

TopGen - Internet Router-Level Topology Generation Based on Technology Constraints

Ingo Scholtes
Systemsoftware and
Distributed Systems
University of Trier
D-54286 Trier, Germany
scholtes@syssoft.uni-trier.de

Jean Botev
Systemsoftware and
Distributed Systems
University of Trier
D-54286 Trier, Germany
botev@syssoft.uni-trier.de

Markus Esch
Faculty of Sciences,
Technology & Communication
University of Luxembourg
1359 Luxembourg,
Luxembourg
markus.esch@uni.lu

Alexander Höhfeld
Systemsoftware and
Distributed Systems
University of Trier
D-54286 Trier, Germany
hoehfeld@syssoft.uni-trier.de

Hermann Schloss
Systemsoftware and
Distributed Systems
University of Trier
D-54286 Trier, Germany
schloss@syssoft.uni-trier.de

Benjamin Zech
Systemsoftware and
Distributed Systems
University of Trier
D-54286 Trier, Germany
zech@syssoft.uni-trier.de

ABSTRACT

In order to realistically simulate algorithms or evaluate P2P overlay topologies, a detailed model of the underlying router topology is required. Since actively measuring this topology is extremely laborious and furthermore a waste of network resources, traditionally topology generators are used in order to create synthetic router-level graphs. For this, usually graph models are selected that are known to generate graphs which are similar to the actual Internet in respect to a certain metric like e.g. vertex degree distribution. These models are often superseded or adjusted when new metrics are being introduced which better differentiate between graphs. Furthermore it has been shown, that graphs that are similar in respect to e.g. vertex degree distribution can be very different from a structural point of view. In this paper TopGen, a generic, extensible and easy-to-use topology generation platform is presented. It contains a topology generation module which bases the generation of router-level graphs on the Internet's underlying principles and the technological constraints of routers rather than trying to effectuate similarity in respect to a certain metric. Apart from describing TopGen's general topology generation approach, graphs created with its Internet topology module are evaluated and found to be encouragingly similar to real-world datasets in various respects.

Categories and Subject Descriptors

C.2.1 [Computer Systems Organization]: Computer-Communication Networks—*Network Architecture and De-*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIMUTOOLS 2008, March 03-07, Marseille, France
Copyright © 2008 ICST 978-963-9799-20-2
DOI 10.4108/ICST.SIMUTOOLS2008.3008

sign; I.6.7 [Computing Methodologies]: Simulation and Modeling

General Terms

Design, Measurement

Keywords

Internet, Network Topology, Topology Generation, Router

1. INTRODUCTION

Synthetically generated Internet topologies are required in many fields of network research. Realistic simulations of distributed algorithms as well as the evaluation of P2P overlay topologies depend on the realism of these synthetically generated graphs. During the last couple of years, the selection of random graph models used for topology generation has been largely extended by the discovery of power-law properties of the Internet's router-level graph [12]. Today most topology generators are based on random graph models like Barabasi/Albert [2], Power-Law Random Graph (PLRG) [1], Watts/Strogatz [27] or Waxman [28], which are known to produce vertex degree distributions similar to those of real-world graphs [12].

As presented in [16], this methodology may lead to graphs that - apart from a certain graph theoretic metric - do not have much in common with real-world graphs. The actual structure and performance of a graph may vary in a wide range yet leaving graph theoretical properties unchanged. For their argumentation, the authors of [16] have compared a real-world network topology with some others that have been generated based on different random graph models. While all of these topologies offer identical degree distributions, it has been shown that the networks largely differ in respect to structure, throughput, utilization and fault-tolerance. Accordingly instead of generating random graphs that are similar in respect to some graph theoretical properties, one needs to provide a generation model that mimics the way how network topologies are built in real life.

When looking at the Internet router graph, the authors of [16] argue that there are economic and technical constraints that have great influence on fundamental graph structures. There is e.g. no router with an arbitrarily large port number or bandwidth. Furthermore the technical capabilities of very expensive and highly available routers that are located in the Internet’s backbone infrastructure differ largely from those of cheaper routers that can be found at the network’s edges. Thus, any graph model that treats the variety of different routers as a homogeneous vertex set is unlikely to produce realistic graphs. Replacing the term vertex type by group membership, this fact has already been recognized by Ravasz and Barabási in [21].

In section 2 we give a brief overview of existing topology generation approaches and motivate our decision to implement TopGen, a generic and extensible topology generation platform which has been designed to specifically address the aforementioned issues. It introduces an intuitive model with which graph generation can be based on different router types and their technical constraints rather than on randomness and a uniform vertex set. TopGen contains a simple development environment that can be used to develop custom topology generation modules in an easy fashion. It also provides a comprehensive set of predefined generation modules which can be combined and instrumented for own purposes. By this means a wide variety of networks can be created without the need to write a single line of code. Apart from that, TopGen provides integrated analysis facilities that allow to measure graph theoretical properties like vertex cover, assortativity [20], relative likelihood [16], diameter and clustering coefficient. Furthermore, the probability distribution function (*pdf*), cumulative distribution function (*cdf*) as well as the complementary cdf (*ccdf*) of vertex degree distributions can be computed. An integrated function fitter can be used to instantly perform linear, exponential and power-law regression as well as extract Gaussian coefficients of a given degree distribution. TopGen has been implemented based on the Microsoft .NET 2.0 framework, own custom topology generation modules can be implemented in any language supported by .NET or the compatible, platform-independent and Open Source runtime environment Mono. For the moment the development environment integrated into TopGen however only supports C#.

Apart from the provision of a modern integrated topology generation platform, the main contribution of this paper is the provision of a ready-to-use Internet topology generation module. It is suitable to create realistic router-level graphs which are solely based on the fundamental technological constraints of routers and principles driving the Internet’s evolution. In section 3 we will describe this Internet topology generation module in detail and by this exemplify TopGen’s underlying graph generation approach. In section 4 we compare a router-level graph generated by our Internet topology generation module with a real-world dataset as well as graphs based on the Barabási/Albert model which is the foundation of many topology generators. These measurements show that TopGen generated graphs are structurally very similar to the actual Internet router-level graph. We will describe our main contributions, open issues, weaknesses and validity threads in more detail in section 5.

