A critical view of the sensitivity of transit ASs to internal failures

Steve Uhlig* and Sébastien Tandel**

Computing Science and Engineering Department
Université catholique de Louvain, Belgium
E-mail:{suh,sta}@info.ucl.ac.be

Abstract. Recent work on hot-potato routing [1] has uncovered that large transit ASs can be sensitive to hot-potato disruptions. Designing a robust network is felt as overly important by transit providers as paths crossed by the traffic have both to be optimal and reliable. However, equipment failures and maintenance make this robustness non-trivial to achieve. To help understanding the robustness of large networks to internal failures, [2] proposed metrics aimed at capturing the sensitivity of ASs to internal failures. In this paper, we discuss the strengths and weaknesses of this approach to understand the robustness of the control plane of large networks, having carried this analysis on a large tier-1 ISP and smaller transit ASs. We argue that this sensitivity model is mainly useful for intradomain topology design, not for the design the whole routing plane of an AS. We claim that additional effort is required to understand the propagation of BGP routes inside large ASs. Complex iBGP structures, in particular route-reflection hierarchies [3], affect route diversity and optimality but it an unclear way.
Keywords: network design, sensitivity analysis, control and data planes, BGP, IGP.

1 Introduction

Designing robust networks is a complex problem. Network design consists of multiple, sometimes contradictory objectives. This problem has been fairly discussed in the literature, in particular [4,5]. Examples of desirable objectives during network design are minimizing the latency, dimensioning the links so as to accommodate the traffic demand without creating congestion, adding redundancy so that rerouting is possible in case of link or router failure and, finally, the network must be designed at the minimum cost. Recent papers have shown that large transit networks might be sensitive to internal failures. In [1], Teixeira et al. have shown that a large ISP network might be sensitive to hot-potato disruptions. [6] extended the results of [1] by showing that a large tier-1 network can undergo significant traffic shifts due to changes in the routing. To measure the sensitivity of a network to hot-potato disruptions, [2] has proposed a set of metrics

* Corresponding author. Steve Uhlig is "chargé de recherches" with the FNRS (Fonds National de la Recherche Scientifique, Belgium). This research was partly carried while Steve Uhlig was visiting Intel research Cambridge.
** Sébastien Tandel is funded by a grant from France Télécom.

that capture the sensitivity of both the control and the data planes to internal failures inside a network.

To understand why internal failures are critical in a large transit AS, it is necessary to understand how routing in a large AS works. Routing in an Autonomous System (AS) today relies on two different routing protocols. Inside an AS, the intradomain routing protocol (OSPF [7] or ISIS [8]) computes the shortest-path between any pair of routers inside the AS. Between ASs, the interdomain routing protocol (BGP [9]) is used to exchange reachability information. Based on both the BGP routes advertised by neighboring ASs and the internal shortest paths available to reach an exit point inside the network, BGP computes for each destination prefix the "best route" to reach this prefix. For this, BGP relies on a "decision process" [10] to choose its a single route called the "best route among several available ones. The "best route" can change for two reasons. Either the set of BGP routes available has changed, or the reachability of the next-hop of the route has changed due to a change in the IGP. In the first case, it is either because some routes were withdrawn by BGP itself, or that some BGP peering with a neighbor was lost by the router. In the second case, any change in the internal topology (links, nodes, weights) might trigger a change in the shortest path to reach the next hop of a BGP route. In this paper we consider only the changes that consist of the failure of a single node or link inside the AS, not routing changes related to the reachability of BGP prefixes.

Our experience with the metrics proposed in [2] provided insight into the sensitivity of the network to internal failures. However, having used this sensitivity analysis on a large tier-1 network also revealed weaknesses of this model to understand the routing plane of transit ASs. We discuss further work required to help operators to understand the control plane of their network, particularly the behavior of iBGP.

Section 2 first presents the methodology to model an AS. Section 3 then introduces the sensitivity model to internal failures. Section 4 discusses the limitations of the model for realistic iBGP structures and Section 5 concludes and discusses further work in the area.

