination. Periodically, an access router computes the primary and backup paths based on its current view of the structural graph and RTT values. For each destination, its RTT value is an average of the measured times between a TCP SYN packet and the corresponding SYN ACK packet of TCP connections to the destination. Occasionally, an access router sends probe messages to deliberately measure the RTT of a route. Thereat, the primary route is chosen to be the one with the shortest RTT to the destination. Furthermore, each packet is sent through the primary route and switched to the backup one when the primary route fails.

The separation of structural and dynamic information along with the placement of route computation in the edge of the network greatly reduces the burden on transit routers as well as message churn. The majority of messages in BGP are not caused by structural information changes but are a result of faults and of traffic engineering techniques [68], which do not appear in this routing architecture. Since the primary and backup routes are supposed to be as distinct as possible, if one fails, reachability is ensured by the other, thus reducing the response time after a failure. Moreover, ACLs of transit routers can be used to control Denial of Service (DoS) attacks [68]. Once a DoS attack has been detected it is often possible to identify its pattern. The victim AS can propagate routing information to its providers, which can then forward to their neighbours, in attempt to configure a filter for the DoS attack.

3.3.4 Pathlet Routing

An overlay network approach for interdomain routing has been presented in [27], where routers are represented by one or more virtual nodes (vnodes) and advertise fragment of paths, dubbed pathlets. A pathlet is a route from a vnode v_1 to other vnode v_2, uniquely identified in the former by a forward identifier (FID) and a sequence of vnodes used to reach v_2 from v_1. Transit routers propagate pathlets using a path-vector protocol as in BGP, and store a forwarding table for each vnode, composed by advertised pathlets. Therefore, the forwarding table of transit routers scales with the number of neighbours rather than the number of possible destinations, as in BGP. Edge routers compute a graph with received pathlets and execute a shortest-path algorithm to select a route to each destination. Each packet carries a source route composed by a sequence of vnodes, resulted from the concatenation of pathlets. In each hop, the ingress vnode in the source route is replaced by the sequence of vnodes identifiers in the forwarding

table of the ingress vnode.

Pathlets can be used to express local policy constraints, *i.e.*, the portions of routes which cross a network. A router can easily define different routes based on incoming traffic by using a different vnode for each neighbour and configuring its forwarding table. Additionally, it is possible to define classes of quality of service on individual segments by tagging each pathlet with a class identifier and setting up multiple pathlets over the same physical path.

3.4 Summary

The separation of the current IP address scheme into identifiers and locators, along with the hierarchical organisation of the locator space, greatly improves scalability of HAIR and LISP. The mapping system of both architectures allows one-to-many identifier-locator bindings, thus supporting multi-homed ASes. However, the mapping system in LISP comprises an explicit inbound load balancing mechanism. In HAIR it is also possible to perform inbound load balancing due to the availability of multiple addresses per destination, though there is no explicit mechanism to perform it. As most of the mapping system is placed close to the edge, both architectures lighten the burden in the core. Furthermore, LISP and HAIR rely upon tunnelling approaches to support legacy hosts when moving from the current architecture.

As regards to alternative routing schemes, scalability is a common concern of all the presented schemes, by reducing the state in each router as well as the number of exchanged messages. Although all presented routing schemes support multi-homed ASes, most of them do not comprise traffic engineering mechanisms. In fact, Feedback based routing only uses one route per destination. Nevertheless, HLP and Pathlet Routing allow ASes to use different paths per destination, whereas NIRA offers path choice at the user level.

None of the presented architectures and routing schemes for interdomain routing comprise all the previously listed design requirements, *e.g.*, traffic engineering, mobility, route differentiation and security. Only HAIR inherently supports mobile users and only LISP contemplates traffic engineering mechanisms along with route differentiation. On the one hand, a route differentiation/traffic engineering mechanism can be further added to the architecture as long as multiple routes exist for a given destination. On the other hand, security should be a design principle since delaying its deployment requires not only adding new architectural components, *e.g.*, public-key infrastructure, but also redefining the message format.

In parallel with the definition of new architectures and alternative routing schemes, several studies have been made regarding the characteristics of large-scale networks, *e.g.*, the Internet, and the suitability of a new routing strategy to those networks, greedy routing, which possesses highly interesting properties, *e.g.*, highly scalable in networks with hundred thousands and millions of nodes. Before tackling the problem of defining a greedy routing scheme for interdomain routing, in the next chapter we start by presenting the main topological characteristics of those networks, designated as scale-free networks, as well as the underlying concepts regarding greedy routing.

4. Scale-free Networks and Greedy Routing

Initially, graph theory focused on graphs of such a scale that their nodes and edges could be easily enumerated. However, since the 1950s complex networks have been studied, *i.e.*, large-scale networks with apparently no design principles, having non-trivial topological properties and exhibiting patterns not purely random. At first, these networks were described as "random graphs", a model proposed as the simplest and most straightforward one. Erdős and Rényi introduced what became a classical model to construct a random graph of N nodes: every pair of nodes is connected independently, according to a given probability p, which follows a certain probability distribution, *e.g.*, Binomial or Poisson. As these distributions depend on N, an Erdős and Rényi random graph cannot grow indefinitely [46].

As the interest in complex networks rose, research on their underlying models and organising principles has emerged. Topologies from all fields were computerised and stored in large databases: biological networks, social networks, cellular networks, collaboration networks, citation networks, the Internet backbone [23], *etc.* . The available computing power allowed researchers to perform numerical studies on real networks with millions of nodes, and supported by those studies, many topological properties shared among the aforementioned networks have been discovered, being the degree distribution one of the most evident differences from Erdős and Rényi random graphs. In fact, one interesting property showed by all these networks is related to the fact that the degree distribution follows a power-law distribution, as we will see in the following section. In order to differentiate these from other complex networks, which degree distributions do not follow a power law, they were designated as scale-free networks since most of their properties are independent of their scale.

Moreover, following the results of an experience in social networks made in 1969, the term *greedy routing* has been recently introduced in [34] to define a new routing scheme, suitable for scale-free networks, since the complexity at each node is not proportional to the size of the network but to the number of its neighbours. Before defining it, the most interesting concepts related to those networks will be presented in the following section.

4.1 Scale-free Networks

One of the topological properties which differentiate scale-free networks from random graphs is the node degree distribution. In scale-free networks the degree distribution follows a power law. A power law distribution of the generic degree d is $P(d \geq k) = k^{-\alpha}$. The main property of this distribution is scale invariance: applying a scale factor to the distribution variable leads only to a proportional scaling of the distribution, thus maintaining its properties. As a generic power law function is represented by $P(k) = k^{-\alpha}$, the scaling invariance property is defined as follows: $P(ck) = (ck)^{-\alpha} = c^{-\alpha}P(k) \propto P(k)$. For most of the aforementioned networks, the power law exponent lies in $1 \leq \alpha \leq 3$. As it can be seen in figure 4.1, a power-law distribution

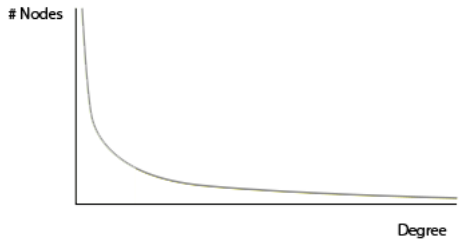


Figure 4.1 A Power-law

has a long right tail of values that are far above the mean. The lowest degrees have the highest probability values, whereas in Binomial and Poisson distributions the highest probability values are close to the mean value.

The main topological properties that characterise scale-free networks are presented in the following sub-sections.

4.1.1 Small World and Network Navigability

In 1969, an interesting experience was performed by Milgram *et al.* [64]. They asked some random individuals (sources) to send a letter to a specific person (destination), from whom they (the sources) only knew his/her name, age, occupation and city of residence. The sources had to pass the letter to people they knew, who were chosen based on the characteristics of the destination in order to maximise the probability of the letter reaching its destination. Surprisingly, 30% of the letters reached their destination and only needed a small number of intermediate people, 5.2 hops on average, even though sources had no global knowledge of the human acquaintance network topology but only their local connections.

The goal of this experience was to find short chains of acquaintances of people who did not know each other, using the *small-world method* which consists in the above method to send letters. From that experience, the fact that, despite the network size, any two nodes are connected through a relatively short path was dubbed as the small-world property. Recently, this property been precisely defined as follows: a network holds the small-world property if the shortest paths between any two pair of nodes scales, at most, logarithmically with the network size [46]. Several scale-free networks hold this property, such as social networks and the Internet backbone.

Norros and Reittu [49] have demonstrated that in graphs with N nodes whose degree distribution follows a power law with exponent $\alpha \in [1,2]$, the distance between any two nodes is in the order of $loglogN$. These networks are called *superscalable* or *ultra-small world* graphs.

4.1.2 Clustering

One characteristic that clearly distinguishes scale-free networks from Erdős and Rényi random graphs is clustering or transitivity [46], *i.e.*, if node X is connected to node Y and node Y is connected to Z, it is highly probable that node X is connected to node Z. In the context of social networks, it means that a friend of your friend is also your friend. The clustering coefficient measures the density of these relations in a network, and can be defined as:

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triples}} \tag{4.1}$$

where a connected triple is a vertex and a pair of two vertices directly connected to it, while a triangle represents the transitivity relationship. The number of triangles is multiplied by 3 to ensure that C lies in the range $0 \leq C \leq 1$, since for each triangle there are 3 triples. Watts and Strogatz [65] have given a local definition of the clustering coefficient:

$$C_i = \frac{\text{number of triangles connected to node } i}{\text{number of triples centred on vertex } i} \tag{4.2}$$

By definition $C_i = 0$ for vertices with degree 0 or 1, for which the numerator and denominator are 0. The average clustering coefficient is then defined as:

$$C = \frac{1}{n} \sum_{i=1}^{n} C_i \tag{4.3}$$

The former definition is normally used in analytical studies, whereas the latter is more suitable for numerical studies since it is easily calculated on a computer. It is important to clearly specify which definition is used since both produce different values. In general, regardless which clustering coefficient definition is used, scale-free networks tend to have considerable higher values than random graphs with a similar number of nodes and vertices [46].

4.1.3 Network Resilience

On Paper [3] a study regarding network resilience of scale-free networks and random graphs with similar number of nodes and vertices is reported. The size of the largest connected component and average path length of both type of networks have been measured in face of random node attacks as well as target attacks to nodes with the highest degrees. Being f the fraction of nodes removed from the original network, S the size of the largest connected component and l the average path length in S, the results of that study can be summarise as follows[1]:

- *Random Attacks*: in random graphs, the size of S decreases almost linearly with the increase of f, while l increases; whereas in scale-free networks the size of S also decreases with

[1]The presented results are valid until the network is formed by small connected components, from which l starts to decrease.

the increase of f, though it reaches 0 at a higher value of f. In addition, as f increases, l increases at a slower rate than in random graphs.

- *Target Attacks*: when removing the highest degree nodes a reversed behaviour happens. The size of S still decreases steadily with the increase of f, however in scale-free networks it reaches 0 at a lower value of f than in random graphs, while l rises at a higher pace than in scale-free networks.

This study [3] has shown that scale-free networks are more robust than random graphs under random node attacks. Since in those networks the majority of nodes has low degree, random removals are likely to affect low degree nodes which play a marginal role in the network functioning. However, as scale-free networks rely on highly connected nodes, the removal of such nodes has a disruptive effect on the network. By contrast, in random graphs nodes tend to have a similar and more balanced role in network functioning. Therefore, they are less robust than scale-free networks in face of random attacks but more robust under target ones to high degree nodes.

Nonetheless, it has been demonstrated [49] that scale-free networks with power-law exponent $\alpha \in [1,2]$ are very robust even if a significant part of the nodes with the highest degrees is removed. In those networks there is a spontaneous emergence of a core network which possesses a property dubbed as *soft hierarchy* [49] composed by three layers:

1. nodes with degree $\in [N^{\varepsilon(N)}\ ;\ \sqrt{N}[\ ;$
2. nodes with degree $\in [\sqrt{N}\ ;\ N^{1/\alpha}[\ ;$
3. nodes with degree $\geq N^{1/\alpha}$.

The value of $N^{\varepsilon(N)}$ is defined as being slightly larger than $1/logN$ [49]. In addition, nodes with degree $> \sqrt{N}$ form almost a clique, *i.e.*, a graph where every two vertices are connected by an edge, and the proportional size of the largest connected component[2] in the core is close to 1. As shortest paths normally pass through the core, it would be expected that the length of shortest paths would greatly increase if the nodes with highest degree were removed. However, due to the density of links in the core, it does not affect connectivity since the graph maintains a giant connected component, *i.e.*, a sub-graph which contains (almost) all nodes in the graph, and the network diameter continues to be in the order of $loglogN$ [49], thus maintaining its *superscalability* property. In fact, even if the whole core is removed, the size of the giant component is asymptotically 1 and the network diameter goes up to the order of $1/\varepsilon(N)$, which is slightly smaller than $logN$ [49].

4.1.4 Assortativity

Assortativity measures selective linking between nodes, *i.e.*, the preference which nodes have to be connected to others of the same type or of other types. The assortativity coefficient measures

[2]The ratio between its size and the number of nodes in a graph.

the fraction of edges that connect nodes of the same type, being 1 if all edges of a network connect nodes of the same type and 0 if the network nodes are totally random mixed. Social-networks have high assortativity coefficient values, since people tend to be related to persons which are similar in some way, and few people are related to others that are different, whereas other scale-free networks, such as the Internet and Biological networks, are *disassortative* [46].

For instance, the Internet can be simply divided in three groups: high-degree nodes (T1 backbone operators), transit nodes (ISPs) and end nodes (stubs). Although it is very unlikely that stubs are connected to T1s, there are several links between backbone operators and ISPs as well as several links between ISPs and stubs, which may overcome the number of connections within each group. Therefore, the Internet has a low assortativity coefficient.

A special case of assortativity is degree correlations, in which the similarity among nodes is specified by their degree. In [51] it was introduced a representation of the assortativity coefficient regarding node degree: the mean degree of its neighbours as a function of the node degree k. If this value increases with k, then the network is assortative. As regards to the Internet, this value decreases with k [51] which confirms that the Internet is a disassortative network.

Although several scale-free networks are disassortative, in such networks there is a spontaneous emergence of a core which is very dense [49] and that is sometimes dubbed as the *rich-club community*.

4.1.5 Network Growth and Network Construction Algorithm

Another characteristic that clearly distinguishes scale-free networks from random graphs is the construction method used. In Erdős and Rényi random graphs nodes are connected randomly with some independent probability, whereas in scale-free networks the connectivity of new nodes is based upon the degree of already existing nodes. In addition, scale-free networks can grow indefinitely using the construction method as the growth method, whereas Erdős and Rényi random graphs cannot, as the probability distributions used on construction depend on the value of N.

The Price's Model [55] was probably the first one proposed to explain the construction mechanism of a real scale-free network: the network of citations among scientific papers. In 1955, Derek Price found that both in and out degree, *i.e.*, the number of papers that cited a given paper and the number of papers that the it cites, follow power-law distributions. Following the work of Herbert Simon [61], who showed that power-laws arise when "the rich get richer", Price defined the *cumulative advantage* mechanism: the rate at which a paper gets new citations is proportional to the number of citations it already has. With this mechanism, a new paper would never receive citations since it has never been cited. Price defined a more general relation, in which the probability of a paper to get a new citation is $k+1$, being k its in-degree. The assumption is that when a paper is published it has a self-citation.

A slightly different model was presented later by Barábasi and Albert [5]. It aimed at modelling the growth of any scale-free network not just the citations network, though it relied upon undirected graphs instead of directed graphs as the Price's Model. Although this model does not

directly represent some real networks as the citations network, since they are directed graphs, it does not have the problem of Price's Model of how a paper gets its first citation, *i.e.*, how a new node will have new ones attaching to it. As a result, they proposed a modified construction mechanism: the rate at which a given node gets new nodes connecting to it is proportional to its degree. They have also proposed a different name for this mechanism to what is nowadays widely known as *preferential attachment* [5, 46].

Both networking constructing models produce scale-free networks, with power-law exponent ranging $2 \leq \alpha \leq 3$ and $\alpha = 3$, respectively. The latter model has been widely studied in the last years and some generalisations have been proposed [12, 21, 36], which overcome some of its limitations to fully represent scale-free networks, *e.g.*, relying upon undirected graphs.

The Milgram's experience showed that social networks are quite navigable in few hops, demonstrating that is possible to route a message between two nodes using only local information, *i.e.*, information regarding direct neighbours. Since other scale-free networks share several characteristics with social networks, especially the small-world property, one may wonder that they are as navigable as social networks. In the next section we present a routing strategy that takes advantage of the navigability properties of those networks.

4.2 Greedy Routing

The term *greedy routing* was firstly introduced by Jon Kleinberg in [34] to characterise the type of routing used in the experience of Milgram *et al.* [64]: (i) each node has only information regarding its neighbours and the destination; (ii) in each hop, the message is routed to the *nearest* neighbour towards the destination node. The notion of the *nearest* neighbour is given by a distance function among network nodes based upon the information associated with each node.

In addition, Jon Kleinberg defined a model in which each node is represented in a coordinate space and it only knows the coordinates of its neighbours. In order to send a message from a source to a destination node, each node sends the message to the *nearest* neighbour towards the destination. Geographically-inspired routing is an example of this type of routing strategy.

If successful, a greedy routing algorithm has the following highly interesting properties:

- **Small routing state**: each node needs only to maintain information regarding its neighbours. Therefore, the amount of routing state is in the order of the number of neighbours, which is quite lower comparing to routing methods based on information regarding (almost) all network. Additionally, there is no routing state maintenance cost in the sense that there is no need to exchange messages in order to perform routing.
- **Small routing stretch**[3]: paths chosen by the the greedy routing algorithm tend to be the shortest path from the source to the destination node [37].

