Towards More Representative Internet Topologies

Bruno Quoitin
INL - IP Networking Lab
CSE Dept, Université Catholique de Louvain (UCL)
Place St. Barbe 2, 1348 Louvain la neuve - Belgium
bruno.quoitin@info.ucl.ac.be

1. INTRODUCTION

In this report, we discuss the parameters that are important for the generation of synthetic topologies. We focus on the parameters that are significant for the evaluation of interdomain routing protocols and interdomain traffic engineering, though this discussion also largely applies to the evaluation of other protocols or applications. Indeed, the evaluation of applications such as Voice/Video over IP or peer-to-peer (P2P) software critically depends on the properties captured by the topology model. For the Voice/Video over IP protocol, the delay between participants as well as their geographic distribution are relevant. For P2P, the heterogeneity of link capacities has an impact on the performance [36]. In the case of routing protocols and traffic engineering methods, another relevant property of the topology is the path diversity, i.e. the existence of alternative paths between a source and a destination. These paths may have differing properties such as the delay, the available bandwidth or even the node- or edge-disjointness.

The problem of obtaining an accurate picture of the Internet topology is not new. We do not know the shape of the Internet today, especially at the router level. There are multiple reasons for this. First, we do not know about the internal structure of domains since their operators are often reluctant to publish the topology of their network. The internal structure of a domain is important since it constrains the paths that intradomain and interdomain routing protocols will select. Second, we do not know how domains are connected together. There is no map of the Internet available today. Looking at BGP routing tables from a small set of monitoring points [40] provides a gross picture of the interdomain graph. However, this approach misses a large number of edges, mainly of the shared-cost type, that are valuable for interdomain routing protocols. In addition, the interdomain graph that we have today indicates when domains are connected together, but it does not provide information on the link redundancy. Finally, an important characteristic of the interdomain graph is the notion of policies. To the opposite of the intradomain graph, not all paths through the interdomain graph are allowed. These paths are constrained by the policies that are enforced by the domains. Hence, the edges of the interdomain graph must be fitted out with attributes defining the relationship [19] between the interconnected domains.

Today, no topology generator is able to produce router-level topologies of the Internet in a satisfactory way. In this chapter, we propose a methodology to build more realistic router-level topologies and we apply it to the construction of an experimental Internet topology. We first survey in Section 2, the approaches currently being used by researchers to infer or generate Internet-like topologies. Second, we summarize in Section 3 what are the important characteristics of Internet topologies suitable for the evaluation of interdomain routing protocols and traffic engineering techniques. Third, we describe our approach in Section 4. We start by describing how real-world network toppologies are designed. We survey the metrics that can be used to evaluate a network design. Then, we define our methodology for building router-level topologies. In Section 5, we apply this methodology to the construction of an experimental Internet topology. Finally, we conclude in Section 6.

2. RELATED WORK

Several approaches to the generation of Internet router-level topologies suitable for simulation have been proposed in the literature. The first and most natural approach was to rely on existing network topologies. This approach is limited due to the difficulty of obtaining the topology of operational networks today. Most network operators still feel nervous when asked to reveal a precise view of their network topology. There has then been proposals for inferring network topologies at the router-level from measurements. Rocketfuel [38] is the most famous of these techniques and it relies on the result of several traceroutes. Unfortunately, since traceroutes only perform a sampling of the real network topology, these techniques sometimes miss multiple paths between routers [26], [44]. In addition, these techniques sometimes fail to resolve router aliases resulting in links and routers that do not really exist [44].

At the AS-level, techniques such as [40] have been proposed to infer the business relationships between domains from multiple BGP routing table dumps. These techniques also provide an undersampling of the real interdomain topology since BGP routing tables only provide the best routes selected by BGP [2]. These topologies are thus not representative of the actual diversity of the AS-level paths. In [9], Willinger et al discuss the completeness of the inferred topologies based on local views. In addition, inferred AS-level topologies do not provide information on the number of peerings between domains nor information on the internal structure of domains. Other inferrence techniques have been proposed later, such as [4] and [12]. The later shows that using BGP updates more peering links are discovered. However these techniques suffer from the same limitations as studied in [9].

Another approach consists in generating synthetic topologies sharing selected properties with the real Internet. Available generators such as BRITE [29] and GT-ITM [8] produce topologies that respect graph properties seen in the real Internet. GT-ITM for instance allows to build router-level topologies with a backbone/access hierarchy. Nodes are placed randomly on a map and connected using a probabilistic model such as Waxman [47]. The problem of this approach is that topologies are generated in order to mimic pure graph properties of real networks. They fail to capture the optimization process that is also at the basis of the real network topologies.

In [3], Alderson et al have presented a novel approach to the design and generation of realistic Internet topologies which reposes on taking into account the economical and technical driving forces of the Internet. Their idea consists in formulating the network design problem as an optimization problem which takes as input a traffic demand and produces a router/host level topology. Later, in [27], Li et al have propsed new metrics for evaluating generated topologies. They have used their metrics to evaluate various generated topologies and compare them to real networks and Heuristically Optimal Tradeoffs (HOT) networks [14] that have the same node-degree distribution. They concluded that topologies generated without taking into account economical and technical constraints perform poorly. They also predicted that future topology generators should not be built on pure graph-theoretic properties but upon more pragmatic properties such as the maximum throughput that can be achieved by the network and its resilience to failures.

3. MOTIVATIONS AND REQUIREMENTS

In this section, we describe the key characteristics that a model of the Internet topology should capture to be suitable for the evaluation of interdomain routing and traffic engineering. These requirements cover both intradomain and interdomain characteristics of the topology.

- **Realistic intradomain structure**. Each domain in the Internet can be composed of several routers. The structure of the interconnection of these routers is an important characteristic of the Internet topology. There are different reasons to take into account the intradomain structure of domains in a model of the Internet topology. First, the characteristics of the intradomain paths such as *the delay or the bandwidth* are components of the interdomain paths characteristics. The internal structure of a domain must therefore include such characteristics. Second, *the paths available to cross a domain influence the selection of interdomain paths*. With BGP for instance, this interaction is known as hot-potato routing, i.e. BGP will prefer to cross the domain with the lowest IGP cost path. The paths used to traverse a domain will depend on the setting of the IGP weights. We showed in Chapter **??** that in the GEANT network, a large number of routing decisions were based on the IGP cost. Finally, inside a domain there are often *multiple paths* to go from an ingress point to an egress point. This diversity of available paths influences the performance of traffic engineering techniques. The internal structure of domains is thus an important component of a realistic model of the Internet topology.

- **Redundancy of interdomain links**. Another relevant characteristic of the Internet topology is the availability of multiple parallel links between two different domains. Internet domains maintain multiple peering links with their neighbors for different reasons. A first reason is the *resilience* of the peering: if one link fails, the peering is still operational. A second reason is the ability to balance the traffic load accross multiple interdomain links, leading to an *increased bandwidth*. A third reason is the provision of egress points that are closer to a given geographical area. Modeling multiple links between two domains is thus important for the evaluation of interdomain traffic engineering solutions. In current approaches such as BRITE, redundancy can occur accidentaly while it is deliberate in the real world.

- **BGP sessions graph**. The physical Internet topology is overlayed by a graph of BGP sessions. This graph has two important characteristics that matter for the evaluation of interdomain routing and traffic engineering. The first one is the presence of *BGP policies* on eBGP sessions. These policies will constrain the authorized interdomain paths [19]. The second characteristic is the *graph of iBGP sessions* within each domain. The default graph is a clique, but it quickly becomes too large when the number of routers in the domain grows. For this reason, hierarchical iBGP topologies have been introduced. These topologies rely on the utilization of route-reflectors. The introduction of route-reflectors in an iBGP topology change can cause important changes in the selection of interdomain routes. Typically, small domains will use a full-mesh of iBGP sessions while larger domains will organize their iBGP topology around route-reflectors.

- **Geographical location of routers**. The geographical location of routers is another important characteristic of the Internet topology. There is a *strong correlation between the location of the routers and the location of urban and industrial areas* [25]. The Points of Presence (PoPs) of a domain will often be located in such places. The location of routers has an impact on their interconnection, i.e. on the intradomain topology structure. It also has an impact on the propagation delay along the links that interconnect the routers. Another aspect of the geographical spread of routers is the *geographical coverage of Internet domains*. Some domains cover a small region while others, such as large international transit networks, can span multiple countries or continents. A last effect of the geographical location of routers is the *location of the interdomain peering links*. Indeed, two domains will often establish their peering links at places where they both have equipment.

