;login:
THE MAGAZINE OF USENIX & SAGE
February 2003 • volume 28 • number 1

inside:

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

introduction to the border gateway protocol (BGP)

by Greg Hankins

Greg Hankins is a senior systems engineer with Riverstone Networks and has been designing and deploying service provider and enterprise networks for the past 10 years. An avid Linux user since early 1994, he is best known for his contributions to the Linux Documentation Project in many different roles.

gregh@twoguys.org

This article presents an introduction and overview of BGP, the routing protocol of choice for large-scale IP routing. BGP has gained a reputation of being somewhat of a black art (even blacker than the art of SCSI), and experienced BGP-savvy network engineers are still at a premium. It certainly is a complex protocol that cannot be thoroughly explained in any one article or even a small book, and takes time and experience to master. However, I hope that after reading this article you will at least have a fundamental understanding of the BGP protocol.

Background

BGP is an Exterior Gateway Protocol (EGP). EGPs were developed to provide inter-domain routing between networks called autonomous systems. An autonomous system (AS) is a set of networks and routers under common administration, which are assigned a globally unique number. EGPs have fundamentally different requirements from an Interior Gateway Protocol (IGP) such as OSPF, IS-IS or RIP. IGPs are run within your network to communicate reachability about networks under your control, while EGPs are run at your network border to provide reachability information about networks outside of your control.

Whereas IGPs were designed to scale to a few thousand routes, EGPs were designed to scale to huge numbers of routes and to provide routing policy mechanisms. EGPs primarily make routing decisions based on the path of networks to a particular destination, not on the hops within each of the individual networks the path traverses. Think of it this way: If you are driving across country, you want a map that shows you the interstate highways, not detailed maps of all the cities along the way.

BGP first became an Internet standard in 1989 and was originally defined in RFC 1105. It was then adopted as the EGP of choice for inter-domain routing. The current version, BGP-4, was adopted in 1995 and is defined in RFC 1771. BGP-4 supports Classless Inter-Domain Routing (CIDR) and is the routing protocol that people use today to route between autonomous systems.

It has proven to be scalable, stable, and extensible, and it provides the mechanisms needed to support complex routing policies. When people talk about BGP today, they implicitly mean BGP-4. There is no need to specify the -4 version number because no one uses earlier versions, and very few vendors even still support them.

BGP continues to evolve through the Internet standards work in the IETF IDR working group; the latest draft version is at *http://www.ietf.org/internet-drafts/draft-ietf-idr-bgp4-18*. As the Internet routing requirements change, BGP is extended to continue to provide the knobs and dials needed to control routing information and to support new network requirements. The base RFC has been extended by several RFCs and I-Ds that define new features. Most recently, for example, BGP has been extended to provide support for building MPLS-based VPNs and a graceful recovery mechanism from router crashes.

BGP is the only protocol that is suitable for use between autonomous systems because of the inherent support for routing policies that the path attributes provide.

Protocol Details

We call BGP a path vector protocol because it stores routing information as a combination of a destination and attributes of the path to that destination. The protocol uses a deterministic route selection process to select the best route from multiple feasible routes using the path attributes as criteria. Characteristics such as delay, link utilization, or router hops are not considered in this process. We will see how BGP uses these path attributes later on.

Unlike most IGP protocols, BGP only sends a full routing update once when a BGP session is established and then only sends incremental changes. BGP only recalculates routing information relative to these updates; there is no regular process that must update all of its routing information like the SPF calculations in OSPF or IS-IS. Although IGP convergence may be faster, an IGP will simply not scale to support the number of routes needed for inter-domain routing. IGPs also lack the path attributes that BGP carries, which are essential for selecting the best route and building routing policies.

BGP is the only protocol that is suitable for use between autonomous systems because of the inherent support for routing policies that the path attributes provide. These policies allow you to accept, reject, or change routing information based on the path attributes before such information is used to make forwarding decisions. This gives network operators the ability to control routing information according to their particular needs, including rejecting routing information they might not want. Neither OSPF or IS-IS provide policies to reject or change routing information and thus should not be run between autonomous systems. RIP provides such policies, but suffers from even greater scalability issues.

BGP runs in two modes: EBGP and IBGP. EBGP (Exterior BGP) is run between routers in different autonomous systems, and IBGP (Interior BGP) is run between routers in the same autonomous system. It is necessary to run IBGP between backbone routers in order to provide each of them with a complete view of the routing table. This allows traffic to take the best exit point out of your network.

Protocol Mechanics

BGP uses TCP to establish a reliable connection between two BGP speakers, or peers, on port 179. Exactly one TCP session is established between each peer for each BGP session. No routing information can be exchanged until the TCP session has been established. This implies that each BGP speaker must have working IP connectivity between them first, which is usually provided by a directly connected interface or the IGP.

Since it uses TCP, BGP does not need to worry about transport issues such as data sequencing or fragmentation. TCP takes care of these problems and simply hands BGP a reliable pipe for transporting its messages. For added security, MD5 signatures can be used to authenticate each TCP segment.

