Modeling the Routing of an Autonomous System with C-BGP

Bruno Quoitin and Steve Uhlig, University of Louvain-la-Neuve

Abstract

Today, the complexity of ISPs' networks make it difficult to investigate the implications of internal or external changes on the distribution of traffic across their network. In this article we explain the complexity of building models of large ISPs' networks. We describe the various aspects important to understanding the routing inside an AS. We present an open source routing solver, C-BGP, that eases the investigation of changes in the routing or topology of large networks. We illustrate how to build a model of an ISP on a real transit network and apply the model on two "what-if" scenarios. The first scenario studies the impact of changes in the Internet connectivity of a transit network. The second investigates the impact of failures in its internal topology.

The Internet consists of a collection of more than 18,000 domains called *autonomous systems* (ASes). Each AS is composed of multiple networks operated under the same authority. An AS can be an enterprise's network, an Internet service provider (ISP), or a campus network. Inside a single domain, an independent interior gateway protocol (IGP) [1] such as Intermediate System to Intermediate System (IS-IS) or Open Shortest Path First (OSPF) is used to propagate routing information. Between ASs, an exterior gateway protocol (EGP) is used to exchange reachability information. Today, Border Gateway Protocol (BGP) [1] is the de facto standard interdomain routing protocol used in the Internet.

Until recently, the main service provided by ISPs was best effort. Today, customers are asking for guaranteed performance and reliability. Now widely deployed services such as virtual private networks (VPNs) or voice over IP (VoIP) require increased performance guarantees from the network. For this reason, ISPs are very sensitive to the resilience and performance of their networks. They try to provide quality assurance to their customers through service level agreements (SLAs). Therefore, ISPs seek to build networks that will accommodate varying traffic loads, and be robust to link and router failures. To satisfy the tight constraints of the SLAs, ISPs engineer their networks to ensure the best performance, by, say, minimizing the delay across the network or preventing congestion from occurring on access links.

An AS is composed of a collection of routers that are interconnected. These routers are usually interconnected using multiple synchronous optical network/synchronous digital hierarchy (SONET/SDH) links and/or Ethernet. An example AS is represented in Fig. 1. We distinguish *core links* that interconnect the routers within an AS and *edge links* that cross AS boundaries. Since edge links connect to routers lying outside of its network, an AS only manages one side of these links. The routers where edge links are established are called the AS's *border routers*. The locations of these border routers are usually called the *points of presence* (POPs) of the AS.

Through edge links, the AS is connected to different kinds of neighbor ASs [2]. The *access links* connect to customer networks. Customer networks buy Internet connectivity from the AS. The *peering links* connect to transit providers and private peers. Most ASes buy Internet connectivity from one or several transit providers. With private peers, the cost of a peering link is shared with the peer. In addition, a private peering link is only used to exchange traffic with the peer and its own customers [2]. No transit traffic will flow through private peering links. Usually, private peerings are established at *interconnection points* (IXPs). An IXP is a collocation crafted with networking equipment where participating ASes can connect to each other.

The physical topology of an AS defines feasible paths that can be used to cross the network. How traffic actually crosses the network depends on the choices made by the routing protocols. These choices depend on two major factors: the diversity of available routes and router configurations. The diversity of the available routes known by an AS depends on the routing information received from neighboring ASs. Among these available routes, the routing protocols choose which one will be used to reach each destination. This choice depends on the goals of the network operators expressed in the router configuration [3].

To date, no modeling tool fully captures both the diversity of the routes announced by the neighboring domains and the details of the routing configuration inside an AS. References [4, 5] described tools that were able to model partial aspects of the routing of an AS. On the side of commercial products, it is unclear whether tools really take into account the BGP information and how they do it. The Cariden traffic engineering tool does not take the interdomain area into account. Product information from WANDL and OPNET claims that

0890-8044/05/$20.00 © 2005 IEEE

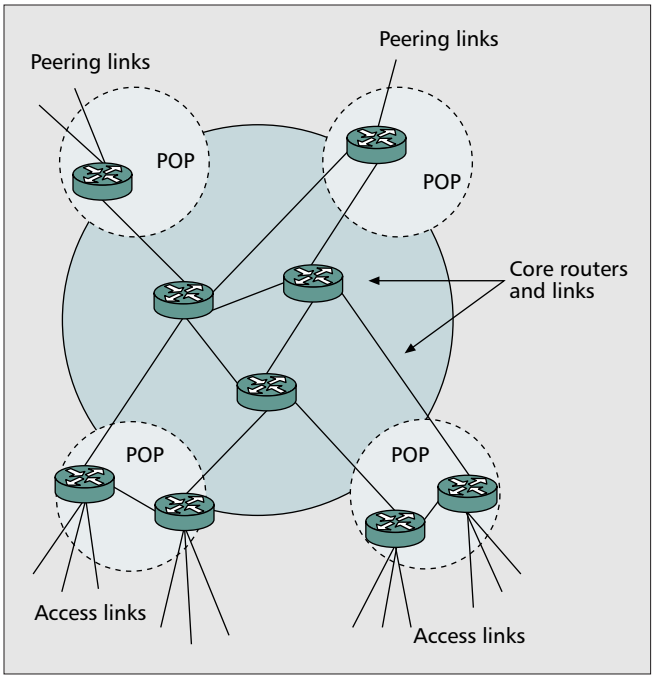


Figure 1. *Topology of an example autonomous system.*

both take BGP into account in some way. However, we were unable to check these claims.