4. ROUTER-LEVEL TOPOLOGIES

In this section, we present a new approach to the design of router-level network topologies. Our approach relies on the use of network design heuristics. The chapter is organized as follows. We will first discuss in Section 4.1 the problem of network design. Basically, real world networks are the outcome of an optimization process but they also follow operational guidelines. We identify the objectives followed by a a real world network designer in practice. Then, we define in Section 4.2 a set of metrics that can be used to evaluate the performance of networks as well as to compare different network designs together. We apply a subset of these metrics on a set of real networks in Section 4.3 to show that real networks sample the spectrum of several design parameters. Finally, we describe our design methodology in Section 4.4.

4.1 Network design

Real world networks are the outcome of a careful design process. The network design problem consists of multiple, sometimes contradictory objectives. No single optimal solution exists, rather a front of possible solutions. The network design problem has been fairly discussed in the literature, in particular by [6, 21]. The objectives of network design may be summarized in **minimizing the**

latency, **dimensioning the links** so that the traffic can be carried without congestion, **adding redundancy** so that rerouting is possible in case of link or router failure and, finally, the network must be designed at **the minimum cost**. None of these objectives are currently explicitly found in degree-based generators such as BRITE [29] or GT-ITM [7].

Usually, a network designer knows the set of nodes that are to be interconnected as well as a prediction of the traffic demand between these nodes. It will then use network design tools such as Cariden MATE [42], Delite [6], WANDL IP/MPLSView [24] or OPNET SPGuru [43] to build a network design that will accomodate the traffic demand. Designing a good network is a time-consuming task though. Indeed, designing an optimal network is computationaly expensive. Its complexity is roughly evaluated to $O(n^5)$ by [21]. This is the reason why network design often relies on heuristics. In addition, the network designer must go through many possible instances of a network design until its objectives are reached and his budget can accomodate it.

Real world networks are often designed with additional constraints in mind. For instance, routers have a maximum degree that corresponds to the maximum number of interfaces they can support [27]. Core routers for instance have an high bandwidth but a limited number of interfaces. In constrast, distribution and access routers can support a larger number of interfaces but their total bandwidth is lower. This leads to a 2- or 3-levels network hierarchy with an increasing aggregation of traffic in the top (core) level. Topology generators such as BRITE or GT-ITM will tend to produce networks with high degree nodes (hubs) in the core. Another pragmatic constraint to the design of a network can be the availability of rack space or power supply in a colocation. In addition to this, network designers apply design guidelines [37], mainly for the design of robust networks. An interesting point to note is that a book such as [37] contains no maths at all. In practice, network design is thus not only an optimization problem.

4.2 Network metrics

In order to measure topologies of real networks as well as to compare topologies generated by network design heuristics, we have selected a set of metrics from the networking literature ([5] is a fair reference). With the following metrics, we capture various aspects of the network design problem. The metrics cover performance properties (delay, redundancy), network design cost and pure graph properties. We use standard graph-theoretic notions. Specifically, let $G(V, E, w)$ be a weighted graph. V is the set of vertices (or nodes) of G and E is the set of edges. The w is the edge weighting function $w : E \rightarrow \mathbb{R}$.

1. **Distance distribution:** A first metric is the distribution of the **distances between pairs of nodes** along the shortest-path route. It is an indication of the delay required to transmit packets between these nodes under the assumption that the largest part of the transmission delay is due to the propagation delay along the links. In particular, the **network diameter** is the length of the longest shortest path. We measure the distribution of the distances between pairs by measuring the length of the shortest-paths between all pairs of nodes.

2. **Path diversity:** To measure the amount of redundancy offered by a network, we use the **path diversity**. This metric measures the availability of diverse paths between pairs of nodes. The availability of diverse paths is important for network robustness and traffic engineering. We compute the path diversity of G in the following way. For each pair of distinct vertices i and j taken in V, we compute the path diversity $\rho(i, j)$ by counting the number of edge-disjoint paths that are available from i to j. First, we compute the shortest-path π_{ij} from i to j using the edges in E. Then we remove the edges of π_{ij} from E and compute another shortest-path. This shortest-path will be edge-disjoint from the first one. We continue until no new path can be found. We repeat this for each pair (i, j). A similar metric was used in [44] to compare the path diversity of the Sprint network and the topologies inferred by Rocketfuel [38].

3. **Connectivity:** Another way to measure the redundancy of a network is to compute the **k-edge-connectivity** [39]. This metric gives the size of a minimum cut in the network. Compared to the path-diversity metric described in the above paragraph, the k-edge-connectivity only gives a lower bound on the number of diverse paths for all the pairs of nodes. To the opposite, the path-diversity metric gives a lower bound for each individual pair.

4. **Node degree:** The distribution of the **node degrees** is a metric which is frequently used to evaluate network topologies. This distribution informs on the existence of hubs, which are nodes with a high degree, where many other nodes connect. It is commonly admitted that networks have a small number of nodes with an high degree (in the backbone) and a large number of nodes with a low degree (access nodes). It has been shown that for some networks, this distribution follows a power law [15]. An interesting fact is that, according to [21], the average node degree of North American carrier networks, is $\bar{d} < 2.5$, while the average node degree of European networks is closer to $\bar{d} > 3.5$. The reason is that in North America, networks span large distances with relatively sparse population in some areas. In addition, North American networks have more often performed economies of scale by using increased capacity links.

5. **Centrality:** The **betweenness-centrality** [5] is a measure of the centrality of vertices or edges in the graph. It basically computes the amount of shortest-paths that go through a vertex or an edge. The centrality of a vertex v is computed as

$$c(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

i.e. the sum for all pairs of sources and destination (s, t) of the fraction of shortest-paths from s to t that pass through v. σ_{st} denotes the number of shortest-paths from s to t and $\sigma_{st}(v)$ denotes the number of shortest-paths from s to t that go through v. A similar definition is used to compute the centrality of edges. Here, the direction of the edges is taken into account.

$$c(u, v) = \sum_{s \neq u, v} \frac{\sigma_{st}(u, v)}{\sigma_{st}}$$

6. **Network cost:** Finally, an important operational constraint is the limited amount of money available to build the network. It is however difficult to define a **network cost** metric. The cost of a network design is difficult to evaluate since it depends on the technology used for links and routers, the bandwidth of links and their length. Though, it is possible to get an idea of some components of the network cost. For instance, the **network span** gives an idea of the total length of the network links.

4.3 Real world networks

We have analysed a set of networks containing large carrier networks as well as regional and national networks. In order to measure the topologies of these networks we have applied the above metrics. Table 1 shows the results of our measurements on a set of 7 network topologies: Abilene, the US research backbone network; GEANT, the pan-european research network; EU-ISP, a european telecom operator; Tiscali; GBLX-EU, the European part of the IP network of Global Crossing; ISP-A and ISP-B, two tier-1 backbone topologies studied in [23].

In Table 1, we show in the two first columns the number of vertices and the number of edges of each topology. Then, the third column ($\bar{\delta}$) shows the average delay between routers. The fourth column ($\bar{\rho}$) gives the average path diversity of each topology. The fifth column ($\bar{d}$) gives the average node degree. The column labelled $\bar{h}$ gives the average number of hops. Note that for EU-ISP, we do not have the distance between the routers, hence we do not give its average delay $\bar{\delta}$ and its total network cost.

	$\|V\|$	$\|E\|$	$\bar{\delta}$	$\bar{\rho}$	$\bar{d}$	$\bar{h}$
Abilene	11	14	25.6	1.63	2.55	2.47
Géant	28	41	21.8	1.62	2.93	3.31
EU-ISP	53	98	NA	1.97	3.69	3.45
Tiscali	39	52	16.7	1.39	2.67	4.89
GBLX-EU	41	77	26.6	1.79	3.76	3.12
ISP-A	20	44	26.9	3.03	4.4	2.25
ISP-B	20	44	28.9	3	4.4	2.25

Table 1: Comparison of network metrics on real world backbone networks.

Based on Table 1, one can assert that real networks sample the spectrum of several metrics. For instance, backbone networks that cover the United States territory have different sizes (number of vertices and edges) and different performance results. Abilene is quite sparse. ISP-A and ISP-B have a high path diversity ($\bar{\rho} \simeq 3$). The average node degree differs largely among all the topologies, ranging from 2.55 for Abilene to 4.4 in ISP-A and ISP-B.

We could not apply to the above networks the metrics related to the network utilization or throughput since we could not obtain the traffic matrices and the link capacities of these networks.

4.4 Topology generation methodology

In this section, we present our methodology to generate synthetic router-level topologies. The literature contains a lot of proposals for building synthetic topologies that have a node-degree distribution similar to real networks [29, 8]. It has been shown that for certain metrics [WHICH ONES ?], the degree-based topology generators better reproduce the hierarchical structure of the Internet than structural generators [41]. However, the choice of metrics presented in [41] has since been criticized [27]. The lack of understanding of the relation between degree-based generated topologies and real networks is an additional obstacle to using degree-based topology generators. Finally, it is possible to generate various different topologies with the same node degree distribution and a very small likelihood to resemble any realistic network [27].

