Network Working Group
Request for Comments: 1771
Obsoletes: 1654
Category: Standards Track

Y. Rekhter
T.J. Watson Research Center, IBM Corp.
T. Li
cisco Systems
Editors

March 1995

A Border Gateway Protocol 4 (BGP-4)

Status of this Memo

Abstract

This document, together with its companion document, "Application of the Border Gateway Protocol in the Internet", define an inter-autonomous system routing protocol for the Internet.

1. Acknowledgements

This document was originally published as RFC 1267 in October 1991, jointly authored by Kirk Lougheed (cisco Systems) and Yakov Rekhter (IBM).

We would like to express our thanks to Guy Almes (ANS), Len Bosack (cisco Systems), and Jeffrey C. Honig (Cornell University) for their contributions to the earlier version of this document.

We like to explicitly thank Bob Braden (ISI) for the review of the earlier version of this document as well as his constructive and valuable comments.

We would also like to thank Bob Hinden, Director for Routing of the Internet Engineering Steering Group, and the team of reviewers he assembled to review the previous version (BGP-2) of this document. This team, consisting of Deborah Estrin, Milo Medin, John Moy, Radia Perlman, Martha Steenstrup, Mike St. Johns, and Paul Tsuchiya, acted with a strong combination of toughness, professionalism, and courtesy.

This updated version of the document is the product of the IETF IDR Working Group with Yakov Rekhter and Tony Li as editors. Certain sections of the document borrowed heavily from IDRP [7], which is the OSI counterpart of BGP. For this credit should be given to the ANSI X3S3.3 group chaired by Lyman Chapin (BBN) and to Charles Kunzinger (IBM Corp.) who was the IDRP editor within that group. We would also like to thank Mike Craren (Proteon, Inc.), Dimitry Haskin (Bay Networks, Inc.), John Krawczyk (Bay Networks, Inc.), and Paul Traina (cisco Systems) for their insightful comments.

We would like to specially acknowledge numerous contributions by Dennis Ferguson (MCI).

The work of Yakov Rekhter was supported in part by the National Science Foundation under Grant Number NCR-9219216.

2. Introduction

The Border Gateway Protocol (BGP) is an inter-Autonomous System routing protocol. It is built on experience gained with EGP as defined in RFC 904 [1] and EGP usage in the NSFNET Backbone as described in RFC 1092 [2] and RFC 1093 [3].

The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of Autonomous Systems (ASs) that reachability information traverses. This information is sufficient to construct a graph of AS connectivity from which routing loops may be pruned and some policy decisions at the AS level may be enforced.

BGP-4 provides a new set of mechanisms for supporting classless interdomain routing. These mechanisms include support for advertising an IP prefix and eliminates the concept of network "class" within BGP. BGP-4 also introduces mechanisms which allow aggregation of routes, including aggregation of AS paths. These changes provide support for the proposed supernetting scheme [8, 9].

To characterize the set of policy decisions that can be enforced using BGP, one must focus on the rule that a BGP speaker advertise to its peers (other BGP speakers which it communicates with) in neighboring 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 neighboring AS intending that the traffic take a different route from that taken by traffic originating in the

neighboring 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.

A more complete discussion of what policies can and cannot be enforced with BGP is outside the scope of this document (but refer to the companion document discussing BGP usage [5]).

BGP runs over a reliable transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing. Any authentication scheme used by the transport protocol may be used in addition to BGP's own authentication mechanisms. The error notification mechanism used in BGP assumes that the transport protocol supports a "graceful" close, i.e., that all outstanding data will be delivered before the connection is closed.

BGP uses TCP [4] as its transport protocol. TCP meets BGP's transport requirements and is present in virtually all commercial routers and hosts. In the following descriptions the phrase "transport protocol connection" can be understood to refer to a TCP connection. BGP uses TCP port 179 for establishing its connections.

This document uses the term 'Autonomous System' (AS) throughout. The classic definition of an Autonomous System is a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASs. Since this classic definition was developed, it has become common for a single AS to use several interior gateway protocols and sometimes several sets of metrics within an AS. The use of the term Autonomous System here stresses the fact that, even when multiple IGPs and metrics are used, the administration of an AS appears to other ASs to have a single coherent interior routing plan and presents a consistent picture of what destinations are reachable through it.

The planned use of BGP in the Internet environment, including such issues as topology, the interaction between BGP and IGPs, and the enforcement of routing policy rules is presented in a companion document [5]. This document is the first of a series of documents planned to explore various aspects of BGP application. Please send comments to the BGP mailing list (bgp@ans.net).

3. Summary of Operation

Two systems form a transport protocol connection between one another. They exchange messages to open and confirm the connection parameters. The initial data flow is the entire BGP routing table. Incremental updates are sent as the routing tables change. BGP does not require periodic refresh of the entire BGP routing table. Therefore, a BGP speaker must retain the current version of the entire BGP routing tables of all of its peers for the duration of the connection. KeepAlive messages are sent periodically to ensure the liveness of the connection. Notification messages are sent in response to errors or special conditions. If a connection encounters an error condition, a notification message is sent and the connection is closed.

The hosts executing the Border Gateway Protocol need not be routers. A non-routing host could exchange routing information with routers via EGP or even an interior routing protocol. That non-routing host could then use BGP to exchange routing information with a border router in another Autonomous System. The implications and applications of this architecture are for further study.

If a particular AS has multiple BGP speakers and is providing transit service for other ASs, then care must be taken to ensure a consistent view of routing within the AS. A consistent view of the interior routes of the AS is provided by the interior routing protocol. A consistent view of the routes exterior to the AS can be provided by having all BGP speakers within the AS maintain direct BGP connections with each other. Using a common set of policies, the BGP speakers arrive at an agreement as to which border routers will serve as exit/entry points for particular destinations outside the AS. This information is communicated to the AS's internal routers, possibly via the interior routing protocol. Care must be taken to ensure that the interior routers have all been updated with transit information before the BGP speakers announce to other ASs that transit service is being provided.

Connections between BGP speakers of different ASs are referred to as "external" links. BGP connections between BGP speakers within the same AS are referred to as "internal" links. Similarly, a peer in a different AS is referred to as an external peer, while a peer in the same AS may be described as an internal peer.

3.1 Routes: Advertisement and Storage

For purposes of this protocol a route is defined as a unit of information that pairs a destination with the attributes of a path to that destination:

- Routes are advertised between a pair of BGP speakers in UPDATE messages: the destination is the systems whose IP addresses are reported in the Network Layer Reachability Information (NLRI) field, and the the path is the information reported in the path attributes fields of the same UPDATE message.

- Routes are stored in the Routing Information Bases (RIBs): namely, the Adj-RIBs-In, the Loc-RIB, and the Adj-RIBs-Out. Routes that will be advertised to other BGP speakers must be present in the Adj-RIB-Out; routes that will be used by the local BGP speaker must be present in the Loc-RIB, and the next hop for each of these routes must be present in the local BGP speaker's forwarding information base; and routes that are received from other BGP speakers are present in the Adj-RIBs-In.

If a BGP speaker chooses to advertise the route, it may add to or modify the path attributes of the route before advertising it to a peer.

BGP provides mechanisms by which a BGP speaker can inform its peer that a previously advertised route is no longer available for use. There are three methods by which a given BGP speaker can indicate that a route has been withdrawn from service:

a) the IP prefix that expresses destinations for a previously advertised route can be advertised in the WITHDRAWN ROUTES field in the UPDATE message, thus marking the associated route as being no longer available for use

b) a replacement route with the same Network Layer Reachability Information can be advertised, or

c) the BGP speaker - BGP speaker connection can be closed, which implicitly removes from service all routes which the pair of speakers had advertised to each other.

3.2 Routing Information Bases

The Routing Information Base (RIB) within a BGP speaker consists of three distinct parts:

a) Adj-RIBs-In: The Adj-RIBs-In store routing information that has been learned from inbound UPDATE messages. Their contents represent routes that are available as an input to the Decision Process.

b) Loc-RIB: The Loc-RIB contains the local routing information that the BGP speaker has selected by applying its local policies to the routing information contained in its Adj-RIBs-In.

c) Adj-RIBs-Out: The Adj-RIBs-Out store the information that the local BGP speaker has selected for advertisement to its peers. The routing information stored in the Adj-RIBs-Out will be carried in the local BGP speaker's UPDATE messages and advertised to its peers.

In summary, the Adj-RIBs-In contain unprocessed routing information that has been advertised to the local BGP speaker by its peers; the Loc-RIB contains the routes that have been selected by the local BGP speaker's Decision Process; and the Adj-RIBs-Out organize the routes for advertisement to specific peers by means of the local speaker's UPDATE messages.

Although the conceptual model distinguishes between Adj-RIBs-In, Loc-RIB, and Adj-RIBs-Out, this neither implies nor requires that an implementation must maintain three separate copies of the routing information. The choice of implementation (for example, 3 copies of the information vs 1 copy with pointers) is not constrained by the protocol.

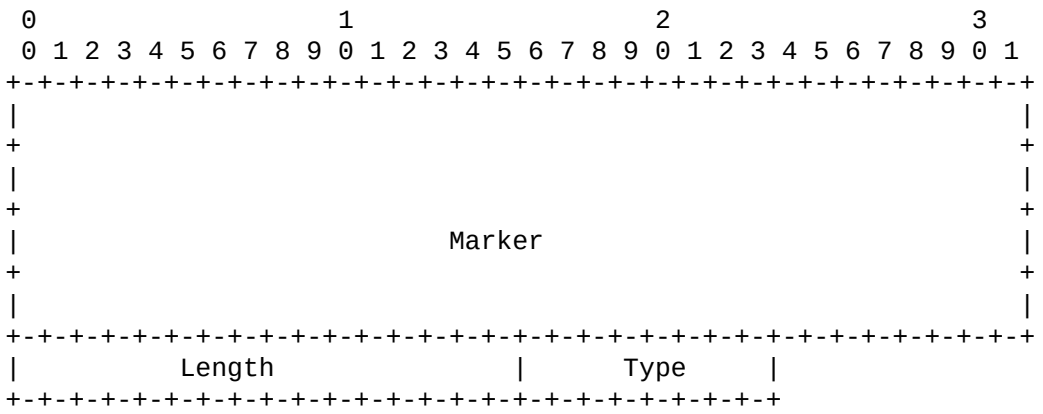
4. Message Formats

This section describes message formats used by BGP.

Messages are sent over a reliable transport protocol connection. A message is processed only after it is entirely received. The maximum message size is 4096 octets. All implementations are required to support this maximum message size. The smallest message that may be sent consists of a BGP header without a data portion, or 19 octets.

4.1 Message Header Format

Each message has a fixed-size header. There may or may not be a data portion following the header, depending on the message type. The layout of these fields is shown below:



Marker:

This 16-octet field contains a value that the receiver of the message can predict. If the Type of the message is OPEN, or if the OPEN message carries no Authentication Information (as an Optional Parameter), then the Marker must be all ones. Otherwise, the value of the marker can be predicted by some a computation specified as part of the authentication mechanism (which is specified as part of the Authentication Information) used. The Marker can be used to detect loss of synchronization between a pair of BGP peers, and to authenticate incoming BGP messages.

Length:

This 2-octet unsigned integer indicates the total length of the message, including the header, in octets. Thus, e.g., it allows one to locate in the transport-level stream the (Marker field of the) next message. The value of the Length field must always be at least 19 and no greater than 4096, and may be further constrained, depending on the message type. No "padding" of extra data after the message is allowed, so the Length field must have the smallest value required given the rest of the message.