2 Methodology

In this section, we describe our approach to build snapshots of real ISP networks. The main point of our relatively heavy methodology is to make the model as easy as possible to match with the context of real transit ASs. We do not make assumptions on the internal graph of the iBGP sessions, even though in the case of GEANT there is a iBGP full-mesh between all border routers. Most large transit ASs do rely on route-reflection, hence putting assumptions on the sensitivity model restricts its applicability. The route solver on which we rely, C-BGP [11], has no restriction on the structure of the iBGP sessions inside an AS. C-BGP has been designed to help the evaluation of changes to the design of the BGP routing inside an AS [12]. Changes to the routing policies of an AS, or the internal configuration of its iBGP sessions is easy with C-BGP.

In this section we describe the methodology we use to model a transit AS. A more detailed discussion on how to model an AS with C-BGP can be found in [12]. We explain in this section how we build snapshots of the routing and traffic matrix of large

ASs. The main steps of our methodology consist in producing snapshots of the routing inside a transit AS, consider that this view of the routing is valid for the whole bin between two time intervals, and based on the routing snapshot and the traffic statistics available to build a traffic demand for the considered time bin. Building more precise views of the routing and traffic matrices is possible by using small enough time intervals.

2.1 Related work

The most closely related works from the literature are [13] and [14]. The aim of [13] was to provide the networking industry with a software system to support traffic measurement and network modeling. This tool is able to model the intradomain routing and study the implications of local traffic changes, configuration and routing. [13] does not model the interdomain routing protocol though. [14] 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, hence it does not reproduce the route filtering process occurring within an AS. Finally, [12] discusses the problem of modeling the routing of an autonomous system and presents an open-source route solver aimed at allowing the study of networks with a large number of BGP routers.

2.2 Modeling the routing of an AS with C-BGP

To build snapshots of the routing inside a transit AS, we rely in this paper on C-BGP [11]. C-BGP is an open-source routing solver aimed at reproducing precisely BGP and making possible the modelling of networks containing a large number of BGP routers. The reason for choosing C-BGP instead of simply gathering BGP RIBs is that we aim at providing a tool that allows a network operator to build "what-if" scenarios, for instance by changing its topology or routing policies, and investigating their impact on the sensitivity of their network [12]. By relying on C-BGP, one can relatively simply change the internal network topology, the configuration of the routers, or even filter the set of BGP routes available inside the network.

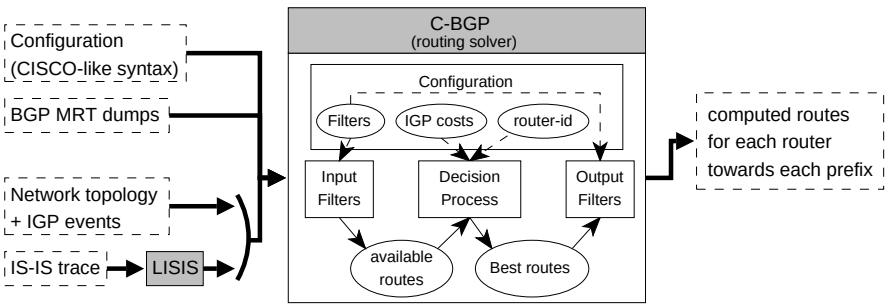

Fig. 1. C-BGP routing solver.

To model a transit network with C-BGP, we use the following steps. These steps reflect the typical way we see how modeling the routing of an AS would be done, even though on particular network instances these steps might have to be slightly changed, depending on the available routing data. First, we infer the topology of the network based on its ISIS data. We create the internal POP-level topology and for each time bin, we read all the routing messages received during the time bin and inject them into C-BGP. For ISIS, we apply all link state packets (LSPs) so that both the reachability and the IGP weights at the end of the time bin are consistent with them. Note that replaying the events at their exact time is useless in C-BGP as the simulator is not even-driven, i.e. it only propagates the routing messages and lets BGP converge inside the BGP router. When only a single solution towards which the real BGP converges, C-BGP will find the same solution. However, when several distinct solutions exist in the real BGP [15], then it is unclear how C-BGP will behave. Sometimes is will converge towards of the the solutions, sometimes it will never converge. The model used by C-BGP provides scalability since it prevents from having to care with the computationally consuming state machines and timers of the real routing protocols.

