BGP-4 Protocol Analysis
<draft-ietf-idr-bgp-analysis-05.txt>

Status of this Document

   This document is an Internet-Draft and is subject to 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/1id-abstracts.html

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

   This document is a product of the IDR Working Group WG. Comments
   should be addressed to the authors, or the mailing list at
   idr@ietf.org.

Copyright Notice

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 [RFC1264]. In order to fulfill the
   requirement, this report augments [RFC1774] and summarizes the key
   features of BGP-4 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-4 protocol was published in
   [[RFC1105]()]. Since then BGP versions 2, 3, and 4 have been developed.
   Version 2 was documented in [[RFC1163]()]. Version 3 is documented in
   [[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 [[RFC1772]()].

   BGP introduced support for Classless InterDomain Routing [[CIDR]()].
   Since earlier versions of BGP lacked the support for CIDR, they are
   considered obsolete and unusable in today's Internet.

[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 also assumes that data
   packets are routed from source towards destination independent of the
   source. 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
   interconnect topology of the autonomous systems. 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. Autonomous System's (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 referred to as a "Path 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.
The path information assist BGP in detecting AS loops thereby
allowing BGP speakers select loop free routes.

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. TCP further assist BGP in limiting congestion to the
advertised window limits.

In addition to incremental updates, BGP has added the concept of

route aggregation so that information about groups of destinations
that use hierarchical address assignment (e.g., CIDR) 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 BGP peers. These BGP events are either mandatory or
optional. They are triggered by the protocol logic as part of the BGP
or using an operator intervention via a configuration interface to
the BGP protocol.

These BGP events are of following types: Optional events linked to
Optional Session attributes, Administrative Events, Timer Events,
TCP Connection based Events, and BGP Message-based Events. Both,
the FSM and the BGP events are explained in details in [BGP4].

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 a BGP speaker to advertise to its
peers during startup various optional features supported by the
speaker (and receive similar information from the peers). 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. Implementation Guidelines

A robust BGP implementation is work conserving. This means that if
the number of prefixes is bound, arbitrarily high levels of route
change can be tolerated with bounded impact on route convergence for

occasional changes in generally stable routes.

A robust implementation of BGP should have the following
characteristics:
1. It is able to operate in almost arbitrarily high levels
    of route flap without loosing peerings (failing to send
    keepalives) or loosing other protocol adjacencies as a
    result of BGP load.

2. Instability of a subset of routes should not affect the
    route advertisements or forwarding associated with the set
    of stable routes.

3. High levels of instability and peers of different CPU speed
    or load resulting in faster or slower processing of routes
    should not cause instability and should have a bounded
    impact on the convergence time for generally stable routes.

Numerous robust BGP implementations exist. Producing a robust
implementation is not a trivial matter but clearly achievable.

6. 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.

6.1. 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 total path attributes (for
all N routes) received from a peer as 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 (P) is

$$BW = O((N + A) * P)$$

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

BGP-4 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 [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-4 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].

6.1.1. **CPU utilization**

An important and fundamental feature of BGP is that BGP's CPU
utilization depends only on the stability of its network which
relates to BGP in terms of BGP UPDATE message announcments. If the
BGP network is stable: all the BGP routers within its network are in
the steady state; 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 the periods of network instability, BGP routers within the
network are generating routing updates that are exchanged using the
BGP UPDATE messages. The greatest overhead per UPDATE message occurs
when each UPDATE message contains only a single network. It should
pointed out that in practice routing changes exhibit strong locality

with respect to the route attributes. That is, routes that change
are likely to have common route attributes. 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]).

6.1.2. 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 unique AS paths as A. Then the worst case memory
requirements (MR) can be expressed as

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

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. We denote average number of routes advertised by
each peer as N, the total number of unique AS paths as A, the mean AS
distance of the Internet as M (distance at the level of an autonomous
system, expressed in terms of the number of autonomous systems),
number of bytes required to store a network as R, and number of bytes
required to store one AS in an AS path as P. 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 * R) + (M * A) * P) * S).

# Networks	Mean AS Distance	# AS's	# BGP peers/per net	Memory Req
(N)	(M)	(A)	(P)	(MR)

| ---------- | ---------------- | ------ | ------------------ | ------------- |
100,000	20	3,000	20	10,400,000
100,000	20	15,000	20	20,000,000
120,000	10	15,000	100	78,000,000
140,000	15	20,000	100	116,000,000

In analyzing BGP's memory requirements, we focus on the size of the
BGP RIB table (and ignoring implementation details). In particular,
we derive upper bounds for the size of the BGP RIB table. For
example, at the time of this writing, the BGP RIB 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 more related to how BGP is
implemented. A robust BGP implementation with a reasonable CPU and
memory should not have issues scaling to such limits.

7. 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 ([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.

7.1. 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
AS2). 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:

```
        AS1 ------> AS2
        /|\          |
         |           |
         |           |
         |          \|/
        AS3 ------- AS4
```
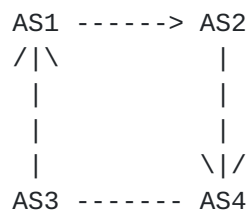
That is, the AS3-AS4 link is intended to be used only when the
AS2-AS4 link is down (for outbound traffic, AS4 simply gives routes
from AS2 a higher local preference). This is a common scenario in
today's Internet. But note that this configuration has another
stable solution:


```
        AS1 ------- AS2
         |           |
         |           |
         |           |
        \|/         \|/
        AS3 ------> AS4
```
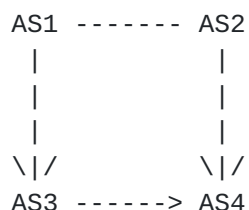

In this case, AS3 does not translate the "depref my route" community
received from AS4 into a "depref my route" community for AS1, and so
if AS1 hears the route to AS4 that transits AS3 it will prefer that
route (since AS3 is a customer). This state could be reached, for
example, by starting in the "correct" 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 [RFC1997] communities. At the same time,
applications of communities by network operators are evolving to
address complex issues of inter-domain traffic engineering.