[3]Stretch is the ratio between the length of paths chosen by a routing algorithm and optimal ones, *e.g.*, shortest paths.

- **Robustness**: changes to node population incur minimal disruptions. Even if a considerable number of simultaneous failures happen, *e.g.*, facing of random removal of 10% of the total nodes, a greedy routing algorithm ensures near full reachability in scale-free networks [37], while chosen paths continue to be close to the shortest ones.

In order to build a successful greedy routing algorithm, several interrelated problems must be solved:

- devise a method to map the network topology into a coordinate space, *i.e.*, an embedding of the network;
- construct a distance function acting on nodes coordinates;
- elaboration of a concrete routing algorithm dealing with route optimality criterion and the dead-end problem, *i.e.*, when a message reaches a node where it cannot make any further progress but to get back through an already known path to find an alternative one.

Earliest versions of greedy routing applications relied upon real geographic position information [7, 32], *e.g.*, as determined by a GPS device, and wireless ad-hoc routing scenarios seemed to be the ideal context to study if the approach would be viable.

Wireless networks with mobile nodes are characterised by having a network model which is completely dynamic. Therefore, it has been a privileged network environment where greedy routing could show its advantages over traditional methods. There are proposals for wireless networks which consider that the network embedding is solved by using GPS sensors along with geographic distance. However, there are several problems concerning with wireless communication which complicate the application of greedy routing in those networks:

- decreased signal strength;
- unknown obstacles;
- weather conditions;
- hidden terminal problem;
- interference with other sources, since used frequencies are frequently public ones;
- multipath propagation.

Nonetheless, even if the embedding problem has been solved, another problem arises: how a node knows the coordinates of a destination node. Several proposals have been presented in the literature, though they are limited by the aforementioned problems.

Moreover, an embedding for the Internet consists in mapping nodes, which can be hosts, routers or even ASes, into a coordinate space. Coordinates can either be geographic or synthetic. In the next paragraphs we present two synthetic coordinate systems which rely upon latency measurements. While the first system (GNP [47]) is based on a location infra-structured with special nodes, *i.e.*, *landmarks*, the second system (Vivaldi [16]) computes coordinates using a decentralised algorithm.

Global Network Positioning (GNP) [47] is a system that uses landmark nodes to compute coordinates for nodes in the Internet. Using a 3-dimentional Euclidean Space, a set of N landmark nodes, in different locations, initially compute their coordinates based on the distance among them, by minimising the error between predicted and real distances. Thereupon, each host derives its own coordinates from the coordinates of the N landmark nodes along with the round trip time between it and each landmark node. The coordinates are chosen in a way that they minimise the overall error between measured and computed host-to-landmark distances.

Vivaldi Coordinate System [16] is based upon an iterative algorithm which leverages the communications between nodes to define their coordinates, so that they reflect their distance in terms of latency. When a new node enters the system, it uses either arbitrary coordinates or the reference origin. As nodes exchange messages, they also compute the latency between them and exchange their coordinates. Using the received coordinates along with the latency values, nodes adjust their coordinates in order to minimise the error between predicted and measured latency values. This continuously coordinates adjustment can be modelled as a mass-spring structure in which masses (nodes) change positions in the space until all springs (latency values) are stabilised.

The above approaches propose specific embeddings for wireless networks and the Internet to use forms of greedy routing relying upon geographic distance and latency, respectively. In the next section we present a generic mechanism to apply greedy routing in scale-free networks.

4.2.1 Greedy routing in scale-free networks

As it has already been introduced, a greedy routing algorithm is based on a method to assign coordinates to nodes along with a distance function using those coordinates. Boguñá *et al.* [6] have defined a general model, based upon the concept of node similarity, as the underlying mechanism to explain the navigability properties of scale-free networks, suitable to apply the greedy routing strategy. This model does not comprise the engineering problems of technological networks as it focuses on the topological properties of scale-free networks. It aims at presenting a model which explains the Milgram's experience and can be applied for other scale-free networks. Nodes characteristics define how similar they are, which is abstracted as a *hidden distance*. Altogether, hidden distances define a *hidden metric space* for a given network which not only guides the routing on the network but it also influences its structure [6]. The hidden distance is coupled with the network structure in the following form:

a) the smaller the hidden distance between two nodes, the higher the probability that they are topologically connected;
b) if a given node A is close to node B and node B is close to another node C, then nodes A and C are also close as a consequence of the triangle inequality [4] in the metric space;
c) it is highly probable that the triangular relationship ABC exists in the network topology, which explains the strong clustering of scale-free networks.

[4] $d(A,C) \leq d(A,B) + d(B,C)$, being d the hidden distance.

A naive greedy routing algorithm can exhibit the following main problems: choosing paths with large number of hops and paths leading to dead ends, where a traveller would be obliged to go back and enter in loop. The former is a part of the general problem of optimality of a routing algorithm, while the latter is a problem of correctness since dead-ends lead to message routing failure. Therefore, the hidden metric space plays a central role in the success of greedy network *navigation* and has a major impact in the embedding and distance function definition.

In the next paragraphs we introduce an intuitive example of how both problems can be avoided in a concrete scenario.

The navigability of scale-free networks can be illustrated using an example of passenger air travel [6]. A travel from Toksook Bay, Alaska, to Ibiza, Spain is simulated using the greedy routing strategy guided by an explicit metric space, using a combination of geography and airport size. At each airport it is chosen the next-hop airport which is geographically closest to the destination. Furthermore, the navigation process has two symmetric phases. The first phase is a coarse-grained search, in which a *zoom-out* mechanism is applied: the travel begins at the small local airport Bethel and a flight to Anchorage, a small hub at a small distance, and from there to Detroit, a larger hub at a larger distance. Being Detroit a large hub airport, it is connected to the majority of other large hub airports. The turning point between the two phases is when the navigation process reaches the closest large hub to the destination. Thereby, the next hop airport is Paris. From here begins the second phase, a fine-grained search towards the destination. A *zoom-in* mechanism is applied, resulting in the travelling to Valencia, a local hub at small distance from the destination, and finally to Ibiza airport.

The navigation process, *i.e.*, the zoom out/zoom in mechanisms, works efficiently if the airport network topology and the underlying metric space exhibits the following two properties [6]:

- the network has enough hub airports to provide an increasing degree sequence during the zoom-out phase;
- the next greedy hop from a remote low-degree node is a node with a higher degree so that greedy paths normally move first to the highly connected network core.

These conditions are fundamental to ensure that local loops do not occur. Specifically to the air travelling example, an airport network without enough clustering would result in a path with several hops among small nearby airports, reaching Ibiza after many hops. In the worst case, when travelling through those small airports, it is possible to reach one that does not have any other connections closer to Ibiza, facing a dead end. This suggests that scale-free networks are suitable for greedy routing since they comprise a large number of hubs, *i.e.*, high-degree nodes, as well as strong clustering.

Following the work of Boguná *et al.* [6], an approach that solves the opposite problem was presented in [37]. There has been defined a metric space in the Hyperbolic plane that naturally leads to the emergence of scale-free networks. The application of the greedy forwarding strategy in the metric space results in 100 % reachability with low stretch, *i.e.*, near optimal paths lengths

[6]. In order to show what network topologies emerge from hyperbolic metric spaces, a network model with the following strategy was defined:

- the Hyperbolic space;
- node density, *i.e.*, the distribution of nodes in the space;
- connection probability, *i.e.*, a pair of nodes at hyperbolic distance x is connected with probability $P(x)$.

Using networks with 10000 nodes and average degree $\bar{k} = 6.5$, the performance of greedy forwarding was evaluated using hyperbolic coordinates [6]. For each generated network, its Giant Connected Component (GCC) was extracted, *i.e.*, the biggest sub-graph where all nodes can be reached from any node, and the greedy forwarding strategy was tested with 10000 random source-destination pairs. Besides the original greedy forwarding strategy (OGF), where each message is dropped when there is no neighbour closer to the destination than the current hop, a modified version (MGF) was also tested, which corresponds to the Gravity Pressure Greedy Routing Algorithm of Cvetkovski and Crovella [15].Two scenarios were studied, static networks and the dynamic networks with link failures:

Static Networks: the success ratio p_s increases while the stretch decreases as the value of γ decreases to 2. For instance, with $\gamma = 2.1$, the greedy forwarding strategy ensures $p_s = 0.99920$ for OGF and $p_s = 0.99986$ for MGF and both greedy forwarding strategies ensure maximum stretch = 1 [6].

Dynamic Networks - Link Failures: p_s and stretch values were measured in two scenarios. In scenario 1, a given percentage p_r, ranging from 0 % to 30 %, of all links in the network was removed, GCC was re-computed as well as the new success ration p_s^{new}. In scenario 2, one link was removed, GCC is also re-computed, and the percentage of successful paths p_s^l, only among those that were previously successful paths and traversed the removed link as well as belonging to the GCC. This process was repeated for 1000 random links. For instance, MGF presented $p_s^{new} > 0.99$ on networks with $\gamma = 2.1$ and $p_r \leq 0.1$ [6]. The percentage p_s^l of MGF paths which used the removed link and found an alternative path is also high, close to 100 % for small values of γ [6].

Although the above remarkable results show that the greedy forwarding strategy can be used to efficiently route messages in scale-free networks, an interesting and very challenging problem is the original abstract one: given any scale-free network, *e.g.*, the Internet AS graph, there is an embedding in a hyperbolic or euclidean space, coordinate computation with no global knowledge of the graph connectivity, that can be used to guide a greedy routing algorithm with similar efficiency?

4.2.2 Greedy Routing Proposals

Some proposals regarding graph or network embeddings into virtual coordinates have been presented in the literature. However, to our best knowledge, an embedding for scale-free networks

which supports the usage of greedy routing on such networks has not been proposed yet. Most of the proposals are concerned with arbitrary graphs and wireless networks. As some of the proposed embeddings do not rely upon Euclidean geometry, but on Hyperbolic geometry, we will present some of its main concepts before discussing each proposal.

4.2.2.1 Hyperbolic Geometry

Hyperbolic geometry is one type of *non-Euclidean geometry*, which verifies all Euclidean postulates except the parallel one. The parallel postulate in Euclidean geometry can be defined as follows: given an Euclidean line L and a point a in $\mathbb{R}$ not in L, there is only one Euclidean line K which passes through a that is parallel to L. By contrast, in the *upper half-plane model* $\mathbb{H} = \{z \in \mathbb{C} \,|\, \mathrm{Im}(z) > 0\}$, given a line l in $\mathbb{H}$, p a point in $\mathbb{H}$ not in l, there are infinitely distinct hyperbolic lines through p that are parallel to l [4]. There are other planar models of the Hyperbolic Plane, each one having its own distance function: the Poincaré Disk Model $\mathbb{D} = \{z \in \mathbb{C} \,|\, |z| < 1\}$, as well as n-dimentional models such as the Klein Disk Model (unit ball) and the Hyperboloid Model. For example, the distance function between two nodes z_1 and z_2 in the Poincaré Disk Model is as follows:

$$\cosh \rho(z_1, z_2) = \frac{2\,|z_1 - z_2|^2}{(1 - |z_1|^2)(1 - |z_2|^2)} + 1\ . \tag{4.4}$$

One common characteristic among distance functions of the different models is the distance expansion towards the boundaries of the model:

- given a pair of points (a,b) at Euclidean distance λ away from the boundary;
- given a pair of points (c,d) at the Euclidean distance λ but close to the boundary;
- unlike the Hyperbolic distance of (c,d), the Hyperbolic distance of (a,b) tend to infinity, though points (a,b) and (c,d) are equally Euclidean distante.

In Hyperbolic geometry, a Möbius transformation is a rational function that has the following form:

$$z \to \frac{az + b}{cz + d} \tag{4.5}$$

satisfying $ad - bc \neq 0$. It is used to represent isometries, *i.e.*, transformations which preserve orientation or distance, *etc.*.

Scale-free networks are composed by heterogeneous, distinguishable nodes, which can be classified into a taxonomy, *i.e.*, nodes can be divided into large groups, consisting of smaller groups, which in turn are composed by smaller sub-groups. The relationship among those groups and sub-groups can be approximated by tree-like structures [6]. Hyperbolic spaces can be considered as "continuous versions" of trees, since the hyperbolic plane is metrically equivalent to an e-ary tree, *i.e.*, a tree having an average branching factor e [6]. Scale-free networks can be embedded in Hyperbolic spaces since trees allow for isometric embeddings, *i.e.*, which preserves distances, into Hyperbolic spaces [38]. In addition, the geometric properties of trees

are closely related to the ones of Hyperbolic spaces [38]. However, the Sequoia framework [56], which will be discussed further on this section, directly produces tree embeddings without using a mapping into a Hyperbolic coordinate space. One clearly difference between Hyperbolic geometry and Euclidean geometry, besides the parallel postulate, is the distance function: while in Euclidean geometry the distance expands linearly towards infinity, in Hyperbolic geometry the distances expands exponentially towards the boundaries of a Hyperbolic space.

For more details regarding Hyperbolic geometry we refer the reader to the book [4].

4.2.2.2 Application to Generic Graphs

Cvetkovski and Crovella [15] defined an embedding for dynamic graphs, assuming that the graph has a single connected component. Firstly, nodes elect a root node r and a spanning tree T is computed. For instance, a minimal depth-tree where each node selects a parent node as the one which has the smallest distance to the root node.

In the Poincaré Disk Model, given two points at infinity[5] $a = e^{i\alpha}$ and $b = e^{i\beta}$ [6], the centre of the Euclidean circle in $\bar{\mathbb{C}} = \{\mathbb{C} \cup \infty\}$, which contains the hyperbolic line whose points at infinity are a and b, and the corresponding radius R, are given by the following formula:

$$c = 1/m^*, R^2 = 1/|m|^2 - 1 \tag{4.6}$$

where $m = (a+b)/2$ is the midpoint of the Euclidean chord joining a and b, and m^* is the complex conjugate of m [4,15]. Before determining the coordinates of all nodes, the coordinates $C(r)$ of the root node r are manually assigned in the hyperbolic plane with angles $\alpha_r = \pi$ and $\beta_r = 2\pi$, corresponding to the ideal points $a_r = e^{i\alpha r}$ and $b_r = e^{i\beta r}$. For each node n in the graph:

1. its parent p_n sends $C(p_n)$, $\alpha_n = \alpha_{p_n}$ and $\beta_n = (\alpha_{p_n} + \beta_{p_n})/2$ and updates $\alpha_{p_n} = \beta_n$;
2. node n calculates c and R with $a_n = e^{i\alpha_n}$ and $b_n = e^{i\beta_n}$ and its coordinate

$$C(n) = \frac{R^2}{(C(p_n))^* - c^*} + c \tag{4.7}$$

and updates $\alpha_n = (\alpha_n + \beta_n)/2$ [15].

The resulting embedding can be seen in picture 4.2.

Cvetkovski and Crovella [15] also proposed a modified version of the greedy routing algorithm : Gravity-Pressure Greedy Routing Algorithm. It has two modes, gravity and pressure, the normal greedy routing mode and the modified version, respectively. When a message reaches a dead-end, *i.e.*, when none of the neighbours is closer to the destination than the current node,

[5] Informally, consider all parallel lines to a given line l. The points at infinity are the ones on which those parallel lines meet at infinity, in both directions.

[6] Remember that a complex number can be represented by polar coordinates:
$z = re^{i\theta} \rightarrow z = r(\cos\theta + i\sin\theta)$.

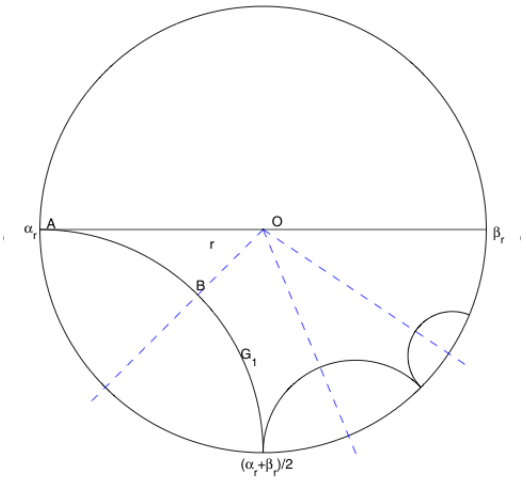


Figure 4.2 Positioning of the root node for a greedy embedding [15]

instead of discarding the message, it is switched to pressure mode. In this mode the message is sent to the closest neighbour towards the destination node, without considering if it is lower than the distance between the current node and the destination. When the message reaches a node which is closer to the destination than the one where the message changed to pressure mode, the message returns to the gravity mode, *i.e.*, the normal functioning of greedy routing.

The embedding in the Hyperbolic Space proposed by Robert Kleinberg [35] also relies upon the construction of a spanning tree T. After electing a root node r, each node $w \neq r$ chooses one neighbour $p(w)$ as its parent, such that edges $(p(w), w)$ form an arborescence rooted at r [35]. Following that, each $p(w)$ transmits to its child w the coefficients of a Möbius transformation μ_w, which then uses it to compute its own coordinates $f(w)$. Following that, each node transmits to its children their coefficients of the Möbius transformation, so that they can compute their coordinates. In order to establish the correctness of the algorithm, for each edge $(p(w), w)$ in T, the function f must map $p(w)$ and w to a pair of adjacent nodes in the greedy embedding of the infinite d-regular tree T [35]. To accomplish that, first the maximum degree of T has to be computed, to determine the degree d of the regular tree.

4.3 Summary

Several real-world networks having thousands and millions of nodes exhibit distinguishable topological properties, namely:

- node degree distribution follows a power-law distribution with exponent lying in $1 \leq \alpha \leq 3$;
- small world property, *i.e.*, shortest paths scale, at most, logarithmically with the network

size;
- strong clustering;
- highly resilient to random attacks, though less resilient in face of target attacks to the highest degree nodes;
- disassortative, in general;
- network growth model based on the principle "the rich get richer", *i.e.*, the probability of a node getting new neighbours is proportional to its degree.

