Internet Engineering Task Force
INTERNET-DRAFT
Olivier Bonaventure, Steve Uhlig and Bruno Quoitin
UCL
July, 2004

The case for more versatile BGP Route Reflectors

Status of this Memo

Copyright Notice

Abstract

The Border Gateway Protocol (BGP) is the standard interdomain routing protocol in the Internet. Inside an Autonomous System (AS), the interdomain routes are often distributed by using BGP Route Reflectors (RR). Today, most RR are simple BGP routers. We show that by adding intelligence to the RR, it is possible to improve both the routing and the packet forwarding in ASes. We show how a versatile RR can help an AS to engineer the flow of its incoming or outgoing interdomain traffic. We also discuss how a versatile RR

could help to reduce the BGP convergence time or reduce the size of the routing tables when providing BGP/MPLS VPN services.

1 Introduction

The Border Gateway Protocol (BGP) [1] is used today by more than 16.000 Autonomous Systems (AS) to exchange their interdomain routes. The stability and performance of BGP are key factors for the stability and performance of the global Internet. Although BGP suffers from a low convergence in case of failure and some BGP routers tend to transmit too many routing messages, recent studies have shown that BGP routing is stable [2, 3], at least when considering the routes towards destinations receiving lots of packets. BGP is also used inside many ISPs to distribute several other types of information such as BGP/MPLS VPN routes [4] or flow specifications [5].

When used for interdomain routing, BGP relies on two types of sessions that are established over TCP connections. Two BGP routers from different domains connected with a physical link will use an eBGP session to exchange their interdomain routes. The interdomain routes received by the border routers of an AS need to be propagated through the AS. This is usually done by relying on iBGP sessions. The initial BGP specification assumed that a full-mesh of iBGP sessions would be established inside each AS to distribute the interdomain routes. A consequence of this full-mesh of iBGP sessions is that a BGP router will not distribute over an iBGP session a route received over another iBGP session. However, this full-mesh quickly appeared unscalable since an AS with N routers needs to support $\frac{N\times(N-1)}{2}$ iBGP sessions.

Two solutions have been proposed to solve this scaling problem. With the confederation approach, each AS is divided into smaller sub-ASes containing each about a few tens of routers. Inside each sub-AS, a full mesh of iBGP sessions are established between the routers of the sub-AS and special eBGP sessions are used between routers of different sub-ASes. A second approach, which, based on discussions with ISP operators, appears to be more often used by large ASes, is to rely on BGP Route Reflectors (RR) [6]. A RR is a special BGP router which is allowed to redistribute over iBGP sessions routes that it has received over some iBGP sessions. A RR has two types of iBGP peers : its client-peers and its non-client peers. The non-client peers are usually other RR. A RR will receive routes from all its iBGP peers and will use its BGP decision process and its IGP table to determine the best routes to reach each destination. If the best route was received on an iBGP session with a client peer, it will be advertised to all the iBGP peers. On the other hand, if the route was received from a non-client peer it will only be advertised to client-peers.

The number of RRs in an IP network is much smaller than the number of routers [7]. A network with several tens of routers would typically have one (or two for redundancy reasons) RR. Larger networks with several hundred of routers in various countries may use up to a few tens of RRs connected in a full mesh or with a RR hierarchy.

Discussions with ISPs indicate that there are three ways to deploy RR. The first solution is to place the RR function on existing backbone routers. In this case, the router needs to have enough CPU and memory capabilities to support the RR function while handling its normal load. Another approach is to use a dedicated router that does not forward IP packets but is equipped with a good CPU and large memory. Finally, smaller ASes sometimes rely on PCs or workstations running open-source RRs.

In most deployments of RRs today, the goal is often to minimise the CPU load on the RR. RRs are often only considered as a way to solve the iBGP distribution problem. In this paper, we assume that the RR service is provided by a carrier-class workstation or a cluster of workstations where the CPU and the memory are not as limited as on current routers.

We show in this article that by correctly exploiting the knowledge of the RR, it is possible to provide new services both inside and between ASes. We discuss several examples that could each lead to entire papers on the topic. We show in section 2 that a more intelligent RR could avoid the forwarding loops that may occur with a badly placed current RR. Then, in section 3 we show how a versatile RR could allow a transit AS to efficiently engineer its interdomain traffic. Finally, in section 4, we discuss the role that versatile RRs could play in ASes using MPLS to support VPN services or interdomain LSPs.

2 Limitations of current RR

The currently deployed RRs advertise their `own` best route to each of their client peers. This allows the RR to compute a single best route, but this creates several problems. The first problem is that routing and even forwarding loops can occur when RR are used. Several of those problems have been described in the literature [8] and reported in real networks [9].