Type:

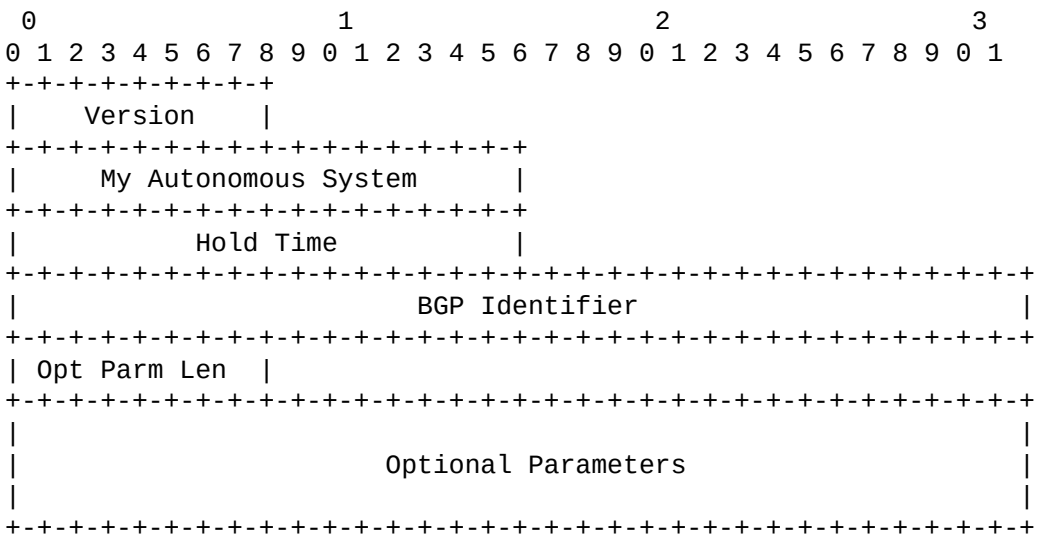
This 1-octet unsigned integer indicates the type code of the message. The following type codes are defined:

1 - OPEN
2 - UPDATE
3 - NOTIFICATION
4 - KEEPALIVE

4.2 OPEN Message Format

After a transport protocol connection is established, the first message sent by each side is an OPEN message. If the OPEN message is acceptable, a KEEPALIVE message confirming the OPEN is sent back. Once the OPEN is confirmed, UPDATE, KEEPALIVE, and NOTIFICATION messages may be exchanged.

In addition to the fixed-size BGP header, the OPEN message contains the following fields:



Version:

This 1-octet unsigned integer indicates the protocol version number of the message. The current BGP version number is 4.

My Autonomous System:

This 2-octet unsigned integer indicates the Autonomous System number of the sender.

Hold Time:

This 2-octet unsigned integer indicates the number of seconds that the sender proposes for the value of the Hold Timer. Upon receipt of an OPEN message, a BGP speaker MUST calculate the value of the Hold Timer by using the smaller of its configured Hold Time and the Hold Time received in the OPEN message. The Hold Time MUST be either zero or at least three seconds. An implementation may reject connections on the basis of the Hold Time. The calculated value indicates the maximum number of seconds that may elapse between the receipt of successive KEEPALIVE, and/or UPDATE messages by the sender.

BGP Identifier:

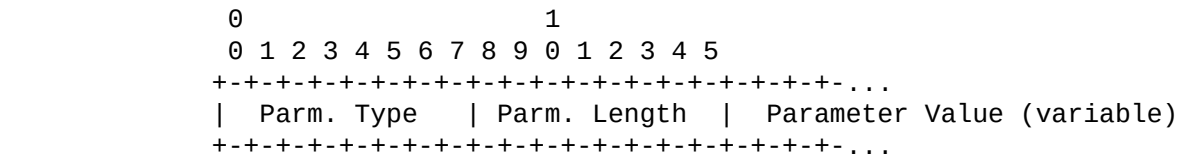
This 4-octet unsigned integer indicates the BGP Identifier of the sender. A given BGP speaker sets the value of its BGP Identifier to an IP address assigned to that BGP speaker. The value of the BGP Identifier is determined on startup and is the same for every local interface and every BGP peer.

Optional Parameters Length:

This 1-octet unsigned integer indicates the total length of the Optional Parameters field in octets. If the value of this field is zero, no Optional Parameters are present.

Optional Parameters:

This field may contain a list of optional parameters, where each parameter is encoded as a <Parameter Type, Parameter Length, Parameter Value> triplet.

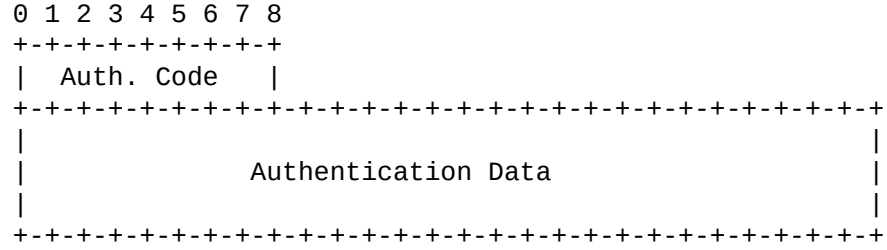


Parameter Type is a one octet field that unambiguously identifies individual parameters. Parameter Length is a one octet field that contains the length of the Parameter Value field in octets. Parameter Value is a variable length field that is interpreted according to the value of the Parameter Type field.

This document defines the following Optional Parameters:

a) Authentication Information (Parameter Type 1):

This optional parameter may be used to authenticate a BGP peer. The Parameter Value field contains a 1-octet Authentication Code followed by a variable length Authentication Data.



Authentication Code:

This 1-octet unsigned integer indicates the authentication mechanism being used. Whenever an authentication mechanism is specified for use within BGP, three things must be included in the specification:

- the value of the Authentication Code which indicates use of the mechanism,
- the form and meaning of the Authentication Data, and
- the algorithm for computing values of Marker fields.

Note that a separate authentication mechanism may be used in establishing the transport level connection.

Authentication Data:

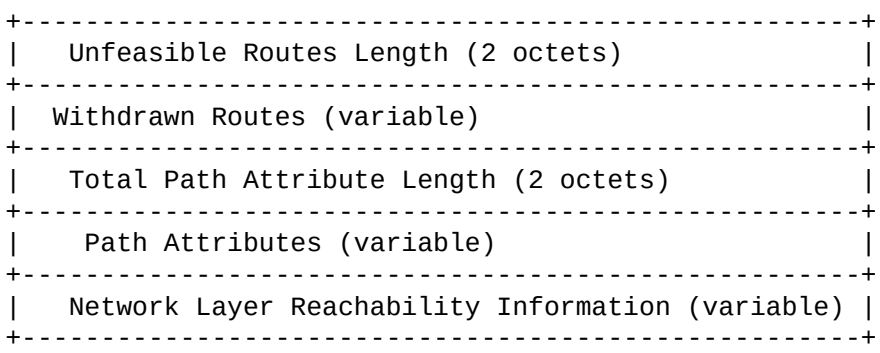
The form and meaning of this field is a variable-length field depend on the Authentication Code.

The minimum length of the OPEN message is 29 octets (including message header).

4.3 UPDATE Message Format

UPDATE messages are used to transfer routing information between BGP peers. The information in the UPDATE packet can be used to construct a graph describing the relationships of the various Autonomous Systems. By applying rules to be discussed, routing information loops and some other anomalies may be detected and removed from inter-AS routing.

An UPDATE message is used to advertise a single feasible route to a peer, or to withdraw multiple unfeasible routes from service (see 3.1). An UPDATE message may simultaneously advertise a feasible route and withdraw multiple unfeasible routes from service. The UPDATE message always includes the fixed-size BGP header, and can optionally include the other fields as shown below:



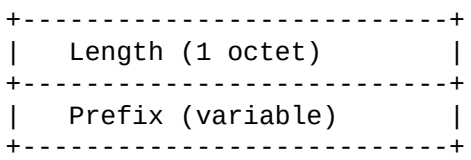
Unfeasible Routes Length:

This 2-octets unsigned integer indicates the total length of the Withdrawn Routes field in octets. Its value must allow the length of the Network Layer Reachability Information field to be determined as specified below.

A value of 0 indicates that no routes are being withdrawn from service, and that the WITHDRAWN ROUTES field is not present in this UPDATE message.

Withdrawn Routes:

This is a variable length field that contains a list of IP address prefixes for the routes that are being withdrawn from service. Each IP address prefix is encoded as a 2-tuple of the form <length, prefix>, whose fields are described below:



The use and the meaning of these fields are as follows:

a) Length:

The Length field indicates the length in bits of the IP address prefix. A length of zero indicates a prefix that matches all IP addresses (with prefix, itself, of zero octets).

b) Prefix:

The Prefix field contains IP address prefixes followed by enough trailing bits to make the end of the field fall on an octet boundary. Note that the value of trailing bits is irrelevant.

Total Path Attribute Length:

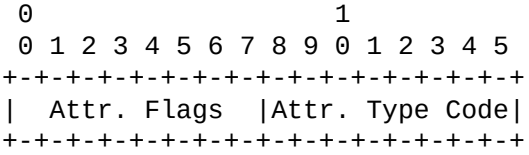
This 2-octet unsigned integer indicates the total length of the Path Attributes field in octets. Its value must allow the length of the Network Layer Reachability field to be determined as specified below.

A value of 0 indicates that no Network Layer Reachability Information field is present in this UPDATE message.

Path Attributes:

A variable length sequence of path attributes is present in every UPDATE. Each path attribute is a triple <attribute type, attribute length, attribute value> of variable length.

Attribute Type is a two-octet field that consists of the Attribute Flags octet followed by the Attribute Type Code octet.



The high-order bit (bit 0) of the Attribute Flags octet is the Optional bit. It defines whether the attribute is optional (if set to 1) or well-known (if set to 0).

The second high-order bit (bit 1) of the Attribute Flags octet is the Transitive bit. It defines whether an optional attribute is transitive (if set to 1) or non-transitive (if set to 0). For well-known attributes, the Transitive bit must be set to 1. (See Section 5 for a discussion of transitive attributes.)

The third high-order bit (bit 2) of the Attribute Flags octet is the Partial bit. It defines whether the information contained in the optional transitive attribute is partial (if set to 1) or complete (if set to 0). For well-known attributes and for optional non-transitive attributes the Partial bit must be set to 0.

The fourth high-order bit (bit 3) of the Attribute Flags octet is the Extended Length bit. It defines whether the Attribute Length is one octet (if set to 0) or two octets (if set to 1). Extended Length may be used only if the length of the attribute value is greater than 255 octets.

The lower-order four bits of the Attribute Flags octet are . unused. They must be zero (and must be ignored when received).

The Attribute Type Code octet contains the Attribute Type Code. Currently defined Attribute Type Codes are discussed in Section 5.

If the Extended Length bit of the Attribute Flags octet is set to 0, the third octet of the Path Attribute contains the length of the attribute data in octets.

If the Extended Length bit of the Attribute Flags octet is set to 1, then the third and the fourth octets of the path attribute contain the length of the attribute data in octets.