Inspired by an experience in social networks, a new routing strategy was defined: greedy routing. Although the main aspects regarding this strategy are simple, the definition of a greedy routing scheme comprises several challenges: *a*) the construction of a mapping of the network topology into a coordinate space; *b*) the definition of a distance function to be used in the coordinate space. These two components of a greedy routing scheme determine its success, *i.e.*, its success ratio and stretch as well as the comprehensiveness of specific domain's routing requirements.

The aforementioned proposals can be divided in three groups:

1. domain-specific [16, 47];
2. generic graphs [15, 35];
3. scale-free networks [6, 37].

They present some limitations or restrictions that difficult their application. The ones for wireless networks normally require several rounds of broadcast messages. In addition, proposals for arbitrary graphs [15, 35] rely upon a spanning tree, leading to a non-utilisation of several links. One of the proposals also demand the the mapping of the spanning tree into a d-regular tree before embedding it in the Hyperbolic space [35]. Moreover, the proposals concerning the Internet [47,56] rely upon volatile distance measures, *i.e.*, latency, which leads to the continuing computation of coordinates. Finally, the Hyperbolic Hidden Metric Space proposal [6] solves the *inverse* problem, however it does not define an embedding for scale-free networks, *e.g.*, the Internet AS graph.

In the next chapter we will present a refresh study of the topological characteristics of the Internet AS graph to support the definition of a greedy routing scheme to the Internet AS graph, which will be presented in the following chapter.

5 . Internet AS Graph as a Scale-free Network

Following the increasing interest on the study of topological properties regarding complex real networks, in 1999 Faloutsos *et al.* [23] have shown that the degree distribution of the Internet AS Graph follows a power-law distribution, with the exponent α ranging in $1 \leq \alpha \leq 2$. Most of the empirical studies regarding Internet topological properties, that followed this seminal work, use data collected by measure and monitoring infrastructures set up by several projects, *e.g.*, Skitter [11], Archipelago [8] and Route Views [50], which continually collect data related to the Internet graph and play a pivotal role in the above mentioned studies. Before presenting the results of those empirical studies, which show that the Internet AS graph is indeed a scale-free network, in the following section some methods to map the structure of the Internet will be briefly discussed.

5.1 Internet Mapping

The mapping of the Internet structure is made at two levels: macroscopic, *i.e.*, mapping of the AS relationships, and microscopic, *i.e.*, mapping at router level. Skitter and Archipelago, which is the evolution of Skitter, focus on the latter level while RouteViews focuses on the former.

Due to privacy and competition issues regarding the internal structure of an AS, traceroute methods based on UDP segments, ICMP messages or TCP are used to construct paths at the router level. As regards to the probing method, the functioning of Skitter and Archipelago is similar: both use ICMP *echo request* messages to construct a router path between two IP addresses. The probing method consists in the following steps:

- an ICMP message is sent with TTL value of 1;
- after the reception of a *destination unreachable* message from the first router in the path, another ICMP message is sent with the TTL incremented by 1;
- until the reception of a *destination unreachable* message from the destination IP address, *echo request* messages continue to be sent. The path is constructed by the IP addresses of the routers which have previously sent *destination unreachable* messages.

While Skitter uses a centralised repository to store the probing results of independent monitor nodes, Archipelago uses a repository inspired from the Linda tuple space [26] to store the probing results of coordinated monitor nodes.

A different method is used in the mapping of the Internet structure at the AS level. The Route Views project continually collects data from vantage points BGP updates exchanged between ASes [50]. As a result, a representation of the real AS graph is constructed and made available on a weekly basis by CAIDA [9]. After extracting all AS links from a RouteViews snapshot, each link is annotated with its commercial relationship, namely: customer-provider, provider-customer, peer-2-peer and sibling. These relationships are inferred using the algorithm presented in [19].

Node degree average and distributions have been used as starting points to model the Internet [23]. Recently, the characterisation of the interconnectivity of neighbourhoods of increasing size has been shown as being able to reproduce arbitrary interconnection metrics, the so called dK-Series model [42]. This model relies on probability distributions evolving relations of different degrees d, between nodes of a given original graph $\mathscr{G}$. As the value of d increases, more properties of graph $\mathscr{G}$ are captured relying on more complex distributions. On the limit, the most complex representation produces isomorphic graphs of the original graph $\mathscr{G}$, *i.e.*, there is a bijective function $f : V(I) \rightarrow V(\mathscr{G})$, being I the generated graph and $V(\mathscr{G})$ the vertices of a graph $\mathscr{G}$. The *0K relation* characterises a given graph by the average degree $\bar{k}$ of its nodes, with $\bar{k} = 2m/n$, being m the number of edges and n the number of nodes. With this characterisation it is not possible to deduce the number of nodes $n(k)$ with degree k. The *1K relation* includes that information, represented by the node degrees distribution $P(k) = n(k)/n$, *i.e.* the probability of having a node with k degree. Nonetheless allowing a richer characterisation than the previous one, it does not contain information about node's connectivity, such as the number of links between k and k' degrees - $m(k,k')$. The *2K relation* comprises such information represented by the *Joint Degree Distribution* (JDD), which is defined by $P(k_1,k_2) = m(k_1,k_2)\mu(k_1,k_2)/(2m)$ where the value of $\mu(k_1,k_2)$ is 2 if $k_1 = k_2$, otherwise 1. The value of $P(k_1,k_2)$ represents the probability of having links between nodes with k_1 degree and nodes with k_2 degree. And so on. For most practical Internet modelling purposes, using up to *2K relations* seems sufficient.

5.2 Topological Properties of the Internet AS Graph

Empirical studies regarding the Internet AS graph rely upon representations of the real ASes structure, a graph made available by CAIDA. Using the release [10] we have computed the distribution of node degrees.

As can be seen in figure 5.1, the distribution of nodes degree in the CAIDA AS graph continues to follow a power law with a very long right tail with degrees far above the mean degree $\bar{k} = 4.4766$. In fact, near 90% of nodes in that graph have degree $< \bar{k}$. Using an estimation method based on the maximum likehood [30] we measured the power-law exponent of the CAIDA AS graph as $\alpha = 1.58$. In the following sub-sections we will present the most important topological properties found in the Internet AS graph.

5.2.1 Assortativity

Inspired by the work of Norros *et al.* [49] we defined a taxonomy [58] relying upon the scale-free topological properties of the Internet, that consists in the following four classes:

1. *periphery* - nodes with degree < 3, which mainly correspond to stub-ASes that do not provide Internet connectivity to others;

2. *intermediate* - nodes with degree ≥ 3 and $< N^{\varepsilon(N)}$ that represent transit ISPs, being N the

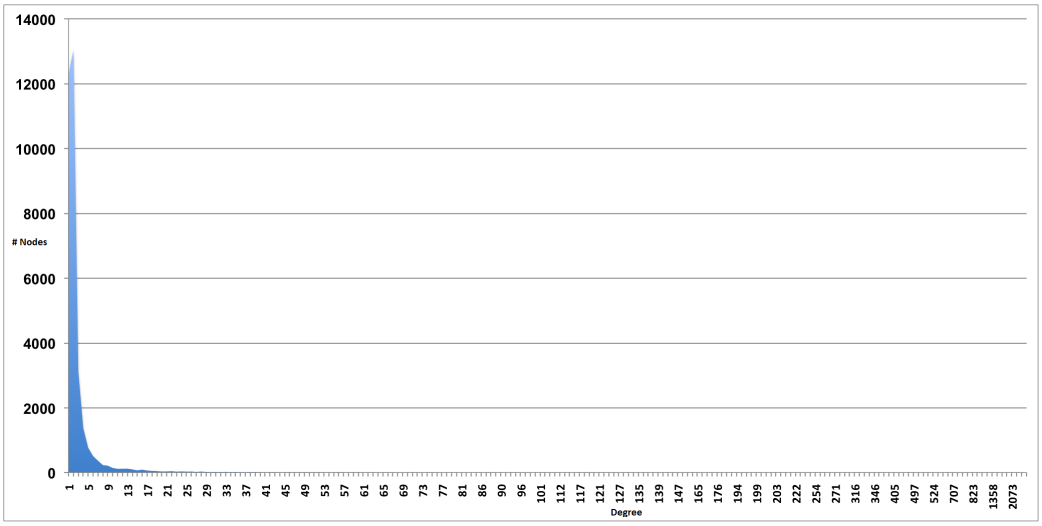


Figure 5.1 Degree distribution of the CAIDA Internet AS graph

number of ASes in the graph;

3. *core* - nodes with degree $\geq N^{\varepsilon(N)}$ and $< \sqrt{N}$, which is the first layer of the soft hierarchy defined in [49];

4. *kernel* - nodes with degree $\geq \sqrt{N}$, which represents the inner core of the soft hierarchy.

In [49] the exact value of $\varepsilon(N)$ has not been defined, though it is specified has being slightly above $1/logN$. The exact value of $\varepsilon(N)$ as well as the underlying concepts regarding this taxonomy are presented further in section 5.3. Table 5.1 presents the statistics per class in the CAIDA AS graph.

Table 5.1 Node Statistics per Class

Class	(%)	# Nodes	Average Degree
Periphery	84.7	28391	1.68
Intermediate	12.3	4108	6.48
Core	2.8	940	38.52
Kernel	0.2	69	572.57
Total	100	33508	4.48

Using this AS taxonomy we measured how assortative the CAIDA AS graph is. Table 5.2 shows how ASes from a given class relate with others. For each class it is shown the proportion of neighbours that belong to each of all classes. In all of these the percentage of neighbours which are of the same class is lower than the percentage of each of the other classes. Periphery nodes do not tend to establish relationships with each other since most ASes classified

as periphery do not provide Internet connectivity. Therefore, the majority of relationships are with ASes that provide Internet connectivity: transit ISPs and ISPs from the core and kernel. By contrast, ASes from the kernel have the majority of their relationships with *lower* hierarchies, to which they provide Internet connectivity. As intermediate nodes mainly represent transit ISPs (regional ISPs), they are likely to establish commercial relationships with periphery ASes to which they provide Internet connectivity as well as to ASes from the core and kernel, from which they purchase worldwide connectivity. These results are in line with the result presented in [46] that the Internet is disassortative and confirms its loose hierarchical structure.

Table 5.2 Distributions of Neighbours per Class

Class	Periphery (%)	Intermediate (%)	Core (%)	Kernel (%)
Periphery	6	16	39	39
Intermediate	28	11	21	40
Core	51	15	11	23
Kernel	47	27	21	5

In fact, in table 5.2 it may seem that the there are few relationships between ASes in the core and the kernel, which contradicts the result presented in [49], that the core is very dense. However, those two classes correspond to two core tiers. Table 5.3 shows the distribution of neighbours per class for the whole core, *i.e.*, merging the two tiers. Although periphery ASes are the majority of core ASes neighbours, the percentage of neighbours from the whole core is higher than the ones for each of the core tiers found in table 5.2.

Table 5.3 Distributions of Neighbours in the Core (two tiers)

	Periphery (%)	Intermediate (%)	Core (two tiers) (%)
Core (two tiers)	49	21	30

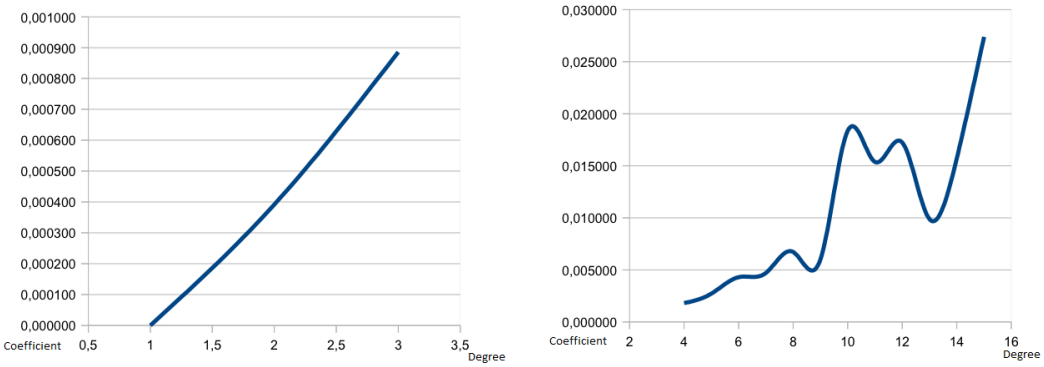
5.2.2 Clustering

We have also computed the clustering coefficient of each node using the following definition:

$$C_i = \frac{\text{number of triangles connected to node } i}{\text{number of triples centred on vertex } i} \tag{5.1}$$

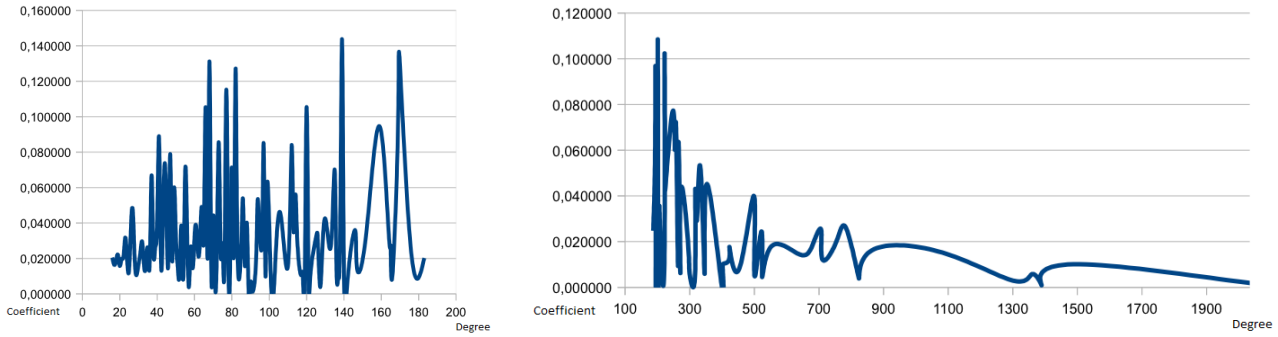
Using the formula $C = \frac{1}{n}\sum C_i$ we have calculated the average clustering coefficient $C \simeq 0.011$. In figure 5.4 it is shown the clustering coefficient for the CAIDA AS graph aggregated by degree, whereas in figures 5.2(a), 5.2(b), 5.3(a), 5.3(b), it is shown the clustering coefficient of periphery nodes, intermediate nodes, core nodes and kernel nodes, respectively.

The average clustering coefficient for each of the four classes is presented in table 5.4. Although the average coefficient C (0.011) of the CAIDA AS graph is much lower than the one



(a) Periphery Nodes (b) Intermediate Nodes

Figure 5.2 Clustering Coefficient of the CAIDA Internet AS graph



(a) Core Nodes (b) Kernel Nodes

Figure 5.3 Clustering Coefficient of the CAIDA Internet AS graph

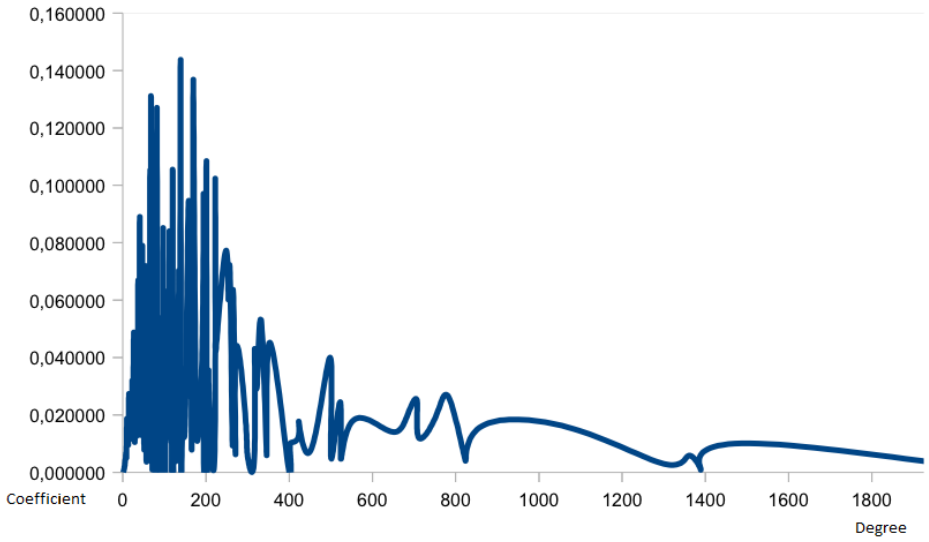


Figure 5.4 Clustering Coefficient of the CAIDA Internet AS graph

presented in [46] (0.39), the presented values are in line with the assortativity values in the last section.

As almost all neighbours of periphery nodes are from other classes which provide Internet connectivity, the clustering coefficient of periphery nodes suggests that is unlikely for the providers of periphery nodes to have connections between them. The substantial difference between the clustering coefficient of periphery ASes to the others is explained by the high percentage of nodes which have clustering coefficient equal to 0, *c.f.* table 5.6.

As regards to core nodes, they present the highest clustering coefficient values since there are several customers, from periphery and intermediate classes, which have a provider from the core in common and also due to the density of relationships in the core. Although there is no kernel node with zero clustering coefficient, these nodes present an average clustering coefficient lower than the one of core nodes. It is less likely to have a node with two connected kernel providers than having two connected core providers, or one provider from the core and other from the kernel.