As an example, consider the topology shown in Figure 1 based on [8]. The arrows show the BGP sessions. The IGP weight of each physical link is also shown. In this network, RX and RY advertise the prefix P. The two RRs prefer the route learned via eBGP and advertise it to their client. If $R1$ receives a packet destined to P, its BGP table forces it to send it to $RR1$. However, the IGP topology will cause the packet to be sent to $R2$. $R2$'s BGP table forces it to send the packet to $RR2$, but to reach this nexthop, $R2$ will send the packet via $R1$ ...

Extensions to BGP [10] have been proposed to solve this problem, but they are not implemented and deployed. The current solution is to apply guidelines when designing iBGP topologies [11, 12, 7]. Those guidelines impose restrictions on the graph of the iBGP sessions. Those restrictions depend on the IGP topology and the location of the RR. In practice, the IGP topology changes frequently as links or routers fail or are added to the network or when traffic engineering tools are used to engineer the intradomain traffic by setting the IGP weights [13]. Ensuring that the guidelines are preserved after each IGP change is not an easy task.

If the CPU of the RR is not a severe bottleneck, a solution to avoid the routing and forwarding loops induced by RRs would be to change the behaviour of the RR. Instead of computing its own best route which is then distributed to all its clients, a RR could compute the best route that would be computed by each client if it had the same BGP table as the RR. Since one step of the BGP decision process uses the IGP distance between the router and the nexthop contained in each BGP route, the RR would need to know the IGP distance between each of its clients and each BGP nexthop. This information can be obtained by computing the IGP table of each client or by defining a new protocol to allow a

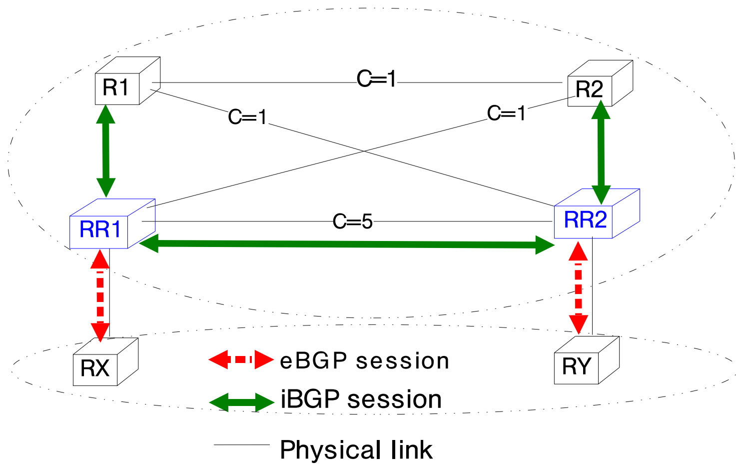

Figure 1: Simple network topology with a forwarding loop

client to report this information to its RR [14]. If the RR recomputes the IGP tables of its clients, they need to be updated after each IGP change. Several algorithms [15] have been proposed to incrementally update the routing table of a router after a topology change. There are also incremental versions of the all-pairs shortest paths algorithms [16]. Based on those algorithms, it should be possible to build incremental algorithms to determine the BGP updates to be sent to the clients of a RR after a BGP or an IGP change.

Another issue with RR is the convergence time in case of failure. Consider again the network topology shown in Figure 1. Assume that the bottom AS is a provider advertising prefix P at both RX and RY. Assume that the forwarding loop problem mentioned above has been solved by forcing each RR to compute the best route for each client. In this case, $R1$ sends its packet to P via $RR2$. If the link $RR2 - RY$ fails, $RR2$ would withdraw its route to P on its iBGP session with $RR1$. $RR1$ would then send a new route to $R1$. If instead of one prefix we consider that 100,000 routes used by $R1$ pass via $RR2$. Then, when the $RR2 - RY$ eBGP session fails, $RR2$ needs to withdraw 100,000 routes on the $RR2 - RR1$ iBGP session. $RR1$ would then need to update the 100,000 routes on the $RR1 - R1$ iBGP session. This could take several seconds or more depending on the performance of the RR. If instead a full-mesh of iBGP sessions was used in this network, $R1$ would have received all the eBGP routes learned by $RR2$ and $RR1$. When the failure of link $RR2 - RY$ is reported by the IGP, $R1$ could consider all the routes via $RR2$ as unreachable and could switch to the routes learned from $RR1$. In large ISPs with a hierarchy of RRs, the impact of the RRs on the BGP convergence time may be even larger.

With today's stringent SLAs, there is a clear need to reduce the convergence time in case of failures. A versatile RR could help to reduce it by advertising several routes to its clients. Knowing the IGP table of each of its clients, the RR can easily determine the best BGP route, but also the *second* route that it would select if the first become unreachable. By using the BGP extensions proposed in [10], the RR could advertise the *best* and the *second* route to each client. This would ensure that the client can quickly switch to a new route when the primary one becomes unreachable. This solution could probably be even more useful in networks providing RFC2547 BGP/MPLS VPN services given their tight SLA constraints.