As the remainder of this article shows, understanding the routing of large ASs requires not only modeling the routing inside the AS, but also taking into account routing information received from neighbor ASs. We explain how routing in an AS works. We describe how to model the routing of an ISP's network. We explain what information is required in order to build such a model. We show how this information is processed by an open source tool we developed. Finally, we provide two applications of our tool to study the behavior of a transit AS.

Routing in an Autonomous System

On its IP-level topology, an AS runs two different routing protocols. First, it runs an IGP such as OSPF or IS-IS in order to compute the interior paths from any AS's router toward the AS's other routers and subnets. The IGP is typically a link state protocol; that is, it floods information about the state of the adjacencies between all routers in the whole AS. The objective of intradomain routing is to find the shortest paths according to a selected metric. ISPs usually use a metric that is proportional to the propagation delay along the path or the bandwidth. Many network operators use the Cisco default metric, which is one over the bandwidth [1]. Some large ASs use a hierarchical IGP, where the AS is divided into different areas. Inside an area all the adjacency information is flooded. Between areas only aggregated information is exchanged.

In addition to the IGP, an AS sometimes uses static routing. Static routes are often used on edge links since routers on both side of these links are not operated by the same authority. Static routes are also used to set up access to small customers that do not use BGP.

Finally, an AS runs BGP [1]. BGP is responsible for the selection of the interdomain paths. It selects the paths toward networks outside the AS. The rationale behind the design of BGP was to provide reachability among domains and the ability for any domain to enforce its own routing policies (i.e., controlling what traffic enters and leaves the domain, and where). In contrast to the intradomain routing protocol, BGP does not optimize a global metric but relies on a *decision process* composed of a sequence of rules.

BGP routers exchange routing information by means of BGP sessions. Each BGP session is established between a pair of routers over a TCP connection. External BGP (eBGP) sessions are established over edge links, while internal BGP (iBGP) sessions are established between the routers of the AS. There is a full mesh of iBGP sessions between the routers of the AS. In some ASs the number of iBGP sessions can be quite large. For this reason, these ASs sometimes deploy route reflectors [6] in their network. Route reflectors are special BGP routers that make possible a hierarchy of iBGP sessions, thereby reducing the number of iBGP sessions. It is also possible to reduce the number of iBGP sessions by using BGP confederations [1].

Through its BGP sessions, each router receives BGP routes toward destination prefixes. Each router uses its decision process on a per-prefix basis to select the routes it will use. The BGP decision process is a sequence of rules that takes a set of routes toward the same destination prefix and selects a single route, called the *best route*, toward this prefix. This route will be installed into the router's routing information base (RIB), copied in the forwarding table, and eventually used to forward packets. Basically, the BGP decision process ranks routes according to their attributes. Each rule of the decision process discards the routes it does not prefer. The surviving routes are then submitted to the next rule, until a single route remains.

The BGP decision process considers several of the BGP route's attributes. The first attribute is the *local-pref*, which corresponds to a local ranking of the route. It is usually attached to the route upon reception by a border router and is never propagated outside the AS. The decision process prefers the routes with the highest local-pref attribute value.

The second attribute is the *as-path*. The as-path contains the sequence of ASs that the route crossed to reach the local AS. The as-path is used for two different purposes: avoiding routing loops and providing a distance metric in AS hops. The decision process prefers the routes with the shortest as-path.

The third attribute is the *multi-exit discriminator* (in short, the med). This attribute is used to rank routes received from the same neighbor AS. Usually, the med attribute is set by the neighbor AS to indicate the preferred peering link to use (e.g., based on the IGP cost in the neighbor). The decision process prefers the routes with the smallest value of the med.

Finally, the route contains a BGP *next-hop* attribute. This attribute indicates the IP address of the router to which the packets must be sent in order to reach their destination. The BGP next hop is often called the *egress* of the route. Note that the BGP next hop may be different from the immediate IP next hop. When a BGP router receives a route, it first checks that the next hop is reachable before considering it in the decision process. The decision process uses the IGP cost of the intradomain path toward the next hop to rank the routes. It prefers the routes with the smallest IGP distance to the next hop. This rule implements hot potato routing [7]. Its aim is to hand packets to a neighbor AS as soon as possible in order to consume as few network resources as possible. In addition, it automatically adapts routing to topology changes that affect the IGP distance to the egress points inside the AS. This step within the BGP decision process is where the IGP and BGP protocols interact.

To date, the only work in the literature to study the interaction between the IGP and BGP protocols is [7], which proposed an analytical model of the sensitivity of hot potato

changes on the BGP route selection process. The analytical formulation proposed in [7], however, reproduces neither the full BGP decision process nor the complexity of the working of BGP inside an AS [1].

Modeling an Autonomous System

Modeling an ISP is a task that includes several aspects, starting with understanding the AS's architecture, gathering network data, building a representation of the AS's network, and ending up with a tool that allows the model to be exploited.

Gathering Routers' Configuration

The first part toward building an AS's model consists of retrieving its configuration. The configuration of routers includes mapping between physical links and layer 3 links, the IGP metric associated with layer 3 links, the IGP hierarchy (areas), the BGP sessions, and the BGP policies enforced on each peering.

However, handling the routers' configuration in a large network is difficult. First, in a large IP network, the volume of information found in the routers' configurations is far too large for a human to be able to deal with manually. Second, the configurations of routers are usually found in separate files, and there are frequently inconsistencies between these files [3, 8]. Finally, the network may be based on heterogeneous equipment; thus, the configurations are written in different configuration languages. Sometimes, some options even depend on the version of the network equipment's operating system. There is therefore a need to automate the process of analyzing the network configuration and properly report inconsistencies.