Table 5.4 Average Clustering Coefficient per Class

Class	Clustering Coefficient
`Periphery`	0.000426
`Intermediate`	0.010850
`Core`	0.035384
`Kernel`	0.023835

Table 5.5 Distribution per Class of Triangular Relationships

Class	Periphery (%)	Intermediate (%)	Core (%)	Kernel (%)
Periphery	2	7	25	66
Intermediate	2	5	17	76
Core	3	6	20	71
Kernel	3	13	35	49

Table 5.6 Proportion of Nodes with Zero Clustering per Class

Class	Periphery (%)	Intermediate (%)	Core (%)	Kernel (%)
	33	3	3	0

5.2.3 Small World and Network Navigability

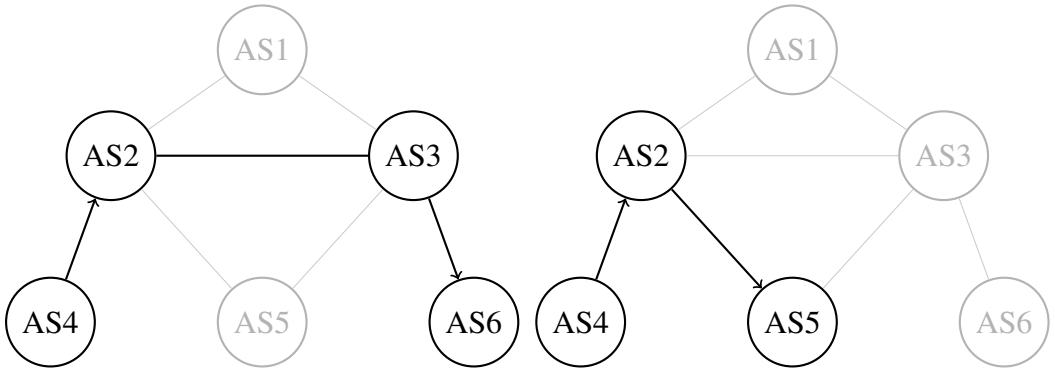
Despite its growth, in the last decade the length of AS paths has remained stable and near to 4 hops [22]. This result is in line with the *superscalability* property defined in [49], since the computed exponent of the degree distribution of a recent instance of the Internet AS graph was 2.1 [41]. Therefore, the Internet also holds the small-world property.

As regards to network navigability, one has to be careful when applying to the Internet AS graph a mechanism defined for other scale-free network. Valid paths in the Internet AS graph are *valley-free*, *i.e.*, it is not possible to have a path which goes from a provider-customer relationship to customer-provider relationship, as represented in figure 5.6. Normal paths in this graph are of the following form: a set of customer-provider edges, followed by a peering edge and then a set of provider-customer edges, *c.f.* figure 5.5 (b). Some paths are only composed by a set of customer-provider edges followed by a set of provider-customer edges, as represented in figure 5.5 (a). Additionally, the usage of a peering edge is restricted to the customers of the ending nodes, to their customers, and so on.

Some scale-free networks do not have restrictions regarding network navigability, though others have some navigability restrictions like the Internet. For instance in railway networks consider two stations, B and C, connected to a station A which is connected to a station D; if a train comes from station B to A it can go to station D, however if it comes from station C in cannot continue to station D as station C is used for regional trains and station D is restricted to international trains. In figure 5.7 it is shown the layout of such network.

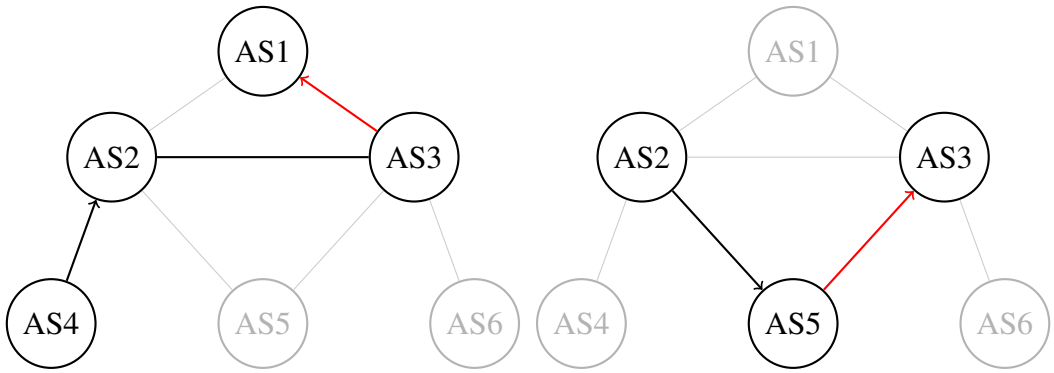
5.2.4 Network Growth and Construction Algorithm

Based on the work of Barábasi and Albert [5], in [60] it was defined a richer model for the growth and construction of the Internet : the Multi-class preferential attachment model (MPA). Firstly, ASes are divided in two different types: ISPs and non-ISPs. A new AS would never connect to an existing non-ISP since it does not provide connectivity. While in the preferential attachment (PA) model of Barábasi and Albert new nodes of the same type are added at a given



(a) provider-customer→peer-peer→customer-provider (b) provider-customer → customer-provider

Figure 5.5 Valid Paths through an AS topology



(a) provider-customer→peer-peer→provider-customer (b) customer-provider → provider-customer

Figure 5.6 Invalid Paths through an AS topology

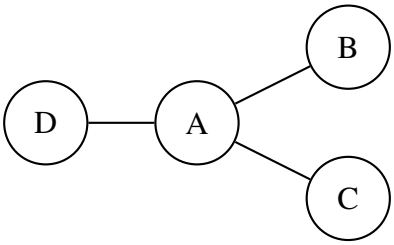


Figure 5.7 Example of a network with navigability restrictions

rate r, in the MPA model ISPs nodes are added at a rate 1 and non-ISPs at a rate ρ, connecting to existing ISPs with linear preference regarding node degree.

Due to cost-saving measures, relationships between ASes change over time. If the amount

of traffic flowing between two ISPs is about the same in both directions, a peering relationship between these two ISPs would help them to decrease their transit costs. Assuming that all customer ASes generate similar volumes of traffic, highly connected ASes would exchange high volume of traffic among themselves, and therefore would establish a peering link. Peering links are added at a rate c and the probability of a pair of ASes establish a peering link is proportional to the product of their degrees.

If an ISP goes bankrupt, it is normally acquired by another ISP which then either merges its infrastructure with the infrastructure of the bankrupted AS or form a *sibling* relationship, *i.e.*, two apparently independent ASes which are controlled by the same organisation. The rate at which bankruptcy happens is denoted by μ.

In order to improve its connectivity and reliability, an AS may decide to connect to more than one ISP, *i.e.*, to become multihomed. Multihomming links appear at a rate ν and the probability that an ISPs I_c node becomes multihomed is proportional to the product of its degree and the degree of the ISP I_p to which it will connect, assuming that I_p has a higher degree than I_c. Additionally, non-ISP nodes form multihome links by an average of m providers each.

This model constructs networks with power-law exponent being equal to:

$$\alpha = 2 + \frac{1-\mu}{1+2\nu+m\rho+2c+\mu} \tag{5.2}$$

From data collected by the RouteViews project, and the results of other studies, the value of α was measured as being $\alpha = 2.114$ which matches the observed empirical results of the power-law exponent of the Internet degree distribution [60]. Note that rates ρ, c, μ and ν are related with economic and commercial factors that influence the evolution of the Internet: commercial success, customers growth, cost of links, relation between the local and long distance links costs.

5.3 Topology Generators and AS Graph Annotations

Based on Internet models, *e.g.*, *dK-Series*, several Internet topology generators (e.g INET [31], BRITE [45], Orbis [41], ...) can be used to build network models of a required size. Some generators like INET [31] and BRITE [45] rely on the *1K relation* using power law degree distributions to model the AS-level Internet graph. Orbis [41] is a recent graph generation and rescaling tool relying upon the *dK-Series* model, which comprises more topological properties of the original graph than the others since it uses *2K relations*. Therefore, generated topologies maintain most of the known topological properties of the original graph used as model. Orbis relies upon the *dK-Series* model to capture the most important connectivity relationships among nodes of a given graph to generate similar graphs of different sizes, as regards to connectivity properties, using the *scaling* technique [42]. It is also possible to maintain the size of the graph and generate similar ones, with different edges, using the *rewiring* technique [42].

Scientific experiments requiring Internet models rely upon topology generators to build network graphs. In order to one be able to use synthetically generated graphs to study the behaviour of protocols in this networks graph model, node and edges must be annotated with properties besides connectivity. Among the most important ones are latency, capacity and type (customer-provider, peering, *etc.*) for edges and type of nodes, *e.g.*, the layer to which a node belongs to. Although good quality topology generators are available, annotation of graphs nodes and edges with attributes is an issue for which well founded methodologies are still lacking. Even if an original graph is annotated with the aforementioned information, topology generators cannot include that information on rescalled graphs.

A typical AS classification is dividing ASes into two layers, stub and transit tiers, where stub ASes are randomly chosen or selected as the ASes with the lowest degree whereas transit ASes are the non-stub ASes. We have proposed a layering approach to annotate (rescaled) AS graphs [58], supported by the aforementioned properties of scale-free networks, using another instance of the AS graph made available by CAIDA [18]. It consists of 20305 ASes automatically annotated by a machine learning algorithm [20] with AS type information, using attributes in Internet Routing Registries (IRRs [57]) such as the number and type of links (peer-to-peer and customer-to-provider), the number of IP prefixes and the size of the address space announced. The AS types considered are:

- `T1` - Large ISPs (large backbone providers, mostly Tier-1 ISPs);
- `T2` - Small ISPs (regional providers, mostly Tier-2 ISPs);
- `EDU_COMP` - Customer ASes (mostly universities and companies);
- `IXP` - Internet exchange points;
- `NIC` - Network information centres;
- `ABSTAINED` - Nodes not classified by the algorithm.

The class distribution of CAIDA AS graph [18] is shown in table 5.7.

Table 5.7 ASes per class from CAIDA AS graph [18]

Class	Size	%
T1	44	0.2
T2	5599	27.6
EDU_COMP	12606	62.1
IXP	33	0.2
NIC	332	1.6
ABSTAINED	1691	8.3
TOTAL	20305	100

Although this classification achieves a precision of 80% it cannot be used in rescalled graphs as the information contained in the IRRs cannot be applied to those graphs. In our AS classification, the AS graph is divided in four layers, representing the four classes previously defined

in sub-section 5.2.1, namely: periphery, intermediate, core, kernel. Recalling from last chapter, in scale-free networks having power-law exponent $\alpha \in [1,2]$ there is a spontaneous emergence of a core which possesses a property dubbed as soft hierarchy [49] composed by three layers:

1. nodes with degree $\in [N^{\varepsilon(N)}, \sqrt{N}]$;
2. nodes with degree $\in [\sqrt{N}, N^{1/\alpha}]$;
3. nodes with degree $\geq N^{1/\alpha}$.

We structure the core with soft hierarchy [49] into two tiers: *kernel* tier composed of nodes with degree $> \sqrt{N}$; *core* tier composed of nodes with degree $\in]N^{\varepsilon(N)}; \sqrt{N}]$.

The value of $N^{\varepsilon(N)}$ is defined as being slightly larger than $1/logN$ [49], though a concrete value is not specified. In order to compute a reasonable value for $\varepsilon(N)$ we used *betweeness* [1] as a shortest path heuristic. We obtained $\varepsilon(N) \simeq 0.33$ and $N^{\varepsilon(N)} \simeq 27.37$ corresponding to a core according to the following criterion, using the CAIDA AS graph [18]:

a) considering ζ as the set of nodes pertaining to the resulting core for a specific value of $\varepsilon(N)$, with size n_{core};

b) considering β as the set of nodes with the highest values of betweeness, with size n_{core};

c) considering μ as the size of $\zeta \cap \beta$;

d) maximise $\phi = \mu / n_{core}$.

This maximises the number of nodes in the which have the highest values of betweeness.

Although the layering approach relies upon nodes degrees, the class distribution of ASes in core and kernel tiers of the CAIDA AS graph - *c.f.* table 5.8 - is in line with the Internet structure since: *a*) the majority of T1 ASes are present in the kernel tier; *b*) all T1 ASes pertain either to kernel or core tiers; *c*) kernel tier contains no EDU_COMP ASes, whereas there are few in the core tier; *d*) only T2 with big infrastructures belong to kernel and core tiers.

5.3.1 Periphery Tier Identification

For the non-core nodes, a common used approach consists in identifying as periphery ASes (stub-ASes), nodes whose degree is ≤ 2. The following table presents the result of applying that approach on the CAIDA AS graph [18].

Since `EDU_COMP` do not provide service to others, the majority of those ASes should be identified as stub-ASes. We identify as periphery ASes nodes whose degree is ≤ 3 in order to include more `EDU_COMP` ASes, as it can be seen in the following table.

[1] Betweeness is the number of shortest paths that pass through a given node.

Table 5.8 Class distribution of ASes in core and kernel tiers

(a) Kernel tier

Class	Count
T1	39
T2	5
EDU_COMP	0
IXP	0
NIC	0

(b) Core tier

Class	Count
T1	5
T2	207
EDU_COMP	2
IXP	2
NIC	5

Table 5.9 ASes per class with degree ≤ 2, from the CAIDA AS graph [18]

Class	Count	%
T1	0	0
T2	3046	54.4
EDU_COMP	10919	86.6
IXP	13	39.3
NIC	247	74.3

Table 5.10 ASes per class with degree ≤ 3, from the CAIDA AS graph [18]

Class	Count	%
T1	0	0
T2	3575	63.9
EDU_COMP	12132	96.2
IXP	16	48.4
NIC	287	86.4

5.3.2 Validation

In order to verify if the presented layering approach is still valid for rescalled graphs, we computed 20 graphs from CAIDA AS graph [18], using `scaleTopology` tool from Orbis, 10 with about 10000 nodes and 10 with about 5000 nodes[2].

In the following tables we present the node distribution per tier for the CAIDA AS graph [18] and the average of node distributions for rescaled graphs with about 10000 and 5000 nodes.

The node distributions per tier are very similar. Therefore, applying the presented layering approach on rescaled graphs preserve the hierarchical structure invariant of the original CAIDA AS graph [18]. Accordingly, this approach can be used to structure AS level graphs, rescaled

[2]It is not known the factor to which a given graph can be rescaled and still preserve its main original characteristics.

Table 5.11 ASes per tier from the CAIDA AS graph [18]

Tier	%
Kernel	0.2
Core	1.1
Intermediate	12.8
Periphery	85.9

Table 5.12 ASes per tier from rescaled graphs

(a) 10000 nodes

Tier	%
Kernel	0.4
Core	2.3
Intermediate	12.9
Periphery	84.4

(b) 5000 nodes

Tier	%
Kernel	0.6
Core	2.8
Intermediate	12.6
Periphery	84.0

from the CAIDA AS graph [18], in four layers (kernel, core, intermediate tier and periphery tier), which induce a classification of ASes based on the layer that each AS pertains.

Additionally, from the graphs mentioned above we computed the distribution of link type: links within the same layer are considered as peer-to-peer (p−p), whereas links connecting different layers are considered as client-provider (c−p and p−c). In the following tables we present the distribution of link type per tier, as a percentage of all links from a tier[3].

Table 5.13 Peering, Provider-Customer and Customer-Provider links per tier from the graph CAIDA AS graph [18]

Tier	p-p (%)	p-c (%)	c-p (%)
Core	15.4	84.6	-
Intermediate	16.6	38.1	45.3
Periphery	6.2	-	93.8

Rescaled graphs, from the graph CAIDA AS graph [18], approximately preserve the distribution of p−p, c−p and p−c links per tier of the graph Ω. Consequently, a classification of links based on connectivity between the identified layers can be used to define link annotations.

The layering approach can be used to define the following models: link capacity, end-node attachment and link-type. Firstly, using the four layers, one may set the capacity of inter-layer links and intra-layer links suitable for a given simulation purpose. Secondly, end-nodes can be randomly attached to periphery-ASes. For instance, in simulations regarding end-users, one can

[3]We merged the two tiers of the core in one for a more adequate analysis.

Table 5.14 Average Peering, Provider-Customer and Customer-Provider links per tier from rescaled graphs with about 10000 nodes

Tier	p-p (%)	p-c (%)	c-p (%)
Core	25.1	74.9	-
Intermediate	15.6	39.2	45.2
Periphery	10.3	-	89.7

Table 5.15 Average Peering, Provider-Customer and Customer-Provider links per tier from rescaled graphs with about 5000 nodes

Tier	p-p (%)	p-c (%)	c-p (%)
Core	20.9	78.9	-
Intermediate	13.9	36.9	49.2
Periphery	10.1	-	89.9

specify how many nodes are attached to each layer so that the attachment model can be adapted to specific simulation environment characteristics. Finally, a classification of client-provider, provider-client and peer-to-peer is applied to the links of a layered AS graph. Additionally, links between application end-nodes and periphery ASes are classified as stub links.

We developed a tool to annotate rescaled AS graphs, that preserves graph CAIDA AS graph [18] properties, which uses the previous defined models. The annotation process can be described as follows:

Input:

- size of the rescaled graph (n);
- distribution of capacity of intra-tier and inter-tier links;
- number of application end-nodes to attach on each layer and the capacity of attachment links (optional).

Process:

- Rescale the CAIDA AS graph to one with size n using `scaleTopology` tool from Orbis;
- Distribute nodes to each layer as defined in the layering approach;
- Attach application end-nodes to each layer according to input distribution (if specified);
- Annotate nodes with layer information and links with capacity values conforming to distribution of link capacities;

Output:

- List of n nodes annotated with layer type (coreTier1, coreTier2, intermediate, periphery and application) along with an ID;

- List of links with IDs of connected nodes and annotated with link type information;
- List of link types with properties of each type.