Our methodology shown in Fig. 1 belongs to the class of structural topology generators. Our approach follows the tasks of a network designer. Usually, a network designer has at hand the set of nodes to interconnect as well as an estimation of their traffic demand. It will then use a network design tool in an incremental manner to build a close-to-optimum topology. For the purpose of generating a model of the Internet topology, we have to generate the router-level topology of about 20.000 domains. We can not afford producing an optimal design for each domain. We rather rely on network design heuristics and produce a plausible topology.

Basically, our methodology follows a bottom-up approach. In a first step, the nodes are **grouped into clusters**. These clusters represent the PoPs of the network. A PoP is composed of two types of nodes. The backbone nodes connect to nodes in other PoPs while access nodes only have connections with routers in the same PoP. We thus end up with a two-level hierarchy composed of a backbone graph and access graphs[1]. In a second step, the structure of **each PoP is built**. We rely on operational practice to build realistic PoP structures. Once each PoP has been generated, a topology for the **backbone is produced**. The backbone topology is a graph which interconnects all the backbone nodes of the PoPs. We rely on various various heuristics to generate the backbone. In a final step, **IGP weights and capacities can be assigned** to each link. Different assignment schemes are possible. We describe each step in more details in the following subsections.

4.4.1 *Grouping nodes into PoPs*

A Point of Presence (PoP) is a physical location where a domain has equipment [37]. The location of a PoP is typically a building in a city, a metropolitan area or a zone of industrial activity [25]. A first characteristic of a PoP is that the routers that it contains are often geographically close to each other (some of them are usually in the same room).

To identify the PoPs of the network, we use clustering methods to group nodes into PoPs. The methods we use are based on the geographical distance between the nodes, based on the traffic demand or a combination of both. We rely on K-Medoids [28] to group the nodes into clusters. This method takes a single parameter K which is the number of clusters that we want to obtain. Another way to group nodes into clusters would be to rely on a lattice. Nodes lying in the same cell are grouped to form a PoP. In this case, the parameter of the clustering algorithm is the size of the cell.

4.4.2 *Building the topology of a PoP*

The structure of a PoP is often carefully designed. There is well-known operational practice to build a PoP [37, 22, 20]. A first topological characteristic of a PoP is that it is the place where **traffic is aggregated**. Usually, at the edge of the network, there is a high number of small capacity links connecting to customers and neighboring domains. These links connect to access routers that have a high degree. The access routers are then connected to backbone routers. A second topological characteristic of a PoP is that it is often designed to be **robust to failures**. Typically, to be resilient to a single link failure, an access router will connect to at least 2 backbone routers in the PoP while backbone routers will be densely connected together. In [22], Iannaccone et al discuss the structure of Sprint, a large international transit network and explain that the backbone routers in a PoP are connected to form a clique. Such PoP structure is common as it has been shown in operator forums for other large networks [20].

Our methodology for building the structure of a PoP is inspired from the above operational practice. For each PoP, we select the n most central nodes (geographically speaking) as backbone routers. The backbone nodes of a PoP are densely connected together using

---

[1] Note that in a topology generator such as GT-ITM, the reverse approach is used. The tool produces a first level with PoPs which are further expanded by generating access nodes. In BRITE, both approaches are used. For the bottom-up approach, BRITE uses a clustering technique based on random walk

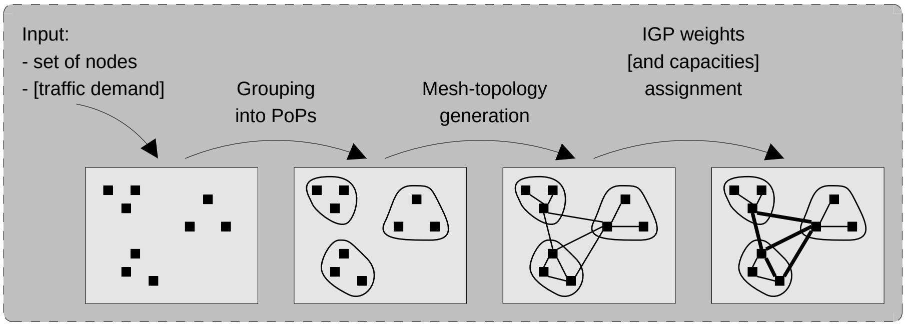


Figure 1: Network design methodology.

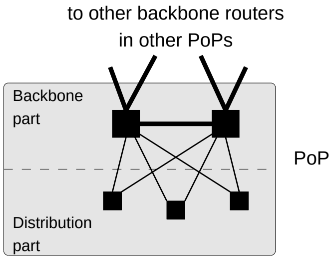


Figure 2: Structure of a PoP ($n = 2$ and $k = 2$).

for instance a tour that guarantees 2-edge-connectivity or a clique. Then, the remaining nodes of the PoP, which model access nodes are connected to the PoP's backbone nodes using at least k edges. Using $k \geq 2$ guarantees redundancy in case of failure ($n \geq 2$ is also needed).

NOTE: if the degree of backbone nodes in the PoP is too high, the methodology could be improved to add an intermediate layer of aggregation.

4.4.3 Building the backbone topology

Once each PoP has been generated, the next step consists in connecting them together. In the real world, the PoPs are usually interconnected with multiple links in the backbone. In [22] for instance, Iannaccone et al indicate that in the Sprint backbone, each PoP is connected to a subset of the other PoPs. A full-mesh would obviously be too expensive. On the other hand, a simple star-topology is not recommended.

In practice, network designers rely on a variety of mesh-generation heuristics [31, 6, 21]. Their operation usually consists in building a seed network topology with built-in requirements such as a maximum number of hop separating each pair of nodes or a minimum connectivity. Then, they proceed iteratively, adding or removing links in order to satisfy additional constraints (path-diversity, link utilization for a given demand prediction). This part is often time-consuming due to the evaluation of many metrics performed at each iteration. At the end, the heuristic leads to a close to optimum mesh design.

In our methodology, we consider only the first step of network design heuristics to generate the backbone. The first advantage is that the seed topology can be computed quickly. The disadvantage is that we do not produce optimal topologies. However, to build a synthetic model of the Internet, we need to generate a large number of topologies. We need not produce optimal designs for each domain.

The first backbone design heuristic we consider is known as **MENTOR** [6] and builds a hybrid minimum spanning tree/shortest-path tree (MST-SPT). The idea behind MENTOR is to find a central node from which to start and use a Dijkstra approach where the labels of nodes are not only the distance from the root, but a linear combination of the distance from the root (start) node and the distance from the previou node. The second component pushes to minimize the total network span. The linear combination is driven by a parameter α which varies between 0 and $+\infty$. With $\alpha = 0$, the heuristic generates a MST while with $\alpha = 1$, the resulting tree is close to an SPT. This heuristic has similarities with the Heuristically Optimized Trade-offs (HOT) proposed by Fabrikant et al [14]. In the HOT approach, nodes arrive uniformly at random in the unit square. The new nodes attach to a previously arrived node based on the same combination of distances than in MENTOR. The difference with MENTOR is that it relies on nodes whose geographical location is known in advance. In MENTOR the starting node is the centroid of the set of vertices.

Since trees are weak networks, another heuristic called **MENTour** [6] can be used. This heuristics directly builds a 2-edge-connected network by computing a minimum length hamiltonian cycle. Since computing a minimum length hamiltonian cycle is an NP complete problem, we rely on a Traveling Salesman Problem (TSP) approximation heuristic to compute the cycle. We use the *furthest-neighbor* heuristic. This heuristic starts with a tour that visits two vertices which are furthest apart. The next vertex to be inserted is the one that increases the length of the current tour the most when this vertex is inserted in the best position on the current tour. The rationale behind the furthest insertion heuristic is that some vertices will be expensive to insert. What are the guarantees in term of tour length ? Higher bound on the length compared to the optimum ? *NOTE: the furthest-insertion heuristic does not guarantee that two edges wil not cross. The only construction heuristic that guarantees this is the sweep heuristic.*

Another way to produce a 2-edge-connected network is the **Two Trees** method [21] which builds 2 MSTs. The TwoTrees method which is due to [31] relies on the combination of two trees to form a network. Good candidate trees are MSTs. The method we have chosen starts with the MST on the complete set of edges E. Then it removes the edges of the first tree from E and searches in the resulting graph a second MST which is thus edge-disjoint with the first MST. The produced graph is the union of both MSTs. Note that other trees may be used to generate such networks [21].

