Network Working Group J. Honig, Cornell Univ. Theory Center
Request for Comments: 1164 D. Katz, Merit/NSFNET
M. Mathis, Pittsburgh Supercomputing Center
Y. Rekhter, T.J. Watson Research Center, IBM Corp
J. Yu, Merit/NSFNET
June 1990

Application of the Border Gateway Protocol in the Internet

Status of this Memo

This RFC, together with its companion RFC-1163, "A Border Gateway Protocol (BGP)", define a Proposed Standard for an inter-autonomous system routing protocol for the Internet.

This protocol, like any other at this initial stage, may undergo modifications before reaching full Internet Standard status as a result of deployment experience. Implementers are encouraged to track the progress of this or any protocol as it moves through the standardization process, and to report their own experience with the protocol.

This protocol is being considered by the Interconnectivity Working Group (IWG) of the Internet Engineering Task Force (IETF). Information about the progress of BGP can be monitored and/or reported on the IWG mailing list (IWG@nri.reston.va.us).

Please refer to the latest edition of the "IAB Official Protocol Standards" RFC for current information on the state and status of standard Internet protocols.

Distribution of this memo is unlimited.

Table of Contents

1. Acknowledgements

The authors would like to thank Guy Almes (Rice University), Kirk Lougheed (cisco Systems), Hans-Werner Braun (Merit/NSFNET), Sue Hares (Merit/NSFNET), and the Interconnectivity Working Group of the Internet Engineering Task Force (chaired by Guy Almes) for their contributions to this paper.

2. Introduction

The Border Gateway Protocol (BGP), described in RFC 1163, is an interdomain routing protocol. The network reachability information exchanged via BGP provides sufficient information to detect routing loops and enforce routing decisions based on performance preference and policy constraints as outlined in RFC 1104 [2].

This memo uses the term "Autonomous System" 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 networks are reachable through it. From the standpoint of exterior routing, an AS can be viewed as monolithic: reachability to networks directly connected to the AS must be equivalent from all border gateways of the AS.

This paper discusses the use of BGP in the Internet environment. Issues such as topology, the interaction between BGP and IGPs, and the enforcement of policy rules with BGP will be presented.

All of the discussions in this paper are based on the assumption that the Internet is a collection of arbitrarily connected Autonomous Systems. The AS is assumed to be administered by a single administrative entity, at least for the purposes of representation of routing information to systems outside of the AS.

3. BGP Theory and Application

3.1 Topological Model

We will be concerned throughout this paper with a general graph whose nodes are ASs and whose edges are connections between pairs of ASs. The notion of AS is discussed above in Section 2. When we say that a connection exists between two ASs, we mean both of two things:

physical connection: there is a shared network between the two ASs, and on this shared network each AS has at least one border gateway belonging to that AS. Thus the border gateway of each AS can forward packets to the border gateway of the other AS without resort to Inter-AS or Intra-AS routing.

BGP connection: there is a BGP session between BGP speakers on each of the ASs, and this session communicates to each connected AS those routes through the physically connected border gateways of the other AS that can be used for specific networks. Throughout this document we place an additional restriction on the BGP speakers that form the BGP connection: they must themselves share the same network that their border gateways share. Thus, a BGP session between the adjacent ASs requires no support from either Inter-AS or Intra-AS routing. Cases that do not conform to this restriction fall outside the scope of this document.

Thus, at each connection, each AS has one or more BGP speakers and one or more border gateways, and these BGP speakers and border gateways are all located on a shared network. Only the AS's border gateways on the connection's shared network may be used by that AS's BGP speakers on that shared network in NEXT_HOP attributes in Update messages. Paths announced by a BGP speaker of one AS on a given connection are taken to be feasible for each of the border gateways of the other AS on the same connection. In all BGP usage, we intend that the flow of packets from one AS to the other correspond to advertised AS paths.

Much of the traffic carried within an AS either originates or

terminates at that AS (i.e., either the source IP address or the destination IP address of the IP packet identifies a host on a network directly connected to that AS). Traffic that fits this description is called "local traffic". Traffic that does not fit this description is called "transit traffic". A major goal of BGP usage is to control the flow of transit traffic.

Based on how a particular AS deals with transit traffic, the AS may now be placed into one of the following categories:

stub AS: an AS that has only a single connection to another AS. Naturally, a stub AS only carries local traffic.

multihomed AS: an AS that has more than one connection to other ASs, but refuses to carry transit traffic.

transit AS: an AS that has more than one connection to other ASs and is designed (under certain policy restrictions) to carry both transit and local traffic.

Since a full AS path provides an efficient and straightforward way of suppressing routing loops and eliminates the "count-to-infinity" problem associated with some distance vector algorithms, BGP imposes no topological restrictions on the interconnection of ASs.