2. BACKGROUND AND RELATED WORK

In a current research project that aims at the provision

of efficient global-scale P2P overlay topologies for highly-interactive and thus latency-sensitive applications, we encountered the need to evaluate the performance of different overlay topologies in terms of latency, load-balancing and their suitability for today’s and tomorrow’s Internet architecture. In order to perform simulations that respect all of these aspects, realistic large-scale router-level graphs of the Internet are required that provide sound per-router bandwidth and end-to-end hopcount information. In order to obtain graphs that are suitable for our needs, we evaluated the market of existing sources.

2.1 Related Work

Whenever there is the need for Internet topology datasets, there are two basic approaches to obtain these data. In the following two paragraphs some prominent examples for each approach will be described.

Topology Generation.

The field of topology generation has been extensively worked on. The authors of [8] give an exhaustive overview of existing topology generation approaches and group them into five categories according to their underlying graph models:

(1) Random Graph Models, (2) Preferential Attachment models, (3) Optimization-based models, (4) Geographic models and (5) Internet-specific models. As it is beyond the scope of this paper to describe all of these approaches, for our purpose of global-scale, latency-sensitive simulations it is sufficient to have a closer view on generators that fall into categories (2), (4) and (5).

A prominent and widely-used example for an Internet-specific topology generator is Inet [29]. It provides a single generation methodology that can be tweaked using a set of fixed parameters. Inet has been designed to generate AS-level graphs in which each vertex represents a collection of routers within an Autonomous System (AS). Accordingly, generated graphs do not contain fine-grained information on routers, bandwidth or latency and are not suitable for our purpose. The BRITE generator [18] is an example for a flexible topology generation platform incorporating several models like Waxman [28] or preferential attachment [2]. Some of these models are suitable to create AS-level graphs, others are specific to router-level topology generation. Considering the modules that operate on router-level, bandwidth and delay values are assigned to routers only after the topology has been generated according to a certain distribution function rather than emerging from the way the network is built. In reality bandwidth and link constraints of routers are - according to [16] - “a significant force shaping network connectivity” and should therefore be considered in topology generation. For the OMNeT++ simulation environment [26], a descriptive approach has been chosen for topology creation. While no statements are made on specific topology generation models and their evaluation, [26] describes a scheme in which different so-called module types can be textually specified. Using this scheme, at least different router classes can be defined although there does not seem to be a possibility to define additional technological constraints that are respected when interconnecting routers.

There are several structural generators, which try to focus on the hierarchical nature of the Internet rather than creating a certain degree distribution. Comparing the structural generators Tiers [10] and Transit-Stub [6] with degree-based

generators it has been shown in [25] that (a) (loose) hierarchical structures may also arise from degree-based generators and (b) existing degree-based generators generally perform better in terms of graph theoretical metrics. Being the first topology generator producing power-law structured network topologies, GridG [7] uses the Tiers structural generator and enforces the outdegree power-law of generated graphs by adding additional links in a downstream process.

It appears obvious that technology constraints applicable to routers largely influence Internet topology. Specifically bandwidth and link number constraints largely influence the way in which providers structure their networks as well as how provider networks are interconnected. To date, to the best of our knowledge, none of the topology generators described above has specific support to define, distribute and respect router-specific constraints when generating router-level networks. Existing generators are built to optimize a certain set of metrics and usually become obsolete or are adapted as soon as additional metrics are introduced that are suitable to better differentiate between graphs.

Active Probing.

Another approach for obtaining data on the Internet topology is to actually measure it by distributed active probing. While the datasets that have been obtained by projects like CAIDA [9] or DIMES [23] are extremely useful in order to compare generated topologies with real-world data, they do not contain information on link capacity, router bandwidth or the geographical region a vertex originates from. While this information is hard to extract from real-world graphs, it is crucial for the simulations we intend to run on the router-level graphs.

While projects like King [13] try to estimate latency between arbitrary end hosts based on online DNS request latency measurements, [30] have used existing delay data in order to deduct metrics that are suitable to create the Internet delay space synthesizer DS^2 . Although being the first attempt to synthetically generate such information it remains uncertain how it can be combined with router connectivity information in order to obtain a comprehensive router-level model of the Internet. Nowadays active probing also leaves the bitter aftertaste of additionally straining the Internet infrastructure for the sake of statistics. Another disadvantage is that some of the world's regions like Africa and Asia are often underrepresented in measurements. Using active probing it also is impossible to obtain a router-level graph that represents the future Internet, something which a good topology generator can possibly provide. It is also hard to scale down a topology to a smaller graph offering the same structural properties, something that is extremely useful for performing realistic and fine-grained simulations in the presence of limited simulation resources.

3. TOPGEN DETAILS

Based on the evaluation of existing graph generation approaches that has been performed in section 2 and the results presented in [16], we decided to implement a topology generator that specifically addresses our requirements. In order not to create a monolithic generator that is constrained in terms of flexibility and parameterizability, we decided to implement a generic, integrated graph generation environment that allows a stepwise refinement of generation methodology. Such an environment should ideally contain everything

that is required to implement, parametrize and run graph generation modules as well as evaluate resulting topologies. Special emphasis was placed on the fact that it must be possible to implement topology modules that rely on a minimum set of technological constraints and principles driving the Internet's evolution. Furthermore the framework should be generic enough to implement arbitrary graph generation models in a way as simple as possible.

The outcome of our efforts - TopGen - will be presented in more detail in this section. It comes with a set of popular random graph generation models like Barabási-Albert, Erdős-Renyi [11], Watts-Strogatz, Waxman as well as several structural modules like tree, mesh etc. Apart from these simple models, a more complex and realistic Internet-style topology module has been implemented. This module rests topology generation on a minimal set of structural and quantitative assumptions. These assumptions have their seeds in fundamental router capabilities and forces driving the Internet's evolution. This section will present TopGen's basic graph generation approach and exemplify each aspect with its usage by the Internet generation module along the way.

