IGen: Generation of Router-level Internet Topologies through Network Design Heuristics

Bruno Quoitin, Virginie Van den Schrieck, Pierre François and Olivier Bonaventure
Université catholique de Louvain (UCLouvain)
Place St. Barbe 2, 1348 Louvain-la-Neuve, Belgium
Email : `firstname.lastname@uclouvain.be`

Abstract— **Network researchers often need to evaluate newly proposed algorithms and protocols over realistic topologies. Many researchers use random networks for such evaluations. Unfortunately, random networks do not accurately model real networks and important parameters such as link metrics or iBGP configurations are often ignored. We propose IGen, a freely available topology generator that uses network design heuristics to generate realistic IP network topologies. We explain the objectives of IGen and describe the topology generation process in details.**

I. INTRODUCTION

It is being increasingly important for the network community to take into consideration network design objectives for the generation of network topologies. The evaluation of network protocols and applications, be it through simulation or over a testbed, requires a careful selection of the underlying network topology. Not all protocols are sensitive to the same topology parameters. Voice/Video over IP protocols for example will be sensitive to the delay between participants as well as to their geographic distribution while for peer-to-peer protocols (P2P), the heterogeneity of link capacities has more impact on the performance [34]. In the case of routing protocols and traffic engineering methods, another relevant property of the topology is its path diversity, i.e. the existence of alternative paths with different properties between a source and a destination.

The problem of obtaining an accurate picture of the Internet topology is not new. We do not know the exact shape of the Internet today, especially at the router level. There are multiple reasons for this. First, the internal structure of domains is very rarely disclosed, operators being reluctant to publish the detailed topology of their network. Attempts at discovering the Internet topology through traceroutes [37], [36] miss several links and suffer from aliasing issues. Second, we only have a partial view of the interconnection of domains. Obtaining it by looking at BGP routing tables from a small set of monitoring points [38] provides a gross picture of the Internet graph. However, a large number of edges, especially private peering links cannot be discovered through this method. In addition, the interdomain graphs that we have do not tell how many links actually connect two domains together. Finally, despite several attempts [38], [14], inferring the routing policies or the business relationships [19] between domains has remained an open problem. This is however of crucial importance when interdomain paths need to be computed.

On the other hand, today, no topology generator is able to produce router-level topologies of the Internet in a satisfactory way. Pure degree-based topology generators such as BRITE [27] and GT-ITM [7] seldom produce realistic topologies as they do not take into account network design objectives and constraints [25]. Objectives such as minimizing the latency, dimensioning the links, adding redundancy or minimizing the network budget are not taken into account by these generators. Moreover, the constraints due to the fixed geographical location of PoPs are ignored. Finally, to our knowledge, the assignment of administrative costs to links, as required by intradomain routing protocols such as OSPF or IS-IS, has never been considered by topology generators. The same is missing at the interdomain level, where routing policies related to the business relationships should be assigned to interdomain links to contrain the selection of paths by BGP.

In this paper, we propose a methodology to build more realistic router-level topologies by relying on network design heuristics. Our methodology addresses the problem of network provisioning, and the assignment to links of administrative costs and routing policies. Our methodology has been implemented in IGen[1], an open-source topology generation toolbox. Through the selection of heuristics and parameters, the user can direct IGen towards certain design goals, such as optimizing the network for delay or throughput. IGen will generate router-level network topologies accordingly. Generated topologies can be arbitrarily scaled in terms of the number of nodes and the size of PoPs.

The paper is organised as follows. We first explain in Section II the requirements of Internet topologies suitable for the evaluation of network applications and protocols. Second, we describe our methodology for generating topologies in two parts. Section III focuses on the generation of the internals of a single domain while Section IV explains how to connect domains together to form an Internet-like topology. We survey in Section V, the approaches currently being used by researchers to infer or generate Internet-like topologies. Finally, we conclude in Section VI.

---

[1] Freely available from `http://inl.info.ucl.ac.be/softwares/igen`

II. MOTIVATIONS AND REQUIREMENTS

In this section, we describe the key characteristics that we believe a model of the Internet topology should capture to be suitable for the evaluation of interdomain routing and traffic engineering. These requirements cover both the intradomain and the interdomain characteristics of the topology. We list them here and explain them in details in the subsequent sections.