3.2 BGP in the Internet

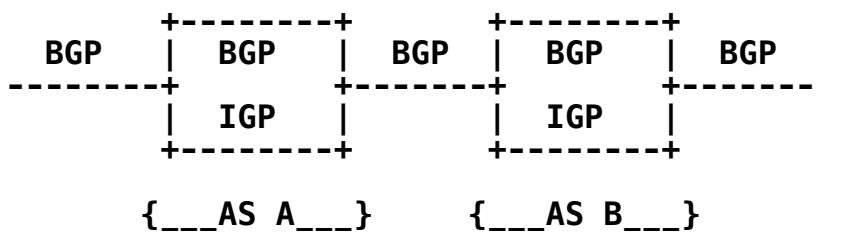
3.2.1 Topology Considerations

The overall Internet topology may be viewed as an arbitrary interconnection of transit, multihomed, and stub ASs. In order to minimize the impact on the current Internet infrastructure, stub and multihomed ASs need not use BGP. These ASs may run other protocols (e.g., EGP) to exchange reachability information with transit ASs. Transit ASs then tag this information as having been learned via EGP or some other method. The fact that BGP need not run on stub or multihomed ASs has no negative impact on the overall quality of inter-AS routing for traffic not local to the stub or multihomed ASs in question.

Of course, BGP may be used for stub and multihomed ASs as well, providing advantage in bandwidth and performance over some of the currently used protocols (such as EGP). In addition, this would result in less need for the use of defaults and in better choices of Inter-AS routes for mulitihomed ASs.

3.2.2 Global Nature of BGP

At a global level, BGP is used to distribute routing information among multiple Autonomous Systems. The information flows can be represented as follows:



This diagram points out that, while BGP alone carries information between ASs, a combination of BGP and an IGP carries information across an AS. Ensuring consistency of routing information between BGP and an IGP within an AS is a significant issue and is discussed at length later in this paper.

3.2.3 BGP Neighbor Relationships

As discussed in the introduction, the Internet is viewed as a set of arbitrarily connected Autonomous Systems (ASs). BGP gateways in each AS communicate with each other to exchange network reachability information based on a set of policies established within each AS. Computers that communicate directly with each other via BGP are known as BGP neighbors. BGP neighbors can be located within the same AS or in different ASs. For the sake of discussion, BGP communications with neighbors in different ASs will be referred to as External BGP, and with neighbors in the same AS as Internal BGP.

External BGP In the case of External BGP, the BGP neighbors must belong to different ASs, but share a common network. This common network should be used to carry the BGP messages between them. The use of BGP across an intervening AS invalidates the AS path information. An Autonomous System number must be used with BGP to specify which Autonomous System the BGP speaker belongs to.

Internal BGP There can be as many BGP gateways as deemed necessary within an AS. Usually, if an AS has multiple connections to other ASs, multiple BGP gateways are needed. All BGP gateways representing the same AS must give a consistent image of the AS to the outside. This requires that the BGP gateways have consistent routing information among them. These gateways can communicate with each other via BGP or by other means. The policy constraints applied to all BGP gateways within an AS must be consistent.

3.3 Policy Making with BGP

BGP provides the capability of enforcing some policies based on various preferences and constraints. Policies are determined by the AS administration and are provided to BGP in the form of configuration information. These policies are enforced within a BGP speaker by affecting the selection of paths from multiple alternatives, and by controlling the redistribution of routing information. Policies are not directly encoded in the protocol.

Non-technical constraints are related to political, security, or economic considerations. For example, if an AS is unwilling to carry traffic to another AS, it can enforce a policy prohibiting this. The following examples of non-technical constraints can be enforced with the use of BGP:

1. A multihomed AS can refuse to act as a transit AS for other ASs. (It does so by not advertising routes to networks other than those directly connected to it.)

2. A multihomed AS can become a transit AS by allowing a certain set of ASs to use it as such. (It does so by advertising routes to networks to this set of ASs.)

3. An AS can favor or disfavor the use of certain ASs for carrying transit traffic from itself to networks advertised with competing AS paths.

A number of performance-related criteria can be controlled with the use of BGP:

1. An AS can minimize the number of transit ASs. (Shorter AS paths can be preferred over longer ones.)

2. The quality of transit ASs. If an AS determines, using BGP, that two or more AS paths can be used to reach a given destination, that AS can use a variety of means to decide which of the candidate AS paths it will use. The quality of an AS can be measured by such things as diameter, link speed, capacity, tendency to become congested, and quality of operation. Information about these qualities might be determined by means other than BGP.