3.1 Router Types

The vast majority of existing graph generation approaches does not respect the fact that vertices in real-world graphs are not homogeneous objects. When looking at a scenario in which each vertex represents an Internet router, edges can often only exist between vertices of certain types. Furthermore there may be restrictions on how many links to other vertices of certain types may exist. In a graph with weighted edges there might as well be a maximum for the sum of weights of all incident edges which resembles the maximum bandwidth of a certain router. Given these considerations one gets a glimpse of the complexity of real-world networks and how this complexity is often oversimplified by topology generators.

In order to respect the constraints mentioned above, in TopGen different router types can be specified. A router type consists of a name, a maximum link number, a maximum bandwidth, an initialization factor (whose purpose will be shortly described) and a deviation value. Whenever a new router instance of a certain type is created, Gaussian distributed random bandwidth and link values will be assigned to it based on the user-specified deviation using a Box-Muller transformation [5]. These deviation values are important because in reality Internet routers of a certain type will not exactly have the same technological constraints but rather fluctuate within a certain spectrum. As the exact distribution of router capabilities within a router type class is unknown and influenced by numerous factors, we think that it is reasonable to assume a normal distribution.

The provision of a router type abstraction is crucially important for TopGen's Internet topology module. Capturing the different constraints in terms of bandwidth and link number for routers at a certain level of the Internet topology (like e.g. IXP Peering, Intra-AS, Access Networks) is simplified by coarsely categorizing them into four classes. These loosely resemble those described in [4]. Each class represents one router type as used by the TopGen Internet topology module.

- *Core*: Core routers are the highest capability routers at the Internet's backbone level. Routers at this level usually have a small number of very high bandwidth

connections that interconnect Autonomous Systems (AS) (as described in [14]) like ISPs or enterprises by means of peering [3]. Core routers are also responsible to route packets between different Internet Exchange Points (IXPs).

- *Border*: Border Routers connect the internal backbone router infrastructure of an AS to the core routers at the IXP. They act as a gateway between the AS-internal network and other networks. Border routers generally have a small number of high bandwidth connections. Each border router loosely resembles the entry point into an AS.
- *Gateway*: Gateway routers are used to accumulate traffic from the network's edges towards border routers in a hierarchical AS-internal structure. They also connect hosts inside an AS that are not connected to the same edge router. We expect routers of this type to have an intermediate number of intermediate bandwidth links.
- *Edge*: Edge routers are used to connect actual hosts and aggregate traffic from end hosts. Examples for these routers are DSL Access Multiplexer routers (DSLAM) in a DSL environment or access switches used for dial-up aggregation [3]. Edge routers are expected to maintain a large number of links with comparably small bandwidth to end users.

In order to extract a set of sound constraints for bandwidth and link number of the router types that have been described above, we intended to use a minimal set of idealized quantitative assumptions which are based on the current state-of-the-art technology and the latest Internet usage statistics.

- The bandwidth of the highest capability routers available is about 1.6 Tbps. This value is justified by a look into the product catalogs of major router providers like Cisco Systems¹ or Juniper Networks².
- The average edge router serves 100 hosts at 1.5 Mbps each (DSL-setting) or 3000 hosts at 56 Kbps each (Dialup-setting). We copy these assumptions from [3]. Please note, that the actual technique used (DSL or dialup) does not influence the graph structure because edges to end hosts are not being considered as they are not part of the router-level graph. At this point we are only interested in the bandwidth between gateway and edge routers, which sums to 150 Mbps using the above assumptions for both settings.

Please note, that all of the above values are configurable in the actual topology generation process. Based on these assumptions we obtain a space of feasible parameters for average link number and maximum bandwidth of all four router types. Using the relation between user bandwidth and link number of edge routers assumed above, we can approximate the average bandwidth of edge routers by 150 Mbps. Assuming a maximum bandwidth of 1.6 Tbps for core routers, a two-level tree structure and setting c as core

router link number, b as border router link number and g as gateway link number, we find that $c \cdot b \cdot g \cdot 150 \text{ Mbps} < 1.6 \text{ Tbps}$, so $c \cdot b \cdot g < 10666$. From the different router type properties, we extract $c < b < g$. From this space of feasible parameters, we chose $c = 10$, $b = 16$ and $g = 64$ but other values may be used as well. With these example values we obtain the following technical constraints for the different router types:

- Edge: maximum bandwidth 150 Mbps
- Gateway: maximum 64 links at 150 Mbps each, maximum bandwidth 9.6 Gbps
- Border: maximum 16 links at 9.6 Gbps, maximum bandwidth 153.6 Gbps
- Core: maximum 10 links at 153.6 Gbps, maximum bandwidth 1.536 Tbps

We set the deviation of link and bandwidth capacities to 90%, 50%, 25% and 15% for edge, gateway, border and core routers respectively. This implies that a router of a certain type with deviation d has a 95% probability to have technical constraints that fall within a $\pm d\%$ range around this router type's base value. The different deviation values for different router types mimic the fact, that the higher the router capacity (and price) of a certain router type is, the less different router models with different characteristics of this type will probably exist.

3.2 Router Creation

The main design goal behind the implementation of TopGen was to create a maximum generic, modular, convenient and modern topology generation platform. Thus any TopGen-based topology module - be it structural or degree-based - should be implementable in a way that is as simple as possible. In order to achieve this, one may think of different approaches to graph generation.

Vertex Connector Approach.

Using this approach, all vertices/routers are generated by the generation platform and passed to generation modules which connect existing vertices according to the implemented model. Graph generation modules do not actually generate the graph but only connect existing vertices by creating edges between them. The main advantage here is, that different generation modules can be easily combined by letting them operate on the same vertex set. Due to TopGen's router type abstraction, vertices are partitioned in different classes. Thus it may be complex to correctly generate the right number of vertices for a certain connector that needs a minimum vertex number of a certain type in order to function correctly. This approach also does not fit well for graph generation models that incrementally "grow" graphs by adding vertices like e.g. Barabási/Albert [2].