- **Network design:** A network topology is the outcome of a careful network design process combined with the application of design guidelines.
- **Geographical location:** Network design choices are constrained by the geographical location of nodes to interconnect.
- **Network provisioning:** The network capacity is provisioned according to a traffic demand.
- **Selection of paths:** Networks are configured so as to prefer some links or paths over others. This selection depends on administrative costs for intradomain paths and on routing policies for interdomain paths. The organization of BGP sessions inside a network can also impact the path selection.

A. 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 well 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**.

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 [9], Delite [6], WANDL IP/MPLSView [43] or OPNET SPGuru [32] 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 designers often rely on heuristics. In addition, the network designers will often need to produce several instances of a network design before their objectives are reached and their 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 [25]. 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. 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 [35], mainly for the design of robust networks. An interesting point to note is that a book such as [35] contains no “maths” at all. In practice, network design is thus more than a pure mathematical optimization problem.

B. Geographical location of routers

Communication networks are built for the purpose of interconnecting computers and people that have a fixed geographical location (at least for wired networks). The Points of Presence (PoPs) of a network, i.e. the places where networking equipment has been deployed, will often be located in areas with significant human or industrial activity. A study by Lakhina et al. [23] has shown that there is a *strong correlation between the location of routers and the location of urban and industrial areas*. This geographical constraint has an important impact on the design of networks. Given the state of communications technology, the distance between two points defines a lower bound on the propagation delay between these points. Whatever the network designer does, this delay cannot be shortened. The designer only has freedom in the selection of how the geographical locations can be interconnected given a financial budget.

The geographical location of networks has also an impact on the interconnection of networks. Networks administered by a single authority are called domains or Autonomous Systems (ASs). Domains need to be connected together to allow global connectivity. When it comes to interconnecting two domains together, the location of their respective PoPs has an impact on where the interdomain links can be deployed. If possible, a domain will often prefer to peer with a neighbor that has local connectivity (often in the same or nearby PoP) rather than relying on the deployment or acquisition of additional long-distance links.

C. Network provisioning

Surprisingly, the assignment of capacities to links has received little attention in the topology generators currently in use [27], [7]. Yet these are of tremendous importance for network protocols. We are not the first to pinpoint the need for taking these parameters into consideration. [25] for example proposed to measure the total network throughput based on the links and routers capacities.

The link capacities will place a bound on the amount of traffic a network can accomodate. However, most topology generators assign these capacities in a random manner. In BRITE [27] 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. An example of a more realistic link capacities assignment was

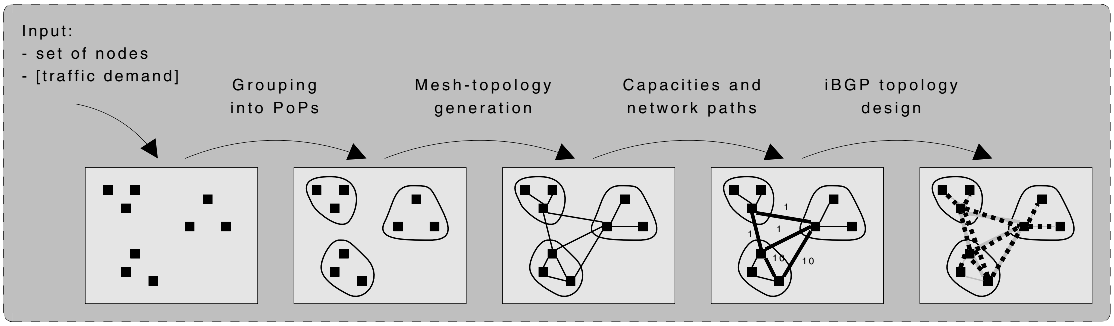

Fig. 1. Network design methodology.

given by Norden [30]. It relied on the computation of link capacities sufficiently large to carry a traffic demand, using linear programming (see [17]).

D. Administrative cost metrics

Another important parameter of IP networks is the assignment of administrative costs to links. These costs or IGP weights are used by routing protocols such as IS-IS or OSPF to compute the least-cost paths. The selection of these weights will therefore influence the selection of the paths used to cross the network. Real networks use a few well-known schemes to assign these weights depending on their objectives. Schemes exist to optimize the network for delay or for throughput or a combination of both. It is possible to integrate such schemes in the generation of topologies.