3 RR-assisted traffic engineering

Another important problem in the global Internet is the need to perform traffic engineering. Several solutions to engineer the flow of the IP packets in the network are used. Some tune the intradomain traffic by setting of the IGP weights [13] or establishing MPLS LSPs [17]. Another problem is to engineer the flow of the inter-domain traffic. As mentioned in [17], "`inter-domain Internet traffic engineering is crucial to the performance enhancement of the global Internet infrastructure.`" However, inter-domain traffic engineering today often relies on tweaking the configurations of the routers [18, 19] and is often more an art than science.

3.1 Reference environment

To perform traffic engineering, a RR needs two types of information. Traffic statistics constitute the first type of information. For intradomain traffic engineering, those statistics are collected as POP-POP or router-router traffic matrices. For interdomain traffic engineering purposes, more precise statistics are required at the granularity of the BGP routes. However, in practice, accurate statistics for each route are not required [20]. Studies of the traffic characteristics in different networks [18, 19] have shown that a small number of prefixes are responsible for most of the traffic. We assume in this paper that per-BGP route volume statistics are maintained by the border routers and sent to the RR for those `heavy` prefixes.

The second type of information required to engineer the flow of the interdomain traffic are the routing tables of the border routers. A RR is ideally placed to obtain this information since it already collects the BGP routes and participates in the IGP.

Figure 2 provides the typical network environment for RR-based traffic engineering. At regular time intervals or when some routing change occur, the RR computes the best route that each ingress BGP router should use to reach a particular destination prefix outside the AS.

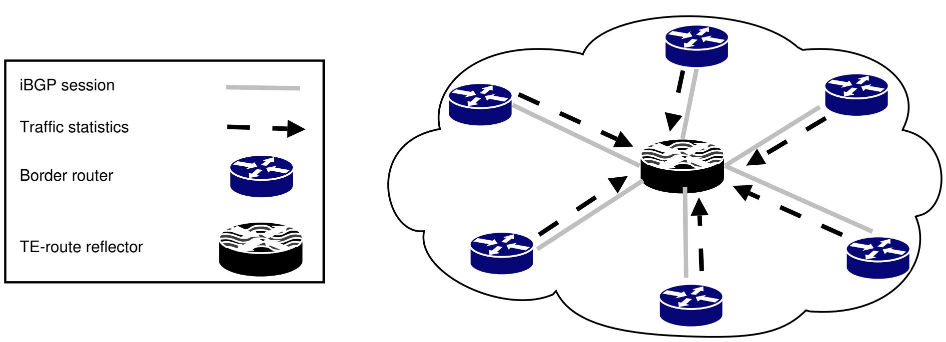

Figure 2: Typical configuration of traffic engineering RR.

When considering interdomain traffic engineering, we need to distinguish between the control of the outbound traffic and the control of the inbound traffic.

3.2 Outbound interdomain traffic engineering

Let us first consider the case of a stub AS that needs to engineer its outgoing traffic to a few transit providers. In principle, this engineering is simple since the network operator can define filters on all its border routers to prefer some upstream provider for some prefixes. However, the size of the BGP routing tables (more than 140.000 routes today) make the search of the ideal configuration difficult [21]. Furthermore, the traffic pattern changes regularly [3] and thus a perfect configuration at time t may become inconvenient at time $t + 1$. In [22, 23], we have shown that by using intelligent route reflectors, it is possible to engineer the flow of the outbound interdomain traffic even when the traffic patterns changes with time.

The principles of the solution described in [22, 23] can be summarised as follows. First, the RR collects traffic statistics regularly as explained above. Second, the RR receives all the routes from the stub's providers. This can be obtained by establishing multi-hop eBGP sessions between the RR and the border router of each provider. Another solution is the BGP extension proposed in [10] to force the stub's border routers to advertise all their routes and not only their best routes. To control the flow of the outgoing traffic, the RR simply has to control the iBGP advertisements that it sends to the stub's border routers.

Based on this routing and traffic information, the RR regularly runs an evolutionary algorithm. This algorithm can be configured with different objective functions such as balancing the traffic among providers, reducing the total cost based on the billing used, ... To fulfil the objective function, the evolutionary algorithm will select from time to time, a few prefixes to be moved from one provider to another. We have shown in [23] that load balancing was possible with only a few iBGP messages per minute, an iBGP load much lower than the normal load of BGP messages in the global Internet.

The RR-based traffic engineering method described above is also applicable for transit ASes. A more detailed description of this approach may be found in [24, 25].