3. Preference of internal routes over external routes.

Non-technical policy will typically override performance issues.

For consistency, combinations of policies and route selection

procedures that might result in equal cost paths must be resolved in a deterministic fashion.

Fundamental to BGP usage is the rule that an AS advertizes to its neighboring ASs only those routes that it uses. This rule reflects the "hop-by-hop" routing paradigm generally used by the current Internet. Note that some policies that cannot be supported by the "hop-by-hop" routing paradigm and which require such techniques as source routing to enforce. For example, BGP does not enable one AS to send traffic to a neighbor AS intending that that traffic take a different route from that taken by traffic originating in the neighbor AS. On the other hand, BGP can support any policy conforming to the "hop-by-hop" routing paradigm. Since the current Internet uses only the "hop-by-hop" routing paradigm and since BGP can support any policy that conforms to that paradigm, BGP is highly applicable as an inter-AS routing protocol for the current Internet.

4. Operational Issues

4.1 Path Selection

One of the major tasks of a BGP speaker for a given AS at a given connection is to evaluate different paths to a destination network from its border gateways at that connection, select the best one, and then advertise it to all of its BGP neighbors at that same connection (subject to policy constraints). The key issue is how different paths are evaluated and compared.

In traditional distance vector protocols (e.g., RIP) there is only one metric (e.g., hop count) associated with a path. As such, comparison of different paths is reduced to simply comparing two numbers. A complication in Inter-AS routing arises from the lack of a universally agreed-upon metric among ASs that can be used to evaluate external paths. Rather, each AS may have its own set of criteria for path evaluation.

A BGP speaker within an Autonomous System builds a routing database consisting of the set of all feasible paths and the list of networks reachable through each path. In an efficient implementation, it will be important to store and process these paths and bundle the networks reachable through them. For purposes of precise discussion, however, it's useful to consider the set of feasible paths for a given destination network. In most cases, we would expect to find only one feasible path in the set. This will often, however, not be the case. All feasible paths must be maintained, and their maintenance speeds adaptation to the loss of the primary path, but only the primary path at any given time will ever be advertised.

The path selection process can be formalized by defining a partial order over the set of all possible paths to a given destination network. One way to define this partial order is to define a function that maps each full AS path to a non-negative integer that denotes the path's degree of preference. Path selection is then reduced to applying this function to all feasible paths and choosing the one with the highest degree of preference.

In actual BGP implementations, criteria for assigning degree of preferences to a path can be specified in a configuration file.

The process of assigning a degree of preference to a path can be based on several sources of information:

1. Information explicitly present in the full AS path.

2. A combination of information that can be derived from the full AS path and information outside the scope of BGP.

The criteria used to assign a degree of preference to a path can be classified as primitive or compound. Possible primitive criteria include:

- AS count. Paths with a smaller AS count are generally better.

- Presence or absence of a certain AS or ASs in the path. By means of information outside the scope of BGP, an AS may know some performance characteristics (e.g., bandwidth, MTU, intra-AS diameter) of certain ASs and may try to avoid or prefer them.

- Path origin. A path whose endpoint is internal to the last AS on the path (BGP is used over the entire path) is generally better than one for which part of the path was learned via EGP or some other means.

- AS path subsets. An AS path that is a subset of a longer AS path to the same destination should be preferred over the longer path. Any problem in the shorter path (such as an outage) will also be a problem in the longer path.

- Link dynamics. Stable paths should be preferred over unstable ones. Note that this criterion must be used in a very careful way to avoid causing unnecessary route fluctuation. Generally, any criteria that depend on dynamic information might cause routing instability and should be treated very carefully.

- Policy consideration. BGP supports policy based routing based

on the policy based distribution of routing information defined in RFC 1104 [2]. A BGP gateway may be aware of some policy constraints (both within and outside of its own AS) and do appropriate path selection. Paths that do not comply with policy requirements are not considered further.

Metrics based on compound criteria can be computed as a weighted combination of the degree of preferences of primitive criteria. The use of compound criteria should be done with extreme caution since it involves comparing potentially uncomparable quantities.

4.2 Syntax and Semantics for BGP Configuration Files

A major task in using BGP is thus to assign a degree of preference to each available AS-path. This degree of preference will generally be a function of the number of ASs in the path, properties of the specific ASs in the path, the origin of the route, and properties of the specific border router to be used in the first hop. In this section we consider how a network administrator might articulate this function by means of a configuration file. In the future, we can imagine using tools based on network management protocols such as SNMP for this task, but the protocols do not currently support this ability.

In addition to controlling the selection of the best path to a given network, the network administrator must control the advertisement of this best path to neighboring ASs. Therefore, path selection and path distribution emerge as the two key aspects of policy expression in BGP usage.