Finally an interesting mesh generation technique consists in computing a **Delaunay triangulation** of the backbone nodes. A Delaunay triangulation is a special type of triangulation graph. It is unique and it is the dual of the Voronoï diagram. The Voronoï diagram is a partition of the space into polygons called sites. Each site contains a single vertex and covers the area of points that are closer to the vertex than to any other vertex. The Delaunay triangulation is a graph that connects two vertices together if their sites in the Voronoï diagram are adjacent. It therefore connects the sites that are close to each other. A Delaunay triangulation can be computed efficiently (typicaly in $O(n.log(n))$ [10]). The Delaunay triangulation also has interesting properties. For instance, it contains the MST. Using a Delaunay triangulation produces a topology with alternate paths between nodes, while minimizing the number of such paths. This is an efficient way of obtaining a cost-effective topology with redundancy.

For clarity reasons, we show in Fig. 3 a visual illustration of how each method performs. The various methods were applied to a randomly generated set of 20 vertices covering Australia.

4.4.4 Assigning IGP weights

After the topology generation step, we assign IGP weights to the links. The IGP weights assignment will influence the selection of the paths used to cross the network. Strangely, this step has received little attention in the topology generators currently in use [29, 7].

In real world networks, different assignment schemes are used. There are two main schemes and a lot of variations. The first common scheme consists in assigning to each link an IGP weight that is proportional to the **link propagation delay** or the **link mileage**. In a least-weight routing scheme, this weights assignment leads to the selection of intradomain paths with the smallest delay. This is the scheme deployed in the Abilene backbone. In Fig. 4 we show a scatterplot of the IGP weights versus the links mileage. There is a clear correlation between the IGP weights and the links mileage in the Abilene network.

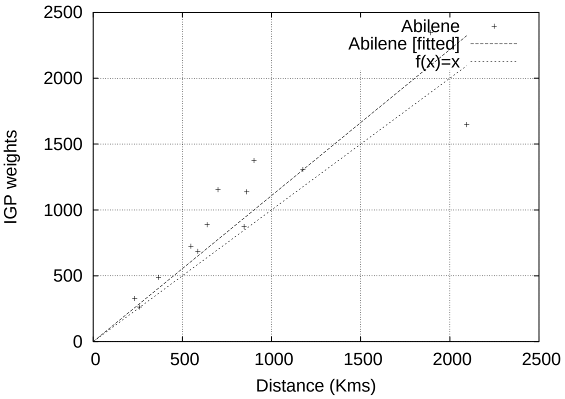


Figure 4: Scatterplot of IGP weights versus links mileage in Abilene.

Another common practice is to assign the IGP weight of a link based on the **inverse of its capacity**. This assignment is also known as the *CISCO default metric* and it is supposed to favour Equal-Cost-Multi-Path (ECMP). It leads to intradomain paths using the highest bandwidth links. In backbone networks, most link capacities are high and similar. In this case, a least-weight routing scheme behaves as minimum-hop routing. In real world networks, the IGP weights deriver from the inverse of the capacity are often manually tuned. This is the case in the GEANT backbone. We show in Fig. 5 the scatterplot of the IGP weights versus the links capacities. We also show the inverse of capacity function ($\frac{10^{12}}{C}$) in order to compare. We observe that the GEANT links have 3 different capacities: 155Mbps, 2.4Gbps and 10Gbps. For most links, the assigned IGP weight is in conformance with the inverse-capacity scheme except for two links.

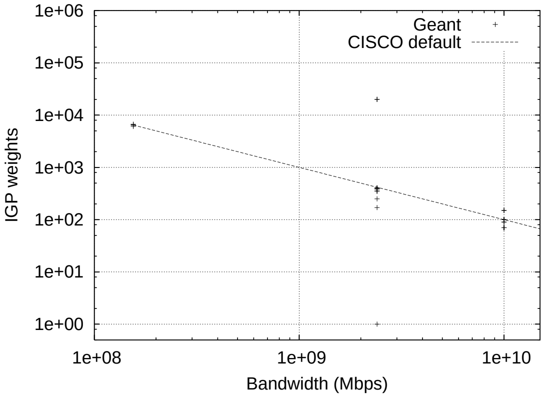


Figure 5: Scatterplot of IGP weights versus link capacity in GEANT.

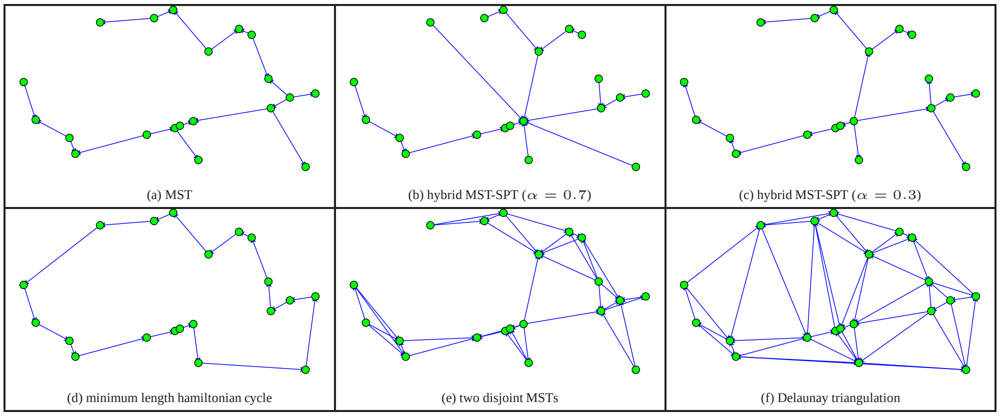
In addition to the above schemes, some operators use smarter assignments such as AT&T's IGP weight optimization method proposed by Fortz et al [18] that minimizes the link utilization. Another possibility is to rely on the method proposed by Nucci et al [34] to optimize IGP weights but without having a negative impact on the IP fast-restoration techniques [22].

In our methodology, IGP weights can be assigned a constant value or they can be based on the link mileage or on the inverse of the links capacities. In the first case, all the links will have the same IGP weight and shortest-path routing will compute minimum-hop paths. The third assignment, based on the inverse of capacities, requires that link capacities be assigned in advance. In the future, we could also rely on Fortz's IGP-WO implemented in the TOTEM toolbox [46]. Using this method would require the links capacities as well as the traffic matrix.

4.4.5 Assigning capacities

In an optional step, capacities can be assigned to the links. The approach of current topology generators consists in assigning links capacities in a random manner. In BRITE [29] for instance, bandwidth can be assigned to links according to uniform, exponential or Pareto distributions. Random assignment of link capacities is interesting since it allows to generate many different assignments. However, it is very unlikely to produce link capacity assignments that are realistic. In real network design, link capacities are **assigned in order to accomodate a traffic demand** between the nodes of the network. In [33] for instance, Norden has evaluated a new QoS routing algorithm. For this purpose, he designed synthetic topologies and relied on a real network design method proposed by Fingerhut et at [17] to compute the bandwidth of the links. The method computes the links capacities that are sufficient to carry a traffic demand that satisfies predefined constraints. The method uses linear programming in order to solve the assignment.

In our methodology, we assigned the link capacities in order to ensure that the demand matrix can be accomodated. For this purpose, we do not rely on linear programming, but we compute the All-Pairs Shortest Paths (APSP) and we simulate the forwarding of traffic demand between all pairs of nodes in order to compute the


(a) MST

(b) hybrid MST-SPT ($\alpha = 0.7$)

(c) hybrid MST-SPT ($\alpha = 0.3$)

(d) minimum length hamiltonian cycle

(e) two disjoint MSTs

(f) Delaunay triangulation

Figure 3: Various mesh designs for a 20-nodes topology located in Australia.

utilization of each link. If there are multiple equal-cost paths, the volume of the demand is splitted equally among the available paths as in [16]. Based on the computed load of each link, the tool selects the suitable capacities.

The utilization of the links is computed based on the computed shortest-paths. The traffic matrix T gives the amount of traffic exchanged between the nodes. In particular, σ_{st} gives the volume sent from the source node t to the destination node t.

$$\Gamma_{u,v} \geq \sum_{(s,t)} \frac{\sigma_{st}(u,v).T_{st}}{\sigma_{st}} \tag{1}$$

where $\Gamma_{u,v}$ is the capacity of link (u,v). The selected capacity must be higher than the computed load of each link. The possible link capacities are selected in a finite set of capacities corresponding to technologies in use today (see Fig. 2).

Technology	**Capacity**
E3	34 Mbps
T3	45 Mbps
STM-1	155 Mbps
STM-4/OC-12	622 Mbps
Gigabit Ethernet	1 Gbps
STM-16/OC-48	2.4 Gbps
STM-64/OC-192	10 Gbps

Table 2: Capacities of links.