Most of the time, discussion with the operator as well as cross-checking the files are required in order to exploit the network configuration.

Representing the Topology

The level of detail of the topology model will depend on what is intended to be studied. The topology model does not need to include physical/facility-level details in order to be able to accurately model how route selection is performed in the AS. Therefore, a model based on a graph of IP routers and layer 3 links is sufficient most of the time.

One difficulty that can be encountered in this part is mapping the nodes and edges of the model with the real networking equipment. Various IP addresses might be used to identify various parts of equipment. Routers might have multiple IP addresses corresponding to different physical interfaces and different loopback addresses. In certain configurations, the IP address used to identify the router in the IGP differs from the IP address that identifies the router in BGP. One solution to this consists of mapping all the addresses of one router to a single IP address. This must be done carefully since routing protocols may make decisions based on this address. This is so for BGP, for instance, where the IP addresses of the routers may be used to break ties in the last step of the decision process.

Routing Data

The third part of modeling an AS is to feed it routing data. Concerning intradomain routing, the routes may be computed based on the adjacencies found in the model's graph and on the IGP metric of the existing edges. For interdomain routing, additional information must be provided to the model. The AS's routers will perform route selection based on the external routes received through BGP by the border routers. These routes have to be captured and injected into the AS's model. To be able to perform very accurate predictions with the model, all the eBGP routes learned by the AS should be collected.

Collecting all the BGP routes learned by an AS is mainly an operational problem. Technically speaking, it is possible to capture all the BGP routes that are received on the peering links of the AS. It is also possible to log in on all the border routers and ask them to dump all their eBGP routes. In practice, however, due to current limitations in the routers' software and the reluctance of operators to perform these operations on production routers, collecting eBGP routes is not that simple. The technique used to collect BGP data depends on the AS's network, but one common technique is to rely on a dedicated workstation running a software implementation of BGP that has passive BGP sessions with the BGP routers of the AS. In large networks the time required to set up these BGP sessions between the workstation and routers is generally considered too large, so only a subset of the routes are collected. New approaches are currently being discussed in the Internet Engineering Task Force (IETF) [9]. It is important to notice that using a subset of the BGP routes may lead to inaccuracies in the model since possible egresses for some destination prefixes will be unknown.

Traffic Data

The fourth part required to model an AS concerns traffic. For an AS, traffic information raises serious problems [10]. In an intradomain model of the AS, only the router-router traffic matrix needs to be considered. This level of detail is sufficient since changes in intradomain routing will only change the paths from router to router, not the volume of data sent from one router to another. In this case one can rely on Simple Network Management Protocol (SNMP) measurements on the external interfaces of the AS and use techniques such as tomography to infer a router-router traffic matrix. The accuracy of these techniques is questionable [11]. Another technique that can be used in ASs where multiprotocol label switching (MPLS) is deployed is to collect per label switched path (LSP) statistics.

When considering a model of an AS that provides transit service, the router-router matrix is not sufficient. One must consider the prefix-prefix matrix since the egress router selected by an ingress router to reach a destination prefix may change, and the ingress router where the traffic from a prefix was received may also change [7]. The techniques described above are not applicable here since they do not provide information on the sources and destinations of the traffic flows.

One solution is to rely on Netflow statistics collected on the border routers. Collecting such statistics is still an issue today [10]. The problems faced by network operators are the following. First, the volume of a prefix-prefix matrix is significantly larger than a router-router matrix. The number of source and destination prefixes is on the order of 150,000. Also, activating Netflow puts an important burden on the border routers. Finally, setting up such a measurement infrastructure requires a significant investment in configuration time and equipment. Consequently, Netflow is usually only activated on the peering interfaces that carry a significant fraction of the traffic. In addition, Netflow sampling is also used in order to decrease the volume of the collected statistics.

C-BGP: A BGP Solver for Large ASs

We are not aware of the existence of any tool that fully captures the aspects described earlier. The most closely related works from the literature are [4, 5]. The aim of [4] was to provide the networking industry with a software system to sup-

port traffic measurement and network modeling. This tool is able to model intradomain routing and study the implications of local traffic changes, configuration, and routing. However, [4] does not model the interdomain routing protocol. Reference [5] proposed a BGP emulator that computes the outcome of the BGP route selection process for each router in a single AS. This tool does not model the flow of the BGP routes inside the AS, so it does not reproduce the route filtering process occurring within an AS. Neither of these tools is publicly available.

In this section we describe C-BGP, an open source routing solver we developed. C-BGP can be used by ISP network operators to study routing what-if scenarios based on routing information collected in their network. The solver takes several sources of information into account. First, it takes a description of the network topology at layer 3. Then it takes the configuration of all the routers present in the topology. This configuration describes the IGP weights of all the links, the BGP peerings of each router, and the BGP policies that must be enforced on each peering. We are able to parse Cisco and Juniper configuration files and generate configurations suitable for C-BGP. Finally, the tool takes the BGP routes learned by the ISP network on its border routers. As output, the solver computes for each router the routes selected toward all the interdomain prefixes. This output can then be used to replay how the traffic was routed by the routers of the AS.

In order to accurately model the routing in an ISP's network, we need to precisely model the path selection performed by the intradomain and interdomain routing protocols. That is, we must compute for each router the next hop that would have been selected to reach each destination prefix. Our solver models the topology of the network, the IGP, the eBGP and iBGP sessions, the iBGP hierarchy with route reflectors, the BGP route filtering, and the complete BGP decision process.