In a stub AS, changing iBGP advertisements is possible since the impact of those advertisements is limited to the stub AS. In a transit AS such as GEANT, an iBGP change can lead sometimes to eBGP changes that could force peers to change their best BGP route. To prevent the BGP route changes to generate instabilities in the rest of the Internet, aggregation could be used by the local AS so that changes in the egress point within the AS do not impact customer ASes. Figure 3 illustrates the use of aggregation to prevent frequent BGP route changes that do not impact the actual path followed by IP packets for the ASes upstream from the flow of the traffic. Suppose that the route reflector RR decides to change the egress point to reach the external prefix `A.B.C.D/Y`, from egress $E1$ to egress $E2$ for ingress point $I1$. Under normal conditions, whenever ingress I1 changes its best BGP route to reach prefix `A.B.C.D/Y`, it requires that a new BGP route be advertised by $I1$ to the external BGP peers. To prevent $I1$ to have to advertise a new BGP route every time the best egress point to be used by $I1$ changes, $I1$ could advertise to its upstream customers an aggregated AS path. This AS path would contain the set of ASes present in the two BGP routes that could be used by $I1$ to reach the destination prefix `A.B.C.D/Y`, as illustrated by Figure 3. In practice, the RR does not need to aggregate the AS-Paths of all the possible routes to a destination, only the routes that it could select with its modified decision process. Often, the best routes for a given destination will be learned from the same peer over different peering sessions. In this case, the aggregation is trivial since all the routes have the same AS-Path.

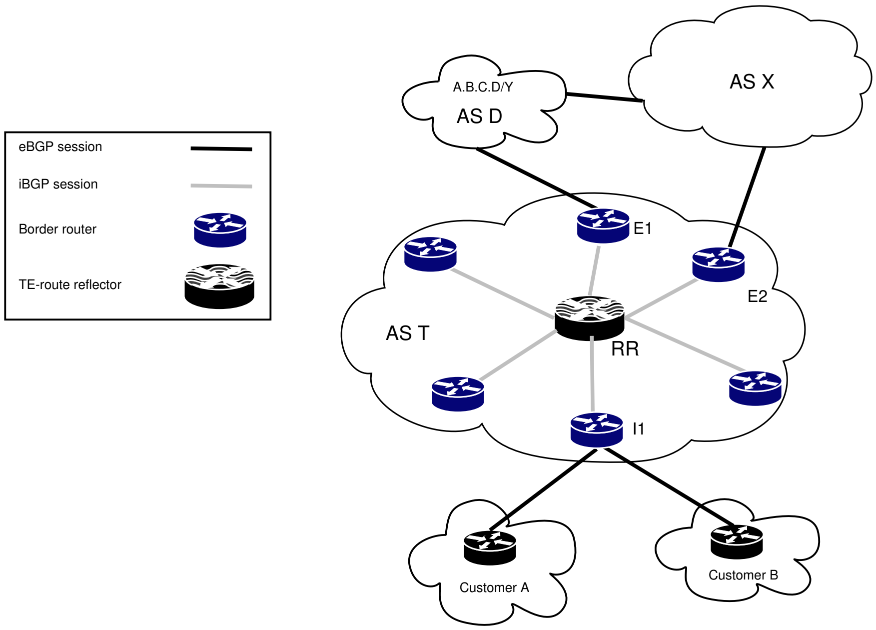

Figure 3: AS path aggregation by RR.

The solution described above could be extended to larger transit ASes that contain more than one (pair of) RR. This would require the definition of protocols that allow RR to exchange routing information and traffic statistics and coordination mechanism between the RRs. For instance, one could choose that each RR is responsible for the ingress routers it has an iBGP session with. Each RR would then compute the best route for its ingress routers towards each destination prefix and send them to these ingress routers. For this solution to be scalable in terms of the BGP advertisements, each RR would advertise to all other RR's of the domain aggregated AS paths.

3.3 Inbound Interdomain Traffic Engineering

Engineering a domain's incoming traffic with BGP is a difficult task [19, 26, 18]. Indeed engineering the incoming traffic of one domain requires the ability to influence how distant domains will select the route that they use to send packets towards the domain. Different techniques exist: announcing more specific prefixes, making selective announcements, prepending the AS-Path and using redistribution communities [19] [27]. However, these methods suffer from several drawbacks. The first two methods increase the size of the BGP routing tables of all routers. AS-Path prepending, while being a widely used method, is known to be coarse and unpredictable. Finally, the redistribution communities are difficult to setup due to the combinatorial explosion of possibilities and the inaccurate view of the topology and policies one has from a single domain's point of view [27].

In this section, we show that a more deterministic approach to engineering the flow of the incoming traffic is possible. Our method relies on a cooperation between the source and destination domains and results in the establishment of interdomain tunnels. A destination domain willing to control how it is reached by a source domain requests the source to establish a tunnel to one of its border router. The tunnel is then used by the source to forward the packets destined to the destination domain. In this way, the packets sent by the given source enter the destination's network through the desired access link.

To explain our approach, let us consider the example topology shown in Figure 4. $AS1$ is a stub that wants to control how it is reached by source $AS2$. On the figure, we can see that there exists multiple interdomain paths between $AS2$ and $AS1$. With the normal BGP, the packets from $AS2$ reach $AS1$ via router $RD1$.