Since different aspects of one AS's policy interact, and since the policies of different ASs interact, it is important to facilitate the analysis of such interactions by means of high-quality and consistent tools.

There is also a need for tools to translate the expression of the network administrator's policy to some technical mechanism within a BGP speaker to implement that policy.

These factors suggest that there should be a globally consistent way of describing policies in the configuration file. The syntax and semantics of these policies should be capable of expressing the path selection phase within the local AS as well as the path redistribution phase to other ASs.

Because it may be desirable to coordinate routing policy at an external level, it may prove worthwhile to create a language to describe this information in a globally consistent way. Policies

expressed in such a language could conceivably be used by some high-level tools to analyze the interaction among the routing policies of different Autonomous Systems.

The following defines one possible syntax and semantics for describing AS path policies from the point of view of the local AS. Alternative syntaxes with equivalent richness of functionality are not precluded. Other mechanisms may be needed to provide a fully functional configuration language.

A complete AS path, supplied by BGP, provides the most important mechanism for policy enforcement. Assigning a degree of preference to a particular AS path can be modelled as a matching between this path and one or more predefined AS path patterns. Each predefined AS path pattern has a degree of preference that will be assigned to any AS path that matches it.

Since patterns are naturally expressed by regular expressions, one can use regular expressions over the alphabet of AS numbers to define AS path patterns and, therefore, to formulate policies.

Since certain constructs occur frequently in regular expressions, the following notational shorthand (operators) is defined:

. matches any AS number. To improve readability, "." can be replaced by "any" so long as this does not introduce ambiguity.

* a regular expression followed by * means zero or more repetitions

+ a regular expression followed by + means one or more repetitions

? a regular expression followed by ? means zero or one repetition

| alternation

() parentheses group subexpressions--an operator, such as * or works on a single element or on a regular expression enclosed in parentheses

{m,n} a regular expression followed by {m,n} (where m and n are both non-negative integers and m <= n) means at least m and at most n repetitions.

{m} a regular expression followed by {m} (where m is a positive integer) means exactly m repetitions.

{m,} a regular expression followed by {m,} (where m is a positive integer) means m or more repetitions.

Any regular expression is generated by these rules.

The Policy Based Routing Language can then be defined as follows:

<Policy-Based-Routing> ::= { <policy-statement> }

Semantics: each policy statement might cause a given possible BGP advertisement (possibility) to be installed into the routing table as the route to a given (set of) networks. Thus, an empty Policy-Based-Routing means that no possibilities will be accepted.

<policy-statement> ::=
<policy-expression> '=' <dop-expression> ';'

Semantics: if a given possibility matches the policy-expression, then that possibility will be accepted with a degree of preference denoted by the integer value dop-expression.

<policy-expression> ::=
<policy-term> |
<policy-term> <policy-operator> <policy-term>

<policy-term> ::=
<network-list> <AS-path> <origin> <distribution-list> |
'(' <policy-expression> ')' |
NOT <policy-expression> |
<>

<policy-operator> ::= OR | AND

Semantics: the intersection of the network list of a possibility and the network-list must be non-empty; the AS-path of the possibility must match the AS-path as a sequence; the origin of the possibility must be a member of the origin set; if these conditions are met, the route denoted by the possibility is accepted as a possible route to those networks of the intersection of the possibility network list and the network-list.

<AS-path> ::= "regular expression over AS numbers"

Semantics: the AS-path of the possibility must be generated by the regular expression <AS-path>.

```
<network-list> ::= '<' { network network-list } '>' |
                   '<' ANY '>'
```

Semantics: A non-empty sequence enumerates the network numbers of the network-list; ANY denotes the set of all network numbers.

```
<origin> ::= IGP | EGP | INCOMPLETE | ANY
```

Semantics: origin enumerates the sequence of acceptable origins; ANY denotes the set of all origins.

```
<distribution-list> ::= '<' { AS } '>' |
                        '<' ANY '>'
```

Semantics: if a given possibility as accepted and installed into the routing table, then distribution-list is the set of (neighboring) autonomous systems to whose border routers we will distribute the BGP-derived routes.

```
<dop-expression> ::= <dop-term> |
                     <dop-term> '+' <dop-term> |
                     <dop-term> '-' <dop-term> |
                     <dop-term> '*' <dop-term> |
                     <dop-term> '/' <dop-term> |
                     REJECT

<dop-term> ::= <integer> |
               <function> |
               '(' <dop-expression> ')'
```

Semantics: if a possibility matches with degree of preference REJECT, then that possibility will not be used. Otherwise, the integer value of the degree of preference indicates the degree of preference of the possibility, with higher values preferred over lower ones.

