INTERNET-DRAFT
draft-ietf-idr-bgp-analysis-02.txt
Category
Expires: October 2003

David Meyer
Keyur Patel

Informational
April 2003

BGP-4 Protocol Analysis
<draft-ietf-idr-bgp-analysis-02.txt>

Status of this Document

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This document is a product of an individual. Comments are solicited and should be addressed to the author(s).

Copyright Notice

INTERNET-DRAFT Expires: October 2003 April 2003

Abstract

The purpose of this report is to document how the requirements for advancing a routing protocol from Draft Standard to full Standard have been satisfied by Border Gateway Protocol version 4 (BGP-4).

This report satisfies the requirement for "the second report", as described in Section 6.0 of RFC 1264 [RFC1264]. In order to fulfill the requirement, this report augments RFC 1774 [RFC1774] and summarizes the key features of BGP protocol, and analyzes the protocol with respect to scaling and performance.

Table of Contents

1. Introduction

BGP-4 is an inter-autonomous system routing protocol designed for TCP/IP internets. Version 1 of the BGP protocol was published in RFC 1105 [RFC1105]. Since then BGP versions 2, 3, and 4 have been developed. Version 2 was documented in RFC 1163 [RFC1163]. Version 3 is documented in RFC 1267 [RFC1267]. Version 4 is documented in the [BGP4] (version 4 of BGP will hereafter be referred to as BGP). The changes between versions are explained in Appendix A of [BGP4]. Possible applications of BGP in the Internet are documented in RFC 1772 [RFC1772].

2. Key Features and algorithms of the BGP protocol

This section summarizes the key features and algorithms of the BGP protocol. BGP is an inter-autonomous system routing protocol; it is designed to be used between multiple autonomous systems. BGP assumes that routing within an autonomous system is done by an intra-autonomous system routing protocol. BGP does not make any assumptions about intra-autonomous system routing protocols deployed within the various autonomous systems. Specifically, BGP does not require all autonomous systems to run the same intra-autonomous system routing protocol (i.e., interior gateway protocol or IGP).

Finally, note that BGP is a real inter-autonomous system routing

protocol, and as such it imposes no constraints on the underlying Internet topology. The information exchanged via BGP is sufficient to construct a graph of autonomous systems connectivity from which routing loops may be pruned and many routing policy decisions at the autonomous system level may be enforced.

2.1. Key Features

The key features of the protocol are the notion of path attributes and aggregation of network layer reachability information (NLRI). Path attributes provide BGP with flexibility and extensibility. Path attributes are partitioned into well-known and optional. The provision for optional attributes allows experimentation that may involve a group of BGP routers without affecting the rest of the Internet. New optional attributes can be added to the protocol in much the same way that new options are added to, say, the Telnet

protocol [RFC854].

One of the most important path attributes is the Autonomous System Path, or AS_PATH. AS reachability information traverses the Internet, this information is augmented by the list of autonomous systems that have been traversed thus far, forming the AS_PATH. The AS_PATH allows straightforward suppression of the looping of routing information. In addition, the AS_PATH serves as a powerful and versatile mechanism for policy-based routing.

BGP enhances the AS_PATH attribute to include sets of autonomous systems as well as lists via the AS_SET attribute. This extended format allows generated aggregate routes to carry path information from the more specific routes used to generate the aggregate. It should be noted however, that as of this writing, AS_SETs are rarely used in the Internet [ROUTEVIEWS].

2.2. BGP Algorithms

BGP uses an algorithm that is neither a pure distance vector algorithm or a pure link state algorithm. It is instead a modified distance vector algorithm that uses path information to avoid traditional distance vector problems. Each route within BGP pairs destination with path information to that destination. Path information (also known as AS_PATH information) is stored within the AS_PATH attribute in BGP. This allows BGP to reconstruct large portions of overall topology whenever required.

BGP uses an incremental update strategy in order to conserve bandwidth and processing power. That is, after initial exchange of complete routing information, a pair of BGP routers exchanges only changes to that information. Such an incremental update design requires reliable transport between a pair of BGP routers to function correctly. BGP solves this problem by using TCP for reliable transport.