The remaining octets of the Path Attribute represent the attribute value and are interpreted according to the Attribute Flags and the Attribute Type Code. The supported Attribute Type Codes, their attribute values and uses are the following:

a) ORIGIN (Type Code 1):

ORIGIN is a well-known mandatory attribute that defines the origin of the path information. The data octet can assume the following values:

Value	Meaning
0	IGP - Network Layer Reachability Information is interior to the originating AS
1	EGP - Network Layer Reachability Information learned via EGP
2	INCOMPLETE - Network Layer Reachability Information learned by some other means

Its usage is defined in 5.1.1

b) AS_PATH (Type Code 2):

AS_PATH is a well-known mandatory attribute that is composed of a sequence of AS path segments. Each AS path segment is represented by a triple <path segment type, path segment length, path segment value>.

The path segment type is a 1-octet long field with the following values defined:

Value	Segment Type
1	AS_SET: unordered set of ASs a route in the UPDATE message has traversed
2	AS_SEQUENCE: ordered set of ASs a route in the UPDATE message has traversed

The path segment length is a 1-octet long field containing the number of ASs in the path segment value field.

The path segment value field contains one or more AS numbers, each encoded as a 2-octets long field.

Usage of this attribute is defined in 5.1.2.

c) NEXT_HOP (Type Code 3):

This is a well-known mandatory attribute that defines the IP address of the border router that should be used as the next hop to the destinations listed in the Network Layer Reachability field of the UPDATE message.

Usage of this attribute is defined in 5.1.3.

d) MULTI_EXIT_DISC (Type Code 4):

This is an optional non-transitive attribute that is a four octet non-negative integer. The value of this attribute may be used by a BGP speaker's decision process to discriminate among multiple exit points to a neighboring autonomous system.

Its usage is defined in 5.1.4.

e) LOCAL_PREF (Type Code 5):

LOCAL_PREF is a well-known discretionary attribute that is a four octet non-negative integer. It is used by a BGP speaker to inform other BGP speakers in its own autonomous system of the originating speaker's degree of preference for an advertised route. Usage of this attribute is described in 5.1.5.

f) ATOMIC_AGGREGATE (Type Code 6)

ATOMIC_AGGREGATE is a well-known discretionary attribute of length 0. It is used by a BGP speaker to inform other BGP speakers that the local system selected a less specific route without selecting a more specific route which is included in it. Usage of this attribute is described in 5.1.6.

g) AGGREGATOR (Type Code 7)

AGGREGATOR is an optional transitive attribute of length 6. The attribute contains the last AS number that formed the aggregate route (encoded as 2 octets), followed by the IP address of the BGP speaker that formed the aggregate route (encoded as 4 octets). Usage of this attribute is described in 5.1.7

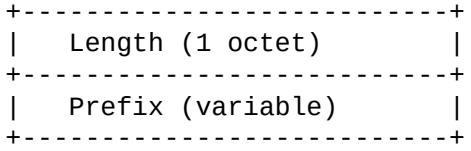
Network Layer Reachability Information:

This variable length field contains a list of IP address prefixes. The length in octets of the Network Layer Reachability Information is not encoded explicitly, but can be calculated as:

UPDATE message Length - 23 - Total Path Attributes Length - Unfeasible Routes Length

where UPDATE message Length is the value encoded in the fixed-size BGP header, Total Path Attribute Length and Unfeasible Routes Length are the values encoded in the variable part of the UPDATE message, and 23 is a combined length of the fixed-size BGP header, the Total Path Attribute Length field and the Unfeasible Routes Length field.

Reachability information is encoded as one or more 2-tuples of the form <length, prefix>, whose fields are described below:



The use and the meaning of these fields are as follows:

a) Length:

The Length field indicates the length in bits of the IP address prefix. A length of zero indicates a prefix that matches all IP addresses (with prefix, itself, of zero octets).

b) Prefix:

The Prefix field contains IP address prefixes followed by enough trailing bits to make the end of the field fall on an octet boundary. Note that the value of the trailing bits is irrelevant.

The minimum length of the UPDATE message is 23 octets -- 19 octets for the fixed header + 2 octets for the Unfeasible Routes Length + 2 octets for the Total Path Attribute Length (the value of Unfeasible Routes Length is 0 and the value of Total Path Attribute Length is 0).

An UPDATE message can advertise at most one route, which may be described by several path attributes. All path attributes contained in a given UPDATE messages apply to the destinations carried in the Network Layer Reachability Information field of the UPDATE message.

An UPDATE message can list multiple routes to be withdrawn from service. Each such route is identified by its destination (expressed as an IP prefix), which unambiguously identifies the route in the context of the BGP speaker - BGP speaker connection to which it has been previously been advertised.

An UPDATE message may advertise only routes to be withdrawn from service, in which case it will not include path attributes or Network Layer Reachability Information. Conversely, it may advertise only a feasible route, in which case the WITHDRAWN ROUTES field need not be present.

4.4 KEEPALIVE Message Format

BGP does not use any transport protocol-based keep-alive mechanism to determine if peers are reachable. Instead, KEEPALIVE messages are exchanged between peers often enough as not to cause the Hold Timer to expire. A reasonable maximum time between KEEPALIVE messages would be one third of the Hold Time interval. KEEPALIVE messages MUST NOT be sent more frequently than one per second. An implementation MAY adjust the rate at which it sends KEEPALIVE

messages as a function of the Hold Time interval.

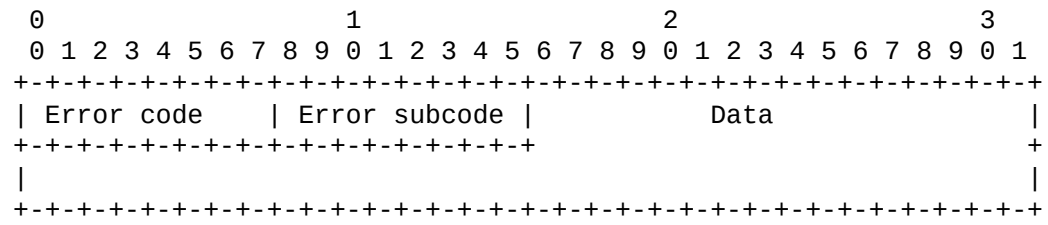
If the negotiated Hold Time interval is zero, then periodic KEEPALIVE messages MUST NOT be sent.

KEEPALIVE message consists of only message header and has a length of 19 octets.

4.5 NOTIFICATION Message Format

A NOTIFICATION message is sent when an error condition is detected. The BGP connection is closed immediately after sending it.

In addition to the fixed-size BGP header, the NOTIFICATION message contains the following fields:



Error Code:

This 1-octet unsigned integer indicates the type of NOTIFICATION. The following Error Codes have been defined:

Error Code	Symbolic Name	Reference
1	Message Header Error	Section 6.1
2	OPEN Message Error	Section 6.2
3	UPDATE Message Error	Section 6.3
4	Hold Timer Expired	Section 6.5
5	Finite State Machine Error	Section 6.6
6	Cease	Section 6.7

Error subcode:

This 1-octet unsigned integer provides more specific information about the nature of the reported error. Each Error Code may have one or more Error Subcodes associated with it.

If no appropriate Error Subcode is defined, then a zero (Unspecific) value is used for the Error Subcode field.

Message Header Error subcodes:

1 - Connection Not Synchronized.
2 - Bad Message Length.
3 - Bad Message Type.

OPEN Message Error subcodes:

1 - Unsupported Version Number.
2 - Bad Peer AS.
3 - Bad BGP Identifier. ’
4 - Unsupported Optional Parameter.
5 - Authentication Failure.
6 - Unacceptable Hold Time.

UPDATE Message Error subcodes:

1 - Malformed Attribute List.
2 - Unrecognized Well-known Attribute.
3 - Missing Well-known Attribute.
4 - Attribute Flags Error.
5 - Attribute Length Error.
6 - Invalid ORIGIN Attribute
7 - AS Routing Loop.
8 - Invalid NEXT_HOP Attribute.
9 - Optional Attribute Error.
10 - Invalid Network Field.
11 - Malformed AS_PATH.

Data:

This variable-length field is used to diagnose the reason for the NOTIFICATION. The contents of the Data field depend upon the Error Code and Error Subcode. See Section 6 below for more details.

Note that the length of the Data field can be determined from the message Length field by the formula:

Message Length = 21 + Data Length

The minimum length of the NOTIFICATION message is 21 octets (including message header).

5. Path Attributes

This section discusses the path attributes of the UPDATE message.

Path attributes fall into four separate categories:

1. Well-known mandatory.
2. Well-known discretionary.
3. Optional transitive.
4. Optional non-transitive.

Well-known attributes must be recognized by all BGP implementations. Some of these attributes are mandatory and must be included in every UPDATE message. Others are discretionary and may or may not be sent in a particular UPDATE message.

All well-known attributes must be passed along (after proper updating, if necessary) to other BGP peers.

In addition to well-known attributes, each path may contain one or more optional attributes. It is not required or expected that all BGP implementations support all optional attributes. The handling of an unrecognized optional attribute is determined by the setting of the Transitive bit in the attribute flags octet. Paths with unrecognized transitive optional attributes should be accepted. If a path with unrecognized transitive optional attribute is accepted and passed along to other BGP peers, then the unrecognized transitive optional attribute of that path must be passed along with the path to other BGP peers with the Partial bit in the Attribute Flags octet set to 1. If a path with recognized transitive optional attribute is accepted and passed along to other BGP peers and the Partial bit in the Attribute Flags octet is set to 1 by some previous AS, it is not set back to 0 by the current AS. Unrecognized non-transitive optional attributes must be quietly ignored and not passed along to other BGP peers.

New transitive optional attributes may be attached to the path by the originator or by any other AS in the path. If they are not attached by the originator, the Partial bit in the Attribute Flags octet is set to 1. The rules for attaching new non-transitive optional attributes will depend on the nature of the specific attribute. The documentation of each new non-transitive optional attribute will be expected to include such rules. (The description of the MULTI_EXIT_DISC attribute gives an example.) All optional attributes (both transitive and non-transitive) may be updated (if appropriate) by ASs in the path.

The sender of an UPDATE message should order path attributes within the UPDATE message in ascending order of attribute type. The receiver of an UPDATE message must be prepared to handle path attributes within the UPDATE message that are out of order.

The same attribute cannot appear more than once within the Path Attributes field of a particular UPDATE message.

5.1 Path Attribute Usage

The usage of each BGP path attributes is described in the following clauses.

5.1.1 ORIGIN

ORIGIN is a well-known mandatory attribute. The ORIGIN attribute shall be generated by the autonomous system that originates the associated routing information. It shall be included in the UPDATE messages of all BGP speakers that choose to propagate this information to other BGP speakers.

5.1.2 AS_PATH

AS_PATH is a well-known mandatory attribute. This attribute identifies the autonomous systems through which routing information carried in this UPDATE message has passed. The components of this list can be AS_SETs or AS_SEQUENCEs.

When a BGP speaker propagates a route which it has learned from another BGP speaker's UPDATE message, it shall modify the route's AS_PATH attribute based on the location of the BGP speaker to which the route will be sent:

a) When a given BGP speaker advertises the route to another BGP speaker located in its own autonomous system, the advertising speaker shall not modify the AS_PATH attribute associated with the route.

b) When a given BGP speaker advertises the route to a BGP speaker located in a neighboring autonomous system, then the advertising speaker shall update the AS_PATH attribute as follows:

1) if the first path segment of the AS_PATH is of type AS_SEQUENCE, the local system shall prepend its own AS number as the last element of the sequence (put it in the leftmost position).