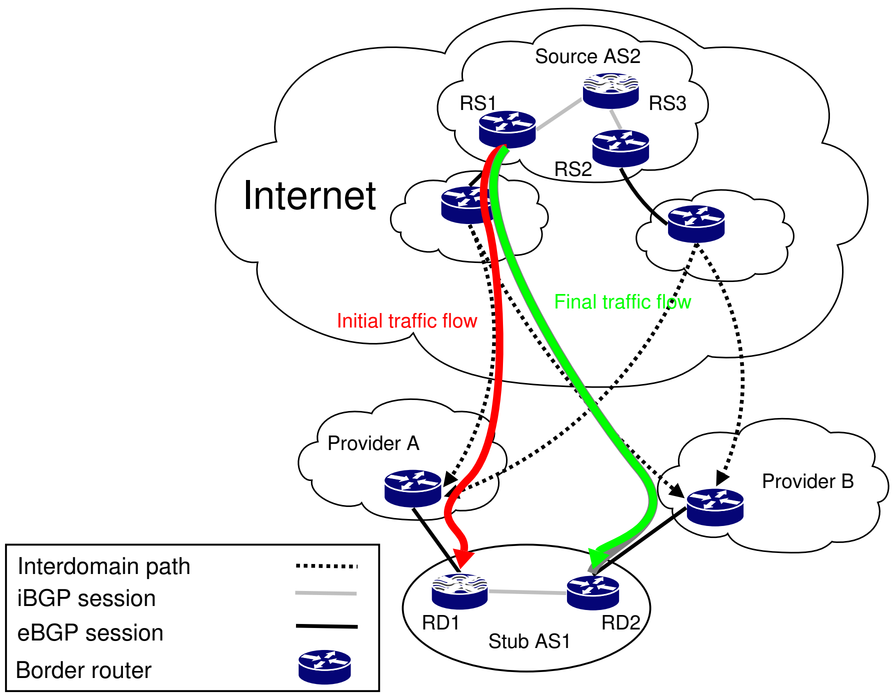

Figure 4: Inbound TE using tunnels.

Assume that to reduce the delay or balance its incoming traffic, AS1 wishes to receive the packets sent by $AS2$ via $ProviderB$ and thus router $RD2$. For this, $AS1$ will request $AS2$ to establish a tunnel with destination $RD2$ to reach all its prefixes. For this, we propose that a route-reflector $RD1$ inside $AS1$ establishes an eBGP session with a route-reflector, $RS3$, in the source domain $AS2$. This multihop eBGP session could be established manually as in the case of peering links or more dynamically. To allow a dynamic establishment of those sessions, $AS2$ must advertise the address of its route-reflector that needs to be contacted. This address can be encoded as an extended community value attached to the route(s) advertised by $AS2$. To avoid security issues, the multi-hop eBGP session should be established over an IPSec tunnel that provides authentication, data integrity and anti-replay. Moreover, BGP extensions such as S-BGP [28] or soBGP [29] should be used to check the validity of the prefixes advertised by $RD1$.

The destination domain will typically advertise its own prefixes over the multi-hop eBGP session and the source domain will not advertise any prefix. Each BGP advertisement will also contain a flexible community value [30] indicating the tunnel endpoint in the destination domain ($RD2$ in our example), the type of tunnel to be used (L2TP, GRE, ...) and possible tunnel parameters such as cookies or identifiers. Instead of using flexible communities, another possibility would be to use MP_BGP and to carry tunnel related information in the MP_REACH_NLRI and a tunnel-SAFI as proposed in [31]. By using as tunnel end-point the IP address of $RD2$ on the link with ProviderB in figure 4, the destination domain can control the ingress link over which the packets will arrive provided that this address is advertised by $ProviderB$.

When $RS3$ has received the route towards the network of $AS1$ over the multi-hop eBGP session, it will select which router(s) will establish the requested tunnel(s) towards the tunnel end-point. It must also update the routes that it distributes with iBGP inside $AS2$ to ensure that the packets towards $AS1$ will be forwarded to the tunnel head-end in $AS2$. Prior to establishing one tunnel towards $AS1$, $RS3$ needs to check that the tunnel end-point is reachable by verifying that it has received at least one BGP route to reach it. Depending on the connectivity of $AS2$, $RS3$ may choose to establish one or several tunnels to reach the endpoint. Since $RS3$ is a route-reflector, it has the most complete knowledge of the available routes towards the tunnel endpoint. $RS3$ will typically select $AS2$'s best egress router to reach the endpoint as the head-end of the tunnel. Note that the selection may depend on other criteria such as the availability of special hardware to perform the required encapsulation on the routers. In order to ask a client to establish a tunnel towards $RD2$, $RS3$ sends to this client an iBGP update containing the tunnel attributes. Upon reception of this update, the client establishes the tunnel. Once the tunnel is up and running, it updates its routing table and sends iBGP advertisements to announce the new route in $AS2$.