In addition to incremental updates, BGP has added the concept of route aggregation so that information about groups of networks may be aggregated and sent as a single Network Layer Reachability (NLRI).

Finally, note that BGP is a self-contained protocol. That is, BGP specifies how routing information is exchanged both between BGP speakers in different autonomous systems, and between BGP speakers within a single autonomous system.

2.3. BGP Finite State Machine (FSM)

The BGP FSM is a set of rules that are applied to a BGP speaker's set of configured peers for the BGP operation. A BGP implementation requires that a BGP speaker must connect to and listen on TCP port 179 for accepting any new BGP connections from its peers. The BGP Finite State Machine, or FSM, must be initiated and maintained for each new incoming and outgoing peer connections. However, in steady state operation, there will be only one BGP FSM per connection per peer.

There may exist a temporary period where in a BGP peer may have separate incoming and outgoing connections resulting into two different BGP FSMs for a peer (instead of one). This can be resolved

following BGP connection collision rules defined in the [BGP4].

Following are different states of BGP FSM for its peers:

IDLE:	State when BGP peer refuses any incoming connections.
CONNECT:	State in which BGP peer is waiting for its TCP connection to be completed.
ACTIVE:	State in which BGP peer is trying to acquire a peer by listening and accepting TCP connection.
OPENSENT:	BGP peer is waiting for OPEN message from its peer.
OPENCONFIRM:	BGP peer is waiting for KEEPALIVE or NOTIFICATION message from its peer.
ESTABLISHED:	BGP peer connection is established and exchanges UPDATE, NOTIFICATION, and KEEPALIVE messages with its peer.

There are different BGP events that operate on above mentioned states of BGP FSM for its peers. These BGP events are used for initiating and terminating peer connections. They also assist BGP in identifying any persistent peer connection oscillations and provide a mechanism for controlling them.

Following are different BGP events:

Manual Start:	Manually start the peer connection.

Manual Stop:	Manually stop the peer connection.
Automatic Start:	Local system automatically starts the peer connection.
Manual start with passive TCP flag:	Local system administrator manually starts the peer connection with peer in passive mode.

Automatic start with passive TCP flag:	Local system administrator automatically starts the peer connection with peer in passive mode.
Automatic start with bgp_stop_flap option set:	Local system administrator automatically starts the peer connection with peer oscillation damping enabled.
Automatic start with bgp_stop_flap option set and passive TCP establishment option set:	Local system administrator automatically starts the peer connection with peer oscillation damping enabled and with peer in passive mode.
Automatic stop:	Local system automatically stops the BGP connection.

Both, Manual Start and Manual Stop are mandatory BGP events. All other events are optional.

3. BGP Capabilities

The BGP Capability mechanism [RFC2842] provides an easy and flexible way to introduce new features within the protocol. In particular, the BGP capability mechanism allows peers to negotiate various optional features during startup. This allows the base BGP protocol to contain only essential functionality, while at the same time providing a flexible mechanism for signaling protocol extensions.

4. BGP Persistent Peer Oscillations

Ideally, whenever a BGP speaker detects an error in any peer connection, it shuts down the peer and changes its FSM state to IDLE. BGP speaker requires a Start event to re-initiate its idle peer connection. If the error remains persistent and BGP speaker generates Start event automatically then it may result in persistent peer flapping. However, although peer oscillation is found to be wide-spread in BGP implementations, methods for preventing persistent peer oscillations are outside the scope of base BGP protocol specification.

5. BGP Performance characteristics and Scalability

In this section, we provide "order of magnitude" answers to the questions of how much link bandwidth, router memory and router CPU cycles the BGP protocol will consume under normal conditions. In particular, we will address the scalability of BGP and its limitations.

It is important to note that BGP does not require all the routers within an autonomous system to participate in the BGP protocol. In particular, only the border routers that provide connectivity between the local autonomous system and their adjacent autonomous systems need participate in BGP. The ability to constrain the set of BGP speakers is one way to address scaling issues.

