Traffic engineering with BGP

Christian Ludwig
(cludwig@cs.tu-berlin.de)

Seminar "Internet Routing" ,
Technical University Berlin

SS 2009 (version from July 2, 2009)

Abstract

This paper will give an overview of how to connect reliably to the Internet using connections to multiple Internet Service Providers (ISPs). Since there are numerous possible ways for packets to enter and leave our own network in such a setup, a network operator might want to control this behavior as best as possible for several reasons. We will discuss how to achieve this goal.

1 Introduction

The Internet can be seen as a network of networks. It is a loose connection of small networks, each one of them forming a separate administrative domain. Neighboring domains are called peers. The responsibility of passing packets from one peer to another is in the hands of every participating party.

In the early days of the Internet the major concern was connectivity, due to bad lines resulting in unstable links. The number of links, which were interconnecting domains was also small. As technology evolved and the Internet grew, more links between peers were built. These are also more and more reliable and stable. Nevertheless, until today, there is no guarantee for links not to fail. Therefore resilient network setups are built and networks are multi-connected. This leaves us with multiple ways in which a packet can be routed to its destination, since there are many redundant ways usable at a time. Also, having these redundant links helps to cope with peaks in bandwidth usage.

The question is which of these ways is the *best* possible one that a packet should be sent to? Since every domain has its own set of local policies, sometimes policies in one domain may contradict to policies in other domains. We will show how this question can be answered in a satisfactory manner for all affected parties.

1.	Higher local preference
2.	Shorter AS path
3.	Lower number in `ORIGIN` attribute
4.	Lower value in `MULTI_EXIT_DISC` attribute
5.	prefer eBGP over iBGP routes
6.	Lower metric to the `NEXT_HOP`

Table 1: Order of precedence for choosing a path in BGP

In the topology of the Internet two main classifications of domains were found by [SARK02], stub domains and transit domains. Stub domains contain only hosts where traffic destines or originates from. These domains do not contain traffic for other domains. ISPs connecting home users use stub domains and get transit services from their upstream peers. Transit domains, on the other hand, do only pipe packets through their network. They do not have sources or sinks of information, instead, they have established agreements with their neighbors to pass packets between them.

Inside of a domain administrators are free to choose whatever routing mechanism suits them best. As one possible setup Multiprotocol Label Switching (MPLS, [RFC3031]) can be used.

Since every peer needs to know where to find everybody else in the Internet, routing information needs to be exchanged between peers. A common protocol was needed to exchange this information using the greatest common denominator available, simple TCP/IP. Over time the Border Gateway Protocol (BGP, [RFC4271]) evolved as the de facto standard routing protocol used by every peer.

2 BGP basics

BGP is a path vector protocol. In BGP terminology, domains are called `Autonomous Systems` (ASes). An administrative domain can consist of multiple ASes, depending on the domain administrator's needs. Routes are represented by IP subnet prefixes. These are exchanged between peers using `UPDATE` messages. This mechanism is commonly known as route advertisement. The receiving router *may* insert the information in its routing table, depending on path length, local network policy and other rules. Announcing a route advertisement to neighboring peers means that the announcing peer will definitely route packets to the IP subnet enclosed inside of the BGP routing advertisement.

To be able to discuss the possible solutions for traffic engineers, we need to understand the criteria, which the BGP protocol defines to choose a route ([RFC4271]). Table 1 shows the order of precedence. The highest influence on routing decisions has the local preference attribute (`LOCAL_PREF`). Assigning a high value to a routing announcement prefers the enclosed route. Therefore the domain administrator has direct influence on how to route traffic. As a second parameter the length of the `AS_PATH` attribute is considered. This attribute contains all ASes on the way leading to the advertised IP prefix. The shorter the way, the better the path. The idea is to reduce the overall packet transfer delay. If there was still no decision possible up to this point, the `ORIGIN` and the `MULTI_EXIT_DISC` (MED) attributes are considered. The last two considerations for the decision process are there for exceptional cases,

where all other values are the same for different announcements.

That means, the final decision on whether an advertised route will be accepted or not by its peers is subject to the local administrator of the receiving AS's router in most cases.