White spaces can be used between symbols to improve readability. "<>" denotes the empty sequence.

There are two built-in functions, PathLength() and PathWeight(). PathLength() takes the AS path as an argument and returns the number of ASs in that path. PathWeight() takes the AS path and an AS weight table as arguments and returns the sum of weights of the ASs in the AS path as defined by the AS weight table. In order to preserve determinism, the AS weight table must always have a default weight which will be assigned to any AS which is not in that table.

The AS path, as used above, is constructed from right to left which

is consistent with BGP), so that the most recent AS in the path occupies the leftmost position.

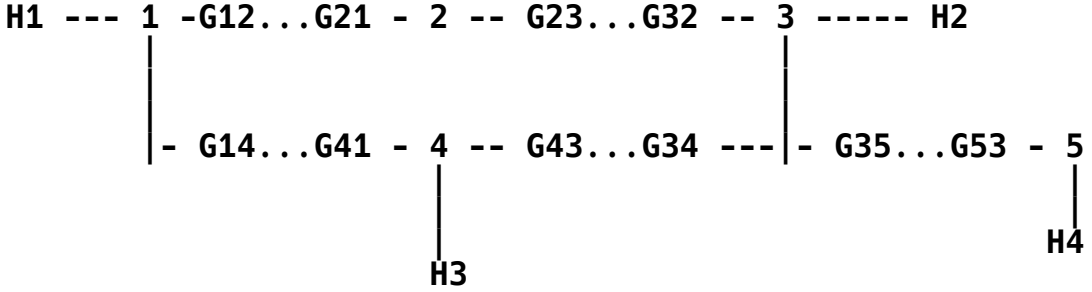
Each network (and its associated complete AS path) received from other BGP neighbors is matched against local Routing Policies.

If either no match occurs or the degree of preference associated with the matched policy is REJECT, then the received information is rejected. Otherwise, a degree of preference associated with the matched policy is assigned to that path. Notice that the process terminates on the first successful match. Therefore, policy-terms should be ordered from more specific to more general.

The semantics of a matched policy is as follows: If a network in <network-list> that was originally introduced into BGP from <origin> is received via <AS-path>, that network should be redistributed to all ASs in <distribution-list>.

The following examples (some taken from RFC 1102 [3]) illustrate how Policy Terms can be written.

In the following topology, H elements are hosts, G elements are Policy Gateways running BGP, and numbered elements are ASs.



In this picture, there are four hosts, ten gateways, and five Autonomous Systems. Gateways G12 and G14 belong to AS 1. Gateways G21 and G23 belong to AS 2. Gateways G41 and G43 belongs to AS 4. Gateways G32, G34, and G35 belong to AS 3. Gateway G53 belongs to AS 5. Dashed lines denote intra-AS connections. Dotted lines denote inter-AS connections.

First, consider AS 2. It has no hosts attached, and models a transit service, such as the NSFNET backbone network. It may have a very simple policy: it will carry any traffic between any two ASs, without further constraint. If AS 1 and AS 3 are neighboring domains, then its policy term could be written as:

```
AS 2: < ANY > < (1 | 3) .* > < IGP > < 1 3 > = 10
```

The first component in this policy, the network list

```
< ANY >
```

says that any network is subject to this policy. The second component, the AS path

```
< (1 | 3) .* >
```

says that routing information that came from either AS 1 or AS 3 matches this policy, including routes from ASs that lie beyond AS 1 and AS 3. The third component, the origin

```
< IGP >
```

says that this route must be interior with respect to the originating AS, implying that routes imported via EGP or some other mechanism would not match this policy. The fourth component, the distribution list

```
< 1  3 >
```

says that this route may be redistributed to both AS 1 and AS 3. Finally, the degree of preference assigned to any route which matches this policy is set to 10.

To improve readability, the above policy can be rewritten as:

```
AS 2: < ANY > < (1 | 3) ANY* > < IGP > < 1  3 > = 10
```

Next, consider AS 3. It is willing to provide transit service to AS 4 and AS 5, presumably due to multilateral agreements. AS 3 should set its policy as follows:

```
AS 3: < ANY > < (4 | 5) > < IGP > < 2  4  5 > = 10
AS 3: < ANY > < 2  .* > < ANY > < 4  5 > = 10
AS 3: < ANY > < 3 > < ANY > < 2  4  5 > = 10
```

This would allow AS 3 to distribute internal routes received from ASs 4 and 5 to ASs 2, 4, and 5, and all backbone routes through AS 2 would be distributed to ASs 4 and 5. AS 3 would advertise its own networks to ASs 2, 4, and 5. Hosts in AS 4 and AS 5 would be able to reach each other, as well as hosts in ASs 1 and 3 and anything beyond them. AS 3 allows any origin in routes from AS 2. This implies that AS 3 trusts AS 2 to impose policy on routes imported by means other than BGP. Note that although the policy statement would appear to allow AS 3 to send ASs 4 and 5 their own routes, the BGP protocol would detect this as a routing loop and prevent it.