E. BGP sessions graph

The physical Internet topology is overlayed by a graph of BGP sessions. This graph has two important characteristics that influences the selection of paths by the interdomain routing protocol, BGP. The first one is the presence of policies on external BGP (eBGP) sessions. These policies constrain the authorized interdomain paths [19]. The second characteristic is the *graph of internal BGP (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 can cause important changes in the selection of interdomain routes [41]. Typically, small domains will use a full-mesh of iBGP sessions while larger domains will organize their iBGP topology around route-reflectors [35].

III. ROUTER-LEVEL TOPOLOGY GENERATION

In this section, we present a new structural approach to the design of router-level network topologies. Our method is similar to the approach followed by a network designer. The network designer typically starts with the set of nodes to be interconnected and possibly an estimation of their traffic demand. It will then use network design heuristics and operational guidelines in an incremental manner to build a close-to-optimum topology.

The methodology is illustrated in Fig. 1 and is implemented in IGen, an open-source topology generator. The methodology is composed of 6 main steps. For each step, the user of IGen can specify its own design goals.We describe each step in more details in the following subsections.

- **Placing routers:** The methodology relies on a set of nodes with their geographical location.
- **Identification of PoPs:** In a first step, the nodes are grouped into PoPs, based on their geographical location. We also identify within each PoP nodes that will be part of the backbone.
- **Building the topologies of PoPs:** In a second step, the structure of each PoP is built. We rely on operational practice to build realistic PoP structures.
- **Building the backbone topology:** In a third step, a topology for the backbone is produced. This is basically a graph which interconnects all the backbone nodes of the PoPs. Various heuristics can be used to generate the backbone.
- **Link capacities and network paths selection:** In an optional step, capacities and IGP weights can be assigned to links. IGP weights will influence the selection of intradomain paths by an IGP.
- **iBGP topology design:** Finally, the network topology can be overlayed with a graph of iBGP sessions. Different iBGP graphs are possible.

A. Placing routers

Our methodology requires an initial set of nodes with geographical coordinates as input. The most obvious source is to rely on an existing network, but it is not always possible to obtain the location of routers for confidentiality reasons.

Another possibility is to generate the set of nodes randomly. For this purpose, we designed a set of polygons, each one representing a continent and we generated router locations using two independent and uniformly distributed random variables X and Y. We constrained the resulting coordinates to fall into a selected subset of polygons to generate regional or international networks.

Another example of suitable geographic locations is provided by IP-geoloc databases. We experimented with MaxMind [1], a database obtained by a technique of geographic mapping of Internet hosts [33]. We crossed the database with a BGP routing table dump obtained from RouteViews[28] at the same date in order to group IP addresses in BGP domains. We were thus able to obtain a realistic source of geographical locations of hosts for a large number of domains. Among these locations, we can pick those contained within a single domain as input for the generation.

B. Identification of PoPs

A Point of Presence (PoP) is a physical location where a domain has equipment [35]. The location of a PoP is typically a building in a city, a metropolitan area or a zone of industrial activity [23]. A first characteristic of a PoP is that the routers that it contains are geographically close to each other (many of them are usually in the same room).

To identify the PoPs of the network, we use clustering methods to build groups of nodes. 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 [26] 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.

C. Generating the PoPs topologies

The structure of a PoP is often carefully designed. There are well-known operational practices to build a PoP [35], [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 are many 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 described in operator forums for other large networks [20].

Our methodology for building the structure of a PoP is inspired from the above operational practice. We show in Fig. 2 the topological structure of a typical PoP design [22]. 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 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).

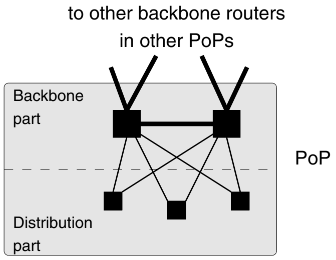

Fig. 2. Structure of a PoP ($n = 2$ and $k = 2$).

D. Generating 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 [29], [6], [21]. Their operation usually consists in building a seed network topology with built-in requirements such as a maximum number of hops 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 predicted demand). 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.

The main heuristics implemented in the current version of IGen are summarized in Table I. We briefly describe each heuristic in the following paragraphs[2].

MENTOR	Hybrid MST-SPT
MENTour	Minimum length hamiltonian cycle
Two-Trees	Union of two edge-disjoint MSTs
Delaunay	Delaunay triangulation
Multi-Tours	Union of Cycles

TABLE I
SUMMARY OF MAIN BACKBONE GENERATION HEURISTICS.