We first inject the static topology of the transit network into C-BGP. This topology contains the routers, links, and IGP weights of the links. The shortest IGP paths inside each AS are computed at each router and BGP sessions are established between the routers. The ISIS data of the two networks up to the beginning of the actual time the simulation is supposed to start are then read and the LSPs converted into changes of the C-BGP topology. The IGP changes are injected into C-BGP and the shortest paths are recomputed at all routers.

For BGP, we restrict the set of prefixes to those largest ones, in a similar way to [16,13,17]. We infer by parsing the BGP RIBs where in the network the prefixes are announced by external peers. Depending on whether the transit AS relies on "next-hop-self" in its routers or not, the visibility of the eBGP routes might be different inside the iBGP sessions of the AS. It is important to understand that relying on BGP messages collected through iBGP sessions as often done in the literature implies that one does not know neither all the messages advertised by the peer routers of the network (eBGP), nor the full diversity of the BGP routes known by the routers. With the current data available today, one can only infer the state of the best routes of some router at some point in time given that it takes some time for messages to propagate inside iBGP. While this might be acceptable in some situations, not having the same set of routes as is available inside the BGP routers implies that simulating the failure of a router inside the network that is used as egress point by another router cannot be accurately simulated since the latter router will choose another best BGP route that one does not know based on the available BGP routing data.

3 Network sensitivity to internal failures

In this section and the following, we describe the main points of our version of the model proposed in [2]. A more detailed presentation of this model can be found in [18].

Let $G = (V, E, w)$ be a graph, V the set of its vertices, E the set of its edges, w the weights of its edges. A graph transformation δ is a function $\delta : (V, E, w) \rightarrow$

(V', E', w') that deletes/adds a vertex or an edge from G. In this paper we consider only the graph transformations δ that consist in removing a single vertex or edge from the graph. In practice, an AS may want to consider more complex failures corresponding to failures that are occur together for instance, or to consider only the failures that have actually occurred in the network for the last few weeks or months. For consistency with [2], we denote the set of graph transformations of some class (router or link failures) by ΔG. The new graph obtained after applying the graph transformation δ on the graph G is denoted by $\delta(G)$. In this paper, we restricted the set of graph transformations as well as the definition of a graph compared to [2], as we do not consider the impact of changes in the IGP cost. Changes to the IGP cost occur rarely in real networks.

To perform the sensitivity analysis to graph transformations, one must first find out for each router how graph transformations impact the egress point it uses towards some destination prefix p. The set of considered prefixes is denoted by P. The BGP decision process $dp(v, p)$ is a function that takes as input the BGP routes known by router v to reach prefix p, and returns the egress point corresponding to the best BGP route. The *region index set* RIS of a vertex v records this egress point of the best route for each ingress router v and destination prefix p, given the state of the graph G: $RIS(G, v, p) = dp(v, p)$.

We introduced the state of the graph G in the *region index set* to capture the fact that changing the graph might change the best routes of the routers. In AS that consist of a full-mesh of iBGP sessions, the impact of a node or router failure on the best route used by a particular router is straightforward to find out. Removing a node or link changes the shortest paths to reach the egress points. Simulating the decision process of the router is enough to predict the best route after the graph transformation. In more complex networks on the other hand, the failure of a link or node not only changes the shortest paths towards the egress points, but new routes might also be advertised in replacement of a previously known route. In such a case, the BGP convergence inside the AS must be replayed to know the exact outcome of the BGP decision process.