Modeling all aspects of BGP is time- and resource-consuming. To keep our model scalable and efficient, we do not model the time-consuming packet exchanges that occur between simulated routers in traditional discrete-event simulators such as SSFNet [12], J-Sim [13], or ns [14]. In addition, we do not model the TCP connections that support BGP sessions. We also do not model BGP timers such as the MRAI. We are therefore able to model large ISP networks.

Topology and IGP Models

In C-BGP we represent the network as a graph where nodes are routers, and edges are links between routers. Each edge is weighted by the IGP metric of the corresponding link. The network graph can be built in many different ways, such as manually building a representation of an existing network, extracting information from an IGP protocol trace captured in the network, or building a synthetic network.

The selection of paths by the intradomain routing protocol is modeled using the computation of the shortest paths based on the weight associated with each edge. In our model we do not simulate the details of the intradomain routing protocol such as the propagation of link state packets. We compute the shortest paths in the solver using Dijkstra's Shortest Path First (SPF) algorithm. These paths do not change until there is a weight change or link/router failure. The model we have implemented currently supports a single area, the most common type of IS-IS deployment in large ISP networks.

In addition to the paths selected by the intradomain routes, the solver also supports the addition of static routes. These routes are typically used for peerings with neighbor domains or to direct traffic toward customers. Static routes do not participate in the intradomain routing protocol.

Interdomain Routing Model

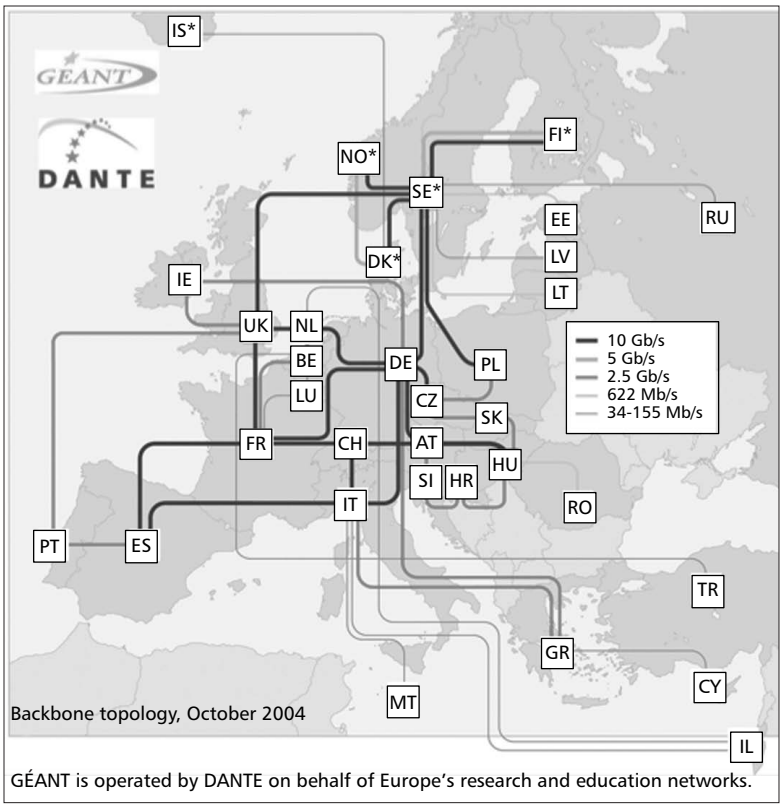
Our model for the interdomain routing protocol relies on the computation of the paths routers know once the BGP routing has converged [15]. For this purpose, we accurately reproduce the route selection performed by BGP in each router [1]. We also model the propagation of BGP messages across routers in a static way since we are not interested in the transient states of routing, only in its outcome. This is reasonable since the large majority of Internet routes are stable over time [16].

The message propagation model of C-BGP relies on a single global linear queue. This queue guarantees that the ordering of messages is kept on each BGP session. In addition, messages issued at a given time are delivered before messages issued at a later time. As opposed to discrete event simulators, the propagation of messages in C-BGP is deterministic. Any run will lead to the same outcome, while in discrete event simulators the outcome of the simulation may depend on the seed of the pseudo random number generator. This has an impact on the convergence of the simulations performed with C-BGP. When a BGP configuration has multiple stable solutions (e.g., see the DISAGREE case [15]), the simulation will not converge. With discrete event simulators, the simulation might converge to one of the solutions in a nondeterministic manner. In a BGP configuration without a stable solution (e.g., the bad-gadget [15]), the behavior of C-BGP will be the same as with discrete event simulators.

In order to model BGP, the nodes in the graph are considered as BGP routers and fitted out with additional data structures: a local RIB (Loc-RIB), adjacent RIBs (Adj-RIBs), and input and output filters. The Loc-RIB is used to store the best BGP routes, while the Adj-RIBs contain routes exchanged with neighbor routers. We distinguish Adj-RIB-in that contains routes received from the neighbor routers from Adj-RIB-out that contains routes announced to neighbor routers.

The model works as follows. Once the network topology is available and the intradomain routes have been computed, the solver begins the propagation of route advertisements. The solver starts with an arbitrary BGP router and advertises the routes known by the router. These routes have previously been captured on the eBGP sessions of the routers being modeled. The solver supports MRTd dumps or manual injection of routes. For each route to be advertised, the solver builds UPDATE messages and sends them to the router's neighbors according to the output filters. For each BGP message to send, the solver looks up in the router's routing table the link along which the message must be forwarded to reach the next hop. The message is forwarded on a hop-by-hop basis until it reaches its final destination. The generated BGP messages are pushed in a single global linear first-in first-out queue that guarantees the BGP messages are received in sequence. In real routers the BGP message ordering is guaranteed by the TCP connections underlying the BGP sessions. The solver does this for all the BGP routers.