Vertex Generator Approach.

In this approach, only parameters are passed to a generation module which will then generate vertices and edges based on the parameters and return the generated graph. This is the principle behind most existing graph generators. While it certainly fits incremental models like Barabási/Albert better, it complicates the application of several graph generation modules on the same graph. For real-world graphs,

¹<http://www.cisco.com>

²<http://www.juniper.net>

it is unlikely that their emergence is based on a single model. They often rather involve a mixture of different models which might be laborious to implement generically in a single generation module.

Hybrid Approach.

In order to combine the advantages of both approaches, a hybrid scheme has been used. Part of the vertices can be pregenerated by the platform, passing them to vertex connector modules while others may be added by a generation module and passed back to TopGen. The user can specify a total router number n and different router types t_1, \dots, t_k . Each router type t_i is assigned an initial factor f_i . Initially TopGen will generate $n \cdot f_i$ routers of each router type t_i and pass those routers to the specified generation modules. By setting the factor f_i to zero, the user can defer the task of creating routers of a certain type to the generation module. Routers generated by a module are passed back to TopGen from where they can be used in downstream generation modules. By this means, different generation modules can be combined to a composite scheme in an easy fashion.

TopGen's Internet topology module makes use of the hybrid approach. Since traffic in the Internet is mostly generated at the network's edges by users, a fixed number of edge routers (which resembles the number of users) and no core, border and gateway routers are passed to the generation module by TopGen. Edge routers will be incrementally attached to a configurable initial set of core, border and gateway routers that will be created as soon as the generation module starts up. Since router-type dependent bandwidth and link number constraints are respected by the generation module this will lead to the successive creation of additional core, border and gateway routers as more and more edge routers are added. These are passed back to TopGen when generation has finished so they can be used by downstream router connection modules. The scheme by which routers and links to existing ones are created is described in more detail in the following section.

3.3 Router Interconnectivity

Based on the router types described above, TopGen introduces the notion of a router type association (RTA) being a tuple of router types. For each RTA, a different topology module can be specified. Topology modules are required to implement a `Generate` method which takes a vector of vertex sets as an input argument. Each set may contain a number of routers all being of the same type. By this means, topology modules are made aware of the router type structure as well as their technological constraints. This facilitates generation modules which for example only connect routers of certain types or respect a maximum bandwidth that has been assigned based on considerations similar to those in section 3.1. By ignoring the distinction between different vertex sets and operating on their union it is just as simple to implement graph models that do not differentiate between vertex types.

For TopGen's Internet topology module, we assume that core routers are interconnected via a redundant high-performance mesh network. Border routers connect several gateway routers to one core router and gateway routers connect several edge routers to one border router. From a router-level perspective each edge router is redundantly connected to a small number of gateway routers only. Figure 1

shows a simple example graph as it might have been generated by the TopGen Internet topology module based on the assumptions that have been postulated above. The different router types are represented by their initial letters. In order to connect routers as seen in Figure 1, we defined an RTA (Edge, Gateway, Border, Core) and implemented a topology module which interconnects routers of certain types as described above.

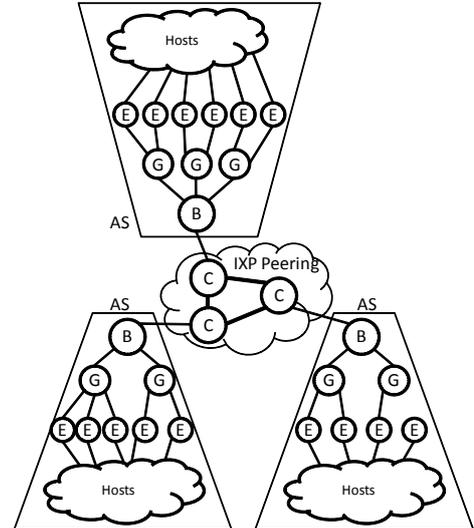


Figure 1: Sample Internet Topology with Core, Border, Gateway and Edge routers

Based on this general idea, we will now give a more detailed description of how the Internet topology module will use this information to generate router-level graphs:

1. Start with an initial number of core routers.
2. Attach a single border router to each core router and a single gateway router to each border router.
3. Incrementally add the user-specified number of edge routers to gateway routers. This resembles the fact, that the router-graph grows because more hosts need to be connected to the network. The probability that a certain edge router is connected to the subtree of a certain border router is chosen to be proportional to the number of edge routers that are already present in the subtree. This resembles the Pareto-principle which is in real-life justified by the fact that huge Internet providers or AS are more likely to become even larger, while smaller ones are more likely to remain small.
4. Whenever a new edge router is connected to the gateway infrastructure, technological constraints of all involved routers will be checked and the router graph will be reorganized according to the following rules:
 - If the cumulative bandwidth of all edge routers in the subtree of a gateway router exceeds the gateway router's maximum bandwidth, a new gateway router is created and attached to the border router. The new gateway router will be attached to a configurable number of existing gateway routers in the same core router subtree. This

mimics the way, how AS and ISP networks grow when the number of connected hosts rises. By interconnecting gateway routers, border routers are relieved from traffic between users present in the same AS.

- If the cumulative bandwidth of all edge routers present in the subtree of a border router (within an AS) exceeds the border router's maximum bandwidth, a new border router is attached to the backbone infrastructure and both border routers will be interconnected. The interconnection of border routers resembles the fact, that provider or AS internal traffic is usually not routed via IXP core routers.
- If the cumulative bandwidth of all edge routers present in the subtree of a core router would exceed the core router's maximum bandwidth, a new core router will be chosen or created to which the new edge router can be connected via border and gateway routers. This scheme resembles the way how IXP provider networks grow when a customer requires additional ports.