5.1. Link bandwidth and CPU utilization

Immediately after the initial BGP connection setup, BGP peers exchange complete set of routing information. If we denote the total number of routes in the Internet by N, the mean AS distance of the Internet by M (distance at the level of an autonomous system, expressed in terms of the number of autonomous systems), the total number of autonomous systems in the Internet by A, and assume that the networks are uniformly distributed among the autonomous systems, then the worst case amount of bandwidth consumed during the initial exchange between a pair of BGP speakers is

$$MR = O(N + M * A)$$

The following table illustrates the typical amount of bandwidth consumed during the initial exchange between a pair of BGP speakers based on the above assumptions (ignoring bandwidth consumed by the BGP Header). For purposes of the estimates here, we will calculate MR = 4 * (N + (M * A)).

# NLRI	Mean AS Distance	# AS's	Bandwidth (MR)
40,000	15	400	184,000 bytes
100,000	10	10,000	800,000 bytes
120,000	10	15,000	1,080,000 bytes
140,000	15	20,000	1,760,000 bytes

[note that most of this bandwidth is consumed by the NLRI exchange]

BGP was created specifically to reduce the size of the set of NLRI entries which have to be carried and exchanged by border routers. The aggregation scheme, defined in RFC 1519 [RFC1519], describes the provider-based aggregation scheme in use in today's Internet.

Due to the advantages of advertising a few large aggregate blocks instead of many smaller class-based individual networks, it is difficult to estimate the actual reduction in bandwidth and processing that BGP has provided over BGP-3. If we simply enumerate all aggregate blocks into their individual class-based networks, we would not take into account "dead" space that has been reserved for future expansion. The best metric for determining the success of BGP's aggregation is to sample the number NLRI entries in the globally connected Internet today and compare it to projected growth rates before BGP was deployed.

At the time of this writing, the full set of exterior routes carried by BGP is approximately 120,000 network entries [ROUTEVIEWS].

5.1.1. CPU utilization

An important and fundamental feature of BGP is that BGP's CPU utilization depends only on the stability of the Internet. If the Internet is stable, then the only link bandwidth and router CPU cycles consumed by BGP are due to the exchange of the BGP KEEPALIVE messages. The KEEPALIVE messages are exchanged only between peers. The suggested frequency of the exchange is 30 seconds. The KEEPALIVE

messages are quite short (19 octets), and require virtually no processing. As a result, the bandwidth consumed by the KEEPALIVE

messages is about 5 bits/sec. Operational experience confirms that the overhead (in terms of bandwidth and CPU) associated with the KEEPALIVE messages should be viewed as negligible.

During periods of Internet instability, changes to the reachability information are passed between routers in UPDATE messages. If we denote the number of routing changes per second by C, then in the worst case the amount of bandwidth consumed by the BGP can be expressed as O(C * M). The greatest overhead per UPDATE message occurs when each UPDATE message contains only a single network. It should be pointed out that in practice routing changes exhibit strong locality with respect to the AS path. That is, routes that change are likely to have common AS path. In this case, multiple networks can be grouped into a single UPDATE message, thus significantly reducing the amount of bandwidth required (see also Appendix F.1 of [BGP4]).

Since in the steady state the link bandwidth and router CPU cycles consumed by the BGP protocol are dependent only on the stability of the Internet, it follows that BGP should have no scaling problems in the areas of link bandwidth and router CPU utilization. This assumes that as the Internet grows, the overall stability of the inter-AS connectivity of the Internet can be controlled. In particular, while the size of the IPv4 Internet routing table is bounded by O(2^32 * M), (where M is a slow-moving function describing the AS interconnectivity of the network), no such bound can be formulated for the dynamic properties (i.e., stability) of BGP. Finally, since the dynamic properties of the network cannot be quantitatively bounded, stability must be addressed via heuristics such as BGP Route Flap Damping [RFC2439]. Due to the nature of BGP, such damping should be viewed as a matter local to an autonomous system matter (see also Appendix F.2 of [BGP4]).