The solver continues the simulation by popping the first message from the queue, and waking up the router corresponding to the current hop of the message. If the BGP message is a WITHDRAW, the router removes from the corresponding Adj-RIB-in the route toward the withdrawn prefix and runs the decision process. If the BGP message is an UPDATE, the router checks if the route it contains is accepted by its input filters. If so, the route is stored in the Adj-RIB-in, and the router's decision process is run. The decision process retrieves from the Adj-RIB-ins all reachable routes for the considered prefix, compares them, and selects the best one. The router then propagates its new best route to its neighbors according to its output filters. The propagation is



■ Figure 2. *An overview map of GÉANT.*

done by pushing new BGP messages on the global linear queue. The solver continues until the message queue is empty, which means that BGP has converged.

The Traffic Model

In our model the traffic information of an AS is a set of triples (ingress router, destination, traffic volume). Each triple represents the traffic volume received by an ingress router to be sent toward the destination. This destination does not need to lie within the AS. These triples can be computed from Netflow statistics collected in the AS on the border routers or generated from synthetic traffic. To replay the flow of traffic across an AS, we take each triple, one at a time. Then we perform a longest matching in the routing table computed by the BGP solver for the considered ingress router in order to find the prefix that contains the destination. We then use the route associated with this prefix to "forward" the traffic. We repeat this step on a hop-by-hop basis. Using this traffic model, we are able to evaluate the impact of various what-if scenarios on the distribution of the traffic inside the AS. For instance, based on the paths followed by the traffic flows, we can compute the load of the internal links as well as the load of the peering links of the AS.

Case Studies

In this section we present two case studies performed on the GÉANT network. The first investigates the addition or removal of peerings on the flow of the traffic. The second studies the routing impact of link failures. GÉANT is the pan-European research network and it is operated by Dante. It carries research traffic from the European national research and education networks (NRENs) connecting universities and research institutions.

GÉANT has POPs in all the European countries. All the routers of GÉANT are border routers. Figure 2 shows an overview of the GÉANT backbone.

Representing the Topology

GÉANT captures a trace of its IS-IS. We obtained the layer-three topology of GÉANT from a one-day IS-IS trace captured on 24 November, 2004. We cross-checked the obtained topology with a map of the network provided by Dante. We model GÉANT with a graph composed of 23 routers, 38 core links, and 53 edge links. All the POP names and peerings have been anonymized in the following case studies by request of Dante.

BGP Routing Data

In GÉANT, the BGP routes were collected using a dedicated workstation running GNU Zebra, a software implementation of BGP. The workstation has an iBGP session with 22 of the 23 border routers of the network. Using this technique, it was possible to collect all the best BGP routes selected by the border routers of the AS. We used a snapshot collected on November 24th, 2004 and obtained the 640,897 BGP routes propagated in the iBGP. Thus, we possess all the best routes currently selected by each router of GÉANT. This is a subset of all the eBGP routes learned by GÉANT on its 53 peerings since we do not know the eBGP routes currently not selected as best. Having only the eBGP routes currently selected as best may affect the results of the experiments in which an unknown eBGP route would have been selected instead of the current best one. Note that collecting all the eBGP routes received by GÉANT would have required the capture of BGP messages received on the 53 peering links.

In order to decrease the size of the model, we grouped the 150,071 destination prefixes advertised by eBGP using a technique similar to [5]. We ended up with 406 destination prefixes. Computing the BGP routes of GÉANT using C-BGP required only 68 s on an Intel P4 running at 2.66 GHz, and the memory footprint was only 69 Mbytes.

Traffic Data

To build an accurate model of the traffic, we obtained the Netflow statistics collected on all the edge links of the GÉANT network. In order to limit the volume of Netflow traces, a Netflow sampling rate of 1/1000 is used. This still generates on the order of 150 Gbytes of gzipped traces per month of traffic.

Optimized Peering

An important problem faced by ISPs is finding the optimal location of peering points. An ISP will search for new peering points in order to improve the efficiency of its interdomain traffic and/or decrease the cost of its peerings. Finding the optimal peering location is a nontrivial problem [17] that depends on both technical and economical factors [18]. In [17] the authors focused on the location of peerings with a single neighbor AS. With our tool, we can investigate the problem of modifying the interdomain connectivity of an AS since we take into account the BGP information. To our knowledge, existing approaches have not taken BGP into account.

In practice, an AS can choose to peer with many different ASes and at several locations. However, all the possible locations do not satisfy the ISP's requirements, and the ISP must decide which one best fits its goals. Let us take the example of a transit provider. Assume that its network is composed of n POPs. Now, suppose that the provider serves new customer ASes that connect to some POP x. These customers cause an