The next step towards a sensitivity model is to compute for each graph transformation δ (link or router deletion), whether a router v will change its egress point towards destination prefix p. For each graph transformation δ, we recompute the all pairs shortest path between all routers after having applied δ, and record for each router v whether it has changed the egress point for its best BGP route towards prefix p. We denote the new graph after the graph transformation δ as $\delta(G)$. As BGP advertisements are made on a per-prefix basis, the best route for each (v, p) pair has to be recomputed for each graph transformation. It is the purpose of the *region shift function* H to record the changes in the egress point corresponding to the best BGP route of any (v, p) pair, after a graph transformation δ:

$$H(G, v, p, \delta) = \begin{cases} 1, \ if \ RIS(G, v, p) \neq RIS(\delta(G), v, p) \\ 0, \ otherwise \end{cases}$$

The *region shift function* H is the building block for the metrics that will capture the sensitivity of the network to the graph transformations.

To summarize how sensitive a router might be to a set of graph transformations, the *node sensitivity* η computes the average *region shift function* over all graph transforma-

tions of a given class (link or node failures), for each individual prefix p:

$$\eta(G, \Delta G, v, p) = \sum_{\delta \in \Delta G} H(G, v, p, \delta) \cdot Pr(\delta)$$

where $Pr(\delta)$ denotes the probability of the graph transformation δ. Note that we assume that all graph transformations within a class (router or link failures) are equally likely, i.e. $Pr(\delta) = \frac{1}{|\Delta G|}, \forall \delta \in \Delta G$, which is reasonable unless one provides a model for link and node failures. Further summarization can be done by averaging the *vertex sensitivity* over all vertices of the graph, for each class of graph transformation. This gives the *average vertex sensitivity* $\hat{\eta}$:

$$\hat{\eta}(G, \Delta G, p) = \frac{1}{|V|} \sum_{v \in V} \eta(G, \Delta G, v, p)$$

The *node sensitivity* is a router-centric concept that performs an average over all possible graph transformations, measuring how much a router will change its egress point for its best routes after the graph transformations on average. Another viewpoint is to look at each individual graph transformation δ and measure how it impacts all routers of the graph. The *impact of a graph transformation* θ is computed as the average over vertices of the *region shift function*:

$$\theta(G, p, \delta) = \frac{1}{|V|} \sum_{v \in V} H(G, v, p, \delta)$$

The *average impact* of a graph transformation $\hat{\theta}$ summarizes the information provided by the *impact* by averaging it over all graph transformations of a given class:

$$\hat{\theta}(G, \Delta G, p) = \sum_{\delta \in \Delta G} \theta(G, p, \delta) \cdot Pr(\delta)$$

4 Discussion

Our use of the sensitivity model sketched in section 3 on different networks has shown the interest and limitations of this model. The first thing to be noted is that the model is very "hot-potato centric". For networks that do not have a complex iBGP structure (no route-reflection), the model reveals the critical links and routers [18]. That is, the model captures the sensitivity due to the concentration of many internal paths that will change after a graph transformation. However, using the model for complex transit networks using route-reflection is more tricky. Contrary to the case of an AS with an iBGP full-mesh, route-reflection introduces opacity into the selection of the best BGP route by a router. In the case of an iBGP full-mesh, each ingress router, i.e. a router that is not itself an exit point inside the AS for the considered destination prefix, chooses as its best route the one that is the "best" one among all the routes advertised by the external peers of the AS. If the routing policies applied inside the AS are consistent among all internal routers, then for each ingress router there exists only one best BGP route that

will be chosen by this router, and the choice of this route will not depend on the best route choice of the other routers of the AS. In the case of route-reflection, the best route chosen by a router depends on the best route choice of the route reflectors on the BGP routes propagation path inside the AS.