In addition, the link capacity can be assigned so as to limit the maximum link utilization to a predefined level τ. This corresponds to real world operational practice. For instance, links are often given a capacity such that the load will be 40 to 50 % [45]. The rationale behind this practice is to keep spare capacity in order to accomodate the variations of the traffic demand as well as its evolution. In the future, we plan to simulate the single-link/router failures and compute the capacity required to accomodate all (or a subset of) the failures. This is what is implemented in commercial network design tools such as WANDL IP/MPLSView [24].

4.5 Comparison of the mesh generation techniques

In order to compare the above mesh generation techniques, we applied them to different sets of vertices. The two first sets are real world networks: Abilene, the US research backbone, and GEANT, the european research backbone. The two second sets are synthetic. They are composed of vertices whose coordinates were generated uniformly at random. The first synthetic network is composed of 20 nodes placed on the Australian continent (Australia-20). The second set is composed of 50 nodes placed in North America (North-America-50).

We generated the following designs for each set of vertices. First, a tour, that we call TSP. Then, two hybrid MST-SPTs with parameters $\alpha = 0.3$ (MENTOR 0.3) and $\alpha = 0.7$ (MENTOR 0.7). The third class of mesh is based on the TwoTrees heuristic (TwoTrees). Finally, we generated a Delaunay triangulation of the vertices (Delaunay). In these topologies, the weight of each link is set according to the link length.

In Fig. 6, we show a measure of the length of the shortest paths in the different generated meshes. On the x-axis, we show the length of the paths. On the y-axis, we show the cumulative fraction of paths that have the corresponding length. A first observation is that for this metric the heuristics can be sorted starting from the best and ending with the worst. The Delaunay triangulation provides the best results, then the TwoTrees, followed by the MST-SPTs and the worst results are provided by the TSP.

For the Abilene and GEANT backbones, we can compare the synthetic meshes with the original networks. The average path-length in the Abilene backbone is 2385.7 kms with a standard deviation of 1296.9 kms. The maximum path lengths is 5029.2 kms. The maximum path length gives an idea of an upper bound on the propagation delay in the network. The Delaunay triangulation provides an average shortest path length equal to 2056.9 kms with a std. dev. of 1077.8 kms. The maximum length is 4126.5 kms. On the other hand, the TSP heuristic leads to an average shortest path length of 3198.8 kms with a std. dev. of 1664.4 kms. The maximum length is 5811.4 kms. If short delays are an objective of the design, the TSP heuristics should be avoided. When the network contains a larger number of nodes, the results of the TSP are going further apart from the other heuristics (see North-America-50).

In GEANT, the average path length is 2166.6 kms with a standard deviation of 1733.3 kms. The larger deviation can be explained by the fact that the GEANT backbone contains two links that cross the atlantic ocean. BLABLABLA

In Fig. 7, we show a measure of the path-diversity in the different generated meshes. The lowest path diversity is obtained with the MST-SPT heuristic since it builds trees. The offered path-diversity in this case is limited to 1. Then, the TSP offers a path-diversity of 2 with one path clockwise and another counterclockwise. The

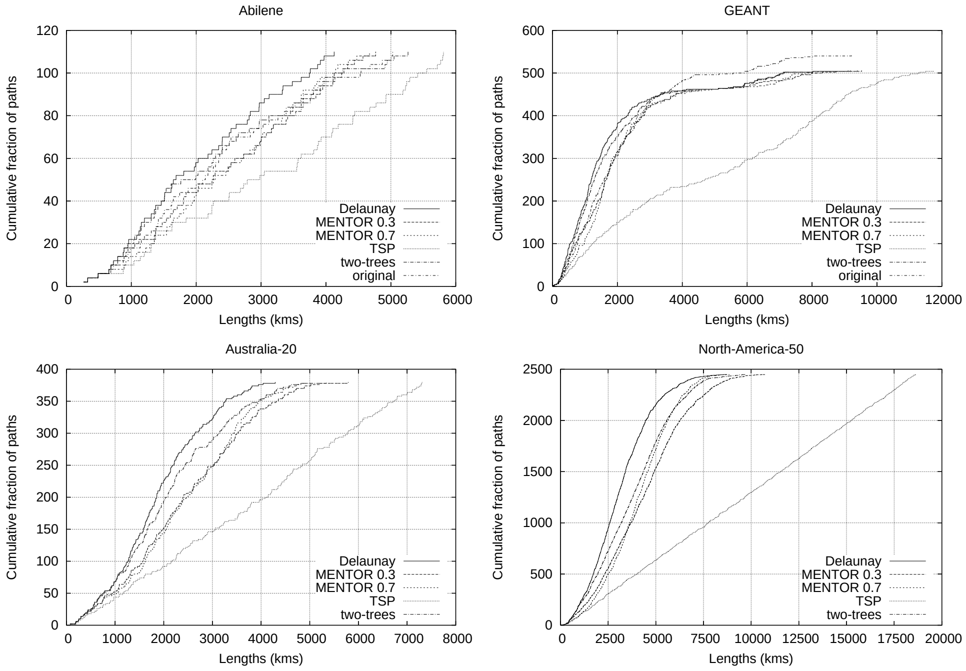


Figure 6: Length of shortest-paths in synthetic meshes.

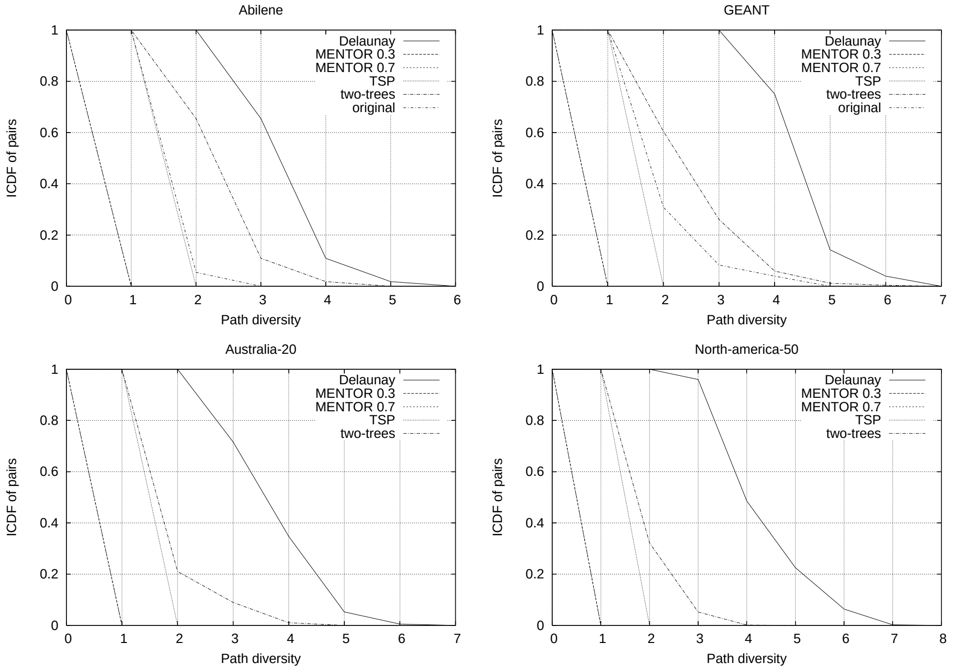


Figure 7: Path diversity in synthetic meshes.

MENTOR heuristic provides a path-diversity

The evaluation performed in this section helps to select which mesh generation heuristic is suitable for what objective. For instance, if we want a minimum cost network with good delay characteristics and if we do not care about the resilience, the MST-SPT approach is suitable. If we need resilience and do not care about the performance, a tour can be used. If resilience and performance matter, the TwoTrees approach can be used since it provides short paths between any pair of vertices. In addition, it is 2-edge-connected since the minimum edge-cut in a TwoTree-generated mesh contains 2 edges. Such a network is resilient to a single link failure. Finally, if path diversity is required, the Delaunay triangulation provides performance and a high path diversity. If the network cost matter, the TwoTrees approach is preferable over the Delaunay triangulation since the latter contains a larger number of edges.