2) if the first path segment of the AS_PATH is of type AS_SET, the local system shall prepend a new path segment of type AS_SEQUENCE to the AS_PATH, including its own AS number in that segment.

When a BGP speaker originates a route then:

a) the originating speaker shall include its own AS number in the AS_PATH attribute of all UPDATE messages sent to BGP speakers located in neighboring autonomous systems. (In this case, the AS number of the originating speaker's autonomous system will be the only entry in the AS_PATH attribute).

b) the originating speaker shall include an empty AS_PATH attribute in all UPDATE messages sent to BGP speakers located in its own autonomous system. (An empty AS_PATH attribute is one whose length field contains the value zero).

5.1.3 NEXT_HOP

The NEXT_HOP path attribute defines the IP address of the border router that should be used as the next hop to the destinations listed in the UPDATE message. If a border router belongs to the same AS as its peer, then the peer is an internal border router. Otherwise, it is an external border router. A BGP speaker can advertise any internal border router as the next hop provided that the interface associated with the IP address of this border router (as specified in the NEXT_HOP path attribute) shares a common subnet with both the local and remote BGP speakers. A BGP speaker can advertise any external border router as the next hop, provided that the IP address of this border router was learned from one of the BGP speaker's peers, and the interface associated with the IP address of this border router (as specified in the NEXT_HOP path attribute) shares a common subnet with the local and remote BGP speakers. A BGP speaker needs to be able to support disabling advertisement of external border routers.

A BGP speaker must never advertise an address of a peer to that peer as a NEXT_HOP, for a route that the speaker is originating. A BGP speaker must never install a route with itself as the next hop.

When a BGP speaker advertises the route to a BGP speaker located in its own autonomous system, the advertising speaker shall not modify the NEXT_HOP attribute associated with the route. When a BGP speaker receives the route via an internal link, it may forward packets to the NEXT_HOP address if the address contained in the attribute is on a common subnet with the local and remote BGP speakers.

5.1.4 MULTI_EXIT_DISC

The MULTI_EXIT_DISC attribute may be used on external (inter-AS) links to discriminate among multiple exit or entry points to the same neighboring AS. The value of the MULTI_EXIT_DISC attribute is a four octet unsigned number which is called a metric. All other factors being equal, the exit or entry point with lower metric should be preferred. If received over external links, the MULTI_EXIT_DISC attribute may be propagated over internal links to other BGP speakers within the same AS. The MULTI_EXIT_DISC attribute is never propagated to other BGP speakers in neighboring AS's.

5.1.5 LOCAL_PREF

LOCAL_PREF is a well-known discretionary attribute that shall be included in all UPDATE messages that a given BGP speaker sends to the other BGP speakers located in its own autonomous system. A BGP speaker shall calculate the degree of preference for each external route and include the degree of preference when advertising a route to its internal peers. The higher degree of preference should be preferred. A BGP speaker shall use the degree of preference learned via LOCAL_PREF in its decision process (see section 9.1.1).

A BGP speaker shall not include this attribute in UPDATE messages that it sends to BGP speakers located in a neighboring autonomous system. If it is contained in an UPDATE message that is received from a BGP speaker which is not located in the same autonomous system as the receiving speaker, then this attribute shall be ignored by the receiving speaker.

5.1.6 ATOMIC_AGGREGATE

ATOMIC_AGGREGATE is a well-known discretionary attribute. If a BGP speaker, when presented with a set of overlapping routes from one of its peers (see 9.1.4), selects the less specific route without selecting the more specific one, then the local system shall attach the ATOMIC_AGGREGATE attribute to the route when propagating it to other BGP speakers (if that attribute is not already present in the received less specific route). A BGP speaker that receives a route with the ATOMIC_AGGREGATE attribute shall not remove the attribute from the route when propagating it to other speakers. A BGP speaker that receives a route with the ATOMIC_AGGREGATE attribute shall not make any NLRI of that route more specific (as defined in 9.1.4) when advertising this route to other BGP speakers. A BGP speaker that receives a route with the ATOMIC_AGGREGATE attribute needs to be cognizant of the fact that the actual path to destinations, as specified in the NLRI of the route, while having the loop-free property, may traverse ASs that are not listed in the AS_PATH

attribute.

5.1.7 AGGREGATOR

AGGREGATOR is an optional transitive attribute which may be included in updates which are formed by aggregation (see Section 9.2.4.2). A BGP speaker which performs route aggregation may add the AGGREGATOR attribute which shall contain its own AS number and IP address.

6. BGP Error Handling.

This section describes actions to be taken when errors are detected while processing BGP messages.

When any of the conditions described here are detected, a NOTIFICATION message with the indicated Error Code, Error Subcode, and Data fields is sent, and the BGP connection is closed. If no Error Subcode is specified, then a zero must be used.

The phrase "the BGP connection is closed" means that the transport protocol connection has been closed and that all resources for that BGP connection have been deallocated. Routing table entries associated with the remote peer are marked as invalid. The fact that the routes have become invalid is passed to other BGP peers before the routes are deleted from the system.

Unless specified explicitly, the Data field of the NOTIFICATION message that is sent to indicate an error is empty.

6.1 Message Header error handling.

All errors detected while processing the Message Header are indicated by sending the NOTIFICATION message with Error Code Message Header Error. The Error Subcode elaborates on the specific nature of the error.

The expected value of the Marker field of the message header is all ones if the message type is OPEN. The expected value of the Marker field for all other types of BGP messages determined based on the presence of the Authentication Information Optional Parameter in the BGP OPEN message and the actual authentication mechanism (if the Authentication Information in the BGP OPEN message is present). If the Marker field of the message header is not the expected one, then a synchronization error has occurred and the Error Subcode is set to Connection Not Synchronized.

If the Length field of the message header is less than 19 or greater than 4096, or if the Length field of an OPEN message is less than the minimum length of the OPEN message, or if the Length field of an UPDATE message is less than the minimum length of the UPDATE message, or if the Length field of a KEEPALIVE message is not equal to 19, or if the Length field of a NOTIFICATION message is less than the minimum length of the NOTIFICATION message, then the Error Subcode is set to Bad Message Length. The Data field contains the erroneous Length field.

If the Type field of the message header is not recognized, then the Error Subcode is set to Bad Message Type. The Data field contains the erroneous Type field.

6.2 OPEN message error handling.

All errors detected while processing the OPEN message are indicated by sending the NOTIFICATION message with Error Code OPEN Message Error. The Error Subcode elaborates on the specific nature of the error.

If the version number contained in the Version field of the received OPEN message is not supported, then the Error Subcode is set to Unsupported Version Number. The Data field is a 2-octet unsigned integer, which indicates the largest locally supported version number less than the version the remote BGP peer bid (as indicated in the received OPEN message).

If the Autonomous System field of the OPEN message is unacceptable, then the Error Subcode is set to Bad Peer AS. The determination of acceptable Autonomous System numbers is outside the scope of this protocol.

If the Hold Time field of the OPEN message is unacceptable, then the Error Subcode MUST be set to Unacceptable Hold Time. An implementation MUST reject Hold Time values of one or two seconds. An implementation MAY reject any proposed Hold Time. An implementation which accepts a Hold Time MUST use the negotiated value for the Hold Time.

If the BGP Identifier field of the OPEN message is syntactically incorrect, then the Error Subcode is set to Bad BGP Identifier. Syntactic correctness means that the BGP Identifier field represents a valid IP host address.

If one of the Optional Parameters in the OPEN message is not recognized, then the Error Subcode is set to Unsupported Optional Parameters.

If the OPEN message carries Authentication Information (as an Optional Parameter), then the corresponding authentication procedure is invoked. If the authentication procedure (based on Authentication Code and Authentication Data) fails, then the Error Subcode is set to Authentication Failure.

6.3 UPDATE message error handling.

All errors detected while processing the UPDATE message are indicated by sending the NOTIFICATION message with Error Code UPDATE Message Error. The error subcode elaborates on the specific nature of the error.

Error checking of an UPDATE message begins by examining the path attributes. If the Unfeasible Routes Length or Total Attribute Length is too large (i.e., if Unfeasible Routes Length + Total Attribute Length + 23 exceeds the message Length), then the Error Subcode is set to Malformed Attribute List.

If any recognized attribute has Attribute Flags that conflict with the Attribute Type Code, then the Error Subcode is set to Attribute Flags Error. The Data field contains the erroneous attribute (type, length and value).

If any recognized attribute has Attribute Length that conflicts with the expected length (based on the attribute type code), then the Error Subcode is set to Attribute Length Error. The Data field contains the erroneous attribute (type, length and value).

If any of the mandatory well-known attributes are not present, then the Error Subcode is set to Missing Well-known Attribute. The Data field contains the Attribute Type Code of the missing well-known attribute.

If any of the mandatory well-known attributes are not recognized, then the Error Subcode is set to Unrecognized Well-known Attribute. The Data field contains the unrecognized attribute (type, length and value).

If the ORIGIN attribute has an undefined value, then the Error Subcode is set to Invalid Origin Attribute. The Data field contains the unrecognized attribute (type, length and value).

If the NEXT_HOP attribute field is syntactically incorrect, then the Error Subcode is set to Invalid NEXT_HOP Attribute. The Data field contains the incorrect attribute (type, length and value). Syntactic correctness means that the NEXT_HOP attribute represents a valid IP host address. Semantic correctness applies only to the external BGP

links. It means that the interface associated with the IP address, as specified in the NEXT_HOP attribute, shares a common subnet with the receiving BGP speaker and is not the IP address of the receiving BGP speaker. If the NEXT_HOP attribute is semantically incorrect, the error should be logged, and the the route should be ignored. In this case, no NOTIFICATION message should be sent.

The AS_PATH attribute is checked for syntactic correctness. If the path is syntactically incorrect, then the Error Subcode is set to Malformed AS_PATH.

If an optional attribute is recognized, then the value of this attribute is checked. If an error is detected, the attribute is discarded, and the Error Subcode is set to Optional Attribute Error. The Data field contains the attribute (type, length and value).

If any attribute appears more than once in the UPDATE message, then the Error Subcode is set to Malformed Attribute List.

The NLRI field in the UPDATE message is checked for syntactic validity. If the field is syntactically incorrect, then the Error Subcode is set to Invalid Network Field.

6.4 NOTIFICATION message error handling.

If a peer sends a NOTIFICATION message, and there is an error in that message, there is unfortunately no means of reporting this error via a subsequent NOTIFICATION message. Any such error, such as an unrecognized Error Code or Error Subcode, should be noticed, logged locally, and brought to the attention of the administration of the peer. The means to do this, however, lies outside the scope of this document.

6.5 Hold Timer Expired error handling.

If a system does not receive successive KEEPALIVE and/or UPDATE and/or NOTIFICATION messages within the period specified in the Hold Time field of the OPEN message, then the NOTIFICATION message with Hold Timer Expired Error Code must be sent and the BGP connection closed.

6.6 Finite State Machine error handling.

Any error detected by the BGP Finite State Machine (e.g., receipt of an unexpected event) is indicated by sending the NOTIFICATION message with Error Code Finite State Machine Error.

6.7 Cease.

In absence of any fatal errors (that are indicated in this section), a BGP peer may choose at any given time to close its BGP connection by sending the NOTIFICATION message with Error Code Cease. However, the Cease NOTIFICATION message must not be used when a fatal error indicated by this section does exist.