Now consider AS 1. AS 1 wishes to use the backbone service provided by AS 2, and is willing to carry transit traffic for AS 4. The policy statements for AS 1 might read:

```
AS 1: < ANY > < 4 > < IGP > < 2 > = 150
AS 1: < ANY > < 2  .* > < ANY > < 4 > = 150
AS 1: < ANY > < 1 > < ANY > < 2  4 > = 150
```

AS 1 will redistribute all routes learned from the AS 2 backbone to AS 4, and vice versa, and distribute routes to its own networks to both AS 2 and AS 4. The degree of preference assigned to any route which matches this policy is set to 150.

AS 5 is a more interesting case. AS 5 wishes to use the backbone service, but is not directly connected to AS 2. Its policy statements could be as follows:

```
AS 5: < ANY > < 3  4 > < IGP > < > = 10
AS 5: < ANY > < 3  2  .* > < . > < > = 10
AS 5: < ANY > < 5 > < . > < 3 > = 10
```

This policy imports routes through AS 2 and AS 3 into AS 5, and allows AS 5 and AS 4 to communicate through AS 3. Since AS 5 does not redistribute any routes other than its own, it is a stub AS. Note that AS 5 does not trust AS 3 to advertise only routes through AS 2, and thus applies its own filter to ensure that it only uses the backbone. This lack of trust makes it necessary to add the second policy term.

AS 4 is a good example of a multihomed AS. AS 4 wishes to use AS 3 as is primary path to the backbone, with AS 1 as a backup. Furthermore, AS 4 does not wish to provide any transit service between ASs 1 and 3. Its policy statement could read:

```
AS 4: < ANY > < 3  .* > < ANY > < > = 10
AS 4: < ANY > < 1  .* > < ANY > < > = 20
AS 4: < ANY > < 4 > < ANY > < 1  3 > = 10
```

Paths to any network through AS 3 are preferred, but AS 1 will be used as a backup if necessary. Note that since AS 4 trusts AS 3 to provide it with reasonable routes, it is not necessary to explicitly import routes from AS 5. Since the redistribution terms are null except for networks within AS 4, AS 4 will never carry any transit traffic.

Given the topology and policies described above, it becomes apparent that two paths of equal preference would be available from AS 2 to any of the networks in AS 4. Since ties are not allowed, an

arbitrary tie-breaking mechanism would come into play (as described above), which might result in less than optimal routes to some networks. An alternative mechanism that would provide optimal routes while still allowing fallback paths would be to provide network-by-network policies in specific cases, and explicit tie-breaking policies for the remaining networks. For example, the policies for AS 2 could be rewritten as follows:

```
AS 2: < 35 > < 1  .* > < IGP > < 3 > = 10
AS 2: < 35 > < 3  .* > < IGP > < 1 > = 20
AS 2: < ANY > < 1  .* > < IGP > < 3 > = 20
AS 2: < ANY > < 3  .* > < IGP > < 1 > = 10
```

Paths to network 35 through AS 1 would be preferred, with AS 3 as a fallback; paths to all other networks through AS 3 would be preferred over those through AS 1. Such optimizations may become arbitrarily complex.

There may be other, simpler ways to assign a degree of preference to an AS path.

The simplest way to assign a degree of preference to a particular path is to use the number of ASs in the AS path as the degree of preference. This approach reflects the heuristic that shorter paths are usually better than longer ones. This policy can be implemented by using the PathLength() built-in function in the following policy statement:

```
< ANY > < .* > < ANY > < ANY > = PathLength(ASpath)
```

This policy assigns to any network with an arbitrary AS path a degree of preference equal to the number of ASs in the AS path; it then redistributes this information to all other BGP speakers. As an example, an AS path which traverses three different Autonomous Systems will be assigned the degree of preference 3.

Another approach is to assign a certain degree of preference to each individual AS, and then determine the degree of preference of a particular AS path as the sum of the degree of preferences of the ASs in that path. Note that this approach does not require the assignment of a specific degree of preference to every AS in the Internet. For ASs with an unknown degree of preference, a default can be used. This policy can be implemented by using the PathWeight() built-in function in the following policy statement:

```
< ANY > < .* > < ANY > < ANY >
                        = PathWeight(ASpath, ASWeightTable)
```

As an example, if Autonomous Systems 145 and 55 have 10 and 15 as their weights in the ASWeightTable, and if the default degree of preference in the ASWeightTable is 50, then an AS path that traverses Autonomous Systems 145, 164, and 55 will be assigned degree of preference 75.