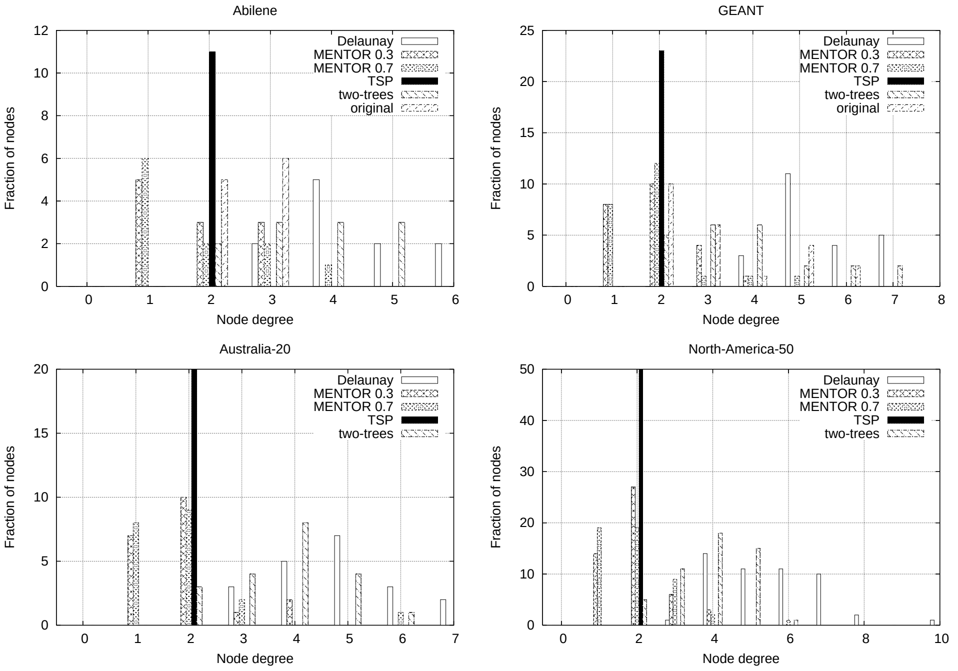


Figure 8: Node degree in synthetic meshes.

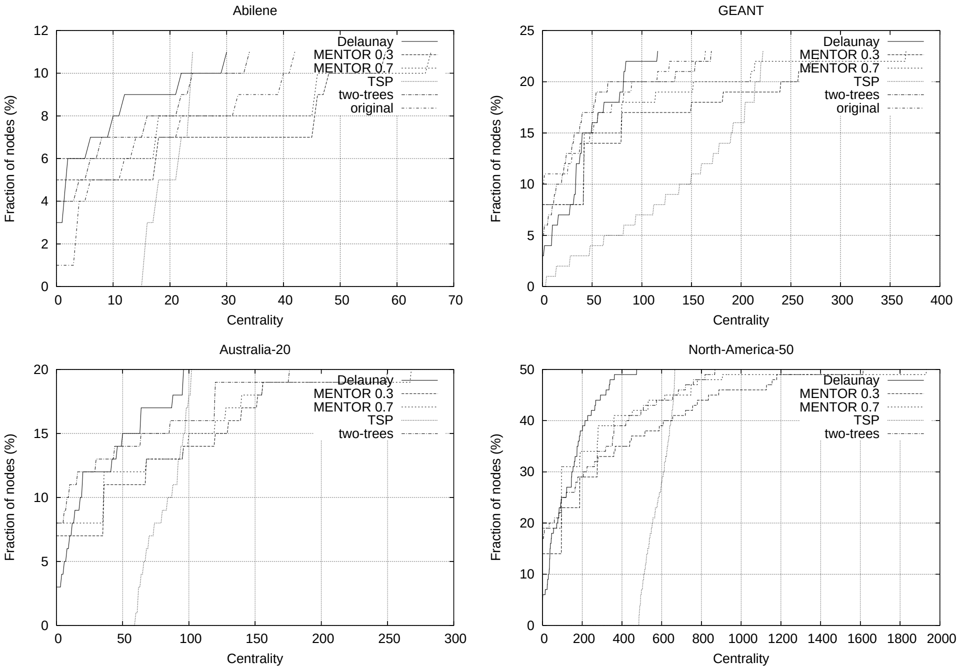


Figure 9: Centrality of vertices in synthetic meshes.

5. AN EXPERIMENTAL INTERNET TOPOLOGY

In this section, we describe an experimental router-level Internet topology designed based on the methodology introduced in Section 4. This topology is not a model of the real Internet topology. This model can not be used for the purpose of reproducing how interdomain routing and interdomain traffic engineering behave in the real Internet.It is rather a model capturing some aspects of the Internet topology such as the geographical spread of routers, the redundancy of peering links, peering policies and router-level intradomain topologies. It can be used to study how these characteristics impact routing and traffic engineering techniques.

The section is organized as follows. We first describe the datasets that we used to build the topology in Section **??**. Then, we describe our methodology in Section **??**.

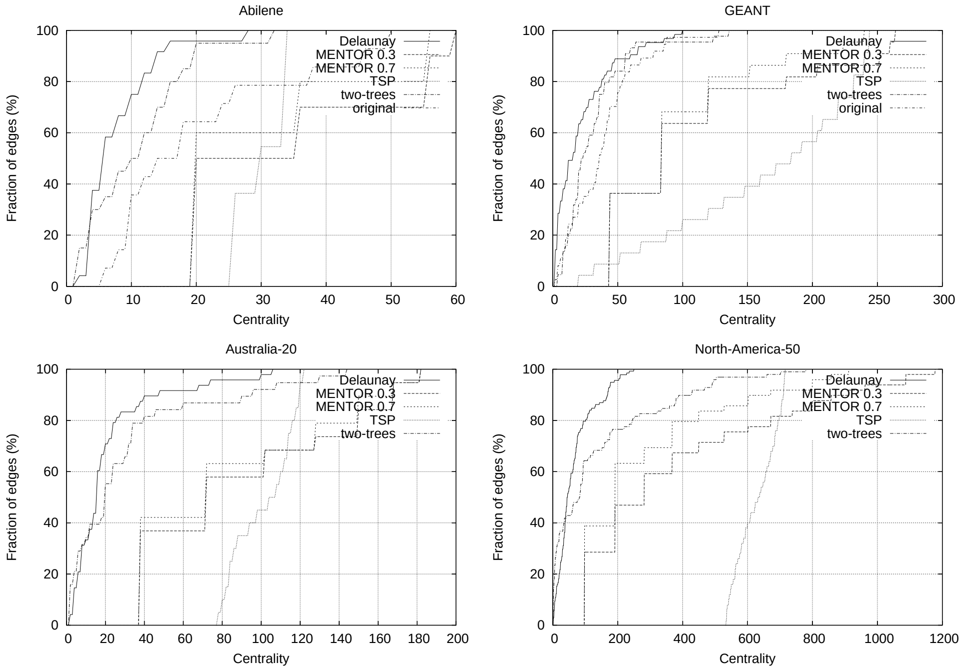


Figure 10: Centrality of edges in synthetic meshes.

5.1 Geographic location of routers

The first step towards the construction of our router-level Internet topology consists in positionnning the routers on the earth. A characterization of the geographic location of routers was performed by Lakhina et al [25]. However, no model exists. BRITE [29] provides a mean to place routers at random on a square map using a uniform or a pareto distributions. However, it is unsure whether or not the resulting geographical distribution has similarities with the real world.

We rely on a dataset providing a **mapping between ranges of IP addresses and geographic coordinates** (latitude and longitude). This dataset was obtained by a geolocation technique [35]. Such technique relies on a mix of whois databases requests (see also [32]), hostname based mapping and various other heuristics to determine with more or less accuracy the location of an Internet host. According to [35], a single technique alone is not sufficient, but the combination of whois, hostname and ad-hoc information works quite well. The potential uses of geographic mapping include content customization, targeted advertising and struggle against credit card fraud. A lot of commercial companies are selling such databases today. Among them, we opted for MaxMind [1] which claims for an high accuracy [2]. The database we use is dated from the 6th of June, 2004 and it contains 1873457 ranges of IP addresses in 118489 geographical locations.

With this dataset, we do not obtain the location of routers but the location of Internet hosts. However, we rely on the fact that blocks of IP addresses represent networks that are connected to a PoP of an ISP. Groups of IP addresses that belong to the same domain and that are located at the same place constitute a PoP for this domain.

5.2 Geographic coverage of domains

We need to identify for each block of IP addresses by which domain it is owned. We rely on a BGP routing table dump for this purpose. This routing table provides us with a mapping between IP prefixes and AS-Paths. We take the last AS in the AS-Path to retrieve the **origin AS of an IP prefix**[3]. With this information, we are able to find the origin AS of prefixes that contain the IP addresses ranges found in the geolocation dataset. The BGP routing table dump we use was collected by the RouteViews project [30] and is dated from the 5th of June, 2004 and contains 139527 different prefixes originated by WWW different ASes.

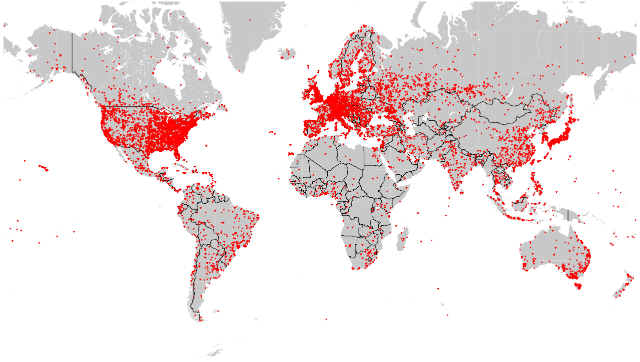


Figure 11: Plot of the points-of-presence on a world map.