One definition is needed before we look at the protocol in more detail. An *IP prefix* is simply an IP network with its mask: for example, 10.0.0.0/8. It is technically incorrect to call this an IP route as it pertains to BGP because the prefix only specifies the network and mask, not how to reach it.

MESSAGES

BGP uses five defined message types to communicate its routing information. You don't need to know all of the details about them, but it is helpful to at least know each one and how BGP uses it. Each message uses a fixed header, with a variable type-and-length field. This allows multiple BGP messages to be sent within one TCP segment.

OPEN

The OPEN message is the first message that is exchanged between BGP peers after the TCP session is established. It contains each peer's configuration information and handles any negotiations on exactly which BGP extensions each peer supports. Only one of these is sent at the beginning of the session.

UPDATE

These messages carry the actual routing information. UPDATE messages are used to signal new routing updates and to withdraw old routing information. The IP prefix, along with the path attributes, is sent in these messages. BGP is very efficient about how it transmits the routing information. If multiple prefixes share the exact same path attributes, BGP will send multiple prefixes in an UPDATE message with one copy of the associated path attributes. UPDATE messages are sent as often as they need to be, but remember that BGP only sends a complete routing update at the beginning of the session. Then it only sends incremental changes.

KEEPALIVE

The KEEPALIVE message is simply a message that keeps the BGP session up, indicating that the router is still operating normally. A timer is reset each time a KEEPALIVE is received. If none are received within a predefined time period, the timer expires. At this point, the other router is presumed to be unreachable and the peering session is torn down.

NOTIFICATION

A NOTIFICATION message is used to communicate errors. All error types are predefined, making it very easy to troubleshoot a BGP peering problem. The NOTIFICATION message simply contains exactly what was wrong in form of an error code and an error subcode. After it is sent, the BGP session is closed.

ROUTE-REFRESH

The ROUTE-REFRESH message is not defined in the base BGP specification but as an extension to BGP. However, it has been so widely implemented that it only makes sense to mention this message here. This message is used to request a complete retransmission of a peer's routing information without tearing down and reestablishing the BGP session (remember, BGP only sends a complete routing update once).

Using this extension, routing policy changes can be made without storing an unmodified copy of the peer's routes on the local router, which in turn saves RAM and resources. If a change is made to the routing policies, then a route refresh is requested from the peer, causing the new policy to take effect.

The ROUTE-REFRESH message was designed to be protocol independent. Thus, for example, you can request a retransmission of a peer's IPv4 unicast routes but none of its IPv6 routes.

BGP is very efficient about how it transmits the routing information.

State Machine

BGP uses a Finite-State Machine with carefully defined events and state transitions. This allows BGP to know exactly what to do next, whenever anything happens. Briefly, the FSM starts out in the Idle state, then transitions through several states as a TCP connection is established, and options are negotiated. Finally, BGP reaches the Established state, and starts exchanging routing information in form of UPDATE messages. If any errors occur along the way, or malformed or invalid routing information is received, BGP shuts down the session and goes back to the Idle state. One FSM is maintained for each BGP session, allowing many peers to exist in different states.

Path Attributes and Routing Policies

No article on BGP would be complete without mentioning some of the path attributes that BGP uses to communicate details about each path to a destination. Though the details can be, well, detailed, I will provide enough to give you the general idea of each. Routing policies can be used to accept, reject, or even change path attributes on routing information that is sent or received between BGP peers. Routing polices are the keys that unlock extremely powerful control over routing information, which can be as granular or as coarse as you need it. For example, you can apply a policy to a single IP prefix (say, 192.168.0.0/24) or all 110,000 routes received from a particular peer.

Let's discuss the most common path attributes briefly, and how each can be used to control routing information.

AS Path

The AS Path is an ordered list of all autonomous systems that an IP prefix has traversed, from right to left. Each autonomous system is represented by an integer from 1 to 65535, and is assigned by the regional registry (ARIN, RIPE, APNIC, etc.). The shorter the AS Path, the more desirable it is. For example, if a prefix has the path "7018 3356 4355" we would know the following: it was originated by AS 4355, it traversed AS 3356, then AS 7018. If your router had two paths to the same destination that were "7018 3356 4355" and "1 4355", it would choose the second because it is a shorter path. You can influence how other networks reach an IP network by making the AS Path shorter or longer, which then makes it more or less desirable.

AS Path regular expressions can be used for matching in routing policies. This gives you a very powerful classification mechanism to make routing decisions. For example, say you wanted to black hole any IP networks owned by Microsoft. A simple regular expression that denies any prefix whose AS Path ends in 8070 (Microsoft's AS number) from entering the routing table can easily be applied.

Local Preference

BGP provides bi-directional metrics for selecting the best route. The Local Preference attribute is used to control how traffic leaves your network, and it is represented as a 4-byte integer. A higher Local Preference means a higher degree of preference. Using a clever combination of this metric, network operators commonly set up primary and backup egress paths. Local Preference tuning is also a popular way of load sharing transit connections.