6.8 Connection collision detection.

If a pair of BGP speakers try simultaneously to establish a TCP connection to each other, then two parallel connections between this pair of speakers might well be formed. We refer to this situation as connection collision. Clearly, one of these connections must be closed.

Based on the value of the BGP Identifier a convention is established for detecting which BGP connection is to be preserved when a collision does occur. The convention is to compare the BGP Identifiers of the peers involved in the collision and to retain only the connection initiated by the BGP speaker with the higher-valued BGP Identifier.

Upon receipt of an OPEN message, the local system must examine all of its connections that are in the OpenConfirm state. A BGP speaker may also examine connections in an OpenSent state if it knows the BGP Identifier of the peer by means outside of the protocol. If among these connections there is a connection to a remote BGP speaker whose BGP Identifier equals the one in the OPEN message, then the local system performs the following collision resolution procedure:

1. The BGP Identifier of the local system is compared to the BGP Identifier of the remote system (as specified in the OPEN message).

2. If the value of the local BGP Identifier is less than the remote one, the local system closes BGP connection that already exists (the one that is already in the OpenConfirm state), and accepts BGP connection initiated by the remote system.

3. Otherwise, the local system closes newly created BGP connection (the one associated with the newly received OPEN message), and continues to use the existing one (the one that is already in the OpenConfirm state).

Comparing BGP Identifiers is done by treating them as (4-octet long) unsigned integers.

A connection collision with an existing BGP connection that is in Established states causes unconditional closing of the newly created connection. Note that a connection collision cannot be detected with connections that are in Idle, or Connect, or Active states.

Closing the BGP connection (that results from the collision resolution procedure) is accomplished by sending the NOTIFICATION message with the Error Code Cease.

7. BGP Version Negotiation.

BGP speakers may negotiate the version of the protocol by making multiple attempts to open a BGP connection, starting with the highest version number each supports. If an open attempt fails with an Error Code OPEN Message Error, and an Error Subcode Unsupported Version Number, then the BGP speaker has available the version number it tried, the version number its peer tried, the version number passed by its peer in the NOTIFICATION message, and the version numbers that it supports. If the two peers do support one or more common versions, then this will allow them to rapidly determine the highest common version. In order to support BGP version negotiation, future versions of BGP must retain the format of the OPEN and NOTIFICATION messages.

8. BGP Finite State machine.

This section specifies BGP operation in terms of a Finite State Machine (FSM). Following is a brief summary and overview of BGP operations by state as determined by this FSM. A condensed version of the BGP FSM is found in Appendix 1.

Initially BGP is in the Idle state.

Idle state:

In this state BGP refuses all incoming BGP connections. No resources are allocated to the peer. In response to the Start event (initiated by either system or operator) the local system initializes all BGP resources, starts the ConnectRetry timer, initiates a transport connection to other BGP peer, while listening for connection that may be initiated by the remote BGP peer, and changes its state to Connect. The exact value of the ConnectRetry timer is a local matter, but should be sufficiently large to allow TCP initialization.

If a BGP speaker detects an error, it shuts down the connection and changes its state to Idle. Getting out of the Idle state

requires generation of the Start event. If such an event is generated automatically, then persistent BGP errors may result in persistent flapping of the speaker. To avoid such a condition it is recommended that Start events should not be generated immediately for a peer that was previously transitioned to Idle due to an error. For a peer that was previously transitioned to Idle due to an error, the time between consecutive generation of Start events, if such events are generated automatically, shall exponentially increase. The value of the initial timer shall be 60 seconds. The time shall be doubled for each consecutive retry.

Any other event received in the Idle state is ignored.

Connect state:

In this state BGP is waiting for the transport protocol connection to be completed.

If the transport protocol connection succeeds, the local system clears the ConnectRetry timer, completes initialization, sends an OPEN message to its peer, and changes its state to OpenSent.

If the transport protocol connect fails (e.g., retransmission timeout), the local system restarts the ConnectRetry timer, continues to listen for a connection that may be initiated by the remote BGP peer, and changes its state to Active state.

In response to the ConnectRetry timer expired event, the local system restarts the ConnectRetry timer, initiates a transport connection to other BGP peer, continues to listen for a connection that may be initiated by the remote BGP peer, and stays in the Connect state.

Start event is ignored in the Active state.

In response to any other event (initiated by either system or operator), the local system releases all BGP resources associated with this connection and changes its state to Idle.

Active state:

In this state BGP is trying to acquire a peer by initiating a transport protocol connection.

If the transport protocol connection succeeds, the local system clears the ConnectRetry timer, completes initialization, sends an OPEN message to its peer, sets its Hold Timer to a large

value, and changes its state to OpenSent. A Hold Timer value of 4 minutes is suggested.

In response to the ConnectRetry timer expired event, the local system restarts the ConnectRetry timer, initiates a transport connection to other BGP peer, continues to listen for a connection that may be initiated by the remote BGP peer, and changes its state to Connect.

If the local system detects that a remote peer is trying to establish BGP connection to it, and the IP address of the remote peer is not an expected one, the local system restarts the ConnectRetry timer, rejects the attempted connection, continues to listen for a connection that may be initiated by the remote BGP peer, and stays in the Active state.

Start event is ignored in the Active state.

In response to any other event (initiated by either system or operator), the local system releases all BGP resources associated with this connection and changes its state to Idle.

OpenSent state:

In this state BGP waits for an OPEN message from its peer. When an OPEN message is received, all fields are checked for correctness. If the BGP message header checking or OPEN message checking detects an error (see Section 6.2), or a connection collision (see Section 6.8) the local system sends a NOTIFICATION message and changes its state to Idle.

If there are no errors in the OPEN message, BGP sends a KEEPALIVE message and sets a KeepAlive timer. The Hold Timer, which was originally set to a large value (see above), is replaced with the negotiated Hold Time value (see section 4.2). If the negotiated Hold Time value is zero, then the Hold Time timer and KeepAlive timers are not started. If the value of the Autonomous System field is the same as the local Autonomous System number, then the connection is an "internal" connection; otherwise, it is "external". (This will effect UPDATE processing as described below.) Finally, the state is changed to OpenConfirm.

If a disconnect notification is received from the underlying transport protocol, the local system closes the BGP connection, restarts the ConnectRetry timer, while continue listening for connection that may be initiated by the remote BGP peer, and goes into the Active state.

If the Hold Timer expires, the local system sends NOTIFICATION message with error code Hold Timer Expired and changes its state to Idle.

In response to the Stop event (initiated by either system or operator) the local system sends NOTIFICATION message with Error Code Cease and changes its state to Idle.

Start event is ignored in the OpenSent state.

In response to any other event the local system sends NOTIFICATION message with Error Code Finite State Machine Error and changes its state to Idle.

Whenever BGP changes its state from OpenSent to Idle, it closes the BGP (and transport-level) connection and releases all resources associated with that connection.

OpenConfirm state:

In this state BGP waits for a KEEPALIVE or NOTIFICATION message.

If the local system receives a KEEPALIVE message, it changes its state to Established.

If the Hold Timer expires before a KEEPALIVE message is received, the local system sends NOTIFICATION message with error code Hold Timer Expired and changes its state to Idle.

If the local system receives a NOTIFICATION message, it changes its state to Idle.

If the KeepAlive timer expires, the local system sends a KEEPALIVE message and restarts its KeepAlive timer.

If a disconnect notification is received from the underlying transport protocol, the local system changes its state to Idle.

In response to the Stop event (initiated by either system or operator) the local system sends NOTIFICATION message with Error Code Cease and changes its state to Idle.

Start event is ignored in the OpenConfirm state.

In response to any other event the local system sends NOTIFICATION message with Error Code Finite State Machine Error and changes its state to Idle.

Whenever BGP changes its state from OpenConfirm to Idle, it closes the BGP (and transport-level) connection and releases all resources associated with that connection.

Established state:

In the Established state BGP can exchange UPDATE, NOTIFICATION, and KEEPALIVE messages with its peer.

If the local system receives an UPDATE or KEEPALIVE message, it restarts its Hold Timer, if the negotiated Hold Time value is non-zero.

If the local system receives a NOTIFICATION message, it changes its state to Idle.

If the local system receives an UPDATE message and the UPDATE message error handling procedure (see Section 6.3) detects an error, the local system sends a NOTIFICATION message and changes its state to Idle.

If a disconnect notification is received from the underlying transport protocol, the local system changes its state to Idle.

If the Hold Timer expires, the local system sends a NOTIFICATION message with Error Code Hold Timer Expired and changes its state to Idle.

If the KeepAlive timer expires, the local system sends a KEEPALIVE message and restarts its KeepAlive timer.

Each time the local system sends a KEEPALIVE or UPDATE message, it restarts its KeepAlive timer, unless the negotiated Hold Time value is zero.

In response to the Stop event (initiated by either system or operator), the local system sends a NOTIFICATION message with Error Code Cease and changes its state to Idle.

Start event is ignored in the Established state.

In response to any other event, the local system sends NOTIFICATION message with Error Code Finite State Machine Error and changes its state to Idle.

Whenever BGP changes its state from Established to Idle, it closes the BGP (and transport-level) connection, releases all resources associated with that connection, and deletes all

routes derived from that connection.

9. UPDATE Message Handling

An UPDATE message may be received only in the Established state. When an UPDATE message is received, each field is checked for validity as specified in Section 6.3.

If an optional non-transitive attribute is unrecognized, it is quietly ignored. If an optional transitive attribute is unrecognized, the Partial bit (the third high-order bit) in the attribute flags octet is set to 1, and the attribute is retained for propagation to other BGP speakers.

If an optional attribute is recognized, and has a valid value, then, depending on the type of the optional attribute, it is processed locally, retained, and updated, if necessary, for possible propagation to other BGP speakers.

If the UPDATE message contains a non-empty WITHDRAWN ROUTES field, the previously advertised routes whose destinations (expressed as IP prefixes) contained in this field shall be removed from the Adj-RIB-In. This BGP speaker shall run its Decision Process since the previously advertised route is not longer available for use.

If the UPDATE message contains a feasible route, it shall be placed in the appropriate Adj-RIB-In, and the following additional actions shall be taken:

i) If its Network Layer Reachability Information (NLRI) is identical to the one of a route currently stored in the Adj-RIB-In, then the new route shall replace the older route in the Adj-RIB-In, thus implicitly withdrawing the older route from service. The BGP speaker shall run its Decision Process since the older route is no longer available for use.

ii) If the new route is an overlapping route that is included (see 9.1.4) in an earlier route contained in the Adj-RIB-In, the BGP speaker shall run its Decision Process since the more specific route has implicitly made a portion of the less specific route unavailable for use.

iii) If the new route has identical path attributes to an earlier route contained in the Adj-RIB-In, and is more specific (see 9.1.4) than the earlier route, no further actions are necessary.

iv) If the new route has NLRI that is not present in any of the routes currently stored in the Adj-RIB-In, then the new route shall

be placed in the Adj-RIB-In. The BGP speaker shall run its Decision Process.

v) If the new route is an overlapping route that is less specific (see 9.1.4) than an earlier route contained in the Adj-RIB-In, the BGP speaker shall run its Decision Process on the set of destinations described only by the less specific route.

9.1 Decision Process

The Decision Process selects routes for subsequent advertisement by applying the policies in the local Policy Information Base (PIB) to the routes stored in its Adj-RIB-In. The output of the Decision Process is the set of routes that will be advertised to all peers; the selected routes will be stored in the local speaker's Adj-RIB-Out.