3.4 Regional Router Composition

Another observation that can be made in the Internet is that the router-level structure and especially end-to-end latency largely depends on geography. Geographical but also political realities can to a large extent influence interconnectivity of routers and latency of links between different regions. There are e.g. only a few high bandwidth links that bridge oceans and interconnect the router-level graphs of different continents. Furthermore, the composition of router types and their capabilities may differ from region to region. In order to facilitate users to respect these facts, TopGen introduces the notion of region graphs. Router types, router type associations as well as graph generation module settings can be specified for each region separately. In a first step, TopGen will use the specified generation modules and settings to create independent regional router-graphs. Special so-called global router type associations that are allowed to span router types present in different regions may be defined. Generation modules can be assigned to these as well just like for regular RTAs. Graphs of different geographical regions will then be assembled to a global router graph based on these modules. By this means, one can e.g. create independent router graphs for each continent or country, interconnect highest bandwidth routers of each region according to their physical adjacency and label intercontinental edges with sound latency values.

The comparably high independence of different region graphs can be utilized in order to maximize generation performance. Due to its multiprocessor awareness, TopGen will automatically generate independent region subgraphs in parallel on different processor cores. The resulting region graphs will then be collected and interconnected in a downstream process based on the global router type associations that have been defined. With this generation approach, a realistic router-level Internet topology graph consisting of more than 200,000 routers and 600,000 links in six regions can be generated in less than 2 minutes on an Intel Core 2 Quad Q6600 machine with 2GB of memory.

For the regional composition of the Internet router-level graph, we used the following assumptions for TopGen's In-

ternet topology module:

- The Internet router-level graph is fragmented into at least six clearly distinguishable regions which represent the continents Africa, Asia, Europe, North America, Oceania and South America. Distance between those most likely implies that only a relatively small number of dedicated high-bandwidth/high-latency links exist between them based on geographic adjacency.
- Links between these regions mainly exists between core routers. This assumption is justified by the way how Internet Exchange Points are organized and interconnected.
- Traffic in the Internet is mostly generated at the network's edges by hosts connected to edge routers.
- In order to estimate the number of necessary edge routers in each of the six regions, current Internet usage statistics from³ have been used. We estimate that in every region, capabilities of edge routers are sufficient to connect 10 % of the total Internet users in parallel at their maximum bandwidth. Based on the data available as of August 2007, we obtain edge router numbers of 43,700 for Asia, 32,200 for Europe, 23,300 for North America, 11,000 for South America, 3,400 for Africa and 1,900 for Oceania.

The generation method that has been described in section 3.3 is applied to each of the six regions using the above mentioned numbers of edge routers that are incrementally added to each region. Core routers within each region are interconnected to a random mesh with a fixed number of links. An example for links being generated based on a global RTA are those existing between core routers of different IXP peering networks in different regions. These are interconnected to a random mesh network based on real-world continent adjacency. Figure 2 shows a simple example graph consisting of two region graphs, each having its own IXP peering network and three AS networks connected to it.

3.5 Graph Analysis

An important task in the process of defining topology modules is to constantly evaluate generated graphs by comparing them to real-world data. For this purpose, graphs defined in text-based adjacency files (as e.g. used for data resulting from the Skitter experiment [9]) can be imported and compared in terms of vertex cover, clustering coefficient and diameter. In order to reflect newer research on scale-free graphs, Top Gen can also compute the relative likelihood [16] as well as assortativity [20] of graphs. The probability distribution function (*pdf*), cumulative distribution function (*cdf*) as well as the complementary *ccdf* of the vertex degree distribution can be displayed. An integrated regression component can be used to perform linear, exponential and power-law regression as well as extract Gaussian and power-law coefficients of a given distribution. By this means the power-law coefficients of two different graphs can be compared by the press of a button.

³<http://www.internetworldstats.com/stats.htm>

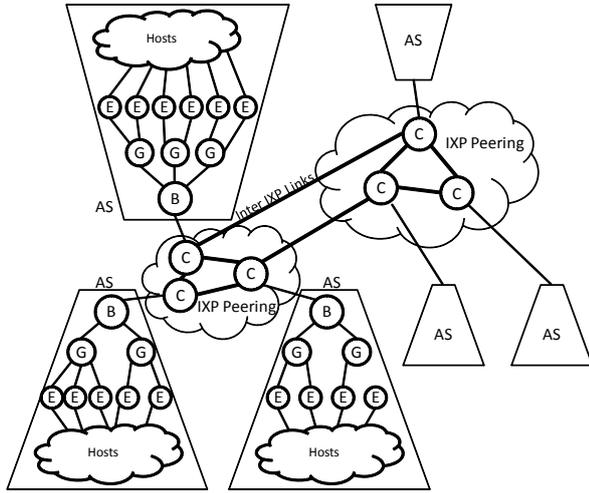


Figure 2: Sample Internet topology consisting of two regions

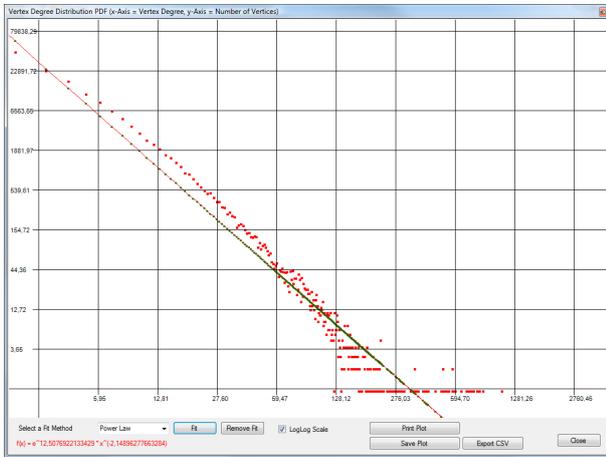


Figure 3: TopGen's integrated regression functionality

4. EVALUATION OF INTERNET TOPOLOGIES GENERATED BY TOPGEN

Although the generation methodology described in section 3.3 resembles the way how the actual Internet grows, there obviously is the need to validate the soundness of Internet topologies generated by TopGen's Internet topology module. For this a router-level graph reconstructed based on the measurements performed in the skitter experiment [9] [24] has been compared to a TopGen generated graph. Furthermore as an example for a graph generated using a simple random graph model, a BA-model graph of comparable size has been created using TopGen's BA topology module. We use this as a (possibly oversimplified) placeholder for graph generators that are based on preferential attachment. A comparison of vertex degree pdf, ccdf and power-law coefficients of the ccdf fit curves of all three graphs can be seen in Figure 4 and 5 as well as in Table 2. Although generation methodologies are completely different, Internet- and BA-model generated graphs as well as the CAIDA graph look

similar from a vertex degree distribution perspective. As seen in Figure 5, the vertex degree ccdf of the Internet-model graph is less regular than that exhibited by the CAIDA dataset or the simple BA-model. The reason for this can most likely be found in the so far rather limited resolution of router classes and regions. An interesting aspect is the exponential cutoff for high vertex degrees (see Figure 5) that can be observed for the synthetic Internet topology graph's vertex degree ccdf as well as for that of CAIDA. This is likely to be caused by the fact, that the probability that a router has a very high link number decreases exponentially.