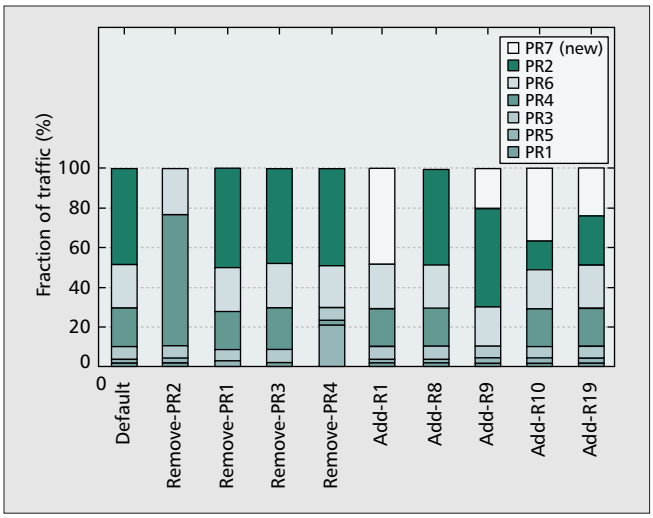


Figure 3. *Impact of addition/removal of peering on the distribution of the traffic among the peering links.*

increased amount of traffic to cross the topology before exiting at other POPs. To prevent the traffic to cross the whole network before reaching the egress points, the transit provider might prefer to add a peering close to the POP *x* that generates more traffic so that the traffic received by this POP exits the network as early as possible. It is thus expected that a given amount of traffic will exit the network through the new peering, but it is difficult to predict how much.

Adding a peering has the potential effect of modifying the best routes of the BGP router connected to this new peering. It is likely that this router will select routes learned through its new peering as best routes. It will then redistribute its new best routes to the other BGP routers through the iBGP sessions. If the latter BGP routers choose to use some of the routes learned through the new peering, it is possible that more traffic than originally planned will exit the network at the new peering. Models of an AS that do not consider BGP routing cannot predict the exact change in the traffic matrix in such a case.

With our modeling tool, we are able to predict what will happen to an AS's traffic when a new peering is added or removed. In order to compare the impact of the various scenarios, we use different metrics. First, we compute the distribution of the traffic over the peering links. Adding a new peering may attract some traffic from the already existing peering links and decrease the likeliness of congestion to occur on these links. Then we compute the IGP cost the traffic undergoes when the new peering is added. If this cost decreases for a significant number of ingress-egress pairs, it means the traffic follows intradomain paths that are shorter in terms of the IGP weights assigned by the ISP.

We performed this evaluation on the GÉANT network. Actually, Dante who operates GÉANT, is currently designing the next version of its network. GÉANT2 will have an increased number of customers, mainly in eastern Europe. GÉANT2 will provide transit to additional NRENs including the Russian NREN, JSCC, for instance. In this context it is important for Dante to know which locations will benefit from additional peerings. In our evaluation we consider the six most important peerings of the GÉANT network we call *PR1*, …, *PR6*. All these peerings use OC-48 links with a 2.4 Gb/s capacity. We study the addition and removal of peerings, and their impact on the traffic coming from all the GÉANT customers.

Figure 3 shows the impact of the removal or addition of a peering, in terms of the distribution of the outgoing traffic over the considered peerings of GÉANT. The x-axis of Fig. 3 gives the different scenarios we simulated. The one labeled *default* gives the distribution of the traffic if we leave the current peerings unchanged. Those labeled *remove-X* concern the scenarios where we removed the existing peering X. The others labeled *add-PRX* are the scenarios where a peering was added at POP *X*. The y-axis of Fig. 3 shows the distribution of the percentage of the total outgoing traffic carried by the considered peerings.

The default scenario on Fig. 3 shows that almost 50 percent of the traffic is carried by a single peering (PR2), and two other peerings each carry about 20 percent. The traffic is thus unevenly balanced over the considered peering links. Removing a peering does not change this uneven distribution. When peering *PR2* is removed, almost all the traffic exits at *PR4* and *PR6*. Removing *PR1* or *PR3* has little effect. Removing *PR4* shifts its traffic to *PR1*. Now let us consider the addition of peerings at POPs *R1*, *R8*, *R9*, *R10*, and *R19*. Adding a peering link at *R1* absorbs all the traffic that previously exited through *PR2*. The explanation is that most of the traffic sent through *PR2* was coming from eastern Europe. To reach *PR2*, these packets now have to pass through router *R1* and thus leave the GÉANT network there. If the purpose of adding this peering is to change the uneven distribution of the traffic among the peering links, adding a peering in *R1* does not help. The situation is similar when adding a peering at *R9*, as it absorbs most of the traffic that previously exited through *PR4*. Adding a peering at *R8* is worthless since the distribution of the traffic is left unchanged. Adding a peering in *R10* or *R19*, on the other hand, improves the balance of the traffic over the considered peering links.

Modifying the peerings of an AS not only changes the distribution of the traffic among the peerings, but also how traffic crosses the intradomain topology. Figure 4 shows the impact of adding or removing a peering on the IGP cost suffered by the traffic to cross the network. On the x-axis of Fig. 4, we show the difference between the IGP cost the traffic is subject to in the default situation and the one in each scenario. A positive difference means an improvement since the IGP cost has been decreased. A negative difference means a deterioration. On the y-axis of Fig. 4, we show the cumulative fraction of the traffic that perceives a change in the IGP cost.

We have seen in Fig. 3 that removing peerings *PR1* and *PR3* does not impact much the balance of the traffic on the peering links. However, we can see that removing *PR3* has an impact on the IGP cost seen by 5 percent of the traffic. The IGP cost for this traffic has been increased by 100. Another observation is that although removing the peering *PR2* has a significant impact on the distribution of the traffic on the peering links, it has a small on the IGP cost ($\Delta = -5$) seen by this traffic. The most interesting scenarios are the addition of a peering in *R10* or *R19* that improves both the IGP cost and the distribution of traffic among the peering links as shown above. If the purpose of adding a single peering link is to improve both the distribution of the traffic over the peering links without worsening the delay across the network, we know that the solution is to add a peering in either *R10* or *R19*.