NOTE: say how many origin AS were found + say how many locations were found in each AS (median, p-5 and p-95 or distribution)

5.3 Domain topologies

Based on the geolocation dataset and on a BGP routing table

---

[2] In addition to affordable prices for education

[3] In the case of aggregated routes with an AS-Path that does not end with an AS_SEQUENCE segment, we are not able to find the origin of the route. This concerns only XXX percent of the routing table

dump, we are able to find for each domain a set of N PoPs. We know the geographical location of each PoP. We apply the methodology described in Section 4 to generate a router-level topology for each domain.

We use the following parameters. For each domain, we group its N locations in $K = \frac{N}{S}$ clusters which leads to clusters with an average size of S. We arbitrarily choose to use $S = 5$, but other values are possible. Then, we generated the structure of each PoP using the methodology of Section 4.4.3. The parameters we use are $n = 2$ backbone routers in each PoP and a redundancy factor $k = 2$. This ensures a robust PoP design. To connect the PoPs together, we choose the Delaunay triangulation since it provides a performant, redundant network at a reasonable cost. The other mesh generation heuristics described in Section 4.4.3 can also be used. Then, we assign IGP weights based on the links mileage. Finally, we assign capacities. Since we do not have a traffic demand, we choose to assign backbone links an high capacity Γ_h and access links a lower capacity Γ_l. We arbitrarily choose 10Gbps and 155Mbps as values for Γ_h and Γ_l respectively.

5.4 Interdomain peerings

In order to connect the domains together, we need to know for each pair of domains if they have a peering together. In this case, we also need to know if it is a customer-provider or peer-to-peer relationship. We use an AS-level topology inferred by Subramanian et al [40]. This dataset contains the interdomain relationships that exist between Internet domains. We are aware of the limitations of such dataset [12]. In particular the utilization of a small set of vantage points located mostly in the core of the Internet misses a large number of shared-cost and backup peerings. Dimitropoulos et al have recently proposed a new approach to the generation of synthetic AS-level topologies with embedded policies [13]. Another possibility would be to rely on an AS-level topology generator that produces an hierachy of domains such as GHITLE [11].

NOTE: Such data can not be produced by BRITE. GT-ITM2 [8] is supposed to produce such policies but it is still not available publicly.

In addition, each time two domains have a peering relationship, we need to decide how many peering links they have between each other and where these links are connected. We decided to determine the number of interdomain links N_{ij} based on the sizes of the domains, N_i and N_j. We use the following formula

$$N_{ij} = 1 + (N-1).\left\lfloor \frac{N_i.N_j}{(\max_i N_i)^2} \right\rfloor \qquad (2)$$

This formula guarantees that $1 \leq N_{ij} \leq N$. We choose arbitrarily $N =$.

To place these links, we rely on the assumption that two domains will preferably connect at places where both are present. Our algorithm to select the endpoints of the N interdomain links is as follows. We start with N links to select. In the first iteration, we search among the $N_i \times N_j$ pairs of routers the shortest link (u_1, v_1). Then, we remove from the set of possible endpoints the vertices u_1 and v_1. We start a second iteration, and so on until N links are placed.

NOTE: the above algorithm has limitations. First, the order in which links are selected has an impact on the solution found. It might be better to select a slightly longer link at first to reach a better solution. I wonder if this is a well-known combinatorial problem. Second, the algorithm does not take the geographical spread of the interconnections into account. This is probably one of the objectives of domains.

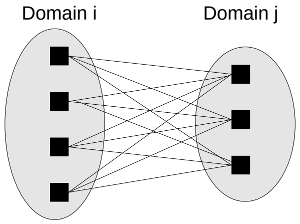


Figure 12: Interconnection of two domains.

5.4.1 Internal BGP setup

In addition to the model of the physical or layer-3 topology, the evaluation of interdomain routing protocols also requires a model of the logical topology. The physical model represents the physical (or layer-3) links that exist between routers. In contrast, the logical topology defines how the routers talk to each other and exchange routing information. The logical topology can be detailed in the logical topology used inside domains and the logical topology used between domains. Typicaly, the internal logical topology models the iBGP sessions between routers while the external topology models the eBGP sessions. The internal and external logical topologies follow different design rules.

The logical topology inside a domain strongly differs from the physical (or layer-3) topology. There are two typical configurations of BGP within a domain. The first one is a full-mesh of iBGP sessions, meaning that each router has an iBGP session with all the other routers. Fig. 13 shows an example of a simple network topology composed of three PoPs on the left and the internal logical topology for the same network if a full-mesh of iBGP sessions is used. The iBGP full-mesh can be modeled as a clique of undirected edges.

The second common logical topology within a domain is an iBGP hierarchy. In such a topology, not all routers and not all iBGP sessions are equal. There are special routers named route-reflectors and special iBGP sessions which are sessions to client routers. The special client iBGP sessions are directed. The route-reflectors are allowed to propagate over a client iBGP session routes received over another iBGP session. A variation of this topology is sometimes deployed in real networks. In this variation, the client routers are connected together in a full-mesh of iBGP sessions. Both topologies are illustrated in Fig. 14.

The external part of the logical topology follows the links of the physical topology. For each link between two domains in the physical topology, there will be an edge in the logical topology, representing an eBGP session. The difference compared to the internal logical topology is that policies are enforced on the external links.

The policies enforced on the external logical links depend on the economical relationships between the domains. These relationships can be grossly categorized in two types [19]: provider-customer and peer-to-peer relationships. The first type of relationship is used between a provider and a customer domains. The customer buys connectivity from its provider and receives from this provider routes towards all the Internet destinations. In addition, the providers advertises to the whole Internet the networks of the customer. The second type of relationship occurs between two domains that have a private peering and that share its cost. Both domains will usually only advertise the routes towards their own clients on this peering in order to avoid to provide transit for Internet routes to the peer domain. In addition, the domains having a peer-to-peer relationship will not advertise to their provider

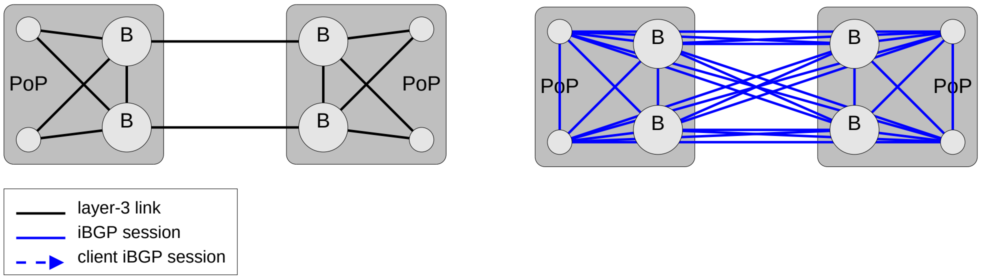


Figure 13: Flat iBGP topology.

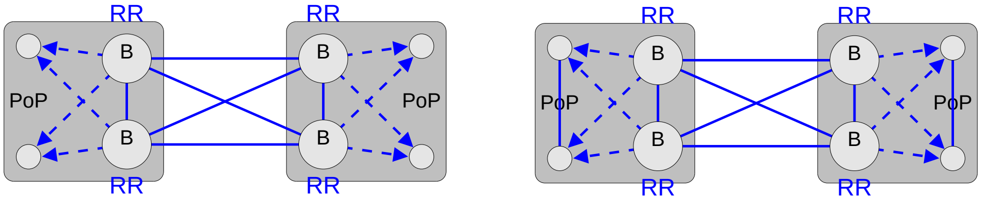


Figure 14: Hierarchical iBGP topologies.

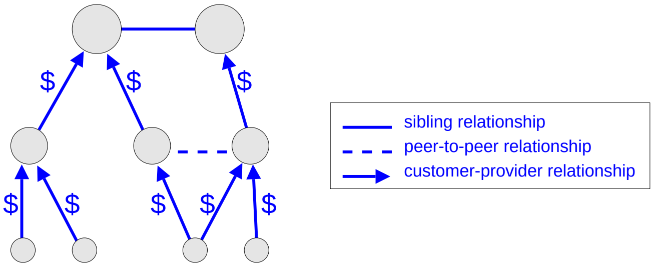


Figure 15: Interdomain relationships.

The above relationships can be implemented using policies on the eBGP sessions. Each type of relationship defines a different policy.