In the case of a stub source domain, the above procedure will only cause iBGP changes. On the contrary, if the source domain is a transit AS, the new routes using the tunnel could be advertised outside the domain. In this case, the BGP updates that are advertised outside the source domain $AS2$ should have an AS-Path that is composed first of $AS2$ itself, followed by the AS-Path of the route followed by the tunnel and finally, the destination domain $AS1$. Using such a path is necessary to allow BGP to continue to detect loops by using the AS-Path attribute. Indeed, without this AS-Path, a transit domain ASX could select $AS2$ has its next-hop to reach $AS1$ while the tunnel used by $AS2$ passes through ASX. The traffic would then pass twice through the same domain which would be a waste of ressources.

IP tunnels such as GRE or L2TP have been often been criticised because of the cost of encapsulating/decapsulating packets and the risk of fragmentation. The first problem is not anymore an issue since several vendors offer interfaces supporting encapsulation/decapsulation at line rate. With Packet over SONET/SDH links, the MTU is less a problem given the available frame size. Furthermore PathMTU discovery is used by almost all endsystems today/widely deployed and used. Compared to other proposals such as [26], the solution described above can be used without deploying new protocols in the transit domains. For example, universities or research networks could use it to control high-bandwidth flows.

4 Route Reflectors and MPLS

Many large ISPs are currently using MPLS to provide BGP/MPLS VPN services to their corporate customers [4]. Today, those services are often provided within a single AS. Three types of routers are usually distinguished in a network providing BGP/MPLS VPN services. A CE router is a router owned and maintained by a customer. A PE router is a router maintained by the network provider and directly attached to a CE router. A PE router will usually learn the routes reachable via each of its attached CE routers through a special IGP or BGP session [4]. To isolate all the different VPNs, a PE router will maintain one VPN Routing and Forwarding table (VRF) for each supported VPN. BGP is used by the PE routers to distribute the content of their VRF to other PE that are attached to the same VPN customers. The forwarding of VPN packets from one PE to another relies on the utilisation of MPLS, GRE or IPSec tunnels. Thanks to the utilisation of those tunnels, the core routers, also called P routers, do not need to maintain per-VPN VRFs. Since BGP is used to distribute the VPN routes inside the network, RR are often used to scale the iBGP full-mesh between the PE routers.

Thanks to the `routeviews` and `RIPE RIS` projects, the behaviour of BGP in the global Internet has received a lot of attention and BGP is better known than a few years ago. Despite of that, few studies have analysed BGP/MPLS VPNs. A recent study [32] revealed that the behaviour of BGP is very different when considering VPN services than when considering the global Internet.

A first difference is the size of the routing tables. In the global Internet, few routes are more specific than `/24` and the BGP routes are very stable. In the BGP/MPLS network analysed in [32], the situation is completely different. First, the BGP/MPLS routing table is already larger than the Internet BGP routing table and is growing quickly. Second, the BGP/MPLS VPN routing table contains much more specific prefixes than the BGP Internet routing table. Figure 5, based on [32] compares the percentage of routes for the most common prefix lengths. This figure shows that 55% of the Internet routes are for `/24` prefixes and other common prefix sizes are `/16` to `/23`. In the BGP/MPLS VPN routing tables, 38% of the routes have a `/32` IPv4 prefix as destination and 9% correspond to a `/30` prefix. A first consequence of those specific routes is that in the network studied in [32], the BGP/MPLS VPN routing table in the RR is already larger than the BGP Internet routing table and the BGP/MPLS routing table. Another consequence is that the BGP/MPLS VPN routes are less stable and the BGP messages are much more frequent in the BGP/MPLS network [32].

The size of the BGP/MPLS routing tables will force operators to utilise route aggregation mechanisms for the BGP/MPLS VPNs. The default BGP aggregation [1] is able to aggregate routes for contiguous prefixes coming from different ASes in a single advertisement. This technique could be applied by the customers on the CE routers. However, a CE router could only aggregate its local routes. A versatile RR, receiving VPN routes from several PE routers could perform a better aggregation by considering all the routes inside each VPN. Given the volatility of some BGP/MPLS routes, the RR would need to be able to change an aggregate dynamically after an event in a customer network.