The resulting graphs can be used as input to simulators or emulators (*e.g.*, Modelnet) in studies requiring tractable models of the Internet, improving simulation/emulation results towards graphs annotated with common ad-hoc heuristics. Next we present a summarised example produced by the tool:

```
Nodes :
     nodeID = 0 nodeType = periphery
(...)

Links :
     src = 10366 dst = 10806 linkType = coreTier1-coreTier1
(...)

Link Type Configuration :
     linkType = coreTier1-coreTier1 capacity = 125 Mbps
(...)
```

Current state-of-the-art to build Internet models rely on topology generators able to produce graphs of different sizes, that mimic quite reasonably most interconnection metrics found in real-world Internet graphs. This is specially true for AS-level Internet graphs since the empirical data made available by CAIDA gives access to a very reasonably accurate AS graph. Although these tools can be used to generate rescaled graphs from an original graph, they generally do not produce annotated graphs with properties such as link and node types, capacities or latencies. Producing them is a major requirement in the current scientific practice related to networking and distributed systems studies. However, current state-of-the-art concerning Internet model graphs annotation is generally performed using intuition-based (or even ad hoc) heuristics.

The classification methodology presented above is based on heuristics derived from several invariant Internet properties and theoretical results concerning large-scale graphs, with properties similar to the Internet, which have node degree power law distributions, and possess a core. As topology properties drive our nodes layering determination, they can be used in the original graph, as well as in the rescaled graphs.

5.4 Summary

We have conducted an updated study regarding the topological properties of the Internet AS graph. Using a recent snapshot we have confirmed that the Internet AS graph is a scale-free networks since:

- the nodes degree distribution continues to follow a power-law distribution, with exponent $\alpha = 1.58$;
- the average length of AS paths has remained stable as 4 hops, which supports the fact that Internet holds the small world property;
- as most scale-free networks, the Internet AS graph is disassortative;
- although the average clustering coefficient of all nodes is lower than the one in other scale-free networks, it is in line with the assortativity results and commercial relationships;
- the network growth and construction model is derived from the model of Barábasi and Albert for any scale-free network.

Furthermore, the Internet AS graph has a very clear layering structure which can be mapped to AS types, being each layer mainly identified by nodes degree. We have shown that rescaled graphs constructed from the Internet AS graph, which comprise the same topological properties of the original graph, exhibit the same properties in what concerns hierarchy structured, defined by the layering taxonomy. From the presented taxonomy supported by the scale-free properties of the Internet AS graph, as well as the annotation models constructed from it, we have built a tool to annotate rescaled AS graphs which can be used to improve simulation results of experiments requiring inter-AS network models.

In the next chapter we will present a preliminary design of a greedy routing scheme for the Internet AS graph, supported by the topological properties detailed in this chapter.

6 . Greedy Routing in the Internet AS Graph

The application of greedy routing in the Internet comprises: *a*) the construction of a mapping of the network topology into a coordinate space; *b*) the definition of a distance function; *c*) the definition of a concrete greedy routing strategy and algorithm. The coordinate space along with the distance function define a metric space $\mathscr{M}$ that supports the definition of a greedy routing algorithm. There are two metrics that are commonly used to evaluate the suitability of a greedy routing algorithm to the Internet:

- stretch: the ratio between the length of paths chosen by the greedy routing algorithm and the length of paths of a reference algorithm, *e.g.*, shortest paths;
- success ratio: the percentage of nodes which are reachable through the greedy routing algorithm.

While in some networks, *e.g.*, wireless networks, overlay, *etc.*, it is possible to compute geographic coordinates for their nodes, or synthetic ones based on latency (*c.f.* embeddings presented in chapter 4), these methods are not well suited for the Internet AS Graph for various reasons:

- there are several ASes which do not have a well-defined geographic location, *e.g.*, tier-1;
- typically, coordinates devised from latency are dynamically computed, thus the coordinate system is not stable and does not ensure convergence;
- latencies are not symmetric, *i.e.*, it is common to have different latency values in both directions of a link;
- it does not always ensure the triangle inequality [67].

We have defined a method to assign synthetic coordinates based on routing requirements. We follow the approach of LISP [29], *cf.* chapter 3-section 2.1, which divides the IP address space into two address ones: locator and identifier space. Although hosts continue to use IP addresses (End-point Identifiers-EID) to communicate with each other, EIDs concern only with the domain where the host resides. Routing Locators (RLOCs) are used to reach hosts, *i.e.*, IP addresses hierarchically organised and bounded to a domain.

We take NIRA [66] as a model to organise the locator space. We consider a set of ASes which verify a definition of the presence of a core in scale-free networks, as the Internet AS graph. In NIRA tier-1 (core) ASes have globally unique IP prefixes from which they allocate non-overlapping sub-prefixes to their customers. There is a provider-customer hierarchy from each tier-1 AS composed by the set of its customers, direct and indirect, *i.e.*, which have a sub-prefix derived from the prefix of the tier-1 AS.

In our model each AS has a coordinate for each provider-hierarchy it pertains, each one representing a different way of reaching the core. As each prefix is bounded to one AS, a mapping component from a prefix to the corresponding set of coordinates is needed in order to perform routing at the inter-AS level. We assume that such component is present in an

architecture where our routing scheme could be applied. In addition, if two ASes having a provider-customer or peering relationship have more than one direct link between them, we only consider one (logical) connection between the two ASes in our routing scheme.

We have defined a preliminary approach in the embedding of the Internet AS graph, *i.e.*, a mapping of its topology in a geometric space, along with a distance function adequate to support a greedy routing algorithm. In the next sections we discuss the details of both.

6.1 Provider-Customer Hierarchies

Using the CAIDA AS graph [10] as input, a snapshot of the real AS graph with 33508 nodes introduced in the previous chapter, we present graph G_o as follows:

- $G_o\ (V_o, E_o)$ - oriented and weighted graph
- V_o refers to vertices and E_o refers to edges
- $\forall e \in E_o,\ e\,(src, dst, weight)$
  - $weight = -1$: customer-provider link
  - $weight = 1$: provider-customer link
  - $weight = 0$: peer-peer link
  - $weight = 2$: sibling-sibling link[1]

We divide the locator space in non-disjoint provider-customer hierarchies, *i.e.*, an AS can be in more than one provider-customer hierarchy. Each hierarchy is rooted in an AS from the core. Following the kernel definition presented in [49], *cf.* chapter 4, the core can be modelled as follows:

- $K_n\ (V_n, E_n)$ - a clique composed by peer-peer links
- $\forall v \in V_n : v \in V_o,\ degree(v) > \sqrt{\#(V_o)},\ \nexists e(v, v_i, -1) \in E_o, v_i \in V_o$

This definition of the core comprises all the ASes which have a degree $> \sqrt{\#(V_o)}$ [49] and that are transit-only, *i.e.*, which are not customer of any other AS. We extend the above definition to include ASes with degree $> \sqrt{\#(V_o)}$, that are not transit-only but are exclusively customers of ASes from the initial core definition. As a result, the core present in the CAIDA AS graph [10] is composed by 14 ASes.

A provider-customer hierarchy G_{v_k}, rooted at node $v_k \in V_n$, is modelled as follows:

- $\forall v_k \in V_n : G_{v_k} = (V_{v_k}, E_{v_k})$ - acyclic sub-graph of G_r
- $V_{v_k}1 = \{v_k\}$
- $\forall v_i \in V_{v_k}i, v_i \in successors_{provider-customer}(v, G_o), v \in V_{v_k}i-1, v_i \notin V_{v_k}1 \cup \ldots \cup V_{v_k}i-1$
  - $\forall s \in successors_{provider-customer}(v, G_o), \exists e(v, s, 1) \in E_o$

[1] A sibling-slibing link connects two ASes managed by the same company.

- $V_{v_k} = V_{v_k 1} \cup \ldots \cup V_{v_k n}$

Each provider-customer hierarchy has on average 27482 nodes and, considering each hierarchy as a tree, the maximum depth of all hierarchies is 9. Nearly 82% of the total nodes in the graph can be reached from each provider-customer hierarchy. However, not all ASes can be reached via these hierarchies. Firstly, there is a set of ASes that are only connected to other ones via peering or sibling links. As these ASes are not involved in global interdomain routing, their absence in the set of reachable ASes via provider-customer hierarchies does not compromise the functioning of the interdomain routing. Secondly, there are small sets of connected components that correspond to Research Networks which use BGP, though they are for private use. For instance, the Research Network in New Zealand composed by REANNZ National Research Network , GNS Science New Zealand, New Zealand Supercomputer Centre, HortResearch Limited, TheLoop Open School Network, Crop and Food Research and others. As a result, the size of the Giant Connected Component is 33300, which corresponds to 99.37% of all graph.

In table 6.1 it is shown the number of ASes which pertain to a given number n of hierarchies, $n \in \{1..\#(V_n)\}$. Two patterns can be identified in table 6.1: 17 % of total nodes pertain to few hierarchies (1 or 2) and 79 % of total nodes are in all hierarchies. Although in the aforementioned greedy routing proposals each node is only represented by a single coordinate, in what concerns interdomain routing it does not allow to control path choice, *i.e.*, how a message passes through the core. As the majority of nodes are present in all hierarchies, in order to exploit such path diversity each AS has multiple coordinates in our model, *i.e.*, one for each provider-customer hierarchy to which it pertains. The variety of paths can be used in load balancing and route differentiation mechanisms, though such mechanisms should not be included in the routing algorithm to maintain its simplicity. In addition, our approach is inspired on the proposal [66].

Table 6.1 Number of nodes in n hierarchies, $n \in \{1..\#(V_n)\}$

# Nodes	Hierarchies
2701	1
2507	2
697	3
436	4
219	5
164	6
61	7
15	8
4	9
26496	14

6.2 An Euclidean Metric Space

We have defined a Metric Space (ξ, ρ) in the Euclidean Plane that supports the greedy routing algorithm. The set ξ is described in the next section, as a sub-set of $\mathbb{R}^2$. In the following section the metric ρ will be defined.

6.2.1 Coordinate Distribution Model

The distribution of coordinates comprises two phases: firstly, the ones of core nodes are manually assigned; secondly, from the one of the core node, coordinates are set along the correspondent provider-customer hierarchy. This process can be defined as follows:

- $\forall v_k \in V_n$, its coordinates are manually assigned;
- $\forall v_k \in V_n, \forall v \in V_{v_k}$ with coordinates = (v_x, v_y),
  $\forall v_s \in successors_{provider-customer}(v, G_{v_k})$, v assigns $(f(v_x, v_s), f(v_y, v_s))$ to v_s

We concentrate our design in the $\mathbb{R}^{2+}$ part of the Euclidean Plane, though it can be applied to the all $\mathbb{R}^2$ as it will be discussed later. Core nodes are disposed in a semi-circumference with radius = $\#(V_n)^4$, in order to guarantee there is enough space for the first customers of core nodes. As core nodes are the ones with the highest degrees, the first level of customers is the one with the highest number of nodes.

Each core node is associated with an arc of the semi-circumference and is placed in the middle of it, as can be seen in the figure 6.1. The lines which pass through the ends of each core node arc delimit the region associated with that node, where the coordinates of its provider-customer hierarchy nodes will be assigned. Additionally, each core node is associated with an angle which corresponds to the rotation applied to determine its coordinates.

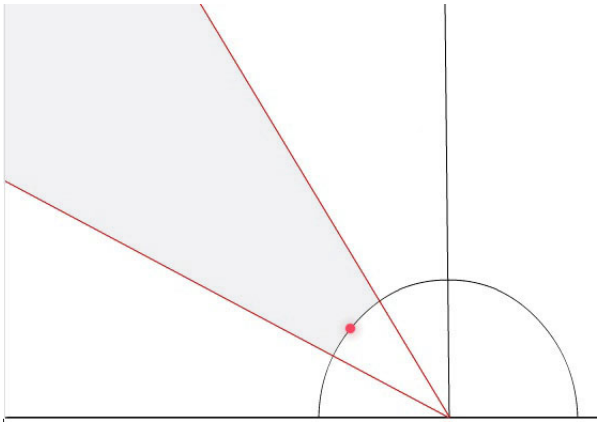


Figure 6.1 Region of a core node

For simplicity, we will only define the coordinate assignment method from a core node, and then describe the difference for further levels of the provider-customer hierarchy. Each core

node has the following information: *a*) distance to centre: core radius; *b*) its angle; *c*) *growth factor*= $(V_n)^2$ of distance to centre; *d*) boundaries of its region in the x-Axis: min, max and its width $\chi = (max - min) =$ kernel_radius$/\#(V_n)$.

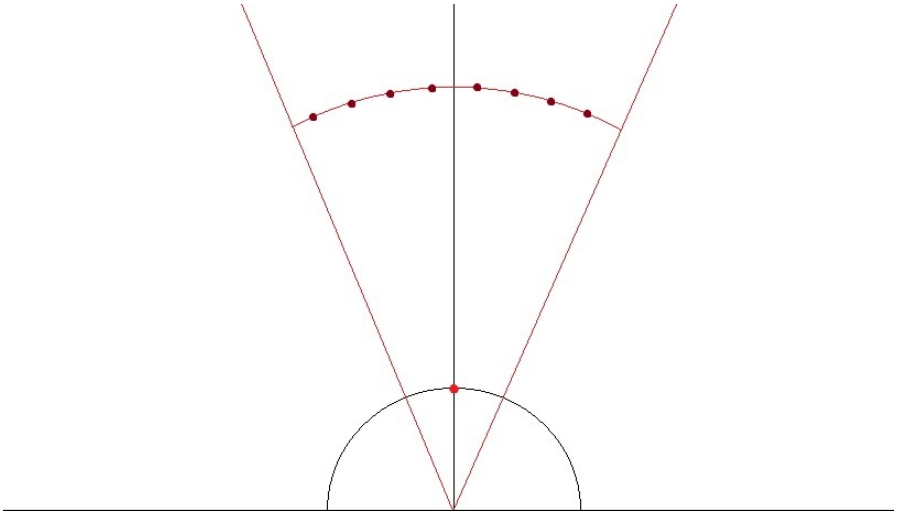


Figure 6.2 Coordinate assignment to customers of a core node

The kernel node customers are placed in an arc of the semi-circumference centred in the origin with *radius* = distance_to_centre × growth_factor, as shown in figure 6.2. Consider n_c as the number of customers of the kernel node, the region width χ is divided in $(3 \times n_c + n_c + 2)$ spaces with length δ. Starting from $min + \delta$, each customer is placed at the middle of its region and within $3 \times \delta$ to the consecutive customer. Since all customers are placed in a semi-circumference centred in the origin, they all have the same norm[2], though the distance between all customers and the provider is not the same. Those spaces are needed to ensure that in each hop, when moving towards a destination node, it is chosen the provider which will lead to the destination node, *i.e.*, the closest node to the destination is the provider of the provider of the ... of the destination node. The discussion on the concrete distance function will be done further in this section. Moreover, each customer has a region of $3 \times \delta$ width and correspondent boundaries $min_c = min + \delta + 3 \times (i-1) \times \delta$ and $max_c = min + \delta + 3 \times (i-1) \times \delta + \delta$, being $i \in \{1, ..., n_c\}$. Finally, a rotation of the kernel node angle is applied to determine the final coordinates of each customer.

In further levels of a provider-customer hierarchy the same method is applied, though with different values from each parent node. The *growth factor* is the same for all levels as well as the kernel node angle in each provider-customer hierarchy. The distance to centre and the boundaries of each node region are the values which are specific to each node.

[2]The norm of a node $a\,(x_a, y_a)$ is its distance to the origin, *i.e.*, $||a|| = \sqrt{x_a^2 + y_a^2}$

Additionally, free sub-regions can be left in each AS region in order to anticipate new customers based on the predicted evolution of the Internet AS graph [60], without having to reformulate the coordinate system. As opposed to IP prefixes which cannot be subdivided indefinitely, the region in each expands till infinity.

6.2.2 Metric

Given two points in $\xi \subset \mathbb{R}^2$, $a(x_a, y_a)$ and $b(x_b, y_b)$, the distance between a and b is given by the following expression:

$$\rho(a,b) = (\parallel a \parallel + \parallel b \parallel) \times \varepsilon(a,b) \tag{6.1}$$

where $\varepsilon(a,b)$ corresponds to the euclidean distance in $\mathbb{R}^2$ and $||a|| = \varepsilon(a,(0,0))$. We start by proving the properties of metric ρ:

1. *non-negativity*: $\rho(a,b) \geq 0 \implies (\parallel a \parallel + \parallel b \parallel) \times \varepsilon(a,b) \geq 0$;
   by definition $\varepsilon(a,b) \geq 0$ and $\parallel a \parallel \geq 0$ also $\parallel b \parallel \geq 0$;
   therefore $\rho(a,b) \geq 0$.

2. *symmetry*: $\rho(a,b) = \rho(b,a) \implies (\parallel a \parallel + \parallel b \parallel) \times \varepsilon(a,b) =$
   $(\parallel b \parallel + \parallel a \parallel) \times \varepsilon(b,a)$;
   by definition $\varepsilon(a,b) = \varepsilon(b,a)$ and $(\parallel a \parallel + \parallel b \parallel) = (\parallel b \parallel + \parallel a \parallel)$;
   therefore $\rho(a,b) = \rho(b,a)$.

3. *triangle inequality*: $\rho(a,c) \leq \rho(a,b) + \rho(b,c)$, *c.f.* appendix.

We combine the norms of each point with their euclidean distance to simulate the distance expansion of hyperbolic geometry models. Although in planar models of hyperbolic geometry there is a distance expansion as we get closer to the boundary, we set the boundary to the point (0,0) and define an opposite behaviour: as nodes get far from the boundary, the distance between them expands with their norms. Informally, if we set the boundary as infinity, the same behaviour as in metrics for hyperbolic geometry models can be modelled.

6.3 Greedy Routing in an Euclidean Metric Space

The classical greedy forwarding strategy does not consider the distance between the current node and its neighbours, only the one between the neighbour and the destination node. We have made a slight modification to the classical greedy forwarding strategy. In our greedy routing algorithm the node selected in each node is the one which matches the following condition:

$$min(\ \rho(current, neighbour) + \rho(neighbour, destination)\ ) \tag{6.2}$$

We start by presenting a base algorithm that makes all packets cross the core. We will present an optimisation of this base algorithm later on.