The selection process is formalized by defining a function that takes the attribute of a given route as an argument and returns a non-negative integer denoting the degree of preference for the route. The function that calculates the degree of preference for a given route shall not use as its inputs any of the following: the existence of other routes, the non-existence of other routes, or the path attributes of other routes. Route selection then consists of individual application of the degree of preference function to each feasible route, followed by the choice of the one with the highest degree of preference.

The Decision Process operates on routes contained in each Adj-RIB-In, and is responsible for:

- selection of routes to be advertised to BGP speakers located in the local speaker's autonomous system

- selection of routes to be advertised to BGP speakers located in neighboring autonomous systems

- route aggregation and route information reduction

The Decision Process takes place in three distinct phases, each triggered by a different event:

a) Phase 1 is responsible for calculating the degree of preference for each route received from a BGP speaker located in a neighboring autonomous system, and for advertising to the other BGP speakers in the local autonomous system the routes that have the highest degree of preference for each distinct destination.

b) Phase 2 is invoked on completion of phase 1. It is responsible for choosing the best route out of all those available for each distinct destination, and for installing each chosen route into the appropriate Loc-RIB.

c) Phase 3 is invoked after the Loc-RIB has been modified. It is responsible for disseminating routes in the Loc-RIB to each peer located in a neighboring autonomous system, according to the policies contained in the PIB. Route aggregation and information reduction can optionally be performed within this phase.

9.1.1 Phase 1: Calculation of Degree of Preference

The Phase 1 decision function shall be invoked whenever the local BGP speaker receives an UPDATE message from a peer located in a neighboring autonomous system that advertises a new route, a replacement route, or a withdrawn route.

The Phase 1 decision function is a separate process which completes when it has no further work to do.

The Phase 1 decision function shall lock an Adj-RIB-In prior to operating on any route contained within it, and shall unlock it after operating on all new or unfeasible routes contained within it.

For each newly received or replacement feasible route, the local BGP speaker shall determine a degree of preference. If the route is learned from a BGP speaker in the local autonomous system, either the value of the LOCAL_PREF attribute shall be taken as the degree of preference, or the local system shall compute the degree of preference of the route based on preconfigured policy information. If the route is learned from a BGP speaker in a neighboring autonomous system, then the degree of preference shall be computed based on preconfigured policy information. The exact nature of this policy information and the computation involved is a local matter. The local speaker shall then run the internal update process of 9.2.1 to select and advertise the most preferable route.

9.1.2 Phase 2: Route Selection

The Phase 2 decision function shall be invoked on completion of Phase 1. The Phase 2 function is a separate process which completes when it has no further work to do. The Phase 2 process shall consider all routes that are present in the Adj-RIBs-In, including those received from BGP speakers located in its own autonomous system and those received from BGP speakers located in neighboring autonomous systems.

The Phase 2 decision function shall be blocked from running while the Phase 3 decision function is in process. The Phase 2 function shall lock all Adj-RIBs-In prior to commencing its function, and shall unlock them on completion.

If the NEXT_HOP attribute of a BGP route depicts an address to which the local BGP speaker doesn't have a route in its Loc-RIB, the BGP route SHOULD be excluded from the Phase 2 decision function.

For each set of destinations for which a feasible route exists in the Adj-RIBs-In, the local BGP speaker shall identify the route that has:

a) the highest degree of preference of any route to the same set of destinations, or

b) is the only route to that destination, or

c) is selected as a result of the Phase 2 tie breaking rules specified in 9.1.2.1.

The local speaker SHALL then install that route in the Loc-RIB, replacing any route to the same destination that is currently being held in the Loc-RIB. The local speaker MUST determine the immediate next hop to the address depicted by the NEXT_HOP attribute of the selected route by performing a lookup in the IGP and selecting one of the possible paths in the IGP. This immediate next hop MUST be used when installing the selected route in the Loc-RIB. If the route to the address depicted by the NEXT_HOP attribute changes such that the immediate next hop changes, route selection should be recalculated as specified above.

Unfeasible routes shall be removed from the Loc-RIB, and corresponding unfeasible routes shall then be removed from the Adj-RIBs-In.

9.1.2.1 Breaking Ties (Phase 2)

In its Adj-RIBs-In a BGP speaker may have several routes to the same destination that have the same degree of preference. The local speaker can select only one of these routes for inclusion in the associated Loc-RIB. The local speaker considers all equally preferable routes, both those received from BGP speakers located in neighboring autonomous systems, and those received from other BGP speakers located in the local speaker's autonomous system.

The following tie-breaking procedure assumes that for each candidate route all the BGP speakers within an autonomous system can ascertain the cost of a path (interior distance) to the address depicted by the

NEXT_HOP attribute of the route. Ties shall be broken according to the following algorithm:

a) If the local system is configured to take into account MULTI_EXIT_DISC, and the candidate routes differ in their MULTI_EXIT_DISC attribute, select the route that has the lowest value of the MULTI_EXIT_DISC attribute.

b) Otherwise, select the route that has the lowest cost (interior distance) to the entity depicted by the NEXT_HOP attribute of the route. If there are several routes with the same cost, then the tie-breaking shall be broken as follows:

- if at least one of the candidate routes was advertised by the BGP speaker in a neighboring autonomous system, select the route that was advertised by the BGP speaker in a neighboring autonomous system whose BGP Identifier has the lowest value among all other BGP speakers in neighboring autonomous systems;

- otherwise, select the route that was advertised by the BGP speaker whose BGP Identifier has the lowest value.

9.1.3 Phase 3: Route Dissemination

The Phase 3 decision function shall be invoked on completion of Phase 2, or when any of the following events occur:

a) when routes in a Loc-RIB to local destinations have changed

b) when locally generated routes learned by means outside of BGP have changed

c) when a new BGP speaker - BGP speaker connection has been established

The Phase 3 function is a separate process which completes when it has no further work to do. The Phase 3 Routing Decision function shall be blocked from running while the Phase 2 decision function is in process.

All routes in the Loc-RIB shall be processed into a corresponding entry in the associated Adj-RIBs-Out. Route aggregation and information reduction techniques (see 9.2.4.1) may optionally be applied.

For the benefit of future support of inter-AS multicast capabilities, a BGP speaker that participates in inter-AS multicast routing shall advertise a route it receives from one of its external peers and if

it installs it in its Loc-RIB, it shall advertise it back to the peer from which the route was received. For a BGP speaker that does not participate in inter-AS multicast routing such an advertisement is optional. When doing such an advertisement, the NEXT_HOP attribute should be set to the address of the peer. An implementation may also optimize such an advertisement by truncating information in the AS_PATH attribute to include only its own AS number and that of the peer that advertised the route (such truncation requires the ORIGIN attribute to be set to INCOMPLETE). In addition an implementation is not required to pass optional or discretionary path attributes with such an advertisement.

When the updating of the Adj-RIBs-Out and the Forwarding Information Base (FIB) is complete, the local BGP speaker shall run the external update process of 9.2.2.

9.1.4 Overlapping Routes

A BGP speaker may transmit routes with overlapping Network Layer Reachability Information (NLRI) to another BGP speaker. NLRI overlap occurs when a set of destinations are identified in non-matching multiple routes. Since BGP encodes NLRI using IP prefixes, overlap will always exhibit subset relationships. A route describing a smaller set of destinations (a longer prefix) is said to be more specific than a route describing a larger set of destinations (a shorted prefix); similarly, a route describing a larger set of destinations (a shorter prefix) is said to be less specific than a route describing a smaller set of destinations (a longer prefix).

The precedence relationship effectively decomposes less specific routes into two parts:

- a set of destinations described only by the less specific route, and

- a set of destinations described by the overlap of the less specific and the more specific routes

When overlapping routes are present in the same Adj-RIB-In, the more specific route shall take precedence, in order from more specific to least specific.

The set of destinations described by the overlap represents a portion of the less specific route that is feasible, but is not currently in use. If a more specific route is later withdrawn, the set of destinations described by the overlap will still be reachable using the less specific route.

If a BGP speaker receives overlapping routes, the Decision Process shall take into account the semantics of the overlapping routes. In particular, if a BGP speaker accepts the less specific route while rejecting the more specific route from the same peer, then the destinations represented by the overlap may not forward along the ASs listed in the AS_PATH attribute of that route. Therefore, a BGP speaker has the following choices:

a) Install both the less and the more specific routes

b) Install the more specific route only

c) Install the non-overlapping part of the less specific route only (that implies de-aggregation)

d) Aggregate the two routes and install the aggregated route

e) Install the less specific route only

f) Install neither route

If a BGP speaker chooses e), then it should add ATOMIC_AGGREGATE attribute to the route. A route that carries ATOMIC_AGGREGATE attribute can not be de-aggregated. That is, the NLRI of this route can not be made more specific. Forwarding along such a route does not guarantee that IP packets will actually traverse only ASs listed in the AS_PATH attribute of the route. If a BGP speaker chooses a), it must not advertise the more general route without the more specific route.

9.2 Update-Send Process

The Update-Send process is responsible for advertising UPDATE messages to all peers. For example, it distributes the routes chosen by the Decision Process to other BGP speakers which may be located in either the same autonomous system or a neighboring autonomous system. rules for information exchange between BGP speakers located in different autonomous systems are given in 9.2.2; rules for information exchange between BGP speakers located in the same autonomous system are given in 9.2.1.

Distribution of routing information between a set of BGP speakers, all of which are located in the same autonomous system, is referred to as internal distribution.

9.2.1 Internal Updates

The Internal update process is concerned with the distribution of routing information to BGP speakers located in the local speaker's autonomous system.

When a BGP speaker receives an UPDATE message from another BGP speaker located in its own autonomous system, the receiving BGP speaker shall not re-distribute the routing information contained in that UPDATE message to other BGP speakers located in its own autonomous system.

When a BGP speaker receives a new route from a BGP speaker in a neighboring autonomous system, it shall advertise that route to all other BGP speakers in its autonomous system by means of an UPDATE message if any of the following conditions occur:

1) the degree of preference assigned to the newly received route by the local BGP speaker is higher than the degree of preference that the local speaker has assigned to other routes that have been received from BGP speakers in neighboring autonomous systems, or

2) there are no other routes that have been received from BGP speakers in neighboring autonomous systems, or

3) the newly received route is selected as a result of breaking a tie between several routes which have the highest degree of preference, and the same destination (the tie-breaking procedure is specified in 9.2.1.1).

When a BGP speaker receives an UPDATE message with a non-empty WITHDRAWN ROUTES field, it shall remove from its Adj-RIB-In all routes whose destinations was carried in this field (as IP prefixes). The speaker shall take the following additional steps:

1) if the corresponding feasible route had not been previously advertised, then no further action is necessary

2) if the corresponding feasible route had been previously advertised, then:

i) if a new route is selected for advertisement that has the same Network Layer Reachability Information as the unfeasible routes, then the local BGP speaker shall advertise the replacement route

ii) if a replacement route is not available for advertisement, then the BGP speaker shall include the destinations of the

unfeasible route (in form of IP prefixes) in the WITHDRAWN ROUTES field of an UPDATE message, and shall send this message to each peer to whom it had previously advertised the corresponding feasible route.

All feasible routes which are advertised shall be placed in the appropriate Adj-RIBs-Out, and all unfeasible routes which are advertised shall be removed from the Adj-RIBs-Out.