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 previous node.

---

[2] More in-depth descriptions can be found in the IGen technical report available from `http://inl.info.ucl.ac.be/softwares/igen`

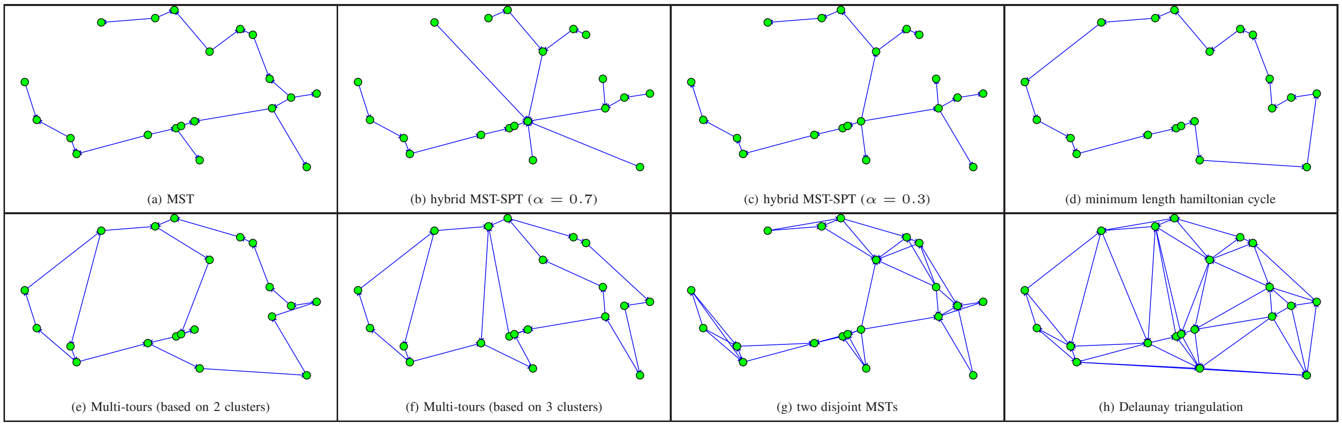
(a) MST
(b) hybrid MST-SPT ($\alpha = 0.7$)
(c) hybrid MST-SPT ($\alpha = 0.3$)
(d) minimum length hamiltonian cycle
(e) Multi-tours (based on 2 clusters)
(f) Multi-tours (based on 3 clusters)
(g) two disjoint MSTs
(h) Delaunay triangulation

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

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. [15]. 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 HOT is that MENTOR 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, i.e. they are not robust to single failures, 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. A variation of MENTour is a heuristic where the set of nodes is initialy partitionned and a tour is computed for each subset. Tours are then connected together. We call this heuristic **Multi-Tour** and it provides networks composed of multiple rings.

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 [29] 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 [11] 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. The Delaunay triangulation has the interesting property that 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. If the Delaunay triangulation is too costly in terms of the number of edges, it is possible to remove the edges that are never used after any single edge failure of the network.

We show in Fig. 3 a visual illustration of the backbones generated by the various heuristics to interconnect a randomly generated set of 20 vertices covering Australia. The (a) network is the MST, it is the network that minimizes the link mileage. Networks (b) and (c) illustrate how the parameter of MENTOR can build trees between the MST and the SPT. The (d) network is composed of a unique tour that traverses each node. Networks (e) and (f) are composed of multiple interconnected tours. Network (g) is the union of two disjoint MSTs. Finally, network (h) is composed of triangles.

E. Assigning IGP weights

Once the topology has been generated, it is possible to assign IGP weights to the links. In the current version of IGen, we provide two schemes to assign IGP weights.

The first 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 used in Abilene, the US research backbone network.

The second scheme consists in basing the IGP weight of links on the **inverse of their capacity** and is known as the *Cisco default metric*. This scheme leads to intradomain paths using the highest bandwidth links and it is supposed to favour

Equal-Cost-Multi-Path (ECMP) in a least-weight routing configuration. GEANT, the pan-european research backbone relies on this scheme with small adaptations to shorten the delay between specific nodes.

F. Assigning capacities

In this step, the generator can assign capacities to links. The capacities are selected among a set of discrete values that correspond to existing link technologies (for example 2.4Gbits/s for an OC-48 link). We can use two different methodologies to assign the capacities.