We divide the functioning of the greedy routing algorithm for interdomain routing in two modes: routing in different hierarchies and routing in the same hierarchy. As regards to routing in different hierarchies, a normal route is of the form: a chain of customer-provider links towards the core, a peering link in the core followed by a chain of provider-customer links towards the destination. The distribution of coordinates along with the metric ρ lead to the following route: shortest path from the source node to the core, one hop in the core, followed by the shortest path from the core to the destination. The choice of coordinates determines from which provider-customer hierarchies the message goes through, towards the core and towards the destination. The complete proof that the metric ρ leads to the mentioned path between ASes from different hierarchies is presented in the appendix. Here we only present a proof sketch.

$$r - \text{core semi-circumference radius}$$
$$g - \text{growth factor of distance to centre}$$
$$\alpha - \text{boundary of the core node region}$$

`Part 1` - *descending phase* towards the core:

$$\textbf{Nodes: } c_1(\alpha g^{level_1+1}, r g^{level_1+1}),\ c_2(\alpha g^{level_1+2}, r g^{level_1+2}),$$
$$p_1(\alpha g^{level_1}, r g^{level_1}),\ dst(-\alpha g^{level_2}, -r g^{level_2})$$

$$\rho(c_1,p_1)+\rho(p_1,dst) < \rho(c_1,c_2)+\rho(c_2,dst)$$

`Part 2` - *ascending phase* towards the destination:

$$\textbf{Nodes: } p_1(0, r g^{level}),\ p_2(\alpha g^{level+1}, r g^{level+1}),$$
$$p_3(-\alpha g^{level+2}, r g^{level+2}),\ dst(-\alpha g^{level+i}, -r g^{level+i})$$

$$\rho(p_2,p_3)+\rho(p_3,dst) < \rho(p_2,p_1)+\rho(p_1,dst)$$

The first part of the proof concerns the *descending* phase towards the core. Here the decreasing of norms of providers takes precedence over the euclidean distance. Node p_1 represents the provider of c_1; c_2 is a customer of c_1 and dst is the destination node in other hierarchy in the

opposite side of the hierarchy of nodes p_1, c_1 and c_2. On the *ascending* phase towards the destination, the inclusion of $\rho(current, neighbour)$ in the distance formula plays a crucial role so that the decreasing of euclidean distance with increasing norms takes precedence over going back to the core, *i.e.*, nodes having smaller norms. The chosen nodes are not related to the ones from first part. Node p_1 represents the root of the provider customer hierarchy, *i.e.*, the node from the core; p_2 is a customer of p_1 and p_3 is a customer of p_2. The destination node *dst* is at a *higher* level than p_3, *i.e.*, it can either be a direct customer of p_3 ($i = 1$) or a customer of a customer of p_3 ($i = 2$), and so on. Since in Euclidean geometry the rotation transformation preserves distances, in order to simplify the notation of the above proof we have rotated a provider-customer hierarchy such that the root node is placed in the $y - Axis$.

In what concerns routing in the same hierarchy, the shortest path would be a chain of customer-provider links, an *inversion* on the path from customer-provider links to provider-customer links, in a common provider of source and destination nodes, followed by a chain of provider-customer links towards the destination. However, with metric ρ the choice of the *inversion* node can be faulty, *i.e.*, it can lead to a dead-end, *c.f.* figure 6.3. Node $AS5$ is the neighbour of $AS2$ that is closer to the DST node, though it is impossible to reach DST node from $AS2$. As source and destination nodes are too close, we enforce the following path: shortest path to the core node of that hierarchy followed by the shortest path towards the destination. This path is identified in figure 6.3 by blue links: $SRC - AS4, AS4 - AS2, AS2 - AS1, AS1 - AS3, AS3 - AS6$ and finally $AS6 - DST$.

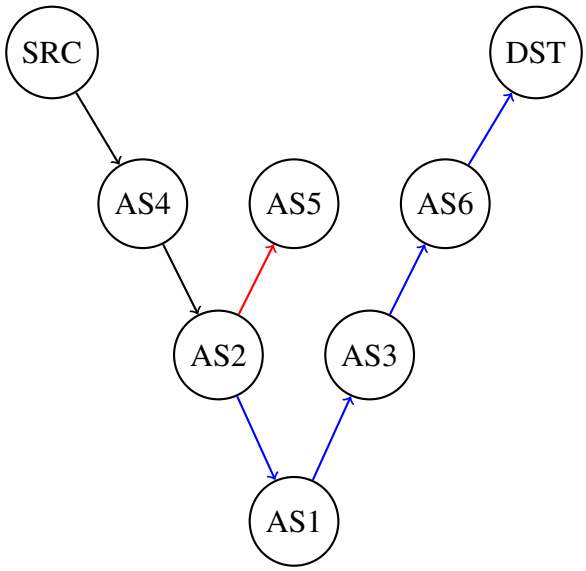


Figure 6.3 Invalid choice of the *Inversion* AS

6.4 Evaluation

We have verified that our approach achieves full success ratio. To evaluate the average stretch of this algorithm, we calculated for all node pairs the ratio between the length of the path chosen by our greedy routing algorithm and the length of the predicted path chosen by BGP. Since for the majority of nodes there are several possible paths between them, we select the shortest one between each pair of nodes. Using the CAIDA AS graph we have computed the predicted paths chosen by BGP using a modified version of the Dijkstra's algorithm. As the CAIDA AS graph only comprises node adjacencies annotated with link relationship type, all the concrete paths used by BGP are unknown[3]. Therefore, we computed the predicted paths of BGP, *i.e.*, the ones that do not break the navigability restrictions discussed in the last chapter. As a result, when two ASes have a peering link, all their clients can use it. Although this is not a completely accurate model of BGP paths, it is in line with the computed paths using our greedy routing scheme.

Table 6.2 shows the aggregated values of average stretch of all nodes, from each of the four classes in the taxonomy presented in the last chapter [58]. In addition, the overall average stretch obtained was approximately 1.516. This value is due to the fact that the above base greedy routing algorithm only explores strict hierarchically paths, ignoring intra and inter-hierarchies peering links. Moreover, since when routing in the same hierarchy packets have to pass through the hierarchy root, *i.e.*, the node from the core, a common provider in the way to the core is not used. Periphery nodes present the highest stretch values since they are the ones which are more penalised by the limitations of the base greedy routing scheme, whereas the nodes from the core are hardly affected since all nodes are directly connected to, at least, one node from the kernel. Intermediate nodes have lower stretch values than periphery ones since they can reach the core in less hops.

Table 6.2 Average stretch for each class of the base greedy routing scheme

Class	Average Stretch
`Kernel`	1
`Core`	1.005
`Intermediate`	1.263
`Periphery`	1.571

These values may seem smaller than expected since only inter-hierarchy are used in the base greedy routing scheme. Next we introduce some results which explain the obtained average stretch values.

As regards to the non-usage of peering links, table 6.3 presents their distribution per level, being level 1 the kernel, level 2 the customer of the kernel, and so on. It shows that peering links are mainly between nodes close to the kernel.

[3]Even though the CAIDA AS graph is computed using BGP messages, the ones regarding peering links are not captured since they are private to the ASes involved in each peering link.

Table 6.3 Distribution of Peering Links per Level

Level	1	2	3	4
1	*182*	320	16	-
2	320	3942	413	7
3	16	413	83	-
4	-	7	-	-

In what concerns routing within the same hierarchy, the impact of forcing packets to unnecessarily pass through the provider-customer hierarchy root is shown in table 6.4. For almost 85% of the intra-hierarchy paths, few hops are added to the shortest path which is not used. The longest intra-hierarchy paths have less impact in the overall average stretch, since they are approximately 7 % of the intra-hierarchy paths.

Table 6.4 Additional Hops in routing between nodes within the same hierarchy

Add. Hops	# Pair of Nodes	Proportion %	Cumulative Prop. %
2	152943698	16.66	16.66
4	306897229	33.43	50.09
6	315434902	34.36	84.45
8	80511178	8.77	93.22
10	41311323	4.5	97.72
12	14504865	1.58	99.29
14	6426206	0.7	100
`Total`	918029401	100	-

6.4.1 Optimisation of the Base Greedy Routing Scheme

Since each peering link between two ASes can only be used by their customers, the above coordinate model and the metric ρ are not compatible with this usage restriction, as can be seen in figure 6.4. The peering link AS2-AS4 is chosen over link AS2-AS1 since AS4 is geometrically closer to DST node. However, it leads to a dead-end since there is no way to reach DST through AS4. The valid path after AS2 using provider-customer links is represented in blue, *i.e.*, AS2-AS1, AS1-AS3 and finally AS3-DST.

Each AS has its own region from which it assigns coordinates and sub-regions to its clients, which in turn repeat the same process to their clients. Having a coordinate and a region, determining if the coordinate pertains to the region is straightforward. Therefore, the two ASes involved in a peering link can exchange their own regions in order to verify the possibility of usage of that peering link. Remember that a peering link can only be used by the direct and indirect customers of the two ASes involved in the peering link.

When choosing the next hop, if the selected neighbour is connected via a peering link, the

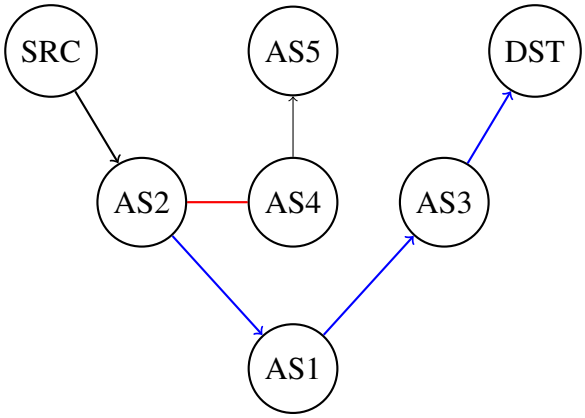


Figure 6.4 Invalid choice of a peering link in an AS topology

following conditions have to be verified:

- the source coordinates belong to a region of the current node, *i.e.*, it is a customer of the current node;
- the destination coordinate refers to a region of the other node in the peering link.

Each AS can control to which customer(s) it allows the peering link to be used, by sending sub-regions of its own region. However, this flexible control increases the amount of data that each AS has to maintain. It ranges from one entry per peering link when the other AS sends its complete region, to thousands of entries when the other AS sends permissions regarding a set of individual end-nodes coordinates. In addition, besides rare modifications, the data concerning the control of peering links does not increase the network traffic since it is only exchanged between the two ASes involved in the peering link.

Since metric ρ leads to an invalid choice of the *inversion* node when routing in the same hierarchy, *c.f.* figure 6.3, in the base greedy routing scheme we inforced packets to pass through the provider-customer hierarchy root in order to avoid dead-ends. If the metric ρ did not have this limitation, the choice of inverting the path from customer-provider links to provider-customer towards the destination would be made at a direct or indirect provider in common with the source and destination nodes. Ultimately, such node is the provider-customer hierarchy root.

The regions of customer nodes are assigned by their providers. Hence, each provider knows the sub-regions of each of its customers. The base greedy routing scheme can be extended in the following way for intra-hierarchy routing: in each hop, the current node verifies if the destination coordinate is from one of the regions of its customers; if the verification succeeds, the node sends the packet to the customer which region contains the destination coordinate. Therefore, this ensures intra-hierarchy shortest-path routing by correctly choosing the *inversion* node, not enforcing packets to unnecessarily pass through the provide-customer hierarchy root.

Although the routing algorithm comprising the aforementioned extensions is not a pure greedy routing algorithm, it does not compromise the scalability foundations of greedy routing.

In the extended greedy routing scheme, the routing state in each node continues to be in the order of the number of neighbours, rather than in the order of the number of nodes in the network. However, it is slightly higher, *i.e.*, in the order of the number of neighbours times the number of ASes in the kernel, in order to support multiple paths to a given node. Nevertheless, the number of ASes in the kernel is a small and almost invariable number - 14 in the CAIDA AS graph.

The aforementioned extensions improve the previous average values of stretch for nodes from core, intermediate and periphery tiers. In fact, an optimal value of stretch (1) is obtained for all nodes in all of the four classes, considering a shortest path metric.

6.4.2 Comparison with BGP

Next we present a discussion on how some of the features of BGP can be performed using our greedy routing scheme.

Our greedy routing scheme does not need to maintain a Forwarding Information Base (FIB) and a Routing Information Base (RIB) as the ones in BGP that concern all received prefixes. Only a FIB-like table is needed, though having only information regarding the direct neighbours.

BGP uses policy filters to prevent advertisements received from provider for being forwarded to another provider as well as advertisements from a peer to another peer. This is done to ensure that packets do not follow paths which violate navigability restrictions (*cf.* chapter 5 - subsection 2.3), *i.e.*, valley-free paths and correct usage of peering links. The paths induced by our greedy routing scheme do not violate those navigability restrictions, as shown earlier on this chapter.

Since our greedy routing scheme allows an AS to be reached via multiple routes it is possible to define a preference mechanism similar to the Local Preference attribute in BGP. When an AS receives the available coordinates of an AS it wants to send packets, *e.g.*, from the mapping system, it can check if the destination AS is one of its direct or indirect customer or one of the customers of one of its peers. As each AS knows its regions and the regions of its peers to which it can send packets, verifying if a given coordinate pertains to a given region is straightforward.

Additionally, if there are more than one physical link between two ASes, in BGP is possible for an AS to express from which point it prefers to receive traffic regarding a given prefix. Since there is a mapping between prefixes and coordinates/regions, ASes can exchange messages to express entry point traffic preference, similar to the configuration of peering links.

We will return to the analysis of what the proposed greedy routing algorithm has achieved in the next chapter. Next we analyse how it could be used in the context of a new architecture for interdomain routing.

6.5 Foundations of a New Architecture for Interdomain Routing

In the next subsections we discuss the components of a preliminary design of an architecture for interdomain routing, which uses the above greedy routing scheme to perform routing among ASes.

6.5.1 Mapping System

The Mapping System is responsible for returning the correspondent coordinate(s) for a given identifier. As our greedy routing scheme supports multiple coordinates per AS, *i.e.*, an AS can be reached via various alternative paths from the core, it is possible to control inbound load-balancing using one-to-many identifier-coordinate mappings. LISP [29], *c.f.* chapter 3-section 2, has a solution which allows to perform inbound load-balancing as well as route differentiation. Each RLOC (a coordinate in our scheme) is annotated with two fields: priority and weight. The former is used to identify a given class of traffic/applications, while the latter is used to specify the amount of traffic of that class which should be sent to that RLOC (coordinate).

As regards to its architecture, the mapping system should have a distributed hierarchical design as in Domain Name System (DNS). Such design would follow the hierarchical organisation of the coordinate space, being core ASes as root-name servers in DNS. Each AS would have its own servers in order to locally control inbound load-balancing mechanisms. Alternatively, in order to avoid having duplicated servers, the mapping system could be introduced in the existing DNS architecture.

6.5.2 Mobility

The mapping system inherently supports mobility. Since mobile hosts move to geographically closed locations, they either move to a different LAN in the same AS or to another in a close AS from the one they were initially. For nodes moving through different LANs within the same AS, Mobile IP [53] could be used. As regards to inter-AS mobility, if geographically closed ASes support identifiers from each other, the following mechanism can be applied to the mapping system [25]:

- consider a host x originally from AS α_1;
- host x moves to AS α_2;
- AS α_2 assigns (a) new coordinate(s) to host x;
- AS α_2 notifies AS α_1 to change the entry for host x identifier with the new coordinate(s).

Since each message carries both the identifier and locator, as soon as the message carrying the new locator is received, it can be inferred by the other host to guarantee connection/session survivability. However, new connections have to wait for the update in the mapping system of the AS α_1 to establish connections to host x.

6.5.3 Security

Although this does not directly concern routing, the architecture should comprise security mechanisms regarding AS authentication for exchanging of configuration messages between ASes. We propose a Public Key Infrastructure (PKI) following an oligarchical model: each Address Assignment Authority, *e.g.*, RIPE, is a Certification Authority (CA) and signs the certificate of the other CAs. Then, each Address Assignment Authority (CA) generates a Public Key Certificate for each AS.

The security protocol for exchanging configuration messages between ASes would be based on a three-way handshake to: *a*) authenticate both ASes; *b*) exchange shared symmetric session keys, which would never be used in further sessions. This mechanism provides the following properties:

- **perfect backward secrecy**: as each session key is never reused, the agreed keys will not be compromised even if an attacker can obtain a session key from a subsequent session, derived from the same long-term secrete (private-key);
- **perfect forward secrecy**: if the agreed key of a session is compromised, an attacker cannot decipher the information exchanged in further sessions;
- **integrity**: verification of data modification by an attacker;
- **confidentiality**: prevents eavesdropping and traffic analysis from an attacker;
- **authentication**: both-sides authentication;
- **non-repudiation**: message signatures using private-keys improve non-repudiation guarantees.

6.5.4 Fault-Management

In the current architecture BGP deals with short and long-term faults, being the former one of the main causes for the high level of message churn in BGP. Our greedy routing scheme does not comprise any mechanism for fault management, as any pure greedy routing scheme. In order to keep the routing protocol as simple as possible, we believe that the management of short-term failures should be done at the periphery. If the destination AS has more than one coordinate, it can be reached by various alternative paths.

Hosts can monitor the availability of the coordinate currently in use, and move to another if it is suspected that the path used by the current coordinate has faced some failure. In order to improve connection reliability a host can send packets to two coordinates in parallel [68]. Furthermore, as regards to long-term faults, they should be advertised to the ASes which are compromised by that failure, *i.e.*, the direct and indirect customers of the AS that is directly connected to the faulty link. After being noticed, each AS can remove the coordinate affected by the failure from its mapping server.

6.5.5 Implications on End-Nodes

In order to support traffic engineering, mobility and fault-management mechanisms, some changes have to be performed on protocols used by end-users. A possible solution is to switch to multipath versions of TCP and UDP that support multiple coordinates for a given identifier, traffic engineering mechanisms such as load-balancing among the available coordinates, monitoring of coordinate availability as well as inference of new coordinates for an existing session of an identifier.