7.2. 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, [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.

8. 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 BGP [BGP4], 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 any internet that is based on IP [RFC791] as the
   internet protocol and "hop-by-hop" routing paradigm.

9. Acknowledgments

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

10. Security Considerations

   BGP provides flexible mechanisms with varying levels of complexity
   for security purposes. BGP sessions are authenticated using BGP
   session addresses and the assigned AS number. Since BGP sessions use
   TCP (and IP) for reliable transport, BGP sessions are further
   authenticated and secured by any authentication and security
   mechanisms used by TCP and IP.

   BGP uses TCP MD5 option for validating data and protecting against
   spoofing of TCP segments exhanged between its sessions. The usage of
   TCP MD5 option for BGP is described at length in [RFC 2385]. The TCP
   MD5 Key management is discussed in [RFC 3562]. BGP data encryption is
   provided using IPsec mechanism which encrypts the IP payload data
   (including TCP and BGP data). The IPsec mechanism can be used in
   both, the transport mode as well as the tunnel mode. The IPsec
   mechanism is described in [RFC 2406]. Both, the TCP MD5 option and
   the IPsec mechanism are not widely deployed security mechanisms for
   BGP in today's Internet and hence it is difficult to guage their real
   performance impact when using with BGP. However, since both the
   mechanisms are TCP and IP based security mechanisms, the Link
   Bandwidth, CPU utilization and router memory consumed by BGP protocol
   using it would be same as any other TCP and IP based protocols.

   BGP uses IP TTL value to protect its EBGP sessions from any TCP (or
   IP) based CPU intensive attacks. It is a simple mechanism which
   suggests the use of filtering BGP (TCP) segments using the IP TTL
   value carried within the IP header of BGP (TCP) segments exchanged
   between the EBGP sessions. The BGP TTL mechanism is described in
   [BTSH]. Usage of [BTSH] impacts performance in a similar way as using
   any ACL policies for BGP.

   Such flexible TCP and IP based security mechanisms, allow BGP to
   prevent insertion/deletion/modification of BGP data, any snooping of
   the data, session stealing, etc. However, BGP is vulnerable to the
   same security attacks that are present in TCP. The [BGP-VUL]
   explains in depth about the BGP security vulneribility. At the time
   of this writing, several efforts are underway for creating and
   defining an appropriate security infrastructure within the BGP
   protocol to provide authentication and security for its routing
   information; some of which include [SBGP] and [SOBGP].

11. IANA Considerations

   This document presents an analysis of the BGP protocol and hence
   presents no new IANA considerations.

12. References

12.1. Informative References

   [BGP4]   Rekhter, Y., T. Li., and S. Hares, Editors, "A
                   Border Gateway Protocol 4 (BGP-4)",
                   draft-ietf-idr-bgp4-20.txt. Work in progress.

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

   [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.

[RFC1774]   Traina, P., "BGP-4 protocol analysis",
                   RFC 1774, March, 1995.

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

[RFC2622]   Alaettinoglu, C., 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.

[BTSH]   Gill, V., Heasley, J., and D. Meyer, "The BGP TTL
                   Security Hack (BTSH)", draft-gill-btsh-02.txt.
                   Work in progress.

[RFC2385]   Heffernan, A., "Protection of BGP Sessions via
                   the TCP MD5 Signature Option", RFC 2385,
                   August, 1998.

[RFC3562]   Leech, M., "Key Management Considerations for
                   the TCP MD5 Signature Option", RFC 3562,
                   July, 2003.

[RFC2406]   Kent, S., Atkinson, R., "IP Encapsulating Security
                   Payload (ESP)", RFC 2406, November, 1998.

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

[SBGP]   Lynn, C., Mikkelson, J., and K. Seo, "Secure BGP S-BGP",
                   Internet-Draft, Work in Progress.

[soBGP]   White, R., "Architecture and Deployment Considerations for
                   Secure Origin BGP (soBGP)", Internet-Draft, Work in
Progress.

[BGP_VULN]   Murphy, S., "BGP Security Vulnerabilities Analysis",
                   draft-ietf-idr-bgp-vuln-00.txt. work in progress

[13]. Author's Addresses

David Meyer
Email: dmm@1-4-5.net

Keyur Patel
Cisco Systems
Email: keyupate@cisco.com

[14]. Full Copyright Statement

[15](). **Intellectual Property**

   The IETF takes no position regarding the validity or scope of any
   Intellectual Property Rights or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; nor does it represent that it has
   made any independent effort to identify any such rights. Information
   on the procedures with respect to rights in RFC documents can be
   found in [BCP 78]() and [BCP 79]().

   Copies of IPR disclosures made to the IETF Secretariat and any
   assurances of licenses to be made available, or the result of an
   attempt made to obtain a general license or permission for the use of
   such proprietary rights by implementers or users of this
   specification can be obtained from the IETF on-line IPR repository at
   [http://www.ietf.org/ipr]().

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights that may cover technology that may be required to implement
   this standard. Please address the information to the IETF at ietf-
   ipr@ietf.org.

[16](). **Acknowledgement**

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