3 Peering relationships

An important step towards understanding the needs for traffic engineering is to have a characterization of Internet traffic. We can use many different approaches to find a taxonomy of Internet traffic, for example, we could use technical values like IP prefixes. A far better method can be employed using an economical point of view, which takes a closer look at the business relationship of each two parties, agreeing on a peering arrangement. Two main peering-models emerged ([SARK02]). First there is the `customer-provider` relationship. This often affects a small stub AS, which acts as a customer in this model. It signs contracts with one or multiple upstream providers to gain Internet access. The provider agrees to route the complete traffic announced by the stub AS and gets paid for that service. Generally, the customer is a smaller player in the field, than the provider. Big providers also have high market power. Having two domains of the same size or market power can lead to a `peer-to-peer` relationship between them. In this model, both parties agree to route traffic originating from another, sharing their traffic fees. The contract details however are in most cases not publicly available.

4 Interdomain traffic patterns

To characterize the amount of traffic and its distribution on the Internet, B. Quoitin, S. Uhlig, et. al. ([QUP+03]) made a statistical analysis of the traffic of three ISPs. They had a special focus on stub ASes, for which they found out, that each of the routing tables revealed nearly the same results. The top 10 sources of traffic contribute to approximately 30% of the overall bandwidth used. Looking at the top 100 sources, they figured out that these contribute to approximately 70% of the total traffic sent. Furthermore, they noticed that the distance to the remote AS for that traffic is mostly about three to four AS hops away.

The concentration of traffic on a relatively small amount of sources means, that there is no need to perform traffic engineering on a global scope. Instead, it will be enough to extract the most traffic intensive sources per domain and find a way to tweak their corresponding BGP announcements. These modifications also need to be able to influence the decision process of ASes, which are a few hops away.

5 Intradomain traffic engineering

Influencing the traffic flow inside the borders of an administrative domain is a rather easy job. All affected routers are under the same administration. In this way the network operator is free to choose his way to distribute routing information among the internal routers. While bound to BGP on the interdomain level, inside a single domain other protocols can be used,

which are aware of traffic engineering requirements and include features to support it. There are numerous alternatives to BGP, which can be chosen here, each having their strengths and weaknesses. The main goal is always to optimize the link-costs of the routes. On the other hand, the network operator needs to consider the protocol translation tasks, when he chooses not to use BGP. If the administrator chooses e. g. MPLS as the intradomain routing protocol, the edge routers will need to translate the routing information between MPLS and BGP. Therefore network administrators mostly choose to use BGP inside their networks, too, connecting routers in a full-meshed tunneled network. When using BGP as interior border gateway protocol (iBGP) on the edge routers, no additional information shipped with the interdomain routes needs to be translated between protocols and is thus unlikely subject to errors. Inside a domain the administrator has the same problems as for the interdomain case, but having all affected routers under control makes planning and deployment of traffic engineering rules easier.

With multiple different routing protocols available, intradomain traffic engineering is a field of its own. It is solely tied to internal company policies, which makes it easier for engineers to implement.

6 Interdomain traffic engineering

The reasons for network operators to consider traffic engineering, or traffic shaping respectively, vary from one another heavily. Consider a stub AS having an uplink to its provider. For redundancy reasons the stub AS may want to have another uplink, which should not be used in normal operation. It should only be used as a failover link. Every traffic generated on the failover link costs the stub AS money, which must be paid in addition to a working primary link. So traffic engineering decisions are pure economical decisions, which therefore highly depend on the ISP's benefit. Since it is possible for some IP prefixes, that routing through the failover link is shorter or faster than routing through the primary link, the stub AS's network operator needs to tell their neighboring peers about his intentions. A transit AS, on the other hand, may want to optimize the traffic flow through their network to balance the load on their routers. They will want to influence their traffic for performance reasons. Content providers will want to shape their outgoing traffic only, whereas access providers will want to shape their incoming traffic.

Unfortunately current BGP does not have any obvious way to specify an AS's traffic engineering requirements. That is why network operators need to have a closer look inside the BGP decision process, as outlined in section 2, to be able to tweak the protocol for fitting their needs best. Since every party has another idea of how interdomain traffic should be routed, it might be possible that traffic engineering decisions in one AS can be contradictory to the ones of others.

In the next sections, we will distinguish between means for outgoing and incoming traffic engineering. Only the combination of both of them leads to appropriate traffic engineering results.