6.5.6 Migration Plan

A new architecture for interdomain routing has to provide a progressive migration plan since it is impossible to *shut-down* the current architecture and then migrate to the new one. Additionally, it is important for a new architecture to support legacy hosts. A common solution of the proposed new architectures for interdomain routing [25, 29] is using UDP tunnelling along with NAT-like components at the borders of each AS. In addition, in order to support legacy hosts that do not have a multipath versions of transport protocols, the mechanisms performed by such version can be done by Ingress and Egress Nodes in the AS border infrastructure.

6.6 Summary

Geographic coordinates spaces and synthetic ones based on latency cannot be used as solutions for embeddings in the AS graph due to several limitations of those models considering the restrictions of interdomain routing. We have defined a method to assign synthetic coordinates to ASes based on routing requirements. Those coordinates along with a distance function support a preliminary definition of a greedy routing scheme for interdomain routing. It achieves 100% success ratio, *i.e.*, it is possible to establish communication between every pair of ASes and an overall average stretch of 1.4. However, it does not allow the correct usage of peering links and does not ensure shortest-path routing within the same hierarchy. Moreover, we defined extension mechanisms to the base greedy routing scheme for peering-link correct usage and to ensure shortest-path intra-hierarchy routing without leading to dead-ends. As a result, the extended greedy routing scheme obtains an optimal overall average stretch (1).

As regards to scalability and convergence, the separation of IP address space into two address ones (identifier-locator) allows a hierarchical organisation of the locator space, in which our greedy routing scheme provides a scalable form of interdomain routing. Additionally, in our greedy routing scheme ASes do not continuously exchange messages concerning the routing protocol. Only configuration messages are rarely exchanged, such as coordinate attribution, peering links configuration and entry point traffic preference. In fact, most of these messages are only exchanged between pairs of ASes and are not propagated to almost the entire network. Moreover, the amount of data that each AS has to maintain is in the order of the number of provider-customer hierarchies times the number of neighbours, which is substantially smaller

than the number of coordinates (prefixes in BGP).

In order to keep the routing scheme as simple as possible, the other critical issues of BGP, as well as the design requirements presented in chapter 3, are addressed by components of an architecture for interdomain routing. We discussed the foundations of such architecture, proposing a new design for security and fault-management components as well as a different architecture for the mapping system. Mobility and traffic engineering components are inspired in LISP [29] and HAIR [25] architectural proposals.

In the next chapter we will return to the analysis of the results achieved by the proposed greedy routing scheme, from a critical point of view.

7 . Closing Remarks

Our main goals were to study the characteristics of the AS graph as a scale-free networks and to define a preliminary path in the definition of a greedy routing scheme for the Internet AS graph, as an alternative solution for BGP. We believe that we have fulfilled these goals since:

- we performed an updated study regarding the topological characteristics of the Internet AS graph, confirming that it continues to be a scale-free network;
- we defined a greedy routing scheme that: *a*) ensures shortest path routing; *b*) offers multiple ways to reach the core (tier-1 ASes); *c*) covers several features of BGP; *d*) improves almost all of BGP critical issues.
- we discuss a sketch of a design for a new architecture for interdomain routing that could use our greedy routing scheme. We next summarise the full scope of the work which resulted in this document.

We started our work by presenting a study of the current state-of-the-art of interdomain routing. We reviewed the functioning of the currently used protocol for interdomain routing (BGP) as well as identifying some problems which compromise its future. Namely, scalability and convergence, route's quality as well as lack of load balancing, quality of service and security. In addition, we discussed some proposals which focus on fixing some issues of BGP, mainly trying to reduce message churn in order to decrease convergence time. These proposals rely upon techniques to prevent withdrawal messages, flushing obsolete paths from the network and defining different delay intervals based on the type of update. However, as these proposals do not solve the most critical issues of BGP which affect its future, *i.e.*, scalability and convergence, they are dubbed as short-term fixes. As these patches may augment the complexity of BGP, their application is probably very limited.

Alternatively to those short-term fixes, researchers have been trying a totally new approach. In order to solve the identified problems of BGP, the definition of a new architecture for interdomain routing is required. We identified the main design requirements of a future design for interdomain routing: scalability, multi-homing support along with traffic engineering mechanisms, mobility support, security mechanisms as well as ways of having less burden in the core. Moreover, we discussed some current proposals for a new architecture that rely upon the principle of separating the IP address space into two address ones: locator and identifier. Some alternative routing schemes to BGP, which mainly focus on improving scalability, were also presented.

In parallel with the definition of new architectures and alternative routing schemes for interdomain routing, several studies concerning with the topological properties of large-scale networks were performed. From those studies, a new type of networks possessing unique properties was designated as scale-free networks. The node degree distribution of these networks follows a power-law distribution, having exponent between 1 and 3. The main property of this

distribution is scale invariance: applying a scale factor to the distribution variable leads only to a proportional scaling of the distribution, thus maintaining its properties. What is more, scale-free networks also possess the small-world property, *i.e.*, the shortest-paths between any pair of nodes in the network scales, at most, logarithmically with the network size. These networks are highly resilient in face of random attacks, though they are less robust in face of target attacks to highly connected nodes. Another important characteristic of scale-free networks is the network growth-construction model. The evolution of these networks follows the cumulative advantage principle: the probability of a given node receiving new connections is proportional to its degree - "the rich get richer".

From the results of an experience made in social-networks, where citizens were asked to forward letters based on their acquaintances, a new routing strategy suitable for scale-free networks was recently designated as greedy routing. Using such strategy, each node forwards messages using only information regarding its direct neighbours; in each hop the message is sent to the closest node towards the destination. Since the complexity at each node is not proportional to the size of the network but to the number of its neighbours, the usage of greedy routing is highly suitable for scale-free networks, as these networks have such scale which traditional routing schemes cannot properly manage. Furthermore, we performed a study on the topological properties of a recent snapshot of the Internet AS graph in order to support the design of a greedy routing scheme. We confirmed that the Internet AS graph still holds the topological characteristics of scale-free networks. We have also shown how these characteristics can be used to annotate rescaled AS graphs, generated from the Orbis tool, with suitable information for simulation and emulation environments which require such networks.

Finally, we devised an euclidean metric space to guide a greedy routing scheme for interdomain routing. The principle of IP address space separation was used as the starting point, since we focus on a hierarchically organised locator space. Inspired by the work of NIRA, we divided the AS graph in several provider-customer hierarchies rooted in one AS from the core. The core in our model follows the definition of emergence of a core in scale-free networks by Norros *et al.* [49]. The base greedy routing scheme allows the usage of multiple ways to transverse the core via shortest paths, though it does not use peering links and not guarantee intra-hierarchy shortest paths. We have made some optimisations that do not involve the dissemination of messages through the whole network. They permit correctly usage of peering links and ensure intra-hierarchy shortest paths. In addition, it covers several BGP features while solving some of the problems which affect the maintenance of BGP. Finally, we discussed a sketch design of a new architecture for interdomain routing that addresses most of the identified design requirements, while using our greedy routing scheme as the routing algorithm among ASes. The architecture comprises new components and other ones inspired from other architectural proposes, though some using a different design.

From a critical perspective, the choice of dividing the AS graph in strict hierarchies makes the functioning of our routing scheme similar to one using IP prefix intervals instead of regions of coordinates. In such scheme, each AS knows the IP interval of its neighbours and routing is performed in the following way: in each hop it is verified to which IP prefix the destination IP

pertains; if this verification returns more than one interval, it is selected the one with the smallest width. Although such scheme seems simpler than our greedy routing scheme, we emphasise that our main goal was not to define a scalable routing scheme as an alternative to BGP but to devise a preliminary path on the application of the greedy routing strategy to interdomain routing. Nevertheless, a routing scheme based on IP prefix intervals limits the growth in height of the AS graph since intervals cannot be subdivided indefinitely, whereas the regions of in our coordinate model continually expand till infinity.

Another possible critique of our work is the usage of the Euclidean space for the embedding of the AS graph instead of using the Hyperbolic space. We recall that the greedy routing proposals which use the Hyperbolic space consider a simpler model of the AS graph: each link results in an undirected edge, regardless of its type of commercial relationship. Additionally, those models do not comprise the choice of multiple paths between two nodes. Since we consider these technological restrictions and divide the AS graph into strict hierarchies, we obtain a simple model that makes an embedding in the Euclidean space feasible.

7.1 Future Work

First and foremost, in order to make the greedy routing scheme more realistic we must revisit the problem of policy control regarding links usage. BGP has base mechanisms that allow any AS to express in a very flexible way which paths are made available to peers/customers for their usage. These BGP mechanisms are so powerful that they even allow to express policies that lead to dead-ends and traffic loss. The requirements of policy routing are somehow orthogonal to routing. These have only been partially addressed in this work but are inevitable as future work.

In the discussed design, the complexity of the routing scheme is moved to other components, *e.g.*, mapping system. Notwithstanding the fact that it seams less complex to maintain information regarding the mapping systems, in comparison with the one managed by BGP, we need to evaluate the impact of the new architectural components to study their improvements.

When a host wants to establish a connection to other, it needs to enquire the mapping system for the available coordinates to the destination host. We need to measure the overhead in terms of message churn introduced by the set of queries sent to the ASes mapping servers. The discussion of caching models to diminish the number of queries is also a relevant one. In addition, it is important to quantify the impact of the identifier-coordinate(s) enquire on the time to establish a connection between two hosts.

The availability of alternative coordinates/paths to a given host can be used to improve traffic engineering and connection reliability. As regards to the latter, we need to measure: *a*) the time is needed to detect a failure that affects a given coordinate; *b*) the percentage of packet loss; *c*) the impact of sending data in parallel.

As regards to mobility, we need to assess the amount of time needed to perform the following operations:

- inference of the new coordinate(s) of a mobile host on the hosts having connections with the mobile one;
- update in the mapping server of the AS which *owns* the identifier of the mobile host.

A . Appendix

A.1 Proof of the path induced by the metric space (ξ, ρ)

Proof.

$$r - \text{core semi-circumference radius}$$
$$g - \text{growth factor of distance to centre}$$
$$\alpha - \text{boundary of the core node region}$$

`Part 1` - *descending phase* towards the core:

$$\textbf{Nodes: } c_1(\alpha\, g^{level_1+1}, r\, g^{level_1+1}),\ c_2(\alpha\, g^{level_1+2}, r\, g^{level_1+2}),$$
$$p_1(\alpha\, g^{level_1}, r\, g^{level_1}),\ dst(-\alpha\, g^{level_2}, -r\, g^{level_2})$$

$$\varepsilon(c_2, dst) = \sqrt{(\alpha\, g^{level_1+2} + \alpha\, g^{level_2})^2 + (r\, g^{level_1+2} + r\, g^{level_2})^2} = \tag{A.1}$$
$$= \sqrt{\alpha^2 (g^{level_1+2} + g^{level_2})^2 + r^2 (g^{level_1+2} + g^{level_2})^2} = \tag{A.2}$$
$$= \sqrt{(\alpha^2 + r^2)(g^{level_1+2} + g^{level_2})^2} = \sqrt{\alpha^2 + r^2}(g^{level_1+2} + g^{level_2}) \tag{A.3}$$

$$\varepsilon(p_1, dst) = \sqrt{\alpha^2 + r^2}(g^{level_1} + g^{level_2}),\ \varepsilon(c_1, dst) = \sqrt{\alpha^2 + r^2}(g^{level_1+1} + g^{level_2}) \tag{A.4}$$

$$\rho(p_1, dst) < \rho(c_2, dst) \iff \varepsilon(p_1, dst)(||p_1|| + ||dst||) < \varepsilon(c_2, dst)(||c_2|| + ||dst||) \tag{A.5}$$
$$\iff \sqrt{\alpha^2 + r}(g^{level_1} + g^{level_2})(||p_1|| + ||dst||) < \sqrt{\alpha^2 + r}(g^{level_1+2} + g^{level_2})(||c_2|| + ||dst||) \tag{A.6}$$
$$\iff g^{level_1} \times r\, g^{level_1} + g^{level_1}||dst|| + g^{level_2} \times r\, g^{level_1} + g^{level_2}||dst|| < \tag{A.7}$$
$$g^{level_1+2} \times r\, g^{level_1+2} + g^{level_1+2}||dst|| + g^{level_2} \times r\, g^{level_1+2} + g^{level_2}||dst|| \tag{A.8}$$
$$\iff g^{2level_1} \times r + g^{level_1}||dst|| + g^{level_2} \times r\, g^{level_1} < \tag{A.9}$$
$$g^{2level_1+4} \times r + g^{level_1+2}||dst|| + g^{level_2} \times r\, g^{level_1+2} \tag{A.10}$$

$$\rho(c_1,p_1) \simeq \rho(c_1,c_2) \tag{A.11}$$

$$\rho(c_1,p_1)+\rho(p_1,dst) < \rho(c_1,c_2)+\rho(c_2,dst) \tag{A.12}$$

`Part 2` - *ascending phase* towards the destination:

$$\textbf{Nodes: } p_1(0,r\,g^{level}),\ p_2(\alpha\,g^{level+1},r\,g^{level+1}),$$
$$p_3(-\alpha\,g^{level+2},r\,g^{level+2}),\ dst(-\alpha\,g^{level+i},-r\,g^{level+i})$$

$$\rho(p_3,dst) = \varepsilon(p_3,dst)(||p_3||+||dst||) = \tag{A.13}$$

$$= \sqrt{(-\alpha\,g^{level+2}+\alpha\,g^{level+2+i})^2+(r\,g^{level+2}-r\,g^{level+2+i})^2}(||p3||+||dst||) = \tag{A.14}$$

$$= \sqrt{(\alpha\,g^{level+2}(-1+g^i))^2+(r\,g^{level+2}(1-g^i))^2}(||p_3||+||dst||) = \tag{A.15}$$

$$= \sqrt{\alpha^2\,g^{2level+4}(-1+g^i)^2+r^2\,g^{2level+4}(1-g^i)^2}(||p_3||+||dst||) = \tag{A.16}$$

$$= \sqrt{g^{2level+4}(-1+g^i)^2\times(\alpha^2+r^2)}(r\,g^{level+2}+r\,g^{level+2+i}) = \tag{A.17}$$

$$= g^{level+2}(-1+g^i)\sqrt{\alpha^2+r^2}(r\,g^{level+2})(1+g^i) = \tag{A.18}$$

$$= g^{2level+4}(g^i-1)^2\sqrt{\alpha^2+r^2}\times r \tag{A.19}$$

$$\rho(p_1,dst) = \varepsilon(p_1,dst)(||p_1||+||dst||) = \tag{A.20}$$

$$\sqrt{(0-\alpha\,g^{level+2+i})^2+(r\,g^{level}-r\,g^{level+2+i})^2}(||p_1||+||dst||) = \tag{A.21}$$

$$= \sqrt{\alpha^2\,g^{2level+4+2i}+r^2\,g^{2level}(1-g^{2+1})^2}(||p_1||+||dst||) = \tag{A.22}$$

$$= \sqrt{g^{2level}(\alpha^2\,g^{4+2i}+r^2(1-g^{2+i})^2)}(r\,g^{level}+r\,g^{level+2+i}) = \tag{A.23}$$

$$= g^{level}r(1+g^{2+i})\sqrt{\alpha^2\,g^{4+2i}+r^2(1-g^{2+i})^2} \tag{A.24}$$

$$\rho(p_2,p_3) = \varepsilon(p_2,p_3)(||p_2||+||p_3||) = \tag{A.25}$$

$$= \sqrt{(-\alpha\,g^{level+1}+\alpha\,g^{level+2})^2+(r\,g^{level+1}-r\,g^{level+2})^2}(||p_2||+||p_3||) = \tag{A.26}$$

$$= \sqrt{(-\alpha\,g^{level+1}(1-g))^2+(r\,g^{level+1}(1-g))^2}(||p_2||+||p_3||) = \tag{A.27}$$

$$= \sqrt{\alpha^2 + g^{2level+2}(1-g)^2 + r^2 g^{2level+2}(1-g)^2}(||p_2|| + ||p_3||) = \tag{A.28}$$

$$= \sqrt{g^{2level+2}((1-g)^2(\alpha^2 + r^2))}(r g^{level+1} + r g^{level+2}) = \tag{A.29}$$

$$= g^{level+1}(1-g)\sqrt{\alpha^2 + r^2} \times r g^{level+1}(1+g) = g^{2level+2}(1-g)^2\sqrt{\alpha^2 + r^2} \times r \tag{A.30}$$

$$\rho(p_2, p_1) = \varepsilon(p_2, p_1)(||p_2|| + ||p_1||) = \tag{A.31}$$

$$= \sqrt{(-alpha\, g^{level+1} - 0)^2 + (r g^{level+1} - r g^{level})^2}(||p_2|| + ||p_1||) \tag{A.32}$$

$$= \sqrt{(-\alpha\, g^{level+1})^2 + (r g^{level}(g-1))^2}(||p_2|| + ||p_1||) = \tag{A.33}$$

$$= \sqrt{\alpha^2 g^{2level+2} + r^2 g^{2level}(g-1)^2}(||p_2|| + ||p_1||) = \tag{A.34}$$

$$= \sqrt{g^{2level}(\alpha^2 g^2 + r^2(g-1)^2)}(r g^{level+1} + r g^{level}) = \tag{A.35}$$

$$= g^{level}\sqrt{\alpha^2 g^2 + r^2(g-1)^2}(r g^{level})(g+1) = g^{2level}\sqrt{\alpha^2 g^2 + r^2(g-1)^2} \times r(g+1) \tag{A.36}$$

$$\rho(p_2, p_3) + \rho(p_3, dst) < \rho(p_2, p_1) + \rho(p_1, dst) \iff \tag{A.37}$$

$$g^{2level+2}(1-g)^2\sqrt{\alpha^2 + r^2} \times r + g^{2level+4}(g^i - 1)^2\sqrt{\alpha^2 + r^2} \times r < \tag{A.38}$$