Multi-Exit Discriminator (MED)

This path attribute is used to control how traffic enters your network. Though the AS Path can be used at a coarse level, MEDs provide finer control. A MED is represented

as a 4-byte integer. A lower number means a higher degree of preference, opposite of the Local Preference. MEDs can also be used to provide redundancy and load sharing, with one caveat: They are only compared between the same autonomous system. Because each network's policy is different, comparing MEDs among different autonomous systems would be like comparing apples and oranges, resulting in some very strange routing. In some cases, the IGP metric can be used as the MED, optimizing the ingress traffic flow even further.

COMMUNITIES

One of the hardest problems in BGP is selecting a few prefixes out of many. Selecting one or all is easy, but how do you choose 500 particular prefixes from 110,000? Communities are simply arbitrary tags that are associated with a prefix. Using communities is a popular way to tag certain prefixes for later matching in a policy. This type of tagging is extremely flexible and, most importantly, dynamic. By using communities, you don't need to rely on lists of IP prefixes that must be updated by hand every time a change in the network is made.

The classification possibilities are endless. For example, some operators assign all prefixes from the same geographic location to the same community. This allows them to make routing policy decision on, say, all networks that are in a particular city, or even continent, without knowing or caring exactly which IP addresses they might be.

When to Use BGP

One of the most important decisions to be made is whether to even run BGP. A lot of thought must be put into this decision, and you should weigh the benefits and drawbacks very carefully. Simply using static routes can save time and a lot of complexity.

As you can see, BGP is a complex protocol, and configuring the routers to run BGP is only a tiny step in implementing BGP in your network. Your network engineers and operators must understand the protocol in great detail in order to make correct design and implementation decisions and to maintain and troubleshoot the network. Additionally, you must understand how to build routing policies, as these are essential in making BGP do what you want. After all, BGP does exactly what you ask it to do, not what you mean it to do.

Here are some very simplified guidelines to help you determine if BGP is right for your network.

- If your network is single-homed to an ISP, you don't need to run BGP. Just use static routing between your network and the ISP for simplicity.
- If your network is multi-homed to one or more ISPs, you might need to run BGP. Again, if one or more static routes will work, each service provider can configure their routers so that traffic is shared between your transit links.
- If your network is multi-homed and you are designing your network for redundancy, load sharing, or want to optimize routing between your Internet transit links, you will need to build BGP routing policies to do this. In this case you need to run BGP with each ISP.

Further Reading and Resources

Unfortunately, many BGP topics were not covered in this article. BGP offers much more than what was discussed here, and we have barely scratched the surface of its capabilities. Hierarchy and scaling, capabilities such as authentication and graceful restart, and many other necessary details are all fascinating topics for further study.

Fully understanding all aspects of the protocol will give you the ability to design, deploy, and scale complex and resilient networks.

All related RFCs and I-Ds can be found on the IDR Working Group Web page (*http://www.ietf.org/html.charters/idr-charter.html*), and this is a good place to start reading if you are interested in the gory details. For added fun you can join the mailing list, where editorial changes and technical issues about the protocol are discussed.

Don't send operational questions to the IDR list though; it is strictly used for work related to developing the protocol itself. For technical questions, you are better off joining one of the many lists that are run by network operators for the purpose of discussing operational issues. A great list of mailing lists can be found on NANOG's ISP Resources page (*http://www.nanog.org/isp.html#lists*).

Additionally, two very good books have been written that cover BGP very nicely:

Bassam Halabi and Danny McPherson, *Internet Routing Architectures*, 2d ed. (Indianapolis: Cisco Press, 2000).

This book is still considered to be the BGP bible. It is an excellent and in-depth book, with many simple and complex practical examples. The configurations are Cisco specific, but the principles apply to any vendor.

John W. Stewart, *BGP4: Inter-Domain Routing in the Internet* (Reading, MA: Addison Wesley Longman, 1999).

A short and easy, vendor-neutral introduction and overview of BGP, Stewart's book does not have many practical examples, but it sure is great to keep handy as a BGP reference.

Finally, if you are interested in learning more about BGP there are many software implementations available for you to use. All you need is a PC to get started:

GNU Zebra (*http://www.zebra.org/*)

Zebra is a fully functional routing engine that runs on most UNIX systems. It supports BGP, OSPF, and RIP for IPv4 and IPv6. This one also features a Cisco-like CLI and is probably the best one to use for learning about routing protocols.

MRT - Multi-Threaded Routing Toolkit (*http://www.mrtd.net/*)

MRT is a freely available implementation that supports IPv4 and IPv6 routing protocols. BGP and RIP are supported. It runs on most UNIXes and MS Windows, too.

GateD (*http://www.gated.org/*)

GateD provides a full implementation of IPv4 and IPv6 routing protocols such as BGP, OSPF, IS-IS and RIP. GateD code is available at no cost to universities and research institutions. Commercial users must pay for a license.