9.2.1.1 Breaking Ties (Internal Updates)

If a local BGP speaker has connections to several BGP speakers in neighboring autonomous systems, there will be multiple Adj-RIBs-In associated with these peers. These Adj-RIBs-In might contain several equally preferable routes to the same destination, all of which were advertised by BGP speakers located in neighboring autonomous systems. The local BGP speaker shall select one of these routes according to the following rules:

a) If the candidate route differ only in their NEXT_HOP and MULTI_EXIT_DISC attributes, and the local system is configured to take into account MULTI_EXIT_DISC attribute, select the routes that has the lowest value of the MULTI_EXIT_DISC attribute.

b) If the local system can ascertain the cost of a path to the entity depicted by the NEXT_HOP attribute of the candidate route, select the route with the lowest cost.

c) In all other cases, select the route that was advertised by the BGP speaker whose BGP Identifier has the lowest value.

9.2.2 External Updates

The external update process is concerned with the distribution of routing information to BGP speakers located in neighboring autonomous systems. As part of Phase 3 route selection process, the BGP speaker has updated its Adj-RIBs-Out and its Forwarding Table. All newly installed routes and all newly unfeasible routes for which there is no replacement route shall be advertised to BGP speakers located in neighboring autonomous systems by means of UPDATE message.

Any routes in the Loc-RIB marked as unfeasible shall be removed. Changes to the reachable destinations within its own autonomous system shall also be advertised in an UPDATE message.

9.2.3 Controlling Routing Traffic Overhead

The BGP protocol constrains the amount of routing traffic (that is, UPDATE messages) in order to limit both the link bandwidth needed to advertise UPDATE messages and the processing power needed by the Decision Process to digest the information contained in the UPDATE messages.

9.2.3.1 Frequency of Route Advertisement

The parameter MinRouteAdvertisementInterval determines the minimum amount of time that must elapse between advertisement of routes to a particular destination from a single BGP speaker. This rate limiting procedure applies on a per-destination basis, although the value of MinRouteAdvertisementInterval is set on a per BGP peer basis.

Two UPDATE messages sent from a single BGP speaker that advertise feasible routes to some common set of destinations received from BGP speakers in neighboring autonomous systems must be separated by at least MinRouteAdvertisementInterval. Clearly, this can only be achieved precisely by keeping a separate timer for each common set of destinations. This would be unwarranted overhead. Any technique which ensures that the interval between two UPDATE messages sent from a single BGP speaker that advertise feasible routes to some common set of destinations received from BGP speakers in neighboring autonomous systems will be at least MinRouteAdvertisementInterval, and will also ensure a constant upper bound on the interval is acceptable.

Since fast convergence is needed within an autonomous system, this procedure does not apply for routes receives from other BGP speakers in the same autonomous system. To avoid long-lived black holes, the procedure does not apply to the explicit withdrawal of unfeasible routes (that is, routes whose destinations (expressed as IP prefixes) are listed in the WITHDRAWN ROUTES field of an UPDATE message).

This procedure does not limit the rate of route selection, but only the rate of route advertisement. If new routes are selected multiple times while awaiting the expiration of MinRouteAdvertisementInterval, the last route selected shall be advertised at the end of MinRouteAdvertisementInterval.

9.2.3.2 Frequency of Route Origination

The parameter MinASOriginationInterval determines the minimum amount of time that must elapse between successive advertisements of UPDATE messages that report changes within the advertising BGP speaker’s own autonomous systems.

9.2.3.3 Jitter

To minimize the likelihood that the distribution of BGP messages by a given BGP speaker will contain peaks, jitter should be applied to the timers associated with MinASOriginationInterval, Keepalive, and MinRouteAdvertisementInterval. A given BGP speaker shall apply the same jitter to each of these quantities regardless of the destinations to which the updates are being sent; that is, jitter will not be applied on a "per peer" basis.

The amount of jitter to be introduced shall be determined by multiplying the base value of the appropriate timer by a random factor which is uniformly distributed in the range from 0.75 to 1.0.

9.2.4 Efficient Organization of Routing Information

Having selected the routing information which it will advertise, a BGP speaker may avail itself of several methods to organize this information in an efficient manner.

9.2.4.1 Information Reduction

Information reduction may imply a reduction in granularity of policy control - after information is collapsed, the same policies will apply to all destinations and paths in the equivalence class.

The Decision Process may optionally reduce the amount of information that it will place in the Adj-RIBs-Out by any of the following methods:

a) Network Layer Reachability Information (NLRI):

Destination IP addresses can be represented as IP address prefixes. In cases where there is a correspondence between the address structure and the systems under control of an autonomous system administrator, it will be possible to reduce the size of the NLRI carried in the UPDATE messages.

b) AS_PATHs:

AS path information can be represented as ordered AS_SEQUENCEs or unordered AS_SETs. AS_SETs are used in the route aggregation algorithm described in 9.2.4.2. They reduce the size of the AS_PATH information by listing each AS number only once, regardless of how many times it may have appeared in multiple AS_PATHs that were aggregated.

An AS_SET implies that the destinations listed in the NLRI can be reached through paths that traverse at least some of the constituent autonomous systems. AS_SETs provide sufficient information to avoid routing information looping; however their use may prune potentially feasible paths, since such paths are no longer listed individually as in the form of AS_SEQUENCEs. In practice this is not likely to be a problem, since once an IP packet arrives at the edge of a group of autonomous systems, the BGP speaker at that point is likely to have more detailed path information and can distinguish individual paths to destinations.

9.2.4.2 Aggregating Routing Information

Aggregation is the process of combining the characteristics of several different routes in such a way that a single route can be advertised. Aggregation can occur as part of the decision process to reduce the amount of routing information that will be placed in the Adj-RIBs-Out.

Aggregation reduces the amount of information that a BGP speaker must store and exchange with other BGP speakers. Routes can be aggregated by applying the following procedure separately to path attributes of like type and to the Network Layer Reachability Information.

Routes that have the following attributes shall not be aggregated unless the corresponding attributes of each route are identical: MULTI_EXIT_DISC, NEXT_HOP.

Path attributes that have different type codes can not be aggregated together. Path of the same type code may be aggregated, according to the following rules:

ORIGIN attribute: If at least one route among routes that are aggregated has ORIGIN with the value INCOMPLETE, then the aggregated route must have the ORIGIN attribute with the value INCOMPLETE. Otherwise, if at least one route among routes that are aggregated has ORIGIN with the value EGP, then the aggregated route must have the origin attribute with the value EGP. In all other case the value of the ORIGIN attribute of the aggregated route is INTERNAL.

AS_PATH attribute: If routes to be aggregated have identical AS_PATH attributes, then the aggregated route has the same AS_PATH attribute as each individual route.

For the purpose of aggregating AS_PATH attributes we model each AS within the AS_PATH attribute as a tuple <type, value>, where "type" identifies a type of the path segment the AS belongs to

(e.g. AS_SEQUENCE, AS_SET), and "value" is the AS number. If the routes to be aggregated have different AS_PATH attributes, then the aggregated AS_PATH attribute shall satisfy all of the following conditions:

- all tuples of the type AS_SEQUENCE in the aggregated AS_PATH shall appear in all of the AS_PATH in the initial set of routes to be aggregated.

- all tuples of the type AS_SET in the aggregated AS_PATH shall appear in at least one of the AS_PATH in the initial set (they may appear as either AS_SET or AS_SEQUENCE types).

- for any tuple X of the type AS_SEQUENCE in the aggregated AS_PATH which precedes tuple Y in the aggregated AS_PATH, X precedes Y in each AS_PATH in the initial set which contains Y, regardless of the type of Y.

- No tuple with the same value shall appear more than once in the aggregated AS_PATH, regardless of the tuple's type.

An implementation may choose any algorithm which conforms to these rules. At a minimum a conformant implementation shall be able to perform the following algorithm that meets all of the above conditions:

- determine the longest leading sequence of tuples (as defined above) common to all the AS_PATH attributes of the routes to be aggregated. Make this sequence the leading sequence of the aggregated AS_PATH attribute.

- set the type of the rest of the tuples from the AS_PATH attributes of the routes to be aggregated to AS_SET, and append them to the aggregated AS_PATH attribute.

- if the aggregated AS_PATH has more than one tuple with the same value (regardless of tuple's type), eliminate all, but one such tuple by deleting tuples of the type AS_SET from the aggregated AS_PATH attribute.

Appendix 6, section 6.8 presents another algorithm that satisfies the conditions and allows for more complex policy configurations.

ATOMIC_AGGREGATE: If at least one of the routes to be aggregated has ATOMIC_AGGREGATE path attribute, then the aggregated route shall have this attribute as well.

AGGREGATOR: All AGGREGATOR attributes of all routes to be aggregated should be ignored.

9.3 Route Selection Criteria

Generally speaking, additional rules for comparing routes among several alternatives are outside the scope of this document. There are two exceptions:

- If the local AS appears in the AS path of the new route being considered, then that new route cannot be viewed as better than any other route. If such a route were ever used, a routing loop would result.

- In order to achieve successful distributed operation, only routes with a likelihood of stability can be chosen. Thus, an AS must avoid using unstable routes, and it must not make rapid spontaneous changes to its choice of route. Quantifying the terms "unstable" and "rapid" in the previous sentence will require experience, but the principle is clear.

9.4 Originating BGP routes

A BGP speaker may originate BGP routes by injecting routing information acquired by some other means (e.g. via an IGP) into BGP. A BGP speaker that originates BGP routes shall assign the degree of preference to these routes by passing them through the Decision Process (see Section 9.1). These routes may also be distributed to other BGP speakers within the local AS as part of the Internal update process (see Section 9.2.1). The decision whether to distribute non-BGP acquired routes within an AS via BGP or not depends on the environment within the AS (e.g. type of IGP) and should be controlled via configuration.

Appendix 1. BGP FSM State Transitions and Actions.

This Appendix discusses the transitions between states in the BGP FSM in response to BGP events. The following is the list of these states and events when the negotiated Hold Time value is non-zero.

BGP States:

1 - Idle
2 - Connect
3 - Active
4 - OpenSent
5 - OpenConfirm
6 - Established

BGP Events:

1 - BGP Start
2 - BGP Stop
3 - BGP Transport connection open
4 - BGP Transport connection closed
5 - BGP Transport connection open failed
6 - BGP Transport fatal error
7 - ConnectRetry timer expired
8 - Hold Timer expired
9 - KeepAlive timer expired
10 - Receive OPEN message
11 - Receive KEEPALIVE message
12 - Receive UPDATE messages
13 - Receive NOTIFICATION message

The following table describes the state transitions of the BGP FSM and the actions triggered by these transitions.

Event	Actions	Message Sent	Next State
Idle (1)			
1	Initialize resources Start ConnectRetry timer Initiate a transport connection	none	2
others	none	none	1
Connect(2)			
1	none	none	2
3	Complete initialization Clear ConnectRetry timer	OPEN	4
5	Restart ConnectRetry timer	none	3
7	Restart ConnectRetry timer Initiate a transport connection	none	2
others	Release resources	none	1
Active (3)			
1	none	none	3
3	Complete initialization Clear ConnectRetry timer	OPEN	4
5	Close connection Restart ConnectRetry timer		3
7	Restart ConnectRetry timer Initiate a transport connection	none	2
others	Release resources	none	1
OpenSent(4)			
1	none	none	4
4	Close transport connection Restart ConnectRetry timer	none	3
6	Release resources	none	1
10	Process OPEN is OK	KEEPALIVE	5
	Process OPEN failed	NOTIFICATION	1
others	Close transport connection Release resources	NOTIFICATION	1

OpenConfirm (5)			
1	none	none	5
4	Release resources	none	1
6	Release resources	none	1
9	Restart KeepAlive timer	KEEPALIVE	5
11	Complete initialization Restart Hold Timer	none	6
13	Close transport connection Release resources		1
others	Close transport connection Release resources	NOTIFICATION	1
Established (6)			
1	none	none	6
4	Release resources	none	1
6	Release resources	none	1
9	Restart KeepAlive timer	KEEPALIVE	6
11	Restart Hold Timer	KEEPALIVE	6
12	Process UPDATE is OK	UPDATE	6
	Process UPDATE failed	NOTIFICATION	1
13	Close transport connection Release resources		1
others	Close transport connection Release resources	NOTIFICATION	1

The following is a condensed version of the above state transition table.