The above examples demonstrate some of the simple policies that can be implemented with BGP. In general, very sophisticated policies based on partial or complete AS path discrimination can be written and enforced. It should be emphasized that movement toward more sophisticated policies will require parallel effort in creating more sophisticated tools for policy interaction analysis.

5. The Interaction of BGP and an IGP

5.1 Overview

By definition, all transit ASs must be able to carry traffic external to that AS (neither the source nor destination host belongs to the AS). This requires a certain degree of interaction and coordination between the Interior Gateway Protocol (IGP) used by that particular AS and BGP. In general, traffic exterior to a given AS is going to pass through both interior gateways (gateways that support IGP only) and border gateways (gateways that support both IGP and BGP). All interior gateways receive information about external routes from one or more of the border gateways of the AS via the IGP.

Depending on the mechanism used to propagate BGP information within a given AS, special care must be taken to ensure consistency between BGP and the IGP, since changes in state are likely to propagate at different rates across the AS. There may be a time window between the moment when some border gateway (A) receives new BGP routing information which was originated from another border gateway (B) within the same AS, and the moment the IGP within this AS is capable of routing transit traffic to that border gateway (B). During that time window, either incorrect routing or "black holes" can occur.

In order to minimize such routing problems, border gateway (A) should not advertise a route to some exterior network X to all of its BGP neighbors in other ASs until all of the interior gateways within the AS are ready to route traffic destined to X via the correct exit border gateway (B). In other words, interior routing should converge on the proper exit gateway before advertising routes via that exit gateway to other ASs.

5.2 Methods for Achieving Stable Interactions

The following discussion outlines several techniques capable of

achieving stable interactions between BGP and the IGP within an Autonomous System.

5.2.1 Propagation of BGP Information via the IGP

While BGP can provide its own mechanism for carrying BGP information within an AS, one can also use an IGP to transport this information, as long as the IGP supports complete flooding of routing information (providing the mechanism to distribute the BGP information) and one-pass convergence (making the mechanism effectively atomic). If an IGP is used to carry BGP information, then the period of desynchronization described earlier does not occur at all, since BGP information propagates within the AS synchronously with the IGP, and the IGP converges more or less simultaneously with the arrival of the new routing information. Note that the IGP only carries BGP information and should not interpret or process this information.

5.2.2 Tagged Interior Gateway Protocol

Certain IGPs can tag routes exterior to an AS with the identity of their exit points while propagating them within the AS. Each border gateway should use identical tags for announcing exterior routing information (received via BGP) both into the IGP and into Internal BGP when propagating this information to other border gateways within the same AS. Tags generated by a border gateway must uniquely identify that particular border gateway--different border gateways must use different tags.

All Border Gateways within a single AS must observe the following two rules:

1. Information received via Internal BGP by a border gateway A declaring a network to be unreachable must immediately be propagated to all of the External BGP neighbors of A.

2. Information received via Internal BGP by a border gateway A about a reachable network X cannot be propagated to any of the External BGP neighbors of A unless/until A has an IGP route to X and both the IGP and the BGP routing information have identical tags.

These rules guarantee that no routing information is announced externally unless the IGP is capable of correctly supporting it. It also avoids some causes of "black holes".

One possible method for tagging BGP and IGP routes within an AS is to use the IP address of the exit border gateway announcing the exterior route into the AS. In this case the "gateway" field in the BGP UPDATE message is used as the tag.

5.2.3 Encapsulation

Encapsulation provides the simplest (in terms of the interaction between the IGP and BGP) mechanism for carrying transit traffic across the AS. In this approach, transit traffic is encapsulated within an IP datagram addressed to the exit gateway. The only requirement imposed on the IGP by this approach is that it should be capable of supporting routing between border gateways within the same AS.

The address of the exit gateway A for some exterior network X is specified in the "gateway" field of the BGP UPDATE message received from gateway A via Internal BGP by all other border gateways within the same AS. In order to route traffic to network X, each border gateway within the AS encapsulates it in datagrams addressed to gateway A. Gateway A then performs decapsulation and forwards the original packet to the proper gateway in another AS.

Since encapsulation does not rely on the IGP to carry exterior routing information, no synchronization between BGP and the IGP is required.

Some means of identifying datagrams containing encapsulated IP, such as an IP protocol type code, must be defined if this method is to be used.

Note, that if a packet to be encapsulated has length that is very close to the MTU, that packet would be fragmented at the gateway that performs encapsulation.