It may also be instructive to compare bandwidth and CPU requirements of BGP with the Exterior Gateway Protocol (EGP). While with BGP the complete information is exchanged only at the connection establishment time, with EGP the complete information is exchanged periodically (usually every 3 minutes). Note that both for BGP and for EGP the amount of information exchanged is roughly on the order of the number of networks reachable via a peer that sends the

information. Therefore, even if one assumes extreme instabilities of BGP, its worst case behavior will be the same as the steady state behavior of its predecessor, EGP.

Operational experience with BGP showed that the incremental update approach employed by BGP provides qualitative improvement in both bandwidth and CPU utilization when compared with complete periodic updates used by EGP (see also presentation by Dennis Ferguson at the Twentieth IETF, March 11-15, 1991, St. Louis).

5.1.2. Memory requirements

To quantify the worst case memory requirements for BGP, we denote the total number of networks in the Internet by N, the mean AS distance of the Internet by M (distance at the level of an autonomous system, expressed in terms of the number of autonomous systems), the total number of autonomous systems in the Internet by A, and the total number of BGP speakers that a system is peering with by K (note that K will usually be dominated by the total number of the BGP speakers within a single autonomous system). Then the worst case memory requirements (MR) can be expressed as

$$MR = O((N + M * A) * K)$$

It is interesting to note that prior to the introduction of BGP in the NSFNET Backbone, memory requirements on the NSFNET Backbone routers running EGP were on the order of O(N *K).

Since a mean AS distance M is a slow moving function of the interconnectivity ("meshiness") of the Internet, for all practical purposes the worst case router memory requirements are on the order of the total number of networks in the Internet times the number of peers the local system is peering with. We expect that the total number of networks in the Internet will grow much faster than the average number of peers per router. As a result, BGP's memory scaling properties are linearly related to the total number of networks in the Internet.

The following table illustrates typical memory requirements of a

router running BGP. It is assumed that each network is encoded as four bytes, each AS is encoded as two bytes, and each networks is reachable via some fraction of all of the peers (# BGP peers/per net). For purposes of the estimates here, we will calculate MR = ((N*4) + (M*A)*2) * K.

# Networks	Mean AS Distance	# AS's	# BGP peers/per net	Memory Req (MR)
100,000	20	3,000	20	1,040,000
100,000	20	15,000	20	1,040,000
120,000	10	15,000	100	75,000,000
140,000	15	20,000	100	116,000,000

In analyzing BGP's memory requirements, we focus on the size of the

forwarding table (and ignoring implementation details). In particular, we derive upper bounds for the size of the forwarding table. For example, at the time of this writing, the forwarding tables of a typical backbone router carry on the order of 120,000 entries. Given this number, one might ask whether it would be possible to have a functional router with a table that will have 1,000,000 entries. Clearly the answer to this question is independent of BGP. Interestingly, in his review of the BGP protocol for the BGP review committee in March of 1990, Paul Tsuchiya noted that "BGP does not scale well. This is not really the fault of BGP. It is the fault of the flat IP address space. Given the flat IP address space, any routing protocol must carry network numbers in its updates." The introduction of the provider based aggregation schemes (e.g., RFC 1519 [RFC1519]) have sought to address this issue, to the extent possible, within the context of current addressing architectures.

6. BGP Policy Expressiveness and its Implications

BGP is unique among deployed IP routing protocols in that routing is determined using semantically rich routing policies. Although routing policies are usually the first thing that comes to a network operator's mind concerning BGP, it is important to note that the

languages and techniques for specifying BGP routing policies are not actually a part of the protocol specification (see RFC 2622 [RFC2622] for an example of such a policy language). In addition, the BGP specification contains few restrictions, either explicitly or implicitly, on routing policy languages. These languages have typically been developed by vendors and have evolved through interactions with network engineers in an environment lacking vendor-independent standards.