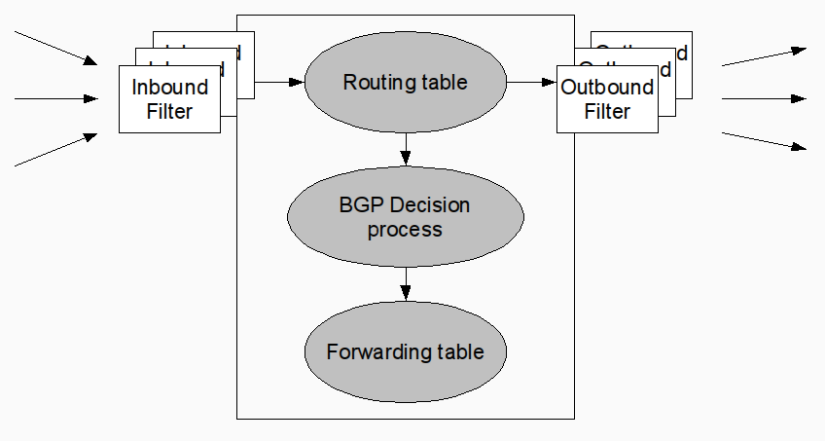

Figure 1: Input and output filters in a BGP router (taken from [QUP+03], slightly modified)

6.1 Influencing outgoing traffic

Sending packets, originating from our own AS to a specific outgoing interdomain link is not hard. As a network operator, we are fully in charge of the intradomain routing policies already and thus can decide where packets should leave our network. We only need to weight our preference on outgoing routes.

Upon receiving a route advertisement from one of our peers, we can apply filters for these incoming `UPDATE` messages, as shown in Figure 1. There can be a separate inbound filter set for each of our peers. These filters can alter BGP attributes, before they reach the decision process. In this way, we can directly manipulate the decisions made by the routing decision engine.

Looking at the BGP routing criteria in Table 1, evaluated by the routing decision process, the only reasonable attribute we can change at the inbound filter level is `LOCAL_PREF`. For example, imagine a domain AS 1 having two upstream providers AS 2 and AS 3 as outlined in Figure 2. AS 1 wants to use AS 2 as primary uplink and AS 3 only for backup purpose. As shown in the figure, the edge router at AS 1 has a direct connection to the edge routers of AS 2 and AS 3. So if AS 1 receives an `UPDATE` notification from AS 3 for a certain IP prefix, the inbound filter alters the corresponding `LOCAL_PREF` attribute to a low value. If another announcement message arrives for the same IP prefix from AS 2 the corresponding edge router of AS 1 sets a higher local preference in the inbound filter rules. Therefore the routing decision engine will always prefer sending packets to AS 2, as requested. Indeed, that filter is not only limited to IP prefixes, but can also trigger for source AS or any other information in the arriving BGP `UPDATE` message. In this way we are able to set a higher weight for preferred routes to the primary upstream provider, having the option to use the secondary route in case we lost the connection to AS 2.

Of course, the outlined setup may become more complex. Each upstream provider could be connected to a separate router in AS 1. Nevertheless, the shown methods stay the same. Both modified route advertisements will be distributed among all intradomain routers in AS 1. One problem is, that both route advertisements will be stored on all internal routers. This increases the size of the routing tables, but therefore the failover works in an automated

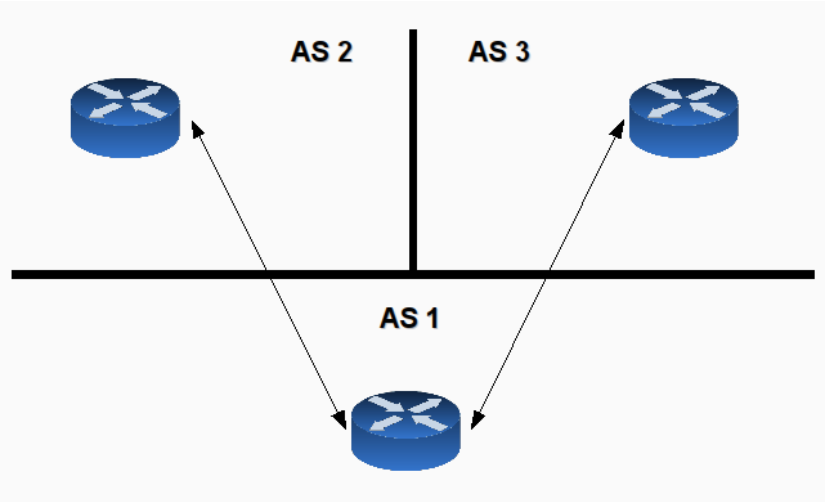

Figure 2: Edge-routers for stub AS 1 with connections to upstream providers AS 2 and AS 3