The next step for the BGP/MPLS VPNs is to provide those services across different ASes. Several solutions are proposed in [4]. One of the possible solutions is to directly interconnect the RR of different ASes with a multi-hop eBGP session to distribute the

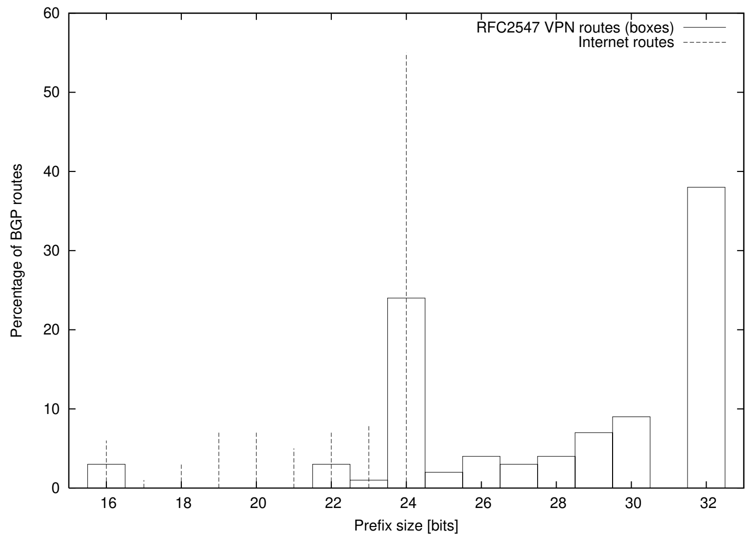

Figure 5: Prefix distribution in the BGP Internet and BGP/MPLS VPN routing tables

inter-provider VPN routes. In this case, the RR should clearly aggregate the VPN routes that it sends over the multi-hop eBGP session.

Another problem with BGP/MPLS VPNs is that important VPN sites are often attached to two different PE routers. This dual attachment is often required for redundancy, but once the two links are established, customers often require to be able to use them for both inbound and outbound traffic. For the packets sent by the CE router to the network provider, this depends only on the customer network. For the packets sent by the VPN provider towards the CE router, the ability to load-balance the traffic between the two PE routers depends on the configuration of PE routers of the VPN provider. A possible solution is to use per-site `route distinguishers` [4] to ensure that each PE receives all the advertisements from all the PE routers attached to the same VPN. However, this increases the size of the BGP/MPLS routing tables. A versatile route reflector could be configured to advertise a single route when scalability is important and several routes, for example by using the BGP extensions proposed in [10], for the VPNs sites where load-balancing must be achieved.

Another situation where RR could play a role in MPLS networks is when interdomain LSPs [33] need to be established with RSVP-TE. To establish a LSP with RSVP-TE, the head-end Label Switching Router (LSR) computes an explicit route. In a single IGP area, this computation relies on the topology distributed by the IGP. Across interdomain boundaries, this computation becomes more difficult since BGP distributes reachability and not topological information. For a primary LSP, the head-end LSR could use the route distributed by BGP. For a disjoint secondary LSP, this becomes more difficult as the head-end usually only receive the best BGP route to each destination. A RR that collects all the candidate routes learned via BGP could select among those routes to find a disjoint route for the secondary LSP.

5 Conclusion

BGP Route Reflectors were designed to solve the scaling problem of the iBGP full-mesh. For this, the RR collects the best routes from all its clients. Instead of only serving as a distributor of iBGP advertisements, we have shown that by exploiting the routing knowledge of the RR it is possible to improve the routing in ASes.

We have then shown several situations where versatile RR could be used to support very useful services in Autonomous Systems. One of those situations is the need to engineer the flow of the outgoing interdomain traffic of a stub or transit AS. Another situation occurs when an AS wishes to control the flow of its incoming traffic. Besides those traffic engineering usages, versatile RR could also be used to reduce the convergence time in case of failure or the size of the BGP/MPLS routing tables.

Acknowledgements

This work was supported by the DGTRE in the framework of the TOTEM project (`http://totem.info.ucl.ac.be`). We would like to thank Nicolas Dubois for the data used in figure 5.

References

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," April 2003, internet draft, draft-ietf-idr-bgp4-20.txt, work in progress.

[2] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "Bgp routing stability of popular destinations," in `Proc. Internet Measurement Workshop`, November 2002.

[3] S. Uhlig, V. Magnin, O. Bonaventure, C. Rapier, and L. Deri, "Implications of the topological properties of internet traffic on traffic engineering," in `ACM Symposium on Applied Computing`, March 2004.

[4] E. Rosen and Y. Rekhter, "BGP/MPLS IP VPNs," September 2003, internet draft, draft-ietf-l3vpn-rfc2547bis-01.txt, work in progress.

[5] P. Marques, N. Sheth, R. Raszuk, J. Mauch, and D. McPherson, "Dissemination of flow specification rules," June 2003, internet draft, draft-marques-idr-flow-spec-00.txt, work in progress.

[6] T. Bates, R. Chandra, and E. Chen, "BGP route reflection - an alternative to full mesh iBGP," April 2000, internet RFC 2796.

[7] B. Halabi, `Internet Routing Architectures`. Cisco Press, 1997.

[8] T. Griffin and G. Wilfong, "Analysis of the MED oscillation problem in BGP," in `ICNP2002`, 2002.