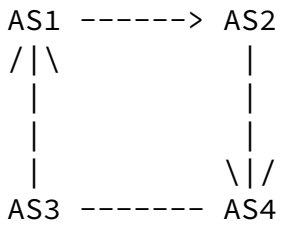
The complexity of typical BGP configurations, at least in provider networks, has been steadily increasing. Router vendors typically provided hundreds of special commands for use in the configuration of BGP, and this command set is continually expanding. For example, BGP communities [RFC1997] allow policy writers to selectively attach tags to routes and use these to signal policy information to other BGP-speaking routers. Many providers allow customers, and sometimes peers, to send communities that determine the scope and preference of their routes. These developments have more and more given the task of writing BGP configurations aspects associated with open-ended programming. This has allowed network operators to encode complex policies in order to address many unforeseen situations, and has opened the door for a great deal of creativity and experimentation in routing policies. This policy flexibility is one of the main reasons

why BGP is so well suited to the commercial environment of the current Internet.

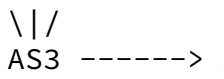
However, this rich policy expressiveness has come with a cost that is often not recognized. In particular, it is possible to construct locally defined routing policies that can lead to unexpected global routing anomalies such as (unintended) nondeterminism and to protocol divergence. If the interacting policies causing such anomalies are defined in different autonomous systems, then these problems can be very difficult to debug and correct. In the following sections, we describe two such cases relating to the existence (or lack thereof) of stable routings.

6.1. Existence of Unique Stable Routings

One can easily construct sets of policies for which BGP can not guarantee that stable routings are unique. This can be illustrated by the following simple example. Consider the example of four Autonomous Systems, AS1, AS2, AS3, and AS4. AS1 and AS2 are peers, and they provide transit for AS3 and AS4 respectively, Suppose further that AS3 provides transit for AS4 (in this case AS3 is a customer of AS1, and AS4 is a multihomed customer of both AS3 and AS4). AS4 may want to use the link to AS3 as a "backup" link, and sends AS3 a community value that AS3 has configured to lower the preference of AS4's routes to a level below that of its upstream provider, AS1. The intended "backup routing" to AS4 is illustrated here:



That is, the AS3-AS4 link is intended to be used only when the AS2-AS4 link is down (for outbound traffic, AS4 simply gives routes from AS2 a higher local preference). This is a common scenario in today's Internet. But note that this configuration has another stable solution:



In this case, AS3 does not translate the "depref my route" community received from AS4 into a "depref my route" community for AS1, and so if AS1 hears the route to AS4 that transits AS3 it will prefer that route (since AS3 is a customer). This state could be reached, for example, by starting in the "correct" backup routing shown first, bringing down the AS2-AS4 BGP session, and then bringing it back up. In general, BGP has no way to prefer the "intended" solution over the anomalous one, and which is picked will depend on the unpredictable order of BGP messages.

While this example is relatively simple, many operators may fail to recognize that the true source of the problem is that the BGP policies of ASes can interact in unexpected ways, and that these interactions can result in multiple stable routings. One can imagine that the interactions could be much more complex in the real Internet. We suspect that such anomalies will only become more common as BGP continues to evolve with richer policy expressiveness. For example, extended communities provide an even more flexible means of signaling information within and between autonomous systems than is possible with RFC 1997 communities. At the same time, applications of communities by network operators are evolving to address complex issues of inter-domain traffic engineering.

6.2. Existence of Stable Routings

One can also construct a set of policies for which BGP can not guarantee that a stable routing exists (or worse, that a stable routing will ever be found). For example, RFC 3345 [RFC3345] documents several scenarios that lead to route oscillations associated with the use of the Multi-Exit Discriminator or MED,

attribute. Route oscillation will happen in BGP when a set of policies has no solution. That is, when there is no stable routing that satisfies the constraints imposed by policy, then BGP has no choice by to keep trying. In addition, BGP configurations can have a

stable routing, yet the protocol may not be able to find it; BGP can "get trapped" down a blind alley that has no solution.

Protocol divergence is not, however, a problem associated solely with use of the MED attribute. This potential exists in BGP even without the use of the MED attribute. Hence, like the unintended nondeterminism described in the previous section, this type of protocol divergence is an unintended consequence of the unconstrained nature of BGP policy languages.