way. This is more comfortable than manually switching routes between the primary and backup provider upon link failure, which would result in a longer lasting downtime. This is unacceptable for all ISPs. So this trade-off is worth the effort. The upside is, that we only need to influence a small number of routes, as we found out in section 4.

The solution provided can also be used to load-balance traffic on multiple uplinks ([QUP+03]). To achieve a good balance the `LOCAL_PREF` attribute has to be set very carefully on advertised prefixes with high traffic volume. This mainly results in monitoring changes in traffic flow and manually adjusting attribute values.

6.2 Influencing incoming traffic

Changing the flow of incoming network traffic is much harder, than influencing outgoing traffic. For outgoing traffic, we are in direct possession of the packets in question. For incoming traffic, our goal has to be to influence the routing decision upstream. We need to predict filter rules for inbound filters, implemented in our peer's edge router. In other words, we need to compete with upstream's outgoing traffic engineering decisions.

One way to set preferences for a certain interdomain link is to split a route advertisement of one IP prefix into smaller prefixes. The forwarding engine in BGP always chooses the most specific route. Consider AS 1 wants to send an advertisement for `192.168.0.0/23` preferring to receive the traffic through AS 2. It could split that IP prefix in two, `192.168.0.0/24` and `192.168.1.0/24`, and announces all three to AS 2. AS 3, on the other hand, will only be sent the original advertisement. All three `UPDATE` messages will be propagated throughout the Internet, so AS 3 will learn at some point that there are more specific routes, and use these instead. The downsides of this solution are bloating routing tables on all routers on the Internet, leading to a performance hit world-wide.

We have to look at the BGP routing criteria outlined in Table 1 once again, to see which attributes we can choose to influence incoming traffic without increasing the size of the routing

tables significantly. As we can see, the AS_PATH attribute can be used ([QUP+03]). Route metric in terms of BGP is measured in AS-hops. The distance between two points in the Internet is measured in how many ASes are on the way to the destination IP prefix. For each UPDATE message sent to a peer, one has to add its own AS number to the AS_PATH attribute value. BGP considers a route to be better than an existing one when the new one has less AS hops in between. It is a common assumption in BGP that a smaller AS path also leads to a smaller number of IP hops, thus reducing network latency for a connection on this link.

Since we do not want to change our own routing table entries, but only alter those of certain peers, an outbound filter for each upstream provider affected needs to be set in place, as it is shown in Figure 1. We need to keep in mind that sending manipulated routing advertisements to certain peers, makes those competing with the original advertisements sent to others. Our upstream providers are most certainly connected to each other on a different path than ours, too.

A network operator can use the AS_PATH attribute and prepend its own AS number more than once. The distance becomes artificially increased. That makes the route less attractive to be considered as primary choice upstream. This solution leads to multiple problems, however. Having crafted a suboptimal route advertisement for a peer, we have no influence on how that peer handles it. On one hand, that route can be inserted into the peer's routing tables. If our primary link fails, the routes for the backup path are already present and can be used. On the other hand, the peer can discard the route for many reasons on its inbound filter level or routing decision process, for example the AS_PATH can be too long to be considered valid. The result for a failing primary link would be a total loss of connectivity. With this technique, not only can we manipulate our direct peer's routing decision, but also for peers several AS hops away. For backup links a big number of ASes is prepended. With a carefully crafted path length we can also achieve a load-balancing effect. Prepending the correct number of ASes to load-balance some links is done in a trial-and-error approach. There are too many considerations to be taken into account. Most of them are in the hands of different network operators, which do not know each other and have different goals about the desired effects.