[9] D. McPherson, V. Gill, D. Walton, and A. Retana, "BGP persistent route oscillation condition," 2002, internet draft, draft-ietf-idr-route-oscillation-01.txt, work in progress.

[10] D. Walton, D. Cook, A. Retana, and J. Scudder, "Advertisement of Multiple Paths in BGP," November 2002, internet draft, draft-walton-bgp-add-paths-01.txt, work in progress.

[11] T. Griffin and G. Wilfong, "On the correctnes of iBGP configuration," in `SIGCOMM'02`, Pittsburgh, PA, USA, August 2002, pp. 17–29.

[12] L. Xiao, J. Wang, and K. Nahrstedt, "Reliability-aware IBGP Route Reflection Topology Design," in `11th IEEE International Conference on Network Protocols (ICNP 2003)`, Atlanta, Georgia, USA, November 2003.

[13] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," `IEEE Communications Magazine`, October 2002.

[14] R. Musunuri and J. Cobb, "A complete solution to stable iBGP," in `IEEE International Conference on Communications (ICC)`, 2004.

[15] C. Alaettinoglu, V. Jacobson, and H. Yu, "Towards millisecond IGP congergence," November 2000, internet draft, draft-alaettinoglu-ISIS-convergence-00.ps, wor k in progress.

[16] C. Demetrescu and G. F. Italiano, "A New Approach to Dynamic All Pairs Shortest Paths," in `Proceedings of the 35th ACM symposium on Theory of computing (STOC'03)`, June 2003, pp. 159–166.

[17] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of internet traffic engineering," May 2002, rFC 3272.

[18] N. Feamster, J. Borkenhagen, and J. Rexford, "Guidelines for interdomain traffic engineering," `SIGCOMM Comput. Commun. Rev.`, vol. 33, no. 5, pp. 19–30, 2003.

[19] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure, "Interdomain traffic engineering with BGP," `IEEE Communications Magazine`, May 2003.

[20] S. Leinen, "Evaluation of candidate protocols for IP flow information ex port (IPFIX)," January 2004, internet draft, draft-leinen-ipfix-eval-contrib-02, work in pr ogress.

[21] T. Ye and S. Kalyanaraman, "A recursive random search algorithm for large-scale network parameter configuration," in `Proc. of ACM SIGMETRICS`, 2003.

[22] S. Uhlig, O. Bonaventure, and B. Quoitin, "Interdomain Traffic Engineering with minimal BGP Configurations," in `Proc. of the` 18^{th} `International Teletraffic Congress, Berlin`, September 2003.

[23] S. Uhlig, "Implications of the traffic characteristics on interdomain tra ffic engineering," Ph.D. dissertation, Computer Science and Engineering Department, Unive rsité catholique de Louvain, March 2004.

[24] ——, "A multiple-objectives evolutionary perspective to interdomain traffic engineering in the internet," in `Workshop on Nature Inspired Approaches to Networks and Telecommunications (NIANT) in PPSN04, Birmingham, UK`, September 2004.

[25] S. Uhlig and B. Quoitin, "BGP-based interdomain traffic engineering for transit ASes."

[26] S. Agarwal, C. Chuah, and R. Katz, "OPCA: Robust Interdomain Policy Rrouting and Traffic Control," in Proceedings of the 6th International Conference on Open Architecture and Network Programming, IEEE OpenArch, April 2003.

[27] B. Quoitin, S. Tandel, S. Uhlig, and O. Bonaventure, "Interdomain Traffic Engineering with Redistribution Communities," Computer Communications, vol. 27, no. 4, pp. 355–363, March 2004.

[28] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," IEEE Journal on Selected Areas in Communications, vol. 18, no. 4, pp. 582–592, April 2000.

[29] R. White, "Securing BGP Through Secure Origin BGP," The Internet Protocol Journal, vol. 6, pp. 15–22, June 2003.

[30] A. Lange, "Flexible BGP Communities," March 2004, internet draft, draft-lange-flexible-communities-02, work in progress.

[31] G. Nalawade, R. Kapoor, and D. Tappan, "Tunnel SAFI," October 2003, internet Draft, draft-nalawade-kapoor-tunnel-safi-01, work in progress.

[32] M. Nicolas, "BGP/MPLS VPN monitoring for troubleshooting, scalability verification and network migration safety," February 2004, presentation at MPLS2004 , Paris (France).

[33] R. Zhang and J. Vasseur, "MPLS Inter-AS traffic engineering requirements," November 2003, internet draft, draft-ietf-tewg-interas-mpls-te-req-02.txt, work in progress.

Authors' addresses

Olivier Bonaventure, Steve Uhlig, Bruno Quoitin
Dept. Computing Science and Engineering
Universite catholique de Louvain (UCL)
Place Sainte-Barbe 2
B-1348 Louvain-la-Neuve
Belgium
http://www.info.ucl.ac.be/people/OBO

Intellectual Property Statement

This Internet-Draft will expire on December 31, 2004.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.