When an iBGP full-mesh is used, it is relatively straightforward to predict the outcome of an internal change on the best route choice performed by BGP. Among several alternative routes having the same quality for BGP[1], the cost of the IGP path to reach the exit point inside the AS will be used to decide which BGP route will be considered as best. Hence in an iBGP full mesh, there is a direct relationship between the IGP shortest paths and the best route choice made by BGP. For instance, Figure 2 shows the internal topology of a simple transit AS relying on an iBGP full-mesh. Inside the iBGP, BGP routers only propagate to other iBGP peers routes they did not receive from an iBGP peer. We do not show all the iBGP sessions on Figure 2, but all routers have an iBGP session with every other router of the topology. We also consider a single destination prefix p. The transit AS has three ingress routers $i1$, $i2$, and $i3$, two egress routers $e1$ and $e2$, and a router located in the middle of the topology. All routers run both the IGP and BGP as in most ASs. The arrows on Figure 2 provide the choice of the best route by BGP towards p at each router of the AS. On Figure 2, two eBGP routes are learned to reach prefix p, one through egress router $e1$ and another through egress router $e2$. With an iBGP full-mesh, $i1$ and $i2$ will use the BGP route learned through $e1$, while $i3$ will use the BGP route learned through $e2$, both due to the IGP cost rule of the decision process. Each router hence relies for its best BGP route the smallest IGP cost path to exit the network.

Now suppose that our AS relies on route-reflection [3] as shown on Figure 3. The sole difference between Figure 2 and Figure 3 in terms of the routing configuration concerns the iBGP sessions. With route-reflection, all BGP routers are not directly connected anymore by an iBGP session, but all border routers are clients of the route-reflector RR. In this case, each routers knows the routes it learned from eBGP sessions, and a single route advertised by RR. The issue with route-reflection is that the best route chosen by RR depends on its own IGP cost to reach the exit point inside the AS, not the one of its clients. RR will choose as its best BGP route the one learned through $e1$ since its IGP cost is smaller than the one through $e2$. In that case, $i1$, $i2$ and $i3$ will all choose as their best route to reach p the route advertised by RR, and thus will use $e1$ as their exit point towards p. The IGP cost of the best route chosen by $i3$ is not optimal in terms of the IGP cost anymore compared to the iBGP full-mesh. Here we used a simple situation with a single route-reflector. In practice, large transit ASs can rely on a hierarchy of route-reflectors. Route-reflection trades-off the number of iBGP sessions inside the AS with a drastic reduction in the diversity of the routes known by the routers, and by a potentially suboptimal IGP cost of the best routes chosen by the routers.

A route reflector today chooses for all its clients routers a single best BGP route. The impact of this choice on the client routers is that the latter's will be sensitive to the graph transformations that affect the path of this best route chosen by their route reflector. Depending on how these client routers are located inside the AS, a particular

[1] Same value of local-pref, AS path length, MED, and route origin type.

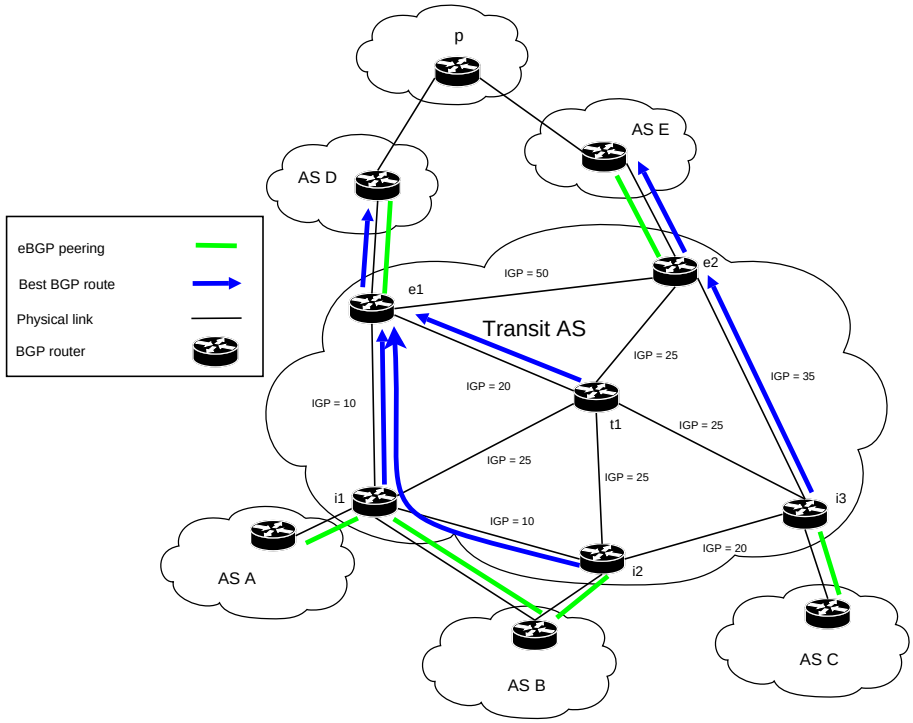

Fig. 2. Example topology with iBGP full-mesh.