In addition to the policies allowing or preventing transit, Internet domains also use preference settings. These preferences define a political ranking on the routes. A typical setting is to prefere the routes learned from a customer over the routes received from a peer or a provider. The rational behind this preference is that using the route received by a customer does not cost money. For the same reason, the routes received from a peer are also prefered over the routes received from a provider. This defines a local ranking of routes based on the types of relationships: customer routes > peer routes > provider routes.

NOTE: the model of interdomain relationships from Gao et al is coarse. (1) It is not unusal to have domains that maintain different relationships over different peering links. (2) In addition, other policies are enforced. For instance, research backbone networks such as Abilene only redistribute routes towards research and education networks. Other local policies are also possible (to provide backup routes for instance). (3) Finally, policies related to traffic engineering are also enforced.

6. CONCLUSION

In this chapter, we have described a pragmatic methodology to build router-level Internet topologies. In the first part of the chapter, we focused on the generation of a single domain topology. We presented an original methodology which, in contrast with traditional topology generators, relies on network design heuristics instead of probabilistic models. We described 4 different network design heuristics and we compared them based on real world networks as well as synthetic sets of vertices. We used different metrics to examine the performance obtained with each heuristic. To cite two heuritics, we concluded that the hybrid MST-SPT heuristic is suitable to build low-cost and performant networks without robustness. On the other hand, a Delaunay triangulation produces a performant network that is resilient to the failure of links and nodes. In addition, we present solutions to the assignment of IGP weights and capacities to the network links.

In the second part, we applied our router-level topology generator to the construction of an Internet-like toplogy. This topology captures different aspects of the real world Internet topology such as the redundancy of peering links, peering policies, geographic spread of routers and router-level domain topologies. We described how to build this topology from 3 datasets obtained by measurement in the real Internet. This topology is more detailled than topologies obtained from probabilistic models. In addition, it is possible to vary its characteristics and study their impact on the performance of interdomain routing and traffic engineering.

NOTE: - talk about links croissing ocean (Atlantic = pond/millpond) - talk about operational practice for iBGP hierarchy: if network over multiple continents, use top-level RRs on both sides of the ocean instead of having all major locations on the top-level mesh.

For the modeling, this may mean that an additional level of RRs is required. See RIPE-49 BGP network design...

7. REFERENCES

[1] MaxMind GeoIP City Database. `http://www.maxmind.com/app/city`, June 2004.

[2] A. Akella, J. Pang, and A. Shaikh. A Comparison of Overlay Routing and Multihoming Route Control. In *Proceedings of ACM SIGCOMM*, Portland, Oregon, August 2004.

[3] D. Alderson, J. Doyle, R. Govindan, and W. Willinger. Toward an Optimization-Driven Framework for Designing and Generating Realistic Internet Topologies. *ACM SIGCOMM Computer Communications Review*, 33(1), January 2003.

[4] G. D. Battista, M. Patrignani, and M. Pizzonia. Computing the Types of the Relationships between Autonomous Systems. In *Proceedings of IEEE INFOCOM*, March 2003.

[5] U. Brandes and T. Erlebach. *Network Analysis: Methodological Foundations, LNCS3418*. Springer-Verlag, 2004.

[6] R. S. Cahn. *Wide Area Network Design: Concepts and Tools for Optimisation*. Morgan Kaufmann, 1998.

[7] K. Calvert, M. Doar, and E. Zegura. Modeling Internet Topology. *IEEE Transactions on Communications*, pages 160–163, December 1997.

[8] K. Calvert, J. Eagan, S. Merugu, A. Namjoshi, J. Stasko, and E. Zegura. In *Proceedings of ACM SIGCOMM Workshop on Models, methods and tools for reproducible network research*, pages 23–27, August 2003.

[9] H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Towards capturing representative AS-level Internet topologies. *Computer Networks*, 44(6):737–755, April 2004.

[10] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarkopf. *Computational Geometry: Algorithms and Applications (2nd edition)*. Springer-Verlag, 2000.

[11] C. de Launois. Generator of Hierarchical Internet Topologies using LEvels. `http://openresources.info.ucl.ac.be/ghitle/`, 2001.

[12] X. Dimitropoulos, D. Krioukov, and G. Riley. Revisiting Internet AS-level Topology Discovery. In *Proceedings of the 6^{th} Passive and Active Measurement Workshop (PAM)*, 2005.

[13] X. Dimitropoulos, G. Riley, D. Krioukov, and R. Sundaram. Towards a Topology Generator Modeling AS Relationships. In *Proceedings of IEEE ICNP*, 2005.

[14] A. Fabrikant, E. Koutsoupias, and C. H. Papadimitriou. Heuristically Optimized Trade-offs: A New Paradigm for Power Laws in the Internet. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, pages 110–122. Springer-Verlag, LNCS2380, 2002.

[15] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *Proceedings of ACM SIGCOMM 1999*, August 1999.

[16] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford. NetScope: Traffic Engineering for IP Networks. *IEEE Network Magazine*, March 2000.

[17] J. A. Fingerhut, S. Suri, and J. S. Turner. Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2):287–309, 1997.

[18] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of IEEE INFOCOM*, pages 519–528, March 2000.

[19] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE Global Internet*, November 2000.

[20] V. Gill and J. Mitchell. AOL Backbone OSPF-ISIS Migration. In *NANOG29*, October 2003.

[21] W. D. Grover. *Mesh-Based Survivable Networks*. Prentice Hall PTR, 2004.

[22] G. Iannaccone, C.-N. Chuah, S. Bhattacharyya, and C. Diot. Feasibility of IP Restoration in a Tier-1 Backbone. *IEEE Network Magazine*, 18(2):13–19, March 2004.

[23] R. Keralapura, C. N. Chuah, G. Iannaccone, and S. Bhattacharyya. Service Availability: A New Approach to Characterize IP Backbone Topologies. In *Proceedings of IEEE IWQoS*, 2004.

[24] W. A. N. D. Laboratory. IP/MPLSView. `http://www.wandl.com`, 2005.

[25] A. Lakhina, J. Byers, M. Crovella, and I. Matta. On the Geographic Location of Internet Resources. *IEEE Journal on Selected Areas in Communications*, 21(6):934–948, August 2003.

[26] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie. Sampling Biases in IP Topology Measurements. In *Proceedings of IEEE INFOCOM*, April 2003.

[27] L. Li, D. Alderson, W. Willinger, and J. Doyle. A First Principles Approach to Understanding the Internet's Router-level Topology. In *Proceedings of ACM SIGCOMM*, August 2004.

[28] J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of 5^{th} Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[29] A. Medina, A.Lakhina, I. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation. In *MASCOTS 2001*, August 2001.

[30] D. Meyer. Route Views Archive Project (University of Oregon). `http://archive.routeviews.org/`, 2005.

[31] C. L. Monma and D. F. Shallcross. Methods for Designing Communication Networks with Certain Two-Connected Survivability Constraints. *Operations Research*, 37(4):531–541, July 1989.

[32] D. Moore, R. Periakaruppan, J. Donohoe, and K. Claffy. Where in the world is netgeo.caida.org? In *Proceedings of INET'2000*, 2000.

[33] S. Norden. Inter-domain routing: Algorithms for QoS guarantees. *Computer Networks*, 2005.

[34] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot. IGP Link Weight Assignment for Transient Link Failures. In *Proceedings of ITC-18*, September 2003.

[35] V. N. Padmanabhan and L. Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proceedings of ACM SIGCOMM*, August 2001.

[36] F. L. Piccolo, G. Neglia, and G. Bianchi. The Effect of Heterogeneous Link Capacities in Bittorrent-like File Sharing Systems. In *Proceedings of HotP2P*, October 2004.

[37] B. Raveendran and P. Smith. *Cisco ISP Essentials*. Cisco Press, 2002.

[38] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP

Topologies with Rocketfuel. In *Proceedings of ACM SIGCOMM*, August 2002.

[39] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM (JACM)*, 44(4):585–591, July 1997.

[40] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. In *Proceedings of INFOCOM*, June 2002.

[41] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network Topology Generators: Degree-Based vs. Structural. In *Proceedings of ACM SIGCOMM*, August 2002.

[42] C. Technologies. MATE. `http://www.cariden.com/`, 2005.

[43] O. Technologies. SPGuru. `http://www.opnet.com/`, 2005.

[44] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker. Characterizing and Measuring Path Diversity of Internet Topologies. In *Proceedings of ACM SIGMETRICS*, June 2003.

[45] T. Telkamp. Traffic Characteristics and Network Planning. In *NANOG26*, October 2002.

[46] TOTEM. A TOolbox for Traffic Engineering Methods. `http://totem.info.ucl.ac.be`, February 2005.

[47] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988.