For the special case that one AS has multiple interdomain uplinks to another AS, the non-transitive BGP multi-exit-discriminator attribute (MED) can be used to tell upstream which link to use ([QUP+03]). This attribute indicates a preference based on a metric value, when there are multiple upstream links available to the same AS ([RFC4271]).

Besides the MED approach, a network operator cannot reliably determine which incoming link will be utilized for certain IP prefixes with high volume traffic. Influencing the traffic flow for ASes, which are not directly peering with us is even harder and ends up in a trial-and-error approach most of the time. The only working way is to communicate with your upstream providers, asking them for their inbound traffic filters and their settings for the routing decision engine. A peering contract then must contain these information from upstream and the proposed action the network operator wants to perform, to set up the inbound traffic engineering to his contentment. In this way both parties can discuss their requirements and find a consensus satisfying both.

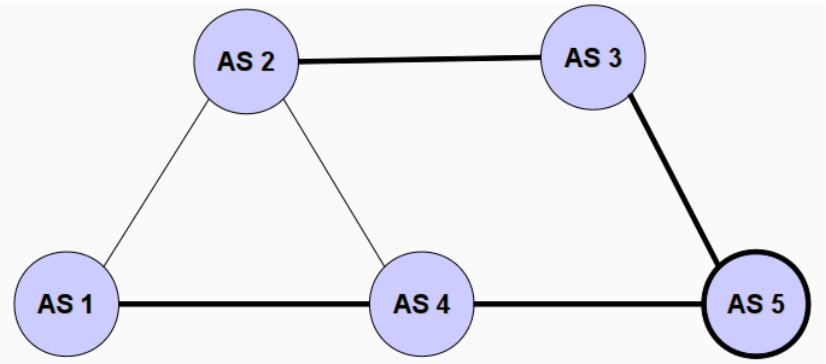

Figure 3: An example network, with focus on AS 5. Preferred paths are highlighted.

6.3 BGP communities

The Internet community recognized the need for an easier way to administer policies in BGP. To accomplish this, an extension to the BGP protocol was made, defining the `communities` attribute ([RFC1997]). This optional attribute can hold multiple 32 bits wide numbers. The network operator can thus append arbitrary additional information to his routing advertisements. This additional information is generic by nature and subject to interpretation. It can have any meaning, from city or country codes to the AS from which a specific announcement was received from.

These numbers only have a meaning between two peering ISPs, so it can be perfectly used for traffic engineering purposes, too. The two ISPs only need to agree on specific numbers and their corresponding meanings on their interdomain link. Therefore downstream has fine-grained control over the redistribution of its routing advertisements, and thus its traffic flow. For example, both could agree, that route advertisements tagged with a certain number in the `communities` attribute may be forwarded only to some of upstream's peers, not everyone.

Technically the sending party alters its `UPDATE` messages towards upstream in the outbound filters on the edge router and inserts a `communities` value, which both agreed upon (cf. Figure 1). The receiving side then resets the `communities` value in its inbound filters and performs the appropriate action necessary to fulfill the agreement.

Communities can also be used inside an AS to carry the meta-information of all peers along with a routing advertisement. The values have to be converted back and forth from a peer-link value to an intradomain value.

To illustrate a more complex example for the use of the `communities` attribute, consider a network like the one outlined in Figure 3. AS 5 receives high volume traffic from AS 1 and AS 2. Therefore it wants to load-balance the incoming traffic for both. Additionally, in normal operation both ASes should take the shortest path. In this scenario AS 5 is unable to influence the incoming traffic itself by `AS_PATH` prepending. AS 4 and AS 5 need to agree on a `communities` value, say 42, which makes AS 4 do an `AS_PATH` prepending of two entries when forwarding a routing advertisement from AS 5 to AS 2. The request does not get altered for any other peer. For its own IP prefixes, AS 5 adds a `communities` value of 42 on the link to AS 4. In return AS 2 now prefers the path over AS 3 to AS 5, rather than ever

using AS 4 in normal operation. AS 1, on the other hand, sends its traffic through AS 4, since this is the shortest path already.

Of course, using `communities` is more work in the first place, but it certainly pays off. Both parties know exactly which traffic needs to be treated in a special way and they also know exactly *how* that traffic should be treated, because in the end they need to agree upon it.