Link and Router Failures

Evaluating the impact of link and router failures on the network is another nontrivial problem. In a large network, determining which link and router failures will change the outcome of the egress selection performed by BGP is a difficult problem [7]. This is important since routing changes can cause traffic shifts and lead to congestion. For an operator, it is important to check that the network will be able to accommodate the traffic load even in the case of single link or router

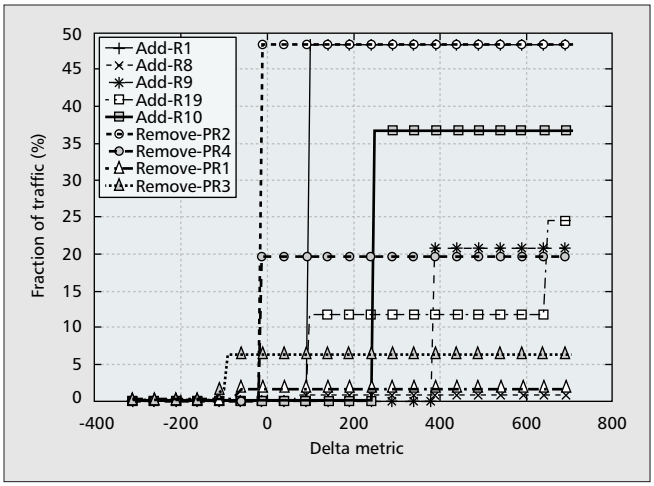


Figure 4. *Impact of addition/removal of peering on IGP cost seen by traffic.*

failures. If not, it is useful to identify which network links should be protected by the addition of parallel links, SONET-SDH protection, or the use of MPLS protection tunnels [19].

Our methodology for studying the impact of intradomain changes on path selection is as follows. First, we build a representation of the network inside the routing solver. We let the solver compute the routes in each router, then store a snapshot of the selected routes. This snapshot corresponds to the state of routing when everything is up and running. Then we remove the failing link or router and let the routing solver recompute the paths.

In order to provide a synthetic view of the impact of each failure, we partition the set of routing changes in four different classes: *peer change*, *egress change*, *intra cost change*, and *intra path change*. The peer change class corresponds to changes in the next-hop AS. If the next-hop AS has not changed but the egress router has, we speak of an egress change. When the egress is unchanged but the IGP cost of the ingress-egress path has changed, the routing change is classified as an intra cost change. Finally, if an ingress-egress path with the same IGP cost has been found, the routing change is put in the intra path change class. This can only occur if there are multiple equal cost paths between an ingress and an egress router in the network.

We simulated all the single-link failures in GÉANT and observed the impact on the BGP routes selected by each GÉANT router. We show our results in Fig. 5. On the x-axis, we show all the internal links of GEANT. On the y-axis, we show the number of routing changes accumulated on all the GÉANT routers. The routing changes are classified as peer, egress, intra cost, and intra path changes as explained above. The links on the x-axis are ordered according to the total number of routing changes caused by their failure. We observe that most of the time, a single-link failure causes many egress changes. Nearly 60 percent of the GÉANT links cause more than 100,000 routing changes when they fail.

The same method can be used to perform the single-router failure analysis or study the impact of changing the IGP cost of one link. Similar results to those found for the link failures have been obtained. For space limitation reasons, we do not present them in this article. We can also observe that in GÉANT, there are few pure intradomain reroutings. That is, there are few routing changes in the intra cost and intra path change classes. These results indicate that a pure intradomain model of the GÉANT network would not capture most of the routing changes that are due to occur under single link failures. This motivates the use of a routing model similar to the one advocated in this article.

Conclusion

In this article we describe the complexity of building a model of the routing of a large AS. We first explain the architecture of an AS and how routing works. Then we describe the essential factors that need to be taken into consideration when building a model of the routing of an AS. We describe C-BGP, an open source tool we developed, especially designed to let ISPs play with a model of their network. We illustrate the use of our tool through two different case studies. The first case study studied the impact on the traffic of a transit AS of changing its Internet connectivity. The second one investigated the impact of link failures on routing changes inside the AS. These two case studies have shown the importance of taking into account the interdomain routing information to understand the routing of a large AS.

As part of our ongoing work, we are currently applying the model presented herein on the network of a large transit AS. This AS contains hundreds of routers and has an iBGP hierarchy with multiple levels. We are also working on studying the interaction between multiple interconnected ASs. C-BGP can be used to compute the outcome of BGP route selection when there are multiple domains. However, we require knowledge of the structure and policies of the other domains. In order to study the impact of changes in one domain on its inbound traffic, for instance, we need to have knowledge of nearly all