Events	Idle (1)	Connect (2)	Active (3)	OpenSent (4)	OpenConfirm (5)	Estab (6)
1	2	2	3	4	5	6
2	1	1	1	1	1	1
3	1	4	4	1	1	1
4	1	1	1	3	1	1
5	1	3	3	1	1	1
6	1	1	1	1	1	1
7	1	2	2	1	1	1
8	1	1	1	1	1	1
9	1	1	1	1	5	6
10	1	1	1	1 or 5	1	1
11	1	1	1	1	6	6
12	1	1	1	1	1	1 or 6
13	1	1	1	1	1	1

Appendix 2. Comparison with RFC1267

BGP-4 is capable of operating in an environment where a set of reachable destinations may be expressed via a single IP prefix. The concept of network classes, or subnetting is foreign to BGP-4. To accommodate these capabilities BGP-4 changes semantics and encoding associated with the AS_PATH attribute. New text has been added to define semantics associated with IP prefixes. These abilities allow BGP-4 to support the proposed supernetting scheme [9].

To simplify configuration this version introduces a new attribute, LOCAL_PREF, that facilitates route selection procedures.

The INTER_AS_METRIC attribute has been renamed to be MULTI_EXIT_DISC. A new attribute, ATOMIC_AGGREGATE, has been introduced to insure that certain aggregates are not de-aggregated. Another new attribute, AGGREGATOR, can be added to aggregate routes in order to advertise which AS and which BGP speaker within that AS caused the aggregation.

To insure that Hold Timers are symmetric, the Hold Time is now negotiated on a per-connection basis. Hold Times of zero are now supported.

Appendix 3. Comparison with RFC 1163

All of the changes listed in Appendix 2, plus the following.

To detect and recover from BGP connection collision, a new field (BGP Identifier) has been added to the OPEN message. New text (Section 6.8) has been added to specify the procedure for detecting and recovering from collision.

The new document no longer restricts the border router that is passed in the NEXT_HOP path attribute to be part of the same Autonomous System as the BGP Speaker.

New document optimizes and simplifies the exchange of the information about previously reachable routes.

Appendix 4. Comparison with RFC 1105

All of the changes listed in Appendices 2 and 3, plus the following.

Minor changes to the RFC1105 Finite State Machine were necessary to accommodate the TCP user interface provided by 4.3 BSD.

The notion of Up/Down/Horizontal relations present in RFC1105 has been removed from the protocol.

The changes in the message format from RFC1105 are as follows:

1. The Hold Time field has been removed from the BGP header and added to the OPEN message.

2. The version field has been removed from the BGP header and added to the OPEN message.

3. The Link Type field has been removed from the OPEN message.

4. The OPEN CONFIRM message has been eliminated and replaced with implicit confirmation provided by the KEEPALIVE message.

5. The format of the UPDATE message has been changed significantly. New fields were added to the UPDATE message to support multiple path attributes.

6. The Marker field has been expanded and its role broadened to support authentication.

Note that quite often BGP, as specified in RFC 1105, is referred to as BGP-1, BGP, as specified in RFC 1163, is referred to as BGP-2, BGP, as specified in RFC1267 is referred to as BGP-3, and BGP, as specified in this document is referred to as BGP-4.

Appendix 5. TCP options that may be used with BGP

If a local system TCP user interface supports TCP PUSH function, then each BGP message should be transmitted with PUSH flag set. Setting PUSH flag forces BGP messages to be transmitted promptly to the receiver.

If a local system TCP user interface supports setting precedence for TCP connection, then the BGP transport connection should be opened with precedence set to Internetwork Control (110) value (see also [6]).

Appendix 6. Implementation Recommendations

This section presents some implementation recommendations.

6.1 Multiple Networks Per Message

The BGP protocol allows for multiple address prefixes with the same AS path and next-hop gateway to be specified in one message. Making use of this capability is highly recommended. With one address prefix per message there is a substantial increase in overhead in the receiver. Not only does the system overhead increase due to the reception of multiple messages, but the overhead of scanning the routing table for updates to BGP peers and other routing protocols (and sending the associated messages) is incurred multiple times as well. One method of building messages containing many address prefixes per AS path and gateway from a routing table that is not organized per AS path is to build many messages as the routing table is scanned. As each address prefix is processed, a message for the associated AS path and gateway is allocated, if it does not exist, and the new address prefix is added to it. If such a message exists, the new address prefix is just appended to it. If the message lacks the space to hold the new address prefix, it is transmitted, a new message is allocated, and the new address prefix is inserted into the new message. When the entire routing table has been scanned, all

allocated messages are sent and their resources released. Maximum compression is achieved when all the destinations covered by the address prefixes share a gateway and common path attributes, making it possible to send many address prefixes in one 4096-byte message.

When peering with a BGP implementation that does not compress multiple address prefixes into one message, it may be necessary to take steps to reduce the overhead from the flood of data received when a peer is acquired or a significant network topology change occurs. One method of doing this is to limit the rate of updates. This will eliminate the redundant scanning of the routing table to provide flash updates for BGP peers and other routing protocols. A disadvantage of this approach is that it increases the propagation latency of routing information. By choosing a minimum flash update interval that is not much greater than the time it takes to process the multiple messages this latency should be minimized. A better method would be to read all received messages before sending updates.

6.2 Processing Messages on a Stream Protocol

BGP uses TCP as a transport mechanism. Due to the stream nature of TCP, all the data for received messages does not necessarily arrive at the same time. This can make it difficult to process the data as messages, especially on systems such as BSD Unix where it is not possible to determine how much data has been received but not yet processed.

One method that can be used in this situation is to first try to read just the message header. For the KEEPALIVE message type, this is a complete message; for other message types, the header should first be verified, in particular the total length. If all checks are successful, the specified length, minus the size of the message header is the amount of data left to read. An implementation that would "hang" the routing information process while trying to read from a peer could set up a message buffer (4096 bytes) per peer and fill it with data as available until a complete message has been received.

6.3 Reducing route flapping

To avoid excessive route flapping a BGP speaker which needs to withdraw a destination and send an update about a more specific or less specific route shall combine them into the same UPDATE message.

6.4 BGP Timers

BGP employs five timers: ConnectRetry, Hold Time, KeepAlive, MinASOriginationInterval, and MinRouteAdvertisementInterval The suggested value for the ConnectRetry timer is 120 seconds. The suggested value for the Hold Time is 90 seconds. The suggested value for the KeepAlive timer is 30 seconds. The suggested value for the MinASOriginationInterval is 15 seconds. The suggested value for the MinRouteAdvertisementInterval is 30 seconds.

An implementation of BGP MUST allow these timers to be configurable.

6.5 Path attribute ordering

Implementations which combine update messages as described above in 6.1 may prefer to see all path attributes presented in a known order. This permits them to quickly identify sets of attributes from different update messages which are semantically identical. To facilitate this, it is a useful optimization to order the path attributes according to type code. This optimization is entirely optional.

6.6 AS_SET sorting

Another useful optimization that can be done to simplify this situation is to sort the AS numbers found in an AS_SET. This optimization is entirely optional.

6.7 Control over version negotiation

Since BGP-4 is capable of carrying aggregated routes which cannot be properly represented in BGP-3, an implementation which supports BGP-4 and another BGP version should provide the capability to only speak BGP-4 on a per-peer basis.

6.8 Complex AS_PATH aggregation

An implementation which chooses to provide a path aggregation algorithm which retains significant amounts of path information may wish to use the following procedure:

> For the purpose of aggregating AS_PATH attributes of two routes, we model each AS as a tuple <type, value>, where "type" identifies a type of the path segment the AS belongs to (e.g. AS_SEQUENCE, AS_SET), and "value" is the AS number. Two ASs are said to be the same if their corresponding <type, value> tuples are the same.

The algorithm to aggregate two AS_PATH attributes works as follows:

a) Identify the same ASs (as defined above) within each AS_PATH attribute that are in the same relative order within both AS_PATH attributes. Two ASs, X and Y, are said to be in the same order if either:

- X precedes Y in both AS_PATH attributes, or - Y precedes X in both AS_PATH attributes.

b) The aggregated AS_PATH attribute consists of ASs identified in (a) in exactly the same order as they appear in the AS_PATH attributes to be aggregated. If two consecutive ASs identified in (a) do not immediately follow each other in both of the AS_PATH attributes to be aggregated, then the intervening ASs (ASs that are between the two consecutive ASs that are the same) in both attributes are combined into an AS_SET path segment that consists of the intervening ASs from both AS_PATH attributes; this segment is then placed in between the two consecutive ASs identified in (a) of the aggregated attribute. If two consecutive ASs identified in (a) immediately follow each other in one attribute, but do not follow in another, then the intervening ASs of the latter are combined into an AS_SET path segment; this segment is then placed in between the two consecutive ASs identified in (a) of the aggregated attribute.

If as a result of the above procedure a given AS number appears more than once within the aggregated AS_PATH attribute, all, but the last instance (rightmost occurrence) of that AS number should be removed from the aggregated AS_PATH attribute.

References

[1] Mills, D., "Exterior Gateway Protocol Formal Specification", RFC 904, BBN, April 1984.

[2] Rekhter, Y., "EGP and Policy Based Routing in the New NSFNET Backbone", RFC 1092, T.J. Watson Research Center, February 1989.

[3] Braun, H-W., "The NSFNET Routing Architecture", RFC 1093, MERIT/NSFNET Project, February 1989.

[4] Postel, J., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", STD 7, RFC 793, DARPA, September 1981.

[5] Rekhter, Y., and P. Gross, "Application of the Border Gateway Protocol in the Internet", RFC 1772, T.J. Watson Research Center, IBM Corp., MCI, March 1995.

[6] Postel, J., "Internet Protocol - DARPA Internet Program Protocol Specification", STD 5, RFC 791, DARPA, September 1981.

[7] "Information Processing Systems - Telecommunications and Information Exchange between Systems - Protocol for Exchange of Inter-domain Routeing Information among Intermediate Systems to Support Forwarding of ISO 8473 PDUs", ISO/IEC IS10747, 1993

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

[9] Rekhter, Y., Li, T., "An Architecture for IP Address Allocation with CIDR", RFC 1518, T.J. Watson Research Center, cisco, September 1993

Security Considerations

Security issues are not discussed in this document.

Editors' Addresses

Yakov Rekhter
T.J. Watson Research Center IBM Corporation
P.O. Box 704, Office H3-D40
Yorktown Heights, NY 10598

Phone: +1 914 784 7361
EMail: yakov@watson.ibm.com

Tony Li
cisco Systems, Inc.
170 W. Tasman Dr.
San Jose, CA 95134

EMail: tli@cisco.com