Tables 1, 2 and 3 compare the three graphs mentioned above in various respects. The size of the graphs that have been used is shown in Table 1. Assuming a power-law ccdf $prob(d \geq x) = e^{-\lambda} \cdot x^{-\tau}$, Table 2 compares parameters τ and λ as well as the determination value R^2 of the vertex degree ccdf that have been measured for the three graphs. The graph generated by TopGen's Internet-model is remarkably close to the CAIDA dataset in respect to power-law coefficients. Finally, Table 3 compares average vertex degree $\frac{\sum_{v \in V} deg(v)}{\#V}$, clustering coefficient as well as assortativity [20] of the three graphs. Since the computation of the vertex cover is known to be NP-complete and we are addressing graphs with large vertex numbers, we have used a heuristic approach in order to determine a near-optimal upper as well as a lower bound.

The Internet's router network does not emerge randomly but has rather been designed to be fault-tolerant and optimize throughput. Looking e.g. at its structure at the very core, one finds a redundant mesh network of high-performance routers with small degree but high link capacity. Highest-degree routers are most probably found at the network's edge. As opposed to designed networks, graphs emerging from random power-law models often exhibit a hub-like core, highest degree vertices being placed right in its center while low-degree vertices are mostly found at the network's edges. This is generally not desirable for technical networks from a fault tolerance perspective. In order to capture structural differences that are invisible if solely comparing vertex degree distributions we have used the assortativity metric as defined by [20]. It can be used to differentiate between graphs having the same vertex degree distribution but showing different structural qualities. Graph generators focusing on creating graphs that provide a certain vertex degree distribution commonly fail to generate graphs being similar to real-world graphs in respect to assortativity [17]. It can roughly be interpreted as the tendency of a vertex to connect to vertices having a similar degree [16], $r < 0$ meaning a graph is disassortative and $r = 1$ representing completely assortative graphs. The assortativity values that have been measured for the three graphs being considered in this section are shown in table 3. The assortativity of BA and ER model graphs is known to converge to 0 as the number of vertices is increased [19]. Assortativity values for the CAIDA dataset as well as for a graph generated by TopGen's Internet module are remarkably close with values of 0.015 and 0.011 respectively.

Another metric that may be used to validate the soundness of TopGen's Internet topology is the composition of router types in different simulated regions. Unfortunately this kind of information is not available in any real-world dataset we are aware of and can therefore not be used for evaluation. When such data is made available in future for

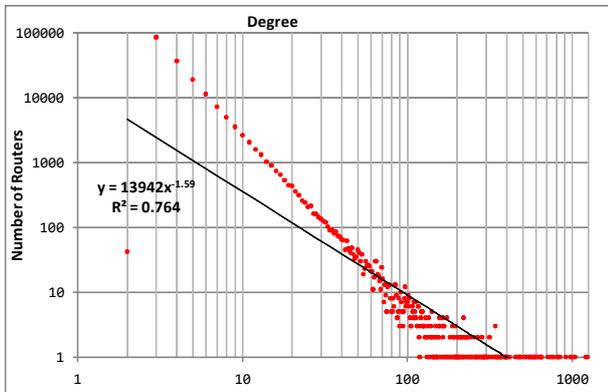
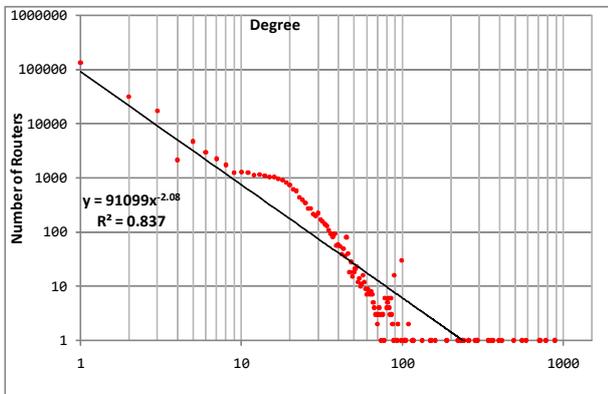
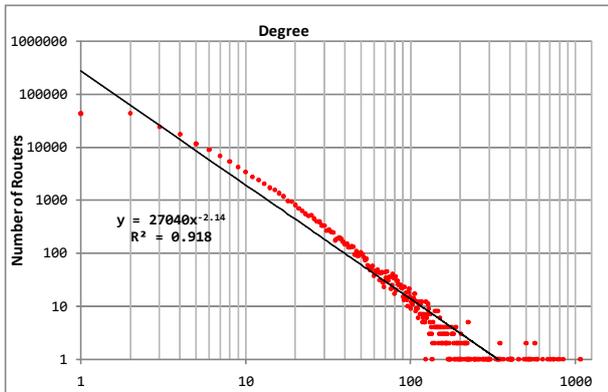


Figure 4: PDF of Vertex Degree (Solid line represents a power-law fit): CAIDA router-level graph (top), TopGen with Internet model (middle), TopGen with simple BA model (bottom)