7. Applicability

In this section we answer the question of which environments is BGP well suited, and for which environments it is not suitable. This question is partially answered in Section 2 of RFC 1771 [RFC1771], which states:

> "To characterize the set of policy decisions that can be enforced using BGP, one must focus on the rule that an AS advertises to its neighbor ASs only those routes that it itself uses. This rule reflects the "hop-by-hop" routing paradigm generally used throughout the current Internet. Note that some policies cannot be supported by the "hop-by-hop" routing paradigm and thus require techniques such as source routing to enforce. For example, BGP does not enable one AS to send traffic to a neighbor AS intending that the traffic take a different route from that taken by traffic originating in the neighbor AS. On the other hand, BGP can support any policy conforming to the "hop-by-hop" routing paradigm. Since the current Internet uses only the "hop-by-hop" routing paradigm and since BGP can support any policy that conforms to that paradigm, BGP is highly applicable as an inter-AS routing protocol for the current Internet."

One of the important points here is that the BGP protocol contains only the functionality that is essential, while at the same time providing a flexible mechanism within the protocol that allow us to extend its functionality. For example, BGP capabilities provide an easy and flexible way to introduce new features within the protocol. Finally, since BGP was designed with flexibility and extensibility in mind, new and/or evolving requirements can be addressed via existing

mechanisms.

To summarize, BGP is well suitable as an inter-autonomous system routing protocol for the IPv4 Internet that is based on IP [RFC791] as the Internet Protocol and "hop-by-hop" routing paradigm. Finally, BGP is equally applicable to IPv6 [RFC2460] internets.

8. Acknowledgments

We would like to thank Paul Traina for authoring previous versions of this document. Tim Griffin and Randy Presuhn also provided many insightful comments on earlier versions of this document.

9. Informative References

[BGP4] Rekhter, Y., T. Li., and S. Hares, Editors, "A Border Gateway Protocol 4 (BGP-4)", draft-ietf-idr-bgp4-19.txt. Work in progress.

[RFC791] "INTERNET PROTOCOL", DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION, RFC 791, September, 1981.

[RFC854] Postel, J. and J. Reynolds, "TELNET PROTOCOL SPECIFICATION", RFC 854, May, 1983.

[RFC1105] Lougheed, K., and Y. Rekhter, "Border Gateway Protocol BGP", RFC 1105, June 1989.

[RFC1163] Lougheed, K., and Rekhter, Y, "Border Gateway Protocol BGP", RFC 1105, June 1990.

[RFC1264] Hinden, R., "Internet Routing Protocol Standardization Criteria", RFC 1264, October 1991.

[RFC1267] Lougheed, K., and Rekhter, Y, "Border Gateway Protocol 3 (BGP-3)", RFC 1105, October 1991.

[RFC1519] Fuller, V., Li. T., Yu J., and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC 1519, September 1993.

[RFC1771] Rekhter, Y., and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[RFC1772] Rekhter, Y., and P. Gross, Editors, "Application of the Border Gateway Protocol in the Internet", RFC 1772, March 1995.

[RFC1997] Chandra. R, and T. Li, "BGP Communities Attribute", RFC 1997, August, 1996.

[RFC2439] Villamizar, C., Chandra, R., and R. Govindan, "BGP Route Flap Damping", RFC 2439, November 1998.

[RFC2460] Deering, S, and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December, 1998.

[RFC2622] C. Alaettinoglu, et al., "Routing Policy Specification Language (RPSL)" RFC 2622, May, 1998.

[RFC2842] Chandra, R. and J. Scudder, "Capabilities Advertisement with BGP-4", RFC 2842, May 2000.

[RFC3345] McPherson, D., Gill, V., Walton, D., and A. Retana, "Border Gateway Protocol (BGP) Persistent Route Oscillation Condition", RFC 3345, August, 2002.

[ROUTEVIEWS] Meyer, D., "The Route Views Project", http://www.routeviews.org

10. Author's Addresses

David Meyer
Email: dmm@maoz.com

Keyur Patel
Cisco Systems
Email: keyupate@cisco.com

11. Full Copyright Statement