subgraph of the AS will be used by the clients to reach the egress point of the BGP route advertised by the route reflector.

The implication for the network sensitivity is that the impact of the graph transformations and the node sensitivity will depend very much on how route-reflectors redistribute the routes they learn to their clients, and which routes they consider as best. If each route reflector was to compute for each of its client the latter would have selected in the case of an iBGP full-mesh and redistribute this route on a per-client basis as proposed in [19], then the sensitivity of the network in the two cases would be the same. However, if route reflectors are left to redistribute their best route without respect to how their clients would have chosen their best route in the iBGP full-mesh, then it is difficult to foresee how this will change the paths used to carry the traffic inside the AS since it depends both on the placement of the route reflectors, the interconnection structure of the iBGP sessions, the diversity of the BGP routes learned from peer ASs, and wherefrom the external routes are learned.

Our experience with a large tier-1 network have shown that the results of the sensitivity analysis between the actual network configuration and an iBGP full mesh are quantitatively much different but qualitatively very alike. However, understanding why differences in the sensitivity arise when the iBGP structure is changed is not answered by the sensitivity model of [2]. For that, one must first understand what filtering is

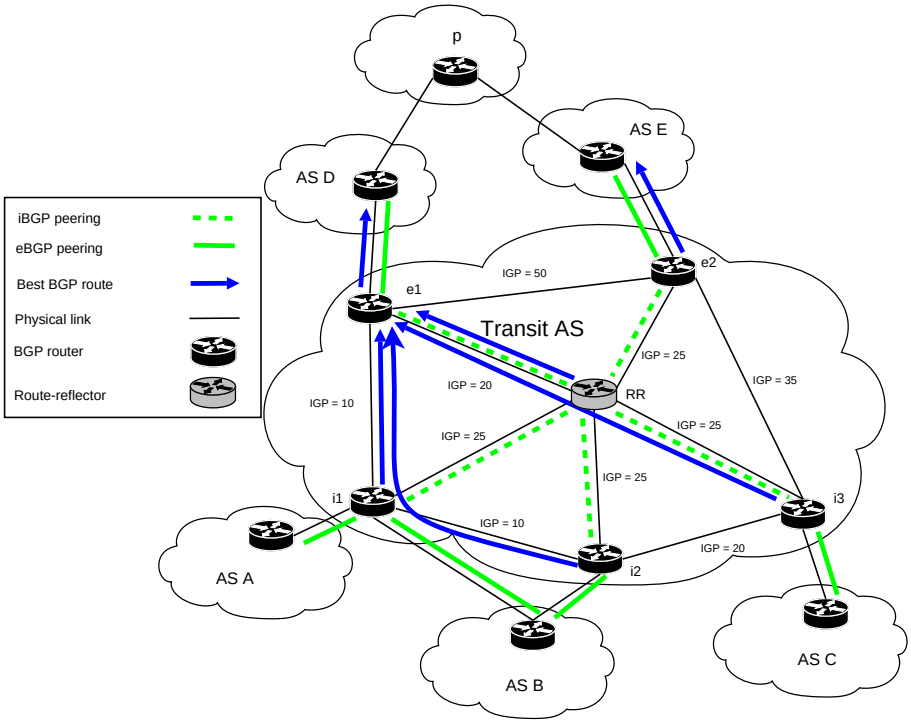

Fig. 3. Example topology with route reflection.

performed by a given iBGP structure compared to the iBGP full mesh. The reason to compare with an iBGP full mesh is that the best route visibility is achieved by the full mesh, while introducing route reflection reduces the diversity of the routes that can be chosen by the BGP routers inside the AS.