$$g^{2level}\sqrt{\alpha^2 g^2 + r^2(g-1)^2}r(g+1) + g^{2level}r(1+g^{2+i})\sqrt{\alpha^2 g^{4+2i} + r^2(1-g^{2+i})^2} \iff \tag{A.39}$$

$$g^2(1-g)^2\sqrt{\alpha^2 + r^2} + g^4(g^i - 1)^2\sqrt{\alpha^2 + r^2} < \tag{A.40}$$

$$\sqrt{\alpha^2 g^2 + r^2(g-1)^2} \times (g+1) + (1-g^{2+i})\sqrt{\alpha^2 g^{4+2i} + r^2(1-g^{2+i})^2} \iff \tag{A.41}$$

$$g^2\sqrt{\alpha^2 + r^2}((1-g)^2 + g^2(g^i - 1)^2) < \tag{A.42}$$

$$\sqrt{\alpha^2 g^2 + r^2 g^2}(g+1) + (1-g^{2+i})\sqrt{\alpha^2 g^{4+2i} + r^2(1-g^{2+i})^2} \iff \tag{A.43}$$

$$g^2\sqrt{\alpha^2 + r^2}((1-g)^2 + g^2(g^i - 1)^2) < \tag{A.44}$$

$$g\sqrt{\alpha^2 + r^2}(g+1) + (1-g^{2+i})\sqrt{\alpha^2 - g^{4+2i} + r^2)1 - g^{2+i})^2} \iff \tag{A.45}$$

$$g^2\sqrt{\alpha^2 + r^2}((1-g)^2 + g^2(g^i - 1)^2) < \tag{A.46}$$

$$g\sqrt{\alpha^2 + r^2}(g+1) + (r+g^{2+i})\sqrt{\alpha^2 - g^{4+2i} + r^2(1 - 2rg^{2+i} + rg^{4+2i})} \iff \tag{A.47}$$

$$g^2\sqrt{\alpha^2 + r^2}((1-g)^2 + g^2(g^i - 1)^2) < \tag{A.48}$$

$$g\sqrt{\alpha^2+r^2}(g+1)+(1+g^{2+i})\sqrt{g^2(\alpha^2-g^{2+2i}+r^2-2rg^i+g2+2i)} \iff \tag{A.49}$$

$$g\sqrt{\alpha^2+r^2}((1-g)^2+g^2(g^{2i}-2g^i+1)) < \tag{A.50}$$

$$\sqrt{\alpha^2+r^2}(g+1)+(1+g^{2+i})\sqrt{\alpha^2+g^{2+2i}(r-1)+r(r-2g^i)} \iff \tag{A.51}$$

$$g\sqrt{\alpha^2+r^2}((1-g)^2+g^2(g^i-1)^2) < \tag{A.52}$$

$$\sqrt{\alpha^2+r^2}(g+1)+(1-g^{2+i})\sqrt{\alpha^2-g^{4+2i}+r^2-2rg^i+rg^{2+2i}} \iff \tag{A.53}$$

$$g\sqrt{\alpha^2+r^2}((1-g)^2+g^2(g^{2i}-2g^i+1)) < \tag{A.54}$$

$$g\sqrt{\alpha^2+r^2}+\sqrt{\alpha^2+r^2}+(1-g^{2+i})\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2} \iff \tag{A.55}$$

$$g\sqrt{\alpha^2+r^2}(1-2g+g^2-g^{2+2i}-2g^{2+i}+g^2) < \tag{A.56}$$

$$g\sqrt{\alpha^2+r^2}+\sqrt{\alpha^2+r^2}+\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2}- \tag{A.57}$$

$$(g^{2+i})\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2} \iff \tag{A.58}$$

$$g\sqrt{\alpha^2+r^2}(1+g(-2+g-g^{2i+1}-2g^{1+i}+g)) < \tag{A.59}$$

$$g\sqrt{\alpha^2+r^2}+\sqrt{\alpha^2+r^2}+\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2}- \tag{A.60}$$

$$(g^{2+i})\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2} \iff \tag{A.61}$$

$$g^2\sqrt{\alpha^2+r^2}(-2+g(1-g^{2i}-2^g i+1)) < \tag{A.62}$$

$$\sqrt{\alpha^2+r^2}+\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2}- \tag{A.63}$$

$$(g^{2+i})\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2} \iff \tag{A.64}$$

$$-2g^2\sqrt{\alpha^2+r^2}+g^3\sqrt{\alpha^2+r^2}(2-g^{2i}-2g^i) < \tag{A.65}$$

$$\sqrt{\alpha^2+r^2}+\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2}- \tag{A.66}$$

$$(g^{2+i})\sqrt{g^i(-g^{4+i}+rg^{2+i}+2r)+\alpha^2+r^2} \iff \tag{A.67}$$

$$(g=\sqrt{r}\to r=g^2) \tag{A.68}$$

$$-2g^2\sqrt{\alpha^2+g^4}+g^3\sqrt{\alpha^2+g^4}(2-g^i(2+g^i)) < \tag{A.69}$$

$$\sqrt{\alpha^2+g^4}+\sqrt{g^i(-g^{4+i}+g^{4+i}+2g^2)+\alpha^2+g^4}- \tag{A.70}$$

$$(g^{2+i})\sqrt{g^i(-g^{4+i}+g^{4+i}+2g^2)+\alpha^2+g^4} \iff \tag{A.71}$$

$$-2g^2\sqrt{\alpha^2+g^4}+g^3\sqrt{\alpha^2+g^4}(2-g^i(2+g^i))< \tag{A.72}$$

$$\sqrt{\alpha^2+g^4}+\sqrt{2g^{2+i}+\alpha^2+g^4}-g^{1+i}\sqrt{2g^{2+i}+\alpha^2+g^4} \iff \tag{A.73}$$

$$-2g^2\sqrt{\alpha^2+g^4}+2g^4\sqrt{\alpha^2+g^4}-g^{4+i}(2+g^i)< \tag{A.74}$$

$$\sqrt{\alpha^2+g^4}+\sqrt{g^2(2g^i+g^2)}-g^{1+i}\sqrt{g^2(2g^i+g^2)} \iff \tag{A.75}$$

$$\sqrt{\alpha^2+g^4}(-2g^3+2g^4)-2g^{4+i}-g^{4+2i}< \tag{A.76}$$

$$\sqrt{\alpha^2+g^4}+g\sqrt{2g^i+g^2}-g^{2+i}\sqrt{2g^i+g^2} \iff \tag{A.77}$$

$$\sqrt{\alpha^2+g^4}(-2g^3)-2g^{4+i}-g^{4+2i}< \tag{A.78}$$

$$\sqrt{\alpha^2+g^4}+g\sqrt{2g^i+g^2}-g^{2+i}\sqrt{2g^i+g^2} \tag{A.79}$$

$$-2g^{4+i}-g^{4+2i}<-g^{2+i}(2gi+g^2) \implies -2g^{4+i}-g^{4+2i}<-g^{2+i}\sqrt{2g^i+g^2} \tag{A.80}$$

$$-2g^{4+i}-g^{4+2i}<-g^{4+i}-2g^{2+2i} \implies \tag{A.81}$$

$$\rho(p_2,p_3)+\rho(p_3,dst)<\rho(p_2,p_1)+\rho(p_1,dst) \tag{A.82}$$

□

A.2 Triangle Inequality of the Metric ρ

In what concerns greedy routing, a metric has to verify the triangle inequality in order to guarantee shortest path routing. However, in our greedy routing scheme the routing choice is not made as in pure greedy routing schemes. We consider the distance between each neighbour and the destination node, but also the distance between the current node and each neighbour, resulting in the following function:

$$min(\ \rho(current,neighbour)+\rho(neighbour,destination)\ ) \tag{A.83}$$

In the above section we verified that this function leads to shortest path routing. Although the metric ρ does not verify the triangle inequality, it does not affect optimality since routing choice is not only based on the value of $\rho(neighbour,destination)$.

Bibliography

[1] Bgp reports. `http://bgp.potaroo.net/`.

[2] Yehuda Afek, Anat Bremler-Barr, and Shemer Schwarz. Improved bgp convergence via ghost flushing. *IEEE Journal on Selected Areas in Communications*, 22(10):1933–1948, 2004.

[3] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[4] James W. Anderson. *Hyperbolic Geometry*. Springer, second edition edition, 2007.

[5] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[6] Marian Boguna, Dmitri Krioukov, and K. C. Claffy. Navigability of complex networks. *Nat Phys*, 5(1):74–80, 2009.

[7] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *DIALM '99: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 48–55, New York, NY, USA, 1999. ACM.

[8] CAIDA. Archipelago measurement infrastructure. http://www.caida.org/projects/ark/.

[9] CAIDA. As relationships. http://www.caida.org/data/active/as-relationships/index.xml.

[10] CAIDA. The caida as relationship dataset, january 2010. http://www.caida.org/data/active/as-relationships/.

[11] CAIDA. Skitter. http://www.caida.org/tools/measurement/skitter/.

[12] Duncan S. Callaway, John E. Hopcroft, Jon M. Kleinberg, M. E. J. Newman, and Steven H. Strogatz. Are randomly grown graphs really random? *Phys. Rev. E*, 64(4):041902, Sep 2001.

[13] CIDR. Cidr report. http://www.cidr-report.org/, July 2005.

[14] D. Clark. The design philosophy of the darpa internet protocols. In *SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 106–114, New York, NY, USA, 1988. ACM.

[15] A. Cvetkovski and M.Crovella. Hyperbolic embedding and routing for dynamic graphs. *INFOCOM*, 2009.

[16] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A decentralized network coordinate system. In *In SIGCOMM*, pages 15–26, 2004.

[17] Amogh Dhamdhere and Constantine Dovrolis. Ten years in the evolution of the internet ecosystem. In *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 183–196, New York, NY, USA, 2008. ACM.

[18] Xenofontas Dimitropoulos. Autonomous system taxonomy. http://www.caida.org/data/active/as_taxonomy/.

[19] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc claffy, and George Riley. As relationships: inference and validation. *SIGCOMM Comput. Commun. Rev.*, 37(1):29–40, 2007.

[20] Xenofontas Dimitropoulos, Dmitri Krioukov, George Riley, and Kc Claffy. Revealing the autonomous system taxonomy: The machine learning approach. In *In Passive and Active Measurement (PAM) Workshop*, 2006.

[21] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Anomalous percolation properties of growing networks. *Phys. Rev. E*, 64(6):066110, Nov 2001.

[22] Ahmed Elmokashfi, Amund Kvalbein, and Constantine Dovrolis. On the scalability of bgp: the roles of topology growth and update rate-limiting. In ACM, editor, *CoNext 2008*. ACM, 2008.

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

[24] Nick Feamster, Hari Balakrishnan, and Jennifer Rexford. Some Foundational Problems in Interdomain Routing. In *3rd ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, San Diego, CA, November 2004.

[25] Anja Feldmann, Luca Cittadini, Wolfgang Mühlbauer, Randy Bush, and Olaf Maennel. Hair: hierarchical architecture for internet routing. In *ReArch '09: Proceedings of the 2009 workshop on Re-architecting the internet*, pages 43–48, New York, NY, USA, 2009. ACM.

[26] David Gelernter. Generative communication in linda. *ACM Trans. Program. Lang. Syst.*, 7(1):80–112, 1985.

[27] P. Brighten Godfrey, Igor Ganichev, Scott Shenker, and Ion Stoica. Pathlet routing. In *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 111–122, New York, NY, USA, 2009. ACM.

[28] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 10(2):232–243, 2002.

[29] IETF Network Research Group. Locator/id separation protocol (lisp. draft-farinacci-list-09.txt, October 2008.

[30] Jacob P. Hoogenboom, Wouter K. Otter den, and Herman L. Offerhaus. Accurate and unbiased estimation of power-law exponents from single-emitter blinking data. *Journal of Chemical Physics*, 125(20):204713–1, 2006.

[31] Cheng Jin, Cheng Jin Qian, and Sugih Jamin. Inet: Internet topology generator, 2000.

[32] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM.

[33] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol (s-bgp). *Selected Areas in Communications, IEEE Journal on*, 18(4):582–592, 2000.

[34] Jon Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.

[35] R. Kleinberg. Geographic routing using hyperbolic space. In *IEEE INFOCOM 2007*, pages 1902–1909. IEEE, May 2007.

[36] P. L. Krapivsky and S. Redner. A statistical physics perspective on web growth. *Computer Networks*, 39(3):261 – 276, 2002.

[37] Dmitri Krioukov, Fragkiskos Papadopoulos, Marián Boguñá, and Amin Vahdat. Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces. *SIGMETRICS Perform. Eval. Rev.*, 37(2):15–17, 2009.

[38] Dmitri V. Krioukov, Fragkiskos Papadopoulos, Marián Boguñá, and Amin Vahdat. Efficient navigation in scale-free networks embedded in hyperbolic metric spaces. *CoRR*, abs/0805.1266, 2008.

[39] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. In *SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 175–187, New York, NY, USA, 2000. ACM.

[40] Anthony Lambert, Marc-Olivier Buob, and Steve Uhlig. Improving internet-wide routing protocols convergence with mrpc timers. In *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 325–336, New York, NY, USA, 2009. ACM.

[41] Priya Mahadevan, Calvin Hubble, Dmitri Krioukov, Bradley Huffaker, and Amin Vahdat. Orbis: rescaling degree correlations to generate annotated internet topologies. *SIGCOMM Comput. Commun. Rev.*, 37(4):325–336, 2007.

[42] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. *SIGCOMM Comput. Commun. Rev.*, 36(4):135–146, 2006.

[43] Zhuoqing Morley Mao, Ramesh Govindan, George Varghese, and Randy H. Katz. Route flap damping exacerbates internet routing convergence. *SIGCOMM Comput. Commun. Rev.*, 32(4):221–233, 2002.

[44] Laurent Mathy and Luigi Iannone. Lisp-dht: towards a dht to map identifiers onto locators. In *CoNEXT '08: Proceedings of the 2008 ACM CoNEXT Conference*, pages 1–6, New York, NY, USA, 2008. ACM.

[45] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. *Proceedings of MASCOTS*, 1, 2001.

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

[47] T. S. Eugene Ng and Hui Zhang. Predicting internet network distance with coordinates-based approaches. In *In INFOCOM*, pages 170–179, 2001.

[48] Ola Nordström and Constantinos Dovrolis. Beware of bgp attacks. *SIGCOMM Comput. Commun. Rev.*, 34(2):1–8, 2004.

[49] I. Norros and H. Reittu. Network models with a 'soft hierarchy': a random graph construction with loglog scalability. *Network, IEEE*, 22(2):40–46, 2008.

[50] University of Oregon. Route views project. http://www.routeviews.org.

[51] Romualdo Pastor-Satorras, Alexei Vázquez, and Alessandro Vespignani. Dynamical and correlation properties of the internet. *Phys. Rev. Lett.*, 87(25):258701, Nov 2001.

[52] Dan Pei, Matt Azuma, Dan Massey, and Lixia Zhang. Bgp-rcn: improving bgp convergence through root cause notification. *Comput. Netw. ISDN Syst.*, 48(2):175–194, 205.

[53] C. Perkins. IP Mobility Support for IPv4.

[54] B. Premore. An experimental analysis of bgp convergence time. In *ICNP '01: Proceedings of the Ninth International Conference on Network Protocols*, page 53, Washington, DC, USA, 2001. IEEE Computer Society.

[55] Derek De Solla Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, pages 292–306, 1976.

[56] Venugopalan Ramasubramanian, Dahlia Malkhi, Fabian Kuhn, Mahesh Balakrishnan, Archit Gupta, and Aditya Akella. On the treeness of internet latency and bandwidth. In *SIGMETRICS '09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, pages 61–72, New York, NY, USA, 2009. ACM.

[57] Internet Routing Registries. `http://www.irr.net`.

[58] Pedro Rodrigues and J. Legatheaux Martins. Improving the accuracy and usefulness of synthetic as-level topology models. In *Actas da ConferÃªncia de Redes de Computadores 2009 (CRC'2009)*. Instituto Superior TÃ©cnico, 10 2009.

[59] Virginie Schrieck, Pierre Francois, Cristel Pelsser, and Olivier Bonaventure. Preventing the unnecessary propagation of bgp withdraws. In *NETWORKING '09: Proceedings of the 8th International IFIP-TC 6 Networking Conference*, pages 495–508, Berlin, Heidelberg, 2009. Springer-Verlag.

[60] Srinivas Shakkottai, Marina Fomenkov, Dmitri Krioukov, Ryan Koga, and kc claffy. Evolution of the internet as-level ecosystem, 2006.

[61] Herbert A. Simon. On a class of skew distribution functions. *Biometrika*, 42:425–440, 1955.

[62] Lakshminarayanan Subramanian, Matthew Caesar, Cheng Tien Ee, Mark Handley, Morley Mao, Scott Shenker, and Ion Stoica. Hlp: a next generation inter-domain routing protocol. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 13–24, New York, NY, USA, 2005. ACM.

[63] Wei Sun, Zhuoqing Mao, and Kang Shin. Differentiated bgp update processing for improved routing convergence. In *ICNP '06: Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*, pages 280–289, Washington, DC, USA, 2006. IEEE Computer Society.

[64] Jeffrey Travers, Stanley Milgram, Jeffrey Travers, and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.

[65] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[66] Xiaowei Yang, David Clark, and Arthur W. Berger. Nira: a new inter-domain routing architecture. *IEEE/ACM Trans. Netw.*, 15(4):775–788, 2007.

[67] Han Zheng, Eng Keong Lua, Marcelo Pias, and Timothy G. Griffin. Internet routing policies and round-trip-times. In *In PAM*, 2005.

[68] Dapeng Zhu, Mark Gritter, and David R. Cheriton. Feedback based routing. *SIGCOMM Comput. Commun. Rev.*, 33(1):71–76, 2003.