7 Bandwidth aspects

In the example of the previous section we found out that AS 5 has diversified its traffic across all available links. As the bandwidth usage grows, the AS 5 decides to extend the available bandwidth by peering with another upstream provider. The network administrators can distribute the total incoming and outgoing bandwidth across all three lines now to be able to cushioning peak bandwidth uses. They have only a very limited visibility of the Internet and the networks around them, so they are unable to predict changes in their incoming traffic flow. Their outgoing traffic flow might also change, since more attractive routes might pop up. As the worst-case scenario, two of the three available uplinks might not be used at all due to bad traffic engineering rules or failed links. That means to be on the safe side, all interdomain links for AS 5 would need to be able to carry the network's full bandwidth capacity. This gets worse with more peering links involved.

Usually ISPs are connected to at least two interdomain links for availability reasons. More interdomain links are established mainly for performance reasons. They also increase availability, since the more interdomain links a peer has the lower is the probability for every link to fail at once. ISPs usually consider the residual risk in Service Level Agreements as part of their contracts with customers. In a disaster scenario however, if multiple links fail at once, traffic engineering can help to redirect traffic over the remaining higher capacity links, instead of going over lower capacity links. The traffic can be adjusted to the new situation. This is feasible for longer lasting failures, but can also be helpful when restructuring interdomain connections.

8 Conclusion

Traffic engineering in current Internet is generally recognized as a valid means to influence outgoing and incoming packets of an AS. Since the original versions of BGP do not allow to directly specify parameters for traffic engineering, the approaches are more or less based on trial and error. This is bad for engineers, which do not have a reliable way to manipulate their traffic flow.

The introduction of the `communities` attribute leverages that problem a bit. Definitions for the meaning of `communities` values only exist between each two peers. They are unknown to the outside world. Although, `communities` are private and should be filtered on edge routers, some ASes do not do this.

The academic society can only observe the use of traffic engineering techniques, most obviously prepending `AS_PATH` and unfiltered `communities`. Out of this data we can observe

the effect on Internet routing. What we do not know is the reason why the traffic flow was altered in some cases. Was a certain flow intended, or was it just a misconfigured router? Due to that lack of that knowledge an attempt to a more transparent use of `communities` was started ([BCHQW03]). The proposal states that there should be well-known `communities` for most traffic engineering tasks, but it lacks a general fitness especially when influencing ASes, which are multiple hops away ([QPBU04]. For that matter, statistical analysis may be the only way to gather such information at the moment.

There are also tools in the works to automatically tune traffic engineering parameters for an AS (e. g. [UQ05]) aiming for a simple configuration of complex interdomain routing policies for a network administrator. These tools are not perfect, yet. They are mostly based on monitoring the traffic flow to optimize the traffic engineering parameters discussed in this paper. They are automating the trial and error approach with faster response times and lower error rates. These kind of tools are greatly appreciated by ISPs, because faster responses to network topology changes and fewer sub-optimal routing leads to satisfied customers.

References

[RFC4271] Rekhter, Y., Li, T., and S. Hares: *A Border Gateway Protocol 4 (BGP-4)*; RFC 4271, January 2006

[SARK02] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz: *Characterizing the internet hierarchy from multiple vantage points*; in INFOCOM 2002, June 2002.

[RFC3031] E. Rosen, A. Viswanathan, R. Callon: *Multiprotocol Label Switching Architecture*; RFC 3031, January 2001

[QUP+03] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen and O. Bonaventure: *Interdomain traffic engineering with BGP*; IEEE Communications Magazine, Volume 41, Issue 5, May 2003, pages 122–128, ISSN: 0163-6804

[RFC1997] Chandra, R., Traina, P., and T. Li: *BGP Communities Attribute*; RFC 1997, August 1996

[BCHQW03] O. Bonaventure and S. De Cnodder and J. Haas and B. Quoitin and R. White: *Controlling the redistribution of BGP routes*; Work in progress, draft-ietf-grow-bgp-redistribution-00.txt, April 2003

[QPBU04] B. Quoitin, C. Pelsser, O. Bonaventure, S. Uhlig: *A performance evaluation of BGP-based traffic engineering*; December 2004

[UQ05] S. Uhlig and B. Quoitin: *Tweak-it: BGP-based Interdomain Traffic Engineering for transit ASs*; NGI Networks, 2005, April 2005 Page(s) 75 – 82