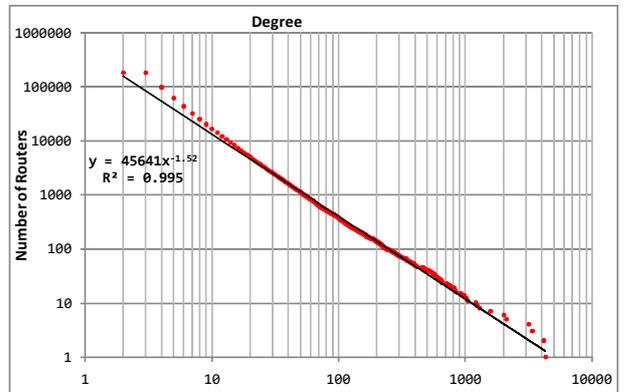
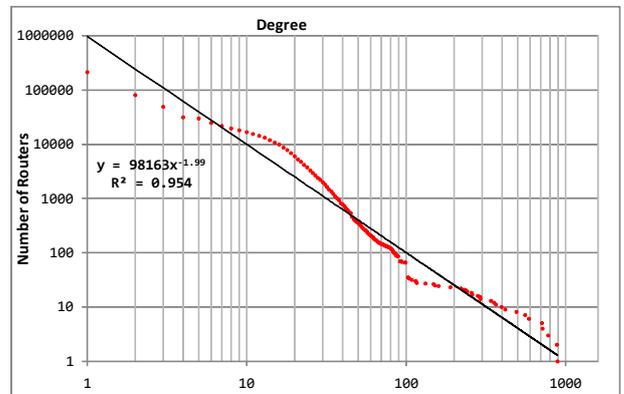
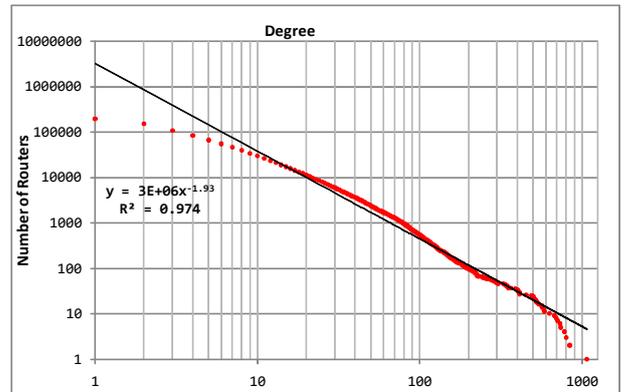


Figure 5: CCDF of Vertex Degree (Solid line represents a power-law fit): CAIDA router-level graph (top), TopGen with Internet model (middle), TopGen with simple BA model (bottom)

Table 1: Comparison of Graph Size

Graph	#V	#E	Avg Vertex Degree
CAIDA	192,244	609,066	6.34
TopGen (Internet)	210,487	631,223	3.18
TopGen (BA)	185,000	554,996	6.00

Table 2: Power-Law Coefficients (CCDF)

Graph	τ	λ	R^2
CAIDA	1.93	14.975	0.974
TopGen (Internet)	1.99	13.800	0.954
TopGen (BA)	1.53	13.031	0.995

real-world networks or for a certain AS, one might consider it as an additional evaluation criteria.

5. CONCLUSION AND FUTURE WORK

Considering the amount of feedback that has been received on a very simple and early version of TopGen that had been published on the private website of one of ours⁴, we presume that there is a considerable demand for a modern and modular topology generation platform within the network research community. Apart from its extensibility and ease-of-use, TopGen also provides a set of predefined graph generation models which might prove sufficient for many researchers and which can be used right away. Due to the set of metrics that can be calculated by the click of a button, TopGen may also be useful for members of the graph theory community, especially those interested in random graphs. TopGen’s multithreaded approach to region graph generation can be utilized to benefit from the current surge in multicore architectures. Bigger graphs can be generated in a short time by using a high regional resolution. More detailed information on how to use TopGen and how to implement custom topology generation modules are available as a technical report [22]. The C# source code of TopGen as well as a binary version and the parameter file that has been used to generate the graphs presented in this paper can be found on⁴. Using the free and Open Source Mono⁵ runtime, TopGen as well as its source code can be used on a number of architectures and operating systems. In order to provide interoperability with simulation environments and facilitate the comparison of TopGen generated graphs with that produced by other topology generators, we are in the process of evaluating whether to use an XML-based output format for generated graphs. Finally, a graphical editor for router classes and the definition of regions is being implemented and will complement the generator. Using this editor, regions and regional router compositions can be defined by drawing on a world map and real-world geographical distances between routers in different regions can be calculated.

As stated in [8], a reason why a pure BA model is unlikely to explain real-world graphs is that it is solely based on the factors preferential attachment and network growth,

⁴<http://syssoft.uni-trier.de/~scholtes/>

⁵<http://www.mono-project.com>

Table 3: Comparison of Graph Theoretic Metrics

Graph	Min/Max Vertex Cover	Clustering Coefficient	Assortativity
CAIDA	0.32/0.63	0.0287	0.015
TopGen (Internet)	0.20/0.39	0.0138	0.011
TopGen (BA)	0.34/0.68	0.0040	-0.002

lacking flexibility in graph generation. Incorporating preferential attachment, incremental growth and hierarchical graph generation, the main contribution of TopGen’s Internet topology model is that it combines those aspects with low-resolution geographic information and the technological constraints of routers. By this means, randomness (being limited to a router-specific narrow spectrum) is repelled in favor of a structured engineering of networks. TopGen also adds router- and link-specific annotations like bandwidth, region or router-type to topology graphs. These can be used e.g. by simulation environments to attach simulated agents only to a certain region or router type (like e.g. edge routers for clients).

As shown in the course of this paper, topology graphs generated by TopGen’s Internet topology module are very similar to available datasets of the Internet’s router-level graph in respect to several common graph theoretical properties that are traditionally used to evaluate the quality of topology generators. Existing generators commonly focus on creating graphs that are similar in respect to a certain metric (e.g. degree distribution) and therefore mostly fail when considering metrics [17] for which they have not been optimized. Our main contribution is the provision of a generation methodology that explicitly respects technical constraints of routers and fundamental principles of Internet connectivity rather than effectuating e.g. a certain degree distribution. It is encouraging to see that generated graphs show many aspects of real-world graphs (degree distribution, power-law coefficients, exponential cutoff in ccdf, small diameter, clustering coefficient, assortativity) although generation is based on mere technical considerations. It remains to investigate how the differences in vertex cover and average vertex degree between the TopGen Internet graph and the CAIDA graph can be compensated by refining model parameters.