The first method relies on the fact that traffic aggregation occurs mostly in the backbone. Therefore, we make the assumption that the bandwidth of the backbone links is usually larger than the bandwidth of the links between backbone and access routers. For the topology generator this means that two distinct link capacities can be assigned: high bandwidth into the backbone, for example 10Gbit/s, and lower bandwidth at the access, for example 1Gbit/s.

The second method is more traffic oriented, and as such, it requires the knowledge of the traffic demand. The link capacities are assigned in order to ensure that the matrix of traffic demand can be accomodated. For this purpose, 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 utilization of each link. If there are multiple equal-cost paths, the volume of the demand is split equally among the available paths as in [16].

The above methodology can be refined so as to limit the maximum link utilization to a predefined level τ. This corresponds to real world operational practice where for instance, links are often given a capacity such that the load will be 40 to 50 % [40]. 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.

G. Design of the iBGP topology

The final step of IGen can automatically generate the topology of BGP sessions within the network. This logical topology can strongly differ from the physical 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. This requires $n.(n-1)/2$ sessions. Fig. 4 shows on the left an example of a simple network topology composed of two PoPs and, on the middle, the full-mesh iBGP topology for the same network.

The second common iBGP deployment is an iBGP hierarchy where routers are divided in two groups: route-reflectors and the other routers named clients. The route-reflectors are allowed to propagate BGP routes to their clients [4]. An example of an hierarchical iBGP topology is shown on the right part of Fig. 4.

In IGen, it is possible to generate the iBGP topology as a full-mesh or as an hierarchy where backbone routers are configured as route-reflectors to which the other routers in the PoPs are connected as clients.

IV. INTERDOMAIN TOPOLOGY GENERATION

This section deals with the generation of topologies composed of multiple domains. The ultimate goal is the generation of complete router-level Internet topologies. We rely on the methodology explained in Section III to build the topology of each domain separately. Then, we merge all the topologies and we add links to connect domains together. We need to solve two problems when connecting domains together. First, we need to obtain or generate a graph of domains where each edge tells what business relationship [19] is established between the connected domains: *customer-provider* or *peer-to-peer*. Second, we need to decide if the interconnection between two domains is organized across multiple peering links and where these links are deployed.

To obtain a graph of domains, we rely on AS-level topologies such as the ones inferred by Subramanian et al. [38]. This dataset[3] contains the interdomain relationships that exist between Internet domains. It is also possible to rely on synthetic AS-level topologies with embedded policies such as those produced by GHITLE [12]. We cannot rely on traditional degree-based AS-level topology generators [27] as they do not care with routing policies.

To determine the number of peering links used to connect a pair of domains together, we rely on the size of these domains. Our hypothesis is that the larger the domains are, the larger the number of peering links will be. We implement this based on an empirical formula $N_{ij} = 1 + \left\lfloor (N-1).\frac{N_i.N_j}{(\max_i N_i)^2} \right\rfloor$ that gives the number of interdomain links N_{ij} between domains i and j as a function of the sizes of these domains in terms of the number of routers. N_i and N_j are the number of routers in domain i and j respectively. The maximum size used as a normalization factor is computed on all the domains in the graph. This formula gives a number of links that is in the interval $[1, N]$. The value N is a parameter fixed by the user.

Finally, to place the 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_{ij} interdomain links is as follows. We start with N_{ij} 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 iterate with the updated set of vertices until N_{ij} links are placed.

V. 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

---

[3] We are aware of the limitations of such dataset [13]. 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.

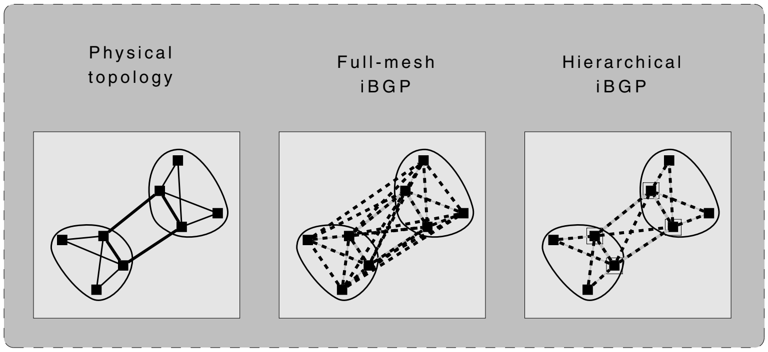

Fig. 4. Full-mesh and hierarchical iBGP topologies.

have then been proposals for inferring network topologies at the router-level from measurements. Rocketfuel [37] 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 [24], [39]. In addition, these techniques sometimes fail to resolve router aliases resulting in links and routers that do not really exist [39], [36].