5 Conclusions and further work

In this paper we have presented our version of the sensitivity model to internal failures based on the work of [2]. We sketched our methodology to study the sensitivity of an AS based on this model and discussed the limitations of this model to provide insight into the behavior of the control plane of large transit ASs due to the presence of route reflectors.

The obvious further work we see is to study the actual impact of route reflection on the diversity and sensitivity of the routers inside a large transit AS. We are currently investigating this problem on a large tier-1 network. The lack of study in the literature concerning the actual impact of route reflection on the diversity and the filtering of the routes inside an AS indicates that our current understanding of iBGP is still very poor. We feel that a proper understanding of iBGP is particularly important for the design of robust networks.

Acknowledgments

This research was partially supported by the Walloon Government (DGTRE) within the TOTEM project [20]. We thank all the people from DANTE that helped in making the GEANT routing and traffic data available, and among them Nicolas Simar specifically. We would also like to thank Bruno Quoitin for developping C-BGP [11].

References

1. R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of hot-potato routing in IP networks," in *Proc. of ACM SIGMETRICS*, June 2004.
2. R. Teixeira, T. Griffin, G. Voelker, and A. Shaikh, "Network sensitivity to hot potato disruptions," in *Proc. of ACM SIGCOMM*, August 2004.
3. T. Bates, R. Chandra, and E. Chen, "BGP Route Reflection - An Alternative to Full Mesh IBGP," Internet Engineering Task Force, RFC2796, April 2000.
4. R. S. Cahn, *Wide Area Network Design: Concepts and Tools for Optimisation*, Morgan Kaufmann, 1998.
5. W. D. Grover, *Mesh-Based Survivable Networks*, Prentice Hall PTR, 2004.
6. R. Teixeira, N. Duffield, J. Rexford, and M. Roughan, "Traffic matrix reloaded: impact of routing changes," in *Proc. of PAM 2005*, March 2005.
7. J. Moy, *OSPF : anatomy of an Internet routing protocol*, Addison-Wesley, 1998.
8. D. Oran, "OSI IS-IS intra-domain routing protocol," Request for Comments 1142, Internet Engineering Task Force, Feb. 1990.
9. J. Stewart, *BGP4 : interdomain routing in the Internet*, Addison Wesley, 1999.
10. Cisco, "BGP best path selection algorithm," `http://www.cisco.com/warp/public/459/25.shtml`.
11. B. Quoitin, "C-BGP, an efficient BGP simulator," `http://cbgp.info.ucl.ac.be/`, September 2003.
12. B. Quoitin and S. Uhlig, "Modeling the routing of an Autonomous System with C-BGP," *IEEE Network Magazine*, November 2005.
13. Anja Feldmann, Albert Greenberg, Carsten Lund, Nick R eingold, and Jennifer Rexford, "NetScope: Traffic Engineering for IP Networks," *IEEE Network Magazine*, March 2000.
14. N. Feamster, J. Winick, and J. Rexford, "A model of BGP routing for network engineering," in *Proc. of ACM SIGMETRICS*, June 2004.
15. T. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *Proc. of ACM SIGCOMM*, September 1999.
16. A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: methodology and experience," in *Proc. of ACM SIGCOMM*, September 2000.
17. J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "BGP Routing Stability of Popular Destinations," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, November 2002.
18. S. Uhlig, "On the sensitivity of transit AShs to internal failures," in *Proc. of the fifth IEEE International Workshop on IP Operations and Management (IPOM2005), Barcelona, Spain*, October 2005.
19. O. Bonaventure, S. Uhlig, and B. Quoitin, "The case for more versatile BGP Route Reflectors," Work in progress, draft-bonaventure-bgp-route-reflectors-00.txt, July 2004.
20. "TOTEM: a TOolbox for Traffic Engineering Methods," `http://totem.info.ucl.ac.be/`.