5.2.4 Other Cases

There may be ASs with IGPs which can neither carry BGP information nor tag exterior routes (e.g., RIP). In addition, encapsulation may be either infeasible or undesirable. In such situations, the following two rules must be observed:

1. Information received via Internal BGP by a border gateway A declaring a network to be unreachable must immediately be propagated to all of the External BGP neighbors of A.

2. Information received via Internal BGP by a border gateway A about a reachable network X cannot be propagated to any of the External BGP neighbors of A unless A has an IGP route to X and sufficient time (holddown) has passed for the IGP routes to have converged.

The above rules present necessary (but not sufficient) conditions for propagating BGP routing information to other ASs. In contrast to

tagged IGPs, these rules cannot ensure that interior routes to the proper exit gateways are in place before propagating the routes to other ASs.

If the convergence time of an IGP is less than some small value X, then the time window during which the IGP and BGP are unsynchronized is less than X as well, and the whole issue can be ignored at the cost of transient periods (of less than length X) of routing instability. A reasonable value for X is a matter for further study, but X should probably be less than one second.

If the convergence time of an IGP cannot be ignored, a different approach is needed. Mechanisms and techniques which might be appropriate in this situation are subjects for further study.

6. Implementation Recommendations

6.1 Multiple Networks Per Message

The BGP protocol allows for multiple networks 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 network 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 flash 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 networks 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 network is processed, a message for the associated AS path and gateway is allocated, if it does not exist, and the new network is added to it. If such a message exists, the new network is just appended to it. If the message lacks the space to hold the new network, it is transmitted, a new message is allocated, and the new network 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 networks share a gateway and common path attributes, making it possible to send many networks in one 4096-byte message.

6.2 Preventing Excessive Resource Utilization

When peering with a BGP implementation that does not compress multiple networks 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 rate limit flash 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.

6.3 Processing Messages on a Stream Protocol

Due to the stream nature of TCP, all the data for received messages does not necessarily arrive at the same time, due to the nature of TCP. 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 (1024 bytes) per peer and fill it with data as available until a complete message has been received.

6.4 Processing Update Messages

In BGP, all Update messages are incremental. Once a particular network is listed in an Update message as being reachable through an AS path and gateway, that piece of information is expected to be retained indefinitely. In order for a route to a network to be removed, it must be explicitly listed in an Update message as being unreachable or with new routing information to replace the old. Note that a BGP peer will only advertise one route to a given network, so any announcement of that network by a particular peer replaces any previous information about that network received from the same peer.

This approach has the obvious advantage of low overhead; if all routes are stable, only KeepAlive messages will be sent. There is no periodic flood of route information.

However, this means that a consistent view of routing information between BGP peers is only possible over the course of a single transport connection, since there is no mechanism for a complete update. This requirement is accommodated by specifying that BGP peers must transition to the Idle state upon the failure of a transport connection.

7. Conclusion

The BGP protocol provides a high degree of control and flexibility for doing interdomain routing while enforcing policy and performance constraints and avoiding routing loops. It is hoped that the guidelines presented here will provide a starting point for more sophisticated and manageable routing in the Internet as it grows.

References

[1] Lougheed, K. and Y. Rekhter, "A Border Gateway Protocol", RFC 1163, cisco Systems and IBM Watson Research Center, June 1990.

[2] Braun, H-W., "Models of Policy Based Routing", RFC 1104, Merit/NSFNET, June 1989.

[3] Clark, D., "Policy Routing in Internet Protocols", RFC 1102, M.I.T., May 1989.

Security Considerations

Security issues are not discussed in this memo.

Authors' Addresses

Jeffrey C. Honig
Theory Center
265 Olin Hall
Cornell University
Ithaca, NY 14853-5201

Phone: (607) 255-8686

Email: JCH@TCGOULD.TN.CORNELL.EDU

Dave Katz
Merit/NSFNET
1075 Beal Ave.
Ann Arbor, MI 48109

Phone: (313) 763-4898

Email: DKATZ@MERIT.EDU

Matt Mathis
Pittsburgh Supercomputing Center
4400 Fifth Ave.
Pittsburgh, PA 15213

Phone: (412) 268-3319

Email: MATHIS@FARADAY.ECE.CMU.EDU

Yakov Rekhter
T.J. Watson Research Center
IBM Corporation
P.O. Box 218
Yorktown Heights, NY 10598

Phone: (914) 945-3896

Email: YAKOV@IBM.COM

Jie Yun (Jessica) Yu
Merit/NSFNET
1075 Beal Ave.
Ann Arbor, MI 48109

Phone: (313) 936-3000

Email: JYY@MERIT.EDU