At the AS-level, techniques such as [38] 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 contain the best routes selected by BGP [2]. These topologies are thus not representative of the actual diversity of the AS-level paths. A systematic study of the completeness of the inferred topologies based on local views was provided by Willinger et al. [10]. In addition, inferred AS-level topologies do not provide information on the number of peering links between domains nor information on the internal structure of domains. Other inferrence techniques have been proposed later, such as [5] and [13]. The latter shows that more peering links can be discovered when using BGP updates. However these techniques suffer from limitations similar to those studied in [10].

Another approach consists in generating synthetic topologies sharing selected properties with the real Internet. Available generators such as BRITE [27] or 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 [42]. The problem of this approach is that topologies are generated in order to mimic pure graph properties. They fail to capture the optimization process that is also at the basis of real ISP 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 [25], Li et al. have proposed 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 [15] 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.

VI. CONCLUSION

In this paper, we have shown that network topologies are the outcome of a careful design process taking into account several objectives such as the minimization of the latency and the accomodation of a traffic demand. Moreover, these optimization objectives are combined with additional constraints such as the location of nodes to interconnect and the robustness to failures. Today, most network researchers still rely on randomly generated topologies that have often nothing in common with real networks.

We have proposed another methodology for building more realistic network topologies. Our methodology uses network design heuristics similar to those used by network designers. We have implemented this methodology in IGen, a publicly available topology generation tool-box. In IGen, the topology structure is explicitly driven by the user's design goals. By selecting among a large choices of mesh generation heuristics as well as parameters such as how the link capacities or IGP weights are assigned, the user can direct IGen to generate a large number of different topologies.

IGen is an initial step towards building more realistic network topologies with explicit design objectives. The methodology has plenty of room for refinements. For example, to assign link capacities, we plan to simulate the single-link/router failures and compute the capacity required to accomodate all (or a subset of) the failures similarly to commercial network design tools [43]. We plan to add optimized IGP weights such as computed by Fortz et al. [18] or Nucci et al. [31].

ACKNOWLEDGMENTS

This work was funded by the Walloon Region under the WIST TOTEM project. This work also received partial support from Trilogy (http://www.trilogy-project.eu), a research project (ICT-216372) partially funded by the European Community under its Seventh Framework Programme. The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document. Pierre Francois is supported by the FNRS (Fonds National de la Recherche Scientifique, Belgium). We thank Cedric de Launois, Cristel Pelsser, Steve Uhlig, Jean Lepropre and Wolfgang Mühlbauer for their comments on earlier versions of IGen and this paper. We also thank the Abilene Observatory for providing so much details about the Internet2 bacbkone network. We thank Nicolas Simar and Otto Kreiter for providing snapshots of the LSDB of the GEANT network. We also thank Mark de Berg for his comments regarding some corner cases of the Delaunay triangulation.

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] T. Bates, R. Chandra, and E. Chen. BGP Route Reflection - An Alternative to Full Mesh IBGP. Internet Engineering Task Force, RFC2796, April 2000.

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

[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. Extending and Enhancing GT-ITM. In *Proceedings of ACM SIGCOMM Workshop on Models, methods and tools for reproducible network research*, pages 23–27, August 2003.

[9] Cariden Technologies. MATE. `http://www.cariden.com/`, 2005.

[10] 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.

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

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

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

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

[15] 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.

[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] 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.

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

[25] 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.

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

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

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

[29] 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.

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

[31] 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.

[32] OPNET Technologies. SPGuru. `http://www.opnet.com/`, 2005.

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

[34] 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.

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

[36] R. Sherwood, A. Bender, and N. Spring. Discarte: a disjunctive internet cartographer. In *Proceedings of ACM SIGCOMM*, pages 303–314, August 2008.

[37] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proceedings of ACM SIGCOMM*, August 2002.

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

[39] 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.

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

[41] S. Uhlig and S. Tandel. Quantifying the impact of route-reflection on BGP routes diversity inside a tier-1 network. In *Proceedings of IFIP NETWORKING 2006*, May 2006.

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

[43] Wide Area Network Design Laboratory (WANDL). IP/MPLSView. `http://www.wandl.com`, 2005.