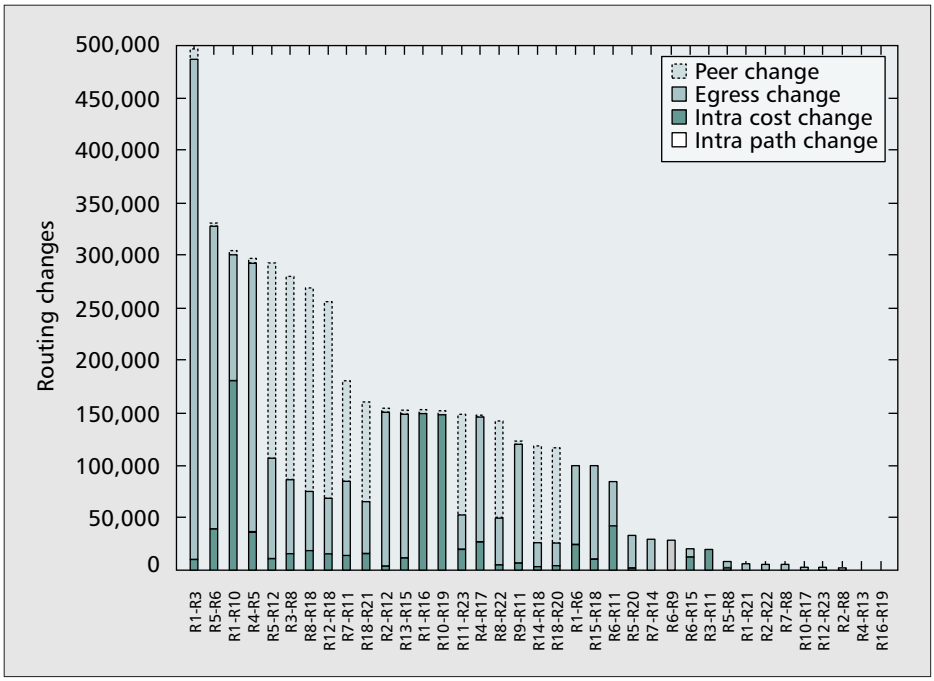


Figure 5. *Single link failure analysis: impact on BGP.*

the Internet domains. We are currently working on building a model of the Internet that can be used for this purpose.

In addition, we are still evolving our tool. The first improvement we are working on concerns a more accurate model of the IGP through support of multiple areas. The second improvement consists of operating the model on a continuous feed of topology, routing data, and traffic data. We believe that our approach to integrate topology, routing data, and traffic data can serve ISP operators to better understand the behavior of an AS and help them investigate improvements in the design of their network.

Acknowledgments

This work was supported by the Walloon Government within the WIST TOTEM project (http://totem.info.ucl.ac.be), the e-NEXT European Network of Excellence. Steve Uhlig is funded by the Belgian National Fund for Scientific Research (FNRS). We are grateful to Tim Griffin and Richard Gass from Intel Research and to Nicolas Simar from Dante for providing us with the GÉANT data. We also thank Olaf Maennel, Olivier Bonaventure, and Sebastien Tandel for their comments on this article.

References

[1] B. Halabi and D. Mc Pherson, *Internet Routing Architectures*, 2nd ed., Cisco Press, Jan. 2000.
[2] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE Global Internet*, Nov. 2000.
[3] A. Feldmann and J. Rexford, "IP Network Configuration for Intradomain Traffic Engineering," *IEEE Network*, Sept./Oct. 2001, pages 46–57.
[4] A. Feldmann *et al.*, "Netscope: Traffic Engineering for IP Networks," *IEEE Network*, Mar. 2000.
[5] N. Feamster, J. Winick, and J. Rexford, "A Model of BGP Routing for Network Engineering," *Proc. ACM SIGMETRICS*, June 2004.
[6] T. Bates, R. Chandra, and E. Chen, "BGP Route Reflection — An Alternative to Full Mesh IBGP," RFC 2796, Apr. 2000.
[7] R. Teixeira *et al.*, "Network Sensitivity to Hot Potato Disruptions," *Proc. ACM SIGCOMM*, Aug. 2004.
[8] N. Feamster and H. Balakrishnan, "Detecting BGP Configuration Faults with Static Analysis," *Proc. 2nd Symp. Networked Syst. Design and Implementation* May 2005.
[9] J. Scudder, "BGP Monitoring Protocol," Internet draft, draft-scudder-bmp-00.txt, Aug. 2005, work in progress.
[10] G. Varghese and C. Estan, "The Measurement Manifesto," *SIGCOMM Comp. Commun. Rev.*, vol. 34, no. 1, 2004, pp. 9–14.
[11] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic Matrix Estimation on a Large IP Backbone — A Comparison on Real Data," *Proc. ACM IMC*, Oct. 2004.
[12] B. J. Premore, "SSF Implementations of BGP-4," http://www.cs.dartmouth.edu/beej/bgp/, 2001.
[13] H. Tyan, "Design, Realization and Evaluation of a Component-Based Compositional Software Architecture for Network Simulation," Ph.D. thesis, Ohio State Univ., 2002.
[14] The Network Simulator ns-2, http://www.isi.edu/nsnam/ns
[15] T. Griffin and G. Wilfong, "An Analysis of BGP Convergence Properties," *Proc. ACM SIGCOMM*, Sept. 1999.
[16] J. Rexford *et al.*, "BGP Routing Stability of Popular Destinations," *Proc. ACM SIGCOMM Internet Measurement Wksp.*, Nov. 2002.
[17] D. O. Awduche, J. Agogbua, and J. McManus, "An Approach to Optimal Peering between Autonomous Systems in the Internet," *Proc. IEEE ICCN '98*, Oct. 1998.
[18] W. B. Norton, *The Art of Peering: The Peering Playbook*, preprint available from wbn@equinix.com, May 2002.
[19] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, and MPLS*, Morgan Kaufmann, 2004.

Biographies

BRUNO QUOITIN (bqu@info.ucl.ac.be) obtained his M.S. degree in computer science from the University of Namur, Belgium. He currently works as a researcher at the University of Louvain-la-Neuve, Belgium. His research interests include interdomain routing and traffic engineering.

STEVE UHLIG (suh@info.ucl.ac.be) obtained his M.S. degree in computer science from the University of Namur in June 2000 and his Ph.D. in applied sciences from the University of Louvain-la-Neuve in March 2004. His current position is chargé de recherches with the FNRS at the University of Louvain. His main research interests are related to the macroscopic aspects of the Internet, particularly understanding the relationships between the traffic characteristics and interdomain routing.