The lack of an evaluation of end-to-end delay data based on a generated router graph certainly is a shortcoming of this publication. It has been omitted for time-constraints and must be considered an open issue for future work. For the same reasons as pictured above we are confident that the distribution of latency values between endpoints is realistic. A future step will be the comparison of such data with those generated by delay synthesizers like DS^2 [30] or King [13], structural generators like Tiers [10] and GridG [7] as well as available real-world data. Since the assortativity metric used in 4 is known to have some flaws when using it for high variability vertex degree distributions [15], we also implemented the relative likelihood metric proposed by [16]. We could not use it for our targeted graph sizes (>180,000 vertices) as it involves quadratic computational and space complexity. Two other metrics that haven’t been used for

the same reason are betweenness and diameter. Both involve the computation of all-pairs shortest paths and are therefore too costly to survey for our targeted graph sizes. The model parameters used to generate the router-level graph presented in this paper do not yet utilize TopGen's capability of using different router type and interconnectivity settings for different regions. When comparing e.g. geographically dispersed regions like Oceania with Europe, we think that it will be useful to consider such aspects in future refinements in order to further improve realism.

Although one cannot expect the Internet topology module presented in this paper to provide realism in every respect, we believe it is a promising first step towards first-principles topology generation as demanded by the authors of [16]. As there has not yet been spent much effort on refining model parameters, it should rather be seen as a basis for future refinement. Using e.g. a higher resolution for space and router classes is likely to result in router graphs that are more realistic. Also a more realistic model of how Internet providers and IXP carriers scale their infrastructures is required. We encourage the network research, topology generation and simulation community to adjust our assumptions, further enhance model parameters and generate small-scale, high-resolution topologies which can then be compared to available real-world data. We also invite researchers to download the TopGen software, the generated router-level graph as well as the parameter file with which it has been generated in order to perform further evaluations and parameter adjustments.

6. REFERENCES

- [1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180, New York, NY, USA, 2000. ACM Press.
- [2] R. Albert and A.-L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 72(1):48–97, 2002.
- [3] D. Alderson, L. Li, W. Willinger, and J. C. Doyle. Understanding internet topology: principles, models, and validation. *IEEE/ACM Trans. Netw.*, 13(6):1205–1218, 2005.
- [4] H. Berkowitz, E. Davies, S. Hares, P. Krishnaswamy, and M. Lepp. Terminology for Benchmarking BGP Device Convergence in the Control Plane. RFC 4098 (Informational), June 2005.
- [5] G. E. P. Box and M. E. Muller. A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, 1958.
- [6] K. Calvert, M. Doar, and E. Zegura. Modeling internet topology. *Communications Magazine, IEEE*, 35, June 1997.
- [7] H. Casanova, A. Chien, and Y.-S. Kee. Realistic modeling and synthesis of resources for computational grids. *Supercomputing, 2004. Proceedings of the ACM/IEEE SC2004 Conference*, 2004.
- [8] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006.
- [9] K. Claffy, B. Huffaker, D. Moore, and D. Plummer. Topology discovery by active probing. *Symposium on Applications and the Internet (SAINT) Workshops. Proceedings.*, pages 90–96, 2002.
- [10] M. Doar. A better model for generating test networks. *Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communications: The Key to Global Prosperity*, 1996.
- [11] P. Erdős and A. Renyi. On random graphs i. In *Publicationes Mathematicae 6*, pages 290–297, 1959.
- [12] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251–262, New York, NY, USA, 1999. ACM Press.
- [13] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary internet end hosts. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pages 5–18, New York, NY, USA, 2002. ACM Press.
- [14] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930 (Best Current Practice), Mar. 1996.
- [15] L. Li. *Topologies of Complex Networks: Functions and Structures*. PhD thesis, California Institute of Technology, 2007.
- [16] L. Li, D. Alderson, W. Willinger, and J. Doyle. A first-principles approach to understanding the internet's router-level topology. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–14, New York, NY, USA, 2004. ACM Press.
- [17] P. Mahadevan, C. Hubble, D. Krioukov, B. Huffaker, and A. Vahdat. Orbis: rescaling degree correlations to generate annotated internet topologies. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 325–336, New York, NY, USA, 2007. ACM Press.
- [18] A. Medina, I. Matta, and J. Byers. On the origin of power laws in internet topologies. *SIGCOMM Comput. Commun. Rev.*, 30(2):18–28, 2000.
- [19] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89, 2002.
- [20] M. E. J. Newman. Mixing patterns in networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 67, 2003.
- [21] E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67, 2003.
- [22] I. Scholtes. Router-level topology generation using TopGen. Technical report, System Software and Distributed Systems Working Group, University of Trier, March 2008. available at <http://syssoft.uni-trier.de/~scholtes>.
- [23] Y. Shavitt and E. Shir. Dimes. *ACM SIGCOMM Computer Communication Review*, 35, 2005.
- [24] Skitter router level topology datasets. <http://www.caida.org/tools/measurement/skitter/>.
- [25] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: degree-based vs. structural. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 147–159, New York, NY, USA, 2002. ACM Press.
- [26] A. Varga and G. Pongor. Flexible Topology Description Language for Simulation Programs. In *Proceedings of the 9th European Simulation Symposium (ESS'97)*, pages 225–229, 1992.
- [27] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature.*, 393, 1998.
- [28] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.
- [29] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report CSE-TR-456-02, University of Michigan, Ann Arbor, MS, 2002.
- [30] B. Zhang, T. S. E. Ng, A. Nandi, R. Riedi, P. Druschel, and G. Wang. Measurement based analysis, modeling, and synthesis of the internet delay space. In *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*, pages 85–98, New York, NY, USA, 2006. ACM Press.